



Agenzia nazionale per le nuove tecnologie,
l'energia e lo sviluppo economico sostenibile



MINISTERO DELLO SVILUPPO ECONOMICO



Ricerca di Sistema elettrico

Metodologie di calcolo del Fabbisogno energetico degli edifici: metodi dinamici. Sviluppo e applicazione di un codice di calcolo I

L. Mazzarella, A. Alongi, A. Angelotti, G. Filippini, A. Gandelli, M. Pasini



POLITECNICO
MILANO 1863

Report RdS/PTR2019/034

METODOLOGIE DI CALCOLO DEL FABBISOGNO ENERGETICO DEGLI EDIFICI: METODI DINAMICI. SVILUPPO E APPLICAZIONE DI UN CODICE DI CALCOLO I

L. Mazzarella, A. Alongi, A. Angelotti, G. Filippini, A. Gandelli, M. Pasini (Politecnico di Milano – Dipartimento di Energia)

Dicembre 2019

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Triennale di Realizzazione 2019-2021 - I annualità

Obiettivo: *N. 1 - Tecnologie*

Progetto: *1.5 - Tecnologie, tecniche e materiali per l'efficienza energetica ed il risparmio di energia negli usi finali elettrici degli edifici nuovi ed esistenti*

Work package: *1 - Edifici ad alta efficienza energetica*

Linea di attività: *LA1.6 - Metodologie di calcolo del Fabbisogno energetico degli edifici: metodi dinamici. Sviluppo e applicazione di un codice di calcolo I*

Responsabile del Progetto: Giovanni Puglisi, ENEA

Responsabile del Work package: Domenico Iatauro, ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione *"Metodologie di calcolo del Fabbisogno energetico degli edifici: metodi dinamici"*

Responsabile scientifico ENEA: Domenico Iatauro

Responsabile scientifico Politecnico di Milano – Dipartimento di Energia: Livio Mazzarella

Indice

SOMMARIO.....	7
INTRODUZIONE.....	9
1 CARATTERISTICHE PRINCIPALI DEGLI ATTUALI CODICI DI SIMULAZIONE DINAMICA	12
1.1 REQUISITI FONDAMENTALI	13
1.1.1 <i>Operatività vs Adeguatezza</i>	13
1.1.2 <i>Valutazione dell'accuratezza</i>	14
1.1.3 <i>Capacità di crescita evolutiva</i>	15
1.2 TIPOLOGIE DI BPST ESISTENTI	16
1.2.1 <i>Special-purpose programs</i>	16
1.2.2 <i>General-purpose programs</i>	17
1.3 INTEROPERABILITÀ E CO-SIMULAZIONE	18
1.3.1 <i>Co-simulazione</i>	19
1.3.2 <i>Modifica del codice</i>	19
1.3.3 <i>Uso inappropriato</i>	20
1.4 ELEMENTI CHIAVE PER LA SCELTA DI UN BPST: LA MODULARITÀ E LA COMMUNITY.....	21
2 INDIVIDUAZIONE DEI REQUISITI PER UN NUOVO CODICE DI CALCOLO.....	22
2.1 PROBLEMI DI COMUNICAZIONE: UNA NUOVA FIGURA PROFESSIONALE INTERDISCIPLINARE	23
2.2 MODULARITÀ DEL SOFTWARE	24
2.3 INTEGRAZIONE NEL BIM	26
2.4 REQUISITI E RUOLO DELL'UTENTE E DEGLI SVILUPPATORI	26
2.5 RESPONSABILITÀ DISTRIBUITE.....	27
2.6 APPROCCIO OPEN SOURCE.....	27
3 CARATTERISTICHE GENERALI DEL NUOVO CODICE: OPENBPS.....	29
3.1 CODICE ORIENTATO AGLI OGGETTI	29
3.2 STRUTTURA DI OPENBPS.....	31
3.2.1 <i>Gestione della simulazione</i>	36
3.2.2 <i>Bilancio energetico e di massa</i>	37
3.2.3 <i>Scambio termico radiativo</i>	38
3.2.4 <i>Simulazione di sistemi impiantistici</i>	39
3.2.5 <i>Dati di input, output e meteo</i>	39
3.2.6 <i>Contribuire ai nuovi sviluppi</i>	40
4 OPENBPS: CARATTERISTICHE SPECIFICHE E MODELLI NUMERICI	41
4.1 DISCRETIZZAZIONE TEMPORALE	41
4.1.1 <i>Convenzione temporale</i>	42
4.1.2 <i>Inizializzazione</i>	42
4.1.3 <i>Passi temporali consentiti</i>	43
4.2 DATI CLIMATICI	44
4.2.1 <i>Formati disponibili</i>	44
4.2.1.1 Formato TMY	45
4.2.1.2 Formato EPW	45
4.2.1.3 Formato WMO.....	45
4.2.1.4 Formato definito dall'utente	46
4.2.2 <i>Gestione dei dati integrali</i>	48
4.2.3 <i>Sincronizzazione dei dati climatici con il tempo della simulazione</i>	49
4.2.3.1 Confronto con EnergyPlus e TRNSYS	51
4.2.3.2 Dettagli su EnergyPlus	52

4.2.3.3	Dettagli su TRNSYS.....	54
4.3	RADIAZIONE SOLARE	55
4.3.1	<i>Nomenclatura specifica e algoritmi base</i>	55
4.3.1.1	Nomenclatura e formule base	55
4.3.1.2	Irradianza solare	58
4.3.2	<i>Inizializzazione – ora di alba e di tramonto</i>	60
4.3.3	<i>Calcolo della posizione del sole: metodo della Serie di Fourier</i>	61
4.3.4	<i>Calcolo della posizione del sole: algoritmo n° 5 di Grena</i>	63
4.3.4.1	Punto di partenza: calcolo della scala temporale	63
4.3.4.2	Calcolo della longitudine eclittica eliocentrica	64
4.3.4.3	Correzione per la nutazione.....	64
4.3.4.4	Calcolo dell'angolo azimutale e zenitale.....	66
4.3.5	<i>Calcolo della posizione del sole: metodo dell'Almanacco Nautico</i>	66
4.3.5.1	Coordinate eclittiche	67
4.3.5.2	Coordinate celesti.....	67
4.3.5.3	Coordinate locali.....	68
4.3.5.4	Correzione per rifrazione e Equazione del tempo	69
4.3.6	<i>Analisi critica degli algoritmi considerati per il calcolo della posizione del sole</i>	69
4.3.7	<i>Approccio di TRNSYS per il calcolo dell'irradianza solare</i>	70
4.3.7.1	Irradiazione diretta o totale sul piano orizzontale.....	71
4.3.7.2	Irradiazione oraria diretta normale	72
4.3.7.3	Calcolo delle irradianze su piano orientato	75
4.3.8	<i>Calcolo dell'irradianza diffusa dalla volta celeste</i>	76
4.3.8.1	Modello di cielo anisotropo di Perez	76
4.3.9	<i>Implementazione in OpenBPS</i>	77
4.3.9.1	Metodo della trasmittanza atmosferica media	78
4.4	SCAMBIO RADIATIVO AD ONDE LUNGHE	81
4.4.1	<i>Scambio termico radiativo interno</i>	81
4.4.1.1	Calcolo del fattore di forma di corpo nero	81
4.4.1.1.1	Calcolo della matrice dei fattori di forma	82
4.4.1.1.2	Matrice dei fattori di forma: distribuzione dell'errore.....	84
4.4.1.1.3	Calcolo del fattore di forma tra poligoni.....	85
4.4.1.2	Calcolo del fattore di radiazione mutua di corpo grigio.....	90
4.4.1.2.1	Metodo esatto	90
4.4.1.2.2	Metodo approssimato	91
4.4.2	<i>Scambio termico radiativo esterno</i>	91
4.4.2.1	Radiazione dal cielo	93
4.4.2.1.1	Modelli implementati in OpenBPS.....	93
4.5	SCAMBIO RADIATIVO AD ONDE CORTE.....	95
4.5.1	<i>Elementi trasparenti: caratteristiche</i>	95
4.5.2	<i>Irradianza Diretta</i>	99
4.5.3	<i>Irradianza Diffusa</i>	100
4.6	APPORTI INTERNI.....	105
4.7	TRASMISSIONE DI CALORE PER CONDUZIONE	107
4.7.1	<i>Modelli implementati</i>	107
4.7.2	<i>Modello matematico della conduzione termica</i>	107
4.7.3	<i>Soluzione analitica monodimensionale in regime periodico stabilizzato</i>	109
4.7.3.1	Soluzione per forzanti armoniche con valore medio nullo	111
4.7.4	<i>Soluzione analitica approssimata: funzioni di trasferimento per la conduzione termica (CTF)</i>	113
4.7.4.1	Rappresentazione dei dati come somma di funzioni elementari.....	114
4.7.4.2	Risoluzione dell'equazione di Fourier mediante l'operatore della Z-trasformata	115
4.7.5	<i>Soluzione numerica: differenze finite (DF)</i>	123
4.7.5.1	Discretizzazione del dominio spazio-temporale	123
4.7.5.2	Equazione alle differenze.....	124
4.7.5.3	Scelta degli operatori.....	124
4.7.5.3.1	Selezione dello schema.....	126
4.7.5.3.2	Metodo Theta	128
4.7.5.4	Sistema di equazioni algebriche	128
4.7.5.5	Pareti multistrato.....	130
4.7.5.5.1	Singolarità dello strato puramente resistivo.....	132

4.7.5.6	Nodi sul contorno	132
4.7.5.6.1	Discretizzazione spaziale del 1° Ordine	132
4.7.5.6.2	Discretizzazione spaziale col volume finito	134
4.7.5.7	Nodi interni (alle interfacce tra strati)	135
4.7.5.7.1	Discretizzazione spaziale del 1° Ordine	135
4.7.5.7.2	Discretizzazione spaziale con i volumi finiti	136
4.7.6	Conduzione termica stazionaria	137
4.8	ZONA: BILANCIO DI ENERGIA TERMICA DEL NODO D'ARIA	141
4.8.1	Ipotesi di miscelamento perfetto; equazione differenziale ordinaria (ODE)	141
4.8.2	ODE: soluzione analitica con parametri costanti	143
4.8.1	ODE: soluzione analitica con parametri variabili linearmente	143
4.8.2	ODE: integrazione numerica	144
4.8.2.1	Metodo di Eulero	145
4.8.2.2	Metodo implicito	145
4.8.2.3	Metodo del trapezio (Crank-Nicolson): estensione al Theta Method	146
5	VALIDAZIONE DEL CODICE DELL'INVOLUCRO	150
5.1.1	Validazione con BESTEST	151
6	CONCLUSIONE E PROSPETTIVE	154
7	APPLICAZIONE AL CASO STUDIO	155
7.1	DESCRIZIONE DELL'EDIFICIO	155
7.1.1	Informazioni generali e contesto esterno	155
7.1.2	Descrizione del fabbricato	157
7.1.3	Descrizione dell'impianto	158
7.2	ASSUNZIONI DI CALCOLO E CONDIZIONI AL CONTO RNO PER LE SIMULAZIONI	159
7.2.1	Dati climatici	160
7.2.2	Zonizzazione	160
7.2.3	Condizioni al contorno	161
7.2.4	Profili di utenza	161
7.2.5	Semplificazioni	165
7.3	RISULTATI PRELIMINARI	166
8	CONCLUSIONI	173
9	RIFERIMENTI BIBLIOGRAFICI	175
10	ABBREVIAZIONI E ACRONIMI	179
11	AUTORI	181
12	ALLEGATO A: CONFRONTO TRA LE PRESTAZIONI DI C#, C++ E FORTRAN	183
12.1	SVILUPPO DEL CODICE: PRO E CONTRO DI C# VS C++	183
12.2	TEMPO DI ESECUZIONE DEGLI ALGORITMI SEQUENZIALI: INFLUENZA DELLA NATURA DELL'ALGORITMO	183
12.2.1	Complessità dell'algoritmo e/o dipendenza dal numero di linee del codice	183
12.2.1.1	Prodotti matriciali	183
12.2.1.2	Impiego delle librerie C++	184
12.3	SOVRACCOSTO PER LA PARALLELIZZAZIONE	185
12.3.1	Casi test di sovraccarico di parallelizzazione: fattorizzazione LU di matrici sparse in C# e C++	185
12.4	PARALLELIZZAZIONE CON MEMORIA CONDIVISA O DISTRIBUITA	188
12.4.1	Implementazione in C++	189
12.4.2	Implementazione in C#	189
12.4.2.1	Memoria distribuita	190
12.4.2.2	Memoria condivisa	190
12.4.2.3	Progetti C# scalabili	190
12.5	LIBRERIE NUMERICHE PER IL FRAMEWORK .NET	190
12.5.1	Codice nativo	191
12.5.1.1	Librerie sequenziali gratuite	191
12.5.1.2	Librerie parallele gratuite	191

12.5.1.3	Librerie sequenziali commerciali	191
12.5.1.4	Librerie parallele commerciali	191
12.5.2	<i>Codice gestito</i>	192
12.5.2.1	Librerie parallele commerciali	192
12.6	TEST FINALI E SCELTA.....	193
12.7	BIBLIOGRAFIA	194

Sommario

L'attività di ricerca esposta in questo rapporto rappresenta una parte del piano triennale di sviluppo di nuovo codice di calcolo per la simulazione delle prestazioni termiche e energetiche degli edifici, chiamato OpenBPS, che possa rispondere ai requisiti della nuova implementazione della direttiva EPBD Recast e successive modifiche. Per raggiungere i nuovi obiettivi di utilizzazione ottimale delle fonti di energia rinnovabili e dell'edificio a energia quasi zero (NZEB) diventa, infatti, ineluttabile l'impiego di modelli di calcolo che lavorino su base oraria e che siano estremamente flessibili per descrivere sistemi sempre più complessi e tra loro interagenti; tali programmi o codici di calcolo vengono identificati con l'acronimo BPSt (Building Performance Simulation tool).

Il codice di calcolo, nel suo sviluppo completo (alla fine del triennio di attività) sarà in grado di simulare le prestazioni termico-energetiche del sistema edificio, cioè del sistema integrato fabbricato e impianti tecnici, sia in regime di riscaldamento che di raffrescamento, con la possibilità di evidenziare il contributo delle fonti energetiche rinnovabili eventualmente utilizzate, in modo da poter essere impiegato per le verifiche richieste dalla legge e non solo.

In questo primo anno lo sviluppo del codice si è concentrato su due filoni principali: l'identificazione dell'architettura e dell'ambiente di sviluppo; la realizzazione e validazione della parte di codice che consente di simulare il comportamento del fabbricato (involucro più strutture) in regime libero (free floating) e con impianto ideale. In via preliminare, i codici di calcolo sviluppati sono stati applicati ad un caso studio fornito dall'ENEA, per un primo confronto con altri metodi di calcolo, approntati in altre linee di attività dello stesso progetto.

La definizione dell'architettura del software è di primaria importanza per la realizzazione di uno strumento che sia non solo efficiente, ma soprattutto facilmente mantenibile e espandibile, cioè che possa facilmente essere aggiornato rispetto all'evoluzione tecnologica di sistemi, componenti e materiali per l'edilizia. Con tale obiettivo è stata effettuata un'analisi delle caratteristiche principali degli attuali codici di simulazione dinamica, cercando di identificarne pregi e lacune, in modo da pervenire alla definizione dei requisiti che deve possedere il nuovo codice di calcolo. In base ai requisiti identificati, si è quindi scelto l'ambiente di sviluppo e il linguaggio di programmazione. Come architettura si è scelta un'architettura aperta, componibile ad oggetti e scalabile, identificando come ambiente di sviluppo più adeguato, perché inter-piattaforma, .NET di Microsoft e come linguaggio di sviluppo C#, un linguaggio orientato agli oggetti. Per sottolineare la scelta di un'architettura aperta e condivisa, al codice di calcolo sviluppato è stato dato il nome di OpenBPS, quale acronimo di Building Performance Simulation Open source code.

La realizzazione della parte di codice che consente di simulare il comportamento del fabbricato si è articolata in due parti principali: lo sviluppo di un'interfaccia utente-codice minimalista, necessaria per lo sviluppo e la validazione del codice stesso, tramite la quale si importano con diverse possibilità i dati relativi all'edificio che si vuole simulare; il vero e proprio codice di calcolo che esegue la simulazione delle prestazioni del fabbricato. Tralasciando di descrivere l'interfaccia, che è solo uno strumento funzionale allo sviluppo del progetto, è invece importante sottolineare l'attività sottostante la mera scrittura del codice di calcolo: la modellistica fisico-matematica-numerica che poi viene tradotta in linguaggio di programmazione. Si ritiene importante dare la possibilità all'utente di poter scegliere tra diversi modelli di uno stesso fenomeno fisico, dandogli quindi la possibilità di utilizzare quello che ritiene più adeguato al particolare caso in studio e/o di fare confronti

tra i risultati ottenibili con i diversi modelli. Quindi si è deciso di implementati più algoritmi di calcolo per diverse tipologie di componenti o processori di carattere generale.

Le caratteristiche del codice realizzato si possono riassumere nei seguenti punti.

- Il codice di calcolo lavora con un passo di integrazione temporale definibile dall'utente di dimensioni qualsivoglia: usuale passo orario, o sub-orario o, eventualmente anche maggiore dell'ora.
- In base alla sua struttura ad oggetti (vedasi scelta del linguaggio di programmazione) i vari componenti l'involucro, se del caso, possono utilizzare un passo temporale specifico, più grande o più piccolo di quello generale di simulazione.
- Per il precedente motivo, è stata posta grande attenzione nel trattamento dei dati climatici, per aver sempre dati consistenti con il modo di lavorare dello specifico solutore di uno specifico problema fisico, e un interpolatore è in grado di fornire ad ogni componente il dato istantaneo ad esso adeguato.
- Sono stati implementati diversi modelli per trattamento della radiazione solare, con qualità e prestazioni diverse, in particolare per la determinazione della posizione del sole e quindi dall'angolo di incidenza solare; ad esempio con il modello dell'Almanacco Nautico è possibile simulare un anno bisestile in modo consistente (non c'è alcun altro codice in grado di farlo).
- Per gli scambi termici radiativi ad onde lunghe tra le superfici dell'ambiente interno, oltre al classico metodo semplificato dei coefficienti di scambio termico radiativo, sfruttando la geometrizzazione della descrizione degli ambienti (vedasi input da ShetchUp), è stato implementato il calcolo diretto della radiazione tramite il metodo dei fattori di radiazione mutua determinati allo start-up in funzione dei fattori di forma per corpo nero e delle emissività delle superfici considerate.
- Riguardo alla conduzione termica in regime variabile nei componenti capacitivi, sono stati implementati diversi modelli di calcolo tra cui l'utente può scegliere e che possono coesistere contemporaneamente all'interno di una stessa simulazione: la parete A usa il modello a, la parete B il modello b. Sono stati quindi implementati:
 - a. le funzioni di trasferimento per la conduzione termica, così come sviluppate da Mitalas, comprensivi del processore che determina i coefficienti delle funzioni stesse a partire dalla stratigrafia del componente;
 - b. uno schema generalizzato alle differenze finite multi-dominio, theta method, che consente tramite un unico parametro di utilizzare uno schema completamente esplicito, o completamente implicito, o Crank-Nicolson o ponderale;
 - c. il metodo armonico delle matrici dell'ammettenza, per testing e confronti.
- Riguardo allo scambio termico attraverso ai componenti trasparenti (finestre e quant'altro), data la loro bassa capacità termica, per la trasmissione termica si usa il modello stazionario, mentre per la trasmissione di radiazione ad onde corte si esegue il calcolo tenendo conto della dipendenza angolare delle proprietà radiative del sistema trasparente (trasmissanza, riflettanza e assorbanza. È possibile procedere in due modi:
 - d. utilizzare i risultati forniti dal programma specializzato Window in termini proprietà angolari, trasmissanza termica e quant'altro, assumendole in input;
 - e. descrive la composizione del sistema trasparente per la determinazione di tali proprietà direttamente da parte del codice OpenBPS.
- Riguardo al "motore" centrale del calcolo, il nodo aria per la singola zona termica, l'equazione di bilancio termico per l'ambiente può, innanzitutto, essere risolta in due modalità diverse:
 - a. Ricerca del carico termico di riscaldamento e raffrescamento, tramite imposizione delle temperature interne di progetto;
 - b. Simulazione del comportamento in free-floating con l'intervento o meno dell'impianto per limitare la banda di oscillazione in funzione della tipologia di regolazione.

Tale equazione differenziale del primo ordine, può poi essere integrata, a scelta dell'utente, con schemi di numerici diversi, quali l'esplicito e l'implicito, ma anche una soluzione analitica approssimata.

Introduzione

L'efficienza energetica degli edifici è importante sia dal punto di vista economico che ambientale ed è il punto focale della politica energetica e ambientale dell'Unione Europea, così come attuata tramite le due direttive EPBD e il loro aggiornamento [1,2,3], e della politica energetica italiana tramite il loro recepimento [4,5,6]. La tecnologia per pervenire a un notevole risparmio energetico nel settore dell'edilizia di fatto esiste già, ma tale potenziale risparmio non è ancora stato pienamente sfruttato, anche a causa di una prassi progettuale non più adeguata ai nuovi obiettivi e necessità. Infatti, per raggiungere gli obiettivi di efficienza posti delle nuove regole legislative, i progettisti hanno sempre più la necessità di utilizzare strumenti che siano adeguati per l'analisi e la comprensione del complesso comportamento degli edifici, sia rispetto al fabbisogno energetico che al soddisfacimento e al mantenimento di altri requisiti prestazionali, quali il benessere termico, acustico e visivo. Fortunatamente, negli ultimi quaranta anni, per fornire una valutazione accurata e dettagliata della prestazione energetica degli edifici e non solo, sono stati sviluppati, dapprima nell'ambito della ricerca, e, successivamente, in ambito commerciale, i Building Performance Simulation Tools (BPST) [7]. I BPST sono software che spaziano, dal punto di vista della complessità, dai semplici fogli di calcolo a strumenti di simulazione avanzati per applicazioni specifiche, e, dal punto di vista dell'integrazione, da strumenti che gestiscono un singolo aspetto della progettazione degli edifici a strumenti che ne integrano i molteplici aspetti. In modo esemplificativo l'evoluzione storica di tali strumenti può essere suddivisa in quattro generazioni. Gli strumenti di prima generazione si fondano su metodi semplificati che si ritrovano sui manuali, cioè su calcoli basati su formulazioni analitiche che comprendono molte ipotesi semplificative e quasi sempre sull'ipotesi di stazionarietà dei fenomeni considerati. Gli strumenti di seconda generazione si basano su metodi che introducono una modellazione semplificata, ancora analitica, della dinamica degli edifici. Gli strumenti attuali, di terza generazione, utilizzano invece metodi numerici e forniscono un'integrazione parziale dei diversi aspetti delle prestazioni degli edifici, ad esempio quelli energetici, illuminotecnici e acustici. La futura quarta generazione, in fase di sviluppo, tende ad una completa integrazione in relazione a diversi aspetti delle prestazioni dell'edificio, con nuovi sviluppi miranti alla realizzazione di interfacce utente che forniscono assistenza intelligente impiegando basi di dati di competenze specifiche, miranti al controllo della qualità delle applicazioni e alla formazione degli utenti. Limitandoci agli strumenti attualmente disponibili, questi possono quindi catturare le specificità della realtà molto meglio di strumenti precedenti (seconda e prima generazione), ma sono più complessi da utilizzare.

Gli strumenti per l'analisi delle prestazioni energetiche dell'edificio sono progettati e realizzati per prevedere il consumo annuo di energia dei sistemi impiantistici presenti. Basati su un sistema di equazioni che definiscono le prestazioni termiche degli edifici e dei sistemi impiantistici considerati, questi strumenti eseguono simulazioni su base oraria o sub-oraria del comportamento dell'edificio, come si definisce oggi il sistema fabbricato-impianto, in funzione delle condizioni al contorno, delle strategie operative e delle azioni di controllo predefinite. Tali strumenti tra cui Carrier SAD, Trane TRACE 700, DOE-2, Equest, EnergyPlus, ESP-r, IDA ICE, TRNSYS, HVACSIM, VA114, SIMBAD, IES, sono in genere utilizzati per calcolare e analizzare le prestazioni a pieno carico e a carico parziale, per analizzare la strategia di funzionamento del sistema, per confrontare le diverse alternative di progetto e così via. All'interno di questa categoria di strumenti si può isolare un sottogruppo che ha come specificità la capacità di analizzare correttamente l'impatto delle diverse tipologie e strategie di controllo, del quale fanno parte tra gli altri ESP-r, EnergyPlus, IDA ICE, TRNSYS. Gli strumenti di simulazione che appartengono a questo sottogruppo, dai quali ci si attende un'accurata

valutazione delle prestazioni del sistema, devono essere in grado di trattare le deviazioni dal comportamento ideale che si verificano nei sistemi reali e di modellare realisticamente sia i controlli che la dinamica del sistema impiantistico, nel seguito identificato dall'acronimo HVAC, il che però non è quasi mai verificato, giacché la quasi totalità dei componenti HVAC è modellata tramite equazioni stazionarie, che non descrivono sempre correttamente la dinamica del componente. L'unica possibilità di avere una buona corrispondenza tra realtà e simulazione è quella di rispettare le ipotesi alla base del modello di simulazione del fabbricato e dell'impianto e di procedere a una accurata calibrazione sui dati misurati. La prima condizione comporta la conoscenza dei modelli dei componenti e delle relative ipotesi semplificative, dei loro tempi caratteristici e quindi della determinazione del minimo passo temporale che comporti il soddisfacimento dell'ipotesi di stato quasi stazionario, che si ha quando le equazioni dello stato stazionario sono in grado di ben approssimare lo stato non stazionario; è evidente che ciò comporta una certa competenza specifica da parte dell'utente dello strumento di simulazione. La seconda condizione, in realtà necessaria solo se si opera su un edificio esistente ai fini di una diagnosi energetica, comporta conoscenze e competenza specifiche ancora maggiori. Come se ciò non bastasse, i diversi approcci di modellazione del sistema HVAC presenti nei diversi strumenti disponibili, quali quelli puramente concettuali, quelli basati sui componenti o su un approccio multidominio, richiedono diversi livelli di abilità da parte degli utenti, diverse risoluzioni temporali e/o spaziale nella modellazione e diversi livelli di capacità di personalizzazione. Un elevato livello di dettaglio nella rappresentazione del sistema richiede una maggiore conoscenza del sistema stesso, a causa del crescente numero di parametri necessari al modello per descriverlo e che spesso sono difficili da ottenere in quanto non vengono forniti dai produttori. Ciò comporta anche una maggiore richiesta computazionale, sia in tempo che in memoria, e un'analisi dei risultati più complicata, ergo maggiori costi. Di conseguenza, un buon modello deve essere il meno complesso possibile, ma tale da mantenere la sua validità in funzione degli obiettivi della simulazione; infine tale minima complessità del modello non è un parametro assoluto ma dipende da cosa si vuole ottenere.

Si deve poi considerare che al variare degli obiettivi della simulazione, quindi della complessità del modello, il costo, in termini di quantità di informazione, tempo necessario per costruirlo, tempo di calcolo e quantità di memoria impiegata, può superare il valore aggiunto prodotto per l'utente e tutto ciò va bilanciato con la necessità di avere un errore cioè una deviazione dei risultati del modello dai dati reali, che sia accettabile). L'errore in un modello che rispecchia fedelmente il sistema descritto è la somma di quello di astrazione, di quello nei dati di ingresso, e di quelli numerici; il primo è dovuto alle astrazioni apportate dalla modellazione a causa dell'incompletezza del modello rispetto al sistema fisico, il secondo è dovuto alle incertezze nei valori nei parametri di ingresso, che possono essere quantificate ottenendo la corrispondente incertezza del risultato del modello, la cosiddetta incertezza predittiva. L'errore di astrazione può dipendere dal livello di complessità, che diminuisce all'aumentare della complessità del modello), ma anche e soprattutto dall'inadeguatezza del modello, dovuta, ad esempio, all'impiego di un modello lineare in un sistema non lineare. Infine gli errori numerici, che dipendono in parte dal passo spaziale di discretizzazione e in parte da quello temporale, possono essere controllati diminuendo il passo, il che aumenta il tempo di calcolo e l'uso di memoria. Come si può ben vedere, si è in presenza di una coperta corta che comporta una competenza particolare per un uso efficiente della simulazione dinamica.

In conclusione, l'applicazione della simulazione dinamica nella progettazione edilizia è a oggi problematica non solo perché gli strumenti di simulazione sono particolarmente complessi e molti progettisti edili non hanno familiarità con le loro proprietà e limitazioni, ma anche perché la stessa realizzazione del modello comporta competenze e conoscenze non usuali. Nella vita reale, la natura del processo di progettazione edile e la carenza di formazione nel settore della modellistica e della simulazione energetica hanno reso difficile sia per l'architetto che per l'ingegnere edile l'impiego efficiente e consistente di tali strumenti, comunque oggi indispensabili.

Come si può ovviare a tale oggettiva situazione di conflitto tra esigenze e competenze? Vi sono due vie che occorre percorrere in parallelo:

- sviluppare negli utilizzatori una migliore comprensione della modellazione e della simulazione energetica per consentire loro di valutare e mettere in pratica le tecniche per realizzare edifici ad alta efficienza energetica, tramite una formazione continue e mirata;
- rendere disponibili dei codici di simulazione dinamica del sistema edificio che, pur consentendo un'analisi dettagliata e di qualità (base di calcolo oraria e sub-oraria), siano utilizzabili da utenti con gradi diversi di competenza, facilmente integrabili in altri strumenti di progettazione, ad esempio nei BIM, e che possano evitare una grande richiesta di dati all'utente, tramite l'impiego di informazioni provenienti direttamente dai costruttori di componenti.

L'attività, di cui all'accordo di collaborazione tra Enea e Politecnico di Milano – Dipartimento di Energia, è proprio motivata e centrata sullo sviluppo di un codice di calcolo innovativo che risponda ai requisiti su citati, pur operando su base oraria o sub-oraria, e che sia in grado di analizzare in modo dettagliato i vari fenomeni termo-energetici che si attuano all'interno del sistema edificio.

Tale attività di ricerca e sviluppo del codice è articolata in tre fasi, corrispondenti a tre annualità, che affrontano rispettivamente,

- fase I:
sviluppo e validazione del codice di calcolo relativo alla prestazione termo-energetica dell'involucro edilizio,
- fase II:
lo sviluppo e la validazione dei codici di calcolo dei componenti impiantistici previsti nel pacchetto normativo sotto mandato della Commissione Europea per l'implementazione della EPBD recast e la loro integrazione nel processo di calcolo dinamico,
- fase III:
definizione di una procedura codificata chiara e trasparente che consenta di assicurare, al termine del progetto, lo sviluppo continuativo, la manutenzione e l'aggiornamento del codice di calcolo, tramite il coinvolgimento di quanti, esperti del settore, vogliano e possano contribuire, prestando la loro competenza in modo coordinato ed efficace in un ambito "open source".

A margine di tale attività è poi prevista un'applicazione del codice sviluppato a dei test case forniti da Enea per consentire una comparazione di diversi metodi e modelli di calcolo, sviluppati sempre nell'ambito dello stesso piano triennale.

1 Caratteristiche principali degli attuali codici di simulazione dinamica

La progettazione o il miglioramento dell'efficienza energetica del sistema edificio non possono prescindere dal mantenimento o dal raggiungimento di altri importanti e concomitanti obiettivi, quali quello della qualità dell'ambiente interno, dei costi e della sostenibilità ambientale. Per gestire un processo progettuale multi obiettivo e con potenziali contrasti tra tali obiettivi, nell'effettuare le scelte è necessario disporre di uno o più strumenti che consentano di "informare" il progettista sugli esiti di tali scelte, in modo da renderle del tutto consapevoli. La natura dell'edificio, costituito da una molteplicità di elementi interagenti quali l'involucro, i sistemi impiantistici, le utenze e i servizi, impone che tali strumenti cerchino di prevedere i diversi aspetti del comportamento di uno o più dei suoi sistemi basandosi su modelli. Tali modelli della realtà, sempre approssimati, possono essere, ad esempio, matematici o fisici o basati su inferenza statistica, e a loro volta devono essere in grado di descrivere in modo adeguato la complessità, le variabilità e le interconnessioni del sistema considerato con il "mondo circostante". In sintesi, uno strumento che cerca di emulare la realtà fisica tramite modelli di questa è ciò che indichiamo come programmi di simulazione dinamica. È comprensibile come la scelta o la costruzione di un modello non adeguato agli obiettivi del sistema simulato possa portare a una valutazione erranea o troppo superficiale delle potenziali azioni progettuali simulate, ed è quindi fondamentale usare in modo competente e cosciente questi programmi, valutandone l'opportunità e le possibilità offerte. Da tutto ciò consegue la necessità di individuare i requisiti cui debba rispondere un moderno programma di simulazione dinamica dell'edificio, anche noto come Building Performance Simulation tool, BPSt (acronico che utilizzeremo in seguito).

Il primo elemento da prendere in considerazione è che oggi il potenziale utente di tali strumenti BPSts non è più unico come nel passato (solo ricercatori): attualmente le figure coinvolte nell'evoluzione di tali strumenti possono essere divise in tre categorie:

- l'utente base;
- l'utente avanzato;
- lo sviluppatore.

Le richieste dell'utente base riguardano principalmente:

- la riduzione del tempo di introduzione dei dati, di modellazione e di analisi dei risultati ottenuti da diverse alternative progettuali per le quali il software fornisca indicatori che ne valutino le prestazioni secondo un numero sufficiente di aspetti economico/funzionali;
- la possibilità di simulare in modo accurato il maggior numero di sistemi esistenti o in via di sviluppo.

Queste due richieste identificano due macro aree di indirizzo: **l'operatività**, che risponde al primo punto, e **l'adeguatezza**, che risponde al secondo ed è relativa alla *completezza* e all'*accuratezza* dei modelli implementati nel software. Come confermato da uno degli ultimi sondaggi proposti agli esperti del settore [8], queste due macro categorie di requisiti, suddivise in diverse sotto-categorie più dettagliate, sono ritenute le più importanti dall'utente base, anche se la loro importanza relativa è influenzata dal background culturale di ciascun utente, architettonico o ingegneristico. Fortunatamente, invece, la caratteristica della facilità di apprendimento ha assunto un'importanza secondaria rispetto a qualche anno fa. Dato il rilievo che ha assunto il tema dell'efficienza e del risparmio energetico in questi ultimi anni, la necessità che il professionista

che si trovi a utilizzare tali strumenti abbia una preparazione adeguata è finalmente e meritatamente stata riconosciuta e accettata.

Oltre alle necessità degli utenti base, anche quelle degli utenti avanzati e degli sviluppatori hanno una rilevanza non trascurabile: l'aggiornamento continuo di tali strumenti è infatti di fondamentale importanza per la loro sopravvivenza, per cui è necessario aggiungere ulteriori requisiti a quelli dell'utente di base, tra i quali la possibilità per lo strumento *di essere facilmente personalizzato, ampliato e mantenuto*.

1.1 Requisiti fondamentali

I requisiti di *operatività* e *adeguatezza* sono stati dettagliati nel sondaggio in precedenza citato [8] nei seguenti sotto-requisiti:

1. usabilità e gestione dell'informazione da parte dell'interfaccia (UIM);
2. integrazione delle conoscenze di base di una progettazione intelligente (IIKB);
3. accuratezza degli strumenti e capacità di simulare in dettaglio componenti dell'edifici complessi (AADCC);
4. interoperabilità della modellazione dell'edificio (IBM);
5. integrazione con il processo di progettazione dell'edificio (IBDP);

ai quali se ne possono aggiungere altri due, data la diffusione dei servizi cloud e la crescente necessità di compiere calcoli che richiedono rilevanti tempi di elaborazione computazionale (Mazzarella et al., 2014):

- l'ubiquità e la condivisibilità del progetto, ad esempio sotto forma di BIM, modelli di simulazione o relazioni finali;
- la riduzione dei tempi di calcolo richiesti, soprattutto nel caso di ottimizzazione multi-obiettivo e/o di simulazioni multi-fisiche quali CFD e ray-tracing, tramite l'ottimizzazione e la parallelizzazione del codice di calcolo.

1.1.1 Operatività vs Adeguatezza

L'**operatività** include al suo interno tutti i sotto requisiti elencati in precedenza tranne uno, l'*adeguatezza*, che è descritta compiutamente dal requisito **AADCC**. Va detto che l'adeguatezza di un software non implica la correttezza dei risultati finali, che dipendono anche dal modo d'impiego dello specifico strumento. Infatti, qualora l'utente introduca un modello rappresentativo inadeguato del sistema in analisi in uno strumento adeguato, lo strumento non potrà fornire risultati accurati; quest'aspetto, dominato dalla competenza dell'utente, non sarà ulteriore oggetto della presente discussione.

Nonostante le due famiglie di requisiti individuate siano ritenute le più importanti dagli utenti di questo tipo di tools, la loro importanza relativa, non uguale per tutti gli utenti, ha determinato alcune delle carenze palesate dagli attuali strumenti, che sono infatti il risultato di un orientamento divergente del mercato di sviluppo di tali software, laddove una tipologia di utenti si è dimostrata più interessata all'automatizzazione del processo di progettazione mediante una IIKB, mentre l'altra alla AADCC [8]. Questo ha portato alcuni sviluppatori a lavorare essenzialmente sulla semplificazione dell'interfaccia grafica e del processo di input dei dati, quindi alla operatività, tralasciando il resto, al contrario di altri, che poco hanno fatto per semplificare il processo di input, concentrandosi invece sulla qualità degli algoritmi di calcolo, cioè sulla adeguatezza. Un programma di simulazione che sia in grado di rispondere a tutti i requisiti elencati deve quindi essere sviluppato rispettando entrambi questi due requisiti principali e ciò potrà accadere solo nel momento in cui le diverse figure in grado di guidare il mercato dello sviluppo di tali tools condivideranno le stesse finalità, come dichiarato in [8]: *“il tipico processo di progettazione unidisciplinare, in cui l'architetto e gli ingegneri*

lavorano in isole separate e senza obiettivi prestazionali, non può raggiungere gli obiettivi del nuovo millennio”.

È necessario quindi adoperarsi per evitare che le iniziative intraprese per soddisfare requisiti di natura più operativa determinino ricadute negative in termini di adeguatezza dei risultati ottenuti. In particolare, bisogna evitare di semplificare troppo il modello di simulazione a prescindere dalla sua reale complessità, così come va evitato di semplificare troppo l'uso di un modello complesso, disattendendone le proprietà. Per capire come sia possibile coniugare la riduzione della complessità, pur mantenendo una sufficiente completezza, con l'accuratezza dei risultati della simulazione è meglio compiere un passo indietro e chiedersi che confidenza abbiamo con il grado di accuratezza dei calcoli eseguiti dai diversi software presenti sul mercato.

1.1.2 Valutazione dell'accuratezza

Negli ultimi anni, una rinnovata attenzione è stata dedicata alla valutazione dell'accuratezza dei risultati ottenuti dai BPSTs. In particolare è stata attribuita una maggiore attenzione agli errori compiuti nella valutazione delle temperature dell'aria negli ambienti simulati, ovvero dei nodi d'aria, risultanti dal calcolo. In passato il principale interesse era rivolto principalmente a una valutazione integrale degli errori in termini di energia consumata, mentre poco veniva detto circa gli errori di valutazione delle diverse temperature presenti nel sistema. Ponendo correttamente sullo stesso piano l'energia consumata e i requisiti di comfort termico, anche una corretta valutazione dei profili temporali delle temperature superficiali interne delle pareti e delle temperature raggiunte dai volumi d'aria considerati costituisce un aspetto di fondamentale importanza.

A tale proposito, risulta di effettivo interesse uno dei risultati ottenuti durante la definizione della metodologia di validazione empirica dei software di simulazione dinamica dell'edificio, sviluppata all'interno dell'Annex 58 del Progetto IEA BCE *“Reliable Building Energy Performance Characterisation Based on Full Scale Dynamic Measurements”* [10]. I risultati ottenuti da un esempio applicativo di tale metodologia di validazione empirica [11], hanno infatti, messo in luce come nessuno dei più diffusi BPST oggi disponibili abbia prodotto risultati che primeggiano rispetto agli altri in termini di accuratezza su tutte le metriche utilizzate.

Per discutere di questi aspetti, è fondamentale disporre di un set di metriche diverse fra loro con le quali valutare gli errori commessi nel calcolo e di un set di variabili di natura diversa, temperature e flussi, cui applicare ciascuna delle metriche precedentemente definite, per comprendere la differente distribuzione degli errori compiuti nella valutazione di una variabile. Nel caso in questione, sono stati definiti due indici “significativi” sui quali basare il confronto inerente all'accuratezza raggiunta in termini di flussi e temperature dei nodi d'aria così come calcolati dai diversi software [11]:

- un indice di “magnitude fit”;
- un indice di “shape fit” o livello di corrispondenza di forma del profilo assunto dalla variabile considerata.

La “magnitude fit” è stata definita come la differenza media assoluta tra la misurazione e la previsione del valore assunto dalla variabile considerata, per ciascun periodo sperimentale e per ciascuna zona termica. La “shape fit” è stata valutata mediante il coefficiente di correlazione di Spearman [12] tra le previsioni e le misure effettuate sulla variabile considerata, applicato ai valori ottenuti, per ciascun periodo sperimentale e per ciascuna zona termica. Le due metriche sono state applicate alle temperature di nodo d'aria e alla potenza fornita dall'impianto di riscaldamento.

Potrebbe essere utile tuttavia aggiungere ancora qualche aspetto alle metriche e alle variabili individuate; ad esempio il set di variabili cui applicare le metriche previste potrebbe essere esteso anche alla valutazione degli errori ottenuti sulle temperature superficiali interne delle pareti delle diverse zone termiche, mentre alle metriche previste, potrebbero essere aggiunte altre valutazioni funzionali alla comprensione dello “shape fit” assunto da ciascuna variabile. Infatti, il coefficiente di Spearman resta un indice di valor medio, mentre potrebbe essere utile avere informazioni inerenti allo sfasamento temporale rilevato sul

raggiungimento di valori di picco delle variabili considerate e all'errore su tali valori di picco, che faceva parte delle metriche valutate in alcuni casi di validazione precedenti, quali l'ASHRAE Standard 140-2011 [13].

Anche questo esempio di validazione ha mostrato differenze rilevanti nella valutazione delle temperature dei nodi d'aria ottenute dalle diverse modellazioni e dai diversi software e ha confermato l'importanza dell'identificazione dei parametri da valutare e della modalità mediante la quale validare l'accuratezza ottenuta dai software attualmente disponibili ai fini della comparabilità dei risultati ottenuti da tali strumenti. Spesso, infatti, i diversi software vengono sottoposti a validazioni diverse e i risultati da essi ottenuti vengono restituiti solo parzialmente, rendendo la loro comparazione più complessa.

1.1.3 Capacità di crescita evolutiva

I software di simulazione, descrivendo sistemi complessi e in continua evoluzione, devono essere soggetti a un continuo aggiornamento. Il modello di evoluzione scelto, ne potrebbe determinare il successo, così come descritto da Fischer [14], che ha individuato quale unico modello applicabile a tali sistemi il SER, "Seeding, Evolutionary growth and Reseeding", mostrato in Figura 1. Il SER prevede che i periodi di attività evolutiva non pianificata e quelli di programmata (ri)strutturazione e valorizzazione di quanto sviluppato in precedenza debbano continuamente alternarsi tra di loro.

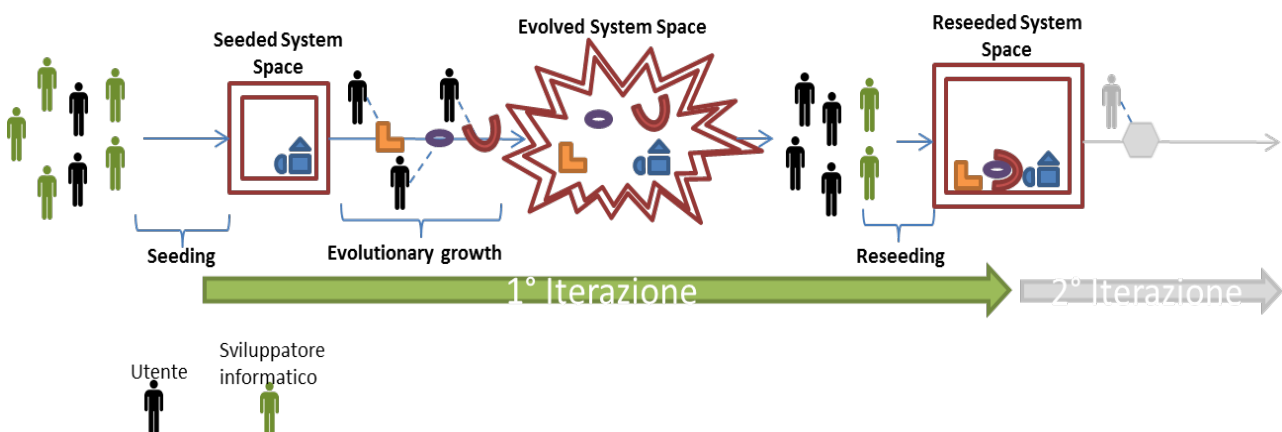


Figura 1 – Seeding, Evolutionary growth and Reseeding: modello di sviluppo di sistemi complessi.

Anche i BPSTs potrebbero dunque beneficiare di una struttura che consenta loro di evolversi insieme all'esperienza dell'utente, mettendoli in grado di adattarsi e rispondere a esigenze non note durante le prime fasi dello sviluppo. Il SER si basa infatti sulla convinzione che l'interazione uomo-computer si evolverà da "facile da usare", che non significa che non siano necessarie alcune competenze relative alla comprensione della complessità intrinseca dello strumento, a "facile da sviluppare" [15]. Esempi di End User Development, EUD, sono rappresentati dalla possibilità di registrare o scrivere una macro in Visual Basic all'interno di un foglio di calcolo o dalla possibilità di usare un particolare linguaggio per la parametrizzazione di un disegno CAD. L'importanza di agevolare una tale Evolutionary growth o "crescita evolutiva" è anche uno dei motori nell'uso di general-purpose programs, all'interno dei quali è resa più semplice appunto la definizione di un nuovo modello matematico da analizzare.

Sarà necessario naturalmente realizzare livelli di "personalizzazione-utente" caratterizzati da diverse difficoltà di sviluppo (semplice, avanzata ed esperta) e per ogni livello dovranno essere individuati e implementati gli strumenti più adatti allo scopo. Non deve essere pertanto necessario che, a prescindere dal livello di implementazione della "personalizzazione" desiderato, l'utente debba diventare un "esperto" informatico, anche se, qualora il livello di modifica sia elevato, potrebbero rivelarsi necessarie maggiori abilità informatiche.

Un aspetto importante che non richiede competenze informatiche potrebbe essere l'arricchimento della banca dati di "componente" disponibile per la simulazione, che potrebbe consistere in una attività di

inserimento tramite pagine web dei dati relativi ai prodotti presenti sul mercato, secondo format di scambio dei dati adeguati ai codici di simulazione dinamica.

Un'attività che richiederebbe invece maggior impegno è quella di implementazione di nuovi modelli o di modifica di quelli esistenti. In questi casi, per permettere una proficua collaborazione fra diversi sviluppatori separati nello spazio e nel tempo, si rivelano di fondamentale importanza la presenza di una struttura chiara del codice esistente, la definizione di regole di sviluppo, la chiara presentazione dei risultati raggiunti e la gestione e documentazione del codice sviluppato.

Questo tipo di evoluzione spesso ritenuto sufficiente a decretare il successo di un software nei confronti di un altro, porta spesso alla creazione di parti di codice create da mentalità diverse per affrontare problemi a volte contingenti, che talvolta risultano poco coordinate o armonizzate e magari in sovrapposizione parziale fra loro e che potrebbero trarre enormi benefici da una migliore strutturazione, generalizzazione e formalizzazione di quanto già implementato. Questa fase di Reseeding o "risemina" dovrebbe spingere gli sviluppatori informatici a collaborare nuovamente con gli esperti del settore di applicazione del software e consentirebbe al nuovo sviluppo di beneficiare di nuovi linguaggi, framework e funzionalità in precedenza non disponibili.

1.2 Tipologie di BPSt esistenti

Nella pagina web "Building Energy Software Tools" [7], fino a poco tempo fa gestita dal Dipartimento di Energia statunitense e dedicata all'elenco dei software di simulazione dinamica dell'edificio, (selezionando come filtro "Whole-building Energy Simulation" si ottiene una lista di 69 software fra i quali motori di calcolo più o meno complessi (fra i più noti EnergyPlus, ESP-r, TRNSYS, IDA ICE, Modelica Buildings library) e interfacce grafiche studiate per la combinazione o il più facile utilizzo di tali motori di calcolo (fra le più note DesignBuilder e OpenStudio). In alcuni casi non è immediato capire se si stia guardando un'interfaccia a un motore di calcolo e quale, o un vero e proprio motore di calcolo.

In alcuni casi, anche se purtroppo non sempre, sono disponibili sia i file di input usati durante il processo di validazione del software, sia la relativa documentazione che ne riassume i risultati (a volte però riportati solo parzialmente), entrambi aspetti di fondamentale importanza ai fini della comprensione del funzionamento e dell'accuratezza del software.

Alcuni dei programmi citati, come IDA ICE e la Modelica Buildings library, sono nati con l'idea di sfruttare linguaggi più simili a quelli dell'ambito di applicazione dei BPSt, insieme a motori di calcolo general-purpose, per rendere la fase di crescita evolutiva del software maggiormente fruibile da un utente non informatico o esperto di analisi numerica.

I programmi ad oggi sviluppati possono essere suddivisi in due principali famiglie:

- special-purpose programs (programmi per applicazioni specifiche);
- general-purpose programs (programmi di applicazione generalizzata).

1.2.1 Special-purpose programs

Gli "special-purpose programs" sono programmi scritti in linguaggio informatico, come il "C" o il "FORTRAN", nelle relative evoluzioni, per la risoluzione di un problema ben definito. Uno sviluppo di questo genere parte dalla definizione di un problema specifico, caratterizzato da una certa struttura e natura, ne identifica il modello matematico (un sistema di equazioni algebriche (AE), differenziali ordinarie (ODE) e/o alle derivate parziali (PDE)) e ne implementa una soluzione efficiente dal punto di vista numerico/computazionale. Quest'approccio consente di raggiungere una buona robustezza del codice e limita il rischio di generare problemi mal definiti, fintanto che i dati di ingresso sono ragionevoli.

La natura specifica di questi programmi non ne preclude una struttura modulare, ovvero più o meno favorevole all'inclusione di nuovi parti di codice. Ad esempio, in TRNSYS è possibile includere facilmente

nuove implementazioni legate a parti impiantistiche del sistema da simulare, mentre si incontrano maggiori difficoltà nella modifica e nella espansione di modelli legati all'edificio. Questa maggiore o minore facilità nella fase evolutiva del codice talvolta è legata a questioni di natura commerciale, anche se più spesso è dovuta alla struttura con cui è stato sviluppato il modello informatico per la simulazione computazionale di un particolare sistema, come quello dell'edificio. In alcuni casi, infatti, ad esempio ESP-r, il modello computazionale, che implementa il modello matematico relativo allo scambio di massa ed energia all'interno di un edificio, viene realizzato costruendo un'unica matrice che poi viene partizionata per ridurre i tempi di calcolo. Tale matrice, infatti, arriva ad assumere dimensioni rilevanti che potrebbero portare a tempi di calcolo, legati alla sua inversione, talvolta troppo lunghi. L'analisi numerica propone metodi per il partizionamento di matrici sparse, volti alla riduzione del relativo tempo di calcolo, ma per aggiungere o modificare una parte del modello matematico, rappresentato computazionalmente dalla matrice, è necessario capirne la struttura e potrebbe essere necessario valutarne il nuovo partizionamento, nel caso quest'ultimo non fosse stato automatizzato seguendo logiche valide "a priori".

Per cercare di ridurre problematiche di questo genere, in alcuni software di questo tipo alle equazioni definite nel modello matematico di alcuni componenti del sistema sono stati aggiunti dei termini "generici" che potessero essere usati per introdurre aspetti precedentemente non considerati.

1.2.2 General-purpose programs

Un programma general-purpose nasce per risolvere qualsiasi tipo di problema venga descritto al suo interno. L'utente di un tale tipo di software, quando vuole simulare un dato sistema deve, in generale, definire prima un modello fisico e poi il suo modello matematico, quindi introdurlo nel programma, tramite un suo specifico linguaggio di programmazione. Per la soluzione di tale modello il programma, dopo una manipolazione simbolica delle equazioni che costituiscono il modello, sfrutta le librerie numeriche che ha in dotazione e che però non sempre sono adeguate alle necessità. Ad esempio, se il modello matematico è descritto da un'equazione differenziale alle derivate parziali (PDE) e le librerie numeriche a disposizione del programma, come nel caso di Modelica e IDA ICE, sono in grado di risolvere solo equazioni differenziali ordinarie (ODE), l'utente dovrà modificare il modello matematico che intende simulare in modo da ricondurlo a ciò che risulta risolvibile dal programma general-purpose che sta utilizzando. Oltre che costringere l'utente a costruire un modello matematico del sistema che si vuole simulare, cosa che comporta competenze specifiche che non tutti i potenziali utenti possiedono, tali programmi spesso richiedono la conoscenza di un linguaggio di programmazione proprietario, che è funzionale alla descrizione formale delle equazioni alla base del modello più che all'implementazione di algoritmi numerici.

Lo scopo di questo tipo di programmi, caratterizzati da elevata estendibilità e flessibilità all'interno del dominio di funzionamento, è quello limitare le responsabilità assegnate allo sviluppatore alla descrizione del modello delegando la loro soluzione numerica alle librerie in essi implementate.

Gli strumenti di uso generale in genere implementano una programmazione dichiarativa (non causale), cioè centrata su ciò che il programma dovrebbe realizzare senza specificare come il programma dovrebbe raggiungere il risultato. anziché procedurale (causale), cioè centrata sulla descrizione del come il programma raggiunge in risultato atteso. Le caratteristiche dei modelli sviluppati seguendo questi due approcci sono:

- Modelli procedurali (causali):
 - sono descritti in una forma più vicina all'algoritmo di risoluzione;
 - il modello di ciascun sistema è dato identificando input e output;
 - sono collegati a un contesto specifico;
 - sono caratterizzati da scarsa riutilizzabilità dei modelli di componenti di base.
- Modelli dichiarativi (non causali):
 - può essere definito da equazioni costitutive;

- sono descritti in una forma indipendente dal contesto;
- sono caratterizzati da complessità nella definizione del modello di simulazione;
- sono caratterizzati da elevata riutilizzabilità dei modelli di componenti di base;
- la loro causalità è determinata dal sistema aggregato dal simulatore.

Tra i programmi general-purpose vanno citati Dymola, OpenModelica e IDA ICE. Modelica è un ambiente general purpose che utilizza un linguaggio dichiarativo di modellazione orientata agli oggetti. Dymola è il front-end commerciale più utilizzato per Modelica, mentre OpenModelica ne è l'alternativa gratuita. Il linguaggio Neutral Model Format, NMF, introdotto alla fine degli anni '80 da Per Sahlin ([16] e sul quale è stato sviluppato IDA ICE, è il linguaggio predecessore di Modelica. Mentre IDA-ICE nasce per la simulazione delle prestazioni degli edifici e quindi è dotata nativamente di una libreria di modelli di componenti specifici richiamabili in modo trasparente all'utente tramite un'interfaccia grafica (che quindi può evitare di costruire i vari modelli matematici tramite il linguaggio di programmazione), Modelica, che mantiene il suo carattere general purpose, supporta tale obiettivo tramite la "Modelica Buildings library", una libreria open source a tal scopo sviluppata. Tuttavia, dal punto di vista dello sviluppatore (e non del mero utente di un BPSt) vi sono casi, come il calcolo dei fattori di forma fra geometrie complesse, in cui il problema maggiore non risiede nella definizione del modello tramite la scrittura di una o più equazioni matematiche, bensì nella gestione ottimizzata del processo di calcolo numerico. In questi casi il linguaggio di programmazione proprietario, finalizzato alla descrizione formale di equazioni, risulta meno efficiente e funzionale di un classico linguaggio di programmazione, come C++, C#, FORTRAN, ecc...

Alcuni svantaggi generati dall'approccio usato in tali programmi sono una scarsa efficienza computazionale, le limitazioni nella soluzione dei sistemi matematici la cui traduzione numerica non sia già compresa nelle librerie numeriche disponibili al programma e le difficoltà nel comprendere messaggi di errore generati dal programma durante la simulazione, a causa della manipolazione simbolica applicata al sistema di equazioni.

Un altro esempio di general-purpose programs comprende i casi di utilizzo di Matlab, accoppiato a Simulink per la definizione di librerie utili alla descrizione del sistema edificio-impianti. In questo caso si ha la possibilità di utilizzare le funzionalità drag and drop offerte da Simulink e la vasta libreria di metodi numerici e statistici implementati in Matlab. Tuttavia, sono stati evidenziati alcuni svantaggi [17], quali:

1. lo sviluppo di modelli procedurali;
2. la necessità di scomporre il sistema in blocchi che interagiscono secondo uno schema causa-effetto;
3. il fatto che si tratta di un ambiente "signal-oriented", per il quale è necessario prestare la dovuta attenzione a cicli che potrebbero causare problemi numerici;
4. la necessità di una licenza Matlab-Simulink.

1.3 Interoperabilità e Co-simulazione

La limitazione intrinseca dei codici di calcolo dedicati a specifici aspetti (special-purpose programs), solitamente più efficienti ma meno flessibili, sono superabili tramite l'applicazione del concetto di interoperabilità o quello, più stringente, di co-simulazione.

L'interoperabilità consiste nella possibilità e capacità di programmi special-purpose diversi di cooperare tra loro scambiandosi in modo affidabile informazioni, in modo da affrontare la risoluzione di più aspetti del sistema in esame senza dovere "istruire" più volte e separatamente programmi diversi, ognuno capace di risolvere un unico aspetto del problema. Molto spesso l'interoperabilità si limita alla definizione di un protocollo standardizzato di scrittura/lettura di file di informazione che i diversi programmi possono scambiarsi; mentre l'interoperabilità necessaria nel campo della simulazione non è collegata all'unica possibilità di scambiare dati statici, ma è legata alla risoluzione matematica e numerica del problema, cioè

ad uno scambio dinamico di informazioni riguardanti lo stato del sistema. Ciò che necessita è un'interoperabilità di runtime che si realizza mediante la co-simulazione [18, 19].

1.3.1 Co-simulazione

La co-simulazione è un ulteriore passo avanti rispetto alla interoperabilità, intesa in senso generale, giacché impone che questo scambio di informazioni avvenga non su dati iniziali/finali dello stato del sistema, ma che avvenga “in continuo” (ad intervalli di tempo prefissati, in genere pari all'intervallo di simulazione) durante l'esecuzione della simulazione del sistema. La co-simulazione risulta totalmente inefficiente se tale scambio di informazione avviene tramite la lettura/scrittura di file di interscambio, mentre può avere una certa efficacia se lo scambio avviene a livello di accesso di memoria RAM condivisa. La formalizzazione di tale tipo di co-simulazione potrebbe indurre una serie di implicazioni positive quali consentire un calcolo distribuito più semplice, responsabilità distribuite nello sviluppo di parti di codice, evoluzione esponenziale delle routine algebriche, garantendo al contempo accuratezza, affidabilità, poco tempo di esecuzione e facilità d'uso. Di contro, i problemi iniziano quando la suddivisione del problema da gestire non riguarda domini interagenti ma diversi (ad esempio, l'involucro da una parte e l'impianto dall'altra), ma la suddivisione in sottodomini di un unico dominio fisico tra risolutori diversi. Questa divisione porta necessariamente a un processo iterativo, spesso completamente senza controllo, perché nessuno dei risolutori coinvolti (i diversi programmi interagenti) ha alcuna conoscenza delle modalità con cui gli altri eseguono i calcoli. Tale occorrenza porta spesso ad un notevole incremento dei tempi di calcolo.

In sintesi, la co-simulazione comporterebbe diversi vantaggi, quali:

- non impone di reinventare la ruota, cioè quando occorre estendere le capacità di un dato programma, non occorre sviluppare l'aspetto relativo, ma basta “collegarsi” ed utilizzare un software specifico già sviluppato;
- dovrebbe essere praticamente “no time-consuming” nella sua applicazione;
- sfrutta gli sviluppi che si verificano in diversi strumenti accessori che si potrebbero utilizzare;
- promuove il mercato liberale di numerosi strumenti diversi, ecc.

Mentre le conseguenze di questo approccio comportano:

- difficoltà nella risoluzione numerica dei problemi accoppiati;
- modifiche al codice esistente per consentirla;
- mancanza di controllo sui modelli esterni utilizzati;
- problemi legati al miglioramento degli strumenti;
- la necessità di essere sempre informati su quale strumento ha guadagnato alcuni “punti” in ciascun calcolo in relazione agli altri.

1.3.2 Modifica del codice

Un esempio di come si stia cercando di far dialogare un sistema monolitico, fortemente specializzato, come quelli di ESP-r e di EnergyPlus, con altri modelli è rappresentato dallo sviluppo di un nuovo gruppo di elementi interni ai codici monolitici, come le “ExternalInterface” in EnergyPlus. L'inclusione di tali elementi, la cui gestione non è ancora del tutto trasparente per l'utente, ha richiesto modifiche al codice di EnergyPlus e l'eventuale introduzione di un gestore per lo scambio di informazioni fra i diversi agenti coinvolti, quale il Building Controls Virtual Test Bed (BCVTB), che può essere evitato nel caso si utilizzi la funzionalità di interfaccia esterna di importazione di Functional Mock-up Units, FMU, prevista in EnergyPlus. In entrambi i casi è fornita la possibilità di modificare alcune delle variabili interne di un software monolitico, assegnando loro valori calcolati esternamente. Per il momento, l'uso di questa opportunità ha riguardato principalmente problemi risolvibili mediante “loose-coupling”, ovvero soluzioni per le quali non fosse “strettamente

necessaria” una interazione di natura iterativa fra i diversi attori. In particolare, in EnergyPlus oggi l'interfaccia esterna è definita per tre tipologie di input:

- ExternalInterface:Schedule, o ExternalInterface:FunctionalMockupUnitImport:To:Schedule;
- ExternalInterface:Actuator o ExternalInterface:FunctionalMockupUnitImport:To:Actuator;
- ExternalInterface:Variable o ExternalInterface:FunctionalMockupUnitImport:To:Variable.

Allo stesso modo sono state definite analoghe interfacce per gli output verso l'esterno:

- Output:Variable;
- EnergyManagementSystem: OutputVariable;
- ExternalInterface:FunctionalMockupUnitImport:From:Variable

Mentre la funzione delle prime due interfacce di input è di immediata comprensione e se ne comprende la natura di “controllore-attuatore”, l'ultima potrebbe sembrare di natura più generale, ma in realtà ha una funzionalità simile alla EnergyManagementSystem:GlobalVariable (oggetto interno ad EnergyPlus per la gestione degli impianti), ovvero quella di passare dei valori a delle variabili di uno specifico linguaggio di modellazione dei sistemi di controllo e gestione interno ad EnergyPlus, chiamato Energy Runtime Language (Erl). Il valore numerico passato ad EnergyPlus dall'interfaccia esterna all'inizio di ogni passo temporale per le zone termiche, rimane tuttavia costante all'interno di questo timestep (passo di integrazione temporale), e quindi non può essere modificato in seguito a iterazioni compiute all'interno del timestep stesso. Una strategia di questo tipo potrebbe rivelarsi in alcuni casi non efficiente in termini computazionali e implementativi, mentre lo sviluppo di un software special-purpose che sia stato implementato secondo una “modularità arricchita” [20], grazie ai costrutti della programmazione a oggetti, consentirebbe di individuare facilmente la parte di codice che si intende sostituire o estendere, mantenendone la “collocazione relativa” rispetto ad altre procedure di calcolo e consentendo di ereditarne le funzionalità, qualora fosse funzionale allo scopo.

1.3.3 Uso inappropriato

Un'altra preoccupazione relativa all'accoppiamento di diversi strumenti riguarda il rischio di un uso inappropriato di modelli che nascevano su alcuni assunti, magari dimenticati negli anni. Spesso infatti alcuni dei programmi che sono stati modificati negli anni da sviluppatori diversi contengono soluzioni soddisfacenti per un determinato momento storico o che rappresentano l'unica possibilità conosciuta all'epoca. Ad esempio, oggi EnergyPlus impone un timestep di calcolo per l'edificio di massimo 3 minuti per il metodo alle differenze finite per la risoluzione dello scambio termico per conduzione all'interno delle pareti, che non si coniuga bene con i tempi di risposta di alcuni impianti e più in generale con un modello stazionario per alcuni degli elementi, soprattutto impiantistici, che si trovano a dialogare con esso. Una modellazione stazionaria degli impianti andava bene fino a quando il passo di discretizzazione temporale era dell'ordine di un'ora e, quindi, è un problema il fatto che nel programma non siano stati implementati messaggi di errore nel momento in cui il passo temporale è inferiore a un tempo di reazione agli stimoli caratteristico di quei modelli descritti in regime stazionario. L'accoppiamento di diversi strumenti non deve mai essere desiderabile solo perché realizzabile, ma deve seguire un'attenta analisi degli strumenti che si desidera accoppiare.

La fase di risemina (reseeding) servirebbe proprio a razionalizzare quanto sviluppato in un lungo periodo, verificandone appropriatezza e consistenza e ricombinandone le diverse parti.

1.4 Elementi chiave per la scelta di un BPSt: la modularità e la community.

Dovendo scegliere un programma di simulazione dinamica da usare per progettare edifici, in modo che siano il più efficienti possibile e in grado di sfruttare al massimo la disponibilità di energia da fonte rinnovabile, edifici NZEB, occorre analizzare le possibilità offerte dai vari software in termini di *accuratezza* e *completezza*, pur tenendo in considerazione facilità di uso e strumenti di supporto alla progettazione eventualmente inclusi.

Purtroppo, per quanto riguarda l'*accuratezza* di un software BPSt, la sua valutazione è ancora un passaggio difficile. Studi comparativi sui software esistenti [10, 11] dimostrano che ancora oggi esistono differenze significative fra i risultati ottenuti mediante i diversi programmi, soprattutto per quanto riguarda la valutazione di variabili ritenute in passato di secondaria importanza. Inoltre, non sempre vengono forniti con i vari software tutti gli strumenti utili per giudicare l'*accuratezza* dei risultati da questi producibili, quali file di input per il software che descrivono il modello delineato nella procedura di valutazione e report "già pronti" contenenti le necessarie elaborazioni dei risultati ottenuti durante il processo di validazione. Di certo, lo scostamento fra previsione e realtà operativa è figlio di un'aleatorietà delle forzanti del sistema che non può essere rappresentata da un singolo scenario e la conoscenza dell'*accuratezza* raggiunta in ogni singolo scenario costituisce solo la base di partenza per poterne considerare molti e per poter valutare le prestazioni del sistema progettato nelle diverse circostanze.

Data la difficoltà nel perseguire contemporaneamente obiettivi potenzialmente in contrasto fra loro, quali *operatività* e *adeguatezza*, si sono moltiplicati gli sforzi di sviluppo dei software di simulazione, arrivando a presentare sul mercato soluzioni molto diverse fra loro.

Nuovi programmi general-purpose hanno cercato di rispondere all'esigenza di rapido sviluppo di nuovi modelli e di accoppiamento fra i modelli realizzati per la simulazione delle diverse parti, mentre molti dei software special-purpose di nuova implementazione hanno puntato sulla facilità di utilizzo e sulla riduzione della complessità del sistema simulato. In risposta alle difficoltà di modifica dei più vecchi programmi special-purpose, ci si è rivolti alla co-simulazione. Tuttavia, non sempre ciò che è accoppiabile dovrebbe esserlo o lo è efficientemente da un punto di vista computazionale e implementativo, per cui si consiglia cautela nell'uso di tali soluzioni.

In conclusione, occorre mettere in evidenza che quello che oggi rende facilmente mantenibile ed estendibile un codice di calcolo non è l'essere general-purpose o special-purpose, ma l'essere stato progettato con un'*architettura di tipo modulare*. Questa, come verrà mostrato chiaramente più avanti, consente, se ben progettata, di aggiornare, sostituire e aggiungere funzionalità ad un codice in modo semplice e naturale.

È importante, infine, ricordare che nessun BPSt può compensare una scarsa competenza dell'utente nel creare un corretto modello rappresentativo del sistema che si vuole simulare. Quindi, allo sviluppo di un codice di calcolo accurato e completo, occorre comunque associare una crescita culturale specifica dei suoi utilizzatori. Questa può essere realizzata tramite una classica formazione basata su un approccio puramente cognitivo della materia, oppure su un approccio pratico e collaborativo, tramite la condivisione di esperienze applicative nell'ambito di una comunità di sviluppatori ed utenti: la "*community*"

2 Individuazione dei requisiti per un nuovo codice di calcolo

Il passaggio da interfacce alfanumeriche a interfacce utente grafiche (GUI) guidate da eventi, l'introduzione di architetture client-server multi-layer, database distribuiti, Internet, la drastica riduzione dei prezzi del PC e il conseguente miglioramento della potenza di calcolo, pur consentendo soluzioni un tempo incredibili, ha causato un notevole aumento della complessità e dei rischi connessi allo sviluppo ed uso del software, come ad esempio:

- requisiti mal definiti;
- requisiti in rapida evoluzione;
- comunicazione difficile tra i ricercatori dell'IT e altri ricercatori;
- l'enorme sfida di mantenere i sistemi esistenti senza documentazione.

Di conseguenza, il processo di sviluppo di un nuovo software per la simulazione dinamica delle prestazioni degli edifici deve affrontare adeguatamente ciascuna delle sue fasi:

- analisi,
- progettazione,
- codifica,
- test,
- implementazione.

Esistono principalmente tre modelli di processo del ciclo di vita di un software: il modello a cascata, il modello iterativo e il modello incrementale-iterativo.

Il modello incrementale-iterativo (Figura 2) è quello più adatto a un sistema particolarmente complesso: secondo tale modello, il sistema viene diviso in sottosistemi, ad ognuno dei quali si applica il processo iterativo.

Tuttavia, la biologia dice che i sistemi complessi e naturali non vengono creati tutti in una volta ma devono evolversi nel tempo [21]. Pertanto, al fine di garantire nel tempo l'utilità di un software, la tradizionale operazione di "manutenzione" non è sufficiente, ma sono necessarie attività di miglioramento, oltre riparare i difetti e correggere i bug.

Come ha rilevato Fischer, è necessario seguire il modello di semina, crescita evolutiva e risemina (SER) [22]. Questo modello postula che, al fine di mantenere sempre attuali strumenti come i BPSt, *"le attività di evoluzione non pianificata e i periodi di (ri)strutturazione e valorizzazione programmati devono alternarsi continuamente"*.

Di conseguenza, gli strumenti BPSt devono:

- evolversi, poiché non possono essere progettati in modo completo prima dell'uso;
- evolvere per mano degli utenti, grazie a strumenti che consentono loro di modificare funzionalità e contenuti;
- devono essere progettato per poter evolvere.

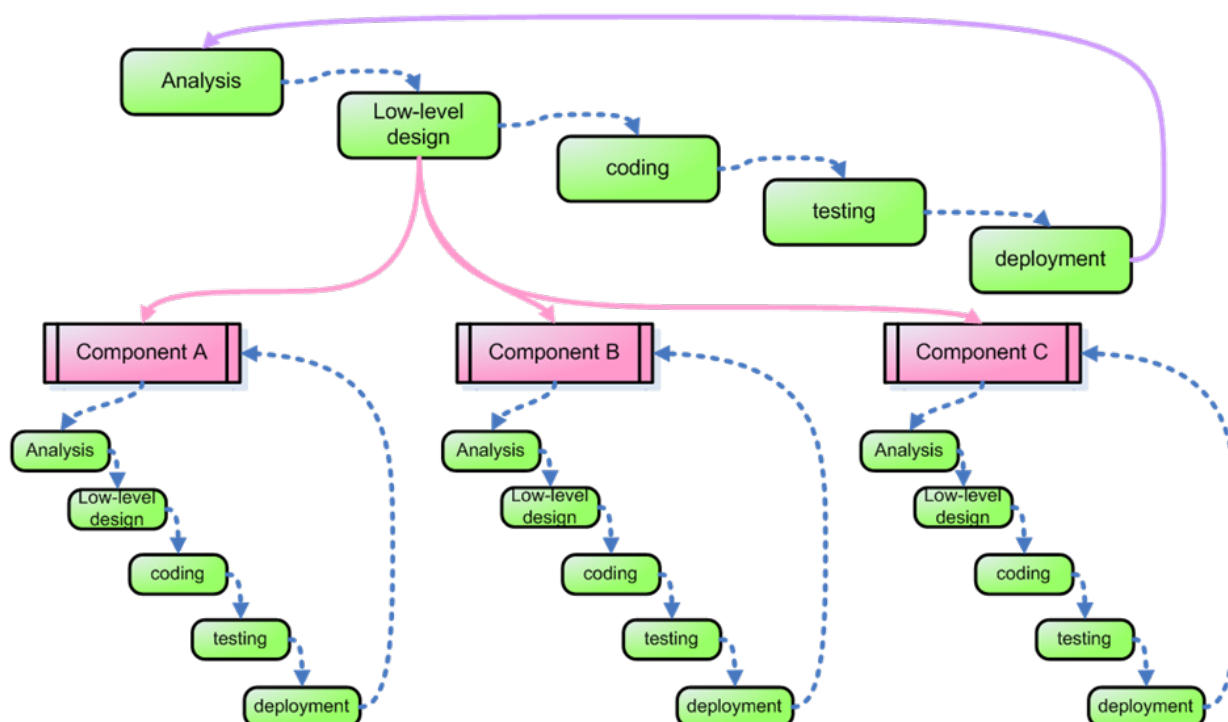


Figura 2 – Il metodo iterativo incrementale.

Occorre che periodicamente lo spazio dell'informazione (una combinazione di annotazioni, progetti parziali e discussioni), che è in continua crescita, venga (ri)strutturato, generalizzato e formalizzato. Questa fase di risemina (reseeding), uno sforzo programmatico di revisione e coordinamento di informazioni e funzionalità, costringe gli sviluppatori IT (i programmatori) a collaborare con i progettisti dello specifico settore di interesse (l'involucro, gli impianti, ecc.). Questa fase di risemina può quindi beneficiare della disponibilità di nuovi linguaggi di programmazione e framework, con funzionalità avanzate. Ad esempio, l'impiego dei diagrammi delle classi può fornire immediatamente, con un'immagine, molte informazioni altrimenti difficilmente sintetizzabili. L'architettura del software è resa ancora più chiara ed evidente, grazie al modo in cui sono rappresentate le classi, le strutture e le interfacce o la relazione di eredità e associazione. Questa caratteristica è cruciale per quanto riguarda la comunicazione dei concetti alla base del codice, l'identificazione dei problemi e la comprensione del codice esistente. Con i diagrammi di classe si può pervenire alla comprensione del modello con una semplice visione alternativa (strutturata a blocchi logici) del codice sorgente. Le modifiche apportate al diagramma di classe si riflettono immediatamente nel codice e viceversa. Strumenti come Visual Studio Class Designer consentono di visualizzare la struttura delle classi e le loro relazioni, creare nuove classi o interfacce utilizzando un ambiente di progettazione visuale, riformattare facilmente le classi e modificare direttamente il loro codice sorgente, con un significativo risparmio di tempo e linee di codice per gli sviluppatori.

2.1 Problemi di comunicazione: una nuova figura professionale interdisciplinare

Nel processo di sviluppo del software, l'interazione più critica è quella tra i ricercatori, che definiscono i modelli matematici e gli sviluppatori di applicazioni. Nella tesi di dottorato di Sahlin [23] si afferma che *"Sembra che il meccanismo dominante di trasferimento tecnologico tra questi attori sia quello di ricercatori che non cambiano professione e diventano invece sviluppatori di software. Un problema pratico è la forma in cui i modelli vengono descritti quando vengono trasferiti da modellisti matematici agli sviluppatori di applicazioni."* Forse i ricercatori, modellisti dei processi fisici, dovrebbero acquisire alcune informazioni sull'IT, almeno rispetto a quegli strumenti, come i diagrammi UML, che, oltre ad essere già una struttura disponibile, è stata sviluppata nel campo dell'ingegneria del software con l'obiettivo di risolvere la

comunicazione dei concetti e mostrare problemi "nascosti". Forse è necessaria una nuova figura professionale, come è già accaduto per la meccatronica, dove è stata finalmente riconosciuta l'importanza di avere un soggetto unico che riassume in se due diverse conoscenze, la meccanica e l'elettronica.

2.2 Modularità del software

Essendo un BPSt un soggetto interdisciplinare, una volta definiti i ruoli dell'interfaccia utente-codice (GUI), l'obiettivo principale dovrebbe essere quello di progettare il codice secondo un principio di modularità che consenta a ciascun esperto di sviluppare facilmente le funzionalità in cui è specializzato, in modo che ogni attore coinvolto svolga compiti a lui congegnali.

Tuttavia, ci sono diversi aspetti della modularità collegati ai diversi livelli di sviluppo delle funzionalità degli strumenti informatici e non impiegati:

- numerici,
- matematici,
- operativi
- evolutivi.

Limitare la necessità della modularità alla sola possibilità di riutilizzo di parte del codice sorgente, precedentemente sviluppato, per i nuovi moduli è un po' riduttivo. Occorre invece comprendere qual è la vera natura della modularità del software e il suo obiettivo.

Possiamo definire un sistema modulare come una struttura in cui l'operazione di inclusione di un nuovo modello è semplice, non richiede la modifica dell'intero sistema e non influisce, in modo negativo, sulle caratteristiche di partenza del sistema, come precisione, robustezza, usabilità, eccetera.

Per quanto riguarda l'aspetto numerico della modularità, si possono fare le stesse considerazioni fatte per la co-simulazione, cioè possono sussistere problemi analoghi ai problemi di convergenza della co-simulazione. Come riportato da Wetter nella sua tesi di dottorato [24], la simulazione del carico termico di riscaldamento e raffreddamento di un edificio e il controllo dell'illuminazione diurna, in EnergyPlus, implica l'uso di *"una decina di solutori che controllano i sottosistemi del modello di simulazione"*. Questa struttura impedisce l'analisi della propagazione degli errori di approssimazione, portando a possibili instabilità. Di conseguenza, non si può dire che questi strumenti siano modulari dal punto di vista numerico, in quanto l'unione di diversi moduli potrebbe modificare l'accuratezza e la stabilità del codice di partenza.

Quindi, quando una struttura è modulare in senso numerico?

Probabilmente, da questo punto di vista, è più modulare un codice che implementa per la soluzione una matrice globale, alla quale si possono aggiungere nuovi modelli matematici e algoritmi per il calcolo parallelo distribuito (in grado di mantenere l'accuratezza del tutto), come quello di IDA/ICE.

Solo una profonda comprensione della natura dei domini fisici da implementare e dei risultati ottenibili nell'analisi numerica, nel calcolo parallelo non lineare, nella potenza di calcolo dell'hardware, ecc., possono aiutare a identificare quali sono le esigenze, o meglio i requisiti, legate alla modularità numerica.

Dal punto di vista dei modelli matematici, la modularità implica che la struttura del codice consenta o è già predisposta per l'implementazione di nuovi modelli, senza richiedere modifiche al codice stesso.

Con l'aspetto operativo della modularità, intendiamo la possibilità che l'utente crei la propria serie di funzionalità, o scenari d'uso, mentre con l'aspetto evolutivo della modularità, intendiamo la possibilità di sviluppare nuovi moduli (senza implicazioni matematiche particolari) che richiedono la scrittura di alcuni codici, senza la necessità di essere esperti IT. Tipicamente, tali tipi di modularità non impattano sulla definizione del processo di simulazione, ma sono correlati al suo uso.

Per quanto riguarda il significato generale della modularità, al fine di realizzare strumenti "modulari", un sistema software dovrebbe avere le seguenti caratteristiche:

- **flessibilità:**
sono consentite modifiche nelle sezioni isolate e limitate del codice e queste non incidono su altre parti del sistema;
- **robustezza:**
le modifiche non portano ad un indebolimento della struttura del codice: ogni modifica interessa solo parti del codice logicamente correlate e non provoca potenziali "crepe" in parti non direttamente correlate;
- **riusabilità:**
è immediato e facile estrarre alcune funzionalità per il loro riutilizzo nell'introduzione di nuove funzionalità.

Per ottenere queste caratteristiche, la struttura del sistema deve avere **basso livello di accoppiamento** (bassa dipendenza di un componente software dagli altri) e **alta coesione** (l'omogeneità di ciascun componente funzionale è elevata e i vari componenti lavorano insieme per ottenere nuove funzionalità ad un livello superiore di complessità). L'elevato livello di accoppiamento può portare a problemi in termini di espandibilità e manutenibilità, mentre la bassa coesione può portare a oggetti che svolgono attività "sparse" e non correlate (vale a dire che troppe "responsabilità" sono state attribuite a quella classe). I criteri per assegnare una responsabilità a una classe in genere non sono riducibili all'idea che coloro che dispongono delle informazioni necessarie per assumersi una specifica responsabilità siano i candidati giusti. Ad esempio, si immagini di dover salvare dei dati in un database. Se ogni classe contenente informazioni da archiviare dovesse salvare i dati sul DB, lo stesso codice verrebbe replicato più volte, invece sarebbe più appropriato assegnare un insieme coerente di responsabilità a una classe "artificiale o di convenienza" che si preoccupi specificatamente di tale compito.

Riassumendo, la struttura di un nuovo BPSt, definita attraverso l'analisi del software esistente, deve:

- consentire all'utente di comprendere meglio il sistema in analisi;
- essere facilmente leggibile (ad es. tramite diagrammi di classe), per facilitare la decisione se un modulo è adatto o se deve essere modificato/adattato agli obiettivi specifici richiesti;
- essere facilmente estensibile;
- consentire il riutilizzo
- consentire lo scambio di moduli tra utenti e sviluppatori tramite un repository basato sul web.

Inoltre, a un livello superiore, per perseguire sempre il principio di chiara allocazione delle responsabilità, la "macroarchitettura" dello strumento dovrebbe separare chiaramente, all'interno di un'architettura a tre livelli, il livello di presentazione, il livello di servizio e il livello persistente.

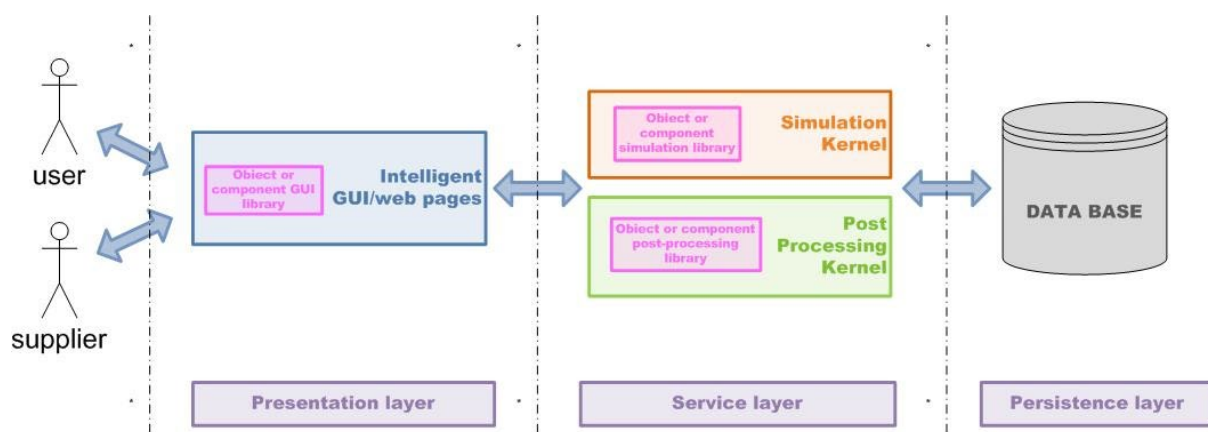


Figura 3 – Macro-architettura dei BPSt.

Siamo in grado di identificare quattro principali macro moduli nel BPSt: una GUI intelligente (interfaccia utente grafica), un kernel di simulazione, un kernel di post-elaborazione e una base di dati.

La GUI intelligente per l'immissione dei dati o la visualizzazione dell'output dovrebbe (seguendo le routine sviluppate dagli esperti, come prescritto dalla crescita evolutiva):

- alleggerire l'inserimento dei dati di input forniti dall'utente;
- essere in grado di adeguarsi alle esigenze del progettista che la sta utilizzando e della progressione delle fasi di progettazione;
- apprendere il modo di progettare di ciascun utente;
- consentire all'utente di scegliere tra alcune predefinite linee di progettazione integrata [25, 26, 27, 28] o modificarle.

Nel kernel di post-elaborazione, oltre alle routine di conversione dei dati, gli sviluppatori dovrebbero implementare la routine per la generazione di diagrammi o algoritmi che mirano a verificare alcune regole di base generali (secondo la politica di distribuzione delle responsabilità), ecc..

2.3 Integrazione nel BIM

Gli strumenti CAD, o la loro evoluzione, le applicazioni BIM, sono oggi molto presenti nella routine di progettazione quotidiana in quanto strumenti ormai indispensabili, affidabili e rispondenti alle esigenze del processo di progettazione, così come inteso nella consuetudine. Offrono una varietà di servizi, si evolvono continuamente con l'obiettivo di ridurre i tempi di progettazione includendo nuove e utili funzionalità che automatizzano o facilitano l'azione di progettazione, ecc.. La tendenza attuale è quella estendere le funzionalità delle applicazioni BIM alla determinazione delle prestazioni energetiche degli edifici, cioè di includere un BPSt nel BIM. Tale inclusione può avvenire con modalità diverse, tramite

- un'interoperabilità limitata allo scambio di dati statici;
- una co-simulazione.

È evidente che, per quanto detto in precedenza, la via da perseguire non è quella di semplicemente "istruire" tramite un BIM il programma di simulazione (di fatti si costruisce il suo file di input e si lancia l'esecuzione del programma BPS), ma è quella della co-simulazione, intendendo in questo caso che il BPSt sia parte integrante del BIM e scambi i dati a livello di mappa di memoria. Tale livello di integrazione presuppone che il BPSt venga sviluppato più come un insieme di librerie dinamiche richiamabili da un programma interfaccia, che in questo caso non è la GUI intelligente, ma lo stesso BIM.

Il principale vantaggio dell'inclusione di un BPSt in un BIM è la maggiore probabilità che questo diventi uno strumento di uso quotidiano nella progettazione, così come lo sta diventando il BIM stesso, soprattutto se il BPSt è stato sviluppato secondo il principio della modularità estesa enunciato. In tal caso, infatti, i diversi modelli di dominio (involucro, impianti, ecc.), più o meno correlati, sono sviluppati e progettati per adattarsi alle esigenze dell'utenza e, eventualmente, modificato dagli stessi utenti finali, mirando a una specializzazione di domini, che non rispondano in modo oscuro a tutte le esigenze, ma ciascuna specifica necessità.

Un altro vantaggio dell'inclusione in un BIM è quello legato allo sviluppo del design collaborativo diffuso, con il sistema di gestione dei file del progetto e lo scambio di informazioni tramite Internet, già presente in diversi applicativi di tale tipo.

2.4 Requisiti e ruolo dell'utente e degli sviluppatori

Al fine di consentire la crescita evolutiva del BPSt, è necessario introdurre o collegare funzionalità avanzate, quali:

- strutture predefinite per lo sviluppo di nuovi modelli di componenti di sistema,
- librerie di modelli e solutori,
- strumenti per la compilazione e la query di database [27],
- strumenti per la personalizzazione dell'interfaccia utente,
- strumenti per il supporto CAD,
- strumenti per attività di ottimizzazione,
- strumenti per la gestione o la presentazione di input e output,
- strumenti a supporto di linguaggi di scripting o servizi web, ecc.

Alcune di queste funzionalità potrebbero essere acquisite tramite l'interoperabilità tra software, altre trarrebbero beneficio da un alto livello di coerenza e quindi dovrebbero essere più profondamente integrate (vedasi la citata integrazione nel BIM). Un esempio che segue questa direzione è il progetto Meta-design (quadro concettuale emergente finalizzato alla creazione di sistemi come entità viventi, al fine di consentire agli utenti di contribuire attivamente allo sviluppo anziché essere consumatori passivi) [15], basato sulla convinzione che l'obiettivo dell'interazione uomo-computer si evolverà dal “facile da usare” (anche se non ancora completamente raggiunto) al “di facile sviluppo”. Alcuni esempi di sviluppo eseguito direttamente degli utenti finali (EUD) sono già presenti in alcuni strumenti odierni, come la registrazione o la scrittura di macro in Visual Basic nei fogli di calcolo Excel, usando lisp per il disegno CAD parametrico [29] o gli script GC per esplorare la relazione dell'oggetto con `GenerativeComponents` [30]. Approcci diversi sono: programmazione per esempi, programmazione visiva, macro e linguaggi di scripting [31]. Come detto in precedenza, ci sarà un diverso livello di sviluppo (modalità di implementazione semplice, modalità di sviluppo avanzata e modalità di sviluppo esperto) e per ogni livello dovrebbero essere individuati e implementati gli strumenti più adeguati.

2.5 Responsabilità distribuite

Un aspetto importante del processo di crescita evolutiva è legato all'arricchimento della libreria dei modelli di simulazione. Questo arricchimento può consistere in un arricchimento compilativo e/o in uno implementativo.

Quello compilativo consiste nell'inserimento di dati relativi a parti di apparecchiature, materiali, ecc. in data base utilizzati per le simulazioni. Tale operazione potrebbe essere fornita, tramite pagine Web opportunamente documentate, direttamente dai vari fornitori/produttori, in modo tale che i loro sistemi possano essere archiviati in data base e utilizzati nelle simulazioni tramite selezione in funzione delle loro caratteristiche. Ciò ridurrebbe l'errore nell'assegnare valori ai parametri delle classi, assegnando la responsabilità di fornire il giusto input direttamente ai produttori. I produttori potrebbero anche essere responsabili della definizione del campo di utilizzo di ciascun componente.

L'arricchimento implementativo è legato invece allo sviluppo di nuove classi, da integrare negli strumenti BPS o, se il modello è proprietario e il sorgente non deve essere diffuso perché coperto da copyright, o da interrogare come servizio web, come ipotizzato da Augenbroe [32], o come librerie dinamiche scaricabili direttamente dal sito del produttore.

2.6 Approccio Open Source

La giusta filosofia di sviluppo per gli strumenti BPS è quella utilizzata da Esp-r ed EnergyPlus, che rendono disponibile a chiunque il codice sorgente. Ciò consente di coinvolgere un insieme allargato di potenziali sviluppatori di modelli di componenti di sistema, e, di conseguenza, il numero di modelli implementati può subire uno sviluppo esponenziale, guidato dagli interessi degli stessi utenti e dall'economia anziché essere

bloccato da una mancanza di risorse intellettuali. In tale scenario è necessario un repository di informazioni basato sul Web per definire le regole di sviluppo, informare sui risultati raggiunti e gestire / documentare il codice già sviluppato. Durante questa fase potrebbe verificarsi un primo tipo di validazione, e in particolare una validazione incrociata. Le informazioni relative alla validazione devono essere ben documentate e chiaramente associate a ciascun componente o modulo presente nel repository.

Per garantire la diffusione di una versione stabile del codice sorgente e/o eseguibile, da collocare in una directory specifica del repository, occorre che ad intervalli prestabiliti si esegua la fase di risemina (reseeding). Tuttavia, poiché la fase di reseeding per essere efficace deve svolgersi con continuità, è necessario individuare un Ente/organizzazione/associazione “inter partes” che possa esserne garante. Un ente pubblico potrebbe avere l’interesse a ricoprire questo ruolo, essendo spesso l’organo di controllo dell'applicazione della legge relativa ai requisiti minimi sulla prestazione energetica, e quindi potenzialmente profondamente coinvolto.

3 Caratteristiche generali del nuovo codice: OpenBPS

OpenBPS è un nuovo programma di simulazione delle prestazioni degli edifici progettato principalmente come libreria software open source multiplatforma (Windows, Mac e Linux). È un motore di simulazione e non esiste un'interfaccia utente formale. A scopo di sviluppo e test, viene fornita una semplice GUI, che è in grado di importare la geometria dell'edificio fornita dal plug-in OpenStudio per SketchUp o di importare direttamente i progetti definiti tramite il file di input EnergyPlus (.idf), anche se non ancora completamente implementati. È codificato con un linguaggio di programmazione orientato agli oggetti multiplatforma, C#, che è un linguaggio open source per .NET Framework basato sugli standard ECMA. Questo linguaggio ha facilmente permesso alla parallelizzazione del codice nativo di sfruttare i processori multi-thread / multi-core oggi disponibili.

3.1 Codice orientato agli oggetti

A seguito dell'analisi condotta sulle caratteristiche dei software di BPS si sono identificate le caratteristiche chiave, per lo sviluppo di un nuovo codice di calcolo che sia all'avanguardia e superi le limitazioni che hanno gli attuali BPSt, in una modularità avanzata e in un approccio opensource. Quindi, uno degli obiettivi principali del progetto OpenBPS è quello di creare una struttura modulare migliorata, rispetto agli standard attuali, che faciliti l'aggiunta di nuove funzionalità e consenta alla libreria di essere utilizzata da qualsiasi programma di hosting. Altro obiettivo primario è poi quello di sviluppare il codice di calcolo in ambito Open Source, per renderne possibile successivamente la massima diffusione. L'insieme di questi due obiettivi ha richiesto di identificare una piattaforma software anch'essa Open Source o, perlomeno, "free-to-use" e, per supportare una modularità avanzata, un linguaggio di programmazione ad oggetti.. L'analisi del "mercato" dell'open source ha condotto alla scelta del binomio framework .NET e C#.

Il framework .NET è un framework di sviluppo software da Microsoft, inizialmente disponibile solo per Windows, che oggi è disponibile anche per i sistemi operativi Linux e iOS. Ciò permette di sviluppare applicazioni cross-platform lavorando solo su una sola piattaforma e di ottenere un codice che sarà ottimizzato "on-the -fly" quando verrà eseguito su qualsiasi altra piattaforma e sull'hardware specifico utilizzato. Infatti, qualsiasi compilatore, specifico ad uno dei linguaggi supportati da .NET, come C#, converte il codice sorgente in un linguaggio intermedio (IL): questo è il codice eseguibile (.exe) dell'applicazione. Quando si esegue il codice .exe, il framework .NET chiama il compilatore JIT (Just-In-Time), che converte l'IL a istruzioni macchina, caricando poi ed eseguendo il codice in istruzioni macchina ottimizzato per hardware specifico.

Un altro motivo per scegliere .NET come ambiente runtime è che .NET è diventato uno standard open-source quando ECMA (ECMA 335) e ISO (ISO / IEC 23271) hanno standardizzato la sua Common Language Infrastructure (CLI). Oggi la Fondazione .NET gestisce la versione open-source, denominato .NET core, che può essere trovata su GitHub. Tale versione open source mentre supporta linguaggio standard C # ECMA, non supporta il linguaggio standard C++ ECMA 372. L'ultima versione di .NET 2015 è supportato da Microsoft su Windows, Linux e Mac OSX come .NET core.

Inoltre, il progetto Mono, sponsorizzato da Xamarin, è una implementazione open source del .NET Framework di Microsoft basata sugli standard ECMA per C # e il Common Language Runtime. Il progetto Mono fornisce .NET core per Windows, Linux e Mac OS e ha sviluppato IDE (MonoDevelop IDE) simile a MS

Visual Studio per coloro che usano tali sistemi operativi. È inoltre disponibile, anche da parte di Microsoft, una versione di MS Visual Studio di uso gratuito, la versione Community.

La scelta del linguaggio di programmazione C#, diventata praticamente obbligatoria se si vuole utilizzare il framework .NET core che consente allo stesso eseguibile di girare su sistemi operativi diversi, è anche supportata dalle considerazioni che seguono.

I principali sviluppatori previsti di OpenPBS sono ingegneri, fisici, matematici, ecc, non specificatamente esperti di linguaggi e tecniche di programmazione, ma più esperti in modellazione numerica dei processi fisici. Quindi il linguaggio di programmazione che assolve meglio al compito richiesto deve essere un linguaggio potente e amichevole, con librerie matematiche disponibili in modo nativo, insieme con la caratteristica di essere uno standard e, possibilmente, un linguaggio open-source. Un altro punto non secondario è che il codice compilato risultante debba essere eseguito il più velocemente possibile su qualsiasi macchina. La scelta è quindi caduta su C #, perché:

- è una implementazione arricchita della programmazione orientata agli oggetti, che include incapsulamento, ereditarietà, polimorfismo e prevalenza del metodo;
- è un linguaggio orientato agli oggetti facile ed efficiente: gli sviluppatori possono tradurre le loro idee / algoritmi per risolvere problemi complessi più facilmente rispetto al C ++;
- è uno dei linguaggi principali che funziona su piattaforme diverse utilizzando .NET framework: è disponibile nei sistemi operativi Windows, Linux e MacOS;
- C# è il linguaggio di programmazione che riflette più direttamente il sottostante Common Language Infrastructure (CLI) di .NET framework;
- è più sicuro rispetto al C ++; le uniche conversioni implicite per impostazione predefinita sono quelle considerate sicure, come ad esempio l'ampliamento di numeri interi; la memoria gestita non può essere liberata esplicitamente; invece, viene gestita automaticamente dalla procedura di "garbage collection";
- può produrre applicazioni che girano veloci come le applicazioni C ++, usando il compilatore Just In Time (JIT), che può regolare con precisione l'ottimizzazione del codice sull'hardware della macchina in esecuzione;
- è oggi un linguaggio di programmazione standard e open source (ECMA-334 e ISO / IEC 23270: 2006).

Nell'Allegato A è riportata un'analisi condotta per confrontare le prestazioni di un codice scritto con C# e con C++, e un'analisi di prestazioni in calcolo seriale e parallelo tra C#, C++ e FORTRAN. Da tale analisi si evince la convenienza all'adozione di C#, che con un minimo di perdita di prestazione di calcolo offre una maggiore semplicità e flessibilità di impegno, portabilità e auto-ottimizzazione.

Una struttura orientata agli oggetti è nativamente modulare e semplifica il riutilizzo di pezzi di codice attraverso il concetto di eredità di classe. Diversi tipi di oggetti hanno spesso una certa quantità di "attributi" e "comportamenti" in comune tra loro. Pareti a cambiamento di fase, pareti traspiranti e pareti ventilate, ad esempio, condividono tutte le caratteristiche delle pareti (geometria, composizione degli strati, orientamento, ecc.). Tuttavia, ciascuno definisce anche funzionalità aggiuntive che le rendono diverse: i muri a cambiamento di fase hanno proprietà del materiale a cambiamento di fase da tenere in considerazione insieme a prestazioni non lineari; le pareti respiratorie hanno proprietà aggiuntive come la porosità e il loro risolutore deve tenere conto dell'avvezione attraverso il materiale poroso delle pareti; le pareti ventilate hanno proprietà di canale aggiuntive che rappresentano la ventilazione forzata / naturale interna alla parete. La programmazione orientata agli oggetti consente alle classi di ereditare lo stato e il comportamento comunemente utilizzati da altre classi. In questo esempio, "Wall" è la superclasse delle classi "pareti a cambiamento di fase", "pareti respiratorie" e "pareti ventilate".

Gli oggetti sono istanze di classe. Ad esempio, la classe "Building" descrive il concetto di cosa sia un edificio. Quando la classe " Building " è specializzata, caricando dati specifici, viene creato l'oggetto, " Building -XYZ", che rappresenta un'istanza particolare della classe Building.

Una struttura orientata agli oggetti aiuta l'esecuzione di calcoli paralleli, che si possono gestire a più livelli: nella esecuzione degli oggetti e/o all'interno del singolo oggetto. Poiché ogni parete di un edificio è un oggetto separato, in ogni fase le sue prestazioni possono essere calcolate parallelamente ad altre pareti (quante dipende da quante CPU sono disponibili) [9]. Inoltre, l'approccio orientato agli oggetti consente di utilizzare intervalli diversi del tempo di integrazione all'interno di ciascun oggetto in modo naturale, consentendo loro di ottimizzare le loro prestazioni indipendentemente dall'intervallo temporale globale della simulazione imposto dall'utente.

Conceptual Development of Enhanced Modular Architecture

Objects Oriented Description of a Building

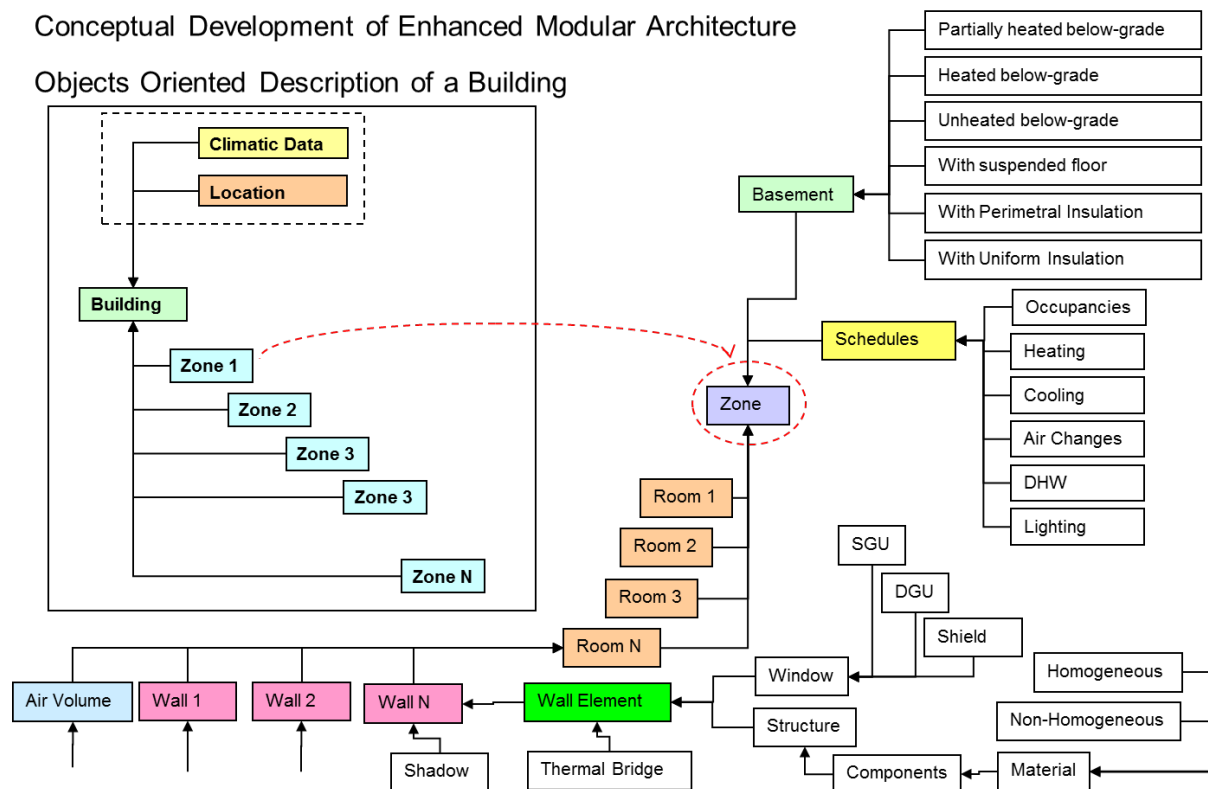


Figura 4 – Struttura modulare migliorata di OpenBPS basata sulle descrizioni orientate agli oggetti dell'edificio.

3.2 Struttura di OpenBPS

La struttura modulare migliorata di OpenBPS, mostrata in Figura 4, segue un'organizzazione gerarchica, portando in cima la classe "Building" che incapsula gerarchicamente tutte le istanze delle altre classi che contribuiscono alla sua definizione. Per implementare tale struttura in un codice gestibile, OpenBPS è stato sviluppato all'interno del framework di sviluppo di MS Visual Studio, e quindi è organizzato in una "Solution" che contiene tutti gli elementi necessari per costruire il suo codice eseguibile, eseguire il debug e testarlo o continuare il suo sviluppo. Questa soluzione comprende diversi "progetti", che sono stati creati per gestire diversi aspetti dello sviluppo del software, in base alla natura modulare dell'intero progetto. Dei progetti inclusi nella nostra Soluzione, uno dà luogo ad un programma eseguibile (exe), altri quattro genereranno ciascuno una libreria dinamica (DLL), mentre gli ultimi due, di supporto allo sviluppo, genereranno "warning" o "report di test".

In particolare, come è possibile vedere nella struttura ad albero presente sulla destra della Figura 5e riportata con maggior dettaglio in Figura 6, abbiamo:

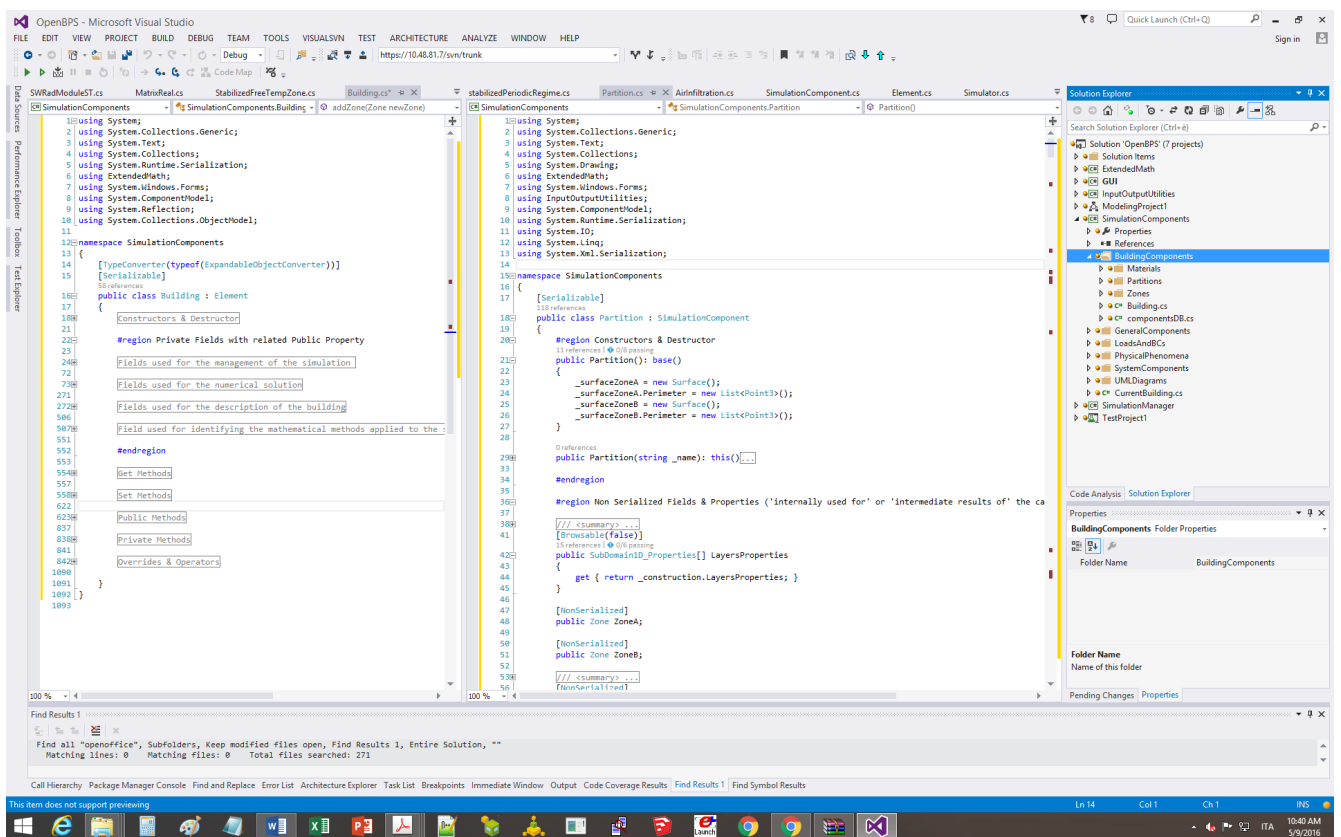


Figura 5 – Ambiente di sviluppo e sotto-strutturazione in “Progetti” della “Soluzione” informatica.

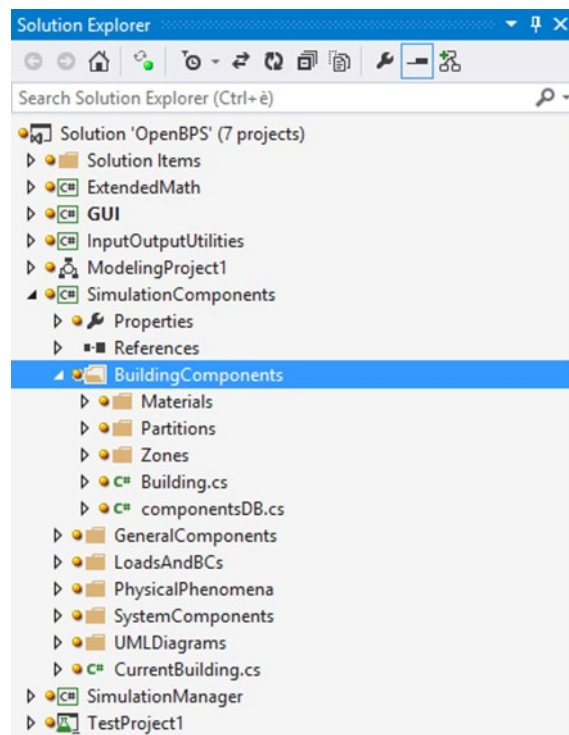


Figura 6 – Struttura modulare della soluzione OpenBPS.

1. un applicativo eseguibile, che costituisce l'interfaccia grafica utente, chiamato **GUI**;
2. quattro progetti di libreria dinamica (dll), identificati come:
 - a. ExtendedMath, contenente gli algoritmi di calcolo usati per la simulazione;
 - b. SimulationManager, contenente l'implementazione del gestore della simulazione;
 - c. SimulationComponents, contenente la descrizione dei componenti costituenti il sistema edificio;
 - d. InputOutputUtilities, contenente gli strumenti per la gestione degli input e degli output;
3. un progetto per la modellazione dell'applicativo: ModelingProject1;
4. un progetto di test dell'applicativo: TestProject1.

Al momento è stato sviluppato un progetto eseguibile (GUI), all'interno del quale è implementata l'interfaccia grafica (Figura 7) per l'uso del kernel di calcolo.

Tale interfaccia consente di importare file di input di EnergyPlus, con alcune limitazioni, o file creati con il plug-in di TRNSYS17 o di OpenStudio in SketchUp, sempre con alcune limitazioni.

È possibile al suo interno usare funzioni di drag-and-drop per assegnare il solutore alle diverse partizioni, o la tipologia di costruzione (strati) delle diverse costruzioni (opache o trasparenti), richiamando le tipologie di costruzione create in precedenza e salvate in un database di progetto.

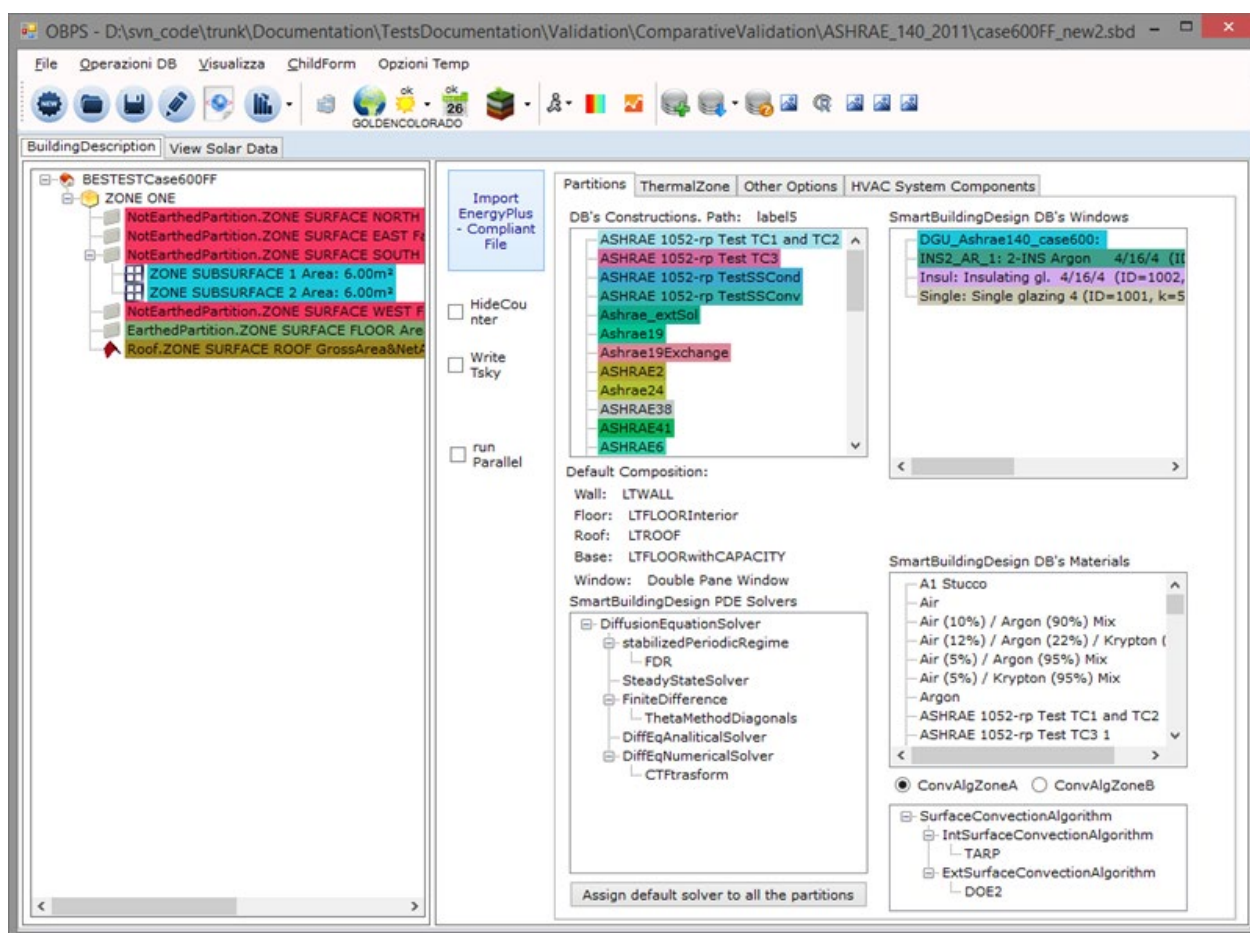


Figura 7 – Interfaccia grafica utente: GUI.

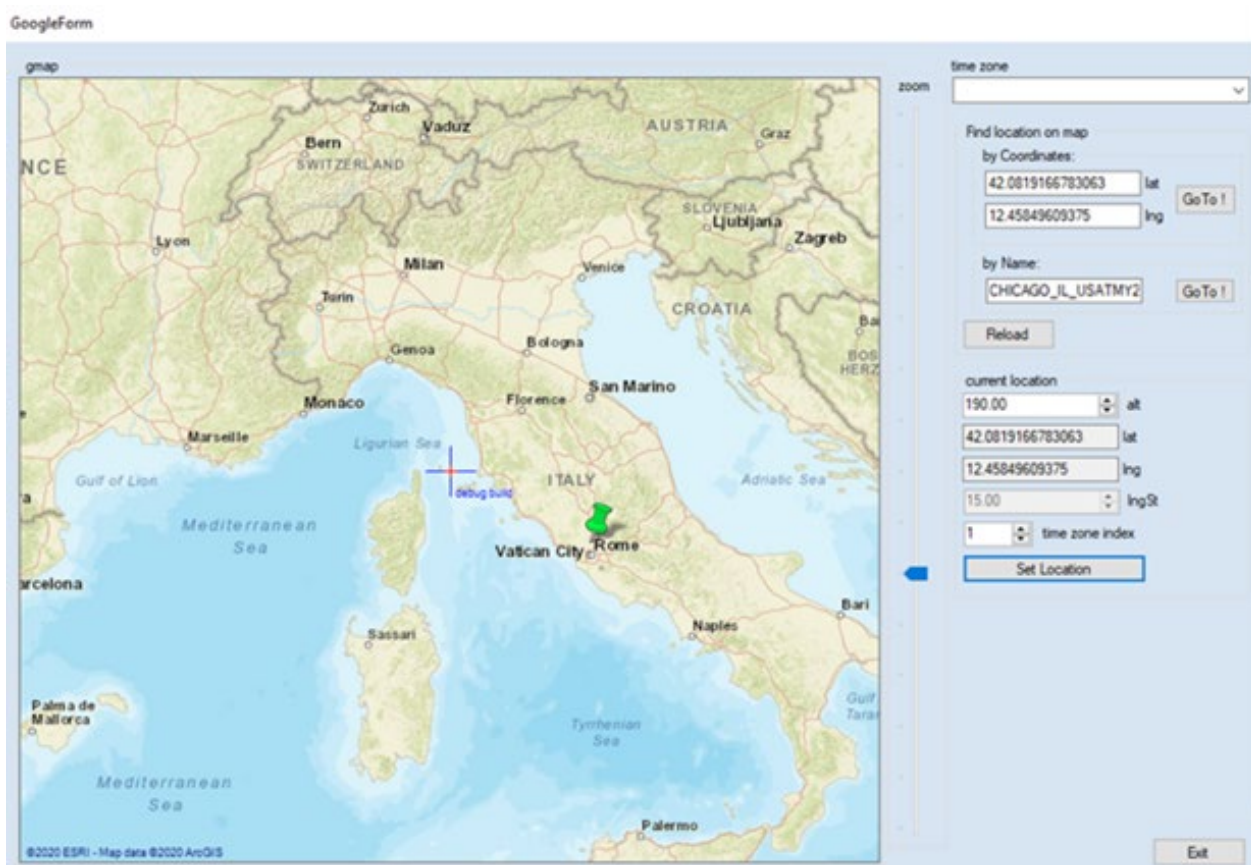


Figura 8 – Strumenti di supporto alla definizione dei dati di input.

Sono stati implementati anche altri supporti per la definizione dei dati di input, come una mappa geografica, che utilizza una libreria legata a GoogleMap, per definire le coordinate della località in cui è ambientata la simulazione (Figura 8).

Se riprendiamo invece le librerie di componente (Figura 2 2), troviamo, ad esempio, all'interno della libreria "SimulationComponent", le strutture per la simulazione degli elementi che costituiscono l'edificio, fra i quali citiamo i più importanti, come:

1. i componenti dell'edificio "BuildingComponents":
 - a. partizioni interne ed esterne, opache e trasparenti
 - b. i volumi d'aria
 - c. i materiali
2. i componenti "LoadsAndBCs", usati per definire i carichi e le condizioni al contorno dei vari componenti costituenti l'edificio, quali schedule, infiltrazioni, apporti interni, ecc.

La filosofia di programmazione scelta per l'implementazione dei modelli descrittivo, matematico e numerico del sistema edificio è quella tipica della programmazione ad oggetti.

La struttura ad oggetti implementata, consente, fra le altre cose, una più agevole:

1. suddivisione delle responsabilità inerenti alle diverse parti di codice;
2. navigazione/comprensione del codice sviluppato;
3. riutilizzo/estensione/manutenzione del codice sviluppato.

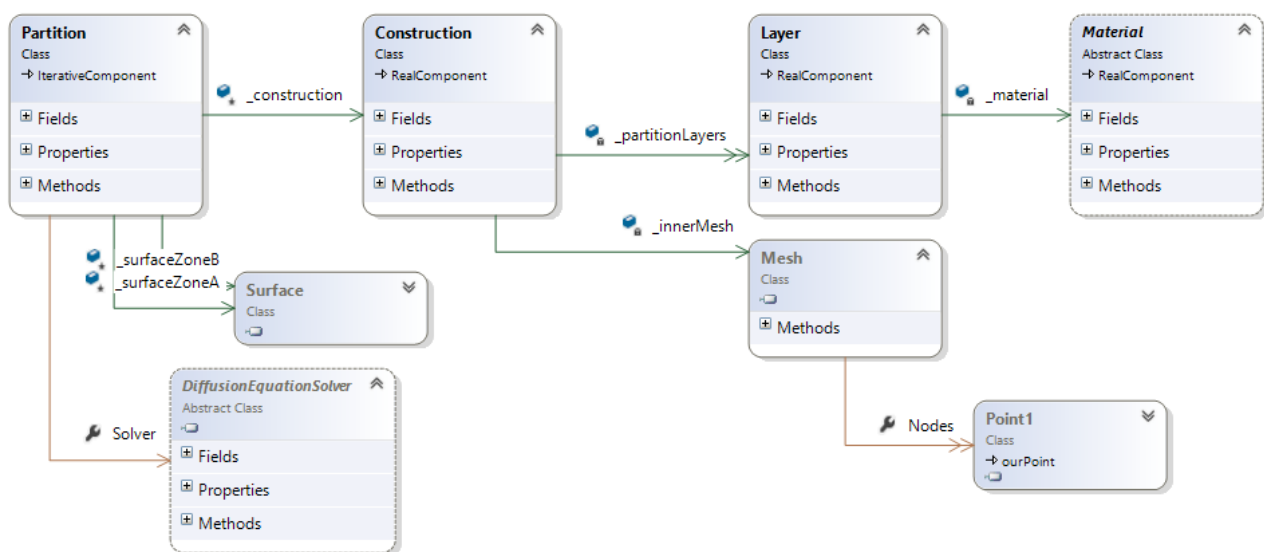


Figura 9 – Struttura Partizione (“partition”).

La suddivisione delle responsabilità all’interno di un modello complesso, è di fondamentale importanza per la facile identificazione di “chi fa cosa”. È infatti importante che ciascuna struttura di codice, racchiusa all’interno di una “classe”, sia identificabile con una “funzionalità” specifica, in modo da essere facilmente identificabile ed il più facilmente possibile “sostituibile/modificabile”, senza la necessità di agire su più parti di codice. Questo consente una facile comprensione, identificazione e manutenzione di quanto sviluppato o ancora da sviluppare.

Un esempio applicativo di tale suddivisione delle responsabilità è visibile dal diagramma di classe di una partizione, come riportato in Figura 9, laddove le frecce uscenti dalla “classe” “Partition” indicano che la partizione “possiede” dei componenti che si occupano di gestire i diversi aspetti descrittivi o numerici funzionali alla sua simulazione.

La partizione ha infatti la responsabilità di contenere al suo interno informazioni relative alla sua collocazione spaziale, alle parti da cui è composta ed al solutore numerico che si intende usare per la sua simulazione dinamica. È responsabilità del solutore numerico quella di eseguire il calcolo numerico necessario alla sua simulazione e della “Construction”, che può essere la stessa per diverse partizioni, quella di gestire i diversi strati e materiali di cui è composta la partizione.

Oltre all’interfaccia grafica ed alle librerie di componenti o strumenti di sviluppo, sono presenti nella Soluzione due progetti di supporto allo sviluppo: il progetto di modellazione e quello di test.

Tali progetti sono rispettivamente funzionali:

1. alla definizione delle relazioni implementate ed implementabili, secondo quanto definito dalle strategie di sviluppo
2. alla fase di test del codice.

Infatti tramite il progetto di modellazione “ModellingProject1” è possibile definire quali interrelazioni fra le diverse librerie siano consentite e quali non lo siano, laddove tale identificazione è stata formalizzata allo scopo di realizzare una migliore strutturazione del codice (Figura 10).

Se qualche sviluppatore dovesse implementare delle interdipendenze che non rispettano tali prescrizioni, il progetto di modellazione bloccherebbe la compilazione del progetto.

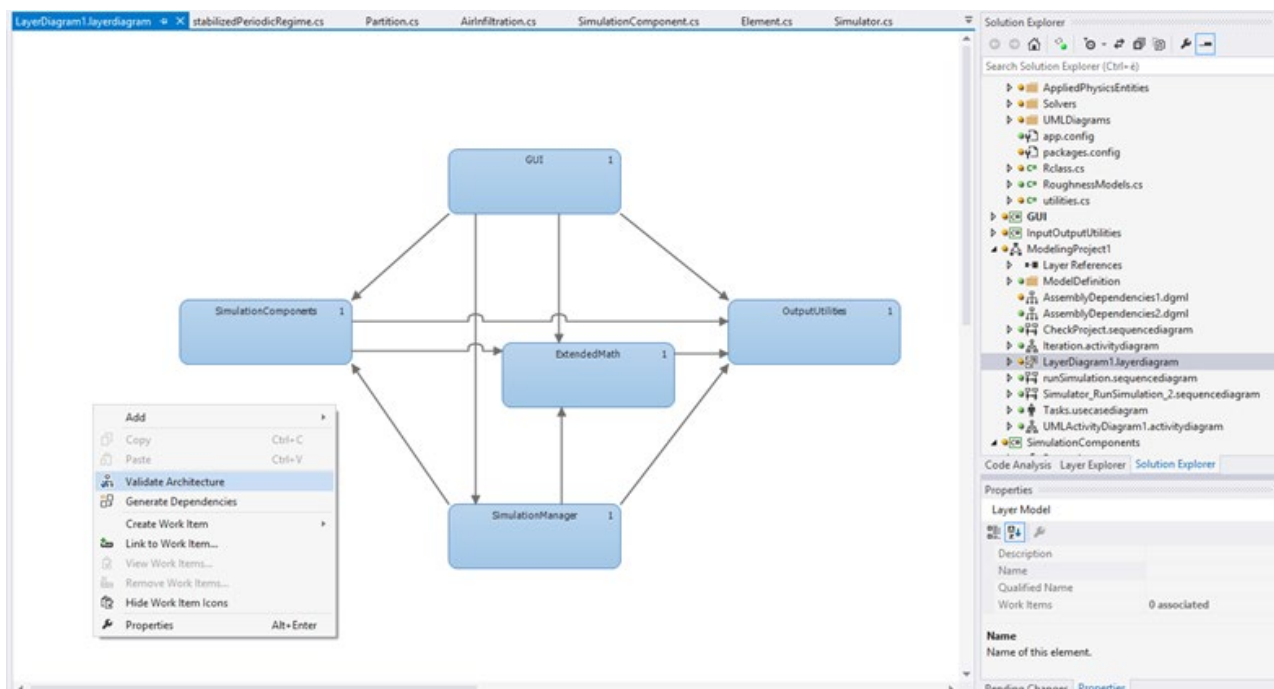


Figura 10 – Validazione architetturale

Il progetto di test, “TestProject1”, è invece funzionale a eseguire in automatico diversi test di parti di codice per i quali siano stati definiti input e output attesi. Questo progetto è utile ad un continuo controllo delle funzionalità del codice, in quando, in seguito a modifiche, i test definiti al suo interno devono essere ancora verificati.

3.2.1 Gestione della simulazione

Tra le quattro librerie principali, che costituiscono il motore di simulazione di OpenBPS, il progetto di `SimulationManager`, contiene il codice del simulatore. Questo è l'oggetto di livello superiore che gestisce la simulazione, controllando le interazioni tra tutti gli oggetti che costituiscono l'edificio. A causa della natura orientata agli oggetti dell'intero codice, la gestione dei componenti viene gestita tramite iterazioni fino a raggiungere la convergenza all'interno di ogni intervallo temporale globale della simulazione. Questo approccio consente di separare l'intervallo temporale globale della simulazione (l'intervallo temporale utilizzato dal Simulatore per aggiornare le sue informazioni) da ciascun intervallo temporale locale che può essere utilizzato da ogni singolo oggetto. In questo modo, è anche possibile implementare in ciascun oggetto strategie che impediscono all'utente di forzare tale componente a lavorare con un intervallo temporale incoerente (rispetto al tempo caratteristico di quel componente). Fondamentalmente, il Simulatore organizza e sincronizza il lavoro di tutti gli oggetti, istruendoli a intraprendere azioni come inizializzare, simulare, esportare risultati, ecc. Può gestire da intervalli sub-orari fino a diverse ore in un periodo di simulazione che va da un giorno a diversi anni.

Il gestore della simulazione, concentrando in un oggetto tutte le regole di gestione, consente di:

- definire meglio le priorità nell'esecuzione degli oggetti;
- allocare la parallelizzazione a un gruppo specifico di oggetti, che può beneficiare più di altri di tale tecnica;
- controllare costantemente il flusso di lavoro;
- aggiungere facilmente nuove classi e oggetti.

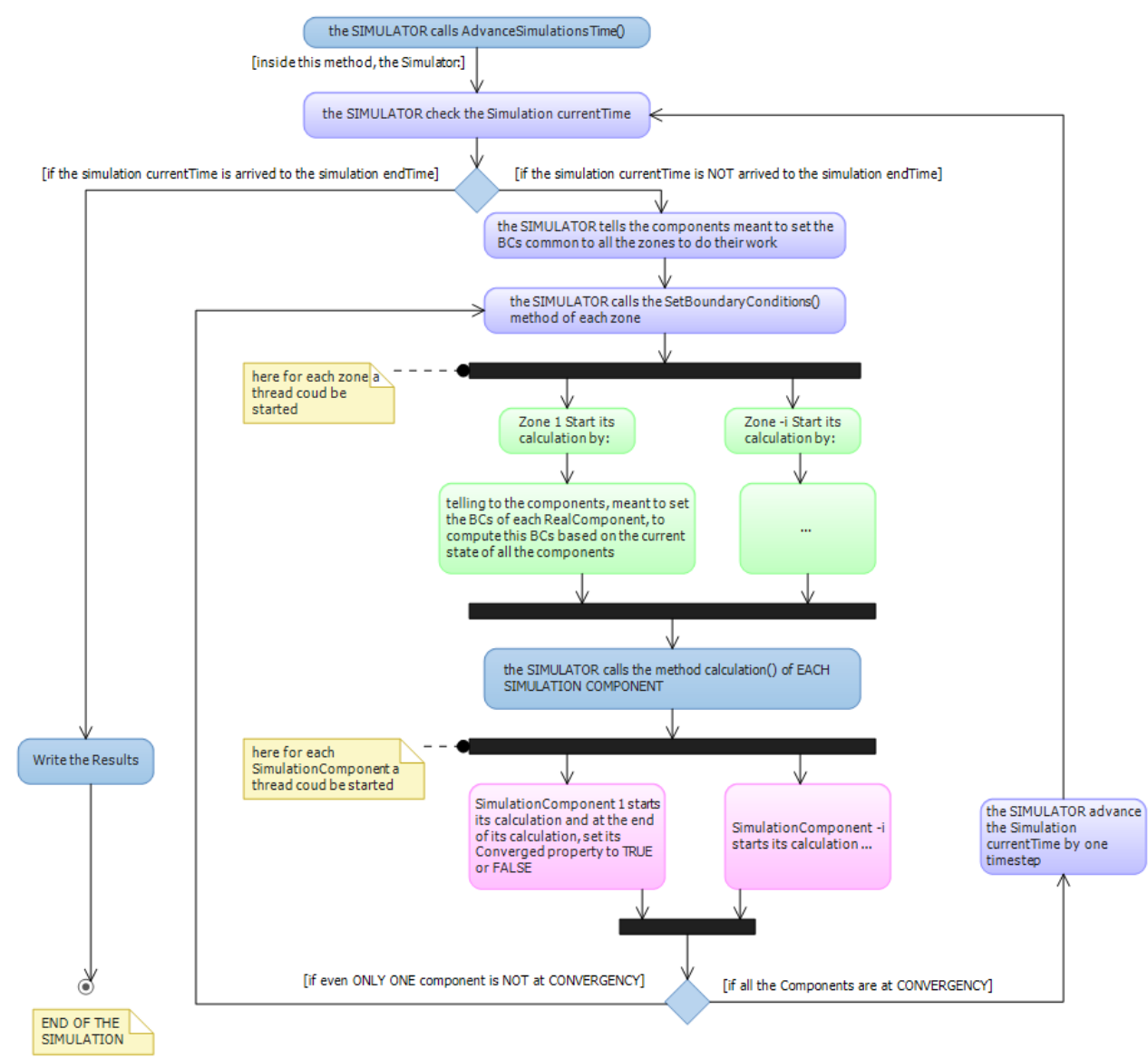


Figura 11 – OpenBPS: gestione del parallelismo a diversi livelli, schema semplificato.

In Figura 11 uno schema semplificato mostra come vengono applicati all'edificio i calcoli paralleli: una prima parallelizzazione viene applicata tra le zone, data la priorità di alcuni calcoli al loro livello, e una seconda parallelizzazione viene applicata tra tutti gli altri componenti di simulazione, coinvolti nel calcolo.

3.2.2 Bilancio energetico e di massa

Il metodo di calcolo dello stato termodinamico della zona termica dell'edificio in OpenBPS è basato su un'equazione di bilancio entalpico integrale, in cui l'aria della zona viene modellata in base all'ipotesi di miscelamento perfetto, ovvero con l'assunzione di una temperatura uniforme in tutto lo spazio della zona[33]. La struttura orientata agli oggetti di OpenBPS consente ovviamente calcoli più dettagliati del semplice impiego del modello basato sui coefficienti di scambio termico convettivo per descrivere i processi convettivi, come i metodi CFD o zonali, che possono essere aggiunti in futuro sviluppo. La parte "sensibile"

dell'equilibrio entalpico costituisce la cosiddetta equazione differenziale ordinaria del "nodo aria", qui semplicemente definita oggetto `Air-Node`.

L'oggetto `Air-Node` gestisce i vari flussi di massa come l'aria di ventilazione, l'aria di espulsione e le infiltrazioni, oltre al trasferimento di calore convettivo con le superfici della stanza (pareti, finestre, soffitti e pavimenti), assunte avere temperatura superficiale uniforme. Tiene conto della capacità termica dell'aria ambiente e valuta direttamente i guadagni termici convettivi dovuti a persone e apparecchiature.

Lo scambio termico attraverso ciascun componente della struttura dell'edificio, qui semplicemente chiamato "partizione", determina le temperature delle superfici della stanza utilizzate nell'oggetto `Air-Node` per calcolare lo scambio termico convettivo. La classe `Partition` consente utilizzare metodi numerici diversi (risolutori) per integrare l'equazione differenziale del secondo ordine che descrive la conduzione termica: metodo delle differenze finite; funzioni di trasferimento per la conduzione e quadrupolo armonico (a fini di test comparativi). Quando viene eseguita una simulazione, qualsiasi oggetto `Partition` può usare un risolutore diverso, in base alle sue esigenze (cioè una parete può essere modellata usando uno schema esplicito alle differenze finite, un'altra il completamente implicito e un'altra la funzione di trasferimento per la conduzione).

3.2.3 Scambio termico radiativo

Lo scambio termico radiativo interno a onde lunghe tra le superfici della zona è modellato in modo dettagliato utilizzando il modello di corpo grigio basato sui fattori di radiazione mutua, che sono calcolati nella fase di inizializzazione tramite i fattori di forma (definiti geometricamente) e le emissività delle superfici. Lo scambio termico radiativo tra tutte le superfici interne viene calcolato ogniqualvolta vengano aggiornate le condizioni al contorno (BC) di ciascun oggetto (componente). Quando le BC vengono aggiornate, viene determinata anche l'irradiazione ad onde corte sia per le superfici di partizione esterne che interne, attraverso il processore di radiazione solare e il modulo di radiazione ad onde corte (`SWRadModule`). È possibile eseguire anche un calcolo semplificato utilizzando i coefficienti superficiali di scambio termico radiativo.

Questo modulo ha la responsabilità di:

- impostare i perimetri di ombreggiamento sulle superfici esterne prendendo in considerazione ostruzioni esterne, come altri edifici, auto-ombreggiature, sporgenze e alette (senza, tuttavia, considerarle anche come possibili riflettori diretti e diffusi)
- impostare la radiazione solare trasmessa attraverso i componenti trasparenti dell'involucro e distribuirla in modo non uniforme tra le superfici della stanza.

Quando si tratta di partizioni trasparenti (finestre, ecc.), `OpenBPS` può operare i due modi diversi per determinare l'apporto radiativo e l'interazione termica: calcolare in linea in fase di esecuzione i coefficienti di trasmissione assorbimento della radiazione solare e luminosa, in funzione dell'angolo di incidenza della stessa, e un valore di trasmittanza termica in funzione della temperatura che si instaura nel sistema, proprietà calcolate inserendo una descrizione strato per strato della sistema, oppure può eseguire un calcolo semplificato inserendo una descrizione di sistema semitrasparente presa da progetti di terze parti, come `WINDOW 7.4` di Berkeley Lab.

Il controllo solare adattativo dei sistemi di schermatura non è stato ancora introdotto, ma l'introduzione di nuovi modelli per il calcolo delle ombre portate e del controllo degli ombreggiamenti è facilitata dalla struttura orientata agli oggetti del progetto. Allo stesso modo, è possibile includere un precalcolo dei fattori di forma per diverse posizioni dell'ombreggiatura e gestire la parte di radiazione ad onde corte (SW) con una descrizione simile a quella adottata per uno strato trasparente, mentre lo scambio termico convettivo con l'aria estrema può essere stimato sia con coefficienti costanti che in funzione delle condizioni ambientali. Per la radiazione solare diffusa su superfici inclinate, è stato implementato algoritmo di cielo anisotropo come descritto da [34]. Questo modello di cielo non isotropo tiene conto della radiazione circumsolare, della luminosità dell'orizzonte e radiazione diffusa isotropa attraverso i "brightening coefficients" derivati empiricamente. Questi coefficienti, funzione della posizione del sole e della copertura nuvolosa, sono stati

implementati dopo l'ultimo aggiornamento della correlazione eseguito da Perez nel 1999, come riportato sia nella documentazione tecnica di TRNSYS 17 che di EnergyPlus 8.6.0.

3.2.4 Simulazione di sistemi impiantistici

Il simulatore dà la priorità ai componenti del fabbricato dell'edificio prima di iniziare a gestire la simulazione dei sistemi impiantistici dell'edificio (HVAC e elettrici). Gli oggetti che rappresentano i componenti HVAC e sistemi elettrici, apparecchiature per la conversione di energia e qualsiasi altro componente necessario dei sistemi tecnici dell'edificio, sono quindi invitati a esporre i loro risultati al momento attuale in sequenze organizzate. La naturale modularità dello schema orientato agli oggetti adottato aiuta a realizzare una simulazione completamente integrata di carichi, sistemi e impianti, ma allo stesso tempo può sollevare problemi di stabilità. Per evitare o limitare il più possibile tali instabilità, gli oggetti appartenenti a un sistema tecnico specifico verranno organizzati letteralmente in "code", ovvero risolti in sequenza in base alla direzione flusso di energia. Con "coda" intendiamo una sequenza di oggetti che vengono risolti in sequenza uno dopo l'altro perché interagiscono principalmente in sequenza, come, ad esempio, un emettitore, i tubi di distribuzione, la caldaia e i loro controlli termostatici. Naturalmente, le reti di tubi e condotte rappresentano "code" a circuito chiuso. Pertanto, ad ogni intervallo temporale globale, il Simulatore deve iterare tra tutti i componenti, riuscendo a risolvere dapprima le incognite per i componenti del fabbricato dell'edificio (principalmente in parallelo) e successivamente per i sistemi tecnici dell'edificio (principalmente in serie). Questa simulazione integrata consente di modellare in modo più realistico i limiti indotti da una potenza installata finita e le strategie di controllo e fornisce un accoppiamento più stretto tra i componenti del fabbricato dell'edificio e i componenti del sistema tecnico. In tal modo si può gestire più facilmente l'interazione, consentendo che uno specifico requisito dell'oggetto, come nel caso delle facciate attive a doppia pelle, in cui un sistema HVAC interagisce strettamente con un componente dell'involucro, possa indurre delle modifiche nella struttura di gestione generale indicata.

La parte che riguarda i sistemi impiantistici dell'edificio sarà oggetto di sviluppo nella seconda fase della ricerca.

3.2.5 Dati di input, output e meteo

Come accennato più volte, OpenBPS è un insieme di librerie a collegamento dinamico progettate principalmente per essere utilizzate da software di terze parti. Pertanto, il front-end dell'utente non rientra nell'obiettivo del progetto, sebbene sia stata sviluppata una semplice GUI per scopi di sviluppo e test. Questa interfaccia consente di importare file di input di EnergyPlus, con alcune limitazioni, o file creati con il plug-in TRNSYS17 o OpenStudio in SketchUp, sempre con alcune limitazioni. Per gestire i test sui componenti, assegnando, ad esempio, diversi risolutori e così via, sono state inoltre implementate per la selezione le funzionalità di trascinamento. Per lo stesso motivo il gestore dell'output consente di selezionare per la stampa quasi tutte le variabili necessarie per analizzare le prestazioni dell'edificio.

Il vero input per il codice di calcolo è comunque lo stesso Building Object: la GUI ha solo la funzione di assegnare i valori numerici a tutte le proprietà richieste da tutti gli oggetti inclusi nel progetto, inclusi i requisiti di simulazione; tale oggetto viene poi passato alla libreria dll che contiene il `SimulationManager` per l'esecuzione della simulazione. Durante la simulazione, l'oggetto di output viene "caricato" con i valori relativi le variabili di output richieste, gli avvisi e qualsiasi altra informazione. Al termine della simulazione, l'oggetto di output viene esposto al programma chiamante, che si assume il compito di produrre grafici, sinossi e qualsiasi altro output specificamente formattato.

L'altro input di dati principali è quello relativo ai dati climatici che viene fornito tramite il file di dati meteorologici letto direttamente dal motore di simulazione. Il codice può leggere direttamente i formati di dati meteorologici standard, come TMY ed EPW, o formati di dati personalizzati. In ogni caso, il processore meteorologico è in grado di produrre le quantità richieste indipendentemente dalla corrispondenza tra la frequenza di dati fornita e la frequenza di dati richiesta. Dopo un allineamento temporale, il processore

meteorologico produce tramite interpolazione i dati richiesti se sono richiesti da un oggetto con una frequenza superiore alla frequenza di registrazione. Se, invece, questa frequenza è inferiore alla frequenza di registrazione, abbiamo due possibilità. Siamo in grado di comunicare a tutti i componenti della simulazione tutti i dati climatici e far loro decidere se vogliono eseguire calcoli multipli senza iterazione o se preferiscono archiviare e manipolare tali dati secondo il loro schema numerico ed eseguire il calcolo una sola volta. Oppure possiamo manipolare i dati climatici prima di esporli ai componenti, nel qual caso sorgono due diverse possibilità: la frequenza è un multiplo della frequenza di registrazione o no. Nel primo caso, i dati non integrali vengono forniti direttamente all'oggetto, mentre i dati integrali vengono cumulati prima di esporli all'oggetto. Nel secondo caso viene eseguita un'interpolazione aggiuntiva per fornire le informazioni richieste.

3.2.6 Contribuire ai nuovi sviluppi

Uno degli obiettivi principali di OpenBPS è incoraggiare uno sviluppo continuo del codice e il suo arricchimento con nuove funzionalità. Per raggiungere tale obiettivo, è stato deciso di adottare un approccio open source e di pianificare lo sviluppo di una comunità di sviluppatori, che sia in grado di assolvere tali compiti in modo autonomo. Tale procedura contributiva, che sarà sviluppata nella terza fase della ricerca, può essere sintetizzata come segue: chiunque può scaricare il codice sorgente e può utilizzarlo o modificarlo, in base al contratto di licenza open source, come crede; per essere invece un collaboratore riconosciuto (ovvero per poter caricare nel repository ufficiale il proprio codice, che entrerà a far parte della successiva versione ufficiale rilasciata), uno sviluppatore deve unirsi alla comunità e seguire le sue regole. La produzione di nuove classi di componenti per il sistema edificio, sia relativi al fabbricato che agli impianti, è prioritaria ed è in linea di principio abbastanza semplice. A causa della struttura orientata agli oggetti e dell'ambiente di sviluppo (MS Visual Studio su Windows, MonoDevelop su Linux e macOS), il processo è relativamente semplice. Innanzitutto, uno sviluppatore definisce un nuovo componente scrivendo il suo modello matematico e numerico, identificando i parametri del modello e le equazioni necessarie, i coefficienti specializzati e tutti gli altri dati necessari. Successivamente, lo sviluppatore scrive il codice (utilizzando lo standard di programmazione OpenBPS), identificando la classe "parent" da cui ereditare proprietà e metodi comuni. Se non è già disponibile una classe da cui ereditare quanto serve, la classe genitore deve essere la classe "SimulationComponent". In effetti, il simulatore all'inizio della simulazione analizza tutti i componenti per trovare quello che eredita da quella classe, grazie al polimorfismo. La classe del componente di simulazione espone metodi che sono sempre chiamati dal simulatore, come il metodo di inizializzazione, il metodo di calcolo, i metodi necessari per gestire la convergenza e il metodo per la scrittura dell'output. Tutti i nuovi componenti, mentre implementano le loro proprietà e metodi specifici, devono ereditare dalla loro particolare classe "genitore" o sovrascrivere l'implementazione disponibile di quei particolari metodi definiti per la classe "SimulationComponent", per essere correttamente chiamati dal simulatore, per ottenere input da altri oggetti e per fornire il loro output.

4 OpenBPS: caratteristiche specifiche e modelli numerici

Il progetto OpenBPS (Open Building Simulation tool) è stato implementato in `c#` seguendo la logica della programmazione orientata agli oggetti. Pertanto, ogni unità concettuale è stata tradotta in una classe a se stante che conterrà, una volta istanziata, i valori numerici delle sue proprietà e sarà responsabile dei calcoli che le competono all'interno della simulazione dinamica. Vi saranno componenti legati ad oggetti fisici e componenti che avranno il compito di gestire l'interazione fra di essi

I file di input per il programma OBPS hanno lo scopo di descrivere un edificio alla volta x consentire di simularne le prestazioni. L'edificio è composto da partizioni, che separano l'edificio dall'esterno e lo suddividono in diverse zone.

Ogni partizione è una singola entità, con i suoi campi, proprietà e metodi. Ogni partizione è responsabile del suo calcolo, ovvero della valutazione del trasferimento di calore e massa attraverso i suoi strati. Altri componenti sono responsabili del settaggio delle condizioni al contorno (BC = Boundary Conditions) di ciascuna partizione, come sarà descritto in seguito.

Ogni zona è una singola entità, con i suoi campi, proprietà e metodi. Ogni zona ha un componente che ha la responsabilità di eseguire i calcoli relativi allo scambio di calore tramite radiazioni a onda lunga (LW) fra le partizioni appartenenti alla zona, i.e. il suo modulo "LWRadiationModule".

Questo `LWRadiationModule` ha la responsabilità di impostare le BCs relative a questo tipo di scambio di calore (temperatura media radiativa e flussi imposti ad onde lunghe, come quelli derivanti da carichi interni dovuti a persone ed apparecchi elettrici) su ciascuna superficie di ciascuna partizione della zona che la possiede.

L'ambiente esterno è rappresentato da un altro componente, chiamato `ExternalZone`, unico all'interno di una simulazione. Il componente `ExternalZone` è responsabile dell'impostazione dei dati meteorologici di ciascun timestamp e della gestione di tutto ciò che riguarda l'ambiente esterno.

Il componente `SWRadModule` è responsabile del calcolo relativo alla radiazione a onde corte (SW), proveniente principalmente dal sole e riflessa dal suolo e da altre entità. Anch'esso è univoco all'interno di una simulazione.

Ogni proprietà o risultato del calcolo è associata ad un timestamp. Il calcolo inizia con un timestamp definito dall'utente. In questo momento, tutte le proprietà e i parametri del sistema dovrebbero essere "conosciuti". Al timestamp iniziale, tutte le proprietà (comprese quelle che sono una soluzione del calcolo) devono avere il loro valore di "inizializzazione". La simulazione calcola il valore di tutti i parametri del sistema (che sono variabili ed oggetto di indagine) dopo un intervallo di tempo o timestep, per il timestamp successivo.

Riassumendo, il passo temporale o time step è un intervallo di tempo, mentre il timestamp è un preciso istante nel tempo. Tutti i risultati principali del calcolo sono valori istantanei riferiti ad un timestamp. Alcuni risultati aggiuntivi possono essere restituiti in seguito ad un post-processing che ne abbia valutato l'integrale medio nel time step precedente al timestamp a cui vengono riferiti.

4.1 Discretizzazione temporale

4.1.1 Convenzione temporale

Come detto nell'Introduzione, il calcolo inizia con un timestamp definito dall'utente. In questo momento, tutte le proprietà e i parametri del sistema devono essere "noti". Al timestamp iniziale, tutte le proprietà (comprese quelle che sono una soluzione del calcolo) devono avere il loro valore di "inizializzazione". La simulazione calcola il valore di tutte le grandezze caratteristiche del sistema (che sono variabili e che si vuole determinare) dopo un passo temporale (time step), cioè per il timestamp successivo. Pertanto, il passo temporale è un intervallo di tempo, mentre il timestamp è un istante esatto nel tempo. Tutti i risultati del calcolo sono valori istantanei riferiti a un timestamp.

Durante la simulazione è possibile utilizzare diversi tipi di tempo, ad esempio:

1. ora locale standard, cioè in funzione del fuso orario in cui si trova il sito considerato:
 - 1.1. con correzione legale, di seguito identificato con `LegalTime`
 - 1.2. senza correzione legale, di seguito identificato con `LocalTime`
2. tempo solare, cioè in funzione della posizione del sole nel cielo, sempre rispetto al sito considerato, di seguito identificato come `SolarTime`.

Poiché normalmente i profili d'utenza, cioè l'orario di lavoro, le routine quotidiane, ecc., sono espressi nell'ora standard locale, è stato scelto questo orario per il timestamp di inizio e l'avanzamento complessivo della simulazione. Pertanto, l'input e l'output verranno forniti nell'ora standard locale.

In particolare, l'output verrà fornito in `LocalTime`, ovvero senza correzione legale. A causa di ciò, durante i periodi in cui si ha correzione legale dell'orario, la simulazione produce uno spostamento temporale dei profili delle quantità considerate, rispetto a quello che potrebbe risultare tramite un monitoraggio di un sistema reale, in cui l'orologio di rilevamento del tempo si aggiorni automaticamente all'ora legale.

Di conseguenza, anche il passo temporale di simulazione è basato su `LocalTime`.

4.1.2 Inizializzazione

Come detto nel paragrafo precedente, al timestamp iniziale, tutte le proprietà (comprese quelle che sono una soluzione del calcolo) devono avere il loro valore di "inizializzazione". Tuttavia, questi valori potrebbero non essere noti all'utente del programma, pertanto questo offre diverse possibilità. L'utente può scegliere tra:

1. inizializzazione predefinita;
2. inizializzazione specifica definita dall'utente;
3. inizializzazione generale definita dall'utente.

L'inizializzazione predefinita prende la temperatura di bulbo secco dell'aria esterna esterno e inizializza:

- tutti i nodi d'aria delle zone a quella temperatura
- tutte le partizioni non contro-terra a quella temperatura
- tutte le partizioni contro-terra, con una distribuzione della temperatura che è la soluzione allo stato stazionario di una partizione con una superficie alla temperatura del suolo e l'altra alla temperatura dell'aria della zona (gli effetti tridimensionali non sono presi in considerazione).

Attualmente il codice non ripete le stesse BC per un certo numero di passaggi temporali o lo stesso giorno per un numero di giorni fino a quando i risultati della simulazione raggiungono una condizione "plausibile", prima di iniziare la simulazione per il periodo. Per ottenere una soluzione il più possibile non influenzata dall'inizializzazione, si dovrebbe eseguire una simulazione dinamica con BC variabili per un numero di giorni (a seconda del comportamento dell'edificio) prima del periodo di interesse.

4.1.3 Passi temporali consentiti

L'utente può scegliere il temporale utilizzato per la simulazione con alcune restrizioni relative all'intervallo temporale con cui sono disponibili i dati meteorologici.

Il passo temporale di simulazione può essere:

- uguale all'intervallo temporale con cui sono disponibili i dati meteorologici (solitamente l'ora);
- un multiplo intero di tale intervallo;
- un sottomultiplo di tale intervallo.

4.2 Dati climatici

I dati climatici per la simulazione dinamica sono usualmente dati medi prodotti su base statistica. Questa scelta è stata fatta per consentire di confrontare, in fase di progettazione, le prestazioni di diverse tecnologie, soggette a forzanti esterne che abbiano una certa validità statistica nella loro occorrenza.

Tuttavia, considerando i requisiti aggiuntivi, che un codice con funzionalità estese deve avere:

- la possibilità di essere validato empiricamente rispetto a prestazioni misurate e quindi dati climatici specifici ottenuti da registrazioni in situ;
- la possibilità di utilizzare i BPSt in fase di gestione del sistema edificio, sia per consentire una migliore conoscenza dello stesso, sia per utilizzarlo nell'ottimizzazione del controllo e/o nella rilevazione e diagnosi dei guasti;

occorre riconsiderare il modo in cui i dati climatici vengono gestiti nei BPSt.

4.2.1 Formati disponibili

I dati climatici sono solitamente disponibili per la maggior parte dei siti di grande interesse nazionale e internazionale, quasi sempre con un intervallo temporale dell'ora tra dato e dato. I dati reperibili sono però di natura e formato diversi tra loro. La natura è relativa al come sono stati prodotti i dati climatici riportati nei file, il formato riguarda quale tipo di dato è riportato, la sua unità di misura, il modo in cui è scritto dentro il file. Nel seguito verranno analizzati solo i formati più ricorrenti, senza entrare nel merito della metodica seguita per generare i valori delle grandezze riportate.

Il codice di calcolo implementa una procedura che è in grado di leggere i diversi tipi di formato di file climatici. Oltre all'indicazione del tipo di formato del file di dati meteorologici, (spesso identificato dal suffisso del file), occorre specificare la variabile booleana della classe `WeatherFileData`: "RealDataNotStatisticallyDerived" (vedasi Figura 12). Se questa variabile è impostata su "True", i dati climatici possono essere assegnati anche per un periodo minore dell'anno, iniziare in qualsiasi mese e terminare ogni volta che è necessario; è consentito inoltre l'impiego di anni bisestili (anni con 367 giorni) se del caso. Il calendario della simulazione (giorno di inizio, giorno di fine) deve essere impostato all'interno dell'intervallo dei dati forniti. Quando quest'opzione è impostata su true, per calcolare la posizione del sole vengono utilizzati gli algoritmi dell'Almanacco astronomico.

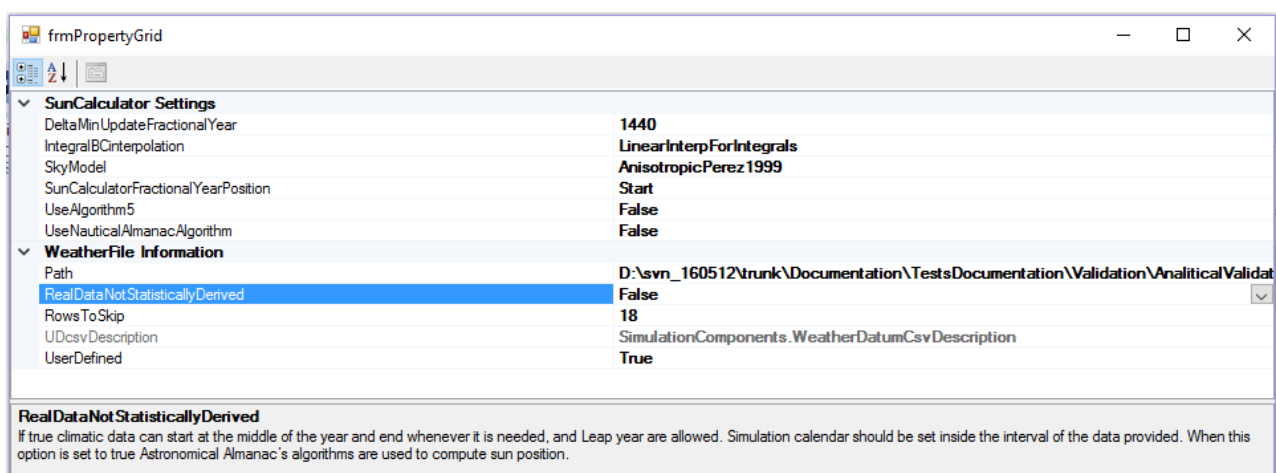


Figura 12 – Maschera per l'introduzione dei parametri relativi ai dati climatici.

4.2.1.1 Formato TMY

Viene di seguito analizzata la struttura del file di dati climatici di tipo TMY, TMY2 e TMY3.

Come descritto in ciascuno dei manuali per i formati di dati TMYx, l'ultimo dei quali è il manuale TMY2 e TMY3 [35] i dati climatici relativi alla radiazione solare sono valori integrali rispetto all'intervallo temporale definito dal timestamp precedente (estremo inferiore) e al timestamp corrente (estremo superiore), mentre gli altri valori (temperatura, umidità, velocità del vento, ecc.) sono valori istantanei al timestamp corrente.

Year	Month	Day	End of Interval: Solar Time	End of Interval: Local Time	Mid of Interval in Solar Time	Global Horizontal Radiation in previous time interval [Wh/m ²]	Direct normal Radiation in previous time interval [Wh/m ²]	Diffuse Horizontal Radiation in previous time interval [Wh/m ²]	Dry Bulb Temperature at the end of the time interval [°C]	Dew Point Temperature at the end of the time interval [°C]	Relative humidity at the end of the time interval [%]
95	1	1	7:57	8:00	7:27	6.94 (I ₇₋₈)	2.22 (I _{bn,7-8})	6.84 (I _{d,7-8})	2.2	-6.1	58.5
95	1	1	8:57	9:00	8:27	93.33 (I ₈₋₉)	67.5 (I _{bn,8-9})	81.58 (I _{d,8-9})	3.3	-5	58.5
95	1	1	9:57	10:00	9:27	116.67 (I ₉₋₁₀)	16.11 (I _{bn,9-10})	111.74 (I _{d,9-10})	7.2	-8.9	19.5

Figura 13 – Esempio di struttura di un file TMY

In un file TMY3 (terza generazione) ogni riga rappresenta una specifica ora di uno specifico giorno di uno specifico mese in un anno di riferimento e i valori dei dati riportati sono separati da virgole. In ogni riga sono poi riportati, con riferimento all'esempio, puramente esplicativo, riportato in Figura 13, i valori al timestamp (ad esempio alle ore 10 di localtime) di irradiazione solare totale oraria, irradiazione solare diretta normale oraria, irradiazione solare diffusa oraria sul piano orizzontale che corrispondono agli integrali tra le 9 e le 10, mentre la temperatura a bulbo secco, temperatura a bulbo umido e umidità relativa corrispondono ai valori istantanei alle 10. Sempre a titolo esplicativo, in Figura 13 sono riportati sia il tempo locale standard che il tempo solare allo stesso timestamp, per evidenziare la differenza che esiste tra questi.

4.2.1.2 Formato EPW

Una descrizione del formato dei dati EPW è disponibile nel manuale dei programmi ausiliari di EnergyPlus8.4.0, a partire dal capitolo "2.9 Dizionario dei dati del file meteorologico (EPW) EnergyPlus".

La "natura" (istantanea o integrale) dei valori riferiti ai dati climatici riportati sono uguali a quelle dei formati TMY.

4.2.1.3 Formato WMO

D'altra parte, l'Organizzazione Mondiale di Meteorologia (WMO) prescrive la restituzione di tutte le misurazioni, anche quelle relative alla radiazione solare, come valori "quasi istantanei", anziché integrati. Con "quasi-istantaneo" si intende che lo strumento di misura raccoglie i dati con una frequenza di campionamento che è più piccola della frequenza di registrazione dei dati. Il valore che viene registrato è il valore medio tra gli ultimi dati campionati prossimi al tempo di registrazione (il numero di tali dati su cui si fa la media dipende dalla frequenza di campionamento e dalla frequenza di registrazione (ad es. una media dei dati campionati degli ultimi 20 secondi, per le registrazioni orarie).

È però alquanto difficile disporre di dati con frequenze di registrazione sub-orarie, e, molto spesso, gli stessi dati WMO, se riportano registrazioni orarie, possono contenere il valore integrale della radiazione invece del valore istantaneo, sicuramente invece presente in quelli con frequenze di registrazione più elevate.

4.2.1.4 Formato definito dall'utente

Per importare i dati meteorologici, si può utilizzare il formato "valore separato da virgola", cioè su una riga di dati la virgola a destra delimita il campo del dato generando così un numero di colonne non necessariamente allineate pari al numero di dati per ogni riga. A numero d'ordine della colonna si associa quindi la natura del dato fornito (temperatura, radiazione solare, ecc.) e un eventuale coefficiente di conversione per riportare il dato dall'unità di misura originale all'unità di misura SI utilizzata dal codice per i calcoli.

La radiazione solare può essere fornita sia come valore istantaneo al timestamp, Irrad (W/m^2) o come valore integrale rispetto al passo temporale precedente, Rad (Wh/m^2). Occorre ovviamente specificare cosa si è scelto.

Allo stesso modo si può richiedere che i valori di temperatura a bulbo secco e a bulbo umido vengano interpolati o no. Se il relativo campo "NotInterpolated" è impostato su "False", i valori riportati in ciascun timestamp del file dati verranno interpolati linearmente, se e quando necessario, per ottenere il valore necessario ad ogni timestamp di simulazione. Ad esempio, intervallo di simulazione 15 minuti e dati climatici orari allineati: alle 10:00 viene fornito in dato non interpolato presente al timestamp 10:00 del file climatico; alle 10:15 si interpola linearmente tra il dato presente al timestamp 10:00 e quello al timestamp 10:30 del file climatico, e così via. Questo meccanismo di interpolazione è adottato di default per i file climatici TMY e EPW che hanno sempre registrazioni orarie.

Se il campo "NotInterpolated" è impostato su "True", il valore della temperatura fornito ad ogni timestamp di simulazione, è il valore corrispondente al dato letto al timestamp dei dati meteorologici subito precedente o uguale, e questo verrà mantenuto costante fintantoché il timestamp di simulazione non eccede o è uguale ai successivi timestamp dei dati climatici. Ciò significa che i valori forniti non saranno interpolati per ottenere il valore ad ogni timestamp di simulazione, portando a variazioni discontinue a gradini delle condizioni contorno relative alle temperature. Ad esempio, intervallo di simulazione 15 minuti e dati climatici orari allineati: alle 10:00 viene fornito in dato non interpolato presente al timestamp 10:00 del file climatico; alle 10:15 viene usato il dato delle 10:00, alle 10:30 ancora lo stesso così come alle 10:45, infine alle 11:00 (tempo di simulazione) si assume il valore riportato nel file climatico al suo timestamp pari a 11:00, e così via.

Quest'opzione è stata introdotta solo per consentire di valutarne l'influenza sui risultati delle simulazioni, ma il suo utilizzo non è fortemente sconsigliato.

In Figura 14 è riportata la maschera di introduzione dei parametri che specificano il formato utente per i dati climatici.

4.2.2 Gestione dei dati integrali

I metodi numerici utilizzati per il calcolo dello scambio termico conduttivo possono richiedere, come dati di input per la definizione delle condizioni al contorno, valori istantanei a ciascun timestamp, sia per le temperature che per i flussi termici di incidenti, tra cui quelli radiativi ad onde corte, cioè la radiazione solare. Di conseguenza, occorre fornire al solutore un valore di potenza istantanea in un determinato timestamp, a partire da un valore integrale disponibile per un intervallo temporale che include tale timestamp. È pertanto necessaria una manipolazione del valore integrale della radiazione solare che possa fornire in modo sufficientemente accurato una stima di tale potenza.

Poiché la legge di variazione dell'irradiazione all'interno di ogni diverso intervallo di integrazione non è nota, per estrarre dai valori integrali dei valori istantanei occorre assumere delle ipotesi di distribuzione.

Tra le ipotesi di lavoro più comunemente adottate, vi sono quelle di considerare:

- i valori integrali assunti pari a valori istantanei medi situati al centro dell'intervallo di tempo su cui sono stati integrati,
- nota a priori la distribuzione temporale di cui il valore è l'integrale.

In Figura 15 è riportato un esempio relativo al caso a): il dato integrale orario riportato nel file climatico alle ore 8:00 viene attribuito quale valore medio istantaneo alle ore 7:30 mentre il dato delle ore 9:00 viene attribuito alle 8:30. In questo caso il valore "istantaneo" esposto ai vari solutori alle ore 8:00 del tempo di simulazione è il valore che si ottiene interpolando linearmente tra il dato medio "istantaneo" attribuito alle 7:30 e il dato attribuito alle 8:30.

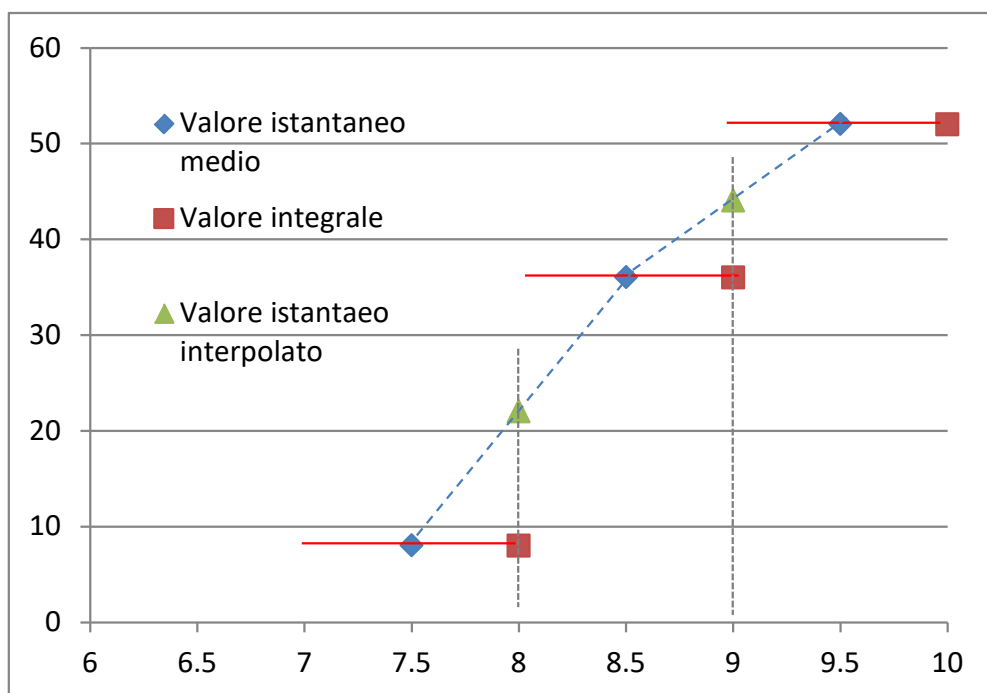


Figura 15 – Valori integrali considerati come valori istantanei medi posti a metà del relativo intervallo di integrazione: interpolazione lineare tra i valori medi "istantanei"

Nel caso b) si ipotizza la conoscenza "a priori" della distribuzione temporale della grandezza considerata, che consente di determinare, a partire dal suo valore integrale sul timestep dei dati, un suo valore sempre integrale, ma su un intervallo di tempo ridotto che corrisponde al timestep di simulazione; poi si assume il suo valore medio sul timestep quale valore istantaneo da attribuire al timestamp corrente della simulazione (questo è il caso del processore solare di TRNSYS, trattato in dettaglio al paragrafo 4.3.7).

4.2.3 Sincronizzazione dei dati climatici con il tempo della simulazione

I dati climatici, sia che siano valori istantanei che integrati nel tempo, sono disponibili a precisi valori discreti nel tempo, i timestamp dei dati climatici, usualmente riferiti al tempo locale standard. Gli intervalli temporali di registrazione dei dati climatici normalmente disponibili sono di un'ora. In tal caso, se il time step di simulazione (intervallo temporale della discretizzazione del calcolo) è pari all'ora, vi è una naturale sincronizzazione tra il dato disponibile e il dato richiesto: i time step sono coincidenti ed è sufficiente iniziare la simulazione con un timestamp uguale a quello del primo dato climatico disponibile.

Se invece, sempre nell'ipotesi di dati climatici disponibili su base oraria, si utilizza un time step di simulazione più piccolo dell'ora, ad esempio da 15 minuti, si ha il problema di "generare" i dati corrispondenti a i timestamp di simulazione compresi nell'intervallo tra due timestamp dei dati climatici. Se invece si utilizza un time step maggiore, ad esempio 2 ore, si ha il problema di gestire la perdita d'informazione legata al valore del dato climatico intermedio, comunque disponibile (ad esempio, si potrebbe avere un brusco abbassamento nel valore del dato nell'ora intermedia che rompe l'apparente monotonicità dell'andamento temporale della grandezza considerata).

Generalizzando, cioè considerando un time step dei dati climatici non necessariamente orario, si possono quindi avere due diverse situazioni:

1. il passo temporale della simulazione è inferiore al passo temporale compreso tra due dati climatici;
2. il passo temporale della simulazione è maggiore del passo temporale compreso tra due dati climatici.

Nel primo caso, quando il passo temporale della simulazione è inferiore al passo temporale dei dati climatici, il dato climatico esposto ai solutori in corrispondenza di un determinato timestamp di simulazione è il risultato dell'interpolazione tra due dati climatici appropriati e adiacenti. Tale interpolazione, usualmente di tipo lineare, può essere in casi particolari anche di tipo non lineare.

Nel secondo caso, quando il passo temporale della simulazione è maggiore del passo temporale dei dati climatici, si può avere una perdita di informazione che può alterare il risultato della simulazione. Vi sono due possibilità:

1. il passo temporale della simulazione è un multiplo del passo temporale dei dati climatici;
2. il passo temporale della simulazione NON è un multiplo del passo temporale dei dati climatici.

Nel primo caso, non ci sono problemi di sincronizzazione, e il processore dei dati climatici espone ai solutori:

- per le grandezze non integrali, i valori dei dati climatici ai timestamp corrispondenti;
- per le grandezze integrali, ci sono due opzioni:
 - a) un valore "istantaneo" basato su un'interpolazione sui dati climatici corrispondenti al timestamp della simulazione corrente e quello antecedente (in tal caso si perde l'informazione relativa al dato intermedio);
 - b) un valore "istantaneo" basato su un'interpolazione sui dati climatici corrispondenti al timestamp della simulazione corrente e al timestamp del dato climatico subito antecedente a quello della simulazione (in tal caso si sfrutta meglio l'informazione disponibile).

Come si può notare dai grafici della Figura 16 e Figura 17, nel caso b) si ha una migliore stima del valore "istantaneo" della grandezza integrale al timestamp di simulazione desiderato.

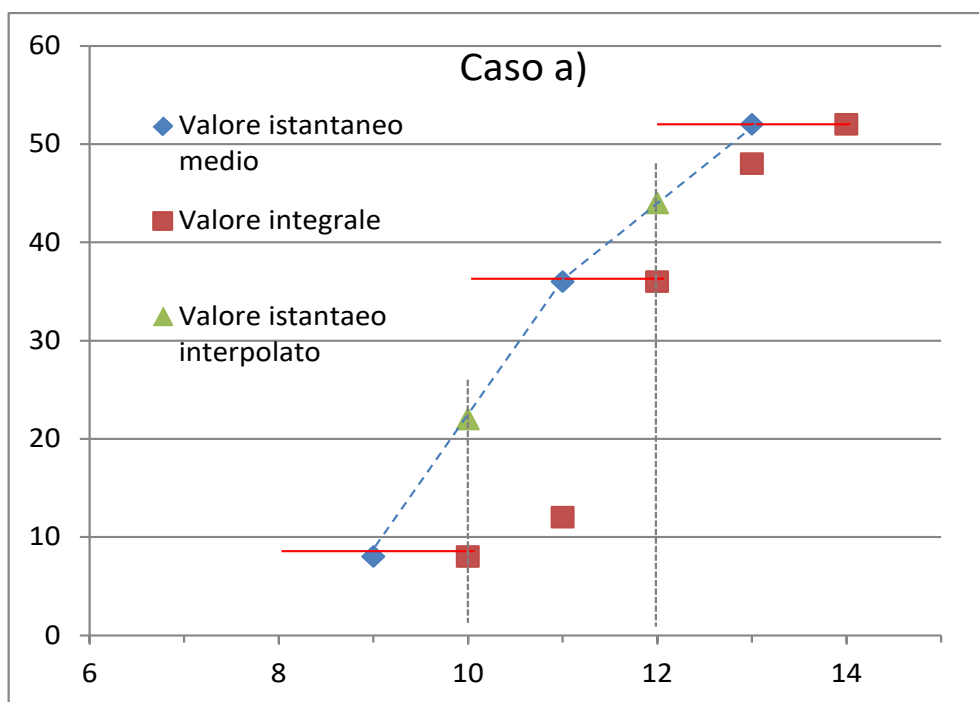


Figura 16 – Grandezze integrali: time step simulazione maggiore del time step dati, caso a)

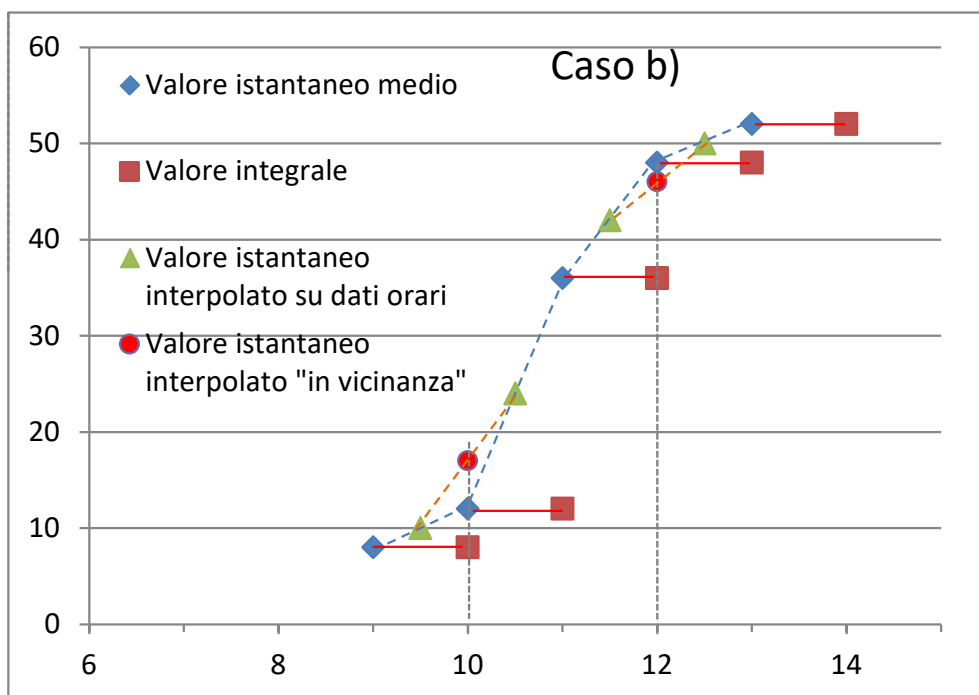


Figura 17 – Grandezze integrali: time step simulazione maggiore del time step dati, caso b)

Nel caso in cui il passo temporale della simulazione NON è un multiplo del passo temporale dei dati climatici, si ha:

- per le grandezze non integrali, l'interpolazione lineare tra i dati climatici subito antecedenti e subito successivi al timestamp di simulazione;
- per le grandezze integrali, alla luce di quanto evidenziato nel caso precedente,;

- si generano i valori medi istantanei associati ad ogni timestamp dei dati climatici (questi sono posizionati indietro nel tempo di metà del time step degli stessi);
- si interpola rispetto al timestamp della simulazione utilizzando i valori medi istantanei generati che siano subito prima e subito dopo tale timestamp.

4.2.3.1 Confronto con EnergyPlus e TRNSYS

Il modo con cui operano i codici dei BPSt non è sempre chiaro e lineare. Si è quindi deciso di analizzare come affrontano tale problematica due dei più noti e utilizzati BPSt, EnergyPlus e TRNSYS, con riferimento alle loro rispettive versioni 8.4.0 e 17. Per condurre tale analisi si è utilizzato il file climatico, DRYCOLD.tmy, contenuto nello standard ASHRAE 140-2011 [13], standard utilizzato nel processo di validazione dell'intero codice OpenBPS.

Nella figura successiva, Figura 18, possiamo vedere rappresentato l'andamento temporale della temperatura di bulbo secco a diversi timestamp, come estratta direttamente dal file DRYCOLD.tmy e fornita dall'output di TRNSYS17 e EnergyPlus8.4.0, per passi temporali di simulazione coincidenti con il passo temporale dei dati climatici e nel caso in cui il primo sia inferiore al secondo (15 minuti contro un ora).

Come si può notare dalla figura, mentre EnergyPlus espone ai solutori esattamente i dati climatici corrispondenti al timestamp dei dati climatici, quando si ha lo stesso time step di 1 ora, e interpola linearmente tra i dati climatici disponibili quando il time step di simulazione è di 15 minuti, TRNSYS ha un comportamento anomalo.

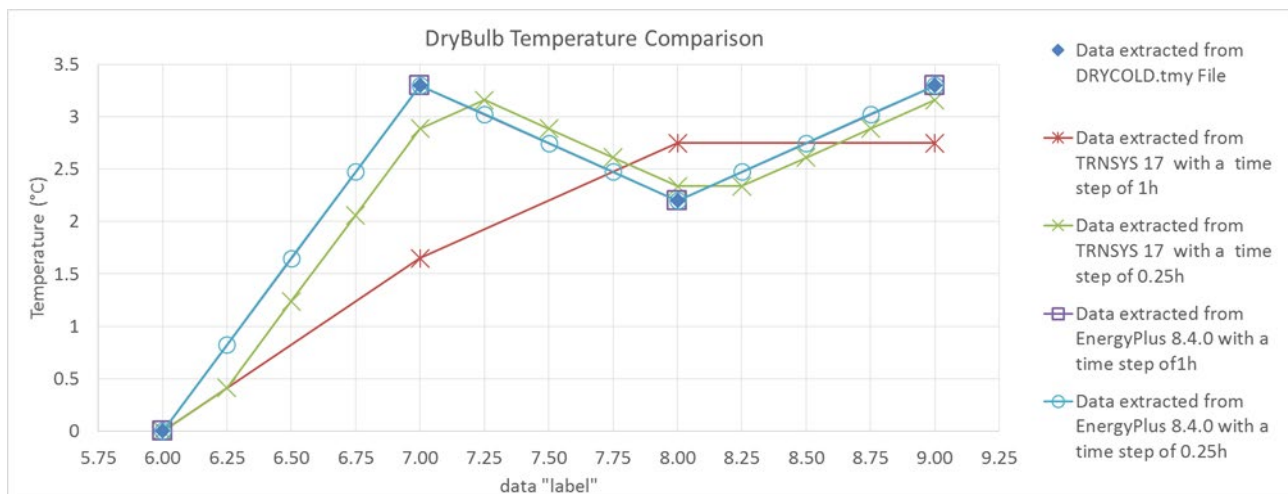


Figura 18 – Valori della temperatura a bulbo secco in funzione del tempo di simulazione sincronizzato con il tempo dei dati climatici con diversi intervalli di simulazione.

TRNSYS opera nel seguente modo:

1. parte dal presupposto che, per le grandezze istantanee, che il dato climatico che vada esposto ai solutori non debba essere quello al dato timestamp della simulazione, ma un valore definito a metà dell'intervallo del tempo di simulazione antecedente (ad esempio, se il passo di simulazione è orario, al tempo di calcolo ore 8:00 la temperatura dell'aria esterna, da passare al modulo che calcola la trasmissione attraverso una parete per determinare il flusso termico scambiato, non è il valore che si ha alle 8:00 ma quello che si avrebbe alle 7:30);
2. determina i valori V delle grandezze istantanee da attribuire ai timestamp $t_{w,i}$ dei dati climatici per interpolazione lineare tra il dato presente nel file al timestamp successivo, $D(t_{w,i+1})$, e il dato presente al timestamp attuale, $D(t_{w,i})$, cioè:

$$V(t_{w,i}) = \frac{D(t_{w,i+1}) + D(t_{w,i})}{2} \quad (4.1)$$

3. se il timestamp di simulazione, $t_{s,i}$, coincide con il timestamp dei dati, $t_{w,i}$, il dato esposto è direttamente $V(t_{w,i})$; se invece non lo è costruisce una sequenza di valori da esporre per i timestamp di simulazione interni al passo temporale dei dati climatici, $V(t_{s,i})$, con $t_{s,i} \geq t_{w,j}$ e $t_{s,i} \leq t_{w,j+1}$, prima ottenendo una distribuzione di valori dei dati all'interno di tale intervallo per interpolazione lineare assumendo dei $t_{w,i} = t_{s,i}$:

$$D(t_{w,i}) = D(t_{w,j}) + \frac{D(t_{w,j+1}) - D(t_{w,j})}{t_{w,j+1} - t_{w,j}} (t_{w,i} - t_{w,j}) \quad (4.2)$$

e poi applicando la regola di cui al punto 2., cioè

$$V(t_{s,i}, t_{w,i}) = \frac{D(t_{w,i+1}) + D(t_{w,i})}{2} \quad (4.3)$$

La conseguenza di tale modo di procedere è che gli output forniti da TRNSYS Simulation Studio, riportati in Figura 18, che sono l'input per i calcoli eseguiti all'interno del TRNSYS Building Module (Type 56) senza alcuna altra modifica, risultano diversi dalla sequenza dei dati climatici disponibili, sia quando il time step di simulazione coincide con quello dei dati climatici, sia quando risultano diversi e interviene un'ulteriore interpolazione.

Questo modo di procedere non è l'ideale per gli sviluppatori di modelli dedicati di componenti del sistema edilizio, perché risulta sensato ed efficiente solo se lo schema numerico di uno specifico solutore, ad esempio l'integrazione numerica della diffusione termica, necessita della condizione al contorno espressa a metà dell'intervallo di integrazione e il timestamp di simulazione è relativo al valore da determinare con l'integrazione numerica. Tale assunto, che può andar bene per alcuni schemi numerici alle differenze finite, è assolutamente sbagliato ad esempio per l'impiego delle funzioni di trasferimento per la conduzione termica per tutti i modelli sviluppato sotto l'ipotesi dello Stato Quasi Stazionario.

4.2.3.2 Dettagli su EnergyPlus

In EnergyPlus, l'utente può scegliere la frequenza di reporting, con la quale EnergyPlus fornisce gli output prescelti, tra:

- Detailed: produce output ad ogni istanza (ad es. timestep variabili nel sistema HVAC)
- Timestep: si riferisce timestep della zona termica e produce output con frequenza pari al numero di timestep nell'ora (Number of Timesteps in hour value);
- Hourly: produce output una volta ogni ora, come **valore medio nell'ora**;
- RunPeriod, Environment e Annual sono la stessa cosa, ovvero sono sinonimi, e fornisce dati o medi o integrali.

(le descrizioni precedenti, ad eccezione di quella "oraria" mancante, sono state prese direttamente dall'interfaccia dell'Editor IDF)

Nel InputOutputReference Manual, a pgg. 85-86 e 1770, è scritto che:

1. le variabili relative ai dati sull'ambiente esterno (come Zone,Average,Site Outdoor Air Drybulb Temperature [C] e Zone,Average,Site Solar Altitude Angle [deg]) sono disponibili al timestamp e con risoluzioni più elevate e possono essere interpolate da punti orari (rif. Interpolazione oraria dei dati meteorologici). **La maggior parte dei valori dei dati rappresenta la "media" nel periodo di risoluzione del reporting.**
2. al fine di abbinare i dati orari in modo continuo per EnergyPlus (dove i timestep potrebbero essere inferiori all'ora), viene eseguita un'**interpolazione lineare** semplice tra i valori di "ultima ora" e "di questa ora" ("*Interpolazione dell'ultima ora*").

- nelle versioni precedenti alla V1.0.3, l'interpolazione è stata fatta tra "quest'ora" e "l'ora successiva" ("Interpolazione dell'ora successiva").
- L'ora 1 per EnergyPlus è l'intervallo di tempo dalle 00:00:01 alle 1:00:00.** L'ora 2 è dalle 1:00:01 alle 2:00:00 (hh:mm:ss). Con lo schema di interpolazione ("Interpolazione dell'ultima ora") e la comunicazione dei valori dei dati meteorologici con la risoluzione del timestep, i valori "orari" riportati dovrebbero corrispondere ai valori dei dati meteorologici orari. (Nota che la risoluzione "hourly" del reporting non produrrà output a meno che il "Number of Timesteps in hour value" sia pari a 1).
- Notare che **questa interpolazione si applica solo ai dati esterni: i valori relativi ai profili d'uso, funzionamento, ecc. non vengono interpolati in questo modo.**

Nell'esempio di dati meteorologici di DRYCOLD, al timestamp 7:00:00 AM è associata una temperatura a bulbo secco del valore di 3,3 °C. EnergyPlus assocerà alla stessa etichetta temporale (timestamp) 7:00:00 AM:

- un valore di 3,3 °C, per una simulazione con un timestep di 60 min e una frequenza di reporting impostata su "hourly" o "timestep" (Figura 4.4b),
- un valore di 2,0625 °C, per una simulazione con un intervallo di tempo di 15 minuti e una frequenza di reporting impostata su "hourly". Questo valore si ottiene calcolando la media dei valori interpolati nei precedenti intervalli di tempo tra le 6:00:01 e le 7:00:00 (Figura 4.4a).

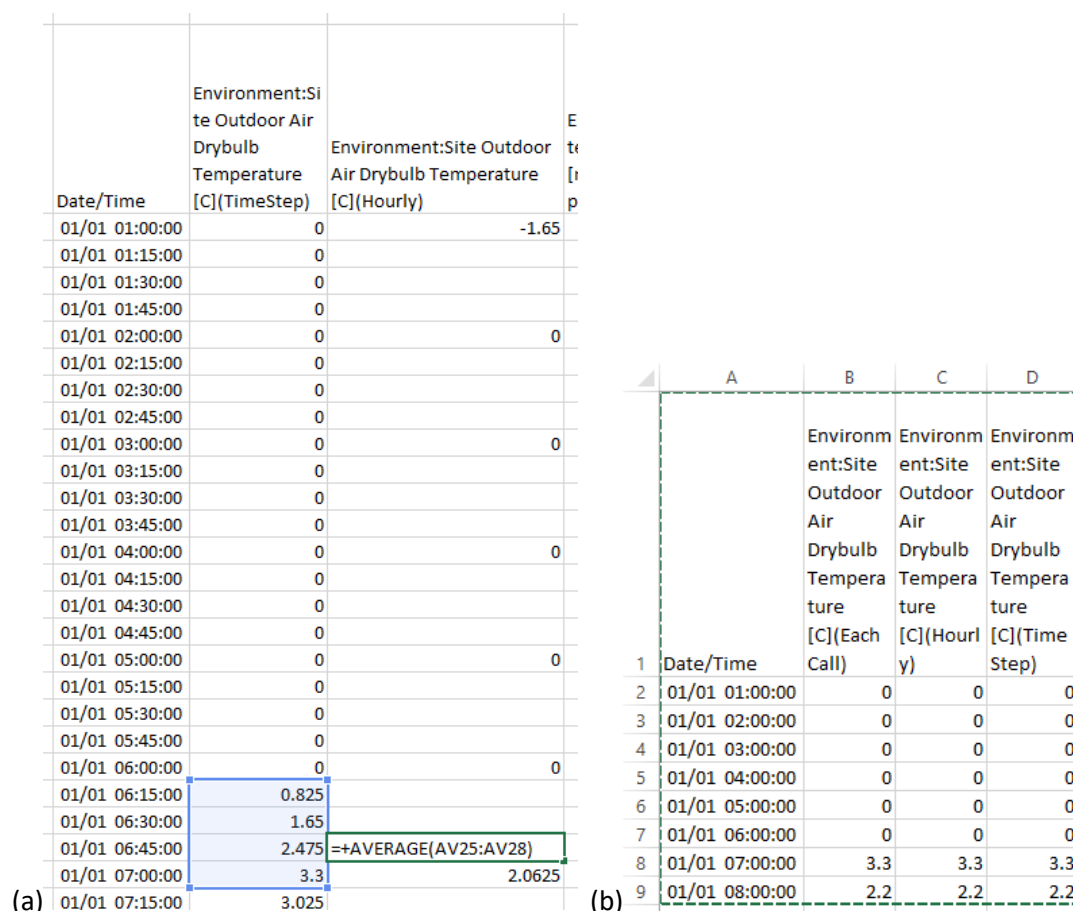


Figura 19 – Timestep vs frequenza di reporting oraria, quando la simulazione è stata eseguita con un timestep di 15 minuti (a) e 60 minuti (b).

Questo comportamento è ambiguo, poiché il valore medio "orario" alle 7:00:00 del dato climatico, tra le 6:00:01 e le 7:00:00 dovrebbe essere unico indipendentemente dal passo di simulazione utilizzato. Dal secondo caso sembra che il valore 3.3 °C sia associato da EnergyPlus al timestamp 6:30:00 AM, mentre dal

primo caso la sua posizione temporale non è più chiara (dovrebbe essere il valore medio tra 6:45:01 e 7:00:00?).

Dal momento che comprendere l'approccio seguito da EnergyPlus in merito a questo aspetto è alquanto difficile tramite la consultazione dei manuali e analizzando gli output stampati (che, per quel che si sa, potrebbero anche essere dati solo forniti in output a titolo informativo per l'utente e non necessariamente utilizzati da nessuno dei moduli del codice), si è analizzato direttamente il codice sorgente, analisi che ha portato alle seguenti considerazioni:

- la posizione del sole è calcolata con i metodi "SetCurrentWeather()" e "DetermineSunUpDown (SOLCOS);" al timestamp corrente;
 - $CurrentTime = (HourOfDay - 1) + TimeStep * (TimeStepFraction);$
 - $HrAngle = (15.0 * (12.0 - (CurrentTime + TodayVariables.EquationOfTime)) + (TimeZoneMeridian - Longitude));$
 - $CosZenith = SinLatitude * TodayVariables.SinSolarDeclinAngle + CosLatitude * TodayVariables.CosSolarDeclinAngle * std::cos(H);$
- nell'interpolazione dei dati meteorologici che non sono correlati alla radiazione solare, il dato fornito dal file di input si trova associato al timestamp fornito (come in OpenBPS) (anche se il campo relativo ai minuti che descrivono questo timestamp è ancora ignorato. Inoltre, nel passaggio dal formato TMY ai file in formato EPW questo campo è impostato su 60 minuti. **Questo, se il campo non fosse ignorato, avrebbe l'effetto di spostare di un'ora il dato meteorologico**);
- per l'interpolazione dei dati meteorologici correlati alla radiazione solare, il dato fornito dal file di input si trova al centro del timestep tra i dati meteorologici e l'interpolazione è "indietro" o "in avanti" a seconda del timestamp corrente (come in OpenBPS);
- alcuni dei valori che, leggendo il manuale, ci aspetteremmo essere mediati nell'intervallo rappresentato da ciascun timestamp, sembrano essere valori istantanei al timestamp;
- N.B.: data la considerazione precedente, anche gli output del calcolo potrebbero essere considerati come valori istantanei al timestamp, se stampati con frequenza del passo temporale e se la simulazione è stata eseguita con un passo temporale sub-orario.

4.2.3.3 *Dettagli su TRNSYS*

L'approccio di TRNSYS, come già visto, è invece il seguente:

- si associa a un timestamp, un intervallo di tempo, corrispondente al passo temporale (timestep) tra timestamp corrente e precedente,
- si associa a ciascun timestamp-timestep che è interno al calcolo, i dati meteorologici che vengono associati o valutati al centro del timestep associato e espone questi valori al solutore del bilancio termico dell'edificio per le sue condizioni al contorno;
- interpola linearmente tutti i dati meteorologici che non sono correlati alla radiazione solare, considerando i dati meteorologici localizzati al momento del timestamp.

Il trattamento della radiazione solare è particolare e verrà trattato nel paragrafo corrispondente.

4.3 Radiazione solare

Per il calcolo del contributo della radiazione solare al bilancio termico di un edificio o per la determinazione dei sistemi di conversione che sfruttano tale fonte rinnovabile di energia, occorre disporre del valore istantaneo di radiazione, l'irradianza solare, nelle due componenti, diretta e diffusa, rispetto ad una superficie comunque orientata. Questo è il compito del "processore di radiazione solare", SunCalculator", che ricevuto in input il dato di radiazione al timestamp corrente dal processore dei dati climatici, lo elabora in modo opportuno per ricavare le quantità richieste. I compiti principali che tale processore deve assolvere sono:

1. il calcolo della posizione del sole, compresi alba e tramonto;
2. il calcolo dello scattering atmosferico per la radiazione diffusa e diretta
3. la sincronizzazione con gli altri dati climatici.

La posizione del sole può essere calcolata secondo tre diverse metodologie, con diversa complessità computazionale e diversi errori nel calcolo, come riassunto da Grena in [36].:

- il "Metodo della serie di Fourier" sviluppato da Spencer nel 1971 (semplice, ma con errori dell'ordine dei decimi di grado, come riportato da Grena [36]);
- algoritmo n° 5 proposto da Grena [36] (media complessità con errore massimo di 0,0027 °, come riportato da Grena [36]);
- il metodo Almanacco nautico (massima complessità con errore massimo di 0,0003 °, come riportato da Grena [36]).

4.3.1 Nomenclatura specifica e algoritmi base

Per il calcolo del flusso termico radiativo dovuto alla radiazione solare incidente su un generico elemento, comunque orientato rispetto alla direzione di provenienza della radiazione solare, nell'ipotesi di conoscere, istante per istante, i valori della irradianza solare nelle sue componenti diretta normale e diffusa sul piano dell'orizzonte, G_{bn} e G_{dh} , occorre considerare la relazione geometrica che esiste tra la posizione del sole nel cielo e la superficie ricevente. Tale relazione si riassume nella determinazione di quello che viene chiamato l'angolo di incidenza della radiazione solare sulla superficie considerata, indicato con il simbolo ϑ .

Nel codice OpenBPS sono stati implementati algoritmi diversi per il calcolo, in particolare, dell'angolo di incidenza della radiazione solare, per la comprensione dei quali è bene introdurre la nomenclatura adottata per descrizione del trattamento della radiazione solare. Di seguito si introducono simboli e termini principali.

4.3.1.1 Nomenclatura e formule base

G	irradianza solare, in W/m^2 , potenza per unità di superficie ricevente della radiazione solare;
I	irradiazione solare oraria, in J/m^2 , integrale orario dell'irradianza;
H	irradiazione giornaliera, in J/m^2 , integrale giornaliero dell'irradianza;
G_{bn}	irradianza solare diretta normale, in W/m^2 , irradianza misurata su una superficie perpendicolare alla direzione della congiunzione sole-terra,
G_b	irradianza solare diretta, in W/m^2 , irradianza su una superficie comunque orientata dovuta alla visione del disco solare da parte della superficie ricevente;
G_d	irradianza solare diffusa, in W/m^2 , irradianza su una superficie comunque orientata che riceve radiazione solare da ogni direzione con esclusione di quella proveniente direttamente dal disco solare;

G_T	irradianza solare totale, in W/m^2 , irradianza complessivamente ricevuta su una superficie comunque orientata ($G_T = G_b + G_d$);
$G_{b,h}$	irradianza solare diretta su un piano orizzontale, in W/m^2 ;
$G_{d,h}$	irradianza solare diffusa su un piano orizzontale, in W/m^2 ;
$G_{T,h}$	irradianza solare totale su un piano orizzontale, in W/m^2 ;
G_{sc}	costante solare, ovvero il flusso radiativo proveniente dal sole ricevuto su un'area unitaria di superficie (irradianza) perpendicolare alla direzione di propagazione della radiazione alla distanza media terra-sole all'esterno dell'atmosfera. Viene utilizzato un valore di G_{sc} di 1367 W/m^2 ($1.960 \text{ cal}/(\text{cm}^2 \text{ min})$, $433 \text{ Btu}/(\text{ft}^2 \text{ h})$ o $4.921 \text{ MJ}/(\text{m}^2 \text{ h})$). [Vedi Iqbal [37] per informazioni più dettagliate sulla costante solare.]

$$G_{sc} = 1367 \text{ W/m}^2 \quad (4.4)$$

G_{on}	irradianza solare extraterrestre (fuori dall'atmosfera) incidente sul piano normale alla radiazione al momento "n" (giorno dell'anno, con $n=1$ il 1° gennaio e $n=365$ il 31 dicembre). In effetti, la variazione della distanza terra-sole porta alla variazione del flusso di radiazioni extraterrestri nell'intervallo di $\pm 3,3\%$. Equazione (4.5) da Spencer [38]; equazione (4.6) da Iqbal [37].
----------	---

$$G_{on} = G_{sc} \left(1 + 0,033 \cos \frac{360 n}{365} \right) \quad (4.5)$$

oppure

$$G_{on} = G_{sc} (1,000110 + 0,034221 \cos B + 0,001280 \sin B + 0,000719 \cos 2B + 0,000077 \sin 2B) \quad (4.6)$$

con

$$B = (n - 1) \frac{360}{365} \quad (4.7)$$

γ_s	angolo azimutale solare, in radianti, angolo che la proiezione sul piano dell'orizzonte della congiungente sole-terra forma con la direzione sud (negativo verso est, positivo verso ovest);
α_s	angolo di altezza solare o elevazione, in radianti, angolo complementare all'angolo zenitale solare ($\alpha_s = \pi/2 - \vartheta_z$);
ϑ_z	angolo solare zenitale o zenit, in radianti, angolo che congiungente sole-terra forma con la normale al piano dell'orizzonte, il cui coseno si calcola come:

$$\cos \vartheta_z = \cos \delta \cos \phi \cos \omega + \sin \delta \sin \phi \quad (4.8)$$

δ	angolo di declinazione solare, in radianti; angolo tra il piano che contiene il circolo di illuminazione (cioè la circonferenza che divide la parte della terra illuminata da quella in ombra) e il piano che contiene l'asse terrestre;
ϕ	latitudine del sito in esame, in gradi;

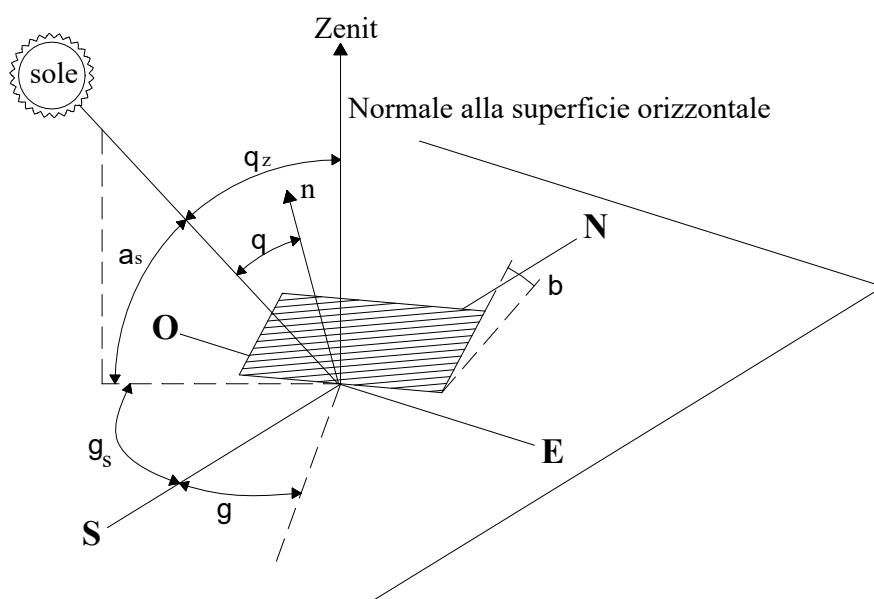


Figura 20 – Sistema di riferimento locale per la definizione della posizione del sole e dell'angolo di incidenza

ω angolo orario, in radianti, spostamento angolare del sole verso est o ovest del meridiano locale dovuto alla rotazione della terra sul suo asse di 15° all'ora; mattina negativo, pomeriggio positivo:

$$\omega = (t_s - 12) \cdot \pi/12 \quad (4.9)$$

t_s tempo solare, in ore, orario basato sul movimento angolare apparente del sole attraverso il cielo con il mezzogiorno solare nel momento in cui il sole attraversa il meridiano dell'osservatore; è calcolabile in funzione del tempo locale come:

$$t_s = t + \frac{1}{15}(L_{st} - L_{loc}) + \frac{12}{\pi}EOT \quad (4.10)$$

t tempo locale, in ore, ora locale definita a partire dall' Universal Time (UT) al meridiano di Greenwich (indicato anche come UT1), più la correzione del fuso orario rispetto all'ora di Greenwich, t_{grw} , senza prendere in considerazione eventuali correzioni legali;

$$t = t_{grw} - L_{st}/15 \quad (4.11)$$

L_{st} longitudine del meridiano di riferimento per il fuso orario locale, in gradi (Roma 15° est di Greenwich, = -15°);

L_{loc} longitudine del sito in esame, in gradi;

EOT equazione del tempo, in radianti, termine di correzione che tiene conto delle perturbazioni della velocità di rotazione terrestre nel calcolo del tempo in cui il sole attraversa il meridiano dell'osservatore, espresso in minuti. Sono due gli effetti astronomici che causano questa non uniformità nell'apparente movimento quotidiano del Sole: l'obliquità dell'eclittica (il piano del moto orbitale annuale della Terra intorno al Sole), che è inclinato di circa 23,44 gradi rispetto al piano dell'Equatore terrestre, e l'eccentricità dell'orbita terrestre attorno al Sole, che è circa 0,0167.

RA ascensione retta del sole, in radianti, distanza angolare fra il meridiano fondamentale (il coluro equinoziale) e il meridiano passante per la posizione del sole, misurata lungo il parallelo passante per la posizione del sole.

ϑ angolo di incidenza, in radianti, angolo che congiungente sole-terra forma con la normale alla genica superficie comunque orientata, il cui coseno è calcolabile, in funzione dell'angolo orario e della declinazione solare, ω e δ , *coordinate equatoriali*, come;

$$\cos \vartheta = \sin \delta (\sin \phi \cos \beta - \cos \phi \sin \beta \cos \gamma) + \cos \delta (\cos \phi \cos \beta + \sin \phi \sin \beta \cos \gamma) \cos \omega + \cos \delta (\sin \beta \sin \gamma) \sin \omega \quad (4.12)$$

oppure, in funzione degli angoli zenitali e azimutali solari, ϑ_z e γ_s , *coordinate orizzontali*,

$$\cos \vartheta = \cos \vartheta_z \cos \beta + \sin \vartheta_z \cos(\gamma_s - \gamma) \quad (4.13)$$

β angolo di inclinazione della superficie ricevente sul piano dell'orizzonte, in radianti;

γ angolo azimutale della superficie, in radianti, angolo che la proiezione sul piano dell'orizzonte della normale alla superficie forma con la direzione sud (negativo verso est, positivo verso ovest);

4.3.1.2 Irradianza solare

Se il dato misurato è disponibile corrisponde a un'irradianza (attributo solare verrà omissso nel seguito), si ha, con riferimento a Figura 23, misurando direttamente la totale e la diffusa:

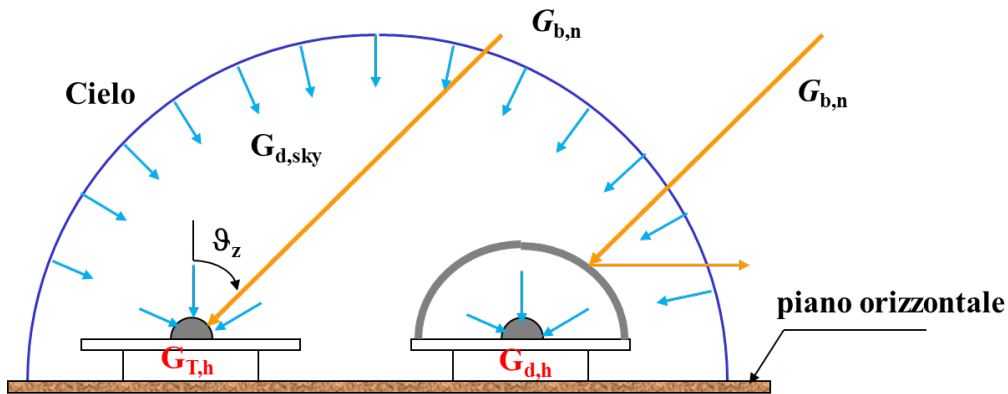


Figura 21 – Misura dell'irradianza diretta e diffusa su un piano orizzontale con due piranometri di cui uno dotato di banda oscurante

$$G_{b,h} = G_{T,h} - G_{d,h} \quad (4.14)$$

dove $G_{d,h} \equiv G_{d,sky}$, avendo indicato con quest'ultima la radiazione solare che arriva sulla superficie orizzontale da tutta la volta del cielo (l'emisfero sopra l'orizzonte).

Per ricavare il dato necessario, l'irradianza diretta normale G_{bn} , occorre introdurre il modello che determina la posizione del sole per poter determinare l'angolo solare zenitale, ϑ_z , e quindi calcolare la G_{bn} come:

$$G_{bn} = G_{b,h} / \cos \vartheta_z \quad (4.15)$$

Su una generica superficie comunque orientata (angolo di inclinazione β e angolo azimutale γ), incide non solo la radiazione diretta, $G_{bn} \cos \vartheta$, e quella diffusa dalla parte di volta del cielo vista dalla superficie, $G_{d,sky \rightarrow (\beta,\gamma)}$, ma anche (in assenza di ostruzioni) la radiazione solare riflessa dal piano dell'orizzonte $G_{d,hor \rightarrow (\beta,\gamma)}$, supposta diffusa, cioè, con riferimento a Figura 22,:

$$G_T = G_{bn} \cos \vartheta + G_{d,sky \rightarrow (\beta,\gamma)} + G_{d,hor \rightarrow (\beta,\gamma)} \quad (4.16)$$

Nell'ipotesi di **cielo isotropo e di riflessione diffusa** della radiazione solare da parte dal piano dell'orizzonte, se due component diffuse possono calcolarsi come:

$$G_{d,sky \rightarrow (\beta,\gamma)} = G_{d,h} \cdot F_{s,sky} = G_{d,h} \cdot \frac{1 + \cos \beta}{2} \quad (4.17)$$

e

$$G_{d,hor \rightarrow (\beta,\gamma)} = \rho_h G_{T,h} \cdot F_{s,hor} = \rho_h G_{T,h} \cdot \frac{1 - \cos \beta}{2} \quad (4.18)$$

con ρ_h coefficiente di riflessione del piano dell'orizzonte (detto anche coefficiente di albedo).

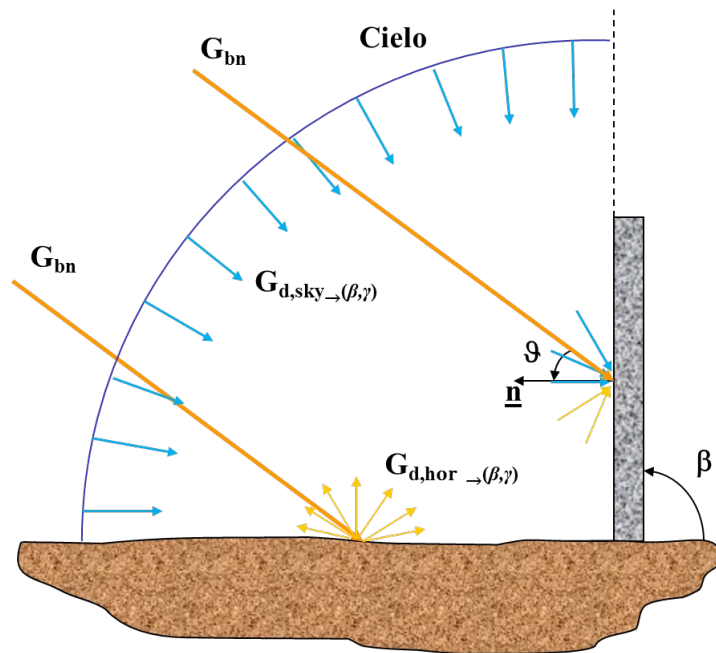


Figura 22 – Componenti dell'irradianza totale solare per una superficie orientata (β,γ)

Da cui la formula, che consente di calcolare l'irradianza totale su una superficie comunque orientata in funzione dei dati solitamente misurati e quindi disponibili, è, **nell'ipotesi di cielo isotropo e orizzonte non oscurato e riflettente in modo diffuso**, è:

$$G_T = (G_{T,h} - G_{d,h}) \cdot \frac{\cos \vartheta}{\cos \vartheta_z} + G_{d,h} \frac{1 + \cos \beta}{2} + \rho_h G_{T,h} \cdot \frac{1 - \cos \beta}{2} \quad (4.19)$$

4.3.2 Inizializzazione – ora di alba e di tramonto

Per potere attribuire correttamente il dato, solitamente integrale, di radiazione solare ad ogni istante del calcolo (timestamp) occorre conoscere l'ora, in tempo locale, a cui sorge e a cui tramonta il sole, valori che vengono predeterminati dal codice in fase di inizializzazione.

La procedura di inizializzazione, infatti, chiama il metodo `SunCalculator.SetSunCalculator`, la cui responsabilità è anche quella di chiamare il metodo `SetSunUpSunDown` per il pre-calcolo di tutti gli angoli di ora dell'alba e del tramonto, per ogni giorno della simulazione.

L'ora di alba e di tramonto possono essere calcolati secondo i tre diversi algoritmi implementati, chiamando il metodo specifico, ad esempio, per il calcolo del tramonto:

- con il "Metodo della serie di Fourier", il metodo chiamato è `public static void SetSunRise(int month, int day)`, che dipende solo dal mese e dal giorno dell'anno;
- con l'algoritmo n°5, il metodo chiamato è `public static void SetSunRise2(int year, int month, int day)`, che dipende dal giorno esatto (anno incluso);
- con il metodo dell'Almanacco Nautico, il metodo chiamato è `public static void SetSunRise(int year, int month, int day)`, che, come con il precedente, dipende dal giorno esatto (anno incluso).

L'ora dell'alba (le stesse considerazioni possono essere applicate al tramonto), che dipende dall'angolo di declinazione solare δ e dalla latitudine del sito ϕ considerato, si calcola con la seguente equazione che determina l'angolo orario di alba, $\omega_{sr} = -|\omega_s|$, simmetrico con l'angolo orario del tramonto, $\omega_{ss} = +|\omega_s|$:

$$\omega_s = -\tan \phi \tan \delta (n) \quad (4.20)$$

con n numero del giorno (preso spesso intero).

Se si vuole determinare l'ora di alba e tramonto in modo più preciso considerando che la declinazione va calcolata all'ora d'alba o tramonto, occorre implementare un processo iterativo, dovendo risolvere in realtà un'equazione trascendente in ω_s :

$$\omega_s = -\tan \phi \tan \delta (\omega_s) \quad (4.21)$$

I passaggi principali, comuni ai tre diversi metodi, sono:

1. determinare una prima declinazione di tentativo assumendo come ora per il calcolo il mezzogiorno solare del giorno considerato uno dei tre algoritmi implementati; cioè $\delta(n, \omega = 0)$
2. utilizzando tale declinazione, calcolare l'angolo dell'ora dell'alba con l'equazione (4.20);
3. ottenere l'ora locale dell'alba dall'equazione (4.9), cioè $\rightarrow t_{s,s} = 12 + \omega_s 12/\pi$;
4. eseguire le seguenti operazioni fintanto che l'ultima verifica non venga soddisfatta:
 - a. utilizzare uno dei tre algoritmi implementati per ottenere la posizione del sole per l'ora locale determinata,
 - b. calcolare il coseno dell'angolo zenitale solare di quella posizione
 - c. aggiungere o sottrarre un minuto all'ora locale dell'alba se il coseno dell'angolo zenitale non è uguale a zero
 - d. se il controllo sullo zero non è soddisfatto entro una tolleranza predefinita si ripetono i passaggi precedenti.

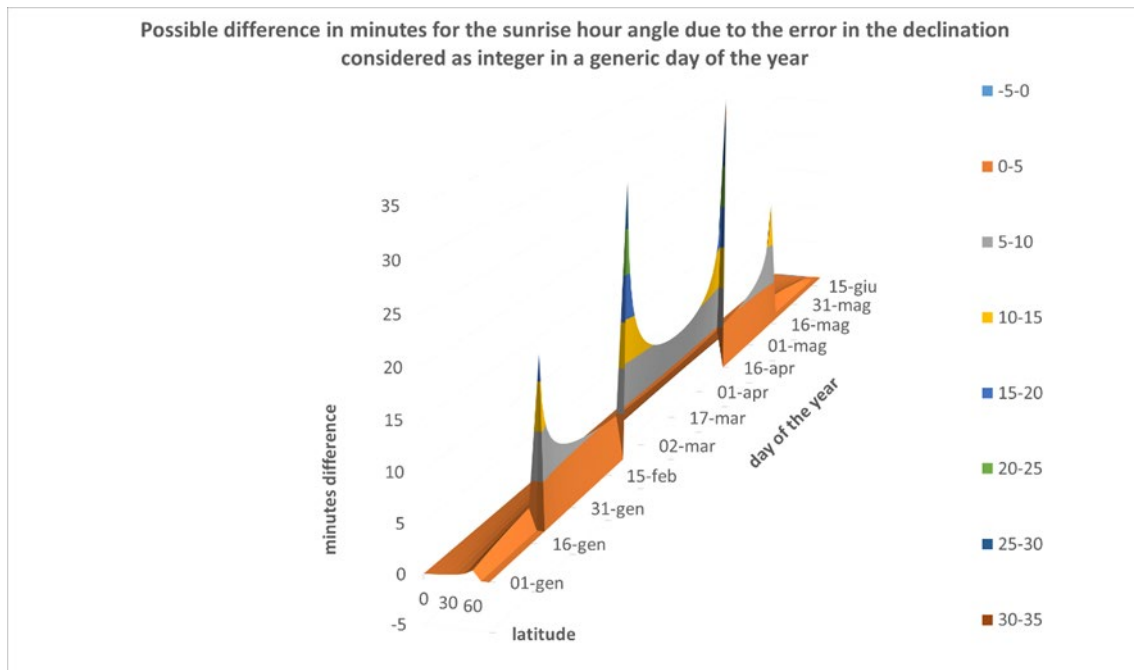


Figura 23 – Errore in minuti sull’angolo d’alba/tramonto se calcolato con un angolo di declinazione definito solo tramite in numero del giorno considerato un intero.

In Figura 23 è riportato l’errore che si commette nel calcolare l’angolo di declinazione solare da utilizzare nell’equazione (4.20) con riferimento al solo numero del giorno considerato intero, cioè con una declinazione sempre riferita allo zenit del sole.

4.3.3 Calcolo della posizione del sole: metodo della Serie di Fourier

Il "Metodo della serie di Fourier" sviluppato da Spencer [38], è il metodo principale utilizzato in campo ingegneristico ed è quello descritto anche nel noto libro di Duffie and Beckman [39], di cui si adotta la nomenclatura.

Questo metodo si basa sulla determinazione diretta delle coordinate solari equatoriali, cioè la declinazione solare, δ e l’angolo orario, ω , funzione dell’equazione del tempo (EOT), entrambe oggetto di un “best-fit” su dati storici eseguito con una serie di Fourier troncata. Infatti, per la precisione richiesta ai fini dei calcoli di prestazione energetica sia di edifici sia di sistemi solari attivi, queste quantità sono state considerate cicliche sul periodo di un anno e sono state quindi espresse in termini di serie di Fourier.

La posizione del sole è determinata dalla sua declinazione e dall’angolo orario, che a sua volta è determinato dall’ora solare che viene calcolata attraverso l’equazione del tempo a partire dall’ora locale (equazioni (4.9) e (4.10)).

Indicando con Υ la velocità orbitale angolare media della Terra in gradi al giorno, questa è data da:

$$\Upsilon = 360^\circ/365.25 \quad [^\circ/\text{giorno}] \quad (4.22)$$

oppure

$$\Upsilon_{approx} = 360^\circ/365 \quad [^\circ/\text{giorno}] \quad (4.23)$$

L’angolo che viene utilizzato per determinare sia la declinazione che l’equazione del tempo tramite una serie di Fourier approssimata, nel primo caso al terzo ordine, nel secondo al secondo ordine, di fatto coincide con

l'angolo B dato dall'equazione (4.7). Però per mettere in evidenza l'effetto di diverse assunzione che spesso vengono fatte in modo arbitrario, si mantiene qui per tale angolo la simbologia di Spencer, T (4.23), e viene calcolato con una velocità angolare terrestre approssimata, cioè, in radianti,:

$$T = d \cdot \gamma_{approx} = d \frac{2\pi}{365} \quad (4.24)$$

dove d è il numero del giorno così come fissato dall'Almanacco Nautico, cioè 0 per il 1° gennaio, 1 per il 2 gennaio, 355 per il 31 dicembre.

In altri riferimenti [39], il giorno dell'anno è identificato da " n " e si pone 1 al 1° gennaio. Tuttavia, se viene utilizzato all'interno dell'angolo sopra menzionato come B dato dall'equazione (4.7), essendovi $(n - 1)$ si ottiene lo stesso risultato.

Non è lo stesso per altri autori, come Vignola, Michalsky, & Stoffel [40], per cui tale giorno dell'anno n viene utilizzato senza sottrazione, cioè di fatto considerando d uguale a 1 il 1° gennaio.

Inoltre, dato che il tempo è una variabile continua, in un determinato istante, il numero del giorno n o d dovrebbe essere un numero reale e non un intero, così come viene normalmente assunto.

Per verificare la sensibilità a questa scelta (intero o reale), è stato aggiunto al codice un campo consentire di aggiornare il numero del giorno ad ogni intervallo di tempo definito dall'utente: `DeltaMinUpdateFractionalYear`, posto per default pari a 1440 min = 24 ore = 1 giorno. In questo caso, numero del giorno, e di conseguenza, l'angolo di declinazione e l'equazione del tempo, sono tenuti costanti per l'intero giorno considerato.

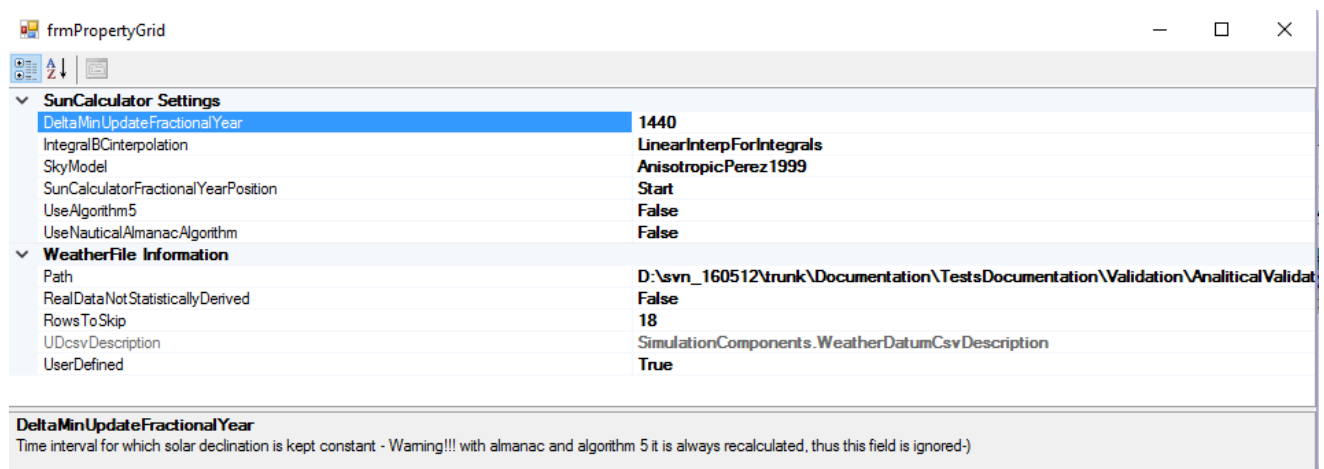


Figura 24 – Scelta sul numero del giorno: intero o reale

Sempre per motivi di verifica, l'utente può decidere se il giorno dell'anno d , che compare nell'equazione (4.24), debba essere calcolato all'inizio (uguale a 0 il 1° gennaio), a metà (uguale a 0,5 il 1° gennaio) o alla fine (uguale a 1 il 1° gennaio) del giorno impostando il campo: `SunCalculatorFractionalYearPosition`. In questo caso la declinazione potrebbe avere una differenza massima nel valore assoluto di massimo 0,4°, se calcolata impostando il giorno dell'anno all'inizio o alla fine della giornata.

Determinato il giorno dell'anno e quindi l'angolo T , si può calcolare sia la declinazione che la correzione dell'ora locale attraverso l'equazione del tempo.

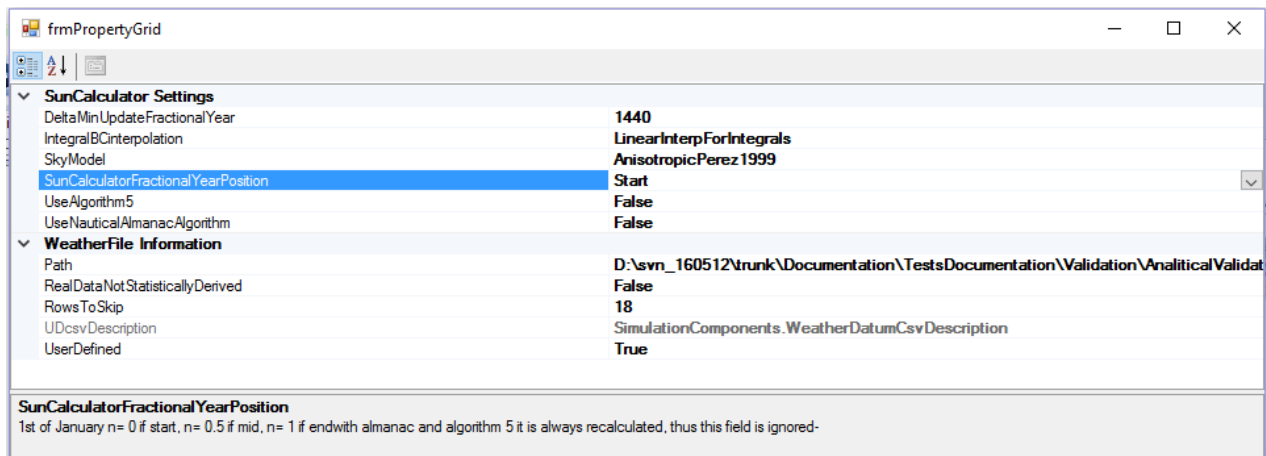


Figura 25 – Scelta sul numero del giorno: ora di partenza del riferimento nel giorno

L'angolo di declinazione secondo Spencer è dato, espresso in radianti, da:

$$\delta = 0.006918 - 0.399912 \cos T + 0.070257 \sin T - 0.006758 \cos 2T + 0.000907 \sin 2T - 0.002697 \cos 3T + 0.001480 \sin 3T \quad (4.25)$$

Per quanto riguarda l'equazione del tempo, una comunicazione personale tra J.W. Spencer, l'autore della formula dell'equazione del tempo, e Mac Oglesby, sottolinea che c'era un errore nella prima stampa e che l'equazione corretta del tempo EOT in radianti, ottenuta dalla serie di Fourier, è:

$$EOT = 0.0000075 + 0.001868 \cos T - 0.032077 \sin T - 0.014615 \cos 2T - 0.040849 \sin 2T \quad (4.26)$$

Se si vuole esprimere tale equazione in minuti, si ottiene, considerando che si ha una correzione dell'ora locale pari a 4 minuti per grado, un fattore di correzione pari a $4 \cdot 180/\pi = 229.18$ min/rd, che va a moltiplicare l'equazione (4.26).

L'equazione (4.26), insieme con l'equazione (4.10) e la (4.9), consente di determinare la seconda coordinata equatoriale del sole, l'angolo orario ω in funzione del tempo locale standard.

Spencer [38] sottolinea l'importanza di utilizzare questi dati per un anno nel mezzo di un ciclo di 4 anni a partire da un bisestile, poiché questo ciclo provoca cambiamenti nella declinazione che induce variazioni dell'ordine di $\pm 10'$ agli equinozi, dove l'effetto è maggiore, a meno di $1'$ nei solstizi, dove l'effetto è minore (riporta inoltre che altre variazioni a lungo termine nella declinazione sono trascurabili rispetto a queste).

4.3.4 Calcolo della posizione del sole: algoritmo n° 5 di Grena

L'algoritmo n° 5 di Grena [36], per ridurre l'errore massimo ben al di sotto di $0,01^\circ$, introduce alcune correzioni di chiara interpretazione fisica (ad esempio, l'effetto di Luna, Venere e Giove) ed altre correzioni armoniche con un significato meno chiaro, dovute ad azioni più complesse (come l'influenza combinata di altri pianeti distanti). In quest'algoritmo la posizione del sole è determinata in funzione delle coordinate orizzontali, angoli zenitali e azimutali del sole, ϑ_z e γ_s .

4.3.4.1 Punto di partenza: calcolo della scala temporale

Il tempo, t_g , utilizzato in quest'algoritmo è il numero di giorni, espresso come numero reale (per tenere conto di ore, minuti e secondi), a partire dall'inizio dell'anno 2060 (circa il punto medio dell'intervallo 2010–2110), secondo l'Universal Time (UT). I passaggi iniziali sono semplicemente la conversione della data e delle ore, espresse in tempo locale standard, in questa scala temporale, espressa in giorni, tramite la formula:

$$t_g = \text{INT}[365,25 (\tilde{y} - 2000)] - \text{INT}[30,6001 (\tilde{m} + 1)] - \text{INT}[0,01 \tilde{y}] + d_m + h/24 - 21958$$

(4.27) con,

$$\begin{aligned} \tilde{y} &= \text{anno} - 1 & ; & \tilde{m} = n^\circ \text{ mese} + 12 & \quad \text{se } n^\circ \text{ mese} \leq 2 \\ \tilde{y} &= \text{anno} & ; & \tilde{m} = n^\circ \text{ mese} & \quad \text{se } n^\circ \text{ mese} > 2 \end{aligned}$$

dove d_m è il numero del giorno nel mese (i.e. da 1 a 31 massimo) e h l'ora, comprensiva di minuti e secondi espressi come frazioni d'ora, in UT, cioè l'ora standard locale meno la correzione del fuso orario rispetto all'ora di Greenwich.

Oltre al tempo t , dipendente dalla rotazione terrestre (si basa sul tempo universale, UT), viene calcolato un altro tempo, t_e , basato sul Terrestrial Time (TT) (tempo siderale) e indipendente dalla rotazione terrestre, aggiungendo a t_g la differenza $\Delta\tau$ (convertita in giorni), cioè:

$$t_e = t_g + 1,574 \cdot 10^{-5} \cdot \Delta\tau \quad [\text{giorni}] \quad (4.28)$$

con

$$\Delta\tau = 96,4 + 0,00158 t_g \quad [\text{s}] \quad (4.29)$$

4.3.4.2 Calcolo della longitudine eclittica eliocentrica

La parte più intensiva dal punto di vista computazionale di questo metodo è quella relativa al calcolo della longitudine eclittica eliocentrica della Terra, L , che è calcolata come espansione in serie armonica rispetto al tempo t_e :

$$\begin{aligned} L &= L_0 + L_1 t_e + \sum_{k=1}^3 [a_k \sin(k\omega_a t_e) + b_k \cos(k\omega_a t_e)] + d_\beta \sin(k\omega_a t_e) \sin(\beta t_e) \\ &\quad + \sum_{i=1}^7 d_i \sin(\omega_i t_e + \phi_i) \end{aligned} \quad (4.30)$$

dove i valori dei vari parametri sono definiti in Tabella 1.

L'ultima frequenza ω_7 corrisponde a un periodo di circa 29,5 giorni e rappresenta chiaramente l'effetto della luna. Altre due frequenze facilmente interpretabili sono ω_3 , corrispondenti a 584 giorni, e ω_4 , corrispondenti a 399 giorni: questi sono, rispettivamente, i periodi tra le due congiunzioni di Venere e Giove con la Terra.

4.3.4.3 Correzione per la nutazione¹

La nutazione agisce con una frequenza $\omega_n = 9,282 \cdot 10^{-4}$ giorni⁻¹. Viene calcolato l'angolo $\nu = \omega_n t_e - 0,8$ e la correzione della longitudine geocentrica eclittica apparente è data da $\Delta\lambda = 8,24 \cdot 10^{-5} \sin \nu$. La longitudine geocentrica eclittica apparente del sole è data quindi da:

Tabella 1– Costanti numeriche per il calcolo della longitudine eclittica eliocentrica

L_0	1.75279
-------	---------

¹ La nutazione è il moto di oscillazione (etimologicamente significa in latino classico "un cenno del capo") dell'asse di rotazione di un oggetto, che si manifesta in combinazione con un moto di precessione.

L_1	$1.720279 \cdot 10^{-2}$					
ω_a	0.017202	a_1	$3.33024 \cdot 10^{-2}$	b_1	$-2.0582 \cdot 10^{-3}$	
		a_2	$3.512 \cdot 10^{-4}$	b_2	$-4.07 \cdot 10^{-5}$	
		a_3	$5.2 \cdot 10^{-6}$	b_3	$-9 \cdot 10^{-7}$	
β	$2.92 \cdot 10^{-5}$	d_β	$8.23 \cdot 10^{-5}$			
ω_1	$1.49 \cdot 10^{-3}$	d_1	$1.27 \cdot 10^{-5}$	φ_1	-2.337	
ω_2	$4.31 \cdot 10^{-3}$	d_2	$1.21 \cdot 10^{-5}$	φ_2	3.065	
ω_3	$1.076 \cdot 10^{-2}$	d_3	$2.33 \cdot 10^{-5}$	φ_3	-1.533	
ω_4	$1.575 \cdot 10^{-2}$	d_4	$3.49 \cdot 10^{-5}$	φ_4	-2.358	
ω_5	$2.152 \cdot 10^{-2}$	d_5	$2.67 \cdot 10^{-5}$	φ_5	0.074	
ω_6	$3.152 \cdot 10^{-2}$	d_6	$1.28 \cdot 10^{-5}$	φ_6	1.547	
ω_7	$2.1277 \cdot 10^{-1}$	d_7	$3.14 \cdot 10^{-5}$	φ_7	-0.488	

$$\lambda = L + \pi + \Delta\lambda \quad (4.31)$$

(il fattore π è dovuto al cambiamento del sistema di riferimento, da eliocentrico a geocentrico).

Si calcola quindi la correzione dell'inclinazione dell'asse terrestre, come:

$$\epsilon = 4,089567 \cdot 10^{-1} - 6,19 \cdot 10^{-9} \cdot t_e + 4,46 \cdot 10^{-5} \cos v \quad (4.32)$$

Si calcolano l'ascensione retta RA e la declinazione δ come:

$$RA = \arctan2(\sin \lambda \cos \epsilon, \cos \lambda) \quad (4.33)$$

NOTA: La funzione $\arctan2(x, y)$ è presente in molti linguaggi di programmazione; in C# è il metodo $\text{Atan2}(\text{Double}, \text{Double})$, e restituisce l'angolo la cui tangente è il quoziente di due numeri specificati.

$$\delta = \sin^{-1}(\sin \lambda \sin \epsilon) \quad (4.34)$$

Viene quindi calcolato l'angolo orario H , con la correzione della nutazione:

$$H = 1,7528311 + 6,300388099 \cdot t + L_{loc} - RA + 0,92 \cdot \Delta\lambda \quad (4.35)$$

dove L_{loc} è la longitudine del sito considerato.

Il valore di H deve essere sostituito dall'angolo corrispondente ω nel suo intervallo convenzionale

$$\omega = \text{MOD}(H + \pi, 2\pi) - \pi \quad (4.36)$$

L'ascensione retta, RA , essendo il risultato di una funzione $\arctan2$, è compresa nell'intervallo $[-\pi, \pi]$; quando è negativa, deve essere spostata in avanti di 2π per ottenere l'intervallo convenzionale corretto.

4.3.4.4 Calcolo dell'angolo azimutale e zenitale

L'angolo azimutale solare, γ_s , si calcola in funzione dell'angolo orario e della declinazione, come:

$$\gamma_s = \arctan2(\sin \omega, \cos \omega \sin \varphi - \tan \delta \cos \varphi) \quad (4.37)$$

angolo a cui non va applicata alcuna correzione per il passaggio da coordinate topocentriche a geocentriche, giacché gli effetti sono trascurabili.

L'angolo zenitale solare, ϑ_z , viene calcolato come l'angolo complementare dell'angolo di elevazione solare:

$$\vartheta_z = \pi/2 - \alpha_s \quad (4.38)$$

L'angolo di elevazione solare, senza correzione per la rifrazione e l'errore di parallasse, è dato da

$$\alpha_{s,0} = \sin^{-1}(\cos \varphi \cos \delta \cos \omega + \sin \varphi \sin \delta) \quad (4.39)$$

mentre la correzione per l'errore di parallasse si calcola come:

$$\Delta_p \alpha_s = -4,26 \cdot 10^{-5} \cos \alpha_{s,0} \quad (4.40)$$

E quindi l'elevazione corretta dell'errore di parallasse è data da:

$$\alpha_{s,p} = \alpha_{s,0} + \Delta_p \alpha_s \quad (4.41)$$

Per la correzione dovuta alla rifrazione si impiega la formula di Reda and Andreas [41], data da:

$$\Delta_r \alpha_s = \frac{0,08422 \cdot P}{273 + T} \frac{1}{\tan \left(\alpha_{s,p} + \frac{0,003138}{\alpha_{s,p} + 0,08919} \right)} \quad (4.42)$$

dove P è la pressione atmosferica in *atm* e T è la temperatura dell'aria in °C.

Quindi, il valore corretto dell'elevazione è:

$$\alpha_s = \alpha_{s,p} + \Delta_r \alpha_s \quad (4.43)$$

da cui si può calcolare l'angolo zenitale come complementare a 90°, cioè $\vartheta_z = \pi/2 - \alpha_s$.

4.3.5 Calcolo della posizione del sole: metodo dell'Almanacco Nautico

Nell'algoritmo dell'Almanacco Nautico [42], le grandezze che determinano la posizione del sole sono primariamente le due coordinate equatoriali, l'angolo orario e la declinazione solare, ω e δ , da cui vengono derivate le due coordinate orizzontali, angoli zenitali e azimutali solari, ϑ_z e γ_s . In quest'algoritmo il tempo, t_a , è definito come la differenza in giorni tra la data giuliana, jd e la data giuliana $JD=2451545.0$, che corrisponde al mezzogiorno del 1 gennaio 2000 in Universal Time (UT), cioè:

$$t_a = jd - 2\,451\,545 \quad (4.44)$$

La data giuliana² corrente viene calcolata a partire da espressioni del tempo locale standard utilizzando la seguente relazione:

$$jd = 2\,432\,916,5 + 365 \, delta + bisestile + giorno + ora/24 \quad (4.45)$$

con

$$delta = anno - 1949 \quad (4.46)$$

e, per tener conto degli anni bisestili,

$$bisestile = \text{INT} (delta/4) \quad (4.47)$$

Il giorno è il giorno progressivo dell'anno a partire da 1 per il 1° gennaio e l'ora è l'UT in ore comprese le sue frazioni costituite dai minuti e dai secondi. La costante numerica nella formula di jd è la data giuliana per la mezzanotte del 0 gennaio 1949 UT. La scelta del 1949 è arbitraria, ma consente di lavorare con numeri positivi nel calcolo di jd per il secolo in questione. L'ora locale va specificata in termini di UT.

4.3.5.1 Coordinate eclittiche

Le coordinate dell'eclittica solare sono calcolate utilizzando le seguenti relazioni:

$$t_a = jd - 2\,451\,545.0 \quad n \in \mathbb{R} \quad (4.48)$$

per il tempo giuliano relativo al mezzogiorno del 1° gennaio 2000, t_a .

$$L_m = 280,460 + 0,9856474 \cdot n \quad (0 \leq L < 360^\circ) \quad (4.49)$$

per la longitudine media eclittica eliocentrica del sole, L_m .

$$g = 357,528 + 0,9856003 \cdot n \quad (0 \leq g < 360^\circ) \quad (4.50)$$

per l'anomalia media dell'eclittica, g .

$$L = L_m + 1,915 \cdot \sin(g) + 0,020 \cdot \sin(2g) \quad 0 \leq l < 360^\circ \quad (4.51)$$

per la longitudine eclittica eliocentrica, L .

$$e_p = 23,439 - 0,00000004 \cdot t_a \quad (^\circ) \quad (4.52)$$

per l'obliquità dell'eclittica

Si noti che la specifica $(0 \leq L < 360^\circ)$ implica che alla quantità L devono essere aggiunti o sottratti multipli di 360° fino a quando non ha un valore compreso nell'intervallo indicato.

4.3.5.2 Coordinate celesti

Per il calcolo dell'ascensione retta RA e della declinazione δ si usano le seguenti formule:

² La data giuliana è il numero di giorni trascorsi a partire dal primo gennaio dell'anno 4713 avanti Cristo.

$$RA = \tan^{-1}[\cos(e_p) \cdot \sin(L/\cos(L))] \quad (4.53)$$

o, in ore, $RA' = RA \cdot 12/\pi$; mentre per la declinazione, in radianti, si ha:

$$\delta = \sin^{-1}[\sin(e_p) \cdot \sin(L)] \quad (4.54)$$

La formula per l'ascensione è scritta in questa forma (nota che $\sin(L)/\cos(L) = \tan(L)$) perché si vuole assegnare all'ascensione un angolo compreso tra 0 e 360°. L'analisi dei segni del numeratore e del denominatore consente di ottenere facilmente tale risultato.

4.3.5.3 Coordinate locali

Per determinare l'angolo azimutale e l'angolo di altezza solare o elevazione, occorre conoscere la declinazione e l'angolo orario. Per calcolare l'angolo orario, occorre conoscere il tempo siderale medio locale (*lmst*) e l'ascensione retta.

Il *lmst* può essere definito come l'ascensione del punto sul meridiano all'equatore celeste. Con questa definizione, l'angolo orario viene rilevato sottraendo l'ascensione dal *lmst*. L'Almanacco tabula il tempo siderale medio di Greenwich (*gmst*) a mezzanotte e la formula per calcolare in modo approssimato *gmst* in un tempo arbitrario è data da:

$$gmst = 6,697375 + 0,0657098242 \cdot t_a + ora(UT) \quad (0 \leq gmst < 24 h) \quad (4.55)$$

La variabile "ora" non viene moltiplicata per un fattore per convertirla in ore siderali perché t_a include già il giorno frazionario e quindi tiene già conto di tale correzione. La longitudine est del sito in questione viene aggiunta a *gmst* per specificare *lmst*. Poiché la longitudine è solitamente espressa in gradi, la longitudine deve essere divisa per 15 per convertirla in ore. La longitudine est è considerata positiva. Perciò:

$$lmst = gmst + \frac{L_{loc}}{15} \quad (0 \leq lmst < 24 h) \quad (4.56)$$

e l'angolo orario H , in ore, è dato da:

$$H = lmst - RA' \quad (-12 < H \leq 12 h) \quad (4.57)$$

da cui il suo valore in radianti è $\omega = H \cdot \pi/12$.

Prima che il sole raggiunga il meridiano, il suo angolo orario è negativo e il suo angolo orario è positivo nell'emisfero postmeridiano. (Si noti che molti ricercatori che operano nel campo del solare usano per ω la convenzione di segno opposto con l'emisfero orientale positivo e l'emisfero occidentale negativo.)

Per calcolare l'angolo di altezza del sole (elevazione) e l'azimut, si impiegano le seguenti trasformazioni:

$$\sin \alpha_s = \cos \delta \cdot \cos \phi \cdot \cos \omega + \sin \delta \cdot \sin \phi \quad (4.58)$$

che coincide con l'equazione (4.8) che fornisce l'angolo zenitale ϑ_z , essendo $\sin \alpha_s = \cos \vartheta_z$; mentre

$$\sin \gamma_s = -\cos \delta \cdot \sin \omega / \cos \alpha_s \quad (0 \leq \gamma_s < 360^\circ) \quad (4.59)$$

L'angolo azimutale così calcolato è misurato dalla direzione nord, per cui $\gamma_s = 0$, e crescente verso est; solitamente invece si assume $\gamma_s = 0$ per la direzione sud e $-180^\circ \leq \gamma_s \leq +180^\circ$ negativo ad est e positivo ad ovest, per cui in tale convenzione occorre sottrarre 180° e riassegnare il corretto quadrante.

4.3.5.4 Correzione per rifrazione e Equazione del tempo

I calcoli sopra riportati sono riferiti al centro del disco solare, che ha un diametro di circa $1/2^\circ$ e per un'atmosfera non riflettente. Quando si osserva il sole vicino all'orizzonte, la rifrazione atmosferica è importante e nel calcolo dell'alba e del tramonto è necessario prendere in considerazione l'estensione angolare del sole, poiché si usa come punto di riferimento il bordo superiore del disco.

La correzione per rifrazione dell'Almanacco si calcola come:

Per $\alpha_s < 9,225^\circ$

$$\Delta_r \alpha_s = 3,51561 \cdot (0,1594 + 0,0196 \cdot \alpha_s + 0,00002 \cdot \alpha_s^2) / (1 + 0,505 \cdot el + 0,0845 \cdot \alpha_s^2) \quad (4.60)$$

per $\alpha_s \geq 9,225^\circ$

$$\Delta_r \alpha_s = 0,0159024 / \tan \alpha_s$$

dove $3,51561 = 1013,2 \text{ mbar} / 288,2 \text{ K}$, cioè il rapporto tra la pressione atmosferica e la temperatura dell'atmosfera standard degli Stati Uniti [43].

Da cui l'angolo di altezza solare corretto per l'effetto della rifrazione è:

$$\alpha_{s,r} = \alpha_s + \Delta_r \alpha_s \quad (4.61)$$

Per calcolare il diametro del sole in funzione del tempo, occorre conoscere la sua distanza dalla terra. Rispetto all'unità astronomica (valore 1 alla distanza media sole-terra), la distanza varia secondo la seguente relazione:

$$\Delta R_{S-T} = 1,00014 - 0,01671 \cdot \cos(g) - 0,00014 \cdot \cos(2g) \quad (4.62)$$

Il raggio del disco solare è quindi calcolato come:

$$r_s = 0,2666 / \Delta R_{S-T} \quad (4.63)$$

Queste correzioni si applicano solo a una particolare struttura dell'atmosfera, che si verifica raramente. È quindi veramente difficile che si abbia una precisione di $0,01^\circ$ quando si osserva il sole a basse elevazioni a causa di questa incertezza.

Infine siccome l'equazione (4.51) per la longitudine eclittica tiene conto delle differenze di lunghezza del giorno solare che danno origine alla correzione chiamata equazione del tempo, questa può essere calcolata come:

$$EOT = L - RA \quad (4.64)$$

che risulta così espressa o in gradi o in radianti.

4.3.6 Analisi critica degli algoritmi considerati per il calcolo della posizione del sole

Durante l'analisi dei vari algoritmi relativi al calcolo della posizione del sole, sono emersi alcuni aspetti critici:

1. un errore nella formula dell'equazione del tempo riportata in Spencer [38] e Duffie-Beckman [39];
2. la possibilità di utilizzare come giorno dell'anno un numero frazionario, anziché un numero intero, per avere una migliore precisione;

3. la possibilità o meno di simulare anche anni bisestili.

Per quanto riguarda il primo punto, in una comunicazione personale tra J.W. Spencer, l'autore della formula dell'equazione del tempo, e Mac Oglesby [44] è chiaramente affermato che vi è un errore di stampa nell'articolo di Spencer [38] e l'equazione corretta del tempo in radianti, ottenuta dalla serie di Fourier, è:

$$EOT = 0.0000075 + 0.001868 \cos T - 0.032077 \sin T - 0.014615 \cos 2T - 0.040849 \sin 2T \quad (4.65)$$

(uno zero in più nei decimali del termine costante in rosso).

Riguardo al secondo punto, si è introdotta la possibilità di scegliere per il calcolo posizione del sole e, quindi, dell'angolo di incidenza della radiazione solare su una superficie comunque orientata, il numero del giorno come:

1. intero 1° gennaio = 1
2. mezza giornata = 0.5
3. in funzione del dato climatico (se sub-orario):
 - a. end
 - b. middle
4. in funzione del timestamp

Per l'ultimo punto è stato introdotto il metodo dell'Almanacco Nautico, che oltre ad essere più preciso, consente di trattare in modo consistente anche gli anni bisestili, cosa che gli altri metodi non consentono.

4.3.7 Approccio di TRNSYS per il calcolo dell'irradianza solare

Il codice di simulazione BPSt TRNSYS, uno dei più noti e diffusi, nato inizialmente proprio per simulare il comportamento dei sistemi solari attivi, utilizza un metodo del tutto particolare per la gestione della radiazione solare. Infatti, siccome la maggior parte dei dati di radiazione solare disponibili sono in realtà dei dati integrati nel tempo dei valori "istantanei" di irradianza, solitamente di irradiazione oraria, il processore dei dati climatici, TYPE 15, non li interpola linearmente, qualora il timestep di simulazione sia inferiore al timestep con cui sono disponibili i dati climatici.

La base di partenza è l'assunto che il valore, ad esempio orario, di irradiazione per la generica componente y , è dato dall'integrale dell'irradianza su l'intervallo temporale con cui sono forniti i dati climatici, timestep dei dati, cioè:

$$I_y \Big|_{\omega_{d-1}}^{\omega_d} = \int_{\omega_{d-1}}^{\omega_d} G_y(\omega) d\omega \quad (4.66)$$

dove gli estremi dell'intervallo sono definiti dagli angoli orari $\omega_d(t_d)$ e $\omega_{d-1}(t_d - \Delta t_d)$ e l'irradianza è espressa in funzione dell'angolo orario $\omega(t)$ con $t \in (t_d - \Delta t_d) \ t_d$. Il dato integrale è associato al timestamp dei dati dell'estremo superiore, cioè t_d .

Per semplicità di notazione nel seguito si pone $t_d - \Delta t_d = t_{d-1}$ e $t_d + \Delta t_d = t_{d+1}$.

Il processore dei dati climatici assegna al timestamp corrente della simulazione, t , un valore del dato di radiazione che è:

- se il timestep di simulazione è di un'ora, e quindi uguale al timestep dei dati climatici, *sincrono* $\rightarrow t \equiv t_d$
- se il timestep di simulazione è inferiore all'ora (un suo sottomultiplo):

- se $t - \Delta t/2 > t_{d-1}$ → il valore attribuito a t è quello del dato al tempo t_{d+1}
- se $t - \Delta t/2 \leq t_{d-1}$ → il valore attribuito a t è quello del dato al tempo t_d

Ad esempio, se il timestep di simulazione è 15 minuti (0,25 ore), al timestamp di simulazione $t = 10$ si ha un valore di radiazione assegnata che corrisponde all'irradiazione tra le 9 e le 10 (timestamp dei dati $t_d = 10$); al timestamp $t = 10,25$ si ha assegnato un valore di radiazione che corrisponde all'irradiazione tra le 11 e le 10 (timestamp dei dati $t_d = 11$); stessa assegnazione si ripete per il timestamp di simulazione $t = 10,50$, $t = 10,75$ e $t = 11,00$, istante in cui il dato climatico viene aggiornato ($t_d = t_d + \Delta t_d$); solo quando si arriva a $t = 11,25$ il valore esposto viene aggiornato ed è quello corrispondente all'irradiazione tra le 12 e le 11 (timestamp dei dati $t_d = 12$). Quindi, il processore dei dati climatici assegna al timestamp della simulazione corrente un valore di radiazione sincrono se il timestamp della simulazione coincide con il timestamp del dato (ma sempre riferito all'intervallo precedente), e un valore antecedente per tutti i timestamp contenuti nell'intervallo temporale tra il timestamp del dato corrente e quello del dato nel futuro. Nel caso dell'esempio, il dato passato è sempre una radiazione integrata su un ora, mentre, se il timestep di simulazione è 15 minuti, servirebbe la radiazione integrata su 15 minuti. Chi si preoccupa di aggiustare le cose è il processore della radiazione solare: la procedura `getIncidentRadiation`.

La procedura `getIncidentRadiation` utilizza direttamente il dato integrale di radiazione, relativo al suo intervallo di integrazione $t_{d2} - t_{d1}$, che gli viene passato dal processore dei dati climatici. Partendo quindi dal presupposto di disporre di un dato integrale relativo ad un intervallo di tempo sempre maggiore o uguale al timestep di simulazione, la principale operazione di adattamento che il processore solare deve fare è quella di ricavare il valore da esporre ai vari solutori quando il timestep di simulazione è inferiore al timestep del dato integrale di radiazione. Se infatti indichiamo con t_x e t_{x-1} gli estremi superiore e inferiore del timestep di simulazione, e con ω_x e ω_{x-1} gli angoli orari calcolati in quei timestamp, per sincronizzazione dei dati si fa sempre in modo da avere $\omega_{d-1} \leq \omega_{x-1}$ e $\omega_d \geq \omega_x$. Quando l'alba o il tramonto cadono all'interno del passo di simulazione, il dato di radiazione viene attribuito tutto alla porzione del timestep relativo alla presenza del sole.

Nel caso ordinario, cioè al di fuori delle singolarità di alba e tramonto, il processore solare opera "stimando" l'irradiazione (quindi l'energia e non la potenza) relativa al timestep di simulazione (per tutto l'intervallo e non a un preciso istante nel tempo) in modo diverso se i dati di radiazione sono riferiti al piano orizzontale, $I_{T,h}$, $I_{b,h}$, $I_{d,h}$ o se uno di essi è l'irradiazione oraria diretta normale, I_{bn} .

4.3.7.1 Irradiazione diretta o totale sul piano orizzontale

Se il dato fornito dal file climatico è **irradiazione diretta o totale sul piano orizzontale**, l'irradiazione al timestamp della simulazione viene ricalcolata dal dato ricevuto come proporzionale al rapporto tra l'irradiazione extraterrestre sul timestep di simulazione e l'irradiazione extraterrestre sul timestep dei dati, e, **diversamente da quanto esposto nel manuale**, in funzione del rapporto inverso tra i due timestep, cioè proporzionale al rapporto delle irradianze medie nei due intervalli temporali:

$$I_{y,h}|_{\omega_{x-1}}^{\omega_x} = I_{y,h}|_{\omega_{d-1}}^{\omega_d} \cdot \frac{I_{o,h}|_{\omega_{x-1}}^{\omega_x}}{I_{o,h}|_{\omega_{d-1}}^{\omega_d}} \cdot \frac{t_d - t_{d-1}}{t_x - t_{x-1}} = I_{y,h}|_{\omega_{d-1}}^{\omega_d} \cdot \frac{\langle G_{o,h} \rangle_{\omega_{x-1}}^{\omega_x}}{\langle G_{o,h} \rangle_{\omega_{d-1}}^{\omega_d}} = I_{y,h}|_{\omega_{d-1}}^{\omega_d} \cdot R_h|_d^x \quad (4.67)$$

dove, per le equazioni (4.5) o (4.6) e la (4.8),

$$I_{o,h}|_{\omega'}^{\omega''} = \int_{\omega'}^{\omega''} G_{on} \cos \vartheta_z d\omega = G_{on} [\cos \delta \cos \phi (\sin \omega'' - \sin \omega') - \sin \delta \sin \phi (\omega'' - \omega')] \quad (4.68)$$

irradiazione misurata su un piano orizzontale alla latitudine del sito ma posto al di fuori dell'atmosfera, calcolata in modo approssimato, cioè **assumendo costante nel giorno la declinazione e la distanza terra-sole**. Da cui, il fattore di adattamento al timestep di simulazione, $R_h|_d^x$ diventa

$$R_h|_d^x = \frac{\cos \delta \cos \phi (\sin \omega_x - \sin \omega_{x-1}) - \sin \delta \sin \phi (\omega_x - \omega_{x-1})}{\cos \delta \cos \phi (\sin \omega_d - \sin \omega_{d-1}) - \sin \delta \sin \phi (\omega_d - \omega_{d-1})} \cdot \frac{t_d - t_{d-11}}{t_x - t_{x-1}} \quad (4.69)$$

NOTA: Il rapporto $\cdot \frac{I_0|_{\omega_{x-1}}^{\omega_x}}{I_0|_{\omega_{d-11}}^{\omega_d}}$ è un rapporto tra due integrali di una stessa quantità (l'irradianza extraterrestre) su intervalli di integrazione diversi, e quindi rappresenta la frazione di irradiazione extraterrestre che si ha nel timestep di simulazione rispetto al suo valore relativo al time step dei dati. Tale rapporto può essere correttamente utilizzato per stimare l'irradiazione sul piano orizzontale nel timestep della simulazione partendo dal dato fornito. Invece il rapporto tra le irradianze extraterrestri medie (così come implementato nel codice) non è consistente. Per verificare tale inconsistenza basta considerare l'esempio limite d'irradianza extraterrestre costante sull'ora, in tal caso $I_0|_{\omega_{x-1}}^{\omega_x} / I_0|_{\omega_{d-11}}^{\omega_d} = 0.25$ per un timestep di simulazione di 15 minuti e una disponibilità dei dati oraria: il dato integrale orario fornito se moltiplicato solo per tale rapporto risulta ridotto al 25% del suo valore, è quindi una stima sufficientemente corretta di quello che potrebbe essere il dato reale sui 15 minuti. Se invece si moltiplica ancora per il rapporto inverso degli intervalli temporali, $(t_d - t_{d-11}) / (t_x - t_{x-1})$, nel nostro esempio paria a 4, di fatto si annulla l'effetto della precedente correzione, ritornando quindi una irradiazione ancora riferita alla base dei dati e non al timestep di simulazione, **SOVRASTIMANDO** quindi la radiazione disponibile, **se poi venisse determinata la potenza media dividendo tale energia per l'intervallo di simulazione**. In realtà, il prodotto $(I_0|_{\omega_{x-1}}^{\omega_x} \cdot (t_x - t_{x-1}) / (I_0|_{\omega_{d-11}}^{\omega_d} \cdot (t_d - t_{d-11}))) \cdot ((t_d - t_{d-11}) / (t_x - t_{x-1}))$ non è mai uguale a 1 per la non linearità rispetto al tempo del primo rapporto. Di fatto, però, **il processore NON calcola la potenza media** rispetto al timestep di simulazione, ma espone il dato trattato, ottenuto ad esempio dalla (4.69), che è comunque un'irradiazione oraria in kJ/m^2 , come se fosse la potenza istantanea al timestamp in $\text{kJ}/(\text{h m}^2)$.

Di fatto, per come vengono forniti i dati dal processore di dati climatici, e da ciò che si dichiara nel manuale, il processore della radiazione solare interpreta il dato d'ingresso di radiazione come se fosse una potenza e non un'energia, in particolare espressa in $\text{kJ}/(\text{h m}^2)$, anche se tale dato viene riconosciuto essere integrale e quindi di fatto espresso in kJ/m^2 . Siccome TRNSYS utilizza come unità di misura per il tempo di simulazione l'ora, il valor medio della potenza nell'intervallo di un'ora espresso $\text{kJ}/(\text{h m}^2)$ COINCIDE NUMERICAMENTE con il valore dell'energia nell'ora espresso in kJ/m^2 . Questo consente di interpretare il dato fornito al processore solare non più un'energia, valore integrale, come dichiarato dal TYPE 15, ma come la potenza media nel timestep precedente al timestamp dei dati climatici. Così reinterpretato il dato in ingresso, la relazione di adattamento (4.67) diventa:

$$\langle G_{y,h} \rangle_{\omega_{x-1}}^{\omega_x} = R_h|_d^x \cdot \langle G_{y,h} \rangle_{\omega_{d-11}}^{\omega_d} \quad (4.70)$$

dove $\langle G_{y,h} \rangle_{\omega_{x-1}}^{\omega_x}$ rappresenta quindi la stima dell'irradianza media sul piano orizzontale per la componente $y = T$ o d (totale o diffusa) nel timestep di simulazione (il fattore risulta identico nei due casi).

Tale stima viene quindi ottenuta non attraverso un'interpolazione tra due dati ma assumendo che la relazione tra il dato da stimare e quello disponibile stiano nello stesso rapporto delle corrispettive irradianze medie extraterrestri sul piano orizzontale.

4.3.7.2 Irradiazione oraria diretta normale

Nel caso in cui il dato climatico disponibile sia invece l'irradiazione oraria diretta normale, I_{bn} , la procedura di stima dell'irradianza diretta normale nel timestep di simulazione è ancora più complessa. In questo caso, infatti, per calcolare il rapporto di adattamento, $R_{bn}|_d^x$, non si fa più riferimento alla irradiazione extraterrestre ma alla irradiazione diretta normale al suolo per cielo sereno (**descrizione assente nel manuale e estratta direttamente dal codice**), così come riportato nella seguente equazione:

$$I_{bn}|_{\omega_{x-1}}^{\omega_x} = I_{bn}|_{\omega_{d-1}}^{\omega_d} \cdot \frac{I_{c,bn}|_{\omega_{x-1}}^{\omega_x}}{I_{c,bn}|_{\omega_{d-1}}^{\omega_d}} \cdot \frac{t_d - t_{d-11}}{t_x - t_{x-1}} = I|_{\omega_{d-1}}^{\omega_d} \cdot \frac{\langle G_{c,bn} \rangle_{\omega_{x-1}}^{\omega_x}}{\langle G_{c,bn} \rangle_{\omega_{d-1}}^{\omega_d}} = I|_{\omega_{d-1}}^{\omega_d} \cdot R_{bn}|_d^x \quad (4.71)$$

dove $I_{c,bn}$ rappresenta l'irradiazione oraria al suolo del sito considerato per cielo sereno e $G_{c,bn}$ la relativa irradianza. Tale irradianza, e quindi l'irradiazione, si calcolata tramite il seguente modello di cielo sereno [39], definito da una trasmittanza dell'atmosfera a cielo sereno sia per la radiazione diretta normale [45] che per la diffusa sul piano orizzontale [46]:

$$\begin{aligned} \tau_b &\equiv \frac{G_{c,bn}}{G_{on}} = a_0(A) + a_1(A)e^{-k(A)/\cos \vartheta_z} \\ \tau_d &\equiv \frac{G_{c,dh}}{G_{o,h}} = 0,271 - 0,294 \cdot \tau_b \end{aligned} \quad (4.72)$$

con

- τ_b trasmittanza solare dell'atmosfera per radiazione diretta normale a cielo sereno;
- τ_d trasmittanza solare dell'atmosfera per radiazione diffusa sul piano orizzontale a cielo sereno;
- $a_0(A)$ coefficiente dipendente dal tipo di clima e funzione dell'altitudine del sito, funzione riportata in Tabella 2, da [39];
- $a_1(A)$ coefficiente dipendente dal tipo di clima e funzione dell'altitudine del sito, funzione riportata in Tabella 2, da [39];
- $k(A)$ fattore di estinzione dipendente dal tipo di clima e funzione dell'altitudine del sito, funzione riportata in Tabella 2, da [39];
- A altitudine del sito considerato in km.

Tabella 2. Coefficienti funzione dell'altitudine caratterizzanti il modello di cielo sereno.

Tipo di clima	a_0	a_1	k
Tropicale	$0,4025 - 0,00780 \cdot (6 - A)^2$	$0,4954 - 0,00583 \cdot (6,5 - A)^2$	$0,2765 - 0,01895 \cdot (2,5 - A)^2$
Estivo medie latitudini	$0,4110 - 0,00796 \cdot (6 - A)^2$	$0,5005 - 0,00589 \cdot (6,5 - A)^2$	$0,2765 - 0,01895 \cdot (2,5 - A)^2$
Invernale subartico	$0,4195 - 0,00813 \cdot (6 - A)^2$	$0,5005 - 0,00589 \cdot (6,5 - A)^2$	$0,2738 - 0,01877 \cdot (2,5 - A)^2$
Invernale medie latitudini	$0,4364 - 0,00846 \cdot (6 - A)^2$	$0,5106 - 0,00601 \cdot (6,5 - A)^2$	$0,2711 - 0,01858 \cdot (2,5 - A)^2$

Da tale modello si ricava che, selezionato il tipo di clima del sito in esame (in TRNSYS in realtà vengono utilizzati dei valori medi di tali coefficienti eliminando la possibilità di tale scelta) e nota la sua altitudine, l'irradianza diretta normale a ciel sereno può essere determinata in funzione dell'irradianza extraterrestre come:

$$G_{c,bn} = G_{on} \cdot \tau_b = G_{on} (a_0(A) + a_1(A)e^{-k(A)/\cos \vartheta_z}) \quad (4.73)$$

con $G_{on}(n)$ dato dall'equazione (4.6) e $\cos \vartheta_z(n; \omega)$ dato dall'equazione (4.8).

L'irradiazione solare diretta normale $I_{c,bn}$ a cielo sereno è quindi calcolabile nell'intervallo $\overline{t_{x-1} t_x}$ come:

$$I_{c,bn} \Big|_{\omega_{x-1}}^{\omega_x} = \int_{\omega_{x-1}}^{\omega_x} G_{on}(n) \cdot \tau_b(n; \omega) d\omega \quad (4.74)$$

dove n è il numero del giorno nell'anno (da 1 a 365), intero o frazionario. Il rapporto tra l'irradiazione sul timestep di simulazione e quella sul timestep dei dati è quindi dato da:

$$\frac{I_{c,bn} \Big|_{\omega_{x-1}}^{\omega_x}}{I_{c,bn} \Big|_{\omega_{d-1}}^{\omega_d}} = \frac{\int_{\omega_{x-1}}^{\omega_x} G_{on}(n) \cdot \tau_b(n; \omega) d\omega}{\int_{\omega_{d-1}}^{\omega_d} G_{on}(n) \cdot \tau_b(n; \omega) d\omega} \quad (4.75)$$

e, si considera $n = \text{costante}$ all'interno del più ampio intervallo $\overline{\omega_{d-1} \omega_d}$, questo è uguale a:

$$\frac{I_{c,bn} \Big|_{\omega_{x-1}}^{\omega_x}}{I_{c,bn} \Big|_{\omega_{d-1}}^{\omega_d}} = \frac{\int_{\omega_{x-1}}^{\omega_x} \tau_b(\omega) d\omega}{\int_{\omega_{d-1}}^{\omega_d} \tau_b(\omega) d\omega} \quad (4.76)$$

Siccome tali integrali non sono eseguibili analiticamente, vengono approssimati con delle sommatorie:

$$\frac{I_{c,bn} \Big|_{\omega_{x-1}}^{\omega_x}}{I_{c,bn} \Big|_{\omega_{d-1}}^{\omega_d}} \cong \frac{\frac{1}{2} \sum_{i=1}^{m-1} \tau_b(\omega_i) \Delta\omega + \frac{1}{2} \sum_{i=2}^m \tau_b(\omega_i) \Delta\omega}{\frac{1}{2} \sum_{i=1}^{m-1} \tau_b(\omega_{d,i}) \Delta\omega_d + \frac{1}{2} \sum_{i=2}^m \tau_b(\omega_{d,i}) \Delta\omega_d} \quad (4.77)$$

con

m numero di suddivisioni intervallo temporale ($m=60$);

$\Delta\omega = \frac{\omega_x - \omega_{x-1}}{m-1}$ valore incrementale discreto dell'angolo orario nell'intervallo di simulazione $\overline{\omega_{x-1} \omega_x}$;

$\Delta\omega_d = \frac{\omega_d - \omega_{d-1}}{m-1}$ valore incrementale discreto dell'angolo orario nell'intervallo dei dati $\overline{\omega_{d-1} \omega_d}$;

$\omega_i = \omega_1 + i \cdot \Delta\omega$ valore attuale dell'angolo orario nell'intervallo di simulazione $\overline{\omega_{x-1} \omega_x}$;

$\omega_{d,i} = \omega_{1d} + i \cdot \Delta\omega$ valore attuale dell'angolo orario nell'intervallo dei dati $\overline{\omega_{d-1} \omega_d}$.

N.B.: la media aritmetica tra le due sommatorie è per ridurre l'effetto "funzione a gradino" dovuto all'integrazione discreta: la prima assume il valore costante nell'intervallo discreto di integrazione assegnato all'estremo inferiore, la seconda all'estremo superiore. Per ottenere lo stesso risultato basta mediare gli estremi e sommare tra 2 e $m-1$, cioè:

$$\frac{1}{2} \sum_{i=1}^{m-1} \tau_b(\omega_i) \Delta\omega + \frac{1}{2} \sum_{i=2}^m \tau_b(\omega_i) \Delta\omega = \left[\frac{\tau_b(\omega_1) + \tau_b(\omega_2)}{2} + \sum_{i=2}^{m-1} \tau_b(\omega_i) \right] \Delta\omega \quad (4.78)$$

Il fattore di adattamento al timestep di simulazione del dato di radiazione disponibile, definito nell'equazione (4.71), diventa:

$$R_{bn} \Big|_d^x = \frac{\left[\frac{\tau_b(\omega_{x-1}) + \tau_b(\omega_x)}{2} + \sum_{i=2}^{m-1} \tau_b(\omega_i) \right] \Delta\omega}{\left[\frac{\tau_b(\omega_{d-1}) + \tau_b(\omega_d)}{2} + \sum_{i=2}^{m-1} \tau_b(\omega_{d,i}) \right] \Delta\omega_d} \cdot \frac{t_{d2} - t_{d1}}{t_2 - t_1} \quad (4.79)$$

Per quanto detto al paragrafo precedente, il dato passato dal processore dei dati climatici, ancorché integrale, viene interpretato come valore di irradianza media; così reinterpretato il dato in ingresso, la relazione di adattamento (4.71), diventa

$$\langle G_{bn} \rangle_{\omega_{x-1}}^{\omega_x} = R_{bn} |_d^x \cdot \langle G_{bn} \rangle_{\omega_{d-1}}^{\omega_d} \quad (4.80)$$

4.3.7.3 Calcolo delle irradianze su piano orientato

Il processore solare di TRNSYS accetta combinazioni diverse di dati climatici di radiazione con cui produce e poi espone i valori di irradianza totale per una superficie comune orientata. Le combinazioni possibili sono:

1. solo irradiazione totale oraria sul piano orizzontale $I_{tot,h} |_{\omega_{d-1}}^{\omega_d}$, assunta numericamente pari all'irradianza media nello stesso periodo, $\langle G_{tot,h} \rangle_{\omega_{d-1}}^{\omega_d}$
2. irradiazione oraria diretta e diffusa sul piano orizzontale, $I_{b,h} |_{\omega_{d-1}}^{\omega_d}$ e $I_{d,h} |_{\omega_{d-1}}^{\omega_d}$ assunte numericamente pari alle irradianze medie nello stesso periodo, $\langle I_{tot,h} \rangle_{\omega_{d-1}}^{\omega_d}$ e $\langle I_{bn,h} \rangle_{\omega_{d-1}}^{\omega_d}$;
3. irradiazione oraria totale sul piano orizzontale e diretta normale, $I_{tot,h} |_{\omega_{d-1}}^{\omega_d}$ e $I_{bn,h} |_{\omega_{d-1}}^{\omega_d}$ assunte numericamente pari alle irradianze medie nello stesso periodo, $\langle G_{tot,h} \rangle_{\omega_{d-1}}^{\omega_d}$ e $\langle G_{bn,h} \rangle_{\omega_{d-1}}^{\omega_d}$;
4. irradiazione oraria totale e diffusa sul piano orizzontale, $I_{tot,h} |_{\omega_{d-1}}^{\omega_d}$ e $I_{d,h} |_{\omega_{d-1}}^{\omega_d}$ assunte numericamente pari alle irradianze medie nello stesso periodo, $\langle G_{tot,h} \rangle_{\omega_{d-1}}^{\omega_d}$ e $\langle G_{d,h} \rangle_{\omega_{d-1}}^{\omega_d}$;
5. irradiazione oraria diretta normale e diffusa sul piano orizzontale, $I_{bn,h} |_{\omega_{d-1}}^{\omega_d}$ e $I_{d,h} |_{\omega_{d-1}}^{\omega_d}$ assunte numericamente pari alle irradianze medie nello stesso periodo, $\langle G_{bn,h} \rangle_{\omega_{d-1}}^{\omega_d}$ e $\langle G_{d,h} \rangle_{\omega_{d-1}}^{\omega_d}$;

Come si è visto nei paragrafi precedenti il fattore di adattamento del dato climatico al timestep di simulazione risulta è diverso in funzione del tipo di dato stesso:

1. $\langle G_{tot,h} \rangle_{\omega_{x-1}}^{\omega_x} = R_h |_d^x \cdot \langle G_{tot,h} \rangle_{\omega_{d-1}}^{\omega_d} \Rightarrow \langle G_{b,h} \rangle_{\omega_{x-1}}^{\omega_x}$ e $\langle G_{d,h} \rangle_{\omega_{x-1}}^{\omega_x}$ ricavate dalla correlazione di Reindl;
2. $\langle G_{b,h} \rangle_{\omega_{x-1}}^{\omega_x} = R_h |_d^x \cdot \langle G_{b,h} \rangle_{\omega_{d-1}}^{\omega_d}$ e $\langle G_{d,h} \rangle_{\omega_{x-1}}^{\omega_x} = R_h |_d^x \cdot \langle G_{d,h} \rangle_{\omega_{d-1}}^{\omega_d}$
 $\Rightarrow \langle G_{tot,h} \rangle_{\omega_{x-1}}^{\omega_x} = \langle G_{b,h} \rangle_{\omega_{x-1}}^{\omega_x} + \langle G_{d,h} \rangle_{\omega_{x-1}}^{\omega_x}$
3. $\langle G_{tot,h} \rangle_{\omega_{x-1}}^{\omega_x} = R_h |_d^x \cdot \langle G_{tot,h} \rangle_{\omega_{d-1}}^{\omega_d}$ e $\langle G_{bn} \rangle_{\omega_{x-1}}^{\omega_x} = R_{bn} |_d^x \cdot \langle G_{bn} \rangle_{\omega_{d-1}}^{\omega_d}$
 $\Rightarrow \langle G_{b,h} \rangle_{\omega_{d-1}}^{\omega_d} = \langle G_{bn} \rangle_{\omega_{d-1}}^{\omega_d} \cos \vartheta_z (\omega_{av}) \Rightarrow \langle G_{d,h} \rangle_{\omega_{x-1}}^{\omega_x} = \langle G_{tot,h} \rangle_{\omega_{x-1}}^{\omega_x} - \langle G_{b,h} \rangle_{\omega_{x-1}}^{\omega_x}$;
4. $\langle G_{tot,h} \rangle_{\omega_{x-1}}^{\omega_x} = R_h |_d^x \cdot \langle G_{tot,h} \rangle_{\omega_{d-1}}^{\omega_d}$ e $\langle G_{d,h} \rangle_{\omega_{x-1}}^{\omega_x} = R_h |_d^x \cdot \langle G_{d,h} \rangle_{\omega_{d-1}}^{\omega_d}$
 $\Rightarrow \langle G_{b,h} \rangle_{\omega_{x-1}}^{\omega_x} = \langle G_{tot,h} \rangle_{\omega_{x-1}}^{\omega_x} - \langle G_{d,h} \rangle_{\omega_{x-1}}^{\omega_x}$;
5. $\langle G_{bn} \rangle_{\omega_{x-1}}^{\omega_x} = R_{bn} |_d^x \cdot \langle G_{bn} \rangle_{\omega_{d-1}}^{\omega_d}$ e $\langle G_{d,h} \rangle_{\omega_{x-1}}^{\omega_x} = R_h |_d^x \cdot \langle G_{d,h} \rangle_{\omega_{d-1}}^{\omega_d}$
 $\Rightarrow \langle G_{b,h} \rangle_{\omega_{d-1}}^{\omega_d} = \langle G_{bn} \rangle_{\omega_{d-1}}^{\omega_d} \cos \vartheta_z (\omega_{av}) \Rightarrow \langle G_{tot,h} \rangle_{\omega_{x-1}}^{\omega_x} = \langle G_{d,h} \rangle_{\omega_{x-1}}^{\omega_x} + \langle G_{b,h} \rangle_{\omega_{x-1}}^{\omega_x}$;

dove, il coseno dell'angolo zenitale ϑ_z è calcolato utilizzando il valore medio aritmetico dell'angolo orario nell'intervallo del timestep di simulazione, $\overline{t_{x-1} t_x}$,

$$\omega_{av} = \frac{\omega_x + \omega_{x-1}}{2} \quad (4.81)$$

Lo stesso angolo medio viene poi utilizzato per calcolare il coseno dell'angolo d'incidenza ϑ , tramite l'equazione (4.12).

In conclusione, l'algoritmo di TRNSYS funziona nel seguente modo:

- Il processore dei dati climatici, rigidamente orari, non interpola i dati di radiazione che fornisce al processore solare;
- il processore solare, ogni timestamp di simulazione, riceve uno o due dati di radiazione oraria che interpreta quali dati di irradianza media sul timestep dei dati climatici (l'ora);
- se il timestep di simulazione coincide con il timestep dei dati climatici ($\Delta t = \Delta t_d$), l'ora, il processore solare considera il dato di radiazione una irradianza media attribuita al baricentro dell'intervallo di simulazione antecedente al timestamp attuale; ad esempio per $t = 10$, il dato letto dal file climatico è $I_{tot,h}|_9^{10} \rightarrow$ il valore assunto dal processore solare è $\langle G_{tot,h} \rangle_9^{10} \rightarrow$ tale valore viene attribuito a $\omega_{xxx} = [\omega(9) + \omega(10)]/2$;
- se il timestep di simulazione è inferiore al timestep dei dati climatici ($\Delta t < \Delta t_d$), ma comunque **sempre un suo sottomultiplo**, si hanno due condizioni diverse:
 - quando $t \equiv t_d$, l'irradiazione calcolata tramite il fattore di adattamento, $R_h|_d^x$ o $R_{bn}|_d^x$, è stimata all'indietro, cioè utilizzando l'irradiazione oraria fornita come dato a tale timestamp e quindi relativa al timestep dei dati precedente; ad esempio $t = 10$ e $\Delta t = 0,25 h$, il dato letto dal file climatico è $I_{tot,h}|_9^{10} \rightarrow$ il valore assunto dal processore solare è $\langle G_{tot,h} \rangle_9^{10} \rightarrow$ si calcola il valore adattato al timestep di simulazione come $\langle G_{tot,h} \rangle_{\omega_{9,75}}^{10} = R_h|_{10}^{10} \cdot \langle G_{tot,h} \rangle_9^{10} \rightarrow$ tale valore viene attribuito a $\omega_{av} = [\omega(9,75) + \omega(10)]/2$;
 - quando $t > t_d$, fino a $t \equiv t_d + \Delta t_d$ escluso, l'irradiazione calcolata è stimata in avanti; cioè utilizzando l'irradiazione oraria fornita come dato al timestamp dei dati che si trova nel "futuro", relativa al timestep dei dati successivo; ad esempio $t = 10,25$ e $\Delta t = 0,25 h$, il dato letto dal file climatico è $I_{tot,h}|_{10}^{11} \rightarrow$ il valore assunto dal processore solare è $\langle G_{tot,h} \rangle_{10}^{11} \rightarrow$ si calcola il valore adattato al timestep di simulazione come $\langle G_{tot,h} \rangle_{\omega_{10}^{10,25}}^{10,25} = R_h|_{11}^{10,25} \cdot \langle G_{tot,h} \rangle_{10}^{11} \rightarrow$ tale valore viene attribuito a $\omega_{av} = [\omega(10) + \omega(10,25)]/2$.

4.3.8 Calcolo dell'irradianza diffusa dalla volta celeste

La radiazione solare che attraversa l'atmosfera viene da questa in parte dispersa generando quella che si chiama "radiazione diffusa", che viene normalmente misurata al suolo sul piano orizzontale. Quando occorre calcolare, a partire da tale dato, l'irradianza diffusa proveniente dal cielo su un piano inclinato rispetto all'orizzonte si possono prendere in considerazione modelli diversi:

- la radiazione diffusa è isotropa (arriva allo stesso modo da ogni punto del cielo);
- la radiazione diffusa è anisotropa (diverso valore dell'intensità in funzione della posizione considerata sulla volta del cielo, circumsolare, etc.).

Nel primo caso, isotropia, il dato, la radiazione diffusa sul piano orizzontale, viene "proiettata" sulla superficie comunque orientata utilizzando il fattore di forma per radiazione isotropa ripotato in equazione (4.17).

Nel secondo caso, viene utilizzato il modello di cielo anisotropo di Perez [47, 48].

4.3.8.1 Modello di cielo anisotropo di Perez

Nel modello proposto da Perez [47 48], la componente dell'irradianza diffusa che incide sulla superficie comunque orientata (β, γ), proveniente dalla volta celeste è definita dalla seguente relazione:

$$G_{d,sky \rightarrow (\beta, \gamma)} = G_{d,h} \cdot \mathcal{F}_{s,sky} = G_{d,h} \cdot \left[(1 - F_1) \cdot \frac{1 + \cos \beta}{2} + F_1 \cdot \frac{a}{b} + F_2 \cdot \sin \beta \right] \quad (4.82)$$

dove F_1 e F_2 sono, rispettivamente, i coefficienti di irradianza circumsolare e dell'orizzonte e a e b sono coefficienti che tengono conto dell'angolo di incidenza solare ϑ sulla superficie considerata. Introducendo l'angolo zenitale del sole ϑ_z espresso in radianti si ha:

$$\begin{aligned} a &= \max[0; \cos \vartheta] \\ b &= \max[0,0872; \cos \vartheta_z] \end{aligned} \quad (4.83)$$

dove 0 sta per il coseno di 90° e 0,0872 è il coseno di 85° .

I coefficienti F_1 e F_2 dipendono dalle condizioni del cielo descritti dai coefficienti di trasparenza ε e di luminosità Δ definiti dalle seguenti equazioni:

$$\begin{aligned} \varepsilon &= \frac{\frac{G_{d,h} + G_{b,h} \sin \alpha_s}{G_{d,h}} + 1,041 \vartheta_z^3}{1 + 1,041 \vartheta_z^3} \\ b &= m \cdot G_{d,h} / G_0 \end{aligned} \quad (4.84)$$

con m , massa ottica dell'aria, funzione soltanto dell'altezza solare α_s , espressa in gradi

$$m = \frac{1}{\sin \alpha_s + 0,50572 \cdot (6,07995 + \alpha_s)^{-1,6364}} \quad (4.85)$$

I coefficienti F_1 e F_2 si calcolano quindi come:

$$\begin{aligned} F_1 &= \max[0; f_{11} + f_{12} \cdot \Delta + f_{13} \cdot \vartheta_z] & 0 \leq F_1 \leq 1 \\ F_2 &= \max[0; f_{21} + f_{22} \cdot \Delta + f_{23} \cdot \vartheta_z] & 0 \leq F_2 \leq 1 \end{aligned} \quad (4.86)$$

con ϑ_z espresso in radianti e dove i coefficienti f_{ij} si desumono dalla tabella in funzione del valore del coefficiente di trasparenza ε .

Tabella 3. Coefficienti per il calcolo dei contributi di irradianza circumsolare e dell'orizzonte.

ε	1,000-1,065	1,065-1,230	1,230-1,500	1,500-1,950	1,950-2,800	2,800-4,500	4,500-6,200	> 6,200
f_{11}	-0,0083117	0,1299457	0,3296958	0,5682053	0,8730280	1,1326077	1,0601591	0,6777470
f_{12}	0,5877285	0,6825954	0,4868735	0,1874525	-0,3920403	-1,2367284	-1,5999137	-0,3272588
f_{13}	-0,0620636	-0,1513752	-0,2210958	-0,2951290	-0,3616149	-0,4118494	-0,3589221	-0,2504286
f_{21}	-0,0596012	-0,0189325	0,0554140	0,1088631	0,2255647	0,2877813	0,2642124	0,1561313
f_{22}	0,0721249	0,0659650	-0,0639588	-0,1519229	-0,4620442	-0,8230357	-1,1272340	-1,3765031
f_{23}	-0,0220216	-0,0288748	-0,0260542	-0,0139754	0,0012448	0,0558651	0,1310694	0,2506212

4.3.9 Implementazione in OpenBPS

A seguito delle analisi fatte, si è deciso di implementare, anche a poter sperimentare l'impatto delle diverse metodiche sul risultato finale della simulazione delle prestazioni energetiche di un edificio, tutti e tre i metodi

per il calcolo della posizione del sole, così come l'interpolazione lineare o l'adattamento integrale sulla radiazione solare, non solo con il metodo descritto di TRNSYS, ma anche con una sua evoluzione originale riportata nel seguito (LinearInterpolationAvgAtmosphericTransmittance), come si può vedere dalla scelta disponibile nella maschera di gestione dei dati climatici.

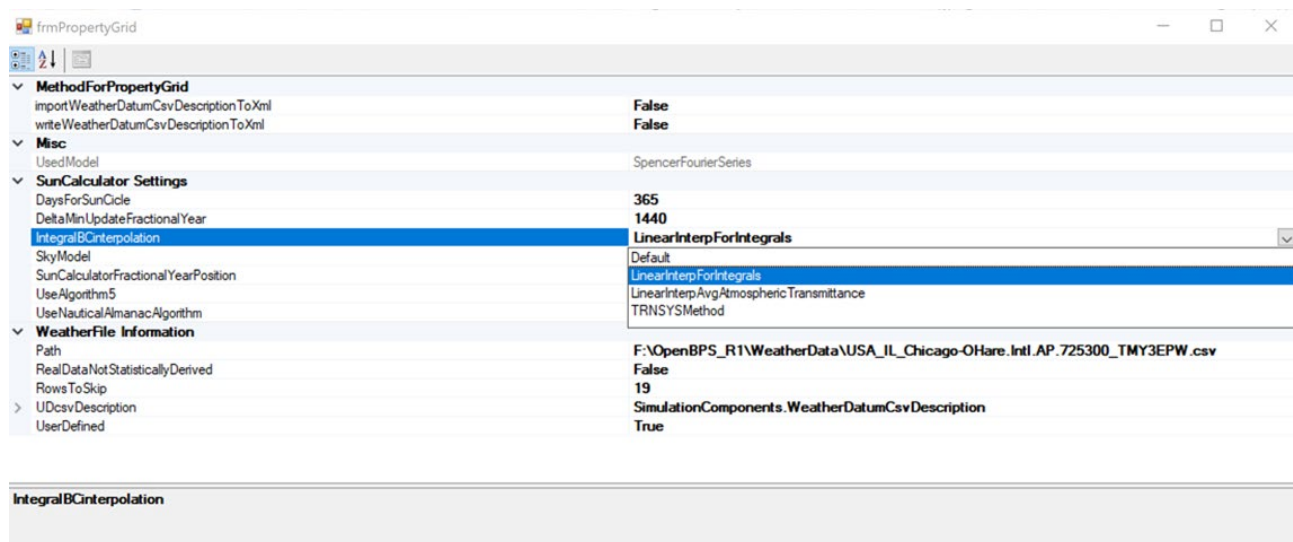


Figura 26 – Opzioni di scelta sul metodo da utilizzare per il trattamento della radiazione solare.

Inoltre il codice implementa il modello di Perez di cielo anisotropo [47, 48] per il calcolo della radiazione diffusa proveniente dal cielo. L'utente, come si vede in Figura 27, può scegliere se utilizzare, per il calcolo della radiazione solare incidente su una superficie comunque orientata, la radiazione diffusa isotropa o anisotropa con tale modello.

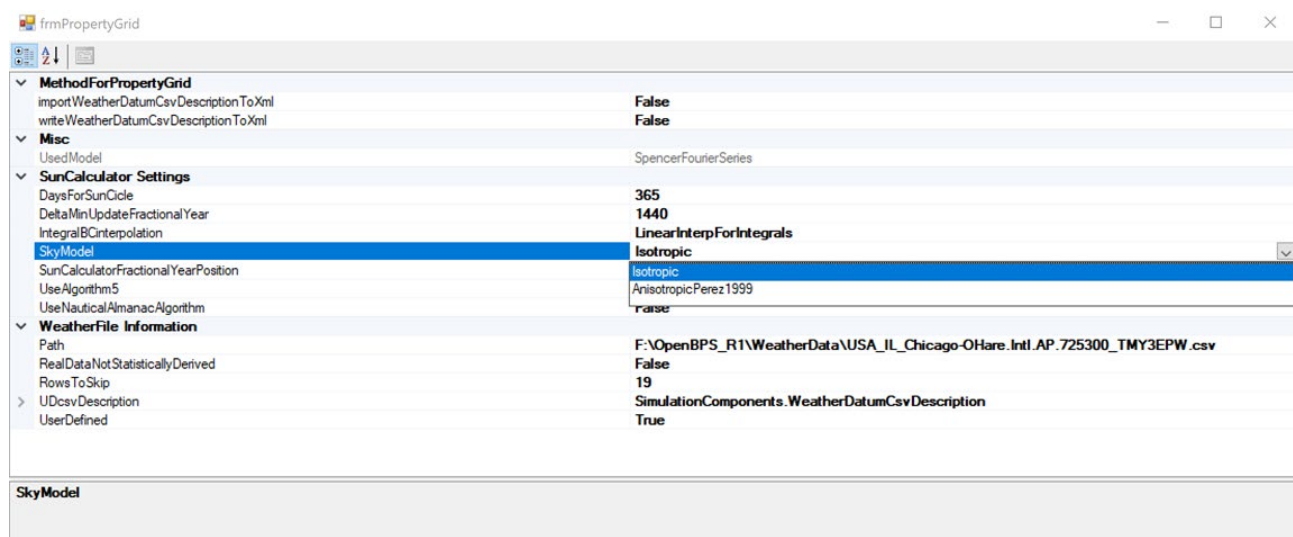


Figura 27 – Opzioni di scelta sul metodo da utilizzare per radiazione solare diffusa che arriva su una superficie comunque orientata.

4.3.9.1 Metodo della trasmittanza atmosferica media

Il metodo implementato in TRNSYS per determinare l'irradianza solare al tempo della simulazione, time stamp t , adattando il dato disponibile a un opportuno tempo dei dati, timestamp t_d , come visto è applicato

in modo diverso se il dato disponibile è su componenti orizzontali della radiazione o se è invece la “irradianza media” diretta normale. Solo in questo secondo caso di fa riferimento alla trasmittanza dell’atmosfera a cielo sereno per determinare, tramite questa, il fattore di adattamento. Il metodo sviluppato e implementato in OpenBPS, come alternativa ai metodi già illustrati, estende l’approccio della trasmittanza dell’atmosfera come criterio per ottenere una migliore “interpolazione” dei dati per passi di simulazione inferiori al timestep dei dati.

Partendo dal modello di cielo sereno riportato in [39] e definito dalla copia di relazioni (4.72), di seguito riproposte

$$\begin{aligned}\tau_{bc}(t) &\equiv \frac{G_{c,bn}(t)}{G_{on}(t)} = a_0(A) + a_1(A)e^{-k(A)/\cos \vartheta_z(t)} \\ \tau_{dc}(t) &\equiv \frac{G_{c,dh}(t)}{G_{o,h}(t)} = \frac{G_{c,dh}(t)}{G_{on}(t) \cos \vartheta_z(t)} = 0,271 - 0,294 \cdot \tau_b(t)\end{aligned}\quad (4.87)$$

è possibile definire delle trasmittanze a cielo reale del sito partendo dai dati di irradiazione oraria disponibili nel file climatico. Indicati con:

- $I_{bn}|_{\omega_x}^{\omega_{x+1}}$ l’irradiazione diretta normale al suolo tra gli angoli orari ω_x e ω_{x+1} , [J/m²];
- $I_{T,h}|_{\omega_x}^{\omega_{x+1}}$ l’irradiazione totale sul piano orizzontale tra gli angoli orari ω_x e ω_{x+1} , [J/m²];
- $I_{d,h}|_{\omega_x}^{\omega_{x+1}}$ l’irradiazione diffusa sul piano orizzontale tra gli angoli orari ω_x e ω_{x+1} , [J/m²];
- $I_{b,h}|_{\omega_x}^{\omega_{x+1}}$ l’irradiazione diretta sul piano orizzontale tra gli angoli orari ω_x e ω_{x+1} , [J/m²];

è possibile determinare i seguenti valori di trasmittanza media oraria:

$$\bar{\tau}_{bn}(\omega_{x+1}) \equiv \frac{I_{bn}|_{\omega_x}^{\omega_{x+1}}}{I_{0n}|_{\omega_x}^{\omega_{x+1}}} \quad (4.88)$$

$$\bar{\tau}_{T,h}(\omega_{x+1}) \equiv \frac{I_{T,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} \quad (4.89)$$

$$\bar{\tau}_{d,h}(\omega_{x+1}) \equiv \frac{I_{d,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} \quad (4.90)$$

$$\bar{\tau}_{b,h}(\omega_{x+1}) \equiv \frac{I_{b,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} \quad (4.91)$$

dove

- $I_{0n}|_{\omega_x}^{\omega_{x+1}}$ l’irradiazione diretta normale extraterrestre tra gli angoli orari ω_x e ω_{x+1} , [J/m²];
- $I_{0,h}|_{\omega_x}^{\omega_{x+1}}$ l’irradiazione diretta extraterrestre su un piano orizzontale parallelo al piano dell’orizzonte del sito agli angoli orari ω_x e ω_{x+1} , [J/m²].;

con

$$I_{on}|_{\omega_x}^{\omega_{x+1}} = \int_{\omega_x}^{\omega_{x+1}} G_{on} d\omega \quad (4.92)$$

$$I_{o,h}|_{\omega_x}^{\omega_{x+1}} = \int_{\omega_x}^{\omega_{x+1}} G_{on} \cos \vartheta_z d\omega \quad (4.93)$$

L'assunzione alla base dell'algoritmo di interpolazione proposto è che sia, in funzione del dato disponibile,:

$$\frac{G_{bn}(\omega_y)}{G_{on}(\omega_y)} \cong \frac{I_{bn}|_{\omega_x}^{\omega_{x+1}}}{I_{on}|_{\omega_x}^{\omega_{x+1}}} = \bar{t}_{bn}(\omega_{x+1}) \quad \omega_x < \omega_y \leq \omega_{x+1} \quad (4.94)$$

$$\frac{G_{T,h}(\omega_y)}{G_{0,h}(\omega_y)} \cong \frac{I_{T,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} = \bar{t}_{T,h}(\omega_{x+1}) \quad \omega_x < \omega_y \leq \omega_{x+1} \quad (4.95)$$

$$\frac{G_{d,h}(\omega_y)}{G_{0,h}(\omega_y)} \cong \frac{I_{d,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} = \bar{t}_{d,h}(\omega_{x+1}) \quad \omega_x < \omega_y \leq \omega_{x+1} \quad (4.96)$$

$$\frac{G_{b,h}(\omega_y)}{G_{0,h}(\omega_y)} \cong \frac{I_{d,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} = \bar{t}_{b,h}(\omega_{x+1}) \quad \omega_x < \omega_y \leq \omega_{x+1} \quad (4.97)$$

Esistono diverse alternative nell'applicazione dell'algoritmo, che riguardano il calcolo delle trasmittanze, funzione del modo di determinare le (4.92) e (4.93):

- a) $G_{on}(n) = \text{cost}$ durante l'arco della giornata (n indice del giorno intero);
- b) $G_{on}(n) = f(\omega)$ variabile durante la giornata (n indice del giorno reale);

tali alternative comportano variazioni minime (minori dell'1%) per cui sono sostanzialmente equivalenti.

Quindi per passi di simulazione inferiori ai passi temporali dei dati climatici, l'irradianza si calcola, in funzione dei dati disponibili, come:

$$G_{bn}(\omega_y) = \frac{I_{bn}|_{\omega_x}^{\omega_{x+1}}}{I_{on}|_{\omega_x}^{\omega_{x+1}}} G_{on}(\omega_y) \quad \omega_x < \omega_y \leq \omega_{x+1} \quad (4.98)$$

$$G_{T,h}(\omega_y) = \frac{I_{T,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} G_{0,h}(\omega_y) \quad \omega_x < \omega_y \leq \omega_{x+1} \quad (4.99)$$

$$G_{d,h}(\omega_y) = \frac{I_{d,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} G_{0,h}(\omega_y) \quad \omega_x < \omega_y \leq \omega_{x+1} \quad (4.100)$$

$$G_{b,h}(\omega_y) = \frac{I_{d,h}|_{\omega_x}^{\omega_{x+1}}}{I_{0,h}|_{\omega_x}^{\omega_{x+1}}} G_{0,h}(\omega_y) \quad \omega_x < \omega_y \leq \omega_{x+1} \quad (4.101)$$

4.4 Scambio radiativo ad onde lunghe

Lo scambio termico radiativo a onde lunghe (LW) avviene sia tra le superfici dell'ambiente esterno e l'edificio e all'interno di ogni zona termica tra le superfici che la delimitano.

Nel caso delle interazioni radiative tra superfici interne il calcolo è più dettagliato che all'esterno, dato si conosce sia la geometria che le proprietà fisiche dei componenti con un buon grado di approssimazione.

Per questo motivo, e non perché i modelli fisici studiati siano diversi, la descrizione dei modelli computazionali implementati per la simulazione di queste due applicazioni dello scambio termico radiativo LW, vengono trattate separatamente nei due paragrafi seguenti.

4.4.1 Scambio termico radiativo interno

Lo scambio di radiazioni ad onde lunghe (LW) tra le superfici interne delle partizioni di ciascuna zona viene calcolato mediante la seguente equazione:

$$\Phi_{int,i}^{LW} = A_i \cdot R_{ij}^{LW} \cdot \sigma \cdot (T_i^4 - T_j^4) \quad (4.102)$$

dove R_{ij}^{LW} è il fattore di radiazione mutua di corpo grigio tra la superficie i e la superficie j .

Questo fattore viene calcolato sulla base del fattore di vista tra la superficie i e la superficie j e delle emissività e riflettività alla radiazione ad onde lunghe di tutte le superfici della zona indagata.

Il componente `LWRadiativeModule`, creato per ogni zona, ha la responsabilità di gestire questo scambio di radiazione ad onde lunghe tra le superfici interne delle partizioni di ciascuna zona.

Questo componente è chiamato a:

1. precalcolare i fattori di radiazione mutua di corpo grigio per ciascuna coppia di superfici, durante la fase di inizializzazione,
2. assegnare a ciascuna superficie eventuali flussi radiativi ad onde lunghe imposti come definiti dall'utente, ad ogni timestamp,
3. assegnare a ciascuna superficie, quale condizione al contorno, la sua temperatura media radiante e il relativo coefficiente di scambio termico superficiale radiativo ad ogni iterazione.

Il primo passo consiste nel calcolare il fattore di radiazione mutua. Per fare ciò è necessario:

1. calcolare i fattori di forma per corpo nero tra tutte le coppie di superfici presenti nella zona;
2. capire se ci sono ostacoli tra una superficie e le altre e gestire tale ostruzione (non ancora implementato);
3. calcolare le inter-riflessioni, seguendo l'approssimazione di superficie diffondente.

I seguenti paragrafi hanno lo scopo di spiegare come questi passaggi sono stati implementati all'interno dello strumento di calcolo.

4.4.1.1 Calcolo del fattore di forma di corpo nero

Una generica zona può contenere quante partizioni "principali" (pareti, tetti, pavimenti) necessita con tutti i componenti secondari necessari (finestre, porte) (vedasi Figura 28).

Tuttavia, attualmente, siccome non è stato ancora implementato un solutore che tratti superfici di forma qualsiasi e un componente per il rilevamento di ostacoli, le SUPERFICI POLIGONALI GENERICHE sono approssimate con una discretizzazione in elementi rettangolari (si hanno errori di approssimazione leggermente più elevati) e, quindi, sono consentite SOLO SUPERFICI che sono:

1. planari, e

2. non si “oscurano” a vicenda.

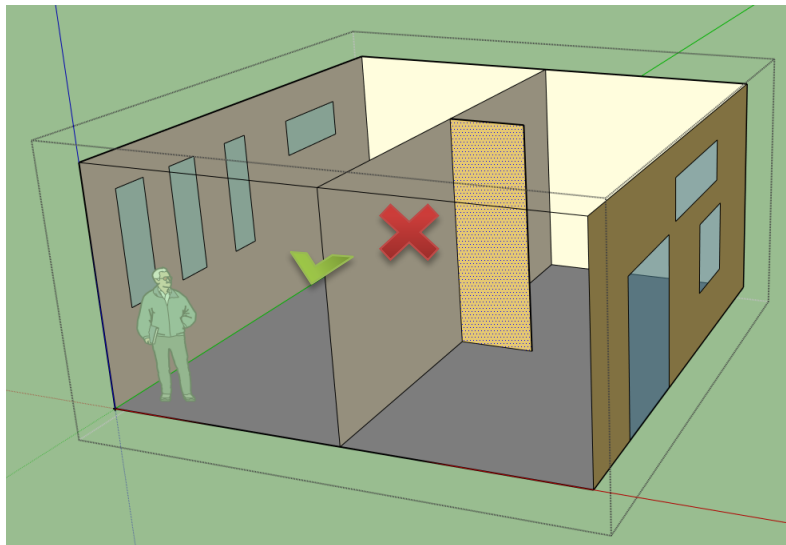


Figura 28 – Le superfici che occupano solo parzialmente lo spazio non sono attualmente gestite.

Se abbiamo una parete interna che separa una zona termica (insieme di ambienti che hanno stessi set point, stessi carichi termici interni, stessi terminali d’impianto, ecc.) in due ambienti (con due volumi d'aria), per il corretto trattamento degli scambi radiativi si devono creare due zone distinte. Il comportamento di ciascuno dei due volumi d'aria contenuto nei due ambienti distinti è ancora quello della zona “completa”, con gli stessi set point e guadagni, ma gli scambi radiativi terranno correttamente conto della partizione che separa i due ambienti. In questo caso, il pavimento, il soffitto e le pareti laterali devono essere divisi di conseguenza tra i due ambienti.

Non è stato qui introdotto il concetto di "spazio o ambiente" come elemento di una "zona", per evitare la moltiplicazione delle classi quando non strettamente necessario. Sarà compito della GUI facilitare la creazione di elenchi di zone che condividono gli stessi guadagni interni, strategie di controllo, infiltrazioni, tipi di elementi terminali HVAC, ecc.

4.4.1.1.1 Calcolo della matrice dei fattori di forma

Una zona può avere un numero generico di partizioni e ogni partizione può avere un numero generico di sottosuperfici.

Per calcolare il flusso di termico radiativo ad onde lunghe scambiato tra la parte “netta” (ovvero parte omogenea o omogenea equivalente) di ciascuna partizione e ciascuna sottosuperficie, è stata implementata la seguente procedura.

Si abbia, ad esempio, una partizione p_1 e una partizione p_2 . Useremo il simbolo p_i come riferimento alla parte “netta” o omogenea della partizione, definendo tale l’area della partizione stessa al netto delle sottosuperfici (finestre, porte, altro, ecc.) contenute nella partizione, indicate con $(w_{k,j})$.

Si calcolano quindi i fattori di forma tra diversi elementi, in particolare, tra:

- la parte netta/omogenea di ciascuna partizione (p_i) e la parte netta/omogenea (p_k) delle altre partizioni comprese nella zona;
- a parte netta/omogenea di ciascuna partizione (p_i) e ciascuna sottosuperficie compresa nella zona ($w_{k,j}$),
- ogni apertura ($w_{i,x}$) compresa in una zona e tutte le altre aperture ($w_{k,y}$).

Con riferimento a Figura 29, ad esempio, la partizione p_1 ha $n = 4$ sottosuperfici e la partizione p_2 ha $m = 3$ sottosuperfici. In tal caso, si determinano i fattori di forma tra p_1 e p_2 , tra p_1 e $w_{2,1}$, $w_{2,2}$ e $w_{2,3}$ e quindi tra $w_{1,1}$ e $w_{2,1}$, $w_{2,2}$ e $w_{2,3}$, tra $w_{1,2}$ e $w_{2,1}$, $w_{2,2}$ e $w_{2,3}$, ecc.

Per calcolare questi fattori di forma per corpo nero occorre eseguire i seguenti passaggi (rappresentati anche in Figura 29):

1. calcolare il fattore di forma tra le superfici ignorando l'esistenza delle loro sottosuperfici, cioè tra $p_i + \sum_{j=1}^n w_{ij}$ e $p_k + \sum_{j=1}^m w_{kj} \rightarrow F_{p_i + \sum_{j=1}^n w_{ij} \rightarrow p_k + \sum_{j=1}^m w_{kj}}$
2. calcolare il fattore di forma tra ciascuna sottosuperficie di una superficie e le altre superfici ignorando l'esistenza delle loro sottosuperfici (a lordo), cioè tra w_{ix} e $p_k + \sum_{j=1}^m w_{kj}$ (fattore $F_{w_{ix} \rightarrow p_k + \sum_{j=1}^m w_{kj}}$), rappresentato con la fase 2' in Figura 29, e tra ogni sottosuperficie e le altre sottosuperfici (fattore $F_{w_{ix} \rightarrow w_{kj}}$), rappresentato con la fase 2'' in Figura 29;
3. calcolare il fattore di forma tra ciascuna superficie lorda (ignorando l'esistenza delle sue sottosuperfici) e la parte netta delle altre superfici, cioè

$$F_{p_i + \sum_{j=1}^n w_{ij} \rightarrow p_k} = F_{p_i + \sum_{j=1}^n w_{ij} \rightarrow p_k + \sum_{j=1}^m w_{kj}} - \sum_{j=1}^m F_{p_i + \sum_{j=1}^n w_{ij} \rightarrow w_{kj}} =$$

$$= F_{p_i + \sum_{j=1}^n w_{ij} \rightarrow p_k + \sum_{j=1}^m w_{kj}} - \sum_{j=1}^m \frac{A_{w_{kj}}}{A_{p_i + \sum_{j=1}^n w_{ij}}} F_{w_{kj} \rightarrow p_i + \sum_{j=1}^n w_{ij}}$$

4. calcolare il fattore di forma tra ogni sottosuperficie e la parte netta delle altre superfici, cioè

$$F_{w_{kx} \rightarrow p_i} = F_{w_{kx} \rightarrow p_i + \sum_{j=1}^n w_{ij}} - \sum_{j=1}^n F_{w_{kx} \rightarrow w_{ij}}$$

5. calcolare il fattore di forma tra la parte netta di ciascuna superficie e la parte netta delle altre superfici

$$F_{p_i \rightarrow p_k} = F_{p_i \rightarrow p_k + \sum_{j=1}^m w_{kj}} - \sum_{j=1}^m F_{p_i \rightarrow w_{kj}}$$

$$= \frac{A_{p_k + \sum_{j=1}^m w_{kj}}}{A_{p_i}} F_{p_k + \sum_{j=1}^m w_{kj} \rightarrow p_i} - \sum_{j=1}^m \frac{A_{w_{kj}}}{A_{p_i}} F_{w_{kj} \rightarrow p_i}$$

Si determina in tal modo la matrice ridotta evidenziata dal rettangolo rosso in Figura 29, che viene poi immagazzinata nella proprietà chiamata `BBViewFactors` nel `LWRadiativeModule` di ogni zona.

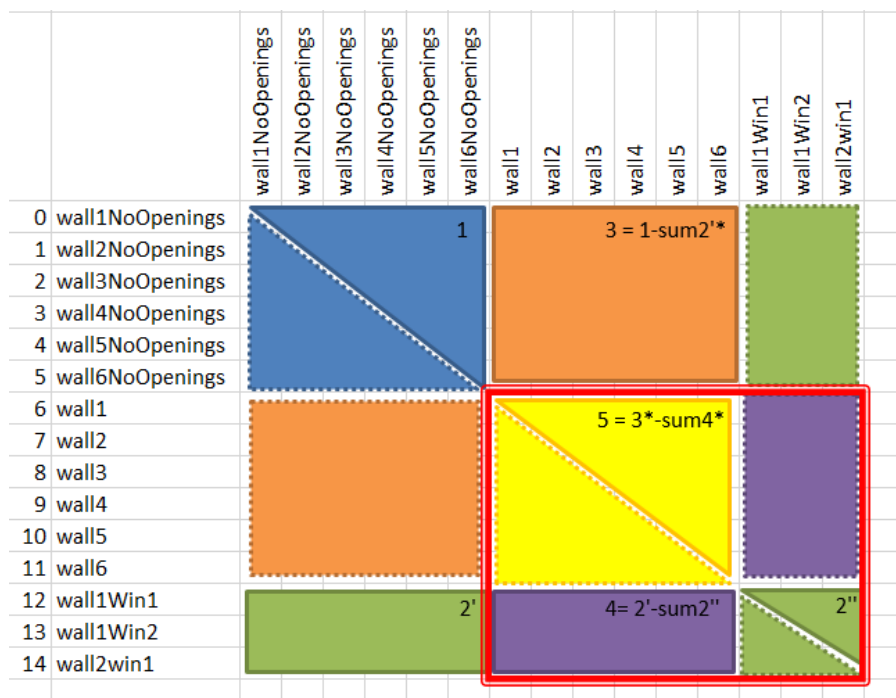


Figura 29 – Calcolo dei fattori di forma di corpo nero.

4.4.1.1.2 Matrice dei fattori di forma: distribuzione dell'errore

La matrice dei fattori di forma tra ciascuna superficie e sottosuperficie può essere affetta da alcuni errori di approssimazione, dovuti al partizionamento delle superfici. L'utente può decidere se desidera distribuire questo errore tra tutti i fattori di forma o meno.

Se l'utente decide di distribuire l'errore, per assicurare la "chiusura" dei fattori di forma (per ciascuna superficie la somma dei fattori deve essere uguale ad uno), preservando la reciprocità tra i vari fattori di forma, è stato implementato il seguente algoritmo, adattato da [49]:

- si calcola la matrice simmetrica **AF**, i cui elementi sono $(AF)_{ij} = A_i \cdot F_{ij}$
- viene calcolato l'errore sulla somma degli elementi di ciascuna riga e sulla somma degli elementi di ciascuna colonna, come: $e_i = [\sum_j (AF)_{ij} - A_i]$ e $e_j = [\sum_i (AF)_{ij} - A_j]$;
- se l'errore in una di queste somme supera l'errore massimo consentito (definito dall'utente), viene eseguita la distribuzione dell'errore su tutti gli elementi della matrice, altrimenti non viene eseguita alcuna correzione.

La correzione viene effettuata nel seguente modo:

- viene costruita la matrice trasposta di **AF**, **AF^T**, e viene controllato l'errore sulla simmetria di **AF**: **E=AF-AF^T**;
- si determinano i fattori di correzione per ciascuna riga r_i e colonna c_j della matrice trasposta **AF^T**, calcolati come il rapporto tra la somma prevista sulla riga/colonna e la somma ottenuta degli elementi della matrice **AF^T**;
- la matrice **AF^T** viene corretta iterativamente, moltiplicando ciascuno dei suoi elementi $(AF)_{ij}$, per la media geometrica dei fattori di correzione della corrispondente riga r_i e colonna c_j ;
- il processo iterativo termina quando viene raggiunto il numero massimo di iterazioni consentito o quando l'errore per ogni somma diventa inferiore all'errore massimo consentito.

La verifica dell'algoritmo implementato è stata eseguita, con riferimento complessa geometria della zona termica di Figura 30 e Figura 31, per confronto con i risultati ottenuti dal software commerciale TRNSYS 17.

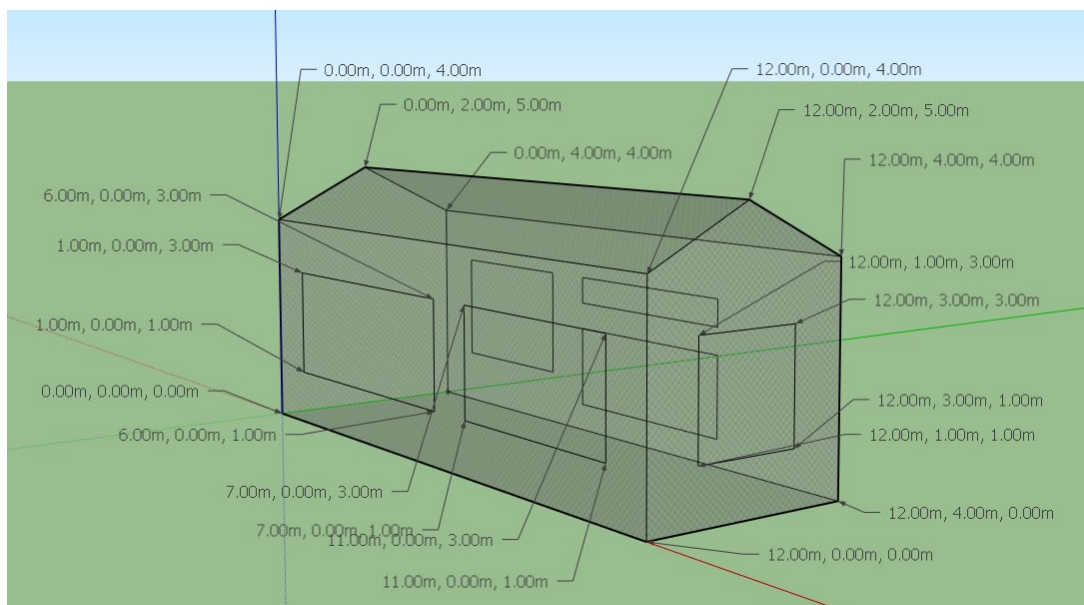


Figura 30 – Caso test Multi2 – Vista sud-est.

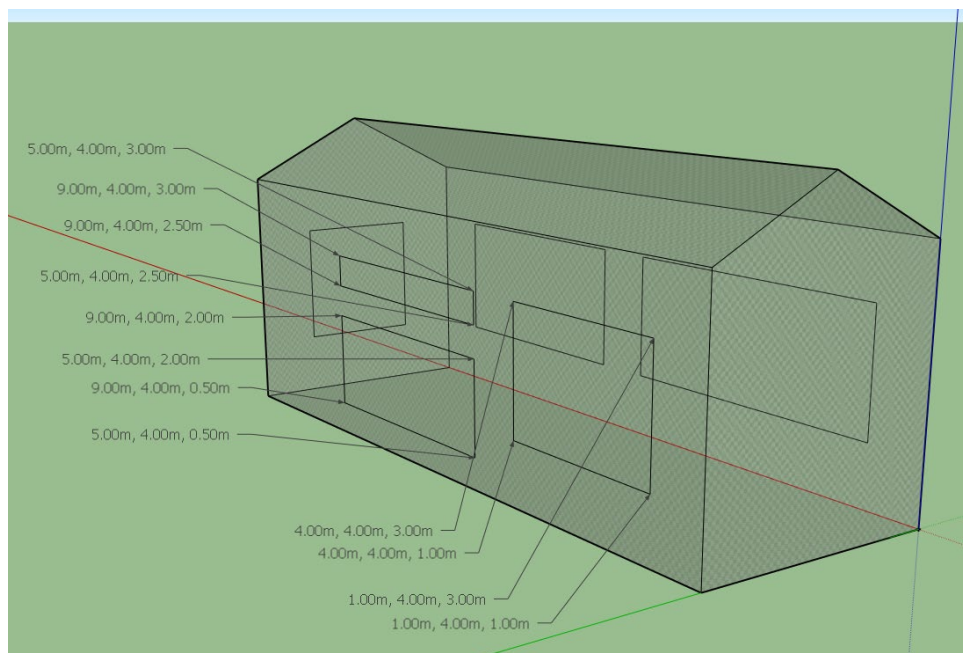


Figura 31 – Caso test Multi2 – Vista nord-ovest.

4.4.1.1.3 Calcolo del fattore di forma tra poligoni

Per la determinazione del fattore di forma tra due superfici piane delimitate, due poligoni, si possono impiegare due diversi metodi:

1. un calcolo analitico “esatto”;
2. un calcolo numerico approssimato.

Se le due superfici sono rettangolari, con almeno due lati paralleli tra loro, vengono utilizzati i metodi analitici, per tutti gli altri casi viene scelto il metodo numerico.

Nelle figure seguenti sono rappresentate alcune delle geometrie usate per verificare la bontà dell'algoritmo implementato, tramite confronto del risultato ottenuto da OpenBPS con quelli ottenuti dalle soluzioni disponibili nelle diverse sottosezioni del sito Web: <http://www.thermalradiation.net/tablecon.html#C>.

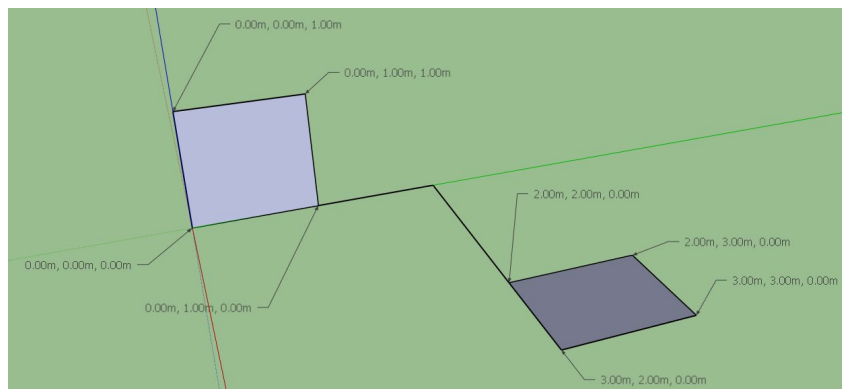


Figura 32 – Test Perp1.

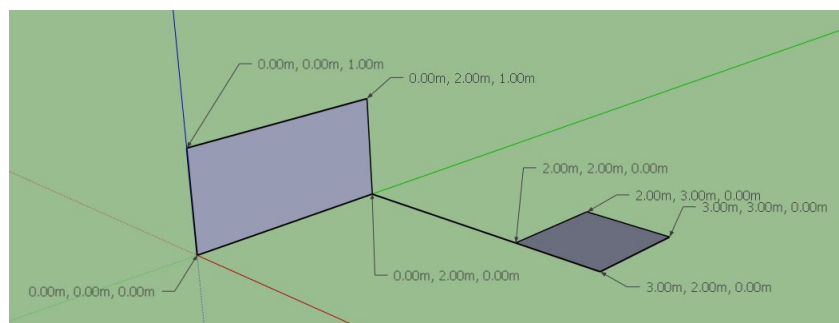


Figura 33 – Test Perp2.

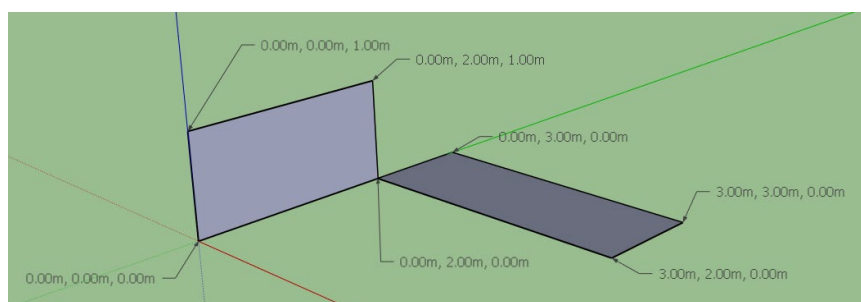


Figura 34 – Test Perp3.

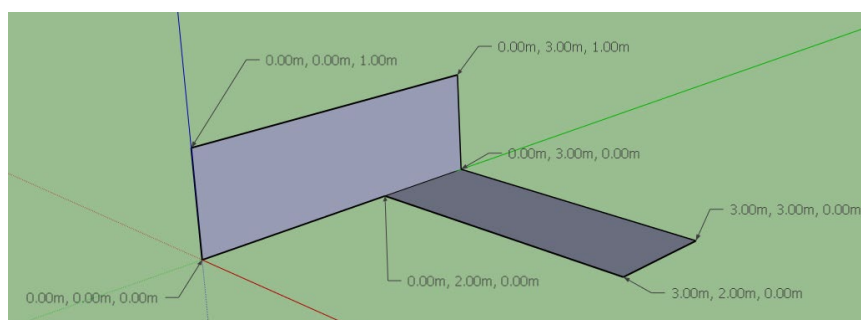


Figura 35 – Test Perp4.

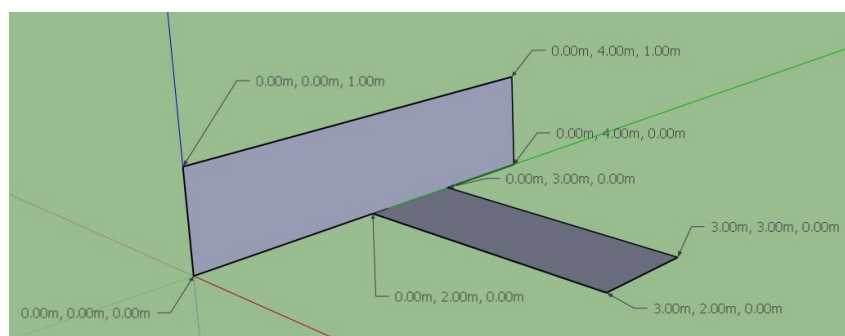


Figura 36 – Test Perp5.

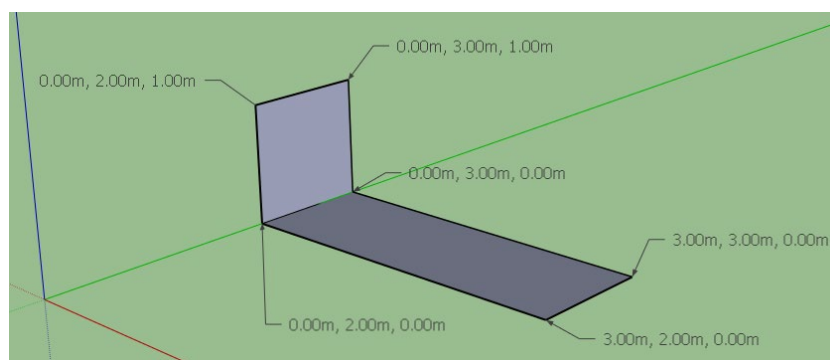


Figura 37 – Test Perp6.

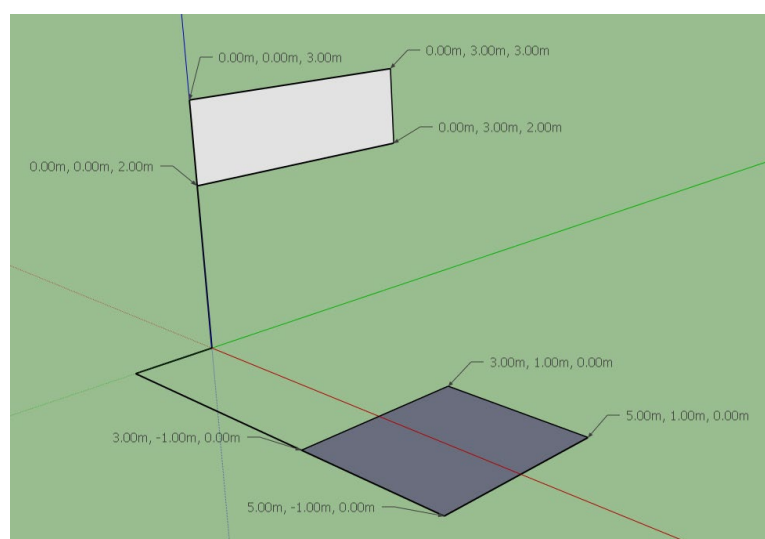


Figura 38 – Test Perp7.

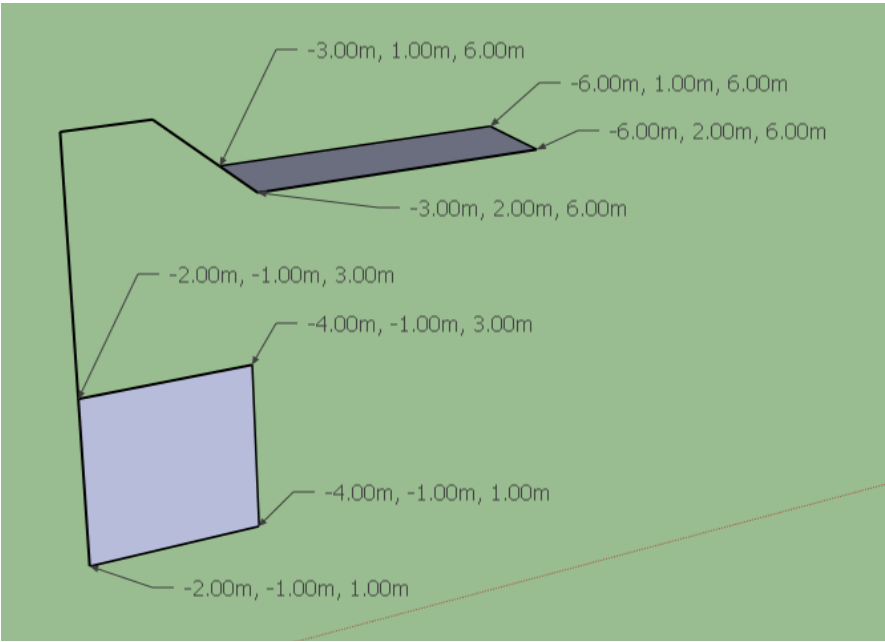


Figura 39 – Test Perp8 (geometria di Perp7 invertita).

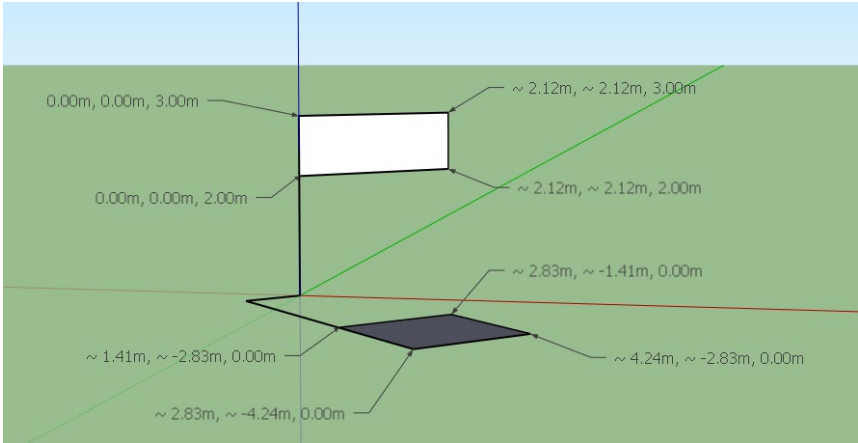


Figura 40 – Test Perp9 (geometria di Perp7 ruotata attorno all’asse z di -45°).

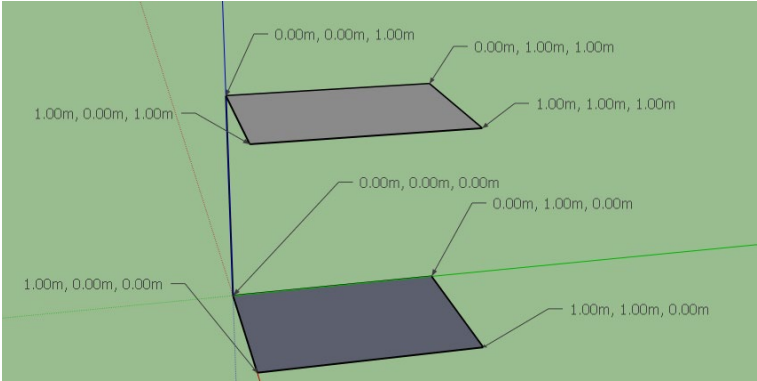


Figura 41 – Test Paralle1

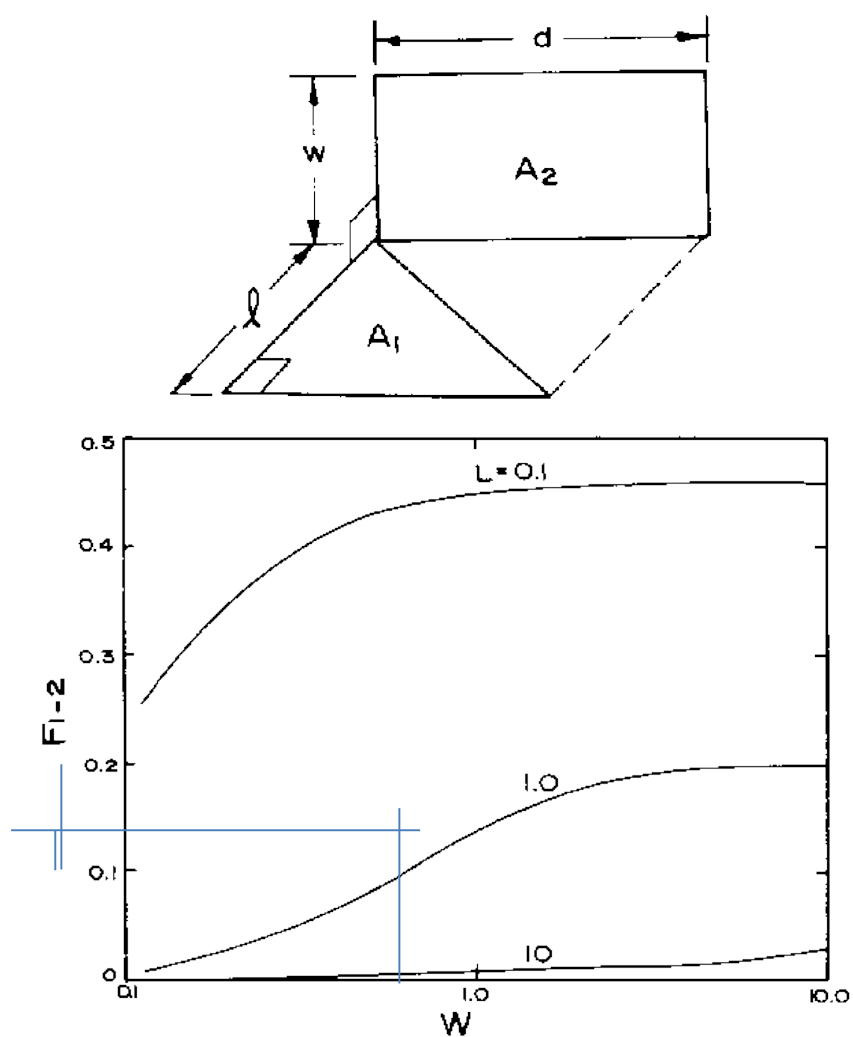


Figura 42 – Test Perp9: $F_{1-2} = 0.138$ per $W=1$ e $L=1$

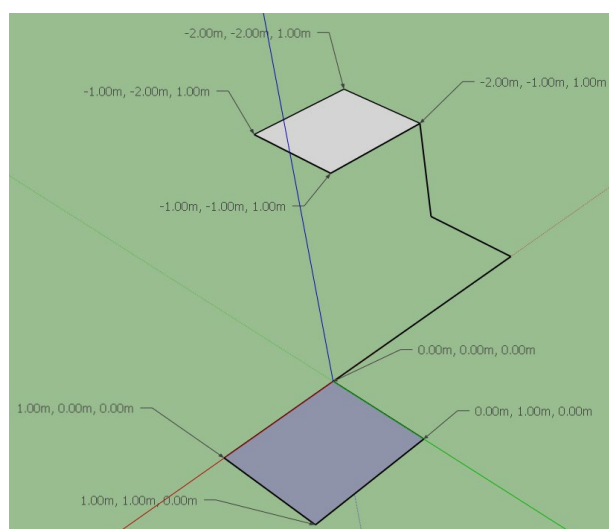


Figura 43 – Test Paralle2

4.4.1.2 Calcolo del fattore di radiazione mutua di corpo grigio

Il calcolo del fattore di radiazione mutua di corpo grigio tra ogni coppia di superfici, R_{ij}^{LW} , viene effettuato solo dopo che siano stati calcolati i fattori di forma di corpo nero F_{ij} tra tutte le superfici che delimitano l'ambiente. Viene quindi costruita una matrice, per ogni zona o ambiente termico, contenente i fattori di radiazione mutua di corpo grigio per ogni coppia di superficie:

1. tramite un metodo di "esatto", i.e. $R_{ij}^{LW} = \varepsilon_i \cdot \varepsilon_j \cdot [\bar{F} (\bar{I} - \bar{D}\bar{F})^{-1}]_{ij}$
2. tramite un metodo "approssimato", ricorrendo al troncamento delle multiple riflessioni (attualmente vengono considerate le riflessioni fino alla terza compresa e non viene consentito all'utente di modificare questo parametro).

Il metodo implicito è stato testato con i risultati ottenuti da TRNSYS 17 per la verifica dei fattori di forma di corpo nero e con un codice Matlab per i termini della matrice di radiazione mutua di corpo grigio.

4.4.1.2.1 Metodo esatto

Considerando che le partizioni, sia trasparenti che opache, siano "opache" alla radiazione ad onde lunghe LW, i.e. $\tau_i^{LW} = 0$, e considerando $\alpha_j^{LW} = \varepsilon_j$, avendo indicato con F_{ij} i fattori di forma per corpo nero e N il numero delle superfici interagenti, abbiamo dalla definizione di fattore di radiazione mutua:

$$\begin{aligned}
 R_{ij}^{LW} &= \frac{\Phi_{i \rightarrow j}^{g.b.}}{\Phi_{i \rightarrow}^{b.b.}} = \frac{A_i \sigma T_i^4 \varepsilon_i F_{ij} \alpha_j^{LW} + \sum_{k=1}^N A_i \sigma T_i^4 \varepsilon_i F_{ik} \rho_k^{LW} F_{kj} \alpha_j^{LW} + \dots}{A_i \sigma T_i^4} \\
 &= \varepsilon_i \cdot \varepsilon_j \left\{ F_{ij} + \sum_{k=1}^N \rho_k^{LW} F_{ik} F_{kj} + \sum_{k=1}^N (\rho_k^{LW} F_{ik} F_{kj})^2 + \dots \right\}. \\
 &= \varepsilon_i \cdot \varepsilon_j \left\{ \bar{F} [\bar{I} + \bar{D}\bar{F} + (\bar{D}\bar{F})^2 + \dots] \right\} = \varepsilon_i \cdot \varepsilon_j \cdot [\bar{F} (\bar{I} - \bar{D}\bar{F})^{-1}]_{ij}^3
 \end{aligned} \tag{4.103}$$

con

$\Phi_{i \rightarrow j}^{g.b.}$ flusso termico emesso dalla superficie grigia i che viene assorbito dalla superficie grigia j

$\Phi_{i \rightarrow}^{b.b.}$ flusso termico emesso dalla superficie grigia i verso tutto l'emisfero sovrastante se fosse nera

$A_i \sigma T_i^4 = \text{sorgente iniziale}$

$F_{ij} \alpha_j^{LW} = \% \text{ che arriva da } i \text{ a } j \text{ ed è assorbito da } j$

$F_{ik} \cdot \rho_k^{LW} = \% \text{ che arriva da } i \text{ a } k \text{ ed è riflesso da } k$

$F_{kj} \cdot \alpha_j^{SW} = \% \text{ che arriva da } k \text{ a } j \text{ ed è riflesso da } j$

$\rho_i^{LW} = 1 - \alpha_i^{LW} = 1 - \varepsilon_i$

$$\bar{I} = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix}; \quad \bar{D} = \begin{bmatrix} \rho_1^{LW} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \rho_N^{LW} \end{bmatrix}; \quad \bar{F} = \begin{bmatrix} F_{11} & \dots & F_{1N} \\ \vdots & \ddots & \vdots \\ F_{N1} & \dots & F_{NN} \end{bmatrix}$$

³ Giacchè per ogni i e j è $F_{ij} < 1$ e $\varepsilon_{ij} < 1$, quindi è $\left| \frac{\bar{D}\bar{F}}{\bar{I}} \right| < 1$; dimostrazione disponibile in Proposition 1.5.31, page 88, di [50].

Cioè la serie delle infinite riflessioni della radiazione scambiata tra le superfici grigie ammette una somma che si determina come prodotto tra matrici che contengono le proprietà geometriche e radiative di tutte le superfici che delimitano l'ambiente.

4.4.1.2.2 Metodo approssimato

Dalla stessa equazione (4.103) si può ricavare una relazione che approssima il fattore di radiazione mutua, consentendo di risparmiare tempo di calcolo qualora le superfici da considerare siano in numero elevato. L'espressione che si ottiene arrestando la serie al terzo termine è:

$$R_{ij}^{LW} \cong \varepsilon_i \cdot \varepsilon_j \left\{ F_{ij} + \sum_{k=1}^N \rho_k^{LW} F_{ik} F_{kj} + \sum_{k=1}^N (\rho_k^{LW} F_{ik} F_{kj})^2 + \sum_{k=1}^N (\rho_k^{LW} F_{ik} F_{kj})^3 \right\} \quad (4.104)$$

4.4.2 Scambio termico radiativo esterno

Lo scambio termico radiativo tra le superficie esterne dell'involucro edilizio e l'ambiente circostante, suddiviso in volta visibile del cielo e piano dell'orizzonte, è modellato tramite equazioni semplificate dell'interazioni tra corpi grigi.

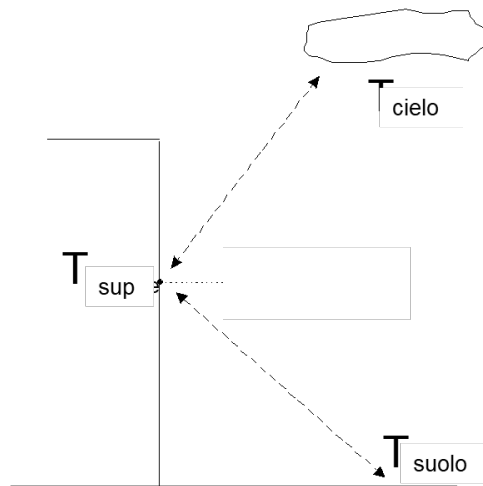


Figura 44 – Interazione radiativa tra involucro, cielo e suolo.

Il modello di calcolo attualmente prevede che la generica superficie dell'involucro interagisca, come mostrato in Figura 44, con solo due elementi: la volta di cielo e di suolo (piano dell'orizzonte) viste dalla superficie.

Le equazioni generali che descrivono tale interazione sono:

$$\Phi_{LW, s \leftrightarrow g} \cong (1 - \cos \beta) / 2 \cdot A \cdot \sigma \cdot (\varepsilon_s \cdot T_s^4 - \varepsilon_g \cdot T_g^4) \quad (4.105)$$

$$\Phi_{LW, s \leftrightarrow sky} \cong (1 + \cos \beta) / 2 \cdot A \cdot \sigma \cdot (\varepsilon_s \cdot T_s^4 - \varepsilon_{sky} \cdot T_{ae}^4) \quad (4.106)$$

con

- β angolo di inclinazione della superficie sul piano dell'orizzonte, in [rd];
- A area della superficie, in [m²];
- σ costante di Stefan-Boltzmann (5,67 10⁻⁸), in [W/(m² K⁴)];
- ε_s emissività della superficie;

- ε_{sky} emissività del cielo;
 T_s temperatura assoluta della superficie, in [K]
 T_g temperatura assoluta del suolo, in [K]
 T_{ae} temperatura assoluta dell'aria esterna, in [K].

NOTA: i termini $(1 + \cos \beta)/2$ e $(1 - \cos \beta)/2$ rappresentano rispettivamente i fattori di forma per radiazione diffusa rispettivamente tra superficie e cielo e tra superficie e piano dell'orizzonte. L'approssimazione risiede nel fatto che si trascurano le inter riflessioni tra superficie e ambiente circostante.

Tale cambio radiativo può essere linearizzato introducendo il coefficiente di scambio termico superficiale radiativo:

$$h_{rd,e} \cong 4\sigma \cdot \bar{T}_m^3 \cdot \sum_{j=1}^N R_{s,j}^{LW} = \varepsilon_s \cdot 4\sigma \cdot \bar{T}_m^3 \quad (4.107)$$

dove

$\bar{T}_m$ è la temperatura media tra superficie e ambiente esterno, differente a seconda che si consideri una:

- a) parete verticale non esposta (canale urbano)

$$\bar{T}_m = \frac{1}{4} (2 \cdot T_w + T_{sky} + T_{a,e}) \text{ [K]}$$

- a) parete verticale esposta

$$\bar{T}_m = \frac{1}{3} (T_w + T_{sky} + T_{a,e}) \text{ [K]}$$

- b) parete orizzontale

$$\bar{T}_m = \frac{1}{2} (T_w + T_{sky})$$

T_w è la temperatura delle facciate degli edifici.

In tal caso lo scambio termico radiativo è valutato tramite la semplice differenza di temperatura tra superficie e ambiente esterno come:

$$\Phi_{LW, s \leftrightarrow AE} \cong A h_{rd,e} (\theta_s - \theta_{LMR,E}) \quad (4.108)$$

con

- $h_{rd,e}$ coefficiente di scambio termico superficiale radiativo, in [W/(m² K)];
 A area della superficie, in [m²];
 θ_s temperatura della superficie, in [°C];
 $\theta_{LMR,E}$ temperatura media radiante lineare esterna, in [°C].

4.4.2.1 Radiazione dal cielo

Difficilmente sono disponibili dati climatici che contengano registrazioni dell'irradianza ad onde lunghe dal cielo, si usano quindi modelli di cielo per la stima di tale radiazione a partire dalla temperatura dell'aria, quale dato climatico disponibile.

In generale sono due i modelli storicamente utilizzati in letteratura:

- **il modello di corpo nero**

che si applica attraverso la definizione di una **temperatura equivalente di corpo nero del cielo**:

si misura la radiazione ad onde lunghe dal cielo, $G_{lw,sky}$, e si ricava dall'irradianza di corpo nero la temperatura equivalente, T_{sky} , come

$$G_{lw,sky} = \sigma \cdot T_{sky}^4 \Rightarrow T_{sky} = \sqrt[4]{G_{lw,sky}/\sigma} \quad (4.109)$$

- **il modello di corpo grigio**

che si applica attraverso la definizione di una **emissività del cielo e della temperatura dell'aria**:

si misura la radiazione ad onde lunghe dal cielo, $G_{lw,sky}$, e si ricava dall'irradianza di corpo grigio, nota la temperatura dell'aria, T_a la emissività del cielo, ε_{sky} , come

$$G_{lw,sky} = \varepsilon_{sky} \cdot \sigma \cdot T_a^4 = \sigma \cdot (\varepsilon_{sky}^{0,25} T_a)^4 = \sigma \cdot T_{sky}^4 \Rightarrow \begin{cases} \varepsilon_a = G_{l,a}/(\sigma \cdot T_a^4) \\ T_{sky} = \varepsilon_a^{0,25} T_a \end{cases} \quad (4.110)$$

Dal secondo modello si può anche ricavare un legame tra temperatura dell'aria e temperatura equivalente di corpo nero del cielo, tramite la sua emissività.

4.4.2.1.1 Modelli implementati in OpenBPS

Sono stati implementati entrambi i modelli che sono presenti in TRNSY e in EnergyPlus.

Il modello di TRNSYS (corpo grigio) è quello di M. Martin e P. Berdahl [51]:

$$\varepsilon_{cls}(t) = \varepsilon_{cs}(t, z) + (1 - \varepsilon_{cs}(t, z)) \cdot \varepsilon_{cl} \cdot c \cdot f_{cloud} = \varepsilon_{cs}(t, z) + (1 - \varepsilon_{cs}(t, z)) \cdot C_c \quad (4.111)$$

con

$$\varepsilon_{cs}(t, z) = \bar{\varepsilon}_{cs} + \Delta \varepsilon_{cs}(t) + \Delta \varepsilon_{cs}(z) = 0.711 + 0.0056 \cdot \bar{\theta}_{dp} + 7.3 \cdot 10^{-5} \cdot \bar{\theta}_{dp}^2 + 0.013 \cdot \cos\left[\frac{\pi \cdot MOD(t, 24)}{12}\right] + 12 \cdot 10^{-5} \cdot (p_{atm}(z) - p_0) \quad (4.112)$$

e

C_c	ammontare nuvoloso infrarosso
ε_{cl}	emissività copertura nuvolosa assunta pari a 0.8
c	copertura nuvolosa relativa (0-1)
f_{cloud}	fattore di copertura nuvolosa $f_{cloud}(z) = e^{-z/8.2}$
θ_{dp}	temperatura a bulbo umido dell'aria, in °C
$p_{atm}(z)$	pressione atmosferica alla quota considerata, z, in mbar;
p_0	1000 mbar.

dove, se non disponibile, l'ammontare nuvoloso infrarosso C_c può essere stimato come []:

$$C_c = e^{-z/z_0} \quad (4.113)$$

con

z l'altezza della base della copertura nuvolosa, in [km]

z_0 8,2 km

Il modello di Energy-Plus (corpo grigio) è quello di G. Clark e C. Allen [52]:

$$\varepsilon_{cls} = \varepsilon_{cs} \cdot (1 + 0.0224 \cdot c - 0.0035 \cdot c^2 + 0.00028 \cdot c^3) \quad (4.114)$$

con

$$\varepsilon_{cs} = 0.787 + 0.764 \cdot \ln(\theta_{dp}) \quad (4.115)$$

e

c copertura nuvolosa relativa (0-1)

θ_{dp} temperatura a bulbo umido dell'aria.

4.5 Scambio radiativo ad onde corte

La radiazione solare costituisce un apporto gratuito per la zona e viene considerata, quando presente, come flusso imposto sulle superfici, interne ed esterne, delle partizioni.

La radiazione solare è:

1. assegnata alle superfici esterne delle partizioni esterne, opache e trasparenti, tenendo conto, per la sua parte diretta, dell'effetto di ombre dovute ad ostruzioni esterne;
2. trasmessa alle superfici interne dalle partizioni trasparenti esterne.

La radiazione solare è assegnata a ciascuna superficie esterna, sia opaca che trasparente, dal componente che simula la zona esterna (classe statica pubblica "ExtZone") che:

1. invoca il componente "SunCalculator", per assegnare a ciascuna superficie esterna un irraggiamento (che potrebbe essere solo diffuso o sia diffuso che diretto)
2. e se la parte di radiazione diretta che colpisce la superficie esterna non è nulla, invoca il componente "SWRadModule" per valutare eventuali ombreggiamenti ed assegnarli alle superfici esterne stesse.

Successivamente, il componente "SWRadModule" distribuirà la radiazione diffusa e proietterà la radiazione diretta, trasmessa dalla parte non ombreggiata delle partizioni trasparenti che confinano con l'esterno, sulle superfici delle partizioni interne, come sarà descritto nei paragrafi seguenti.

Per ogni partizione esterna trasparente, a partire dalla sua superficie esterna, se interessata da radiazione incidente, diffusa e/o diretta:

1. verrà invocato il metodo "assingDirSWFlux" che si occuperà della radiazione diretta trasmessa da fonte primaria, ovvero direttamente dall'esterno:
2. in seguito, verrà invocato il metodo "assignDiffSWFlux" che si occuperà della radiazione diffusa trasmessa da fonte primaria, ovvero direttamente dall'esterno.

La "problematica" della trasmissione di radiazione diffusa fra zone separate da partizioni trasparenti viene quindi affrontata partendo dall'esterno e procedendo con le trasmissioni, riflessioni ed assegnazioni, finché il flusso in gioco non risulti trascurabile.

4.5.1 Elementi trasparenti: caratteristiche

Gli elementi trasparenti dell'involucro di un edificio, sia che siano costituiti da laste di vetro o altro materiale semitrasparente alla radiazione solare, sono caratterizzati dalle proprietà di trasmissione, riflessione e assorbimento della radiazione ad onde corte, oltre che essere comunque delle superfici che scambiare energia all'infrarosso con gli ambienti circostanti.

Essendo le citate proprietà di natura spettrale e dipendenti dall'angolo di incidenza della radiazione, occorre fornire tali dati:

1. con dipendenza angolare;
2. integrati sullo spettro solare;
3. per entrambi i lati dell'unità trasparente e, poiché la radiazione ad onde corte può colpire un'unità vetrata su entrambi i lati (per una finestra esterna, direttamente dall'esterno e indirettamente dall'interno, trasmessa all'interno da un'altra finestra e riflessa da altri superfici) .;
4. per un massimo di tre "elementi" (vetro o ombreggiatura) caratterizzati dai valori delle proprietà termiche e ottiche;

- la avvezione attraverso un elemento non è attualmente stata considerata, ovvero solo gestite solo componenti con le cavità chiuse tra i diversi strati di materiale trasparente (intercapedini).

Ogni partizione trasparente è definita da:

- le sue condizioni al contorno sul lato A e sul lato B (rif. "A" nella Figura 6.1: Descrizione della partizione trasparente)
- la sua InnerConstruction (rif. "B" in Figura 45), che contiene:
 - gli strati dell'elemento trasparente, caratterizzati dalle loro proprietà termiche, come conducibilità, densità e calore specifico (rif. "B1" in Figura 45);
 - la descrizione ottica dell'unità di smaltatura, caratterizzata da valori angolari delle sue proprietà ottiche, quali trasmissione visibile, riflettanza e assorbimento (rif. "B2" in Figura 45).

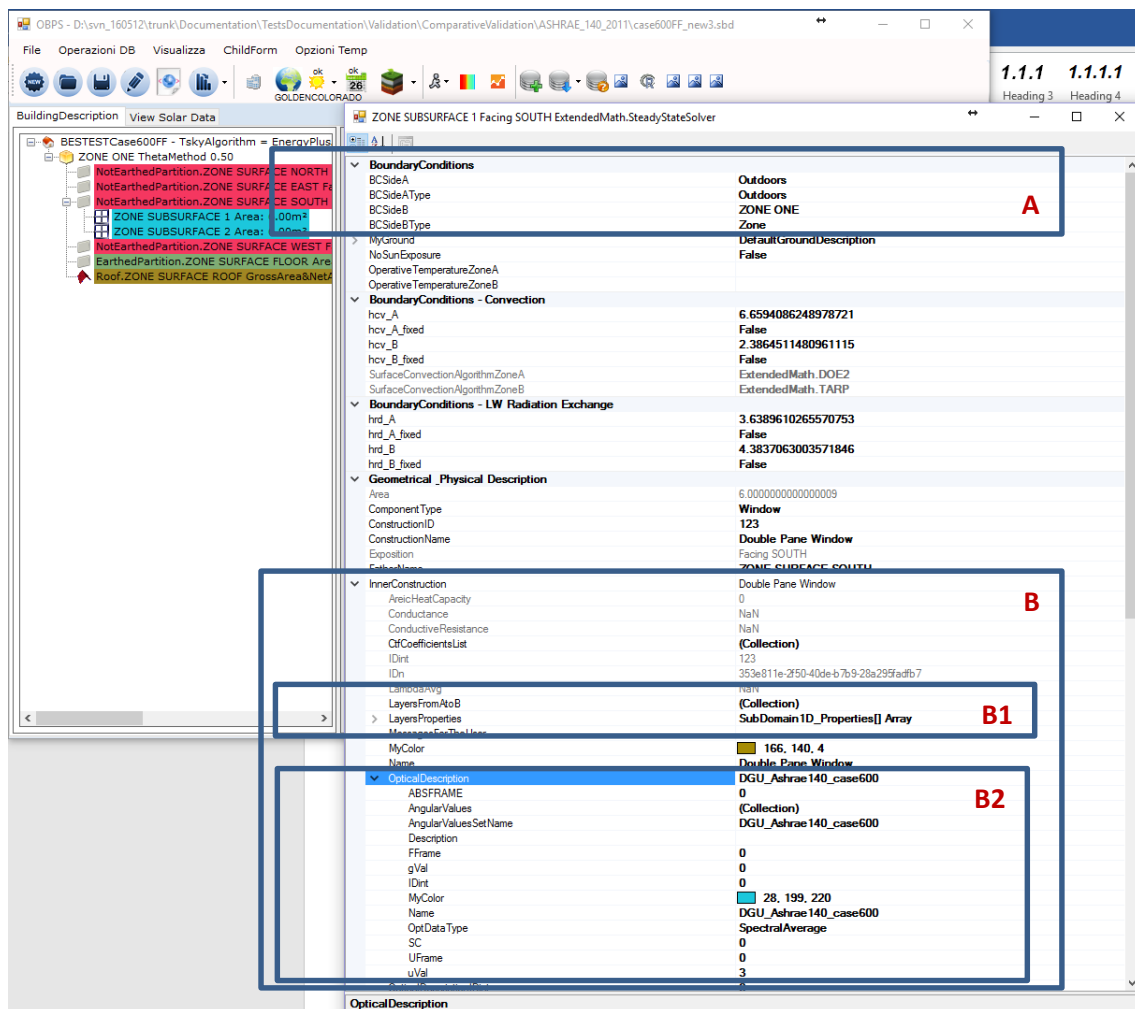


Figura 45 – Descrizione dell'elemento trasparente.

I dati principali delle proprietà ottiche richieste dal codice per ogni valore dell'angolo di incidenza della radiazione solare sono:

- la trasmittanza solare totale per la radiazione solare (cioè integrata su tutte le lunghezze d'onda caratteristiche dello spettro solare) per l'intero sistema trasparente, cioè il rapporto tra la radiazione trasmessa dal lato opposto di quella incidente;

- la riflettanza solare totale, cioè il rapporto tra la radiazione riflessa dalla superficie e la radiazione incidente su tale superficie; tale proprietà deve essere assegnata per entrambi le superfici (esterna e interna), perché l'elemento potrebbe essere non simmetrico;
- l'assorbanza solare totale di ogni singolo elemento (lastra) costituente l'elemento trasparente;

In particolare, le proprietà ottiche angolari richieste dal codice per un numero discreto e finito di valori dell'angolo di incidenza sono, con riferimento a Figura 46, le seguenti:

1. assorbanza solare totale del singolo strato:
 - 1.1. $AbsGlass1FromAtoB_{RadFromA}$ = Assorbanza della prima lastra trasparente da ZoneA verso ZoneB, se la radiazione proviene dalla ZoneA, per lo specificato angolo di incidenza;
 - 1.2. $AbsGlass1FromAtoB_{RadFromB}$ = Assorbanza della prima lastra trasparente da ZoneA verso ZoneB, se la radiazione proviene dalla ZoneB, per lo specificato angolo di incidenza;
 - 1.3. $AbsGlass2FromAtoB_{RadFromA}$ = Assorbanza della seconda lastra trasparente da ZoneA verso ZoneB, se la radiazione proviene dalla ZoneA, per lo specificato angolo di incidenza;
 - 1.4. $AbsGlass2FromAtoB_{RadFromB}$ = Assorbanza della seconda lastra trasparente da ZoneA verso ZoneB, se la radiazione proviene dalla ZoneB, per lo specificato angolo di incidenza;
 - 1.5. $AbsGlass3FromAtoB_{RadFromA}$ = Assorbanza della terza lastra trasparente da ZoneA verso ZoneB, se la radiazione proviene dalla ZoneA, per lo specificato angolo di incidenza;
 - 1.6. $AbsGlass3FromAtoB_{RadFromB}$ = Assorbanza della terza lastra trasparente da ZoneA verso ZoneB, se la radiazione proviene dalla ZoneB, per lo specificato angolo di incidenza;
2. angolo di incidenza: angolo compreso tra la normale alla superficie e la congiungente superficie-sole ;
3. riflettanza solare totale:
 - 3.1. $Rbsol_ZoneA$ = riflettanza solare della superficie rivolta verso la zona A , per lo specificato angolo di incidenza;
 - 3.2. $Rfsol_ZoneB$ = riflettanza solare della superficie rivolta verso la zona B , per lo specificato angolo di incidenza;
4. Trasmittanza solare totale.

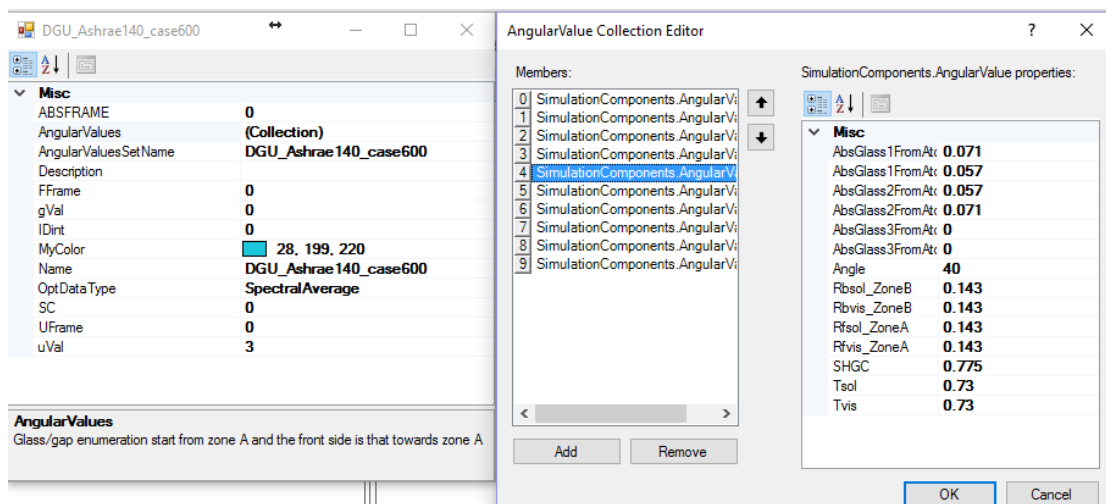
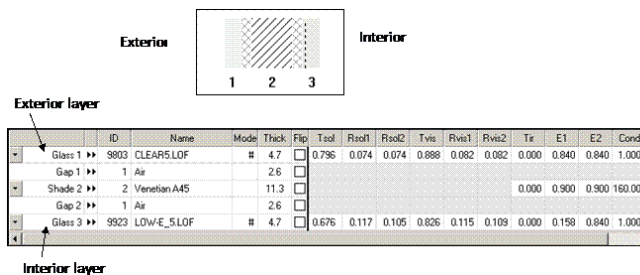


Figura 46 – Descrizione delle proprietà ottiche dell'elemento trasparente,

Per angoli di incidenza compresi tra due valori per cui si hanno i dati assegnati, le proprietà citate vengono interpolate linearmente.

Tutti i dati descritti, necessari, sono ricavabili da software gratuiti come il Berkeley Lab WINDOW 7.4.8.

Se occorre fornire i dati relativi a un doppio vetro che non sia simmetrico, può essere necessario (così come accade con WINDOW 7.4.8) eseguire tali calcoli preliminari due volte, in modo da ricavare i dati di assorbanza richiesti. Di fatto, alcuni software, forniscono i valori di assorbanza degli strati componenti il sistema solo relativamente alla radiazione incidente sulla superficie dichiarata esterna., ad esempio, la prima superficie a sinistra nello schema di WINDOW 7.4.8 (vedasi Figura 47). Quindi, in tal caso, per ottenere le assorbanze di stato per una radiazione proveniente dall'interno e diretta verso l'esterno, occorre invertire l'ordine degli strati e rilanciare l'esecuzione (Figura 48).



	ID	Name	Mode	Thick	Flp	Tsol	Rsol1	Rsol2	Tvis	Rvis1	Rvis2	Tr	E1	E2	Cond
Exterior layer															
Glass 1	9803	CLEAR5_LOF	#	4.7		0.796	0.074	0.074	0.888	0.082	0.082	0.000	0.840	0.840	1.000
Gap 1	1	Air	#	2.6											
Shade 2	2	Venetian A45	#	11.3								0.000	0.900	0.900	160.000
Gap 2	1	Air	#	2.6											
Interior layer															
Glass 3	9923	LOW-E_5_LOF	#	4.7		0.676	0.117	0.105	0.826	0.115	0.109	0.000	0.158	0.840	1.000

Figura 47 – Superfici interne ed esterne per WINDOW 7.4.8

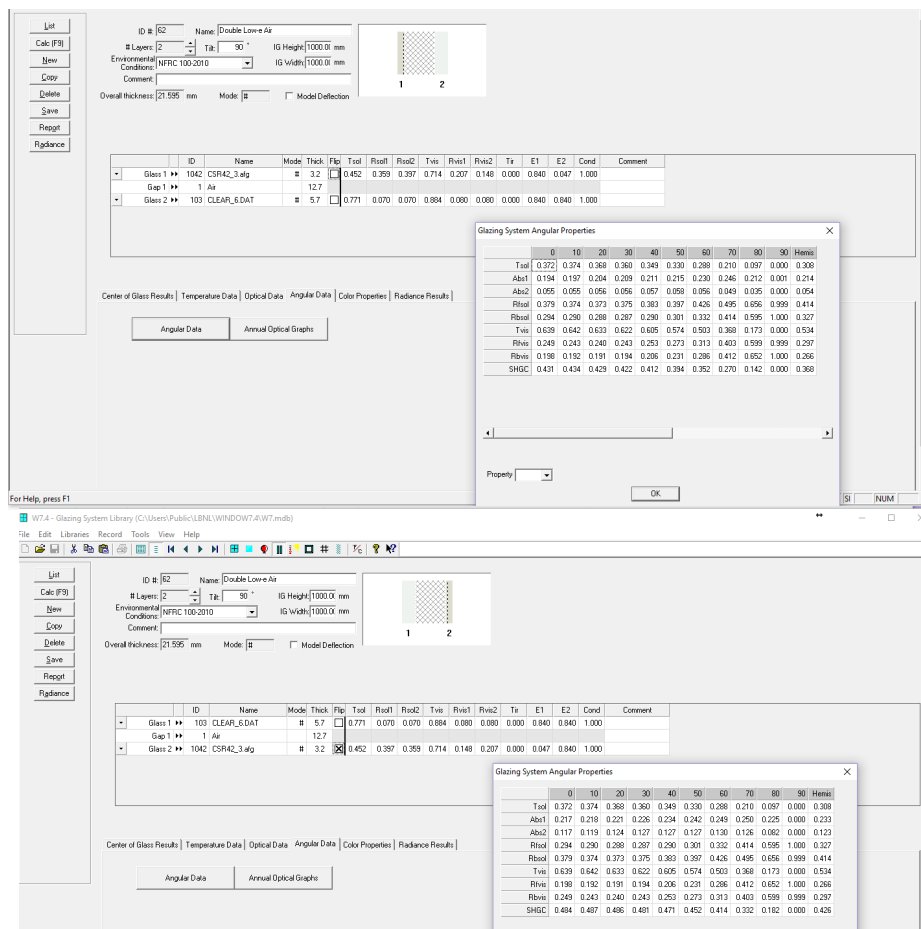


Figura 48 – Input per un elemento trasparente non simmetrico da WINDOW 7.4.8

ATTENZIONE: Nel precedente suggerimento su come ottenere i dati di input da WINDOW 7.4.8, è stata fatta un'approssimazione. Le proprietà ottiche fornite dal software sono quelle totali, cioè integrate nello spettro della radiazione solare esterna, e quando si ribaltano gli strati per calcolare le proprietà per radiazione il programma usa sempre l'interazione sullo spettro solare esterno. Tuttavia, se l'utente ha disponibile un software più sofisticato per produrre l'input richiesto da OpenBPS, l'integrazione sullo spettro potrebbe prendere in considerazione uno spettro filtrato, in funzione delle caratteristiche dello stesso componente trasparente e delle caratteristiche spettrali delle riflettanze delle superfici interne.

4.5.2 Irradianza Diretta

Se la superficie esterna di una componente trasparente è colpita dall'irradiazione solare diretta, il perimetro della superficie interna del componente o il suo perimetro non ombreggiato, impostato durante il calcolo precedente, viene proiettato su tutte le superfici interne delle partizioni appartenenti alla zona.

Per evitare proiezioni che richiedono tempo quando la radiazione diretta trasmessa è molto bassa, l'utente può ridefinire la proprietà dell'edificio, denominata "MinumumTransmittedDirectRadiation", che per impostazione predefinita è 10 W, al di sotto della quale l'irradianza solare che avrebbe dovuto essere trasmessa viene invece completamente assorbita dal componente trasparente.

Pertanto, solo se viene superato il MinumumTransmittedDirectRadiation, l'irradianza solare viene proiettata sulle superfici interne e viene calcolato il flusso radiativo (trascurando effetti minori come la diffrazione, ecc.) che, in base al perimetro della sorgente di radiazione solare, colpisce una o più superfici interne della zona.

Tuttavia, siccome per adesso lo scambio termico per conduzione viene risolto nell'ipotesi di monodimensionalità, cioè non si tiene conto di condizioni al contorno diversificate in funzione della posizione di un generico punto sulla superficie, questo flusso, che in realtà incide solo su una parte della superficie della partizione, viene distribuito su tutta l'area della superficie. In futuro, quando le condizioni al contorno saranno date tramite coordinate tridimensionali a un dominio multidimensionale, la risoluzione del problema conduttivo potrà trarre beneficio da tale algoritmo geometrico già implementato.

La radiazione diretta incidente, trasmessa all'interno della zona ancora come radiazione diretta (a meno che non vi sia uno schermo diffondente), dopo aver raggiunto le superfici interne, viene riflessa:

1. in maniera diffusa, se incidente su di una superficie opaca;
2. specularmente, se incidente su di una superficie trasparente.

Come anticipato nel paragrafo introduttivo, per ogni partizione esterna trasparente, a partire dalla sua superficie esterna, se interessata da radiazione incidente, diffusa e/o diretta verrà invocato il metodo "assingDirSWFlux" che si occuperà della radiazione diretta trasmessa da fonte primaria, ovvero direttamente dall'esterno secondo il seguente schema:

1. partendo dal perimetro della superficie trasparente, o da quello della sua parte non ombreggiata, sarà invocato il metodo "ProjectRay" che si occupa della proiezione di tale perimetro sulle superfici delle altre partizioni:
 - 1.1. trovata l'intersezione con una **partizione opaca**, la frazione assorbita di radiazione solare viene assegnata tramite il metodo "AddImposedSWDensityFlux" come flusso imposto sulla superficie e, se la radiazione riflessa supera una certa soglia (valore di default o impostato dall'utente), viene riflessa come radiazione diffusa, invocando il metodo "assignDiffSWFlux" al quale viene passata l'area della partizione riflettente, che costituisce la sorgente di radiazione diffusa "secondaria", altrimenti viene interamente assorbita dalla partizione opaca;

1.2. trovata l'intersezione con una **partizione trasparente**:

- la radiazione assorbita e trasmessa è assegnata alla superficie in questione, tramite la sua proprietà "IncRadSideA.Add" per il successivo calcolo degli assorbimenti nelle diverse lastre di vetro
- se la radiazione trasmessa supera un valore limite "MinumumTransmitted DirectRadiation" viene trasmessa verso eventuali altre zone confinanti con la partizione trasparente interna in esame, altrimenti viene assegnata alla superficie in esame come flusso imposto mediante il metodo "AddImposedSWDensityFlux";
- fino ad un numero massimo di riflessioni la radiazione solare incidente viene riflessa specularmente ed assegnata alle altre superfici invocando nuovamente il metodo "ProjectRay", se la radiazione riflessa supera una certa soglia (valore di default o impostato dall'utente) in caso contrario viene assegnata come flusso imposto alla superficie che dovrebbe rifletterla;
- oltre il numero massimo di riflessioni "MaximumSolarReflectionNumber" (valore di default o impostato dall'utente), viene riflessa in maniera diffusa se la radiazione riflessa supera una certa soglia (valore di default o impostato dall'utente), in caso contrario viene assegnata come flusso imposto alla superficie che dovrebbe rifletterla. Un valore nullo del parametro MaximumSolarReflectionNumber, comporta anche la prima riflessione diffusa da parte delle superfici trasparenti

2. in seguito, verrà invocato il metodo "assignDiffSWFlux" che si occuperà della radiazione diffusa trasmessa da fonte primaria, ovvero direttamente dall'esterno.

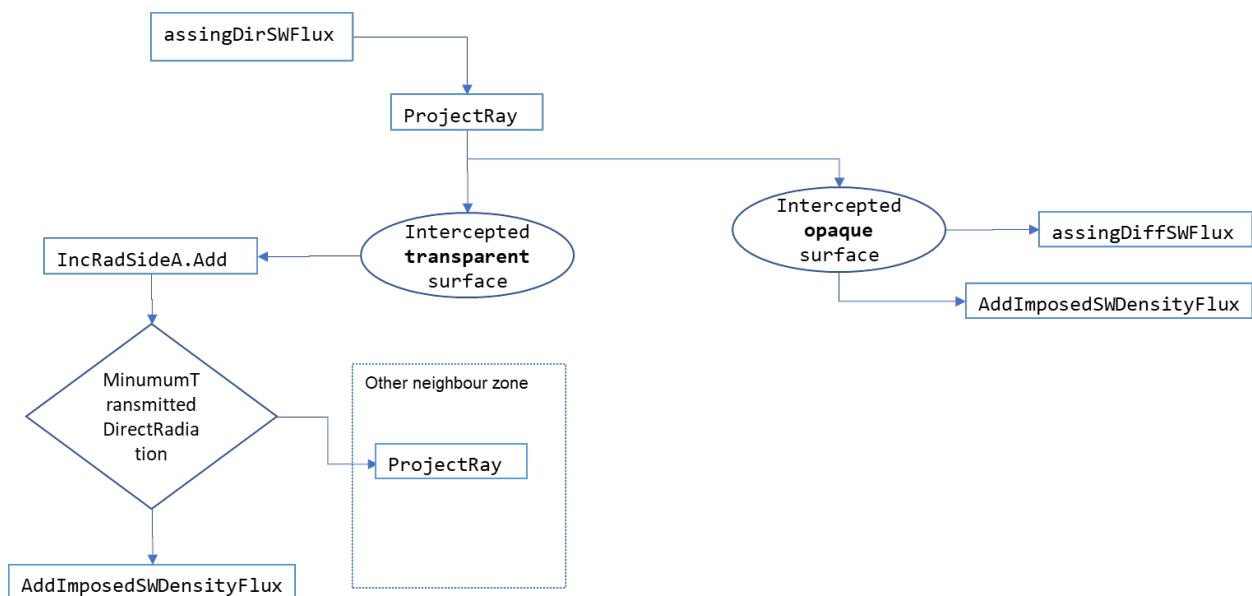


Figura 49 – Schema logico trattamento radiazione solare diretta

4.5.3 Irradianza Diffusa

Il componente SWRadModule ha il compito di calcolare la matrice di riflessione mutua per mezzo della quale distribuirà la radiazione diffusa uscente da una superficie sulle altre superfici interne di una zona. L'unica sostanziale differenza, rispetto al caso trattato per la radiazione ad onde lunghe, è che il fattore di radiazione

mutua in questo caso non è riferito al flusso emesso da un corpo nero a pari temperatura della superficie grigia per cui si calcola, ma è riferito al flusso radiativo generato dalla sorgente diffondente, in questo caso una superficie semitrasparente diffondente; cioè:

$$R_{ij,A}^{SW} = \frac{\Phi_{i \rightarrow absByj}^{SW}}{\Phi_{i \rightarrow}^{SW}} \quad (4.116)$$

con

$\Phi_{i \rightarrow absByj}^{SW}$ flusso radiativo diffuso ad onde corte che dall'elemento trasparente-diffondente i viene assorbito dalla superficie opaca j ;

$\Phi_{i \rightarrow}^{SW}$ flusso radiativo diffuso ad onde corte che complessivamente lascia la superficie dall'elemento trasparente-diffondente i

La sorgente di radiazione diffusa ad onde corte può essere “primaria” o “secondaria”, infatti tale radiazione può raggiungere la superficie interna di una zona per i seguenti motivi:

1. perché arriva direttamente dall'esterno e viene trasmessa all'interno da una superficie trasparente, che costituisce la sorgente di radiazione diffusa “primaria”;
2. perché arriva da un'altra zona e viene trasmessa all'interno da una superficie trasparente che divide le due zone e che costituisce la sorgente di radiazione diffusa “secondaria”;
3. in seguito a riflessione diffusa della radiazione diretta incidente su una partizione interna, la cui area illuminata costituisce la sorgente di radiazione diffusa “secondaria”; (in questo caso la sorgente di radiazione diffusa secondaria coinciderebbe con la sola parte irraggiata della partizione, ma per non dover ricalcolare ad ogni timestamp i fattori di radiazione mutua, viene considerata l'intera superficie della partizione riflettente).

La “problematica” della trasmissione di radiazione diffusa fra zone separate da partizioni trasparenti viene quindi affrontata partendo dall'esterno e procedendo con le trasmissioni, riflessioni ed assegnazioni, finché il flusso in gioco non risulti trascurabile.

A questo punto possono essere prese due strade:

1. ci si può interessare della sola radiazione assorbita dalla partizione (trasparente o opaca) ed ignorare la radiazione che fosse eventualmente trasmessa dalla partizione trasparente, in quanto il più delle volte viene trasmessa nuovamente all'esterno non costituendo un termine utile al resto del calcolo;
2. si può valutare la problematica che incontreremmo qualora due zone siano divise da una partizione trasparente, in quanto parte della radiazione incidente da un lato della partizione interna, verrebbe trasmessa nella zona accanto.

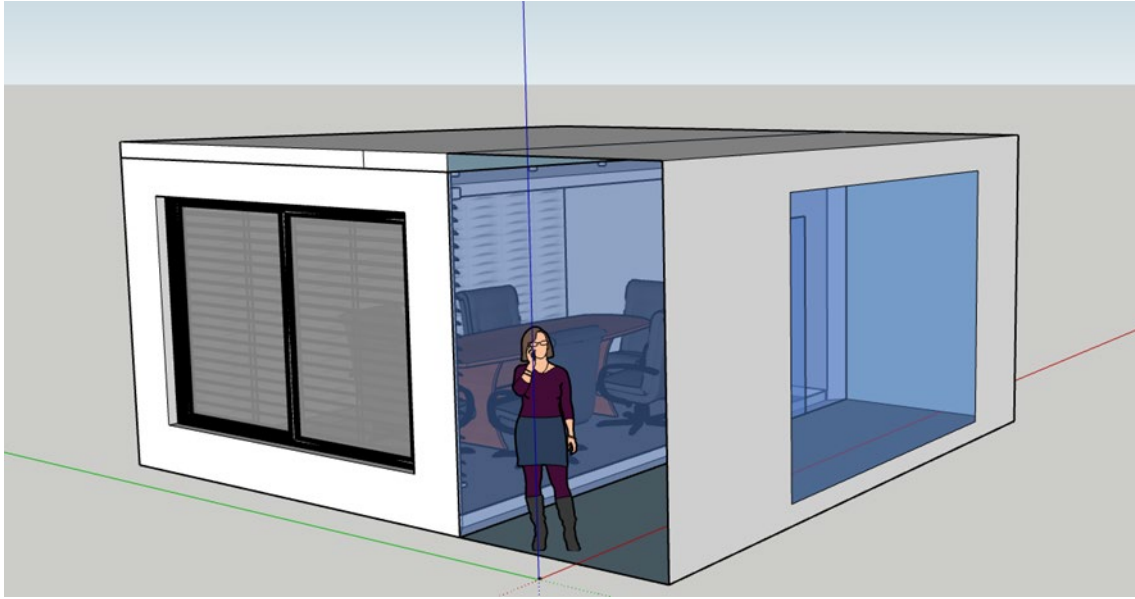


Figura 50 – Doppia trasmissione esterno ambiente A – ambiente B.

Nel primo caso, radiazione incidente su elemento opaco, avremmo:

$$R_{ij,A}^{SW} = \frac{\Phi_{i \rightarrow absByj}^{SW}}{\Phi_{i \rightarrow}^{SW}} = \frac{A_i \cdot \tau_i^{SW} \cdot G_i \cdot F_{ij} \cdot \alpha_j^{SW} + \sum_k A_i \cdot \tau_i^{SW} \cdot G_i \cdot F_{ik} \cdot \rho_k^{SW} \cdot F_{kj} \cdot \alpha_j^{SW} + \dots}{A_i \cdot \tau_i^{SW} \cdot G_i} \quad (4.117)$$

Nel secondo caso, radiazione incidente su elemento trasparente, avremmo:

$$\begin{aligned} R_{ij,A}^{SW} &= \frac{\Phi_{i \rightarrow abs\&transmByj}^{SW}}{\Phi_{i \rightarrow}^{SW}} \\ &= \frac{A_i \cdot \tau_i^{SW} \cdot G_i \cdot F_{ij} \cdot (1 - \rho_{j,A}^{SW}) + \sum_k A_i \cdot \tau_i^{SW} \cdot G_i \cdot F_{ik} \cdot \rho_{k,A}^{SW} \cdot F_{kj} \cdot (1 - \rho_{j,A}^{SW}) + \dots}{A_i \cdot \tau_i^{SW} \cdot G_i} \end{aligned} \quad (4.118)$$

dove

$$A_i \cdot \tau_i^{SW} \cdot G_i = \text{ sorgente iniziale}$$

$$\Phi_{i \rightarrow absByj}^{SW} = \text{flusso proveniente da i assorbito da j}$$

$$F_{ij} \cdot \alpha_j^{SW} = \% \text{ che arriva da i a j ed è assorbita da j}$$

$$F_{kj} \cdot \alpha_j^{SW} = \% \text{ percentuale che arriva da k a j ed è assorbita da j}$$

$$\Phi_{i \rightarrow abs\&transmByj}^{SW} = \text{flusso proveniente da i assorbito e trasmesso fuori dall'ambiente da j}$$

$$F_{ij} \cdot (1 - \rho_{j,A}^{SW}) = \% \text{ che arriva da i a j ed è assorbita e trasmessa fuori dall'ambiente da j}$$

$$F_{ik} \cdot \rho_k^{SW} = \% \text{ che arriva da i a k ed è riflessa da k}$$

$F_{kj} \cdot (1 - \rho_j^{SW}) = \% \text{ che arriva da } k \text{ a } j \text{ ed è assorbito e trasmesso fuori dall'ambiente da } j$

Possiamo dunque scrivere:

$$\mathbf{R}_{ij,A}^{SW} = (1 - \rho_j^{SW}) \left[F_{ij} + \sum_k F_{ik} \cdot \rho_k^{SW} \cdot F_{kj} + \sum_{k,l} F_{ik} \cdot \rho_k^{SW} \cdot F_{kl} \cdot \rho_l^{SW} \cdot F_{lj} + \dots \right] \quad (4.119)$$

$$\mathbf{R}_{ij,A}^{SW} = (1 - \rho_j^{SW}) \{ \bar{F} [\bar{I} + \bar{D}\bar{F} + (\bar{D}\bar{F})^2 + \dots] \}_{ij} = (1 - \rho_j^{SW}) [\bar{F} (\bar{I} - \bar{D}\bar{F})^{-1}]_{ij} \quad (4.120)$$

dove:

$$\bar{I} = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix}; \quad \bar{D} = \begin{bmatrix} \rho_1^{SW} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \rho_N^{SW} \end{bmatrix}; \quad \bar{F} = \begin{bmatrix} F_{11} & \dots & F_{1N} \\ \vdots & \ddots & \vdots \\ F_{N1} & \dots & F_{NN} \end{bmatrix}$$

Quando la superficie ricevente è una partizione opaca, il flusso solare diffuso, trasmesso dalla finestra esterna, viene diviso per l'area netta della partizione e considerato come un flusso imposto in una condizione al contorno "combinata", insieme alla temperatura operativa (nella maggior parte dei casi).

Viceversa, quando si considera una partizione trasparente, il calcolo è più articolato, di fatto

1. il fattore di radiazione mutua per radiazione ad onde corte relativo a due superfici appartenenti ad una zona (zona A, per esempio), $\mathbf{R}_{ij,A}^{SW}$, è stato calcolato utilizzando per il componente trasparente un coefficiente di assorbimento "globale", $\alpha_{j,A}^{SW}$, questo nel caso in cui la radiazione proviene dalla zona per il quale il fattore è stato calcolato. Tale coefficiente è stato calcolato come $\alpha_{j,A}^{SW} = 1 - \tau_j^{SW,diff} - \rho_{j,A}^{SW,diff}$, e potrebbe essere differente per un componente trasparente se la radiazione solare lo colpisce sul suo lato A (direttamente dall'esterno) o dal suo lato B (trasmessa all'interno della zona da un altro componente trasparente).
2. poiché per il calcolo della prestazione del componente trasparente è necessario considerare l'effettivo coefficiente di assorbimento di ogni strato, il flusso solare ottenuto moltiplicando quello incidente per $\mathbf{R}_{ij,A}^{SW}$, viene diviso per il coefficiente di assorbimento globale, $\alpha_{j,A}^{SW}$, e considerato come flusso imposto (che deve essere assorbito dalle lastre del componente trasparente rispettando l'assorbimento globale) $\rightarrow \alpha_{j,A}^{SW} = \sum_{i=1}^{n \text{ glasses}} \alpha_{j,A,i}^{SW}$ alla superficie del componente dal lato in cui la radiazione arriva:

- 2.1. se il componente trasparente è interno, i.e. l'altra superficie è rivolta verso una zona interna, il flusso imposto è moltiplicato per il coefficiente di trasmissione diffusa e

- 2.1.1. se il flusso trasmesso è maggiore del valore soglia impostato dall'utente con la proprietà dell'edificio chiamata "MinumumTransmittedDiffuseRadiation", che per default è posta pari a 10 W, questo è assegnato a tutte le partizioni dell'altra zona, tramite il fattore di radiazione mutua per onde corte dell'altra zona

- 2.1.2. se il flusso trasmesso è minore del valore soglia impostato dall'utente con la proprietà dell'edificio chiamata "MinumumTransmittedDiffuseRadiation" che per default è posta pari a 10 W, questo è assegnato come flusso imposto (che deve essere completamente assorbito dalle lastre costituenti il componente trasparente) sulla superficie del componente trasparente dal lato in cui la radiazione arriva;

- 2.2. se il componente trasparente è rivolto verso l'esterno, il flusso trasmesso non viene calcolato e il calcolo prosegue con un altro componente.

Ad esempio, si abbiano tre zone A , B e C, di cui la zona A abbia una finestra verso l'esterno e una finestra verso la zona B interna (cioè senza finestre verso l'esterno) e la zona C abbia anch'essa una finestra verso l'esterno e una finestra verso la zona B. Supponiamo che dalla finestra della zona A con esposizione nord, passi una radiazione solare diffusa pari a 1242 W, e che la finestra interna con la zona B abbia un $R_{ij,A}^{SW}$ pari a 0.22 e un coefficiente di trasmissione diffusa pari a 0,8 , mentre per la zona B dalla finestra (sempre a nord) passi una radiazione solare diffusa inferiore pari a 800 W, con finestra interna con la zona B che abbia le stesse proprietà della zona A, cioè $R_{ij,A}^{SW}$ pari a 0.22 e coefficiente di trasmissione diffusa pari a 0,8, si ha che:

1. passano direttamente nella zona B circa 220 W e
 - 1.1 dopo una prima riflessione dalla zona B, ritornano nella zona A, ad esempio, circa 14 W, di cui 1 W verrebbe ritrasmessi nella zona B per ulteriori inter-riflessioni nella zona, ma poiché sotto i 10 W viene imposto come completamente assorbito delle partizioni della zona A;
 - 1.2 dalla zona B vanno nella zona C, attraverso la seconda finestra interna, circa 42 W, i quali poi vengono assegnati tutti come flussi assorbiti alle partizioni della zona C in quanto la parte potenzialmente ritrasmessa indietro risulta è inferiore a 1 W
- 2 passano direttamente nella zona C circa 141 W, che vengono tutti assegnati alle superfici della zona 5 perché le ritrasmissioni sarebbero inferiori ai 10 W

4.6 Apporti interni

Gli apporti interni sono dovuti alla presenza di persone, dispositivi elettrici, illuminazione artificiale e quant'altro rilasci energia sotto forma convettiva, avveviva e radiativa nella generica zona termica. Tali apporti sono di fatto dati del problema che devono essere assegnati dall'utente del codice, in funzione del meccanismo di interazione tra "sorgente" e zona termica, vedasi Figura 51. Con meccanismo di interazione si intende distinguere gli apporti energetici dovuti a persone e cose che occupano un dato spazio interno in:

- apporti di tipo convettivo,
cioè dovuti allo scambio termico convettivo tra superfici di persone e cose e nodo aria;
- apporti di tipo radiativo a onde lunghe,
cioè dovuti allo scambio termico radiativo ad onde lunghe tra superfici di persone e cose e le pareti della zona termica;
- apporti di tipo radiativo ad onde corte,
legati all'illuminazione artificiale;
- apporti di tipo latente,
cioè produzione di vapore da parte di persone e cose all'interno dello spazio della zona termica.

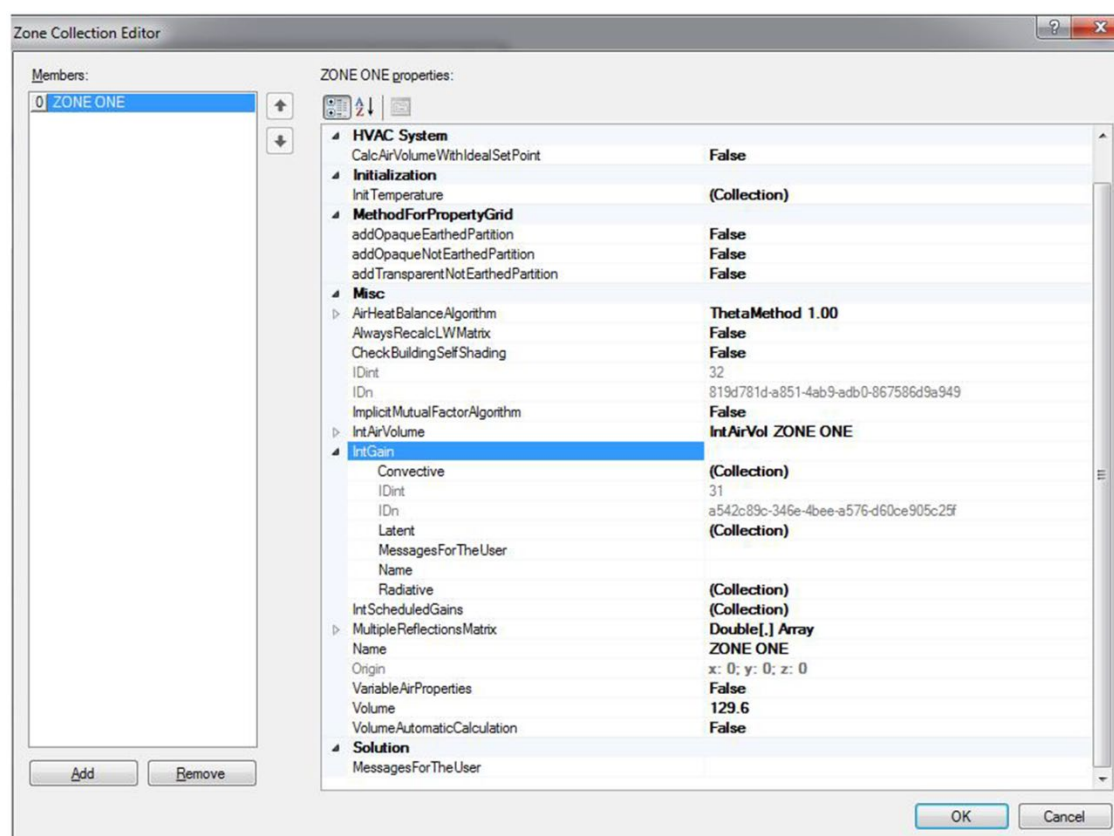


Figura 51 – Maschera per l'assegnazione degli apporti interni alla singola zona termica

Per ognuna delle citate tipologie di apporti interni occorre specificare, oltre il valore della potenza erogata o della quantità di vapore prodotta, la distribuzione temporale, sia in termini di distribuzione oraria giornaliera, sia in termini di periodo di presenza di tale apporto (da che data a che data) (Figura 52).

I dettagli di come vengono computati tali apporti nel bilancio termico della zona sono descritti in tale paragrafo, a cui si rimanda.

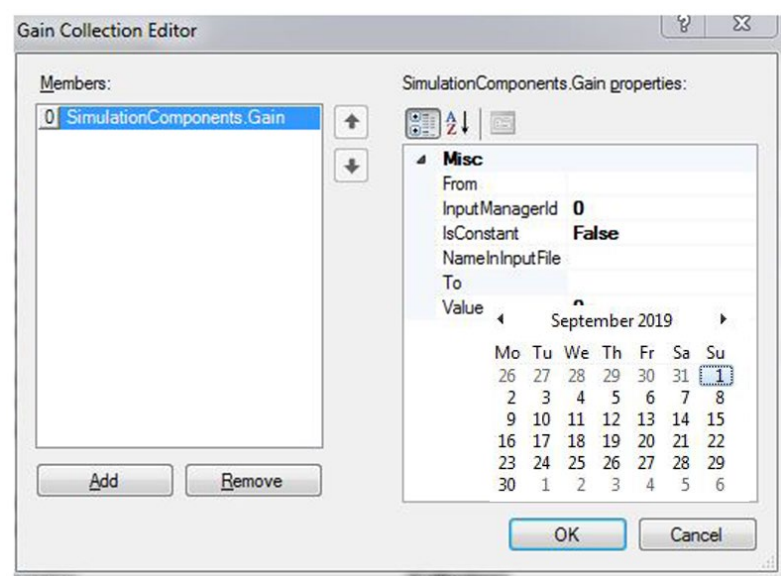


Figura 52 – Maschera per l’assegnazione del profilo temporale degli apporti interni

4.7 Trasmissione di calore per conduzione

I componenti del fabbricato interagiscono con l'ambiente intero e l'ambiente esterno in modo dinamico accumulando e cedendo energia; il meccanismo principale che regola questa interazione è quello della conduzione termica. Nel seguito vengono indicati e spiegati i modelli che sono stati implementati nel codice di calcolo per simulare tale comportamento.

4.7.1 Modelli implementati

Sono stati implementati in OpenBPS due principali famiglie di modelli per lo scambio termico conduttivo:

1. modelli dinamici, usati principalmente per i component opachi;
2. il modello stazionario, usato per i componenti poco capacitivi, in particolare quelli trasparenti.

Nei paragrafi seguenti, si dettagliano i modelli implementati.

4.7.2 Modello matematico della conduzione termica

L'equazione della conduzione termica è modellata matematicamente con un'equazione parabolica alle derivate parziali [53]. Per ottenere un problema ben posto, tale equazione parabolica deve essere accoppiata con condizioni al contorno e iniziali consistenti, come mostrato nel seguente sistema di equazioni:

$$\begin{cases} \frac{\partial(\rho c \vartheta)}{\partial t} + L\vartheta = f & t > 0, \bar{x} \in \Omega \\ \vartheta(\bar{x}, 0) = \vartheta_0(\bar{x}) & t = 0, \bar{x} \in \Omega \\ BC_1(\bar{x}, t) & t > 0, \bar{x} \in \Gamma_1 \\ BC_2(\bar{x}, t) & t > 0, \bar{x} \in \Gamma_2 \end{cases} \quad (4.121)$$

dove:

- l'incognita è la distribuzione di temperatura, $\vartheta = \vartheta(\bar{x}, t)$ valutata alla generica coordinata spaziale $\bar{x}$, nel dominio d -dimensionale, e allo specifico tempo t ,
- $\vartheta(\bar{x}, 0) = \vartheta_0(\bar{x})$ è la condizione iniziale assegnata,
- Ω è un dominio di $\mathbb{R}_d$, $d = 1, 2, 3$, che comprende il suo confine $\partial\Omega$,
- $f = f(\bar{x}, t)$ è una possibile funzione forzante nota,
- $L = L(\bar{x})$ è il generico operatore ellittico applicato all'incognita $\vartheta = \vartheta(\bar{x}, t)$, che per la conduzione termica è pari a $L\vartheta = -\text{div}[\bar{\lambda} \cdot \text{grad}(\vartheta)]$, o, se il dominio d -dimensionale è omogeneo e la conduttività è non dipendente dalla temperatura, diventa $L\vartheta = -\lambda \cdot \Delta\vartheta$
- $\{\Gamma_1, \Gamma_2\}$ è una partizione del contorno del dominio che soddisfa le seguenti condizioni: $\Gamma_1 \cup \Gamma_2 = \partial\Omega$, $\Gamma_1^\circ \cap \Gamma_2^\circ = \emptyset$, dove la notazione Γ° viene usata per indicare l'interno di Γ ,
- $BCxs$ sono le condizioni al contorno sul confine dello spazio d -dimensionale, che possono essere di diverso tipo su sottodomini diversi del confine.

Quando si risolve il problema differenziale su un intervallo temporale finito, per esempio tra $0 < t < T$, la regione $QT = \Omega \times (0, T)$ è identificata come "cilindro nello spazio $\mathbb{R}_d \times \mathbb{R}^+$ " come rappresentato in Figura 53.

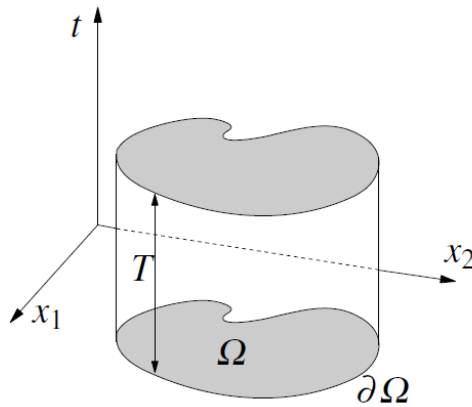


Figura 53 – Cilindro $Q_T = \Omega \times (0, T)$, $\Omega \subset \mathbb{R}_2$

Nel caso di uno spazio monodimensionale, 1D, con proprietà materiali che sono indipendenti dalla soluzione e dalle coordinate spaziali, e senza generazione interna di calore, si ha:

$$\begin{cases} \frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} & t > 0, x \in \Omega \\ T(x, 0) = T_0(x) & t = 0, x \in \Omega \\ BCs(x, t) & t > 0, x \in \partial\Omega \end{cases} \quad (4.122)$$

con

$\alpha = \lambda/(\rho \cdot c)$ diffusività termica del mezzo solido

Le condizioni al contorno assumono la seguente forma generale:

$$T(x, t) = \varphi(x, t), \quad x \in \partial\Omega \quad e \quad t > 0,$$

$$\partial T(x, t)/\partial n = \psi(x, t), \quad x \in \partial\Omega_n \quad e \quad t > 0$$

La prima equazione rappresenta la condizione al contorno di Dirichlet (funzione nota imposta sul contorno). La seconda equazione descrive invece sia le condizioni al contorno di Newman (derivata della funzione incognita funzione nota sul contorno) che di Robin (derivata della funzione incognita funzione nota dell'incognita sul contorno), da cui si ottiene la condizione di Newman se la funzione ψ non è dipendente dalla soluzione.

L'integrazione (cioè la determinazione della soluzione) del problema differenziale della conduzione termica può essere effettuato con tecniche diverse, ognuna con i suoi pro e contro. Una prima distinzione tra i metodi di integrazione è tra:

- metodi analitici;
- metodi numerici.

La distinzione è solo nell'approccio iniziale perché tutti i metodi in realtà fanno uso sia di procedure analitiche, sia di tecniche numeriche. In ogni caso le soluzioni trovate saranno in qualche modo approssimate in quanto:

- i dati che descrivono le condizioni al contorno sono noti in forma discreta e non continua, e quindi soggetti a qualche procedura di interpolazione;
- l'impossibilità pratica di determinare una soluzione esatta del problema se non per casi molto semplici.

4.7.3 Soluzione analitica monodimensionale in regime periodico stabilizzato

Dominio di soluzione:

1. dominio monodimensionale,
2. sistema multistrato (ogni strato è omogeneo),
3. stessa periodicità per entrambi le condizioni al contorno,
4. condizioni al contorno funzioni sinusoidali/cosinusoidali del tempo,
5. condizioni al contorno implementate: di Dirichlet e Robin.

La soluzione a regime (periodico stabilizzato) è data dalla sovrapposizione della soluzione stazionaria, riferita alle componenti medie temporali delle condizioni al contorno, e alla soluzione armonica per condizioni al contorno armoniche, periodiche di periodo P , con valore medio nullo. Se infatti consideriamo il valore medio sul periodo P , della funzione incognita $T(x, t)$, si ha:

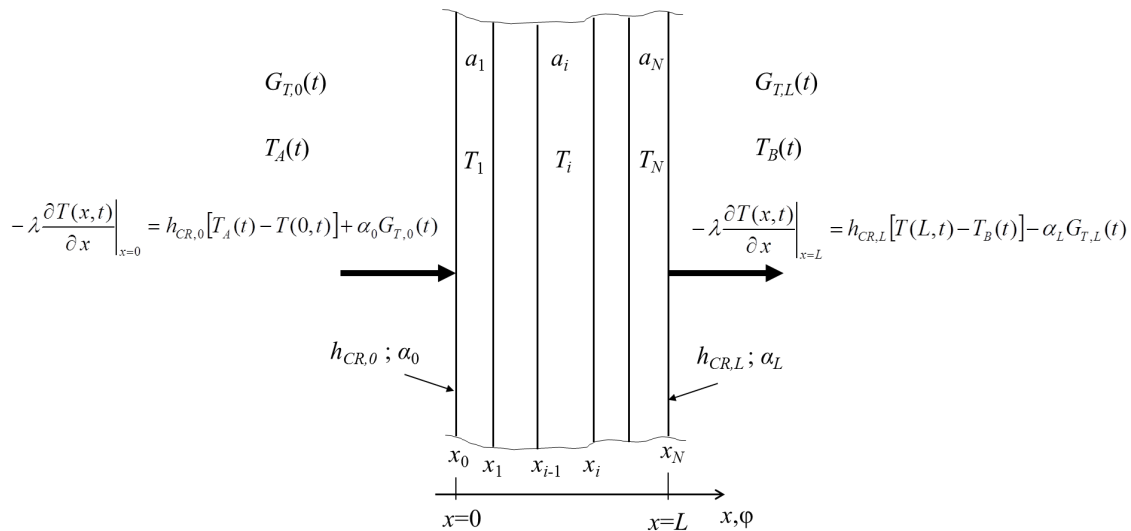


Figura 54 – Parete piana indefinita multistrato con condizioni al contorno di Robin

$$\bar{T}(x) \equiv \frac{1}{P} \int_0^P T(x, t) dt \quad (4.123)$$

$$\vartheta(x, t) = T(x, t) - \bar{T}(x) \quad (4.124)$$

dove

- $\bar{T}(x)$ parte stazionaria della soluzione;
 $\vartheta(x, t)$ parte periodica della soluzione.

Applicando tale approccio anche alle condizioni al contorno, quelle di Robin scritte in termini di flusso termico areico scambiato all'interfaccia tra parete e ambienti, cioè:

$$\begin{aligned} -\lambda \frac{\partial T(x,t)}{\partial x} \Big|_{x=0} &= h_{cv,0}[T_{ai}(t) - T(0,t)] + h_{rd,0}[T_{MRL,I}(t) - T(0,t)] + \alpha_0 G_{T,0}(t) \\ -\lambda \frac{\partial T(x,t)}{\partial x} \Big|_{x=L} &= h_{cv,L}[T(L,t) - T_{ae}(t)] + h_{rd,L}[T(L,t) - T_{MRL,E}(t)] - \alpha_L G_{T,L}(t) \end{aligned} \quad (4.125)$$

dove:

- L è lo spessore complessivo della parete [m];
- $T_{ai}(t)$ è la temperatura dell'aria interna, lato della parete a $x = 0$, [K];
- $T_{ae}(t)$ è la temperatura dell'aria esterna, lato della parete a $x = L$, [K];
- $T_{MRL,I}(t)$ è la temperatura media radiante lineare dell'ambiente interno, lato $x = 0$, [K];
- $T_{MRL,E}(t)$ è la temperatura media radiante lineare dell'ambiente esterno, lato $x = L$, [K];
- $h_{cv,0}$ è il coefficiente superficiale di scambio termico convettivo per $x = 0$, [W/(m² K)];
- $h_{cv,L}$ è il coefficiente superficiale di scambio termico convettivo per $x = L$, [W/(m² K)].
- $h_{rd,0}$ è il coefficiente superficiale di scambio termico radiativo per $x = 0$, [W/(m² K)];
- $h_{rd,L}$ è il coefficiente superficiale di scambio termico radiativo per $x = L$, [W/(m² K)].
- α_0 è il coefficiente di assorbimento della radiazione solare per $x = 0$, [-];
- α_{L0} è il coefficiente di assorbimento della radiazione solare per $x = L$, [-].
- $G_{T,0}$ è l'irradianza solare totale sulla superficie della parete per $x = 0$, [W/m²];
- $G_{T,L}$ è l'irradianza solare totale sulla superficie della parete per $x = L$, [W/m²].
- per

$$\bar{G}_T \equiv \frac{1}{P} \int_0^P G_T(t) dt \quad (4.126)$$

$$I_T(t) = G_T(t) - \bar{G}_T \quad (4.127)$$

si ottiene la seguente decomposizione:

$$\begin{aligned} \bar{\varphi}_0 &= -\lambda \frac{d \bar{T}(x)}{dx} \Big|_{x=0} = h_{cv,0}[\bar{T}_{ai} - \bar{T}(0)] + h_{rd,0}[\bar{T}_{MRL,I} - \bar{T}(0)] + \alpha_0 \bar{G}_{T,0} \\ \bar{\varphi}_L &= -\lambda \frac{d \bar{T}(x)}{dx} \Big|_{x=L} = h_{cv,L}[\bar{T}(L) - \bar{T}_{ae}] + h_{rd,L}[\bar{T}(L) - \bar{T}_{MRL,E}] - \alpha_L \bar{G}_{T,L} \end{aligned} \quad (4.128)$$

e

$$\begin{aligned}\varphi_0(t) &= -\lambda \frac{\partial \vartheta(x, t)}{\partial x} \Big|_{x=0} = h_{cv,0}[\vartheta_{ai}(t) - \vartheta(0, t)] + h_{rd,0}[\vartheta_{MRL,I}(t) - \vartheta(0, t)] + \alpha_0 \Gamma_{T,0}(t) \\ \varphi_L(t) &= -\lambda \frac{\partial \vartheta(x, t)}{\partial x} \Big|_{x=L} = h_{cv,L}[\vartheta(L, t) - \vartheta_{ae}(t)] + h_{rd,L}[\vartheta(L, t) - \vartheta_{MRL,E}(t)] - \alpha_L \Gamma_{T,L}(t)\end{aligned}\quad (4.129)$$

La separazione delle condizioni al contorno consente la separazione del problema differenziale in due diversi problemi, uno puramente stazionario e uno puramente periodico. Risolti separatamente, la soluzione generale sarà data dalla sovrapposizione degli effetti, cioè:

$$T(x, t) = \bar{T}(x) + \vartheta(x, t) \quad (4.130)$$

Per la soluzione della componente stazionaria si rinvia al paragrafo 4.7.6, mentre la soluzione per la componente periodica in regime armonico viene riportata nel paragrafo successivo.

4.7.3.1 Soluzione per forzanti armoniche con valore medio nullo

Per la soluzione di un problema differenziale in cui le forzanti siano espresse da funzioni armoniche del tempo,

$$\begin{aligned}\vartheta_{ai}(t) &= \Delta T_{ai} \cos(\omega t + \alpha_{ai}) \\ \vartheta_{ae}(t) &= \Delta T_{ae} \cos(\omega t + \alpha_{ae}) \\ \vartheta_{MRL,I}(t) &= \Delta T_{MRL,I} \cos(\omega t + \alpha_I) \\ \vartheta_{MRL,E}(t) &= \Delta T_{MRL,E} \cos(\omega t + \alpha_E) \\ \Gamma_{T,0}(t) &= \Delta G_{T,0} \cos(\omega t + \beta_0) \\ \Gamma_{T,L}(t) &= \Delta G_{T,L} \cos(\omega t + \beta_L)\end{aligned}\quad (4.131)$$

si utilizza la seguente equivalenza con la notazione complessa esponenziale:

$$A(t) = A_0 \cos(\omega t + \alpha) = \text{Re}[A_0 e^{i\alpha} e^{i\omega t}] = \text{Re}[\tilde{A}_0 e^{i\omega t}] = \text{Re}[\tilde{A}(t)] \quad (4.132)$$

dove $\tilde{A}(t)$ è un vettore rotante in campo complesso con velocità angolare ω .

Le condizioni al contorno si possono riscrivere, in campo complesso, come:

$$\begin{aligned}\tilde{\vartheta}_{ai}(t) &= \tilde{\vartheta}_{ai} e^{i\omega t} \\ \tilde{\vartheta}_{ae}(t) &= \tilde{\vartheta}_{ae} e^{i\omega t} \\ \tilde{\vartheta}_{MRL,I}(t) &= \tilde{\vartheta}_{MRL,I} e^{i\omega t} \\ \tilde{\vartheta}_{MRL,E}(t) &= \tilde{\vartheta}_{MRL,E} e^{i\omega t} \\ \tilde{\Gamma}_{T,0}(t) &= \tilde{G}_{T,0} e^{i\omega t} \\ \tilde{\Gamma}_{T,L}(t) &= \tilde{G}_{T,L} e^{i\omega t}\end{aligned}\quad (4.133)$$

e le incognite del problema, temperature e flussi areici, come:

$$\begin{aligned}\tilde{\vartheta}(x, t) &= \tilde{\vartheta}(x) e^{i\omega t} \\ \tilde{\varphi}(x, t) &= \tilde{\varphi}(x) e^{i\omega t}\end{aligned}\quad (4.134)$$

Posto,

$$\begin{aligned}\tilde{\theta}_0 &= \tilde{\theta}(x=0) & ; & & \tilde{\theta}_L &= \tilde{\theta}(x=L) \\ \tilde{\phi}_0 &= \tilde{\phi}(x=0) & ; & & \tilde{\phi}_L &= \tilde{\phi}(x=L)\end{aligned}\quad (4.135)$$

la soluzione in termini di ampiezze armoniche è determinata tramite la matrice della conduttanza **C**, che coincide con la matrice dell'ammettenza per condizioni al contorno di Dirichlet, cioè:

$$\mathbf{C} = \begin{bmatrix} \tilde{c}_{11} & \tilde{c}_{12} \\ \tilde{c}_{21} & \tilde{c}_{22} \end{bmatrix} = \mathbf{Y}^{(C)} = \frac{1}{\tilde{z}_{12}^{(C)}} \begin{bmatrix} -\tilde{z}_{11}^{(C)} & 1 \\ 1 & -\tilde{z}_{22}^{(C)} \end{bmatrix} \quad \text{con} \quad \tilde{c}_{12} = \tilde{c}_{21} \quad (4.136)$$

dove i termini $\tilde{z}_{ij}^{(C)}$ sono i termini della matrice di trasmissione armonica del mezzo multistrato con condizioni al contorno di temperatura imposta, $\mathbf{Z}_T^{(C)}$, che si calcola a partire dalle matrici di strato come:

$$\mathbf{Z}_T^{(C)} = \prod_{i=N}^1 \mathbf{Z}_i = \prod_{i=N}^1 \begin{bmatrix} \tilde{z}_{11,i} & \tilde{z}_{12,i} \\ \tilde{z}_{21,i} & \tilde{z}_{22,i} \end{bmatrix}^{(C)} = \begin{bmatrix} \tilde{z}_{11}^{(C)} & \tilde{z}_{12}^{(C)} \\ \tilde{z}_{21}^{(C)} & \tilde{z}_{22}^{(C)} \end{bmatrix} \quad (4.137)$$

con

$$\begin{aligned}\tilde{z}_{11,i} &= \tilde{z}_{22,i} = \cosh \tilde{\gamma}_i L_i \\ \tilde{z}_{12,i} &= -\frac{\sinh \tilde{\gamma}_i L_i}{\lambda_i \tilde{\gamma}_i} \\ \tilde{z}_{21,i} &= -\lambda_i \tilde{\gamma}_i \sinh \tilde{\gamma}_i L_i\end{aligned}\quad (4.138)$$

dove

$$\tilde{\gamma}_i = (1+i) \sqrt{\frac{\omega}{2a_i}} = (1+i) \sqrt{\frac{\omega \rho_i c_i}{2\lambda_i}} \quad (4.139)$$

Nota la matrice della conduttanza **C**, questa correla i flussi areici conduttivi con le temperature superficiali tramite la seguente relazione matriciale:

$$\begin{Bmatrix} \tilde{\phi}_0 \\ -\tilde{\phi}_L \end{Bmatrix} = \mathbf{C} \cdot \begin{Bmatrix} \tilde{\theta}_0 \\ \tilde{\theta}_L \end{Bmatrix} = \begin{bmatrix} \tilde{c}_{11} & \tilde{c}_{12} \\ \tilde{c}_{21} & \tilde{c}_{22} \end{bmatrix} \cdot \begin{Bmatrix} \tilde{\theta}_0 \\ \tilde{\theta}_L \end{Bmatrix} \quad (4.140)$$

Combinando la (4.140) con le condizioni al contorno, espresse dalle (4.129), si ottengono le seguenti relazioni che esprimono le ampiezze armoniche temperature superficiali della parete in funzione delle ampiezze armoniche delle forzanti:

$$\begin{aligned}\tilde{\theta}_0 &= \frac{\tilde{c}_{12}}{\tilde{D}} (h_{cv,L} \tilde{\theta}_{ae} + h_{rd,L} \tilde{\theta}_{LMR,L} + \alpha_L \tilde{G}_{T,L}) - \frac{\tilde{c}_{22} + h_{CR,L}}{\tilde{D}} \cdot (h_{cv,0} \tilde{\theta}_{ai} + h_{rd,0} \tilde{\theta}_{LMR,0} + \alpha_0 \tilde{G}_{T,0}) \\ \tilde{\theta}_L &= \frac{\tilde{c}_{12}}{\tilde{D}} (h_{cv,0} \tilde{\theta}_{ai} + h_{rd,0} \tilde{\theta}_{LMR,0} + \alpha_0 \tilde{G}_{T,0}) - \frac{\tilde{c}_{11} + h_{CR,0}}{\tilde{D}} (h_{cv,L} \tilde{\theta}_{ae} + h_{rd,L} \tilde{\theta}_{LMR,L} + \alpha_L \tilde{G}_{T,L})\end{aligned}\quad (4.141)$$

dove

$$\begin{aligned}\tilde{D} &= \tilde{c}_{12}^2 - (\tilde{c}_{11} + h_{CR,0}) \cdot (\tilde{c}_{22} + h_{CR,L}) \\ h_{CR,0} &= h_{cv,0} + h_{rd,0} \\ h_{CR,L} &= h_{cv,L} + h_{rd,L}\end{aligned}\quad (4.142)$$

L'ampiezza armonica dei flussi termici areici, convettivi e radiativi, è poi determinata come:

$$\begin{aligned}
 \tilde{\phi}_{cv,0} &= h_{cv,0}(\tilde{\theta}_{ai} - \tilde{\theta}_0) \\
 \tilde{\phi}_{rd,0} &= h_{rd,0}(\tilde{\theta}_{MRL,I} - \tilde{\theta}_0) \\
 -\tilde{\phi}_{cv,L} &= h_{cv,L}(\tilde{\theta}_{ae} - \tilde{\theta}_L) \\
 -\tilde{\phi}_{rd,L} &= h_{rd,L}(\tilde{\theta}_{LMR,E} - \tilde{\theta}_L)
 \end{aligned}
 \tag{4.143}$$

Note le ampiezze armoniche complesse, la soluzione armonica reale è data da:

$$\begin{aligned}
 \vartheta_0(t) &= \text{Re}[\tilde{\theta}_0 e^{i\omega t}] = \text{Re}[\theta_0 e^{i(\omega t + \alpha_0)}] = \theta_0 \cos(\omega t + \alpha_0) \\
 \vartheta_L(t) &= \text{Re}[\tilde{\theta}_L e^{i\omega t}] = \text{Re}[\theta_L e^{i(\omega t + \alpha_L)}] = \theta_L \cos(\omega t + \alpha_L)
 \end{aligned}
 \tag{4.144}$$

e i flussi areici da

$$\begin{aligned}
 \varphi_{cv,0}(t) &= h_{cv,0}[\Delta T_{ai} \cos(\omega t + \alpha_{ai}) - \theta_0 \cos(\omega t + \alpha_0)] \\
 \varphi_{rd,0}(t) &= h_{rd,0}[\Delta T_{MRL,I} \cos(\omega t + \alpha_I) - \theta_0 \cos(\omega t + \alpha_0)] \\
 \varphi_{cv,L}(t) &= h_{cv,L}[\theta_L \cos(\omega t + \alpha_L) - \Delta T_{ae} \cos(\omega t + \alpha_{ae})] \\
 \varphi_{rd,L}(t) &= h_{rd,L}[\theta_L \cos(\omega t + \alpha_L) - \Delta T_{MRL,E} \cos(\omega t + \alpha_E)]
 \end{aligned}
 \tag{4.145}$$

NOTA: Questo modello è stato implementato nel codice non tanto per eseguire le simulazioni dinamiche, quanto per consentire la determinazione di soluzioni analitiche che vengono poi usate per validare e determinare l'errore di approssimazione introdotto dai modelli numerici, che sono d'impiego più generale.

I limiti di questo modello è quello di essere applicabile solo

- per condizioni al contorno armoniche;
- coefficienti di scambio termico superficiale convettivi e radiativi costanti,
- materiali dello strato della parete con proprietà costanti.

4.7.4 Soluzione analitica approssimata: funzioni di trasferimento per la conduzione termica (CTF)

Il metodo delle funzioni di trasferimento è un metodo in parte analitico e in parte numerico: analitico per quanto attiene alla forma della soluzione, numerico in quanto i coefficienti che vi compaiono sono determinati numericamente in forma approssimata.

La soluzione del problema della conduzione termica monodimensionale in regime variabile descritta dalla (4.122) prevede, secondo il metodo delle CTF i seguenti passaggi:

- decomposizione dei dati rappresentanti le condizioni al contorno in somma di funzioni elementari;
- risoluzione dell'equazione di Fourier mediante l'operatore della Z-trasformata;
- individuazione, in forma matriciale, delle funzioni di trasferimento per la parete piana indefinita OMOGENEA;
- determinazione delle proprietà della parete multistrato attraverso il prodotto delle matrici contenenti le proprietà dei singoli strati (stesso principio della matrice di trasmissione armonica);
- determinazione dei flussi termici alle interfacce in corrispondenza delle forzanti elementari;
- calcolo degli andamenti temporali dei flussi termici alle interfacce in corrispondenza dei dati di ingresso discreti sfruttando la linearità del sistema e quindi la sovrapposizione degli effetti.

4.7.4.1 Rappresentazione dei dati come somma di funzioni elementari

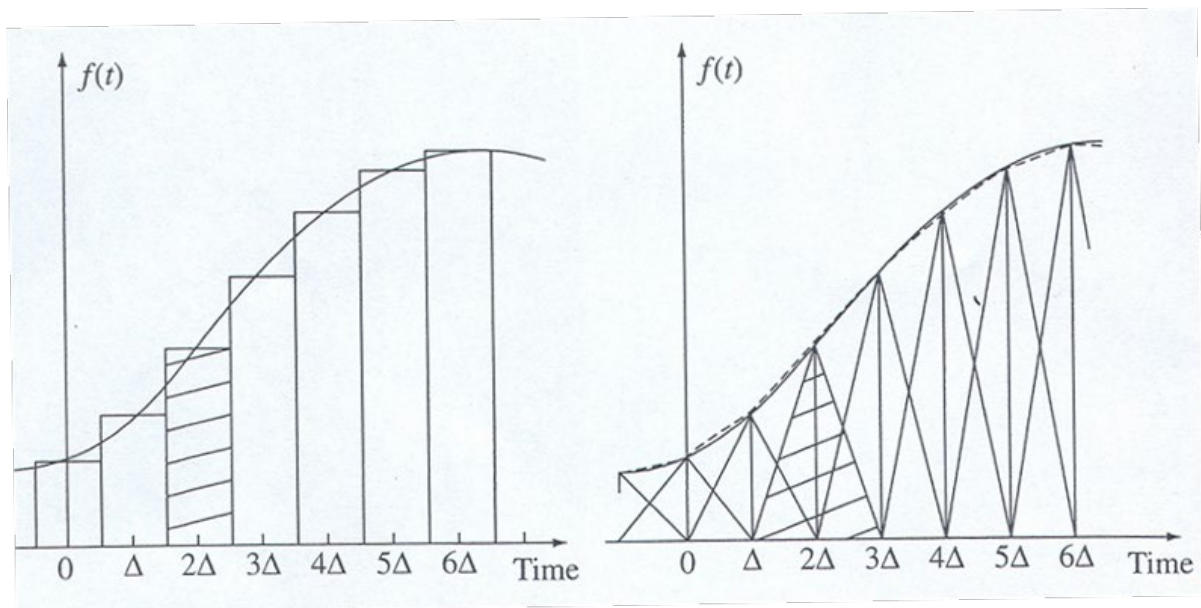


Figura 55 – Approssimazione di una funzione continua del tempo in serie temporale di impulsi elementari: rettangolari, triangolari

Una funzione continua del tempo $f(t)$ può essere APPROSSIMATA da una successione temporale di “impulsi” di forme diverse. In Figura 55 a sinistra la funzione $f(t)$ è approssimata da una successione di impulsi rettangolari tutti di durata Δt ma con altezze diverse paria al valore della funzione per $t + \Delta t/2$ che formano una “funzione a gradini”. In Figura 55 a destra la funzione $f(t)$ è invece approssimata da una successione di impulsi triangolari, tutti di durata $2\Delta t$, ma parzialmente sovrapposti per Δt e il cui vertice corrisponde al valore della funzione per $t + \Delta t$; il risultato è un’approssimazione della funzione con una spezzata costituita da segmenti di retta, cioè i dati (i valori della funzione ad ogni intervallo Δt) vengono interpolati linearmente all’interno dell’intervallo stesso. La qualità della “stima dei valori” della funzione tra i dati campionati dipende quindi dalla scelta della forma dell’impulso.

Per lo sviluppo della CTF è stata scelta l’approssimazione con una successione di impulsi triangolari parzialmente sovrapposti e quindi un’approssimazione con spezzata poligonale. Tale successione temporale di valori prende il nome di **serie temporale** (*time-series*).

È importante poter decomporre la funzione temporale, che descrive le condizioni al contorno del problema conduttivo (ad esempio una temperatura), in una serie temporale di “impulsi” con la stessa base temporale e diversa altezza, perché questo ci consente, sfruttando la **linearità del problema** di sfruttare il **principio di sovrapposizione degli effetti**. Cioè si può determinare la soluzione per una condizione al contorno “elementare” (il singolo impulso) e poi sommare il risultato che ogni soluzione elementare ha prodotto per ottenere il risultato complessivo dovuto al succedersi temporale dei singoli impulsi.

In realtà, non si dispone di funzioni continue delle condizioni al contorno, ma i dati sono discreti nel tempo e quindi già in origine sono a tutti gli effetti serie temporali. La scelta del tipo di impulso da utilizzare per la risoluzione del problema conduttivo è quindi più un problema interpretativo sulla presunta regolarità o meno della reale distribuzione temporale dei dati tra due diversi timestamp, che un’effettiva decomposizione di una funzione analitica nota.

Nel seguito la decomposizione della funzione continua in impulsi elementari che si succedono ad intervalli di tempo costanti Δt , cioè l’insieme dei valori che essa assume per $\Delta t, 2\Delta t, 3\Delta t, \dots, n \cdot \Delta t$, viene indicato come $f(n \cdot \Delta t) \Rightarrow f_n$, dove f_n rappresenta la serie temporale degli “impulsi” o “valori campionati”.

4.7.4.2 Risoluzione dell'equazione di Fourier mediante l'operatore della Z-trasformata

Il metodo delle CTF discende da un altro metodo analitico, quello dei **fattori di risposta** (*Response Factors, RF*) [54, 55], di cui rappresenta la trasposizione in un dominio temporale discreto.

Nella determinazione dei RF si risolve l'equazione monodimensionale della conduzione termica con diffusività costante (omogenea e isotropa) per la singola parete piana indefinita applicando l'operatore integrale $\mathcal{L}[\]$ di Laplace:

$$\mathcal{L}\left[\frac{\partial T(x,t)}{\partial t} - \alpha \frac{\partial^2 T(x,t)}{\partial x^2}\right] = 0 \Rightarrow \frac{\partial^2 \hat{T}(x;s)}{\partial x^2} - \frac{s}{\alpha} \hat{T}(x;s) + T(x,0) = 0 \quad (4.146)$$

dove $\hat{T}(x,s)$ è la funzione trasformata di Laplace di $T(x,t)$ e s è un generico numero complesso parametro della trasformata.

NOTA: l'equazione (4.247) contiene la condizione iniziale $T(x,0)$, che per la determinazione dei RF viene posta **identicamente nulla** (C.I. omogenea).

Considerando condizioni al contorno di Dirichlet, anch'esse trasformate, e condizione iniziale identicamente nulla il problema differenziale diventa ordinario, cioè:

$$\begin{cases} \frac{d^2 \hat{T}(x;s)}{dx^2} - \frac{s}{\alpha} \hat{T}(x;s) = 0 \\ x = 0 \quad \hat{T}(0;s) = \hat{T}_0(s) \\ x = L \quad \hat{T}(L;s) = \hat{T}_L(s) \end{cases} \quad (4.147)$$

che ammette la seguente soluzione:

$$\hat{T}(x;s) = \hat{T}(0;s) \frac{\sinh[L\sqrt{s/a} \cdot (L-x)]}{\sinh L\sqrt{s/a}} + \hat{T}(L;s) \frac{\sinh[L\sqrt{s/a} \cdot x]}{\sinh L\sqrt{s/a}} \quad (4.148)$$

da cui si può derivare la trasformata del flusso termico areico come

$$\hat{\varphi}(x;s) = -\lambda \frac{d\hat{T}(x;s)}{dx} = \lambda \sqrt{\frac{s}{a}} \left\{ \hat{T}(0;s) \frac{\cosh[\sqrt{s/a} \cdot (L-x)]}{\sinh L\sqrt{s/a}} - \hat{T}(L;s) \cdot \frac{\cosh[\sqrt{s/a} \cdot x]}{\sinh L\sqrt{s/a}} \right\} \quad (4.149)$$

L'equazione (4.149) esprime il legame causa-effetto, cioè quali sono i flussi termici scambiati dalla parete in funzione delle temperature imposte alle superfici, da cui, posto

$$\tau = L^2/a \quad \text{tempo caratteristico [s]} \quad (4.150)$$

$$R = L/\lambda \quad \text{resistenza termica areica dello stato [m}^2 \text{ K/W]} \quad (4.151)$$

si ha

$$\begin{aligned} \Rightarrow L\sqrt{s/a} &= \sqrt{L^2 s/a} = \sqrt{\tau s} \\ \Rightarrow \lambda\sqrt{s/a} &= \lambda/L \sqrt{L^2 s/a} = \sqrt{\tau s}/R \end{aligned} \quad (4.152)$$

e, quindi, alle interfacce per $x = 0$ e $x = L$ si ha, per $\hat{T}(0;s) = \hat{T}_0$ e $\hat{T}(L;s) = \hat{T}_L$:

$$\hat{\phi}_0 = \hat{\varphi}(0; s) = \hat{T}_0 \cdot \frac{\sqrt{\tau s} \cosh \sqrt{\tau s}}{R \sinh \sqrt{\tau s}} - \hat{T}_L \cdot \frac{\sqrt{\tau s}}{R \sinh \sqrt{\tau s}} \quad (4.153)$$

$$\hat{\phi}_L = \hat{\varphi}(L; s) = \hat{T}_0 \cdot \frac{\sqrt{\tau s}}{R \sinh \sqrt{\tau s}} - \hat{T}_L \cdot \frac{\sqrt{\tau s} \cosh \sqrt{\tau s}}{R \sinh \sqrt{\tau s}} \quad (4.154)$$



Figura 56 – Quadripolo rappresentante la parete piana omogenea nello spazio della trasformata di Laplace.

Questo legame causa-effetto può essere riscritto in forma matriciale utilizzando lo stesso approccio seguito per il metodo armonico, cioè determinando una matrice di trasferimento nello spazio della trasformata di Laplace $\hat{\mathbf{Z}}$, (Figura 56), definita come:

$$\begin{bmatrix} \hat{T}_L \\ \hat{\phi}_L \end{bmatrix} = \hat{\mathbf{Z}} \cdot \begin{bmatrix} \hat{T}_0 \\ \hat{\phi}_0 \end{bmatrix} = \begin{bmatrix} \hat{z}_{11} & \hat{z}_{12} \\ \hat{z}_{21} & \hat{z}_{22} \end{bmatrix} \cdot \begin{bmatrix} \hat{T}_0 \\ \hat{\phi}_0 \end{bmatrix} \quad (4.155)$$

con

$$\hat{z}_{11} = \hat{z}_{22} = \cosh \sqrt{\tau s} \quad ; \quad \hat{z}_{12} = -\frac{R}{\sqrt{\tau s}} \sinh \sqrt{\tau s} \quad ; \quad \hat{z}_{21} = -\frac{\sqrt{\tau s}}{R} \sinh \sqrt{\tau s} \quad (4.156)$$

La (4.156) consente di determinare la matrice di trasferimento totale della parete multistrato costituita da più strati omogenei con un semplice prodotto matriciale, cioè:

$$\hat{\mathbf{Z}}_T = \prod_{i=N}^1 \hat{\mathbf{Z}}_i = \prod_{i=N}^1 \begin{bmatrix} \hat{z}_{11,i} & \hat{z}_{12,i} \\ \hat{z}_{21,i} & \hat{z}_{22,i} \end{bmatrix}^{(C)} = \begin{bmatrix} \hat{z}_{11}^{(T)} & \hat{z}_{12}^{(T)} \\ \hat{z}_{21}^{(T)} & \hat{z}_{22}^{(T)} \end{bmatrix} \quad (4.157)$$

Per le stesse proprietà delle matrici di trasferimento del caso armonico, $\det[\hat{\mathbf{Z}}_i] = 1 \quad \forall i$, si ha:

$$\hat{z}_{11}^{(T)} \cdot \hat{z}_{22}^{(T)} - \hat{z}_{12}^{(T)} \hat{z}_{21}^{(T)} = 1 \quad (4.158)$$

da cui, esprimendo nuovamente il legame tra temperature e flussi alle interfacce della parete multistrato come legame causa-effetto, si ha:

$$\begin{Bmatrix} \hat{\phi}_0 \\ -\hat{\phi}_L \end{Bmatrix} = \hat{\mathbf{C}} \cdot \begin{Bmatrix} \hat{T}_0 \\ \hat{T}_L \end{Bmatrix} = \begin{bmatrix} \hat{c}_{11} & \hat{c}_{12} \\ \hat{c}_{21} & \hat{c}_{22} \end{bmatrix} \cdot \begin{Bmatrix} \hat{T}_0 \\ \hat{T}_L \end{Bmatrix} \quad (4.159)$$

dove $\hat{\mathbf{C}}$ rappresenta l'equivalente della matrice di conduttanza armonica nello spazio della trasformata di Laplace, con

$$\begin{bmatrix} \hat{c}_{11} & \hat{c}_{12} \\ \hat{c}_{21} & \hat{c}_{22} \end{bmatrix} = \frac{1}{\hat{z}_{12}^{(T)}} \begin{bmatrix} -\hat{z}_{11}^{(T)} & 1 \\ 1 & -\hat{z}_{22}^{(T)} \end{bmatrix} \quad (4.160)$$

Viene quindi introdotto il concetto di **funzione di trasferimento**, $\widehat{TF}$, quale rapporto, nello spazio della trasformata, tra la singola causa (*input*) e la singola risposta a tale causa (*output*), cioè, con riferimento a Figura 57,:

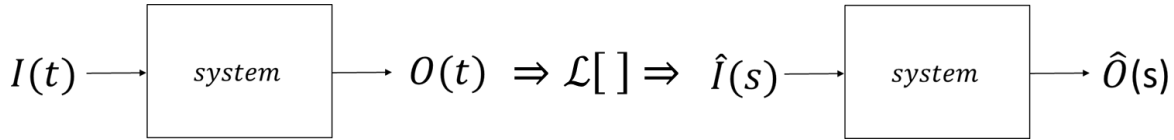
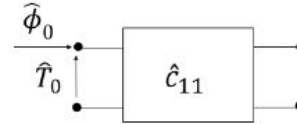


Figura 57 – Funzione di trasferimento nello spazio della trasformata di Laplace.

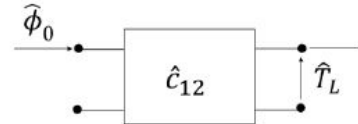
$$\widehat{TF}(s) \equiv \widehat{O}(s)/\widehat{I}(s) \quad (4.161)$$

Dalla (4.159) si desumono le seguenti quattro funzioni di trasferimento:

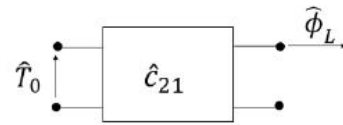
$$\hat{c}_{11} = \frac{\widehat{\phi}_0}{\widehat{T}_0} = \frac{-\hat{z}_{11}^{(T)}}{\hat{z}_{12}^{(T)}} \quad (4.162)$$



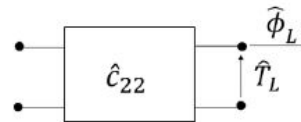
$$\hat{c}_{12} = \frac{\widehat{\phi}_0}{\widehat{T}_L} = \frac{1}{\hat{z}_{12}^{(T)}} \quad (4.163)$$



$$\hat{c}_{21} = \frac{-\widehat{\phi}_L}{\widehat{T}_0} = \frac{1}{\hat{z}_{12}^{(T)}} \quad (4.164)$$



$$\hat{c}_{22} = \frac{-\widehat{\phi}_L}{\widehat{T}_L} = \frac{-\hat{z}_{22}^{(T)}}{\hat{z}_{12}^{(T)}} \quad (4.165)$$



NOTA: le quattro funzioni di trasferimento sono frazionarie e hanno tutte lo stesso denominatore.

Se si è in grado di determinare la singola funzione di trasferimento $\hat{c}_{ij}$ e soprattutto se si è in grado di eseguire l'antitrasformata di Laplace, $\mathcal{L}^{-1}[\]$, della funzione di trasferimento, la convoluzione integrale tra l'antitrasformata e l'input, equazione (4.251), determina il legame funzionale nel tempo tra flusso termico causato e temperatura di parete imposta (Figura 58).

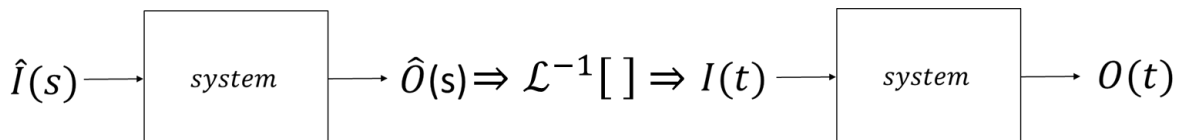


Figura 58 – Convoluzione antitrasformata della funzione di trasferimento e input.

In generale per

$$\hat{c}_{ij} = \frac{\widehat{\phi}_i}{\widehat{T}_j} \Rightarrow \widehat{\phi}_i(s) = \hat{c}_{ij}(s) \cdot \widehat{T}_j \quad (4.166)$$

e per

$$c_{ij}(t) = \mathcal{L}^{-1}[\hat{c}_{ij}(s)] \quad (4.167)$$

si ha:

$$\phi_i(t) = c_{ij}(t) * T_j(t) = \int_0^{\infty} c_{ij}(\tau) * T_j(t - \tau) d\tau \quad (4.168)$$

Mentre è possibile determinare analiticamente la $\hat{c}_{ij}(s)$ del singolo strato omogeneo, ricavare la sua antitrasformata $c_{ij}(t)$, e quindi determinare la soluzione tramite la (4.251) avendo definito la condizione al contorno $T_j(t)$, la determinazione della forma analitica della funzione di trasferimento $\hat{c}_{ij}(s)$ per una parete con un numero di strati maggiore di due è praticamente improponibile. Di conseguenza, i coefficienti della matrice di trasferimento per una parete multistrato, $\hat{z}_{ij}^{(T)}$, possono essere determinati solo in forma numerica, a partire dalla determinazione numerica dei coefficienti della matrice di trasferimento di ogni singolo strato, $\hat{z}_{ij}$. Si lascia al riferimento [54, 55] la spiegazione della metodica implementata per la determinazione di tali coefficienti, che comunque sia per la natura dei dati sia per la scelta della funzione descrivente le condizioni al contorno, "impulso", risultano essere delle successioni numeriche che si autoestinguono al crescere del tempo. La generica risposta flusso termico areico ad un'interfaccia in funzione di una temperatura superficiale imposta come "impulso" sulla stessa o altra interfaccia, viene quindi descritta da una relazione del seguente tipo:

$$\phi_i^{\tau} = \phi_i(\tau \cdot \Delta t) \cong \sum_{k=0}^{N_a} a_k T_j^{(\tau-k)} \quad (4.169)$$

con

ϕ_i^{τ} flusso termico areico all'interfaccia i all'istante temporale discreto τ

τ numero intero definito positivo che rappresenta l'avanzamento temporale a passi di intervalli Δt :
 $t = \tau \cdot \Delta t \quad \tau \in \mathbb{N}$;

a_k coefficienti che rappresentano la antitrasformata della funzione di trasferimento considerata, discreta nel tempo, di magnitudine decrescente con k ;

N_a numero di coefficienti che assicurano un errore accettabile;

$T_j^{(\tau-k)}$ ampiezza dell'impulso di temperatura superficiale imposta all'istante iniziale ($\tau = 0$) sulla superficie i che "scorre" indietro all'avanzare del tempo; $(\tau - k)$ è l'indice del tempo, che identifica la condizione al contorno, che è non nulla al crescere di τ solo per ogni $(\tau - k) = 0$.

Applicando la ricomposizione degli effetti (possibile per la linearità del sistema), se $\{T_j\} = \{T_j^0, T_j^1, T_j^2, \dots, T_j^{\tau}, \dots\}$ è la serie temporale che rappresenta la successione dei dati di temperatura imposta sul contorno j , la risposta sarà data dalla somma delle risposte al singolo impulso espresse dalla (4.169). La forma di tale somma formalmente coincide con la (4.169), essendo i coefficienti identici per ogni impulso imposto ed esistendo ogni impulso applicato per un solo istante temporale discreto: all'istante $\tau = 0$ viene applicato l'impulso di ampiezza T_j^0 che moltiplica il coefficiente a_0 (i valori T_j^{-k} sono tutti nulli), all'istante $\tau = 1$ viene applicato l'impulso di ampiezza T_j^1 che moltiplica il coefficiente a_0 , mentre l'impulso precedente moltiplica solo il coefficiente a_1 , e così via. Quindi la risposta all'applicazione della serie temporale $\{T_j\}$ che costituisce il treno di impulsi è descritta ancora da:

$$\phi_i^{\tau} = \phi_i(\tau \cdot \Delta t) \cong \sum_{k=0}^{N_a} a_k T_j^{(\tau-k)} \quad (4.170)$$

dove però la singola temperatura $T_j^{(\tau-k)}$ rappresenta adesso il valore dell'impulso presente nella serie temporale $\{T_j\}$ al tempo discreto adimensionale τ .

Considerando adesso che il flusso termico areico all'interfaccia i è causato sia dall'applicazione della temperatura alla stessa interfaccia T_i , che all'interfaccia dal lato opposto, T_j , la formula che fornisce il flusso complessivo all'interfacce $x = 0$ e $x = L$ è costituita dalla somma di due equazioni (4.170), cioè:

$$\phi_0^\tau = \phi_0^\tau(T_0) + \phi_0^\tau(T_L) = \sum_{k=0}^{N_a} a_k T_0^{(\tau-k)} - \sum_{k=0}^{N_b} b_k T_L^{(\tau-k)} \quad (4.171)$$

$$\phi_L^\tau = \phi_L^\tau(T_i) + \phi_L^\tau(T_j) = \sum_{k=0}^{N_a} b_k T_0^{(\tau-k)} - \sum_{k=0}^{N_c} c_k T_L^{(\tau-k)} \quad (4.172)$$

dove a_k , b_k e c_k sono i **fattori di risposta**.

NOTA: i coefficienti b_k che compaiono nella (4.216) e nella (4.217) sono gli stessi per la simmetria della matrice delle funzioni di trasferimento (matrice di conduttanza nello spazio di Laplace).

Il calcolo dei flussi termici scambiati dalla parte con gli ambienti circostanti tramite l'impiego dei fattori di risposta non risulta molto efficiente, giacché il numero di coefficienti necessari per avere una buona convergenza delle sommatorie sulle serie originali può essere molto alto (per una parete N_a , N_b e N_c possono superare anche i 20-30 coefficienti).

Il metodo della Z-trasformata, sviluppato da Stephenson e Mitalas [56], è una derivazione del metodo su esposto della trasformata di Laplace e si basa sulla considerazione che i dati disponibili sono serie temporali e non funzioni continue del tempo. Viene quindi introdotto l'operatore $\mathcal{Z}[\]$, che trasforma una serie temporale in una rappresentazione funzionale complessa.

Data una serie temporale, f_n , la Z-trasformata unilaterale è:

$$\hat{F}(z) = \mathcal{Z}[f_n] = \sum_{n=0}^{\infty} f_n \cdot z^{-n} \quad (4.173)$$

dove

z parametro della Z-trasformata, generico numero complesso.

Se adesso si considera la definizione di funzione di trasferimento data dalla (4.161) in termini di Z-trasformata, si ha:

$$\widehat{TF}(z) \equiv \hat{O}(z)/\hat{I}(z) \quad (4.174)$$

con

$$\hat{O}(z) = \sum_{n=0}^{\infty} o_n \cdot z^{-n}$$

$$\hat{I}(z) = \sum_{n=0}^{\infty} i_n \cdot z^{-n}$$

dove o_n e i_n sono le serie temporali rispettivamente dell'output e dell'input.

Se la funzione di trasferimento nel dominio della Z-trasformata è esprimibile come un rapporto caratteristico della natura del sistema, cioè non dipendente dallo specifico input impiegato, cioè:

$$\widehat{TF}(z) = \widehat{N}(z)/\widehat{D}(z) \quad (4.175)$$

con

$$\begin{aligned} \widehat{N}(z) &= \sum_{n=0}^{\infty} \alpha_n \cdot z^{-n} \\ \widehat{D}(z) &= \sum_{n=0}^{\infty} \beta_n \cdot z^{-n} \end{aligned}$$

dove α_n e β_n sono le serie temporali rispettivamente del numeratore e del denominatore, eguagliando le (4.174) e (4.175) si ottiene:

$$\frac{\widehat{N}(z)}{\widehat{D}(z)} = \frac{\widehat{O}(z)}{\widehat{I}(z)} \Rightarrow \widehat{D}(z) \cdot \widehat{O}(z) = \widehat{N}(z) \cdot \widehat{I}(z) \quad (4.176)$$

relazione che riporta and ambo i membri il prodotto di due trasformate. Tale prodotto corrisponde, nel dominio temporale, alla convoluzione integrale per la trasformata di Laplace, e alla convoluzione discreta per la Z-trasformata, che viene spesso indicata come l'analogo nel discreto della trasformata di Laplace nel continuo. La convoluzione discreta tra due serie temporali f_n e g_n è definita come:

$$f_n * g_n = \sum_{m=0}^{\infty} f_m \cdot g_{n-m} \quad (4.177)$$

La grande comodità dell'uso della Z-trasformata deriva proprio dalla forma della convoluzione discreta; infatti per la (4.176) e la (4.177) si ha:

$$\widehat{D}(z) \cdot \widehat{O}(z) = \alpha_n * o_n = \sum_{m=0}^{\infty} \alpha_m \cdot o_{n-m} = \widehat{N}(z) \cdot \widehat{I}(z) = \beta_n * i_n = \sum_{m=0}^{\infty} \beta_m \cdot i_{n-m} \quad (4.178)$$

da cui

$$\alpha_0 \cdot o_n = \sum_{m=0}^{\infty} \beta_m \cdot i_{n-m} - \sum_{m=1}^{\infty} \alpha_m \cdot o_{n-m} \quad (4.179)$$

e quindi

$$o_n = \sum_{m=0}^{\infty} \frac{\beta_m}{\alpha_0} \cdot i_{n-m} - \sum_{m=1}^{\infty} \frac{\alpha_m}{\alpha_0} \cdot o_{n-m} \quad (4.180)$$

si ottiene l'output all'istante n in funzione della serie temporale d'input e della storia pregressa dell'output. La (4.180) rappresenta quella che si chiama **una forma ricorsiva** dell'output.

Posto

$$\frac{\beta_m}{\alpha_0} = a_m \quad ; \quad \frac{\alpha_m}{\alpha_0} = d_m \quad (4.181)$$

e troncando le serie ai termini N_a e N_d per ottenere l'approssimazione desiderata, avendo indicato con k l'indice delle sommatorie, si ottiene infine:

$$o_n \cong \sum_{k=0}^{N_a} a_k \cdot i_{n-m} - \sum_{k=1}^{N_d} d_k \cdot o_{n-k} \quad (4.182)$$

La (4.241) mostra che per determinare l'output in funzione della serie temporale in ingresso, input, non occorre operare l'antitrasformata della funzione di trasferimento Z, ma per la sua proprietà di essere frazionaria, si determina direttamente la forma della convoluzione discreta in funzione dei coefficienti del numeratore e del denominatore della Z-trasformata (unici termini che occorre calcolare per determinare il risultato).

Tornando al problema della conduzione termica monodimensionale in regime variabile e alla definizione delle quattro funzioni di trasferimento introdotte con la trasformata di Laplace, assodato che:

- esiste anche per la Z-trasformata, analoga della trasformata di Laplace nel discreto, una matrice delle funzioni di trasferimento per una parete multistrato;
- le quattro funzioni di trasferimento Z hanno le stesse caratteristiche delle quattro funzioni di trasferimento di Laplace date dalle (4.162), (4.163), (4.164) e (4.165), cioè forma frazionaria con denominatore comune;
- la convoluzione discreta può essere determinata direttamente in funzione dei coefficienti del numeratore e del denominatore delle funzioni di trasferimento;
- sovrapponendo gli effetti per i flussi scambiati alle interfacce, così come già fatto con le (4.171) e (4.172),
- assumendo la superficie a $x = 0$ la superficie rivolta verso l'ambiente esterno, e quella per $x = L$ la superficie rivolta verso l'ambiente interno, anche se la descrizione degli strati della parete è data dall'interno verso l'esterno, e il flusso areico positivo se transita dall'esterno verso l'interno, convenzione adottata dall'ASHRAE Handbook of Fundamentals (1997) [56] e fatta propria nelle implementazioni del metodo da TRNSYS e EnergyPlus;
- rimandano al riferimento [57], per la metodica di determinazione dei coefficienti delle funzioni di trasferimento, basata sull'analogia con la trasformata di Laplace;

la soluzione può essere scritta nella forma ricorsiva rispetto al flusso areico:

$$\phi_{cd,int}^{(\tau)} = \sum_{k=0}^{N_b} b_k \cdot T_{S,ext}^{(\tau-k)} - \sum_{k=0}^{N_c} c_k \cdot T_{S,int}^{(\tau-k)} - \sum_{k=1}^{N_d} d_k \cdot \phi_{cd,int}^{(\tau-k)} \quad (4.183)$$

$$\phi_{cd,ext}^{(\tau)} = \sum_{k=0}^{N_a} a_k \cdot T_{S,ext}^{(\tau-k)} - \sum_{k=0}^{N_b} b_k \cdot T_{S,int}^{(\tau-k)} - \sum_{k=1}^{N_d} d_k \cdot \phi_{cd,ext}^{(\tau-k)} \quad (4.184)$$

dove con $T_{S,ext}^{(\tau)}$ e $T_{S,int}^{(\tau)}$ si intendono le temperature imposte rispettivamente sulle superfici esterne e interne all'istante τ , e con $\phi_{cd,int}^{(\tau)}$ e $\phi_{cd,ext}^{(\tau)}$ i flussi termici areici conduttivi che si hanno all'istante τ rispettivamente alla superficie interna e esterna, positivi se diretti dall'esterno verso l'interno.

Per determinare le temperature superficiali, e quindi i flussi, in funzione delle effettive condizioni al contorno di Robin (temperatura dell'ambiente e flussi radiativi ad onde corte direttamente applicati alle superfici), si aggiungono le seguenti equazioni ausiliarie, derivate dalla congruenza dei flussi areici alle interfacce,

$$T_{S,ext}^{(\tau)} = \frac{(h_I^{(\tau)} + c_0) \cdot K_E^{(\tau)} + b_0 \cdot K_I^{(\tau)}}{(h_I^{(\tau)} + c_0)(h_E^{(\tau)} + a_0) - b_0^2} \quad (4.185)$$

$$T_{S,int}^{(\tau)} = \frac{(h_E^{(\tau)} + a_0) \cdot K_I^{(\tau)} + b_0 \cdot K_E^{(\tau)}}{(h_I^{(\tau)} + c_0)(h_E^{(\tau)} + a_0) - b_0^2} \quad (4.186)$$

con

$$K_E^{(\tau)} \equiv h_E^{(\tau)} \cdot T_{AE}^{*(\tau)} - \left[\sum_{k=1}^{N_a} a_k \cdot T_{S,ext}^{(\tau-k)} - \sum_{k=1}^{N_b} b_k \cdot T_{S,int}^{(\tau-k)} - \sum_{k=1}^{N_d} d_k \cdot \phi_{cd,ext}^{(\tau-k)} \right] \quad (4.187)$$

$$K_I^{(\tau)} \equiv h_I^{(\tau)} \cdot T_{AI}^{*(\tau)} + \left[\sum_{k=1}^{N_b} b_k \cdot T_{S,ext}^{(\tau-k)} - \sum_{k=1}^{N_c} c_k \cdot T_{S,int}^{(\tau-k)} - \sum_{k=1}^{N_d} d_k \cdot \phi_{cd,int}^{(\tau-k)} \right] \quad (4.188)$$

dove

$h_E^{(\tau)}$ coefficiente di scambio termico superficiale misto convettivo-radiativo alla superficie esterna all'istante τ , in $[W/(m^2 K)]$;

$h_I^{(\tau)}$ coefficiente di scambio termico superficiale misto convettivo-radiativo alla superficie interna all'istante τ , in $[W/(m^2 K)]$;

$T_{AE}^{*(\tau)}$ temperatura sole-ambiente all'istante τ riferita all'ambiente esterno e definita come:

$$T_{AE}^{*(\tau)} = T_{AE}^{(\tau)} + \frac{\phi_{ext}^{(SW)(\tau)}}{h_E^{(\tau)}} = \left(\frac{h_{cv,est}^{(\tau)} T_{ae}^{(\tau)} + h_{rd,est}^{(\tau)} T_{LMR,est}^{(\tau)}}{h_E^{(\tau)}} \right) + \frac{\phi_{ext}^{(SW)(\tau)}}{h_E^{(\tau)}}$$

$T_{AI}^{*(\tau)}$ temperatura sole-ambiente all'istante τ riferita all'ambiente interno e definita come:

$$T_{AI}^{*(\tau)} = T_{AI}^{(\tau)} + \frac{\phi_{int}^{(SW)(\tau)}}{h_I^{(\tau)}} = \left(\frac{h_{cv,int}^{(\tau)} T_{ai}^{(\tau)} + h_{rd,int}^{(\tau)} T_{LMR,est}^{(\tau)}}{h_I^{(\tau)}} \right) + \frac{\phi_{int}^{(SW)(\tau)}}{h_I^{(\tau)}}$$

con

$T_{ae}^{(\tau)}$ temperatura dell'aria esterna, in $[^{\circ}C]$;

$T_{ai}^{(\tau)}$ temperatura dell'aria interna, in $[^{\circ}C]$;

$h_{cv,est}^{(\tau)}$ coefficiente di scambio termico superficiale convettivo alla superficie esterna, in $[W/(m^2 K)]$;

$h_{rd,est}^{(\tau)}$ coefficiente di scambio termico superficiale radiativo alla superficie esterna, in $[W/(m^2 K)]$;

$h_{cv,int}^{(\tau)}$ coefficiente di scambio termico superficiale convettivo alla superficie interna, in $[W/(m^2 K)]$;

$h_{rd,int}^{(\tau)}$ coefficiente di scambio termico superficiale radiativo alla superficie interna, in $[W/(m^2 K)]$;

NOTA: l'approccio seguito consente di modificare ad ogni istante temporale i coefficienti scambio termico superficiale, se richiesto.

NOTA: Il vantaggio di utilizzare la forma ricorsiva ottenuta consente, rispetto al metodo dei RF, di ridurre di molto il numero dei coefficienti utilizzati e quindi l'efficienza computazionale del metodo

Le metodiche di calcolo dei coefficienti delle Z-trasformate utilizzate nella soluzione ricorsiva, chiamata Conduction Transfer Function (CTF), possono essere diverse: State-Space method (SS) [58] implementato in EnergyPlus, Direct Root Finding method (DRF), implementato in TRNSYS [59]. A seguito di indagini comparative [60], si è deciso di implementare in OpenBPS il metodo originale [61], così come definito da Mitalas e Arsenault.

4.7.5 Soluzione numerica: differenze finite (DF)

La soluzione del problema della conduzione termica monodimensionale in regime variabile descritta dalla (4.122) prevede, secondo il metodo numerico delle differenze finite (DF) i seguenti passaggi:

- la discretizzazione del dominio spazio-temporale;
- approssimazione dell'equazione differenziale di Fourier con un'equazione alle differenze;
- scelta degli operatori alle differenze: schemi algebrici conseguenti;
- costruzione di un sistema di equazioni algebriche il cui numero è correlato al numero di discretizzazioni spaziali operate;
- risoluzione del sistema di equazioni attraverso i metodi dell'algebra matriciale numerica.

4.7.5.1 Discretizzazione del dominio spazio-temporale

L'approccio alle differenze finite per la soluzione delle equazioni alle derivate parziali si basa sulla discretizzazione del dominio di definizione della funzione incognita, $T(x, t)$, che da continua si trasforma in discreta.

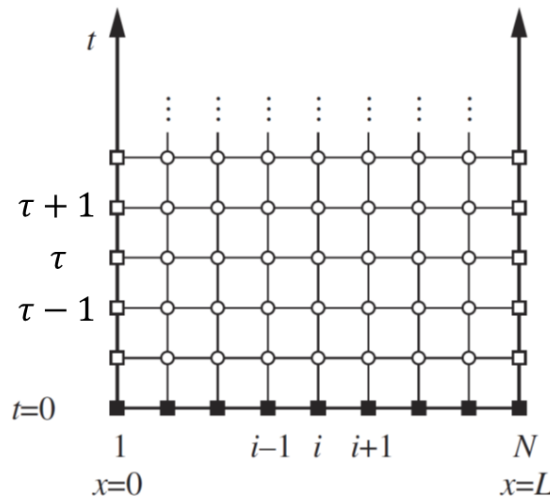


Figura 59 – Discretizzazione del dominio di definizione della funzione incognita .

$$\begin{cases} x = i \cdot \Delta x & i \in \mathbb{N} \\ t = \tau \cdot \Delta t & \tau \in \mathbb{N} \end{cases} \quad T(x, t) \Rightarrow T(i\Delta x, \tau\Delta t) = T_i^\tau \quad (4.189)$$

La funzione incognita è definita in un insieme discreto di punti dello spazio e del tempo, detti nodi spazio-temporali, e quindi anche l'insieme dei valori risulta discreto.

4.7.5.2 Equazione alle differenze

L'equazione differenziale della conduzione termica, monodimensionale omogenea, è descritta dall'operatore differenziale lineare $L_c[]$,

$$L_c[] = \frac{\partial}{\partial t} - a \nabla^2 \quad (4.190)$$

mentre la sua versione alle differenze è $L_d[]$,

$$L_d[] = \frac{\Delta}{\Delta t} - a \Delta^2 \quad (4.191)$$

quindi la forma alle differenze dell'equazione della conduzione monodimensionale è:

$$\frac{\Delta}{\Delta t} T_i^\tau - a \frac{\Delta^2}{\Delta x^2} T_i^\tau \quad T_i^\tau \in D \quad (4.192)$$

dove D è un insieme discreto di valori.

La (4.232) è una forma del tutto generale che va specializzata scegliendo gli operatori alle differenze che vi compaiono: $\Delta^2/\Delta x^2$ e $\Delta/\Delta t$.

Perché la soluzione di tale equazione alle differenze sia sufficientemente corretta e stabile, questa deve soddisfare le tre seguenti condizioni:

- consistenza $\lim_{\Delta x, \Delta t \rightarrow 0} |L_c[T] - L_d[T_i^\tau]| = 0$
- convergenza $\lim_{\Delta x, \Delta t \rightarrow 0} |T(x, t) - T_i^\tau| = 0$
- stabilità $|T(x, t) - T_i^\tau| < \epsilon \quad \forall x, t$

Non esiste un modo univoco per definire i singoli operatori alle differenze, ma tra tutti i modi possibili c'è ne è uno che automaticamente soddisfa la condizione di convergenza: il metodo dell'espansione in serie di Taylor, serie convergente sulla funzione che rappresenta.

4.7.5.3 Scelta degli operatori

L'espansione di Taylor in un'intorno di un punto di una funzione continua consente di stimare il valore della funzione sul bordo di tale intorno in funzione delle sue derivate calcolate nel punto. Ad esempio, considerata la funzione continua $T(x, t)$ in un punto fisso nello spazio, $\bar{x}$, l'espansione nell'intorno di un punto nel tempo, t_0 , può essere effettuata nell'intorno sinistro ($t_0 - \Delta t$), che nell'intorno destro ($t_0 + \Delta t$):

$$T(\bar{x}, t_0 + \Delta t) = T(\bar{x}, t_0) + \sum_{n=1}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial t^n} \right|_{t_0} (\Delta t)^n \quad (4.193)$$

$$T(\bar{x}, t_0 - \Delta t) = T(\bar{x}, t_0) + \sum_{n=1}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial t^n} \right|_{t_0} (-\Delta t)^n \quad (4.194)$$

Dalla (4.193) si ottiene:

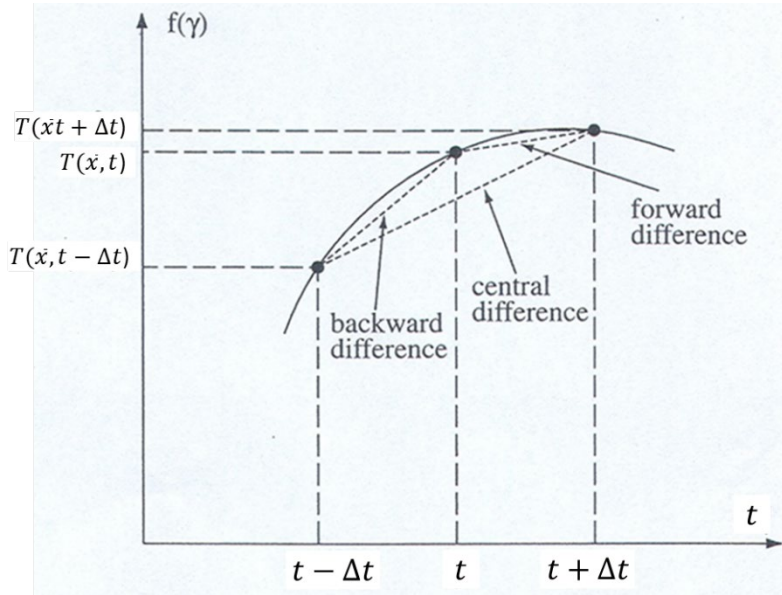


Figura 60 – Approssimazione della funzione continua nell'intorno di un punto con spezzate poligionali

$$\frac{T(\bar{x}, t_0 + \Delta t) - T(\bar{x}, t_0)}{\Delta t} = \left. \frac{\partial T}{\partial t} \right|_{t_0} + \sum_{n=2}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial t^n} \right|_{t_0} (\Delta t)^{n-1} \quad (4.195)$$

Dalla (4.234)

$$\frac{T(\bar{x}, t_0) - T(\bar{x}, t_0 - \Delta t)}{\Delta t} = \left. \frac{\partial T}{\partial t} \right|_{t_0} - \sum_{n=2}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial t^n} \right|_{t_0} (\Delta t)^{n-1} \quad (4.196)$$

E sottraendo dalla (4.193) la (4.234) si ottiene

$$\frac{T(\bar{x}, t_0 + \Delta t) - T(\bar{x}, t_0 - \Delta t)}{2\Delta t} = \left. \frac{\partial T}{\partial t} \right|_{t_0} + \frac{1}{2} \sum_{n=3}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial t^n} \right|_{t_0} (\Delta t)^{n-1} \quad (4.197)$$

Indicato con $\mathcal{O}[\quad]$ l'ordine di approssimazione che si ha troncando la serie, cioè trascurando la sommatoria a secondo membro, e considerando il generico punto t si ha:

$$\frac{\partial T}{\partial t} \cong \frac{T(\bar{x}, t + \Delta t) - T(\bar{x}, t)}{\Delta t} \quad \mathcal{O}[\Delta t] \quad (F.D.) \quad (4.198)$$

$$\frac{\partial T}{\partial t} \cong \frac{T(\bar{x}, t) - T(\bar{x}, t - \Delta t)}{\Delta t} \quad \mathcal{O}[\Delta t] \quad (B.D.) \quad (4.199)$$

$$\frac{\partial T}{\partial t} \cong \frac{T(\bar{x}, t + \Delta t) - T(\bar{x}, t - \Delta t)}{2\Delta t} \quad \mathcal{O}[\Delta t^2] \quad (C.D.) \quad (4.200)$$

da cui si vede che la stessa derivata può essere approssimata in tre diversi modi: differenza in avanti (F.D.), differenza indietro (B.D) e differenza centrale (C.D), di cui l'ultima presenta un migliore ordine di approssimazione.

Se si espande la $T(x, t)$ in un punto fisso nel tempo, $\bar{t}$, nell'intorno di un punto nello spazio, x_0 , rispetto all'intorno sinistro ($x_0 - \Delta x$), e all'intorno destro ($x_0 + \Delta x$), si ottiene:

$$T(x_0 + \Delta x, \bar{t}) = T(x_0, \bar{t}) + \sum_{n=1}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial x^n} \right|_{x_0} (\Delta x)^n \quad (4.201)$$

$$T(x_0 - \Delta x, \bar{t}) = T(x_0, \bar{t}) + \sum_{n=1}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial x^n} \right|_{x_0} (-\Delta x)^n \quad (4.202)$$

da cui si possono riottenere le stesse approssimazioni per le derivate prime riscavate rispetto al tempo, ma anche un'approssimazione per la derivata seconda rispetto allo spazio. Infatti per:

$$T(x_0 + \Delta x, \bar{t}) = T(x_0, \bar{t}) + \left. \frac{\partial T}{\partial x} \right|_{x_0} \Delta x + \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_0} \frac{(\Delta x)^2}{2} + \sum_{n=3}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial x^n} \right|_{x_0} (\Delta x)^n \quad (4.203)$$

$$T(x_0 - \Delta x, \bar{t}) = T(x_0, \bar{t}) - \left. \frac{\partial T}{\partial x} \right|_{x_0} \Delta x + \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_0} \frac{(\Delta x)^2}{2} + \sum_{n=3}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial x^n} \right|_{x_0} (-\Delta x)^n \quad (4.204)$$

sommando le (4.203) e (4.204) la derivata prima si elide e si ottiene:

$$\frac{T(x_0 + \Delta x, \bar{t}) - 2T(x_0, \bar{t}) + T(x_0 - \Delta x, \bar{t})}{\Delta x^2} = \left. \frac{\partial^2 T}{\partial x^2} \right|_{x_0} + \frac{1}{2} \sum_{n=4}^{\infty} \frac{1}{n!} \left. \frac{\partial^n T}{\partial x^n} \right|_{x_0} (\Delta x)^{n-2} \quad (4.205)$$

da cui

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{T(x + \Delta x, \bar{t}) - 2T(x, \bar{t}) + T(x - \Delta x, \bar{t})}{\Delta x^2} \quad \mathcal{O}[\Delta x^2] \quad (C.D.) \quad (4.206)$$

Considerando sviluppi in serie su intorni di ampiezza doppia, $\pm 2\Delta x$, si possono ricavare le seguenti altre approssimazione per la derivata seconda spaziale:

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{T(x + 2\Delta x, \bar{t}) - 2T(x + \Delta x, \bar{t}) + T(x, \bar{t})}{\Delta x^2} \quad \mathcal{O}[\Delta x] \quad (F.D.) \quad (4.207)$$

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{T(x - 2\Delta x, \bar{t}) - 2T(x - \Delta x, \bar{t}) + T(x, \bar{t})}{\Delta x^2} \quad \mathcal{O}[\Delta x] \quad (B.D.) \quad (4.208)$$

che risultano però avere un più basso ordine di approssimazione.

4.7.5.3.1 Selezione dello schema

Dalla (4.192) si possono ottenere diversi schemi di equazioni alle differenze in base alla scelta dell'operatore primo rispetto al tempo e all'operatore secondo rispetto allo spazio, schemi che sono caratterizzati da specifici pro e contra. Gli schemi più usati e conosciuti, ma anche i più elementari, sono il completamente esplicito (F.E., Full Explicit) e il completamente implicito (F.I., Full Implicit).

Lo **schema F.E.** si ottiene scegliendo:

$$\frac{\partial T}{\partial t} \cong \frac{T(\bar{x}, t + \Delta t) - T(\bar{x}, t)}{\Delta t} = \frac{T_i^{\tau+1} - T_i^{\tau}}{\Delta t} \quad \mathcal{O}[\Delta t] \quad (F.D.) \quad (4.209)$$

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{T(x + \Delta x, \bar{t}) - 2T(x, \bar{t}) + T(x - \Delta x, \bar{t})}{\Delta x^2} = \frac{T_{i+1}^{\tau} - 2T_i^{\tau} + T_{i-1}^{\tau}}{\Delta x^2} \quad \mathcal{O}[\Delta x^2] \quad (C.D.)$$

ottenendo per sostituzione nella (4.192)

$$\frac{T_i^{\tau+1} - T_i^{\tau}}{\Delta t} = a \frac{T_{i+1}^{\tau} - 2T_i^{\tau} + T_{i-1}^{\tau}}{\Delta x^2} \quad (4.210)$$

Introducendo il numero locale di Fourier, $Fo = a\Delta t/\Delta x^2$, la (4.210) si può riscrivere come:

$$T_i^{\tau+1} = Fo T_{i+1}^{\tau} + (1 - 2Fo)T_i^{\tau} + Fo T_{i-1}^{\tau} \quad \forall i, \tau \quad (F.E.) \quad (4.211)$$

equazione algebrica nodale che consente di determinare in modo diretto la temperatura della coordinata spaziale adimensionale i nel futuro ($\tau + 1$) in funzione dai valori attuali nello stesso punto e nei nodi limitrofi: $T_i^{\tau}, T_{i+1}^{\tau}$ e T_{i-1}^{τ} .

NOTA: tale schema elementare presenta però una **condizione per la verifica di stabilità** che condiziona la scelta degli intervalli di discretizzazione: **$Fo < 0,5$** .

L'altro schema elementare che si può ricavare è lo **schema F.I.** che si ottiene scegliendo:

$$\frac{\partial T}{\partial t} \cong \frac{T(\bar{x}, t) - T(\bar{x}, t - \Delta t)}{\Delta t} = \frac{T_i^{\tau} - T_i^{\tau-1}}{\Delta t} \quad \mathcal{O}[\Delta t] \quad (B.D.) \quad (4.212)$$

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{T(x + \Delta x, \bar{t}) - 2T(x, \bar{t}) + T(x - \Delta x, \bar{t})}{\Delta x^2} = \frac{T_{i+1}^{\tau} - 2T_i^{\tau} + T_{i-1}^{\tau}}{\Delta x^2} \quad \mathcal{O}[\Delta x^2] \quad (C.D.)$$

ottenendo per sostituzione nella (4.192)

$$\frac{T_i^{\tau} - T_i^{\tau-1}}{\Delta t} = a \frac{T_{i+1}^{\tau} - 2T_i^{\tau} + T_{i-1}^{\tau}}{\Delta x^2} \quad (4.213)$$

Introducendo il numero locale di Fourier, $Fo = a\Delta t/\Delta x^2$, la (4.213) si può riscrivere, traslando il tempo adimensionale di una unità, come:

$$T_i^{\tau} = -Fo T_{i+1}^{\tau+1} + (1 + 2Fo)T_i^{\tau+1} - Fo T_{i-1}^{\tau+1} \quad (F.I.) \quad (4.214)$$

equazione algebrica nodale che non consente di determinare in modo diretto la temperatura della coordinata spaziale adimensionale i nel futuro ($\tau + 1$), in quanto compaiono, come incognite, anche le temperature dei nodi limitrofi e il solo dato disponibile è il valore attuale nel punto i : T_i^{τ} .

NOTA: tale schema elementare non presenta alcuna condizione per la verifica di stabilità e viene detto **incondizionatamente stabile**.

Come mostrato si possono costruire con il metodo indicato diversi schemi alle differenze finite, ognuno con caratteristiche differenti. Il metodo adottato in OpenBPS, considerato che si vuole costruire uno strumento che sia il più flessibile e accurato possibile, è stato quello che viene indicato con il nome di “Metodo Theta”.

4.7.5.3.2 Metodo Theta

Il metodo θ è uno schema numerico alle differenze finite molto utilizzato che corrisponde ad una media pesata dei due schemi precedentemente illustrati, il F.E. a il F.I., rispetto alla determinazione temporale delle differenze seconde nello spazio:

$$\frac{T_i^{\tau+1} - T_i^{\tau}}{\Delta t} = a\theta \frac{T_{i+1}^{\tau+1} - 2T_i^{\tau+1} + T_{i-1}^{\tau+1}}{\Delta x^2} + a(1-\theta) \frac{T_{i+1}^{\tau} - 2T_i^{\tau} + T_{i-1}^{\tau}}{\Delta x^2} \quad (4.215)$$

dove $0 \leq \theta \leq 1$, rappresenta il fattore di pesatura ($\theta = 0$, schema F.E.; $\theta = 1$, schema F.I.), che, introducendo il numero locale di Fourier, $Fo = a\Delta t/\Delta x^2$, può essere riscritta come:

$$T_i^{\tau+1} - T_i^{\tau} = \theta \cdot Fo \cdot (T_{i+1}^{\tau+1} - 2T_i^{\tau+1} + T_{i-1}^{\tau+1}) + (1-\theta) \cdot Fo \cdot (T_{i+1}^{\tau} - 2T_i^{\tau} + T_{i-1}^{\tau}) \quad (4.216)$$

o anche, separando le incognite dai dati, come

$$\begin{aligned} -\theta \cdot Fo \cdot T_{i-1}^{\tau+1} + (1 + 2\theta \cdot Fo) \cdot T_i^{\tau+1} - \theta \cdot Fo \cdot T_{i+1}^{\tau+1} \\ = (1-\theta)Fo \cdot T_{i-1}^{\tau} + (1 - 2(1-\theta)Fo) \cdot T_i^{\tau} + (1-\theta)Fo \cdot T_{i+1}^{\tau} \end{aligned} \quad (4.217)$$

Come si può notare lo schema theta consente di passare con la semplice definizione del calore del parametro θ da uno schema ad un altro e a l'insieme infinito di combinazioni lineari degli stessi, compreso il noto schema Crank-Nicolson, che si ottiene per $\theta = 1/2$.

4.7.5.4 Sistema di equazioni algebriche

Per determinare la soluzione del problema conduttivo occorre applicare l'equazione algebrica alle differenze ad ogni nodo spaziale i con cui si è discretizzato il dominio spaziale $]0, L[$, cioè per gli N nodi tale per cui, dato L lo spessore della parete piana omogenea (Figura 61), risulti:

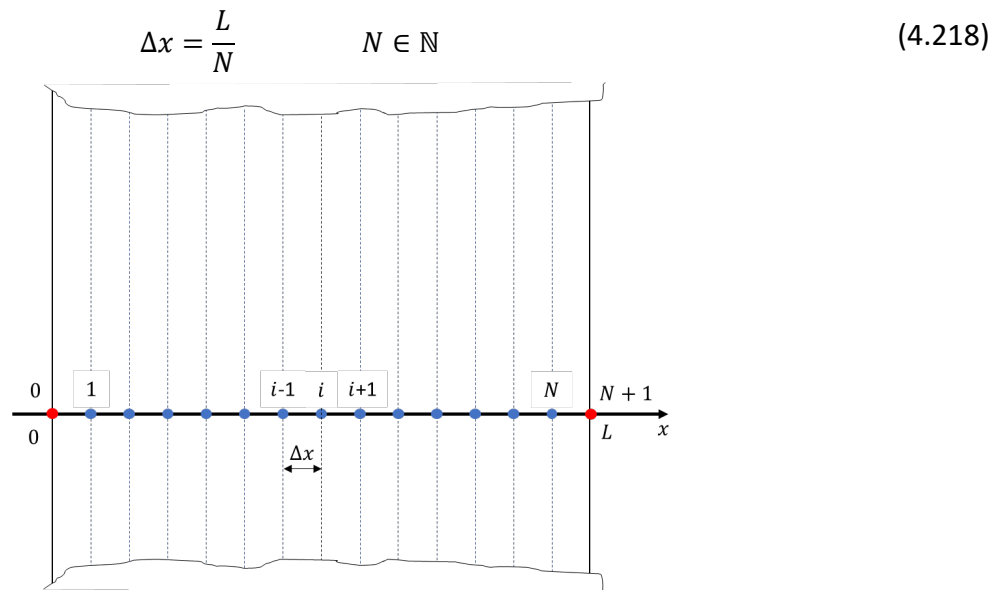


Figura 61 – Discretizzazione spaziale sovrapposta alla parete di spessore L : nodi spaziali.

Come si vede da Figura 61, il numero di nodi interni al dominio di definizione della funzione incognita sono N contati a partire da 1, e N è il numero di partizioni spaziali corrispondenti nello spessore L della parete, mentre il numero complessivo di nodi è $(N + 2)$, dato che bisogna aggiungerli quelli disposti sulle frontiere del dominio.

Siccome l'equazione differenziale (4.122) è definita solo all'interno del dominio spaziale $]0, L[$, anche la sua approssimazione alle differenze finite (4.192), indipendentemente dallo schema adottato, è definita nei nodi da $i = 1$ a $i = N$. Di conseguenza l'equazione differenziale è approssimata da un sistema di N equazioni alle differenze, che può essere efficacemente espresso in forma matriciale per ognuno dei vari schemi esaminati.

Per lo **schema F.E.** è

$$\begin{bmatrix} T_1 \\ T_i \\ T_N \end{bmatrix}^{\tau+1} = \begin{bmatrix} 1-2Fo & Fo & & & \\ Fo & 1-2Fo & Fo & & \\ & Fo & 1-2Fo & Fo & \\ & & Fo & 1-2Fo & Fo \\ & & & Fo & 1-2Fo \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_i \\ T_N \end{bmatrix}^{\tau} \quad (4.219)$$

la cui soluzione è DIRETTA:

$$\mathbf{T}^{\tau+1} = \mathbf{A}_E \cdot \mathbf{T}^{\tau} = \quad (4.220)$$

indicati con $\mathbf{T}$ il vettore delle temperature nodali e con $\mathbf{A}_E$ la matrice tridiagonale simmetrica dei coefficienti.

Per lo **schema F.I.** è

$$\begin{bmatrix} T_1 \\ T_i \\ T_N \end{bmatrix}^{\tau} = \begin{bmatrix} 1+2Fo & -Fo & & & \\ -Fo & 1+2Fo & -Fo & & \\ & -Fo & 1+2Fo & -Fo & \\ & & -Fo & 1+2Fo & -Fo \\ & & & -Fo & 1+2Fo \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_i \\ T_N \end{bmatrix}^{\tau+1} \quad (4.221)$$

la cui soluzione è INDIRETTA, dato che occorre procedere prima all'inversione della matrice tridiagonale simmetrica dei coefficienti $\mathbf{A}_I$ per poter risolvere il sistema:

$$\mathbf{T}^{\tau} = \mathbf{A}_I \cdot \mathbf{T}^{\tau+1} \Rightarrow \mathbf{T}^{\tau+1} = \mathbf{A}_I^{-1} \cdot \mathbf{T}^{\tau} \quad (4.222)$$

NOTA: è l'inversione della matrice dei coefficienti che rende incondizionatamente stabile lo schema completamente implicito.

Per lo schema del **Metodo Theta** è:

$$\begin{bmatrix} 1+2\theta Fo & -\theta Fo & & \\ -\theta Fo & 1+2\theta Fo & -\theta Fo & \\ & -\theta Fo & 1+2\theta Fo & -\theta Fo \\ & & -\theta Fo & 1+2\theta Fo \\ & & & -\theta Fo & 1+2\theta Fo \end{bmatrix} \cdot \begin{bmatrix} T_1^{\tau+1} \\ T_i \\ T_N \end{bmatrix} = \begin{bmatrix} 1-2(1-\theta)Fo & (1-\theta)Fo & & \\ (1-\theta)Fo & 1-2(1-\theta)Fo & (1-\theta)Fo & \\ & (1-\theta)Fo & 1-2(1-\theta)Fo & (1-\theta)Fo \\ & & (1-\theta)Fo & 1-2(1-\theta)Fo & (1-\theta)Fo \\ & & & (1-\theta)Fo & 1-2(1-\theta)Fo \end{bmatrix} \cdot \begin{bmatrix} T_1^{\tau} \\ T_i \\ T_N \end{bmatrix} \quad (4.223)$$

che, indicata con $\mathbf{A}_L$ la matrice a primo membro e con $\mathbf{A}_R$ la matrice a secondo membro, può essere riscritta in forma compatta come:

$$\mathbf{A}_L \cdot \mathbf{T}^{\tau+1} = \mathbf{A}_R \cdot \mathbf{T}^{\tau} \quad (4.224)$$

Da cui la soluzione del sistema si ottiene per inversione della matrice $\mathbf{A}_L$ e suo prodotto con la matrice $\mathbf{A}_R$, cioè:

$$\mathbf{T}^{\tau+1} = (\mathbf{A}_L^{-1} \cdot \mathbf{A}_R) \cdot \mathbf{T}^{\tau} \Rightarrow \mathbf{T}^{\tau+1} = \mathbf{A}_{LR} \cdot \mathbf{T}^{\tau} \quad (4.225)$$

Come però si è già detto tali equazioni matriciali rappresentano solo l'approssimazione dell'equazione differenziale e non il problema differenziale, per la cui soluzione occorre aggiungere le condizioni al contorno, cioè derivare le equazioni alle differenze che descrivono le condizioni al contorno nei nodi sul contorno, cioè $i = 0$ a $i = N + 1$ (vedasi Figura 61).

4.7.5.5 Pareti multistrato

Nei paragrafi precedenti è stata trattata la conduzione termica in regime variabile per una parete piana indefinita omogenea con il metodo delle differenze finite. Se si vuole risolvere lo stesso problema per una parete piana composta da più strati omogenei (parete multistrato) si può ancora applicare quanto visto, ma con qualche complicazione in più.

Per una parete composta da N strati omogenei (nell'esempio di Figura 62 sono tre), si può scrivere per ognuno dei singoli strati l'equazione matriciale che approssima l'equazione differenziale all'interno dei singoli domini spaziali (nell'esempio $]0_1 L_1[$, $]0_2 L_2[$, $]0_3 L_3[$), cioè:

$$\mathbf{T}_k^{\tau+1} = \mathbf{A}_k \cdot \mathbf{T}_k^{\tau} \quad \forall k \quad (4.226)$$

dove k è l'indice dello strato ($k = 1 \div N$) e $\mathbf{A}_k$ è la matrice dei coefficienti di strato che dipenderanno dal metodo scelto (F.I., F.E., Theta).

A queste N equazioni matriciali occorre aggiungere le equazioni alle differenze che esplicitano le condizioni al contorno per ciascuno dei singoli domini spaziali, due delle quali rappresentano le condizioni al contorno del problema differenziale relativo all'intera parete multistrato. Il numero di tali equazioni ausiliarie è apparentemente $2N$, in realtà sono $2 + (N - 1)$, in quanto quelle relative ai "contorni" interni sono condivise tra i due strati adiacenti: *condizioni di congruenza dei flussi all'interfaccia*.

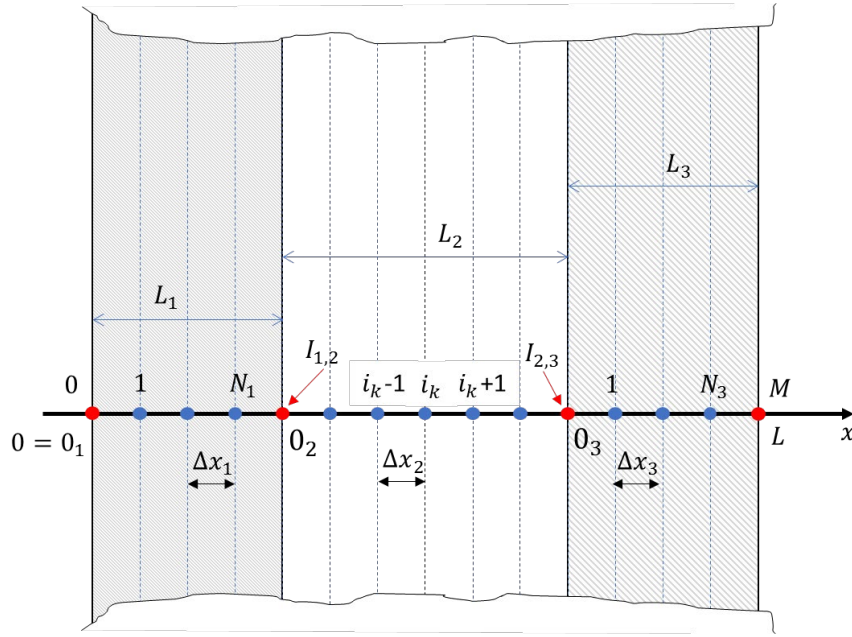


Figura 62 – Parete multistrato composta da tre strati omogenei: coordinate dei nodi spaziali i_1 , i_2 e i_3 .

Indicato con N_k il numero di nodi spaziali i_k interni allo strato k , con 0 e M i nodi sulle superfici sinistra e destra della parete multistrato, con $I_{k,k+1}$ i nodi all'interfaccia tra due strati, la forma generale del sistema da risolvere è data da:

$$\mathbf{T}_k^{\tau+1} = \mathbf{A}_k \cdot \mathbf{T}_k^{\tau} \quad k = 1 \div N \quad (\text{equazioni matriciali di strato})$$

$$\Phi_k^{\tau+1} = \Phi_{k+1}^{\tau+1} \quad k = 1 \div N - 1 \quad (\text{equazioni di congruenza alle interfacce}) \quad (4.227)$$

$$\Phi_0^{\tau+1} = f_0^{\tau+1} \quad ; \quad \Phi_M^{\tau+1} = f_M^{\tau+1} \quad (\text{condizioni al contorno})$$

con

$\Phi_k^{\tau+1}$ flussi termici scambiati all'interfaccia tra lo strato k e lo strato $k + 1$;

$\Phi_0^{\tau+1}$ flusso termico scambiato al contorno sinistro ($x = 0$);

$\Phi_L^{\tau+1}$ flusso termico scambiato al contorno destro ($x = L$);

$f_0^{\tau+1}$ funzione generalizzata che include le tre possibili condizioni al contorno in $x = 0$;

$f_M^{\tau+1}$ funzione generalizzata che include le tre possibili condizioni al contorno in $x = L$.

Le equazioni di congruenza e le condizioni al contorno vengono esplicitate, in funzione dello schema adottato, in equazioni per la determinazione delle variabili non incluse nelle equazioni matriciali presenti nella (4.227), le temperature dei nodi d'interfaccia. In forma generale, alle interfacce, intere ed esterne, si ha quindi:

$$\begin{aligned} T_{I_k}^{\tau+1} &= f_k(T_{I_k}^{\tau}, T_{k,i_k}^{\tau}, T_{k+1,i_{k+1}}^{\tau}, T_{k,i_k}^{\tau+1}, T_{k+1,i_{k+1}}^{\tau+1}) \quad k = 1 \div N - 1 \\ T_0^{\tau+1} &= f_0(T_0^{\tau}, T_{A_0}^{\tau}, T_{1,i_1}^{\tau}, T_{A_0}^{\tau+1}, T_{1,i_1}^{\tau+1}) \\ T_L^{\tau+1} &= f_L(T_M^{\tau}, T_{A_L}^{\tau}, T_{N_N,i_N}^{\tau}, T_{A_L}^{\tau+1}, T_{N_N,i_N}^{\tau+1}) \end{aligned} \quad (4.228)$$

È quindi possibile ricondurre la soluzione del problema conduttivo modellato con le differenze finite alla soluzione di un unico sistema lineare rappresentato da un'unica equazione matriciale, che comprende l'insieme delle equazioni matriciali di strato (4.226), le equazioni di congruenza alle interfacce interne e le condizioni al contorno, equazioni (4.228), nella forma generale:

$$\mathbf{T}^{\tau+1} = \mathbf{A}_{LR} \cdot \mathbf{T}^{\tau} + \mathbf{A}_L^{-1} \cdot \mathbf{B}^{\tau} + \mathbf{A}_L \cdot \mathbf{B}^{\tau+1} \quad (4.229)$$

dove

$\mathbf{B}^{\tau}$ vettore delle condizioni al contorno all'istante τ ;

$\mathbf{B}^{\tau+1}$ vettore delle condizioni al contorno all'istante $\tau + 1$.

4.7.5.5.1 Singolarità dello strato puramente resistivo

Se uno degli strati della parete viene dichiarato puramente resistivo, come ad esempio un'intercapedine d'aria, ponendo la capacità termica specifica per unità di massa $c_k = 0$, la diffusività termica dello strato k risulta infinita, $\alpha_k \rightarrow \infty$, e di conseguenza anche il numero di Fourier tale strato diventa singolare, $Fo_k \rightarrow \infty$, facendo "esplodere" numericamente lo schema, qualsiasi esso sia.

Per ovviare a tale condizione, se lo strato k viene dichiarato puramente resistivo ($c_k = 0$), lo schema numerico di calcolo viene modificato di conseguenza come segue:

- si considera un solo nodo interno allo strato, 1_k , e quindi un solo valore di temperatura, T_{1k} ;
- a sinistra e a destra di tale nodo esistono solo due resistenze equivalenti conduttive:
 - se è un'intercapedine d'aria: $c_k = 0$, $\rho_k = 0$, $\lambda_k = h_{int}L_k$, $\Delta x_k = L_k/2$;
 - se invece è un materiale solido: $c_k = 0$, $\rho_k = 0$, $\lambda_k = \lambda_{mat}$, $\Delta x_k = L_k/2$

da cui esiste un'unica equazione nodale per lo strato data dalla congruenza del flusso conduttivo equivalente, cioè:

$$-\lambda_k \frac{T_{I_{k-1}}^{\tau+1} - T_{1k}^{\tau+1}}{\Delta x_k} = +\lambda_k \frac{T_{I_{k+1}}^{\tau+1} - T_{1k}^{\tau+1}}{\Delta x_k} \quad (4.230)$$

da cui

$$T_{1k}^{\tau+1} = (T_{I_{k-1}}^{\tau+1} + T_{I_{k+1}}^{\tau+1})/2 \quad (4.231)$$

4.7.5.6 Nodi sul contorno

Di seguito vengono riportate le forme che assumono le equazioni (4.228) che descrivono le condizioni al contorno in funzione sia della tipologia delle stesse sia dei possibili diversi metodi di approssimazione.

4.7.5.6.1 Discretizzazione spaziale del 1° Ordine

Lato 0 – Condizioni al contorno di Robin

$$-\lambda_1 \cdot \frac{\partial T(x, t)}{\partial x} \Big|_{x=0} = h_0(t) \cdot (T_{A_0}(t) - T(0, t)) + \phi_0(t) \quad (4.232)$$

La forma alle differenze finite è

$$-\lambda_1 \cdot \frac{T_1^{\tau+1} - T_0^{\tau+1}}{\Delta x} = h_0^{\tau+1} \cdot (T_{A_0}^{\tau+1} - T_0^{\tau+1}) + \phi_0^{\tau+1} \quad (4.233)$$

da cui

$$\left[\frac{\lambda_1}{\Delta x_1} + h_0^{\tau+1} \right] \cdot T_0^{\tau+1} - \frac{\lambda_1}{\Delta x_1} T_1^{\tau+1} = h_0^{\tau+1} \cdot T_{A_0}^{\tau+1} + \phi_0^{\tau+1} \quad (4.234)$$

Lato 0 – Condizioni al contorno di Neumann

$$-\lambda_1 \cdot \frac{\partial T(x, t)}{\partial x} \Big|_{x=0} = \phi_{A_0}(t) \quad (4.235)$$

La forma alle differenze finite è

$$-\lambda_1 \cdot \frac{T_1^{\tau+1} - T_0^{\tau+1}}{\Delta x_1} = \phi_0^{\tau+1} \quad (4.236)$$

da cui

$$\left[\frac{\lambda_1}{\Delta x_1} \right] \cdot T_0^{\tau+1} - \frac{\lambda_1}{\Delta x_1} T_1^{\tau+1} = \phi_0^{\tau+1} \quad (4.237)$$

Lato 0 – Condizioni al contorno di Dirichlet

$$T_0^{\tau+1} = \bar{T}_0^{\tau+1} \quad (4.238)$$

Lato L – Condizioni al contorno Robin

$$\lambda_N \cdot \frac{\partial T(x, t)}{\partial x} \Big|_{x=L} = h_L(t) \cdot (T_{A_L}(t) - T(L, t)) + \phi_L(t) \quad (4.239)$$

La forma alle differenze finite è

$$\lambda_N \cdot \frac{T_M^{\tau+1} - T_{M-1}^{\tau+1}}{\Delta x_N} = h_L^{\tau+1} \cdot (T_{A_L}^{\tau+1} - T_M^{\tau+1}) + \phi_L^{\tau+1} \quad (4.240)$$

da cui

$$-\frac{\lambda_N}{\Delta x_N} T_{M-1}^{\tau+1} + \left[\frac{\lambda_N}{\Delta x_N} + h_L^{\tau+1} \right] \cdot T_M^{\tau+1} = h_L^{\tau+1} \cdot T_{A_L}^{\tau+1} + \phi_L^{\tau+1} \quad (4.241)$$

Lato L – Condizioni al contorno di Neumann

$$\lambda_N \cdot \frac{\partial T(x, t)}{\partial x} \Big|_{x=L} = \phi_L(t) \quad (4.242)$$

La forma alle differenze finite è

$$\lambda_N \cdot \frac{T_M^{\tau+1} - T_{M-1}^{\tau+1}}{\Delta x_N} = \phi_L^{\tau+1} \quad (4.243)$$

da cui

$$-\frac{\lambda_N}{\Delta x_N} T_{M-1}^{\tau+1} + \left[\frac{\lambda_N}{\Delta x_N} \right] \cdot T_M^{\tau+1} = \phi_L^{\tau+1} \quad (4.244)$$

Lato L – Condizioni al contorno di Dirichlet

$$T_L^{\tau+1} = \bar{T}_L^{\tau+1} \quad (4.245)$$

NOTA: le equazioni risultanti sono “sincrone” e non “evolutive”; ciò vuol dire che l’incognita si determina in base ai dati che sono sincroni nel tempo (non ci sono effetti capacitivi e quindi memoria temporale). Questo introduce un errore di approssimazione, giacché corrisponde a ignorare la capacità termica corrispondente a due strati superficiali di spessore $\Delta x/2$. La sincronicità fa anche sì che le condizioni al contorno così espresse abbiano la stessa forma per i tre schemi citati (F.E, F.I. Theta).

4.7.5.6.2 Discretizzazione spaziale col volume finito

In questo caso non si approssima la derivata parziale presente nella condizione al contorno con una differenza finita, ma si isola una porzione di parete contenente il contorno pari a $\Delta x/2$, area tratteggiata in Figura 63, per la quale si scrive un’equazione di bilancio energetico in forma discreta, cioè approssimando le derivate che compaiono nel bilancio con delle differenze finite. Da un punto di vista formale scrive un’equazione integrale del tipo:

$$\rho c \frac{\Delta x}{2} \frac{d\langle T \rangle_V}{dt} = h_0 (T_{A_0} - \langle T \rangle_V) + \phi_0 + \lambda \cdot \left. \left\langle \frac{\partial T}{\partial x} \right\rangle_s \right|_{x=\frac{\Delta x}{2}} \quad (4.246)$$

dove del “volume” di spessore Δx solo $\Delta x/2$ è dotato di capacità termica (il solido) e i flussi areici, convettivo-radiativo e imposto, ϕ_0 , sono scambiati alla superficie per $x = 0$, mentre il flusso conduttivo è scambiato alla superficie fittizia per $x = \Delta x/2$.

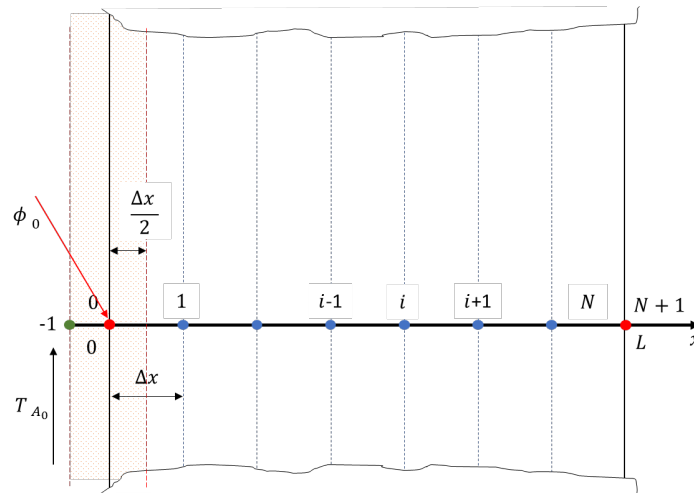


Figura 63 – Bilancio energetico su volume finito fittizio all’interfaccia: -1 = nodo temperatura ambiente.

Lato 0 – Condizioni al contorno di Robin (Theta method)

$$\begin{aligned}
 & \rho_1 c_1 \frac{\Delta x_1}{2} \cdot \frac{T_0^{\tau+1} - T_0^{\tau}}{\Delta t} = \\
 & = \theta [h_0^{\tau+1} \cdot (T_{A_0}^{\tau+1} - T_0^{\tau+1}) + \phi_0^{\tau+1}] + (1 - \theta) [h_0^{\tau} \cdot (T_{A_0}^{\tau} - T_0^{\tau}) + \phi_0^{\tau}] \\
 & + \lambda_1 \theta \frac{T_1^{\tau+1} - T_0^{\tau+1}}{\Delta x_1} + \lambda_1 (1 - \theta) \frac{T_1^{\tau} - T_0^{\tau}}{\Delta x_1}
 \end{aligned} \tag{4.247}$$

e raccogliendo rispetto alle variabili incognite:

$$\begin{aligned}
 & \left[\left(\rho_1 c_1 \frac{\Delta x_1}{2 \Delta t} + \theta \left(\frac{\lambda_1}{\Delta x_1} + h_0^{\tau+1} \right) \right) T_0^{\tau+1} - \theta \left(\frac{\lambda_1}{\Delta x_1} \right) T_1^{\tau+1} \right. \\
 & = \left[\left(\rho_1 c_1 \frac{\Delta x_1}{2 \Delta \tau} - (1 - \theta) \left(\frac{\lambda_1}{\Delta x_1} + h_0^{\tau} \right) \right) T_0^{\tau} + (1 - \theta) \left(\frac{\lambda_1}{\Delta x_1} \right) T_1^{\tau} \right. \\
 & \quad \left. + \theta (h_0^{\tau+1} T_{A_0}^{\tau+1} + \phi_0^{\tau+1}) + (1 - \theta) (h_0^{\tau} T_{A_0}^{\tau} + \phi_0^{\tau}) \right]
 \end{aligned} \tag{4.248}$$

E introducendo in numero di Fourier e di Biot:

$$Fo_1 = a_1 \Delta t / \Delta x_1^2 = \lambda_1 \Delta t / (\rho_1 c_1 \Delta x_1^2) \quad ; \quad Bi_0 = h_0 \Delta x_1 / \lambda_1$$

si ha:

$$\begin{aligned}
 & [1 + 2Fo_1 \theta (1 + Bi_0^{\tau+1})] \cdot T_0^{\tau+1} - 2Fo_1 \theta \cdot T_1^{\tau+1} \\
 & = [(1 - 2Fo_1 (1 - \theta) (1 + Bi_0^{\tau})) \cdot T_0^{\tau} + 2Fo_1 (1 - \theta) \cdot T_1^{\tau} \\
 & \quad + \theta 2Fo_1 Bi_0^{\tau+1} (T_{A_0}^{\tau+1} + \phi_0^{\tau+1} / h_0^{\tau+1}) + (1 - \theta) 2Fo_1 Bi_0^{\tau} (T_{A_0}^{\tau} + \phi_0^{\tau} / h_0^{\tau})]
 \end{aligned} \tag{4.249}$$

Lato L – Condizioni al contorno di Robin (Theta method)

Analogamente per l'altra superficie essendo N l'indice dell'ultimo strato e introducendo il numero di Fourier e di Biot:

si ha:

$$Fo_N = a_N \Delta t / \Delta x_N^2 = \lambda_N \Delta t / (\rho_N c_N \Delta x_N^2) \quad ; \quad Bi_L = h_L \Delta x_N / \lambda_N$$

$$\begin{aligned}
 & [1 + 2Fo_N \theta (1 + Bi_L^{\tau+1})] \cdot T_M^{\tau+1} - 2Fo_N \theta \cdot T_{M-1}^{\tau+1} \\
 & = [(1 - 2Fo_N (1 - \theta) (1 + Bi_L^{\tau})) \cdot T_M^{\tau} + 2Fo_N (1 - \theta) \cdot T_{M-1}^{\tau} \\
 & \quad + \theta 2Fo_N Bi_L^{\tau+1} (T_{A_L}^{\tau+1} + \phi_L^{\tau+1} / h_L^{\tau+1}) + (1 - \theta) 2Fo_N Bi_L^{\tau} (T_{A_L}^{\tau} + \phi_L^{\tau} / h_L^{\tau})]
 \end{aligned} \tag{4.250}$$

4.7.5.7 Nodi interni (alle interfacce tra strati)

4.7.5.7.1 Discretizzazione spaziale del 1° Ordine

Per k indice dello strato e $I_k = I_{k,k+1}$ posizione dell'interfaccia tra due strati adiacenti, la congruenza dei flussi all'interfaccia è data da:

$$-\lambda_k \frac{\partial T_k(x, t)}{\partial x} \Big|_{I_k} = -\lambda_{k+1} \frac{\partial T_{k+1}(x, t)}{\partial x} \Big|_{I_k} \quad (4.251)$$

La sua approssimazione alle differenze finite è:

$$\lambda_k \frac{T_{I_k}^{\tau+1} - T_{I_k}^{\tau}}{\Delta x_k} = \lambda_{k+1} \frac{T_{I_{k+1}}^{\tau+1} - T_{I_k}^{\tau+1}}{\Delta x_{k+1}} \quad (4.252)$$

da cui

$$-\frac{\lambda_k}{\Delta x_k} T_{I_{k-1}}^{\tau+1} + \left(\frac{\lambda_k}{\Delta x_k} + \frac{\lambda_{k+1}}{\Delta x_{k+1}} \right) T_{I_k}^{\tau+1} - \frac{\lambda_{k+1}}{\Delta x_{k+1}} T_{I_{k+1}}^{\tau+1} = 0 \quad (4.253)$$

NOTA: anche in questo caso le equazioni di congruenza risultano sincrone e quindi non tengono in considerazione la capacità termica di due strati di parete, rispettivamente di spessori, $\Delta x_k/2$ e $\Delta x_{k+1}/2$.

4.7.5.7.2 Discretizzazione spaziale con i volumi finiti

$$\begin{aligned} \left(\rho_k c_k \frac{\Delta x_k}{2} + \rho_{k+1} c_{k+1} \frac{\Delta x_{k+1}}{2} \right) \frac{T_{I_k}^{\tau+1} - T_{I_k}^{\tau}}{\Delta \tau} &= \lambda_k \theta \frac{T_{I_{k-1}}^{\tau+1} - T_{I_k}^{\tau+1}}{\Delta x_k} \\ + \lambda_k (1 - \theta) \frac{T_{I_{k-1}}^{\tau} - T_{I_k}^{\tau}}{\Delta x_k} &+ \lambda_{k+1} \theta \frac{T_{I_{k+1}}^{\tau+1} - T_{I_k}^{\tau+1}}{\Delta x_{k+1}} + \lambda_{k+1} (1 - \theta) \frac{T_{I_{k+1}}^{\tau} - T_{I_k}^{\tau}}{\Delta x_{k+1}} \end{aligned} \quad (4.254)$$

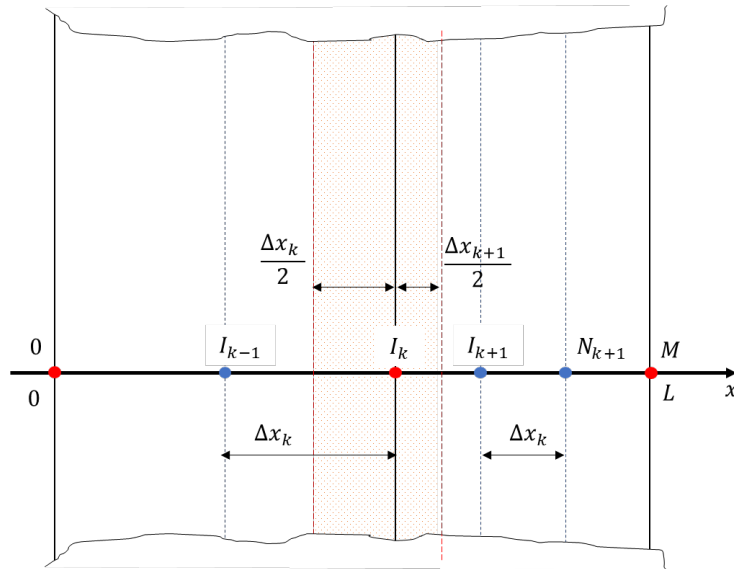


Figura 64 – Bilancio energetico su volume finito fittizio all'interfaccia tra due strati: I_k nodo all'interfaccia.

Definiti i seguenti numeri di Fourier

$$Fo_k^- = \frac{\lambda_k \Delta t / \Delta x_k}{\rho_k c_k \Delta x_k + \rho_{k+1} c_{k+1} \Delta x_{k+1}} \quad (4.255)$$

$$Fo_k^+ = \frac{\lambda_{k+1} \Delta t / \Delta x_{k+1}}{\rho_k c_k \Delta x_k + \rho_{k+1} c_{k+1} \Delta x_{k+1}} \quad (4.256)$$

si ha:

$$\begin{aligned} (T_{I_k}^{\tau+1} - T_{I_k}^{\tau}) &= 2\theta Fo_k^- (T_{I_{k-1}}^{\tau+1} - T_{I_k}^{\tau+1}) + 2(1-\theta) Fo_k^- (T_{I_{k-1}}^{\tau} - T_{I_k}^{\tau}) \\ &+ 2\theta Fo_k^+ (T_{I_{k+1}}^{\tau+1} - T_{I_k}^{\tau+1}) + 2(1-\theta) Fo_k^+ (T_{I_{k+1}}^{\tau} - T_{I_k}^{\tau}) \end{aligned} \quad (4.257)$$

E mettendo in evidenza le variabili

$$\begin{aligned} -2\theta Fo_k^- \cdot T_{I_{k-1}}^{\tau+1} + [1 + 2\theta(Fo_k^- + 2\theta Fo_k^+)] \cdot T_{I_k}^{\tau+1} - 2\theta Fo_k^+ \cdot T_{I_{k+1}}^{\tau+1} \\ = +2(1-\theta) Fo_k^- \cdot T_{I_{k-1}}^{\tau} + [1 - 2(1-\theta)(Fo_k^- + Fo_k^+)] \cdot T_{I_k}^{\tau} + 2(1-\theta) Fo_k^+ \cdot T_{I_{k+1}}^{\tau} \end{aligned} \quad (4.258)$$

Nel caso in cui uno dei due strati limitrofi sia uno strato non capacitivo (solo resistivo) (ad esempio un'intercapedine d'aria), lo schema resta identico, con l'unica differenza che, se è k lo strato puramente resistivo, si ha automaticamente per $c_k = 0$:

$$Fo_k^- = \frac{\lambda_k \Delta t / \Delta x_k}{\rho_{k+1} c_{k+1} \Delta x_{k+1}} \quad (4.259)$$

$$Fo_k^+ = \frac{\lambda_{k+1} \Delta t / \Delta x_{k+1}}{\rho_{k+1} c_{k+1} \Delta x_{k+1}} \quad (4.260)$$

con

$$\lambda_k = h_{int} \cdot L_k \quad ; \quad \Delta x_k = L_k / 2 \quad (4.261)$$

h_{int} coefficiente di scambio termico convettivo-radiativo dell'intercapedine, in $[W(m^2 K)]$;

L_k spessore dell'intercapedine, [m].

Nel caso in cui l'utente imponesse la presenza di due strati non capacitivi adiacenti, invece di impedire l'esecuzione, viene adottato in modo automatico il seguente schema:

$$0 = \lambda_k \frac{T_{I_{k-1}}^{\tau+1} - T_{I_k}^{\tau+1}}{\Delta x_k} + \lambda_{k+1} \frac{T_{I_{k+1}}^{\tau+1} - T_{I_k}^{\tau+1}}{\Delta x_{k+1}} \quad (4.262)$$

da cui

$$T_{I_k}^{\tau+1} = \frac{\left(\frac{\lambda_k}{\Delta x_k}\right) T_{I_{k-1}}^{\tau+1} + \left(\frac{\lambda_{k+1}}{\Delta x_{k+1}}\right) T_{I_{k+1}}^{\tau+1}}{\frac{\lambda_k}{\Delta x_k} + \frac{\lambda_{k+1}}{\Delta x_{k+1}}} \quad (4.263)$$

4.7.6 Conduzione termica stazionaria

Il modello in stato stazionario è usato principalmente per trattare gli elementi trasparenti, che sono poco capacitivi e quindi che soddisfano l'ipotesi dello stato quasi stazionario. Quindi, la descrizione del modello stazionario è di seguito riferita a un elemento trasparente.

Un componente trasparente è definito da un numero di strati, alcuni solidi (lastre), altri gassosi (intercapedini riempite di gas).

La soluzione del problema stazionario è determinata per un numero di “nodi” in uno schema elettrico equivalente (Figura 65), uguale al numero di lastre trasparenti moltiplicato per tre.

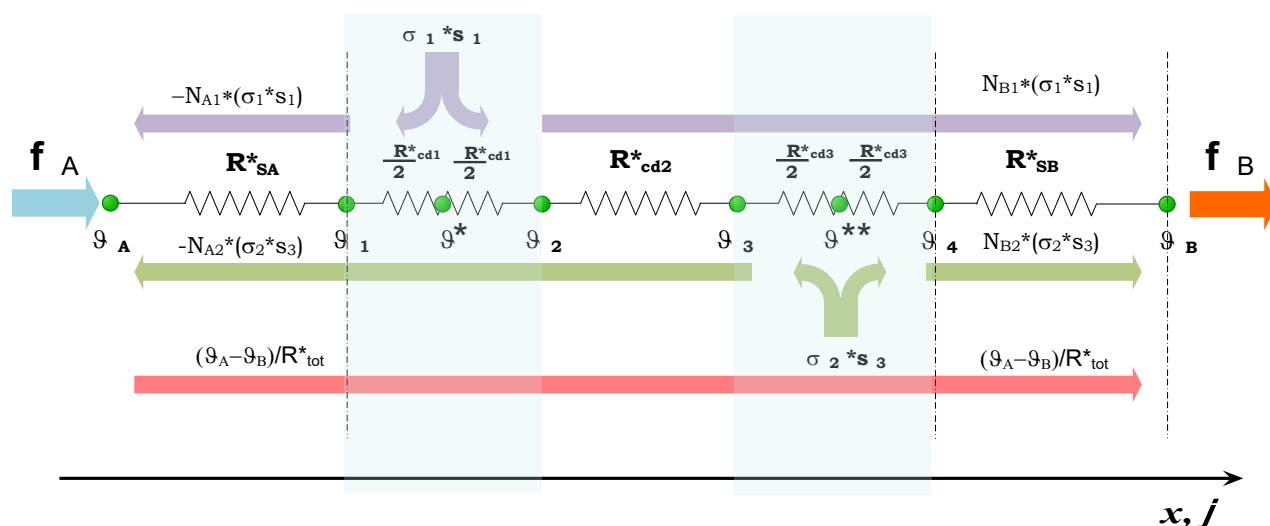


Figura 65 – Schema elettrico equivalente trasmissione termica in regime stazionario

Nell'implementare il modello della rete elettrica equivalente, sono stati presi in considerazione i due seguenti aspetti principali:

1. i calcoli relative alla generazione interna di calore internamente alle lastre di materiale trasparente viene trattata con il “bipolo elettrico equivalente”, inserendo il flusso termico imposto conseguente all’assorbimento tra due resistenze conduttive, una “backward” e una “forward” che ripartiscono in due la resistenza conduttiva della lastra;
2. alcune resistenze (quelle relative alle intercapedini con gas) possono avere le resistenze termiche dipendenti dal risultato (problema non lineare).

Per calcolare queste resistenze, “forward” e “backward”, insieme con il vettore delle resistenze (resistenze degli strati), viene definito un vettore aggiuntivo, che contiene tali resistenze aggiuntive:

```
IncrementalResistanceFromAtoB = new double[NodesNum];  
SingularResistanceFromAtoB = new double[NodesNum];
```

Durante l'inizializzazione dell'oggetto, le resistenze costanti vengono immagazzinate nel vettore `singularResistance`.

Le equazioni risolvanti lo schema elettrico equivalente sfruttano il principio di sovrapposizione degli effetti e sono date, con riferimento a Figura 65, da:

$$\varphi_B = U \cdot (\vartheta_A - \vartheta_B) + N_{B1} \cdot (\sigma_1 \cdot s_1) + N_{B2} \cdot (\sigma_2 \cdot s_2) + N_{B1} \cdot (\sigma_3 \cdot s_3) \quad (4.264)$$

$$\varphi_A = U \cdot (\vartheta_A - \vartheta_B) - N_{A1} \cdot (\sigma_1 \cdot s_1) - N_{A2} \cdot (\sigma_2 \cdot s_2) - N_{A1} \cdot (\sigma_3 \cdot s_3) \quad (4.265)$$

dove:

- ϑ_A temperatura operante ambiente A, in [°C];
- ϑ_B temperatura operante ambiente B, in [°C];
- U trasmittanza termica componente, in [W/(m² K)];
- σ_1 sorgente termica omogenea equivalente (ad es, radiazione solare assorbita) all'interno della lastra 1, [W/m³];
- σ_2 sorgente termica omogenea equivalente (ad es, radiazione solare assorbita) all'interno della lastra 2, [W/m³];
- σ_3 sorgente termica omogenea equivalente (ad es, radiazione solare assorbita) all'interno della lastra 3, [W/m³];
- s_1 spessore lastra 1, in [m];
- s_2 spessore lastra 2, in [m];
- s_3 spessore lastra 3, in [m];
- N_{A1} fattore di ripartizione della potenza termica generata all'interno della lastra 1 e dissipata verso l'ambiente A, adimensionale;
- N_{A2} fattore di ripartizione della potenza termica generata all'interno della lastra 2 e dissipata verso l'ambiente A, adimensionale;
- N_{A3} fattore di ripartizione della potenza termica generata all'interno della lastra 3 e dissipata verso l'ambiente A, adimensionale;
- N_{B1} fattore di ripartizione della potenza termica generata all'interno della lastra 1 e dissipata verso l'ambiente B, adimensionale;
- N_{B2} fattore di ripartizione della potenza termica generata all'interno della lastra 2 e dissipata verso l'ambiente B, adimensionale;
- N_{B3} fattore di ripartizione della potenza termica generata all'interno della lastra 3 e dissipata verso l'ambiente B, adimensionale;

con

$$U = \frac{1}{R_{Tot}^*} = 1 / \left(R_{SA}^* + R_{SB}^* + \sum_{i=1}^{N_L} R_{cdi}^* + \sum_{i=1}^{N_{int}} R_{int,i}^* \right) \quad (4.266)$$

dove

- R_{Tot}^* resistenza termica areica totale, in [m² K/W];
- N_L numero delle lastre;
- N_{int} numero delle intercapedini;
- λ_i conduttività termica del materiale della lastra i-esima, in [W/(m K)];

$h_{int,i}$ coefficiente di scambio termico convettivo-radiativo dell'intercapedine i-esima, in [W/(m² K)];

$R_{SA}^* = 1/h_A$ resistenza termica areica superficiale convettivo-radiativa con l'ambiente A, in [m² K/W];

$R_{SB}^* = 1/h_B$ resistenza termica areica superficiale convettivo-radiativa con l'ambiente B, in [m² K/W];

$R_{cdi}^* = s_i/\lambda_i$ resistenza termica areica conduttiva della lastra i-esima, in [m² K/W];

$R_{int,i}^* = 1/(h_{int,i}s_i)$ resistenza termica areica convettivo-radiativa della intercapedine i-esima, in [m² K/W];

e con

$$\begin{aligned} N_{A1} &= (R_{cd1}^*/2 + R_{cd1}^* + R_{int1}^* + R_{cd2}^* + R_{int2}^* + R_{cd3}^* + R_{SB}^*)/R_{Tot}^* \\ &= (R_{Tot}^* - R_{cd1}^*/2 - R_{SA}^*)/R_{Tot}^* \end{aligned} \quad (4.267)$$

$$N_{A2} = (R_{cd2}^*/2 + R_{int2}^* + R_{cd3}^* + R_{SB}^*)/R_{Tot}^* \quad (4.268)$$

$$N_{A3} = (R_{cd3}^*/2 + R_{SB}^*)/R_{Tot}^* \quad (4.269)$$

$$N_{B1} = (R_{SA}^* + R_{cd1}^*/2)/R_{Tot}^* \quad (4.270)$$

$$N_{B2} = (R_{SA}^* + R_{cd1}^* + R_{int,1}^* + R_{cd2}^*/2)/R_{Tot}^* \quad (4.271)$$

$$\begin{aligned} N_{B3} &= (R_{SA}^* + R_{cd1}^* + R_{int,1}^* + R_{cd2}^* + R_{int,2}^* + R_{cd3}^*/2)/R_{Tot}^* \\ &= (R_{Tot}^* - R_{cd3}^*/2 - R_{SB}^*)/R_{Tot}^* \end{aligned} \quad (4.272)$$

4.8 Zona: Bilancio di energia termica del nodo d'aria

4.8.1 Ipotesi di miscelamento perfetto; equazione differenziale ordinaria (ODE)

Sotto l'ipotesi di miscelamento perfetto dell'aria contenuta nella zona termica (volume di controllo), si ha:

$$\langle T_{ai} \rangle_V(t) = \frac{1}{V} \int_V T_{ai}(\vec{x}, t) dV \rightarrow T_{ai}(t) \quad (4.273)$$

Con tale ipotesi e considerando trascurabili le dissipazioni interne nell'aria dovute ai gradienti di velocità e i processi quasi ovunque isobari, [62], l'equazione integrale di bilancio entalpico sensibile dell'aria che occupa il volume della zona può essere scritta come:

$$V c_p \rho \frac{dT_{ai}}{dt} = \sum_{i=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i} - T_{ai}) + \sum_{j=1}^{N_{in}} \dot{m}_j c_{p,j} (T_{a,j} - T_{ai}) + \dot{Q}_{p\&f}^{(s;cv+ad)} + \dot{Q}_{sys}^{(s;cv+ad)} \quad (4.274)$$

dove

- V volume occupato dall'aria della zona termica, in [m³];
- c_p capacità termica massica dell'aria umida in [J/(kg^(AS) K)] ($c_p = c_p^{(AS)} + x_{ai} \cdot c_p^{(H_2O)}$) nelle condizioni medie di volume.
- x_{ai} umidità massica media di volume dell'aria interna, in [kg^(H₂O)/kg^(AS)];
- ρ massa volumica dell'aria secca, valore medio di volume, in [kg/m³];
- A_i area delle superficie i-esima della zona con cui l'aria scambia per convezione, in [m²];
- $h_{cv,i}$ coefficiente di scambio termico convettivo della superficie i-esima, in [W/(m² K)];
- $T_{s,i}$ temperatura della superficie i-esima, in [°C];
- T_{ai} temperatura media di volume dell'aria interna, in [°C];
- $\dot{m}_j$ portata massica entrante di aria secca attraverso aperture/fenditure nelle superfici di contorno della zona, in [kg^(AS)/s];
- $c_{p,j}$ capacità termica massica dell'aria umida in [J/(kg^(AS) K)] nelle condizioni di umidità dell'aria immessa;
- $T_{a,j}$ temperatura media di superficie (di immissione) dell'aria immessa attraverso aperture/fenditure nelle superfici di contorno della zona, in [°C];
- $\dot{Q}_{p\&f}^{(s;cv+ad)}$ potenza termica sensibile ceduta all'aria interna a causa di persone e cose che scambiano per convezione e/o avvezione con l'aria interna, in [W];
- $\dot{Q}_{sys}^{(s;cv+ad)}$ potenza termica sensibile ceduta/estratta all'aria interna dal sistema di climatizzazione che scambia per convezione e/o avvezione con l'aria interna, in [W].

Separando il termine avvevativo dovuto alle infiltrazioni dirette di aria esterna dai termini di interscambio tra zone limitrofe e isolando a primo membro il termico con l'incognita T_{ai} , si ottiene:

$$V c_p \rho \frac{dT_{ai}}{dt} + \left(\dot{m}_{inf} c_{p,ext} + \sum_{i=1}^{N_{surf}} A_i h_{cv,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,i} \right) T_{ai} = \sum_{i=1}^{N_{surf}} A_i h_{cv,i} T_{s,i} \quad (4.275)$$

$$+ \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} T_{z,j} + \dot{m}_{inf} c_{p,e} T_{ae} + \dot{Q}_{p\&f}^{(s;cv+ad)} + \dot{Q}_{sys}^{(s;cv+ad)}$$

dove:

$\dot{m}_{inf}$ portata massica di aria secca esterna di infiltrazione, in $[\text{kg}^{(AS)}/\text{s}]$;

$c_{p,e}$ capacità termica massica dell'aria umida esterna in $[\text{J}/(\text{kg}^{(AS)} \text{ K})]$ nelle condizioni di umidità dell'aria esterna;

T_{ae} temperatura aria esterna, in $[\text{°C}]$;

$T_{z,j}$ temperatura aria interna zone adiacenti, in $[\text{°C}]$;

Indicata la capacità termica dell'aria umida contenuta nella zona alle condizioni di umidità media di volume x_{ai} , come

$$C = V \rho c_p(x_{ai})$$

e

$$A(t) = \frac{1}{C} \left(\sum_{i=1}^{N_{surf}} A_i h_{cv,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} + \dot{m}_{inf} c_{p,ext} \right)$$

$$f(t) = \frac{1}{C} \left(\sum_{i=1}^{N_{surf}} A_i h_{cv,i} T_{s,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} T_{z,j} + \dot{m}_{inf} c_{p,e} T_{ae} + \dot{Q}_{p\&f}^{(s;cv+ad)} + \dot{Q}_{sys}^{(s;cv+ad)} \right)$$

la (4.275) può essere riscritta come:

$$\frac{dT_{ai}}{dt} + A(t) \cdot T_{ai} = f(t) \quad (4.276)$$

equazione differenziale ordinaria lineare del primo ordine non omogenea, il cui integrale generale ha la seguente forma:

$$T_{ai}(t) = e^{-\int A(t)dt} \cdot \int f(t) e^{\int A(t)dt} dt \quad (4.277)$$

Per poter svolgere gli integrali presenti a secondo membro della (4.277) occorre però conoscere a priori la forma analitica della dipendenza di $A(t)$ e $f(t)$ dal tempo, cosa quasi mai possibile.

Nei seguenti paragrafi sono riportate alcune soluzioni analitiche e numeriche che si possono ottenere sotto specifiche ipotesi semplificative, soluzioni che sono state tutte implementate in OpenBPS, a scopo comparativo.

at a precise time $t = t_i$, we have an ordinary differential equation representing the single node of the air zone (all the following terms are evaluated at the same time $t = t_i$):

4.8.2 ODE: soluzione analitica con parametri costanti

Nel seguito è riportata la soluzione analitica nell'ipotesi di parametri costanti:

$$\text{IP1: } \rightarrow A(t) = \bar{A} = \text{cost} ; f(t) = \bar{f} = \text{cost}$$

cioè $h_{cv,i}, \dot{m}_j c_{p,j}, \dot{m}_{inf} c_{p,e}, T_{s,i}, T_{z,j}, T_{ae}, \dot{Q}_{p\&f}^{(s;cv+ad)}, \dot{Q}_{sys}^{(s;cv+ad)} \rightarrow$ costanti all'interno dell'intervallo di integrazione.

In tal caso la (4.277) si riduce a:

$$T_{ai}(t) = e^{-\bar{A}t} \cdot \left[\bar{f} \int e^{\bar{A}t} dt + C_1 \right] = e^{-\bar{A}t} \cdot \left[\frac{\bar{f}}{\bar{A}} e^{\bar{A}t} + C_1 \right] = \frac{\bar{f}}{\bar{A}} + C_1 e^{-\bar{A}t} \quad (4.278)$$

Per determinare la costante C_1 poniamo la condizione

$$T_{ai}(t_1) = \frac{\bar{f}}{\bar{A}} + C_1 e^{-\bar{A}t_1} \Rightarrow C_1 = T_{ai}(t_1) e^{\bar{A}t_1} - \frac{\bar{f}}{\bar{A}} e^{\bar{A}t_1} \quad (4.279)$$

da cui

$$T_{ai}(t) = \frac{\bar{f}}{\bar{A}} (1 - e^{-\bar{A}(t-t_1)}) + T_{ai}(t_1) e^{-\bar{A}(t-t_1)} \quad (4.280)$$

Infine se consideriamo il tempo $t_2 = t_1 + \Delta t$, la temperatura dopo un intervallo temporale Δt è pari a:

$$T_{ai}(t_2) = \frac{\bar{f}}{\bar{A}} + \left(T_{ai}(t_1) - \frac{\bar{f}}{\bar{A}} \right) \cdot e^{-\bar{A}\Delta t} \quad (4.281)$$

NOTA: la (4.281) consente di determinare analiticamente la variazione di temperatura del nodo aria qualora si considerino costanti tutti i parametri dell'equazione di bilancio termico all'interno dell'intervallo di integrazione Δt . È però possibile variare tali parametri nel successivo intervallo di integrazione, ottenendo così un'approssimazione integrale a gradini (rispetto ai parametri).

4.8.1 ODE: soluzione analitica con parametri variabili linearmente

Nel seguito è riportata la soluzione analitica nell'ipotesi di coefficienti costanti e temperature e potenze variabili nel tempo linearmente:

$$\text{IP2: } \rightarrow A(t) = \bar{A} = \text{cost} ; f(t) = f(t_1) + \frac{f(t_2) - f(t_1)}{t_2 - t_1} (t - t_1) = a + b(t - t_1)$$

cioè $h_{cv,i}, \dot{m}_j c_{p,j}, \dot{m}_{inf} c_{p,e} \rightarrow$ costanti all'interno dell'intervallo di integrazione.

In tal caso la (4.277) si riduce a:

$$T_{ai}(t) = e^{-\bar{A}t} \cdot \left[\int f(t) e^{\bar{A}t} dt + C_1 \right] = e^{-\bar{A}t} \cdot \left[\frac{a\bar{A} + b[\bar{A}(t - t_1) - 1]}{\bar{A}^2} e^{\bar{A}t} + C_1 \right] \quad (4.282)$$

Per determinare la costante C_1 poniamo la condizione

$$T_{ai}(t_1) = \frac{a\bar{A} - b}{\bar{A}^2} + C_1 e^{-\bar{A}t_1} \quad \Rightarrow \quad C_1 = T_{ai}(t_1) e^{\bar{A}t_1} - \frac{a\bar{A} - b}{\bar{A}^2} e^{\bar{A}t_1} \quad (4.283)$$

da cui

$$T_{ai}(t) = \frac{a\bar{A} - b}{\bar{A}^2} (1 - e^{-\bar{A}(t-t_1)}) + \frac{b}{\bar{A}} (t - t_1) + T_{ai}(t_1) e^{-\bar{A}(t-t_1)} \quad (4.284)$$

Infine se consideriamo il tempo $t_2 = t_1 + \Delta t$, la temperatura dopo un intervallo temporale Δt è pari a:

$$T_{ai}(t_2) = \frac{a\bar{A} - b}{\bar{A}^2} + \frac{b}{\bar{A}} \Delta t + \left(T_{ai}(t_1) - \frac{a\bar{A} - b}{\bar{A}^2} \right) \cdot e^{-\bar{A}\Delta t} \quad (4.285)$$

NOTA: la (4.285) consente di determinare analiticamente la variazione di temperatura del nodo aria qualora si considerino costanti i coefficienti e variabile linearmente la forzante dell'equazione di bilancio termico all'interno dell'intervallo di integrazione Δt , consentendo una migliore approssimazione rispetto al caso di parametri costanti.

4.8.2 ODE: integrazione numerica

Il problema di Cauchy (detto anche problema ai valori iniziali) consiste nel determinare la soluzione di un'equazione differenziale ordinaria (in breve nel seguito ODE, dall'inglese *ordinary differential equation*), scalare o vettoriale, completata da opportune condizioni iniziali. In particolare, nel caso scalare, indicato con I un intervallo di $\mathbb{R}$ contenente un punto t_0 , il problema di Cauchy associato ad una ODE del primo ordine si formula come:

trovare una funzione $y \in C^1(I)$ a valori reali tale che

$$\begin{cases} y'(t) = f(t, y(t)) & t \in I \\ y(t_0) = y_0 \end{cases} \quad (4.286)$$

dove $f(t, y)$ è una funzione assegnata nella striscia $S = I \times (-\infty + \infty)$, a valori reali e continua rispetto ad entrambe le variabili. Nel caso particolare in cui f dipenda da t solo attraverso y , l'equazione differenziale si dice autonoma. La trattazione che segue si limita allo studio di una singola equazione differenziale (caso scalare).

Osserviamo che se f è una funzione continua rispetto a t e si integra la (4.286) fra t_0 e t si ottiene

$$y(t) - y_0 = \int_{t_0}^t y'(\tau) d\tau = \int_{t_0}^t f(\tau, y(\tau)) d\tau \quad (4.287)$$

Pertanto $y \in C^1(I)$ soddisfa all'equazione integrale (4.287). Viceversa, se y è una soluzione della (4.287), continua in I , allora si ha $y(t_0) = y_0$ e y è una primitiva della funzione continua $f(\cdot, y(\cdot))$. Di conseguenza $y \in C^1(I)$ e verifica l'equazione differenziale $y'(t) = f(t, y(t))$. Sussiste dunque un'equivalenza fra il problema di Cauchy (4.286) e l'equazione integrale (4.287), che sfrutteremo a livello numerico.

4.8.2.1 Metodo di Eulero

Il metodo di Eulero per l'integrazione di un'equazione differenziale lineare ordinaria del primo ordine consiste nell'approssimare la derivata prima nel tempo con il suo rapporto incrementale, il che equivale ad approssimarla con una differenza in avanti, cioè:

$$\frac{dT_{ai}}{dt} \cong \frac{T_{ai}(t + \Delta t) - T_{ai}(t)}{\Delta t} \quad (4.288)$$

da cui, sostituendo nella (4.274) e introducendo la variabile discreta T_{ai}^τ e l'indice temporale adimensionale τ , tale che $t = \tau \cdot \Delta t$, si ottiene:

$$\begin{aligned} C \frac{T_{ai}^{\tau+1} - T_{ai}^\tau}{\Delta t} = & \sum_{i=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i}^\tau - T_{ai}^\tau) + \sum_{j=1}^{N_{coupl}} \dot{m}_j^\tau c_{p,j} (T_{z,j}^\tau - T_{ai}^\tau) + \dot{m}_{inf}^\tau c_{p,e} (T_{ae}^\tau - T_{ai}^\tau) \\ & + \dot{Q}_{p\&f}^\tau + \dot{Q}_{sys}^\tau \end{aligned} \quad (4.289)$$

Da cui si ottiene un'equazione algebrica esplicita:

$$\begin{aligned} T_{ai}^{\tau+1} = T_{ai}^\tau + \frac{\Delta t}{C} \sum_{i=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i}^\tau - T_{ai}^\tau) + \sum_{j=1}^{N_{coupl}} \dot{m}_j^\tau c_{p,j} (T_{z,j}^\tau - T_{ai}^\tau) + \dot{m}_{inf}^\tau c_{p,e} (T_{ae}^\tau - T_{ai}^\tau) \\ + \dot{Q}_{p\&f}^\tau + \dot{Q}_{sys}^\tau \end{aligned} \quad (4.290)$$

che può essere direttamente utilizzata in tale forma per l'integrazione numerica.

4.8.2.2 Metodo implicito

Il metodo implicito nasce dalla constatazione che il metodo di Eulero di fatto coincide con l'approssimazione della derivata temporale ordinaria con una differenza finita in avanti, e consiste nell'utilizzare per tale approssimazione una differenza indietro, cioè:

$$\frac{dT_{ai}}{dt} \cong \frac{T_{ai}(t) - T_{ai}(t - \Delta t)}{\Delta t} \quad (4.291)$$

da cui, sostituendo nella (4.274) e introducendo la variabile discreta T_{ai}^τ e l'indice temporale adimensionale τ , tale che $t = \tau \cdot \Delta t$, si ottiene con una traslazione unitaria del tempo:

$$\begin{aligned} C \frac{T_{ai}^{\tau+1} - T_{ai}^\tau}{\Delta t} = & \sum_{i=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i}^{\tau+1} - T_{ai}^{\tau+1}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j^{\tau+1} c_{p,j} (T_{z,j}^{\tau+1} - T_{ai}^{\tau+1}) \\ & + \dot{m}_{inf}^{\tau+1} c_{p,e} (T_{ae}^{\tau+1} - T_{ai}^{\tau+1}) + \dot{Q}_{p\&f}^{\tau+1} + \dot{Q}_{sys}^{\tau+1} \end{aligned} \quad (4.292)$$

Da cui si ottiene la seguente equazione algebrica:

$$T_{ai}^{\tau+1} \left(1 - \frac{\Delta t}{C} \sum_{i=1}^{N_{surf}} A_i h_{cv,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j^{\tau+1} c_{p,j} + \dot{m}_{inf}^{\tau+1} c_{p,e} \right) = T_{ai}^{\tau} + \frac{\Delta t}{C} \left(\sum_{i=1}^{N_{surf}} A_i h_{cv,i} T_{s,i}^{\tau+1} + \sum_{j=1}^{N_{coupl}} \dot{m}_j^{\tau+1} c_{p,j} T_{z,j}^{\tau+1} + \dot{m}_{inf}^{\tau+1} c_{p,e} T_{ae}^{\tau+1} + \dot{Q}_{p\&f}^{\tau+1} + \dot{Q}_{sys}^{\tau+1} \right) \quad (4.293)$$

E posto

$$D^{\tau+1} = \left(1 - \frac{\Delta t}{C} \sum_{i=1}^{N_{surf}} A_i h_{cv,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j^{\tau+1} c_{p,j} + \dot{m}_{inf}^{\tau+1} c_{p,e} \right) \quad (4.294)$$

la (4.293) può essere riscritta come:

$$T_{ai}^{\tau+1} = \frac{1}{D^{\tau+1}} T_{ai}^{\tau} + \frac{\Delta t}{D^{\tau+1} C} \cdot \left(\sum_{i=1}^{N_{surf}} A_i h_{cv,i} T_{s,i}^{\tau+1} + \sum_{j=1}^{N_{coupl}} \dot{m}_j^{\tau+1} c_{p,j} T_{z,j}^{\tau+1} + \dot{m}_{inf}^{\tau+1} c_{p,e} T_{ae}^{\tau+1} + \dot{Q}_{p\&f}^{\tau+1} + \dot{Q}_{sys}^{\tau+1} \right) \quad (4.295)$$

La (4.295) rappresenta lo schema implicito dell'equazione del nodo aria, che necessita per la sua utilizzazione della conoscenza delle grandezze al passo temporale attuale $\tau + 1$ e non il loro valore pregresso (all'istante τ), ad esclusione ovviamente del valore pregresso della temperatura dell'aria ambiente.

La formulazione implicita viene solitamente impiegata quando si utilizza lo schema completamente implicito per la soluzione della conduzione termica nelle pareti. In tal caso è conveniente raggruppare l'equazione del nodo aria e le equazioni matriciali di tutte le pareti in un unico sistema algebrico, costruendo una matrice unica che contenga tale equazione e tutte quelle delle pareti.

4.8.2.3 Metodo del trapezio (Crank-Nicolson): estensione al Theta Method

Il metodo del trapezio, detto anche di Crank-Nicolson, è uno dei metodi numerici di integrazione di ODE più utilizzato, e consiste nell'approssimare l'integrale a secondo membro della (4.287) con la formula del trapezio (sviluppata per il calcolo approssimato delle aree), ottenendo il seguente schema numerico:

$$y_{n+1} - y_n = \frac{\Delta t}{2} [f(t_n) + f(t_{n+1})] \quad (4.296)$$

con

Δt intervallo di discretizzazione del dominio temporale;

t_n tempo discretizzato, $t_n = n \cdot \Delta t$

n indice intero avanzamento temporale, $n \in \mathbb{N}$;

$f(t_n)$ funzione integranda valutata al tempo discreto t_n .

Il bilancio termico del nodo aria, definito dalla (4.274), può essere riscritto come:

$$V c_p \rho \frac{dT_{ai}}{dt} = \sum_{i=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) + \dot{Q} \quad (4.297)$$

dove si sono esplicitati i flussi avvevivi di infiltrazione, $\dot{m}_{inf}$, e si è compattato in un unico termine sorgente, $\dot{Q}_{tot}$, il contributo delle potenze sensibili dovuti a persone e cose, $\dot{Q}_{p\&f}^{(s;cv+ad)}$ e $\dot{Q}_{sys}^{(s;cv+ad)}$.

Applicando alla (4.297) lo schema della (4.296) e introducendo la variabile discreta T_{ai}^τ e l'indice temporale adimensionale τ , tale che $t = \tau \cdot \Delta t$, si ottiene:

$$\begin{aligned} C \left[\frac{dT_{ai}}{dt} \right]_{\frac{t_{i+1}+t_i}{2}} &= C \frac{T_{ai}^{\tau+1} - T_{ai}^\tau}{\Delta t} \\ &= \frac{1}{2} \left[\sum_{i=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) + \dot{Q}_{tot} \right]^{\tau+} \\ &\quad + \frac{1}{2} \left[\sum_{i=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) + \dot{Q}_{tot} \right]^\tau \end{aligned} \quad (4.298)$$

Possiamo generalizzare la forma espressa dalla (4.297), sostituiscono ai pesi $\frac{1}{2}$ il generico parametro θ , compreso tra 0 e 1, ottenendo:

$$\begin{aligned} &C \frac{T_{ai}^{\tau+1} - T_{ai}^\tau}{\Delta t} \\ &= \theta \left[\sum_{j=1}^{N_{surf}} A_j h_{cv,j} (T_{s,j} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) + \dot{Q}_{tot} \right]^{\tau+} \\ &\quad + (1 - \theta) \left[\sum_{j=1}^{N_{surf}} A_j h_{cv,j} (T_{s,j} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) + \dot{Q}_{tot} \right]^\tau \end{aligned} \quad (4.299)$$

Esplicitando le variabili si ha:

$$\left[C + \theta \left(\sum_{j=1}^{N_{surf}} A_j h_{cv,j} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} + \dot{m}_{inf} c_{p,e} \right) \right]^{\tau+1} T_{ai}^{\tau+1} \quad (4.300)$$

$$+(1-\theta) \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} T_{s,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} T_{z,j} + \dot{m}_{inf} c_{p,e} T_{ae} + \dot{Q}_{tot} \right]^{\tau}$$

Posto

$$D^{\tau+1} = \left[\frac{C}{\Delta t} + \theta \left(\sum_{j=1}^{N_{surf}} A_i h_{cv,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} + \dot{m}_{inf} c_{p,e} \right)^{\tau+1} \right] \quad (4.301)$$

la (4.300) si può riscrivere come

$$\begin{aligned} T_{ai}^{\tau+1} = & \frac{1}{D^{\tau+1}} \left[\frac{C}{\Delta t} - (1-\theta) \left(\sum_{j=1}^{N_{surf}} A_i h_{cv,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} + \dot{m}_{inf} c_{p,e} \right)^{\tau+1} \right] \cdot T_{ai}^{\tau} \\ & + \frac{\theta}{D^{\tau+1}} \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} T_{s,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} T_{z,j} + \dot{m}_{inf} c_{p,e} T_{ae} + \dot{Q}_{tot} \right]^{\tau+1} \\ & + \frac{1-\theta}{D^{\tau+1}} \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} T_{s,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} T_{z,j} + \dot{m}_{inf} c_{p,e} T_{ae} + \dot{Q}_{tot} \right]^{\tau} \end{aligned} \quad (4.302)$$

Si può derivare anche una formulazione alternativa, moltiplicando la (4.299) per $\Delta t/C$, ottenendo:

$$\begin{aligned} T_{ai}^{\tau+1} = T_{ai}^{\tau} + \theta \frac{\Delta t}{C} \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) \right. \\ \left. + \dot{Q}_{tot} \right]^{\tau+1} + (1-\theta) \frac{\Delta t}{C} \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) \right. \\ \left. + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) + \dot{Q}_{tot} \right]^{\tau} \end{aligned} \quad (4.303)$$

e raccogliendo solo rispetto a $T_{ai}^{\tau+1}$ si ottiene:

$$\begin{aligned} T_{ai}^{\tau+1} \left(1 + \theta \frac{\Delta t}{C} \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} + \dot{m}_{inf} c_{p,e} \right]^{\tau+1} \right) = \\ = T_{ai}^{\tau} + \theta \frac{\Delta t}{C} \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} T_{s,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} T_{z,j} + \dot{m}_{inf} c_{p,e} T_{ae} + \dot{Q}_{tot} \right]^{\tau+1} \\ + (1-\theta) \frac{\Delta t}{C} \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) + \dot{Q}_{tot} \right]^{\tau} \end{aligned} \quad (4.304)$$

Posto

$$D^{\tau+1} = \left(1 + \theta \frac{\Delta t}{C} \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} + \dot{m}_{inf} c_{p,e} \right]^{\tau+1} \right) \quad (4.305)$$

dalla (4.304) si ricava

$$\begin{aligned} T_{ai}^{\tau+1} = T_{ai}^{\tau} + \theta \frac{\Delta t}{D^{\tau+1} C} & \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} T_{s,i} + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} T_{z,j} + \dot{m}_{inf} c_{p,e} T_{ae} + \dot{Q}_{tot} \right]^{\tau+1} \\ + (1 - \theta) \frac{\Delta t}{D^{\tau+1} C} & \left[\sum_{j=1}^{N_{surf}} A_i h_{cv,i} (T_{s,i} - T_{ai}) + \sum_{j=1}^{N_{coupl}} \dot{m}_j c_{p,j} (T_{z,j} - T_{ai}) + \dot{m}_{inf} c_{p,e} (T_{ae} - T_{ai}) \right. \\ & \left. + \dot{Q}_{tot} \right] \end{aligned} \quad (4.306)$$

5 Validazione del codice dell'involucro

È in via di sviluppo una documentazione di tutti i test di verifica e validazione effettuati, che raccoglie la descrizione e risultati dei test eseguiti ed è a sua volta suddivisa in più sottocategorie inerenti alle:

1. verifiche di codice;
2. verifiche di calcolo;
3. validazioni:
 - a. empiriche
 - b. analitiche
 - c. comparative

Le verifiche di codice sono spesso relative al controllo del codice implementato a prescindere dagli algoritmi presenti al suo interno, come l'import di un file di EnergyPlus, o la gestione di diverse schermature insistenti su una partizione.

All'interno della documentazione inerente alle verifiche di calcolo, sono riportati i risultati di test eseguiti su singole parti di codice. Tali test sono funzionali alla verifica della corretta implementazione di un singolo algoritmo di calcolo.

Con validazione invece si intende un test più completo, che riguardi la simulazione così come potrebbe essere eseguita dall'utente. Tali validazioni sono suddivise in analitiche, empiriche e comparative. Tale suddivisione è relativa alla generazione dei risultati con i quali vengono valutati i risultati ottenuti dal software. Nel primo caso i risultati sono ottenuti mediante procedure che hanno una soluzione analitica, nel secondo caso i risultati ottenuti con il software sono confrontati con dati misurati in apposite campagne sperimentali e nell'ultimo caso il confronto è eseguito con i risultati ottenuti da altri software disponibili sul mercato. Quest'ultimo è l'approccio dello standard ASHRAE 140 [13] che utilizza i risultati del progetto BESTEST, un'attività congiunta dei membri del "Testing and Validation of Building Energy Simulation Tools Experts Group" dell'Agenzia internazionale dell'energia (IEA), gruppo composto da esperti del IEA Solar Heating and Cooling (SHC) Programme, Task 34, e del Energy Conservation in Buildings and Community Systems (ECBCS) Programme Annex 43.

OpenBPS è continuamente in fase di validazione, essendo ancora in fase di sviluppo, così come la documentazione dei test, che sarà disponibile in forma completa solo al termine del progetto triennale, e omessa da questo rapporto intermedio per non appesantirlo, salvo la validazione comparativa riportata nel seguito.

5.1.1 Validazione con BESTEST

Per validare il codice di calcolo nel suo complesso, è stato utilizzato, così come indicato dalla norma internazionale UNI EN ISO 52016-1:2017, lo standard ASHARE 140 (BESTEST), con esito positivo. Si riportano nel seguito i risultati sintetici delle valutazioni comparative, effettuate per gli edifici e il clima campione descritti nello standard, come previsto dal BESTEST per i seguenti casi di test:

- **caso 600FF:** in Figura 66 è riportato l'andamento della temperatura interna in regime libero (senza impianto) con pareti di colore chiaro (basso assorbimento della radiazione solare) in condizioni invernali (4 di gennaio);
- **caso 650FF:** in Figura 67 è riportato l'andamento della temperatura interna in regime libero (senza impianto) con pareti di colore chiaro (basso assorbimento della radiazione solare) in condizioni estive e con ventilazione notturna;
- **caso 900FF:** in Figura 68 è riportato l'andamento della temperatura interna in regime libero (senza impianto) con pareti più capacitive rispetto al caso 600FF, in condizioni invernali;
- **caso 950FF:** Figura 69 è riportato l'andamento della temperatura interna in regime libero (senza impianto) con pareti più capacitive rispetto al caso 650FF in regime estivo e sempre con ventilazione notturna.

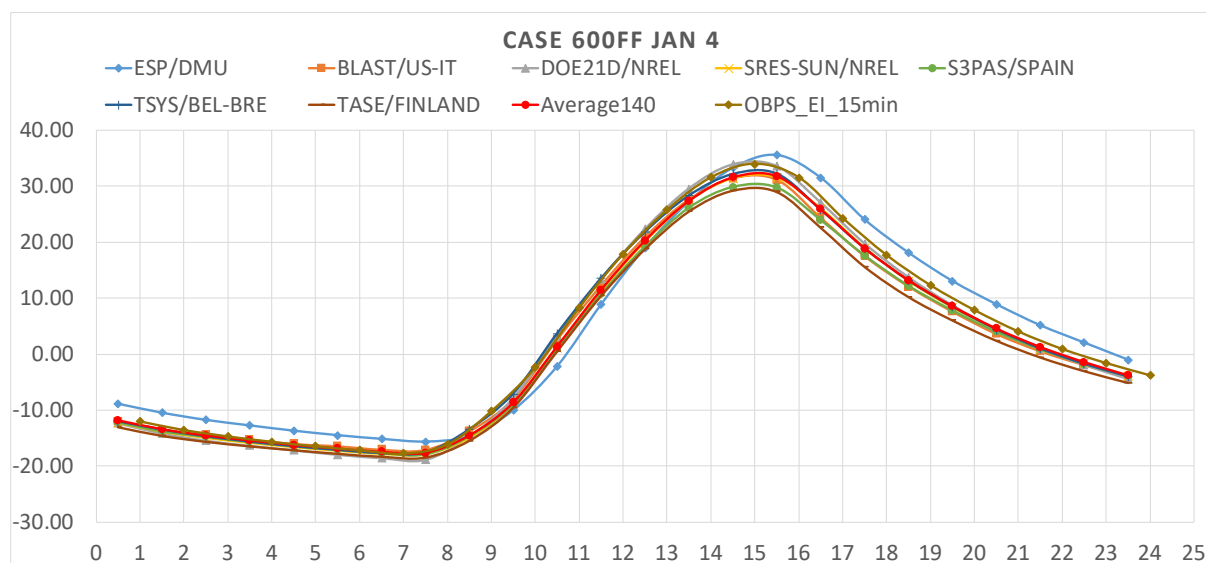


Figura 66 – ASHRAE 140 Standard validazione di OpenBPS: caso 600FF -invernale.

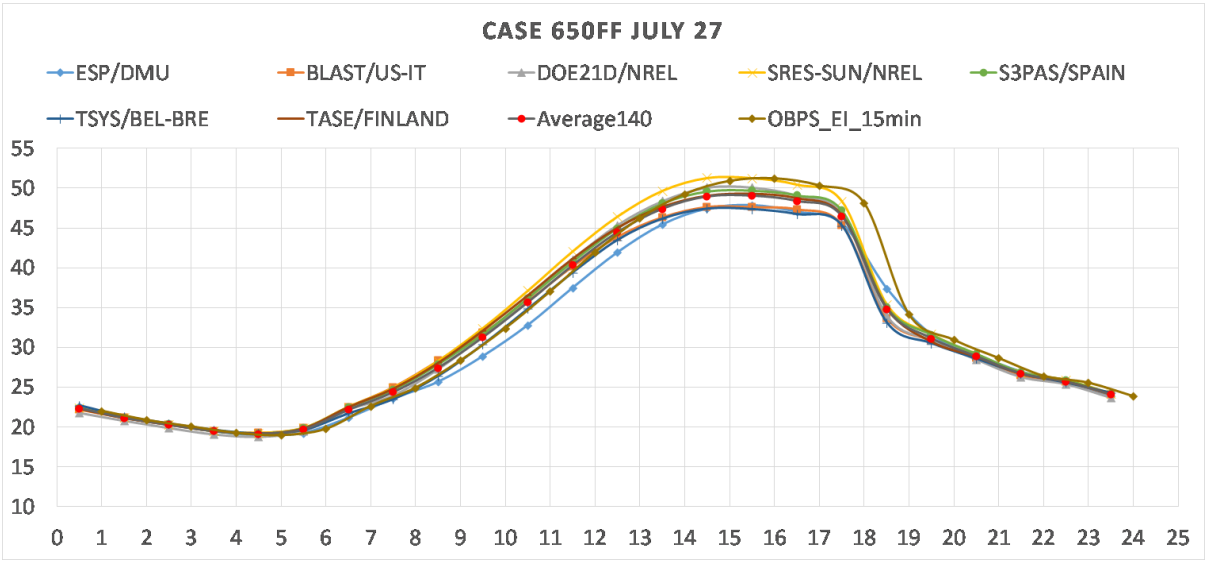


Figura 67 – ASHRAE 140 Standard validazione di OpenBPS: caso 650FF -estivo.

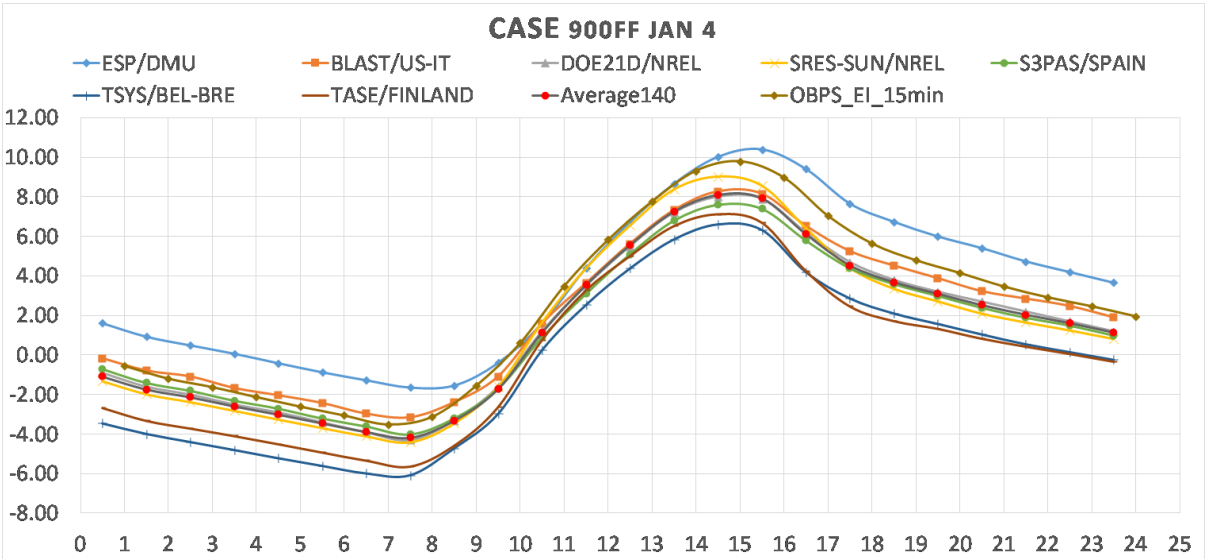


Figura 68 – ASHRAE 140 Standard validazione di OpenBPS: caso 900FF -invernale.

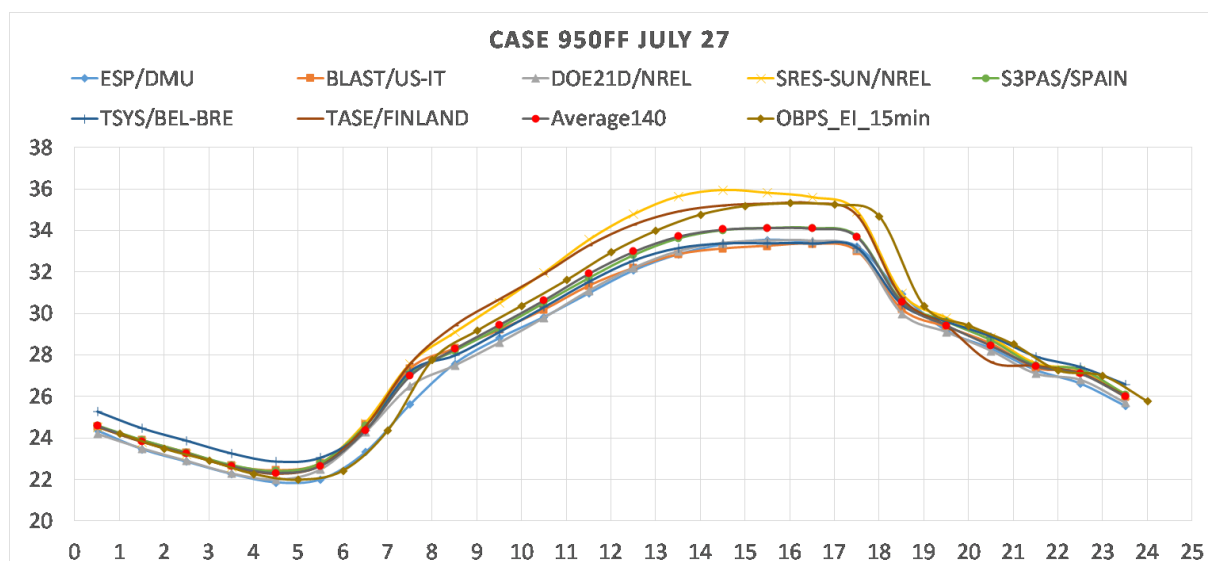


Figura 69 – ASHRAE 140 Standard validazione di OpenBPS: caso 950FF -estivo.

Dall'esame dei grafici si può veder come in tutti i casi OpenBPS si comporta in modo analogo agli altri software di riferimento, posizionandosi quasi sempre leggermente sopra l'andamento medio.

6 Conclusione e prospettive

Lo strumento di simulazione delle prestazioni dell'edificio orientato agli oggetti OpenBPS non è ancora pronto per essere distribuito alla comunità della simulazione energetica perché manca dei modelli dei componenti dei sistemi impiantistici. Questa parte è attualmente in fase di sviluppo cercando di modellare i componenti di sistema come componenti dinamici anche se semplificati. Dall'altro lato, si deve prendere una decisione su come avviare la comunità degli sviluppatori, per creare un vero progetto open source. L'idea principale, che può supportare e aumentare il suo sviluppo e la sua diffusione, è di lanciarlo come “standard de facto” in sostituzione dell'intero set di standard EPBD, che sarà in vigore alla fine del 2017. Questo nuovo set di standard EPBD implica che ogni software house tecnica, che vende programmi per valutare le prestazioni energetiche degli edifici, deve riscrivere il proprio motore di calcolo secondo i nuovi standard. Successivamente, devono richiedere una convalida legale prima di metterli sul mercato. Una joint venture tra OpenBPS e tali software house tecniche può risolvere il problema di entrambi: un unico motore di calcolo validato completo, che può essere utilizzato all'interno di qualsiasi software su qualsiasi piattaforma, completamente documentato ed espandibile di volta in volta.

7 Applicazione al caso studio

7.1 Descrizione dell'edificio

Nei successivi paragrafi vengono illustrate le informazioni e i dati relativi al contesto in cui sorge l'edificio oggetto di analisi, alle caratteristiche del fabbricato e degli impianti esistenti. La documentazione fornita da ENEA e le assunzioni coordinate con le altre università partecipanti al progetto sono state utilizzate come riferimento per la caratterizzazione e la modellazione dell'edificio.

7.1.1 Informazioni generali e contesto esterno

L'edificio oggetto di analisi è un edificio residenziale condominiale sito nel quartiere San Lorenzo (Roma), tra le stazioni ferroviarie di Roma Termini e Roma Tiburtina (Figura 70).



Figura 70. Inquadramento territoriale (da Google Maps).

L'edificio è caratterizzato da una forma trapezoidale (Figura 71) con un lato adiacente all'edificio confinante; gli altri tre lati affacciano rispettivamente su Via degli Enotri, Via dei Reti e Viale dello Scalo San Lorenzo. È orientato principalmente verso nord-ovest (il prospetto su Via degli Enotri è orientato a 55° nord-ovest). È inserito in un contesto prevalentemente residenziale e commerciale con edifici più bassi o di pari altezza rispetto al caso studio, che non risulta, quindi, particolarmente ombreggiato dagli edifici circostanti. Infatti, le facciate esposte ad est e sud-est si affacciano su ampi spazi aperti privi di edifici costruiti (Figura 71).



Figura 71. Individuazione dell’edificio (da Google Maps).

L’edificio oggetto di analisi è costituito da 7 piani fuori terra (Figura 72) e un piano seminterrato. Le principali caratteristiche geometriche dell’edificio sono riportate in Tabella 4. Il piano terra, non oggetto di analisi, ospita quattro locali commerciali (in grigio in Figura 72) e un locale condominiale, mentre gli altri sei piani fuori terra ospitano 3 alloggi ciascuno, per un totale di 18 alloggi, e gli spazi di servizio (vano scala comune). I piani intermedi (dal piano 1 al piano 4) sono caratterizzati da piante simili, con alloggi di 48, 57 e 87 m² (rispettivamente in azzurro, arancione e giallo in Figura 72). Gli ultimi due piani, invece, presentano un arretramento della facciata su Via dei Reti per la creazione di un terrazzo (sulla copertura del quarto piano) sul piano quinto e di un balcone sul piano sesto. Gli alloggi presentano metrature inferiori rispetto ai piani sottostanti, in particolar di 39, 47 e 85 m² (rispettivamente in verde, rosso e blu in Figura 72). In Figura 73 vengono riportati i prospetti dell’edificio oggetto di analisi.

Tabella 4. Principali caratteristiche geometriche dell’edificio.

Parametro	Simbolo	U.M.	Valore
Volume lordo riscaldato	V_l	m ³	5280
Superficie netta di pavimento riscaldata	A_f	m ²	1091
Superficie d’involucro disperdente	A_{env}	m ²	1958
Rapporto di forma	A_{env} / V_l	m ⁻¹	0,37



Figura 72. Piante del piano terra, piani 1-4 (pianta tipo) e piani 5/6 (pianta tipo).



Figura 73. Prospetti sud-est (a – Viale dello Scalo San Lorenzo), nord-est (b – Via dei Reti) e nord-ovest (c – Via degli Enotri).

7.1.2 Descrizione del fabbricato

L'edificio, edificato nel 1926, presenta una struttura a setti portanti in laterizio e pietra, e un involucro edilizio prevalentemente non isolato. La classificazione dei componenti d'involucro è stata realizzata a partire dalla documentazione fornita da ENEA e secondo le specifiche delle norme tecniche.

Per quanto riguarda l'involucro opaco, si è fatto riferimento alla descrizione della stratigrafia riportata nella documentazione tecnica, e le caratteristiche dei materiali sono state ricavate dalla normativa tecnica. Le principali caratteristiche termofisiche dei componenti edilizi opachi sono riportate in Tabella 5 e Tabella 6.

Le pareti perimetrali esterne, ad esclusione degli ultimi due piani, hanno uno spessore di 70 cm e sono caratterizzate da due filari di laterizi pieni separati da uno strato di ciottoli e pietre frantumati, con rivestimento interno ed esterno in intonaco di calce e sabbia. A seguito di una probabile sopraelevazione, le pareti esterne degli ultimi due piani si presentano più leggere rispetto alle pareti sottostanti. Sono caratterizzate, infatti, da uno spessore minore (33 cm) e sono caratterizzate da due filari di mattoni semipieni separati da un'intercapedine d'aria non ventilata, e rivestimenti in intonaco. In corrispondenza del sottofinestra, le pareti esterne presentano una stratigrafia (unico filare di laterizi semipieni e rivestimento in intonaco di calce e sabbia) e uno spessore differenti (27 cm). In Tabella 5 sono riportate le caratteristiche termofisiche dei componenti opachi verticali (partizioni interne comprese).

Tabella 5. Caratteristiche dei componenti edilizi opachi verticali.

Cod	Descrizione	s	U	M_s	Y_{ie}	κ_i
		[m]	[W·m ⁻² K ⁻¹]	[kg·m ⁻²]	[W·m ⁻² K ⁻¹]	[kJ·m ⁻² K ⁻¹]
M1	Parete esterna (piani terra-4)	0,70	0,94	1188	0,02	62,51
M2	Tramezzo interno (1)	0,10	2,25	112	1,72	55,59
M3	Tramezzo interno (2)	0,20	1,55	252	0,63	64,16
M4	Parete esterna (piani 5-6)	0,33	1,25	332	0,390	59,67
M5	Parete esterna sottofinestra	0,27	1,78	379	0,65	67,13

Per quanto riguarda i componenti opachi orizzontali, le strutture disperdenti sono rappresentate dal solaio che separa il piano terra (non oggetto d'analisi) e il primo piano, e il solaio di copertura. Si tratta, in generale, di solai in laterocemento con tavelloni portanti o blocchi in laterizio e sottofondo in calcestruzzo. I solai interpiano sono caratterizzati da spessori di 20 cm (tra il piano terra e i piani 1-4) e di 30 cm (tra gli ultimi due piani), entrambi rivestiti con piastrelle in marmo. La copertura piana calpestabile è invece caratterizzata da

uno spessore di 33 cm e rivestimento in piastrelle in ceramica. In Tabella 6 sono riportate le caratteristiche termofisiche dei componenti opachi orizzontali (partizioni interne comprese).

Tabella 6. Caratteristiche dei componenti edilizi opachi orizzontali.

Cod	Descrizione	s	U	M_s	Y_{ie}	κ_l
		[m]	[W·m ⁻² K ⁻¹]	[kg·m ⁻²]	[W·m ⁻² K ⁻¹]	[kJ·m ⁻² K ⁻¹]
S1	Solaio p. terra/p. primo	0,20	2,27	300	1,07	100,17
S2	Solaio di copertura	0,33	1,82	470	0,57	106,88
S3	Solaio interpiano (piani 1-4)	0,20	1,94	267	1,08	91,95
S4	Solaio interpiano (piani 5-6)	0,30	1,62	397	0,61	96,44

I serramenti presenti nel fabbricato sono composti da telaio in metallo e vetri singoli (per i serramenti dei piani 1-4) e telaio in legno e vetri doppi (per i serramenti dei piani 5 e 6). Le chiusure oscuranti sono invece costituite da persiane in legno a doppia anta, e sono installate su tutte le aperture, ad eccezione delle finestre di dimensione 0,5 × 1,8 m nei piani intermedi. In Tabella 7 vengono riportate le principali caratteristiche dei componenti vetrati.

Tabella 7. Caratteristiche dei componenti edilizi trasparenti.

Cod	Descrizione	A_w	F_F	U_w	U_{w+shut}	$g_{gl,n}$	g_{gl+sh}
		[m ²]	[-]	[W·m ⁻² K ⁻¹]	[W·m ⁻² K ⁻¹]	[-]	[-]
W1	Finestra a due ante con vetro singolo	1,80	0,23	5,979	4,144	0,75	0,75
W2	Finestra a un'anta con vetro singolo	0,90	0,24	6,004	4,168	0,75	0,75
W3	Finestra a due ante con vetro doppio	1,80	0,23	3,188	2,555	0,85	0,85
W4	Finestra a un'anta con vetro doppio	0,90	0,24	3,186	2,556	0,85	0,85
W5	Porta-finestra a due ante con vetro doppio	3,36	0,18	3,120	2,500	0,85	0,85
W6	Finestra a due ante con vetro doppio	2,16	0,20	3,141	2,517	0,85	0,85
W7	Finestra a due ante con vetro doppio	5,17	0,12	3,012	2,413	0,85	0,85

7.1.3 Descrizione dell'impianto

Gli alloggi presenti nell'edificio oggetto di analisi sono serviti da un impianto centralizzato per il riscaldamento; gli spazi comuni, invece, non risultano riscaldati. Il generatore di calore per il riscaldamento è una caldaia localizzata in centrale termica. In particolare, si tratta di un generatore a basamento in ghisa con bruciatore atmosferico ad aria aspirata (Figura 74), con potenza termica utile nominale di 152,5 kW e rendimento termico utile alla potenza nominale di 0,904. I fabbisogni elettrici di generazione sono stati assunti pari a 63 W. La circolazione viene effettuata tramite elettropompe con portata massima di 26,4 m³/h ed assorbimento elettrico pari a 400 W (Figura 75).

Per quanto riguarda il sottosistema di distribuzione, i terminali di emissione sono costituiti da radiatori in ghisa posizionati sulle pareti esterne, e l'impianto di distribuzione è caratterizzato da montanti non isolati

correnti nel lato interno delle pareti esterne. Risulta assente un sistema di regolazione (la regolazione avviene manualmente).



Figura 74. Generatore di calore.



Figura 75. Pompe di circolazione.

7.2 Assunzioni di calcolo e condizioni al contorno per le simulazioni

Si riportano di seguito le assunzioni di calcolo e le condizioni al contorno che sono state fissate per l'esecuzione della simulazione, eseguita su base oraria utilizzando il codice di calcolo sviluppato nell'ambito della ricerca, e, in particolare, il metodo delle funzioni di trasferimento, per il calcolo della trasmissione termica attraverso l'involucro edilizio.

Le assunzioni e condizioni al contorno di seguito riportate sono state concordate con il gruppo di lavoro del Politecnico di Torino, che esegue, sempre nell'ambito del Work package WP.1 Edifici ad alta efficienza,

un'analoga analisi ma utilizzando una metodica differente [63], per consentire un successivo confronto tra le metodologie adottate.

7.2.1 Dati climatici

L'edificio è stato analizzato in tre diverse zone climatiche, corrispondenti a Milano (zona climatica E), Roma (zona climatica D) e Palermo (zona climatica B). I dati climatici per ciascuna delle tre località considerate si riferiscono all'anno climatico tipo su base oraria elaborato dal Comitato Termotecnico Italiano (CTI) nel 2015 [64]. I rispettivi valori medi mensili sono contenuti nella norma UNI 10349-1 [65] e riportati in Figura 76 e in Figura 77.

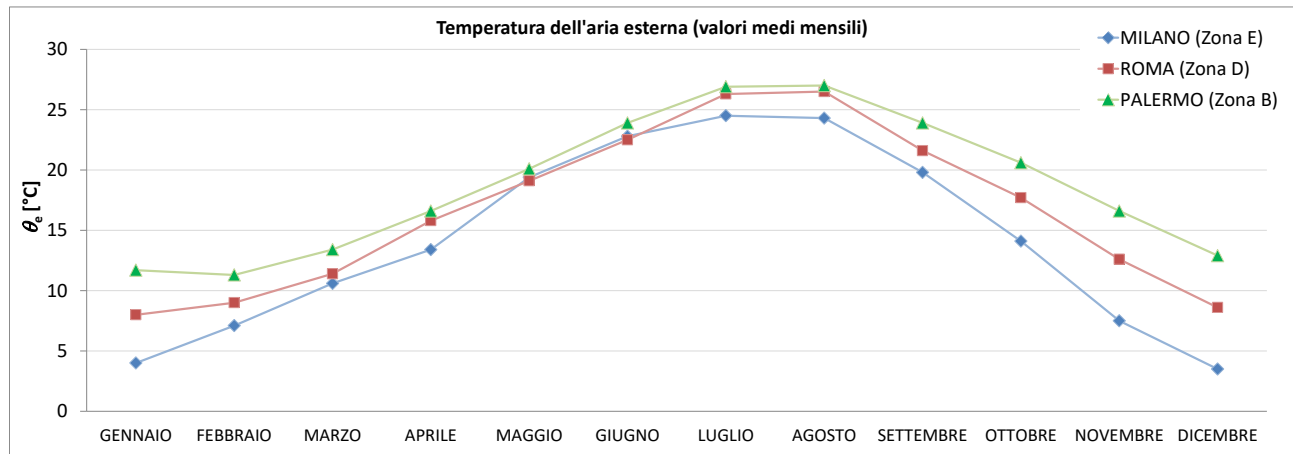


Figura 76. Valori medi mensili della temperatura dell'aria esterna (fonte: UNI 10349-1 [15]).

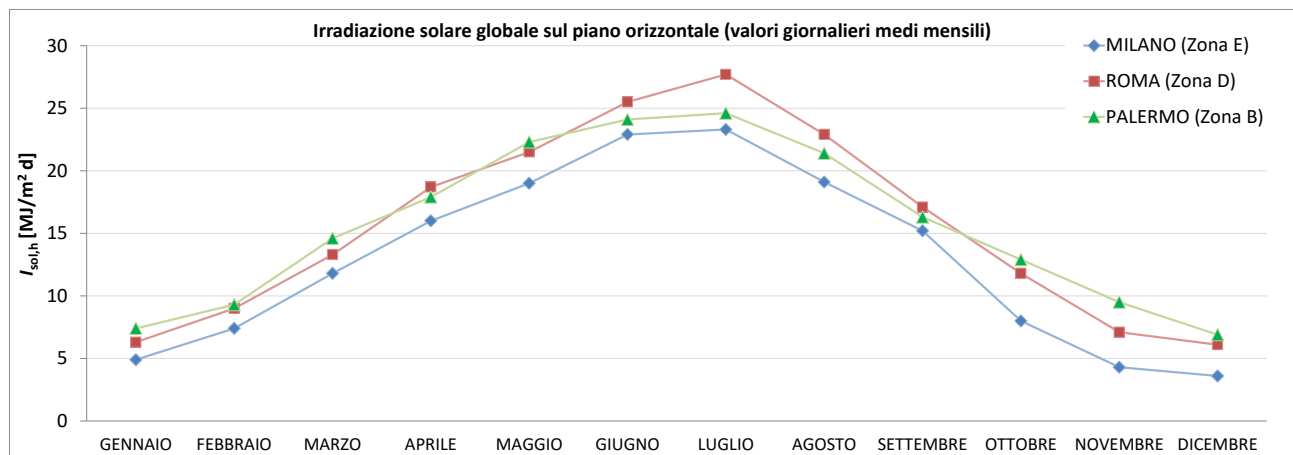


Figura 77. Valori medi mensili dell'irradiazione solare globale sul piano orizzontale (fonte: UNI 10349-1 [15]).

7.2.2 Zonizzazione

L'edificio è stato suddiviso nelle seguenti zone termiche (Figura 78):

- zona termica 1: 1° piano (2° F.T.);
- zona termica 2: dal piano 2° (3° F.T.) al piano 4° (5° F.T.);
- zona termica 3: dal piano 5° (6° F.T.) al piano 6° (7° F.T.).

Il piano terra, che ospita una diversa destinazione l'uso, è stato escluso dall'analisi poiché riscaldato non dall'impianto termico centralizzato, ma da un diverso impianto.

Si riportano in Tabella 8 i principali dati geometrici delle zone termiche.

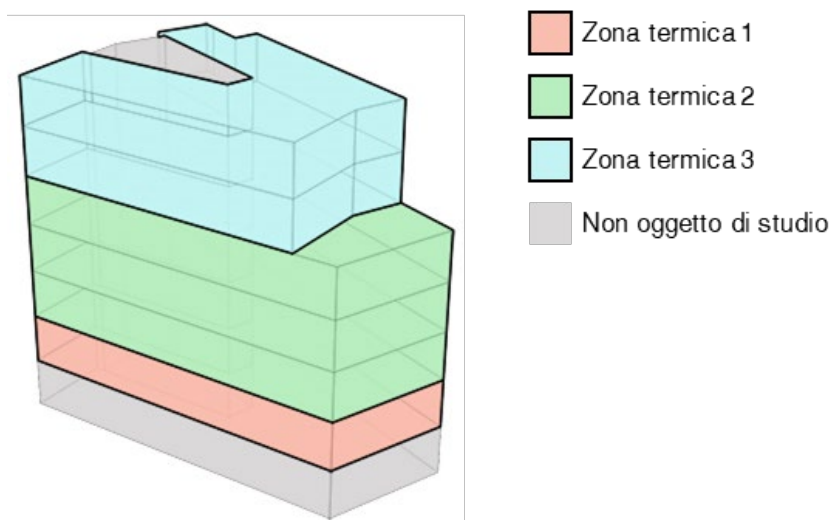


Figura 78. Schematizzazione tridimensionale delle zone termiche nelle quali è stato suddiviso lo spazio climatizzato dell'edificio.

Tabella 8. Quadro dei principali dati geometrici delle zone termiche.

	A_f [m ²]	V_i [m ³]	A_{env}/V_i [m ⁻¹]
Zona termica 1	182	940	0,31
Zona termica 2	541	2 806	0,32
Zona termica 3	368	1 535	0,51

7.2.3 Condizioni al contorno

Il piano terra è stato assunto climatizzato a temperatura costante, come segue:

- 20 °C durante la stagione invernale;
- 26 °C da giugno ad agosto;
- 23 °C nel restante periodo.

Le stesse condizioni al contorno sono state fissate per gli edifici costruiti in aderenza alle testate del caso studio che affacciano su viale dello Scalo di San Lorenzo e via degli Enotri.

Le ostruzioni del contesto circostante sono state modellate geometricamente come indicato nella UNI EN ISO 52016-1.

7.2.4 Profili di utenza

I dati relativi all'utenza sono stati ricavati prevalentemente dalla bozza di Appendice Nazionale della norma UNI EN 16798-1:2019 [66,67]. In Tabella 9 si riporta il quadro d'insieme dei dati d'utenza assunti nel calcolo, con l'indicazione, per ogni parametro, dell'equazione o del criterio per la sua determinazione, il valore numerico (in assenza di equazione), la fonte del dato e alcune note esplicative.

Si riporta in Tabella 10 il profilo di occupazione assunto per l'edificio residenziale [66]. Da Figura 79 a Figura 81 sono rappresentati i profili giornalieri su base oraria, rispettivamente, degli apporti di calore relativi alle sorgenti interne (Figura 79), del tasso di ricambio orario (Figura 80) e del funzionamento dell'impianto di

riscaldamento per ciascuna località, quest'ultimo da utilizzarsi nella valutazione adattata all'utenza (Figura 81). Per semplicità i primi due profili si riferiscono alla zona termica 1; per le altre zone termiche si registrano variazioni trascurabili. Tutti i profili sono stati costruiti secondo quanto indicato in Tabella 9.

Nella valutazione adattata all'utenza (*tailored*) si è assunto che nella stagione di riscaldamento, stabilita in funzione della zona climatica, la temperatura di regolazione sia impostata a 20 °C nelle ore di funzionamento dell'impianto, e a 16 °C in regime di attenuazione. Nella valutazione standard, invece, all'interno della stagione di riscaldamento, l'impianto presenta un funzionamento continuo sulle 24 ore a temperatura di regolazione costante pari a 20 °C.

Difformemente da quanto assunto dal gruppo di lavoro del Politecnico di Torino, la stagione di raffrescamento è stata assunta non bloccata al periodo dal 1° giugno al 31 agosto, ma lasciata libera in modo complementare alla stagione di riscaldamento, sempre con un regime di funzionamento dell'impianto continuo sulle 24 ore e una temperatura di regolazione di 26 °C per tutte le zone climatiche. Tale deviazione risulta giustificata dai risultati delle simulazioni che mostrano in tale mese ancora una significativa richiesta di raffrescamento quando si utilizza il profilo d'uso standard, sia per Palermo che per Roma.

Le chiusure oscuranti installate nell'edificio, che sono persiane in legno a doppia anta, sono considerate in funzione 12 ore al giorno, dalle ore 20:00 alle ore 08:00.

Tabella 9. Quadro riepilogativo dei dati d'utenza per l'edificio residenziale (appartamenti).

	Parametro	Equazione/criterio	Valore	Unità di misura	Fonte	Note
Periodo di utilizzo	Ora del giorno, INIZIO		0	h	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
	Ora del giorno, FINE		24	h	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
	Interruzioni, all'interno dell'intervallo		0	h	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
	Giorni settimanali		7	d	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
	Ore giornaliere		24	h	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
	Ore all'anno		8760	h	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
Occupanti	Numero di persone per appartamento	Per appartamento monolocale o con una sola camera da letto: 2 persone. Per ogni camera da letto aggiuntiva: - se la superficie è inferiore a 14 m ² , 1 persona in più; - se la superficie è uguale o superiore a 14 m ² , 2 persone in più.			UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.19	Per l'edificio oggetto di studio, si ipotizzano 7 persone per piano (2 persone in ciascuno dei due appartamenti con una sola camera da letto, e 3 persone nell'appartamento con una camera da letto aggiuntiva di piccole dimensioni).
	Profilo orario di occupazione				UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
Apporti termici interni	Occupanti - apporti per persona (totale)		118	W	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.20	Da moltiplicare per il numero di persone, secondo il profilo di occupazione.
	Occupanti - apporti per persona (quota sensibile, convettiva)		41	W	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.20	
	Occupanti - apporti per persona (quota sensibile, radiativa)		39	W	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.20	
	Occupanti - apporti per persona (quota latente)		38	W	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.20	
	Occupanti - produzione di vapore per persona	$G_{w,v,oc} = (q_{int,oc,lat}/h_{vap}) \times 3600$	54,7	g/h	UNI EN 16798-1 Appendice Nazionale ^(*) - Equazione A.8 e Tabella A.20	
	Apparecchiature e illuminazione - apporti interni quota sensibile	$q_{int,eq,sen} = 2,2 + 136/A_{cf}$		W/m ²	UNI EN 16798-1 Appendice Nazionale ^(*) - Equazione A.5	Modulato secondo il profilo di occupazione. A_{cf} = superficie di pavimento climatizzata [m ²]
	Apparecchiature - produzione di vapore	$G_{w,v,eq} = 0,32 + 28,8/A_{cf}$		g/(h m ²)	UNI EN 16798-1 Appendice Nazionale ^(*) - Equazione A.6	Modulato secondo il profilo di occupazione. A_{cf} = superficie di pavimento climatizzata [m ²]
Ventilazione	Portata di ventilazione minima per la qualità dell'aria	$q_{tot} = n \cdot q_p + A_{cf} \cdot q_B$, dove: $q_p = 7$ l/(s pers), $q_B = 0$ l/(s m ²)		l/s	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabelle A.11 e A.12	Categoria II dell'ambiente interno. Segue il profilo di occupazione
Condizioni termo-igrometriche di set-point	$T_{op,min}$ - riscaldamento		20	°C	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.2	Categoria II dell'ambiente interno
	$T_{op,max}$ - raffrescamento		26	°C	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.2	Categoria II dell'ambiente interno
	$T_{op,min}$ in regime di attenuazione		16	°C	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
	$T_{op,max}$ in regime di attenuazione		32	°C	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella 26b	
	Umidità relativa minima		40	%	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.13	Categoria II dell'ambiente interno
	Umidità relativa massima		60	%	UNI EN 16798-1 Appendice Nazionale ^(*) - Tabella A.13	Categoria II dell'ambiente interno
Durata giornaliera di accensione dell'impianto	Riscaldamento - Milano (zona E)		14	h/d	D.P.R. 74/2013	Regime attenuato dalle 22:00 alle 06:00 e dalle 13:00 alle 15:00
	Riscaldamento - Roma (zona D)		12	h/d	D.P.R. 74/2013	Regime attenuato dalle 22:00 alle 06:00 e dalle 12:00 alle 16:00
	Riscaldamento - Palermo (zona B)		8	h/d	D.P.R. 74/2013	Regime attenuato dalle 22:00 alle 06:00 e dalle 10:00 alle 18:00
	Raffrescamento (tutte le zone climatiche)		24	h/d	-	Regime di funzionamento continuo
	Riscaldamento (tutte le zone climatiche)		24	h/d	Standard energy rating (UNI/TS 11300-1)	Regime di funzionamento continuo
	Raffrescamento (tutte le zone climatiche)		24	h/d	Standard energy rating (UNI/TS 11300-1)	Regime di funzionamento continuo

(*) Bozza del dicembre 2019

Tabella 10. Profilo di occupazione dell'edificio residenziale (appartamenti) (fonte: UNI EN 16798-1, Appendice Nazionale, bozza del dicembre 2019 [66,67]).

h	Profilo di occupazione - Uso residenziale, appartamenti		
	Giorni feriali	Sabato	Domenica/Festività
1	1,0	1,0	1,0
2	1,0	1,0	1,0
3	1,0	1,0	1,0
4	1,0	1,0	1,0
5	1,0	1,0	1,0
6	1,0	1,0	1,0
7	0,5	0,8	0,8
8	0,5	0,8	0,8
9	0,5	0,8	0,8
10	0,1	0,8	0,8
11	0,1	0,8	0,8
12	0,1	0,8	0,8
13	0,1	0,8	0,8
14	0,2	0,8	0,8
15	0,2	0,8	0,8
16	0,2	0,8	0,8
17	0,5	0,8	0,8
18	0,5	0,8	0,8
19	0,5	0,8	0,8
20	0,8	0,8	0,8
21	0,8	0,8	0,8
22	0,8	0,8	0,8
23	1,0	1,0	1,0
24	1,0	1,0	1,0

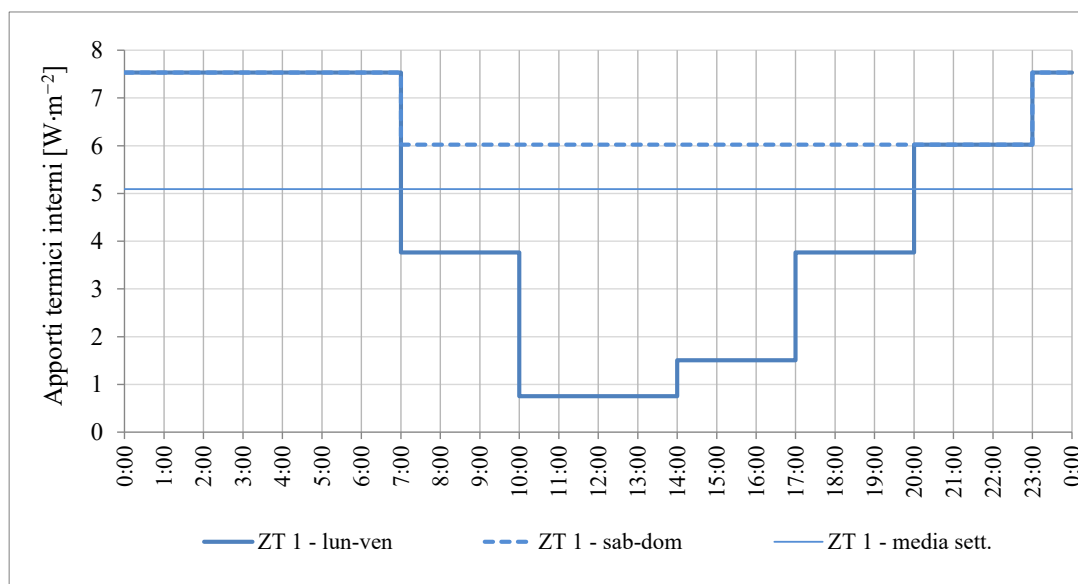


Figura 79. Profilo orario degli apporti interni di calore (quota sensibile) dovuti a occupanti, apparecchiature e illuminazione. Esempio per la zona termica 1.

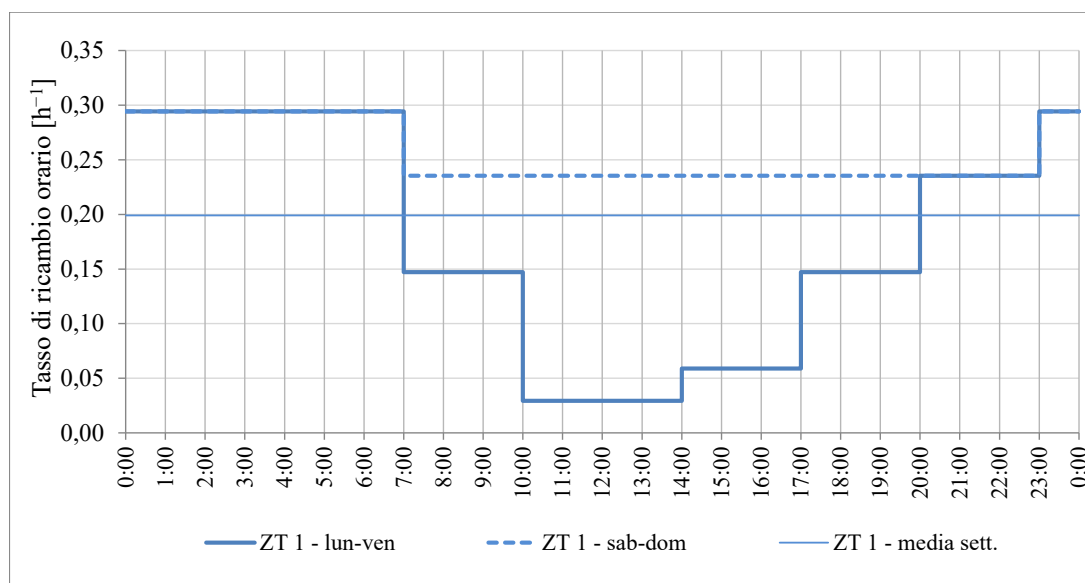


Figura 80. Profilo orario del tasso di ricambio d'aria (ventilazione naturale). Esempio per la zona termica 1.

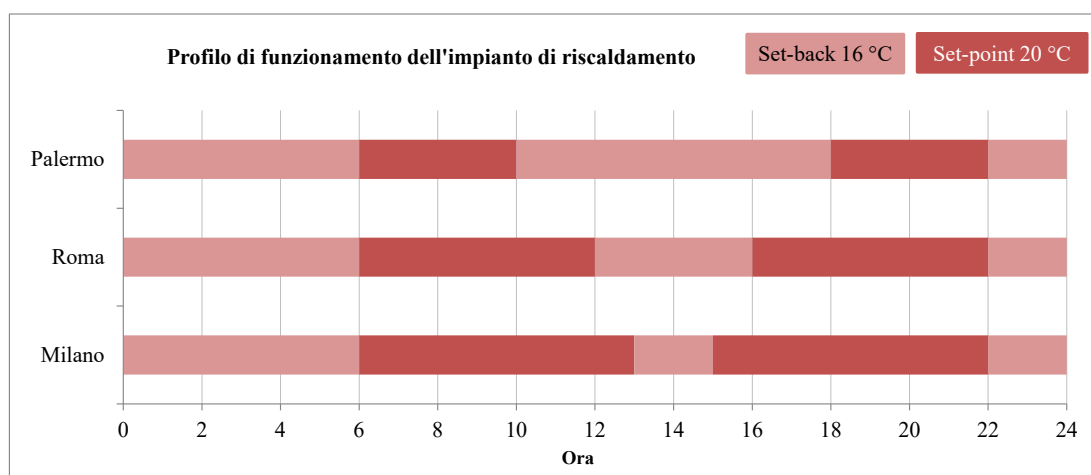


Figura 81. Profilo orario di funzionamento dell'impianto di climatizzazione invernale per le tre zone climatiche.

7.2.5 Semplificazioni

All'interno della modellazione sono state effettuate le seguenti semplificazioni:

- assenza di ponti termici;
- vano scala alla temperatura dell'aria esterna;
- i coefficienti di scambio termico superficiale convettivi sono stati assunti costanti sia all'interno che all'esterno, rispetto al calcolo automatico tramite correlazioni in funzione della temperatura; i valori assunto sono: 4 W/m²K per il coefficiente convettivo interno e 20 W/m²K per quello esterno.
- In questa analisi non è stato preso in considerazione la presenza della caldaia, ma si è valutato esclusivamente il comportamento del fabbricato e calcolato di conseguenza il fabbisogno termico e non quello energetico.

7.3 Risultati preliminari

Nel seguito sono riportati i risultati preliminari delle simulazioni orarie eseguite per le tre zone climatiche identificate con Milano, Roma e Palermo.

Per ognuna delle tre zone climatiche si è determinato il fabbisogno termico sia per il riscaldamento (limitato al periodo di legge) sia per il raffrescamento (possibile nel periodo dell'anno complementare al periodo di riscaldamento), secondo i due profili d'uso descritti: quello standard e quello adattato all'utenza (tailored).

In Figura 82 e Figura 83 sono riportati per le tre zone climatiche i fabbisogni termici ANNUI rispettivamente di riscaldamento e di raffrescamento. Quello che si può notare da quest'analisi preliminare è la prevista significativa riduzione del fabbisogno di riscaldamento al decrescere della latitudine, e piccola riduzione che si ha di tale fabbisogno nel passare dal profilo d'uso standard (riscaldamento attivo 24/24) al profilo tailored con attenuazione notturna. Tale risultato è spiegato dall'elevato tempo caratteristico dell'edificio che presenta una struttura significativamente capacitiva. Se si analizza Figura 83, che riporta il fabbisogno termico per raffrescamento, si osserva simmetricamente un incremento del fabbisogno al decrescere della latitudine per il profilo d'uso standard (raffrescamento attivo 24/24). Di contro, rispetto al riscaldamento, la differenza di richiesta termica è radicalmente diversa quando si considera il profilo tailored. In questo caso si ha una forte riduzione della richiesta termica. Si ricorda che in regime di "attenuazione" notturno il raffrescamento "parte" solo se la temperatura supera i 30°C. Ciò, insieme alle caratteristiche estremamente capacitive dell'edificio oggetto dell'analisi, comporta il superamento di tale temperatura poche volte e quindi una bassa richiesta di energia. Si può inoltre notare un comportamento apparentemente anomalo, nel caso tailored, passando da Roma a Palermo: si va in controtendenza rispetto al caso standard, Palermo richiede meno energia di Roma. Questa, che apparentemente potrebbe sembrare una contraddizione, è probabilmente giustificata da una diversa distribuzione della temperatura dell'aria nei mesi estivi: una temperatura notturna più bassa per Palermo rispetto a Roma, che favorirebbe il night cooling (i ricambi d'aria notturni sono maggiori di quelli diurni) e lo scambio radiativo con il cielo. Per indagare meglio, su tale aspetto, è in programma un'analisi comparativa dei dati climatici orari utilizzati.

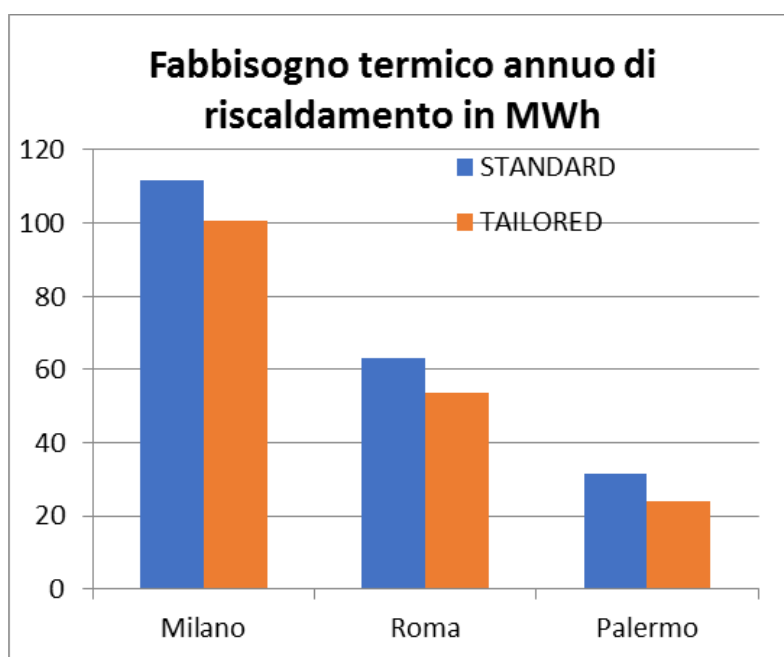


Figura 82. Fabbisogno termico di riscaldamento ANNUO per le tre zone climatiche.

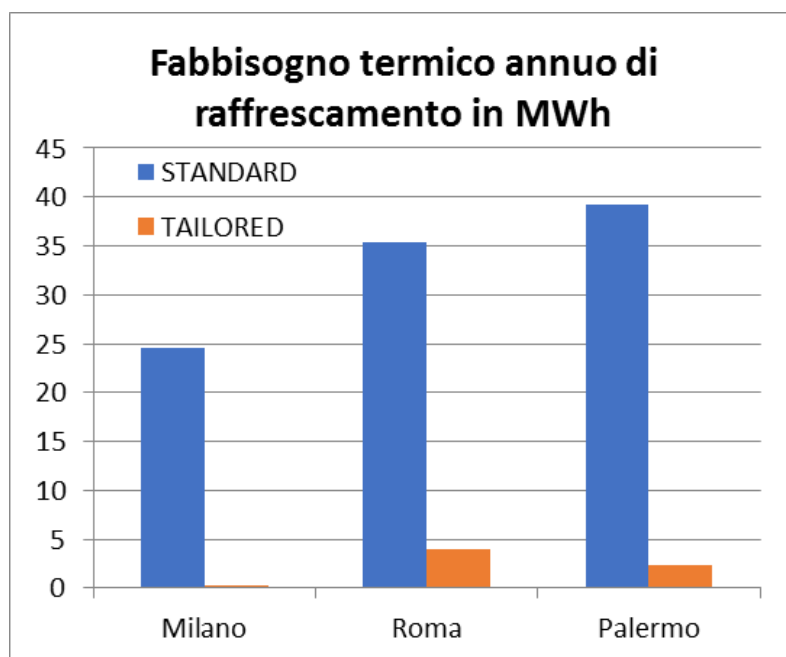


Figura 83. Fabbisogno termico di raffrescamento ANNUO per le tre zone climatiche.

In Figura 84 e Figura 85 sono riportati i dettagli mensili del fabbisogno termico per il riscaldamento ambientale, nella prima riferiti ai profili d'uso standard, nella seconda a quello tailored.

L'andamento mensile rispecchia esattamente quanto già visto a livello annuale. Il dettaglio mensile consente però di vedere come la differenza percentuale tra i due profili d'uso non sia ugualmente distribuita tra i mesi: risulta maggiore in novembre e dicembre. Anche qui, per meglio comprendere tale comportamento occorrerà comparare i dati climatici dei vari mesi tra loro, giacché una possibile giustificazione a tale comportamento può essere, ancora una volta, una significativa differenza tra temperature medie giornaliere ed escursione termica.

In Figura 86 e Figura 87 sono riportati i dettagli mensili del fabbisogno termico per il raffrescamento ambientale, nella prima riferiti ai profili d'uso standard, nella seconda a quello tailored.

L'andamento mensile rispecchia anche qui quanto già visto a livello annuale. Il dettaglio mensile consente però di vedere come, rispetto al profilo d'uso standard, il fabbisogno di Roma risulti in luglio leggermente maggiore di quello di Palermo e in agosto solo leggermente più basso. Palermo ha infine un maggior fabbisogno estivo annuo perché presenta una stagione più lunga. Altra singolarità, in giugno Milano richiede più di Roma e quasi quanto Palermo. Nel caso di profilo d'uso tailored, come già visto sul fabbisogno annuale, Roma richiede più energia per raffrescamento di Palermo, in particolare nel mese di agosto. Anche in questo caso, per meglio comprendere tale comportamento occorrerà comparare i dati climatici dei vari mesi tra loro, giacché una possibile giustificazione a tale comportamento può essere, ancora una volta, una significativa differenza tra temperature medie giornaliere ed escursione termica, che può esaltare fenomeni di night cooling a Palermo rispetto a Roma.

Per completare questa prima analisi preliminare, in Figura 88 e in Figura 89 sono riportati i profili della temperatura interna nella zona intermedia nel mese di maggio, rispettivamente per il profilo d'uso standard e quello tailored. Si può osservare come risultino identici, giacché in tale zona non si raggiunge mai la temperatura di attivazione del raffrescamento (26 °C). Il fabbisogno non nullo in tale mese è dovuto alla zona superiore di caratteristiche termofisiche meno preformanti e con superficie esposta decisamente superiore avendo il tetto come parte del suo involucro.

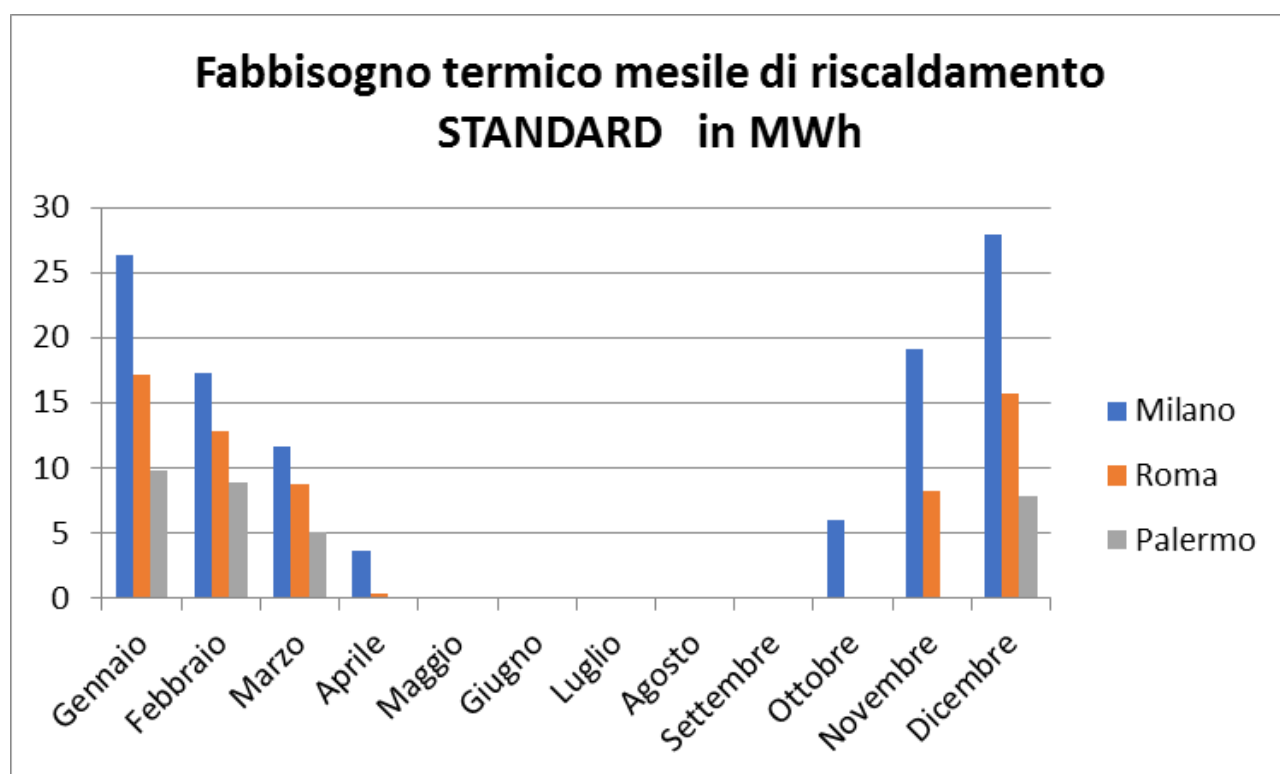


Figura 84. Distribuzione mensile del fabbisogno termico di riscaldamento per le tre zone climatiche in condizioni STANDARD.

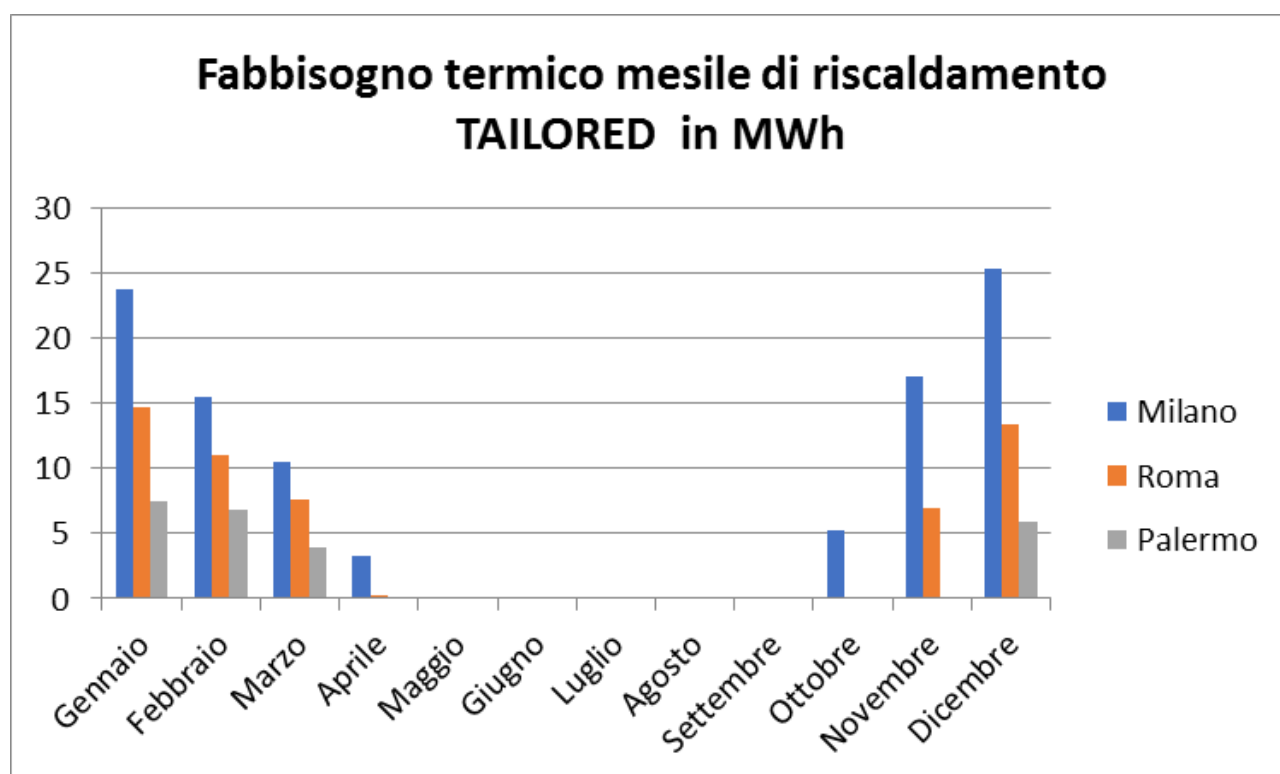


Figura 85. Distribuzione mensile del fabbisogno termico di riscaldamento per le tre zone climatiche in condizioni TAILORED.

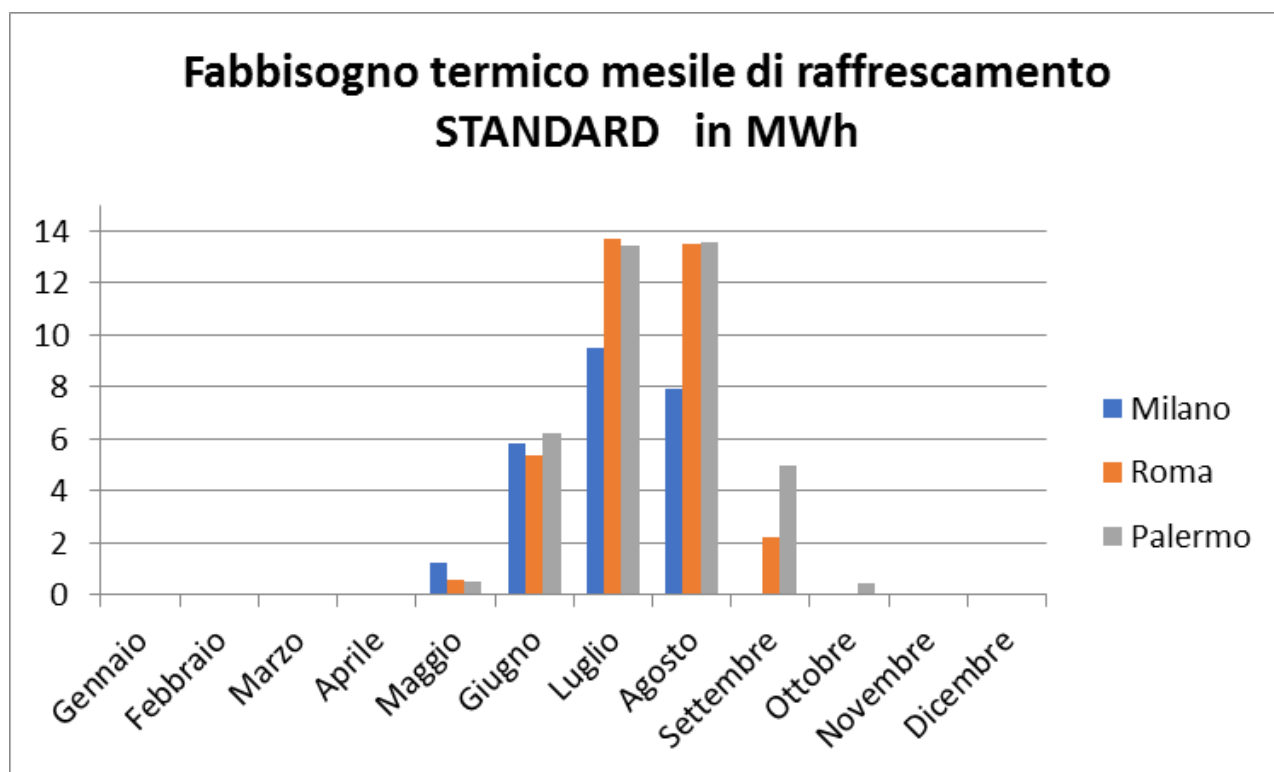


Figura 86. Distribuzione mensile del fabbisogno termico di raffrescamento per le tre zone climatiche in condizioni STANDARD.

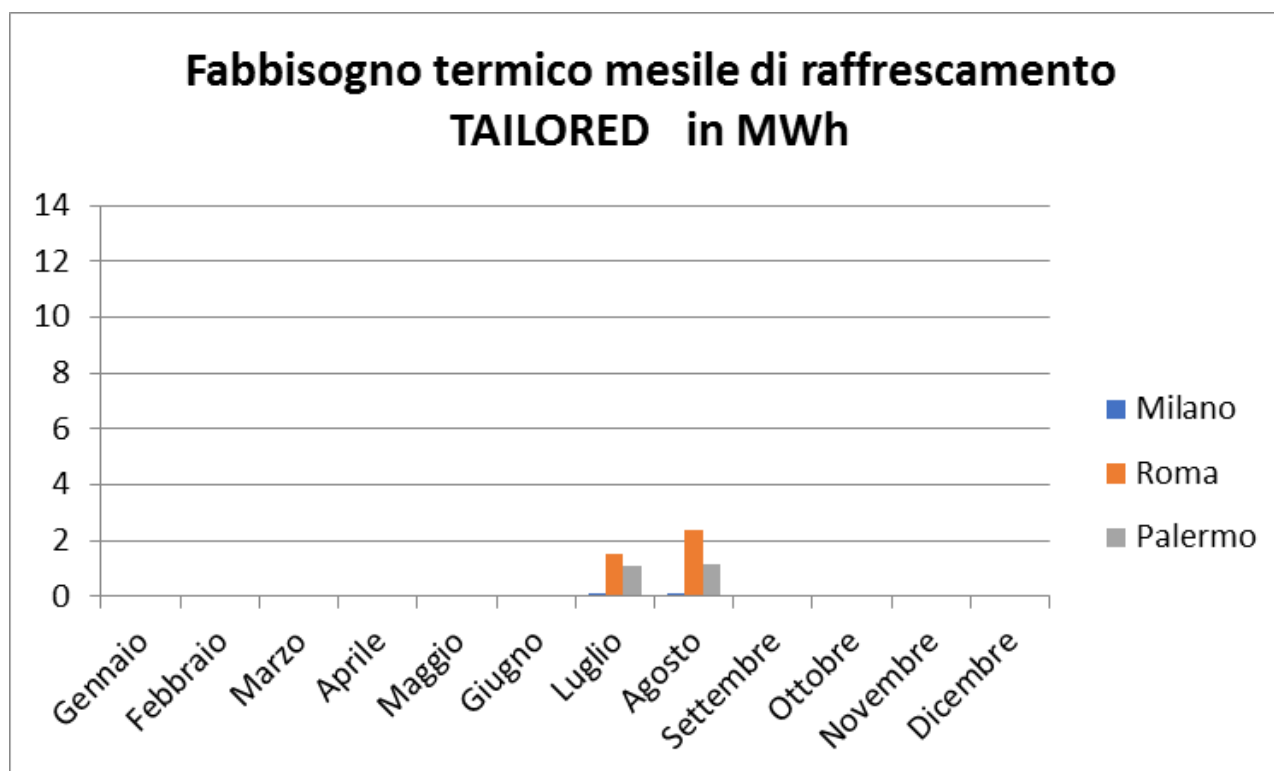


Figura 87. Distribuzione mensile del fabbisogno termico di raffrescamento per le tre zone climatiche in condizioni TAILORED.

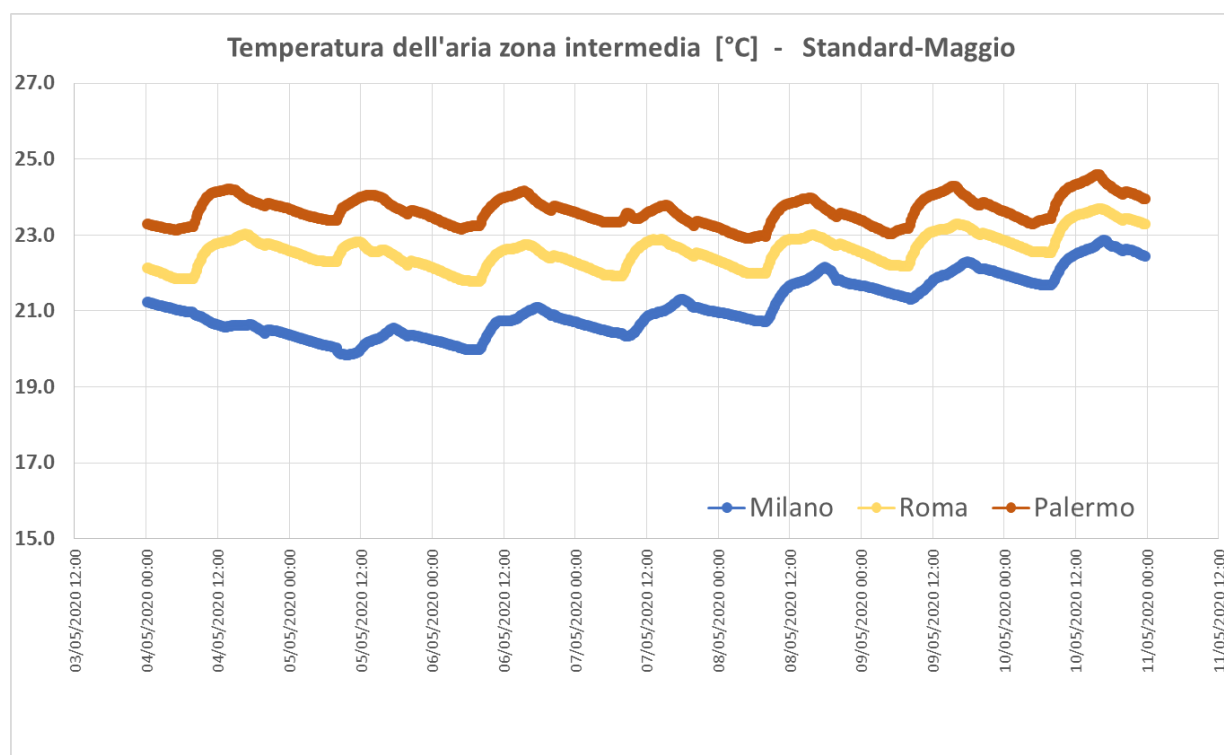


Figura 88. Temperatura dell'aria nella zona intermedia nel mese di maggio per le tre zone climatiche: profilo d'uso STANDARD.

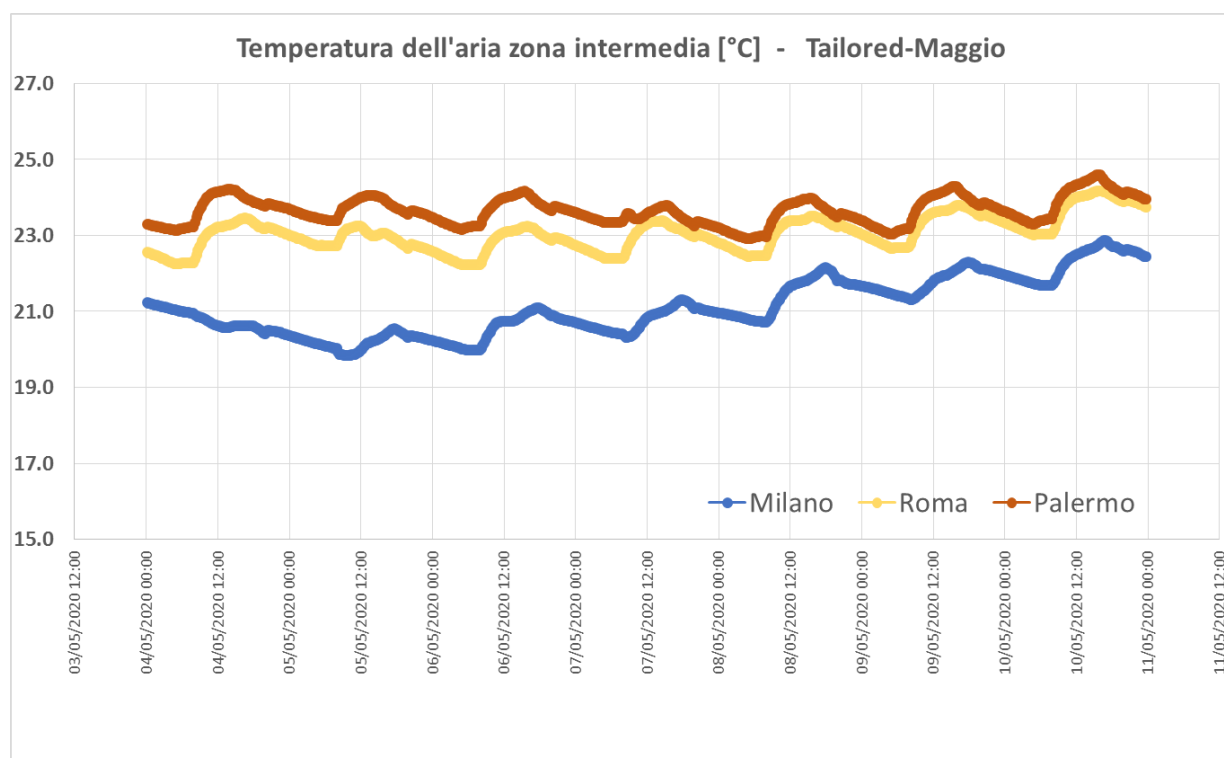


Figura 89. Temperatura dell'aria nella zona intermedia nel mese di maggio per le tre zone climatiche: profilo d'uso TAILORED.

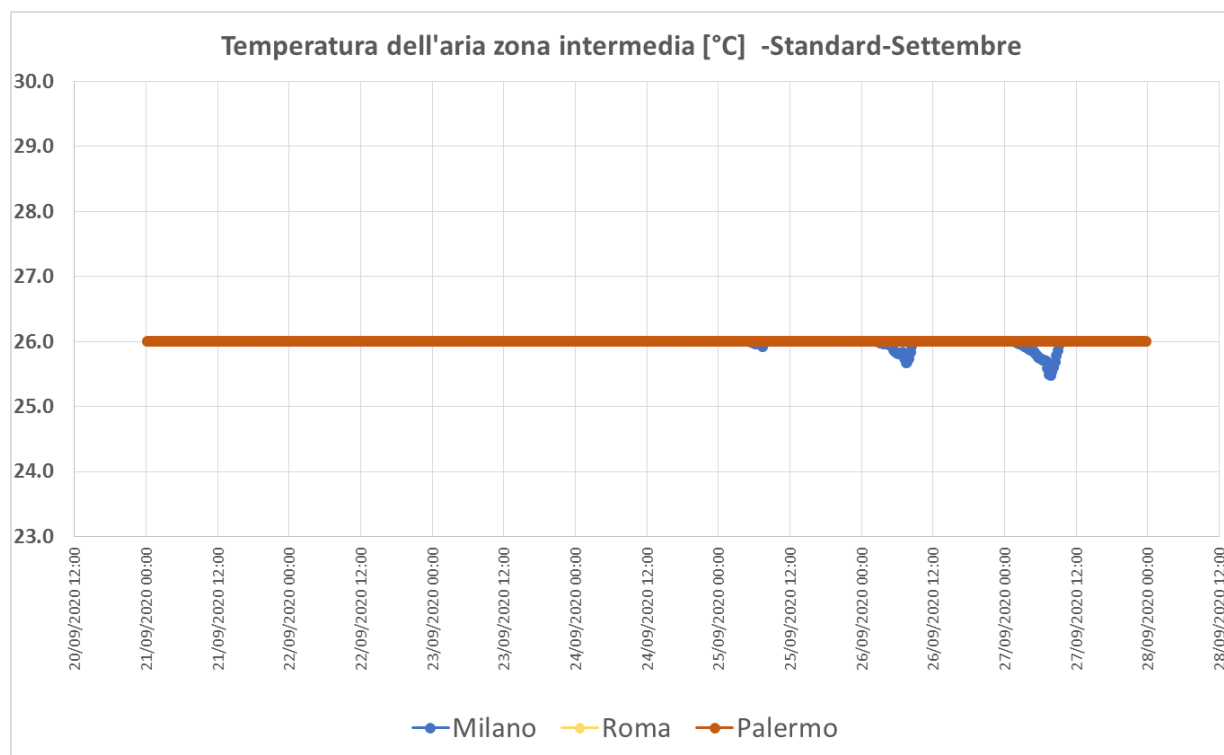


Figura 90. Temperatura dell'aria nella zona intermedia nel mese di settembre per le tre zone climatiche: profilo d'uso STANDARD.

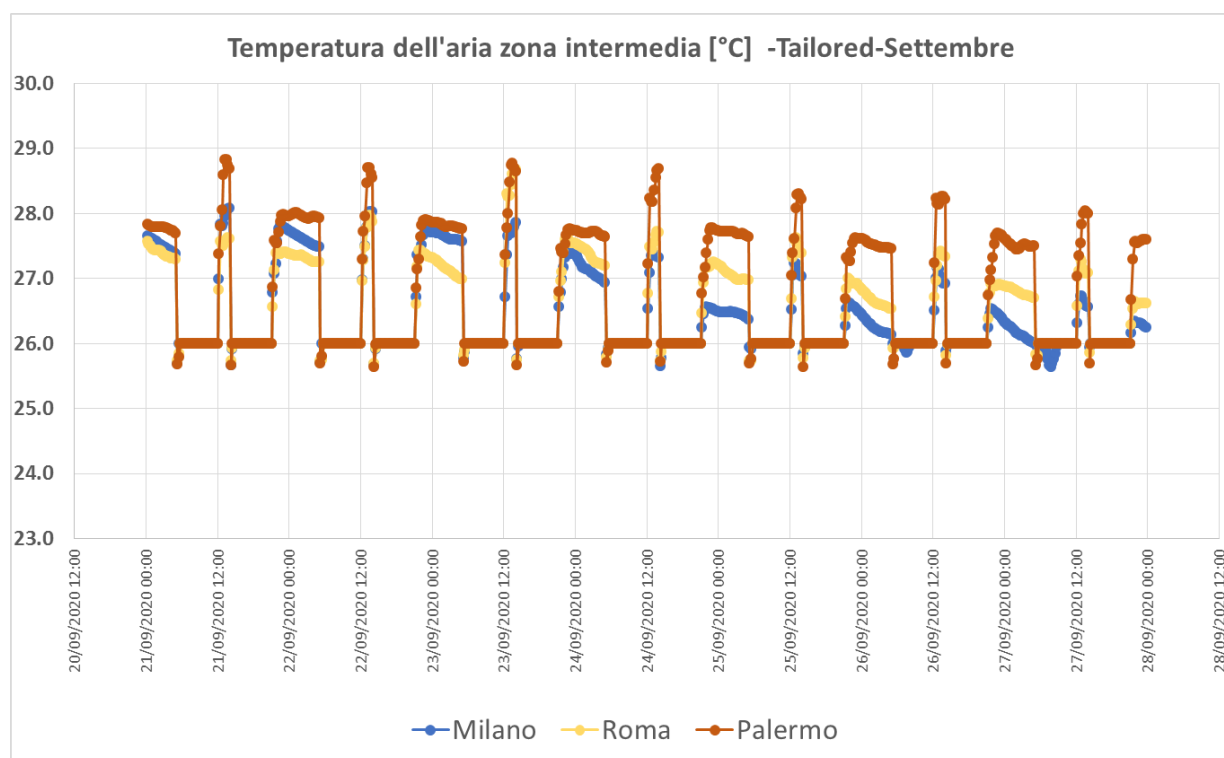


Figura 91. Temperatura dell'aria nella zona intermedia nel mese di settembre per le tre zone climatiche: profilo d'uso TAILORED.

Infine in Figura 90e Figura 91, sono riportati gli analoghi profili per il mese di settembre, dai quali si può notare come, ad esclusione di qualche giorno per Milano, il sistema di raffrescamento deve intervenire con continuità per tutto il mese (profilo d'uso standard). Mentre nel caso di profilo d'uso tailored (riaccensione notturna solo se la temperatura interna supera i 30 °C), si osserva come ci sia un trend discendente, giorno dopo giorno, nel decadimento notturno della temperatura interna, più marcato per Milano, un po' meno per Roma, e quasi assente per Palermo, coerentemente con l'andamento climatico diverso dei tre siti considerati. In tutti e tre i casi comunque la temperatura interna non raggiunge mai i 30°C nel periodo di attenuazione e ciò giustifica la forte riduzione del fabbisogno rispetto al caso di profilo d'uso standard.

8 Conclusioni

Questo rapporto rappresenta solo il primo anno di attività del progetto di ricerca mirante a realizzare un codice di calcolo per la simulazione dinamica delle prestazioni energetiche del sistema edificio che, pur consentendo un'analisi dettagliata e di qualità (base di calcolo oraria e sub-oraria), sia utilizzabile da utenti con gradi diversi di competenza, facilmente integrabili in altri strumenti di progettazione, ad esempio nei BIM, e che possa evitare una grande richiesta di dati all'utente, tramite l'impiego di informazioni provenienti direttamente dai costruttori di componenti.

Per realizzare tale obiettivo, questo primo anno di ricerca si è focalizzato sulla definizione della struttura generale del codice, del linguaggio di programmazione, sui supporti alla stessa (scelta di ambiti e supporti open source e "free-to-use"), sull'implementazione di una logica ad oggetti facilmente parallelizzabile, in modo da realizzare uno strumento proiettato verso le future piattaforme informatiche e di calcolo e facile da mantenere e aggiornare.

Dopo varie analisi e lo studio delle caratteristiche dei maggiori codici di simulazione oggi esistenti a livello mondiale, EnergyPlus e TRNSYS, è stato scelto come ambiente di sviluppo inter-piattaforma (cioè compresente su tre maggiori sistemi operativi (Windows, Linux e IOS), il framework dot-Net (.Net) con il suo linguaggio di programmazione ad oggetti, C#.

Dopo la scelta degli strumenti per lo sviluppo del codice, che ne potessero assicurare la funzionalità e longevità, si è iniziato lo sviluppo vero e proprio dello stesso, sia in termini di architettura informatica, sia in termini di algoritmi per la "simulazione dinamica" delle prestazioni dei componenti dell'involucro edilizio, o meglio del fabbricato, intendendo con questo termine l'involucro più tutte le sue strutture interne, che ovviamente partecipano alla definizione della sua prestazione.

Nello spirito dell'obiettivo, più volte sottolineato, di realizzare uno strumento multi-obiettivo, in grado cioè di soddisfare sia le esigenze di analisi avanzata e ricerca, che l'uso ai fini di verifiche convenzionali di legge, la scelta e implementazione nel codice di modelli dei fenomeni fisici di interesse è stato il secondo focus dell'attività svolta nel primo anno.

La prima scelta fondamentale è stata quella di costruire un codice che si basi sulla "geometrizzazione" dell'edificio, cioè l'edificio è descritto nella sua spazialità tramite le relazioni geometriche tra i suoi componenti. La ragione di questa scelta è semplice: di contro a una maggiore complessità nell'introduzione dell'informazione in termini di relazioni geometriche, nel breve termine superabile con l'interazione con il BIM, si ha la possibilità di trattare in modo più rigoroso i fenomeni fisici che avvengono nell'ambiente, estendendo la possibilità, per lo stesso codice di calcolo, di trattare nel futuro anche altre tipologie di prestazione, come ad esempio tra tutte quella illuminotecnica o caustica, che richiedono modelli geometrici dell'edificio.

Altra scelta importante è stata, dopo un'accurata analisi comparativa tra soluzioni diverse, quella di implementare più modelli e, quindi, solutori per la simulazione di stessi fenomeni fisici, in modo da dare all'utente, secondo le sue abilità, la possibilità di poter utilizzare lo strumento ritenuto al uopo più adeguato. Così sono stati, ad esempio, analizzati e implementati più modelli per il trattamento della radiazione solare, che forniscono gradi di accuratezza differenti con impegni di calcolo leggermente diversi. Allo stesso modo sono stati implementati modelli diversi per la determinazione della conduzione termica in regime variabile

nelle partizioni dell'edificio: quadripolo armonico, funzioni di trasferimento e differenze finite, con diverse possibili combinazioni di schemi.

La radiazione termica (onde lunghe), sfruttando la natura geometrica del modello dell'edificio è modellata usando i fattori di forma per radiazione isotropa e i fattori di radiazione mutua per corpo griccio, approccio che consente poi di modellare in modo geometrico anche la radiazione solare entrante dalle finestre e ridistribuirla in modo più realistico della semplice ipotesi di isotropia.

Per la zona termica, l'equazione fondamentale di bilancio termico, chiamata brevemente equazione del nodo aria, è stata modellata in modi differenti, anche qui, per dare la possibilità di calcoli più accurati ma più pesanti da eseguire, e soluzioni più rapide ma meno precise. Si dispone di due solutori analitici, uno a parametri costanti, l'altro a parametri funzioni lineari del tempo, e di diversi solutori numerici, basati di fatto su schemi alle differenze finite.

Tutti i vari modelli, schemi e algoritmi implementati sono stati accuratamente verificati tramite controlli di consistenza, soluzioni al limite, comparazione con soluzioni analitiche "esatte", in modo da assicurare che ogni mattoncino dell'intera impalcatura fosse di per se verificato. La documentazione di tutti i singoli test è in via di sviluppo e sarà parte degli obiettivi del terzo anno di ricerca, la realizzazione di una documentazione esaustiva a supporto dello sviluppo e manutenzione del codice.

Quello che invece è stato riportato nel presente rapporto è la validazione del codice di calcolo come insieme, tramite il BESTEST (standard ASHRE 140), che ha dimostrato come OpenBPS si comporti esattamente come la media dei maggior codici di simulazione dinamica per il sistema fabbricato, cioè l'edificio senza il controllo dell'impianto in regime libero.

In ultimo è stata fatta un'applicazione preliminare del codice di calcolo ad un edificio tipologicamente significativo del parco edilizio nazionale, un edificio residenziale massivo, con il duplice scopo di evidenziare le diverse previsioni che il calcolo dinamico può indicare in funzione di versi criteri di gestione dell'edificio, e di analizzare le differenze che si hanno tra modelli e approcci diversi. A tale scopo il modello di edificio, fornito da ENEA, è stato il più possibile concordato e coordinato tra le tre diverse unità di ricerca coinvolte nel progetto, in modo da poter produrre dati facilmente confrontabili, per le future analisi comparative.

Un primo risultato, in tal senso, è stato già raggiunto dall'evidenza che, per il modello oggetto di questa ricerca, OpenBPS, il considerare un regime di attenuazione notturna, invece che considerare il set point fisso 24 ore su 24, comporta: per il riscaldamento differenze piccole e sostanzialmente indipendenti dalla latitudine del sito; per il raffrescamento differenze sostanziali e non sempre strettamente dipendenti dalla latitudine. È evidente che nel caso del raffrescamento la grande capacità termica della struttura dell'edificio recita un ruolo da comprimario, che invece non ha nel riscaldamento. E quindi come sia necessario avere un codice di calcolo dinamico per avere una buona stima del fabbisogno termico per il raffrescamento.

In conclusione, in questo primo anno di ricerca, è stata realizzata quella parte di codice di calcolo che consente la determinazione dinamica del fabbisogno termico di un edificio, con livelli di accuratezza variabili a scelta dell'utente. Il codice è stato validato e utilizzato per delle preanalisi su un edificio campione.

Nei successivi due anni della ricerca, verrà portato avanti lo sviluppo dei componenti d'impianto, in modo da poter simulare al termine del secondo anno un edificio campione comprensivo di impianto di climatizzazione, e, nel terzo anno, il protocollo contenente la procedura di sviluppo, validazione e gestione in ambito Open Source, compressivo di Public EU Open Source License, il sito web del progetto OpenBPS. Il sito Repository del codice sorgente, che consenta lo sviluppo collaborativo tramite Git (software di controllo versione), in modo da poter rendere disponibile a chiunque fosse interessato una versione dimostrativa del software OpenBPS, scaricabile dal repository, costituita dalle librerie OpenBPS e un'interfaccia utente semplificata.

9 Riferimenti bibliografici

1. Parlamento Europeo, "Direttiva 2002/91/EC del Parlamento europeo e del Consiglio del 16 dicembre 2002 sul rendimento energetico nell'edilizia", Gazzetta ufficiale delle Comunità europee. L 1/65, 04.1.2003.
2. Parlamento Europeo, "Direttiva 2010/31/UE del Parlamento europeo e del Consiglio del 19 maggio 2010 sulla prestazione energetica nell'edilizia (rifusione)", Gazzetta ufficiale delle Comunità europee. L 153/13, 18.6.2010.
3. Parlamento Europeo, "Direttiva 2018/844/UE del Parlamento europeo e del Consiglio del 30 maggio 2018 che modifica la direttiva 2010/31/UE sulla prestazione energetica nell'edilizia e la direttiva 2012/27/UE sull'efficienza energetica", Gazzetta ufficiale delle Comunità europee. L 156/75, 19.6.2018.
4. Presidente della Repubblica, "Decreto Legislativo 19 agosto 2005, n. 192. Attuazione della direttiva 2002/91/CE relativa al rendimento energetico nell'edilizia", Gazzetta Ufficiale n. 222 del 23 settembre 2005 - suppl. ord. n. 158.
5. Presidente della Repubblica, "Decreto-legge 4 giugno 2013, n. 63, Disposizioni urgenti per il recepimento della Direttiva 2010/31/UE del Parlamento europeo e del Consiglio del 19 maggio 2010, sulla prestazione energetica nell'edilizia per la definizione delle procedure d'infrazione avviate dalla Commissione europea, nonché altre disposizioni in materia di coesione sociale. (13G00107)", Gazzetta Ufficiale n.130 del 5-6-2013
6. Presidente della Repubblica, "LEGGE 3 agosto 2013, n. 90. Conversione in legge, con modificazioni, del decreto-legge 4 giugno 2013, n. 63, recante disposizioni urgenti per il recepimento della Direttiva 2010/31/UE del Parlamento europeo e del Consiglio del 19 maggio 2010, sulla prestazione energetica nell'edilizia .. ", Gazzetta Ufficiale Serie Generale n.181 del 03-08-2013.
7. IBPSA-USA, "BEST Directory - Building Energy Software Tools", disponibile presso: <http://www.buildingenergysoftwaretools.com/> [ultimo accesso: novembre, 2019].
8. Attia S., Hensen J.L.M., Bertran L., De Herde A., "Selection criteria for building performance simulation tools: contrasting architects and engineers needs", Journal of Building Performance Simulation, 5(3), 155-169.
9. Mazzarella, L.; Pasini, M.; Hoonejani Shahmandi N., "Challenges, Limitations, and Success of Cloud Computing for Parallel Simulation of Multiple Scenario and Co-Simulation", 2014 ASHRAE/IBPSA-USA Building Simulation Conference Atlanta, GA September 10-12, 2014.
10. IEA, "EBC Annex 58. Reliable Building Energy Performance Characterisation Based on Full Scale Dynamic Measurements", web reference: <https://www.iea-ebc.org/projects/project?AnnexID=58>
11. Strachan, P., Svehla, K., Heusler, I. and Kersken, M., "Whole model empirical validation on a full-scale building", Journal of Building Performance Simulation", Volume 9, 2016 - Issue 4, pp 331-350.
12. Kendall, M., and J. D. Gibbons, "Rank Correlation Methods", 5th ed. (1990), pp.69-77. London: Edward Arnold.
13. ASHRAE, "ASHRAE Standard 140-2017. Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs", (2017), ASHRAE, Atlanta, GA.
14. Fischer, G., McCall, R., Ostwald, J., Reeves, B., Shipman, F., "Seeding, evolutionary growth and reseeding", Proc. of the SIGCHI conference on Human factors in computing systems, 1994, Boston, United States, p.292-298.
15. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G., Mehendjiev, N., "Meta-Design: A Manifesto for End-User Development", Communications of the ACM, vol. 47(9) (2004), pp.33-37.

16. Sahlin, P. & Sowell, E.F., "A neutral format for building simulation models", Proc. of IBPSA '89 Conference, 1989, Vancouver, Canada, pp.147-154.
17. Zupančič, B., Sodja, A., "Object oriented modelling of variable envelope properties in buildings", WSEAS Transactions on Systems and Control, vol. 3(12) (2008), 1046-1056
18. Hensen, J., "A comparison of coupled and de-coupled solutions for temperature and air flow in a building", in ASHRAE Transactions vol. 105:2 (1999), pp. 962-969.
19. Zhai, Z., "Developing an Integrated Building Design Tool by Coupling Building Energy Simulation and Computational Fluid Dynamics Program", PhD thesis, 2004, Massachusetts Institute of Technology.
20. Mazzarella L., Pasini M., "Building energy simulation and object-oriented modelling: review and reflections upon achieved results and further developments", Proceedings of 11th International IBPSA Conference, 2009, Glasgow, 638-645.
21. Dawkins, R., "The Blind Watchmaker", W.W. Norton, 1987, New York, USA.
22. Fischer G., "Seeding, Evolutionary Growth, and Reseeding: Constructing, Capturing, and Evolving Knowledge in Domain- Oriented Design Environments", Automated Software Engineering, October 1998, Volume 5, Issue 4, pp 447–464
23. Sahlin P., "Modelling and simulation methods for modular continuous systems in buildings", PhD thesis, 1996, Department of Building Sciences, Royal Institute of Technology, Stockholm.
24. Wetter, M., "Simulation-Based Building Energy Optimization", PhD thesis, 2004, University of California, Berkeley, USA.
25. Clarke J.A., Hand, J.W., Hensen, J.L.M, Johnsen, K., Wittchen, K., Madsen, C. and Compagnon, R., "Integrated Performance Appraisal of Daylight-Europe Case Study Buildings", Proc. 4th European Conference on Solar Energy in Architecture and Urban Planning, 1996, Bedford, UK
26. Hand, J.W., "Removing Barriers To The Use Of Simulation In The Building Design Professions", PhD thesis, Energy System Research Unit, Department of Mechanical Engineering, 1998, University of Strathclyde, Glasgow, UK.
27. Citherlet S., "Towards the Holistic Assessment of Building Performance Based on an Integrated Simulation Approach", PhD thesis, Architecture Department, 2001, Swiss Federal Institute of Technology (EPFL), Lausanne.
28. Prazeres, L.M.R., "An Exploratory Study about the Benefits of Targeted Data Perceptualisation Techniques and Rules in Building Simulation", PhD thesis, Department of Mechanical Engineering, 2006, University of Strathclyde, Glasgow, UK.
29. Katz, N.C., "Parametric Modeling in AutoCAD", AECbytes Viewpoint #32 (May 16, 2007).
30. <http://www.bentley.com/en-US/Markets/Building/GenerativeComponents/default.htm>, ultimo accesso Ottobre 2019.
31. Beringer, J., Fischer, G., Mussio, P., Myers, B., Paternò, F., de Ruyter, B., "The next challenge: from easy-to-use to easy-to-develop", Proceedings of the Conference on Human Factors in Computing Systems 2008, Florence, Italy, pp. 2257-2260.
32. Augenbroe, G. & Malkawi, A. (Eds.), "Advanced building simulation", Spon Press, 2004, New York and London.
33. Mazzarella L., Martina P., "Development of a new tool for the co-simulation of multiple autonomous object". In "Building Simulation 2013, Proceedings of BS2013: 13th Conference of International Building Performance Simulation Association", Etienne Wurtz Ed., Chambéry, France, August 26-28 2013 , (pp. 3794- 3801), ISBN 978-2-7466-6294-0
34. Perez, R., P. Ineichen, R. Seals, J. Michalsky and R. Stewart. Modeling Daylight Availability and Irradiance Components from Direct and Global Irradiance. Solar Energy 44 (1990), 271-289.
35. Wilcox S. and Marion W., "Users Manual for TMY3 Data Sets", Technical Report NREL/TP-581-43156 Revised May 2008, Golden, Colorado.
36. Grena R., "Five new algorithms for the computation of sun position from 2010 to 2110", Solar Energy 86 (2012), 1323-1337.
37. Iqbal M., "An Introduction to Solar Radiation", 1983, Academic Press, Elsevier Inc. I
38. Spencer, J. W., "Fourier Series Representation of the Position of the Sun.", Search, 2 (5), (1971), 172

39. Duffie J.A. & Beckman W.A., "Solar Engineering of Thermal Processes", 4th Ed., 2012, John Wiley & Sons Inc
40. Vignola F., Michalsky J., Stoffel T., "Solar and infrared radiation measurements", 2012, CRC press
41. Reda I., Andreas A., "Solar position algorithm for solar radiation applications", Solar Energy 76, 2004, 577–589.
42. Michalsky J., "The Astronomical Almanac's algorithm for approximate solar position (1950-2050)", Solar Energy Vol , 40, No 3, pp. 227-235, 1988
43. U.S. Committee on Extension to Standard Atmosphere, "U.S. Standard Atmosphere", U.S. Gov't. Printing Office, Washington, DC (1962).
44. Mac Oglesby, Sat, 7 Feb 1998 15:49:41 -0500 (EST) E-mail retrieved from <https://www.mail-archive.com/search?l=sundial@uni-koeln.de&q=subject:%22Fourier+paper%22&o=newest&f=1>
45. Hottel H. C., "A Simple Model for Estimating the Transmittance of Direct Solar Radiation Through Clear Atmospheres.", Solar Energy, 18, 1976, pp.129.
46. Liu B. Y. H. and Jordan R. C., "The Interrelationship and Characteristic Distribution of Direct, Diffuse and Total Solar Radiation.", Solar Energy, 4 (3), 1960, pp.1.
47. Perez, R. et al, "A new simplified version of the Perez diffuse irradiance model for tilted surfaces", Solar Energy 39, 3, 1987, pp 221-231.
48. Perez, R. et al, "Modelling daylight availability and irradiance components from direct and solar irradiance", Solar Energy 44,5, 1990, pp 271-289.
49. Knight P.A., Ruiz D., Uçar B., "A Symmetry Preserving Algorithm for Matrix Scaling", SIAM Journal on Matrix Analysis and Applications, Society for Industrial and Applied Mathematics, 2014, 35 (3), pp.25.
50. Hubbard, J. H., & Hubbard, B. B., "Vector calculus, linear algebra, and differential forms: a unified approach.", Second edition , 2002, ISBN 0-13-041408-5, Prentice Hall
51. Martin M. and Berdahl P., "Characteristics of infrared sky radiation in the United States," Sol. Energy **33**, 1984, 321–336.
52. Clark G and Allen C., "The estimation of atmospheric radiation for clear and cloudy skies.", in: Proceedings of the 2nd national passive solar conference (AS/ISES). 16–18 March 1978, pp.675–678, University of Pennsylvania
53. Quarteroni, A. "Numerical models for differential problems" Vol. 2, 2014, Springer Science & Business Media
54. Mitalas G.P. and Stephenson D.G., "Room thermal response factors", ASHRAE Transactions, Vol.73, part I, 1967
55. Stephenson, D. G. and Mitalas, G. P., "Cooling Load Calculations by Thermal Response Factor Method", ASHRAE Transaction, Vol. 73, Part I, 1967.
56. Stephenson, D. G. and Mitalas, G. P., "Calculation of heat conduction transfer functions for multi-layer slabs", ASHRAE Transaction, Vol. 77, Part. 2, 1971, pp. 117-126
57. ASHRAE Handbooks - Fundamentals Handbook (SI Edition), 1997, ASHRAE
58. Seem J.E. "Modeling of Heat Transfer in Buildings", Ph.D. Thesis, Department of Mechanical Engineering, University of Wisconsin-Madison, 1987.
59. Delcroix B., Kummer M., Daoud A., Hiller M., "Improved Conduction Transfer Functions Coefficients Generation in TRNSYS Multizone Building Model", 13th Conference of IBPSA, 2013, Chambéry, France, August 26-28
60. Mazzarella L. e Pasini M., "CTF vs FD based numerical methods: accuracy, stability and computational time's comparison", Energy Procedia, Volume 78, 2015, Pages 2620–2625
61. Mitalas, G. P and Arseneault J.G., "Fortran IV Program to Calculate z-Transfer Functions for the Calculation of Transient Heat Transfer through Walls and Roofs", 1972, National Research Council Canada, Ottawa
62. Mazzarella L., "The air energy balance equation paradox", Proceedings of Building Simulation Applications BSA 2013, 1st IBPSA Italy conference, Bolzano, 30th January –1st February 2013, pages 15-28, ISSN 2531-6702
63. UNI, "UNI EN ISO 52016-1:2018 - Prestazione energetica degli edifici - Fabbisogni energetici per riscaldamento e raffrescamento, temperature interne e carichi termici sensibili e latenti - Parte 1: Procedure di calcolo", UNI, 2018, Milano.

64. Murano G., "La serie UNI 10349: nuova norma per il calcolo energetico – L'aggiornamento dei dati climatici", Ingenio, Novembre 2016
65. UNI, "UNI 10349-1:2016 - Riscaldamento e raffrescamento degli edifici - Dati climatici - Parte 1: Medie mensili per la valutazione della prestazione termo-energetica dell'edificio e metodi per ripartire l'irradianza solare nella frazione diretta e diffusa e per calcolare l'irradianza solare su di una superficie inclinata", UNI, 2016, Milano.
66. UNI, "UNI EN 16798-1:2019 - Prestazione energetica degli edifici - Ventilazione per gli edifici - Parte 1: Parametri di ingresso dell'ambiente interno per la progettazione e la valutazione della prestazione energetica degli edifici in relazione alla qualità dell'aria interna, all'ambiente termico, all'illuminazione e all'acustica - Modulo M1-6", UNI, 2019, Milano.
67. CTI, "Appendice nazionale alla UNI EN 16798-1:2019 – Bozza dicembre 2019", Comitato Termotecnico Italiano, 2019, Milano

10 Abbreviazioni e acronimi

AADCC	A ccuracy and A bility of tools to simulate D etailed and C omplex building C omponents
AE	A lgebraic E quation
B.C.	B ackward D ifference
BC	B oundary C onditions
BCVTB	B uilding C ontrols V irtual T est B ed
BPSt(s)	B uilding P erformance S imulation t ool(s)
C.D.	C entral D ifference
CFD	C omputational Thermo- F luid D ynamics
CLR	C ommon L anguage R untime
CPU	C entral P rocess U nit
CTR	C onduction T ransfer F unction
DF	D ifferenze F inite
DLL	D ynamically L inked L ibrary
DRF	D irect R oot F inding
EUD.	F orward D ifference
F.E.	Full E xplicit
F.I.	Full I mplicit
FNU	F unctional M ock-up U nits
gmst	G reenwich m ean s idereal t ime
GUI	G raphical U ser I nterface
IBDP	I ntegration with B uilding D esign P rocess
IBM	I nteroperability of B uilding M odelling
IICB	I ntegration of I ntelligent design K nowledge- B ase
IT	I nformation T echnology
lmst	local m ean s idereal t ime
LW	L ong W ave R adiation
NUMA	N on- U niform M emory A ccess
ODE	O rdinary D erivatives E quation
PDE	P artial D erivatives E quation
RAN	R andom A ccess M emory
RF	R esponse F actors

SER	S eeding, E volutionary growth and R eseeding
SH	S hort W ave Radiation
SMP	S ymmetric M ulti- P rocessor
SS	S tate- S pace Method
TT	T errestrial T ime
UIM	U sability and I nformation M anagement of interface
UMA	U niform M emory A ccess
UT	U niversal T ime
WMO	World Meteorological Organization

11 Autori

Prof. Livio Mazzarella

Laureato in Ingegneria Meccanica indirizzo Energetico al Politecnico di Milano nel 1982 con il massimo dei voti, consegue il titolo di Dottore di Ricerca in Energetica, primo ciclo, nel luglio 1987. Ricercatore di ruolo dal 1990, professore incaricato dal 1993 al 1997 presso l'Università di Pavia, diventa professore associato di ruolo in Fisica Tecnica Ambientale nel 1998 e successivamente Professore Straordinario sempre presso la Facoltà d'Ingegneria del Politecnico di Milano dal 1° settembre 2000. Dal 1° settembre 2003, confermato nel ruolo di Professore Ordinario di Fisica Tecnica Ambientale (ING-IND/11), è stato docente di numerosi insegnamenti nell'ambito della fisica dell'edificio. Ha svolto e svolge attualmente attività di ricerca, documentata da più di 200 pubblicazioni, sui temi dell'Energetica e della Fisica Tecnica, e più precisamente nei settori della termo fluidodinamica, modellistica numerica dei sistemi edificio-impianto, energetica negli edifici, termofisica degli edifici, conversione termica dell'energia solare, sistemi per l'accumulo di energia termica: interstagionali e non, pompe di calore, trasporto di massa ed energia nei mezzi porosi, acustica applicata. Partecipa da più di vent'anni all'attività di normativa nel settore dell'energia negli edifici e degli impianti tecnici tramite il Comitato Termotecnico Italiano, ove ha ricoperto la carica di vice-presidente del SC1 del CTI e ricopre attualmente la carica di presidente del SC5. È inoltre REHVA fellow, coordinatore del COP group di REHVA, membro della giunta direttiva di AICARR e di IBPSA-Italy.

Ing. Andrea Alongi

Laureato in Ingegneria Edile con lode nell'aprile 2007 presso il Politecnico di Milano, ha conseguito il Dottorato di Ricerca in Energetica con merito presso lo stesso Ateneo nel 2012 con uno studio sui componenti d'involucro edilizio permeabili all'aria. Dal 2007 al 2019 è stato assegnista di ricerca presso il Dipartimento di Energia, dove ha svolto attività di ricerca sull'analisi numerica della prestazione energetica di edifici: lo studio, sia in termini di fabbisogno energetico in regime dinamico sia valutandone il comportamento in condizioni standardizzate da procedura per certificazione energetica, di edifici reali esistenti o in fase di progettazione; l'analisi parametrica tramite metodologia cost optimality di varie categorie di edifici virtuali (analisi per archetipi di residenze nuove ed esistenti, uffici nuovi ed esistenti, scuole). Dall'ottobre 2010 è stato assunto dallo stesso Ateneo nel ruolo tecnico.

Prof. Adriana Angelotti

Adriana Angelotti laureata in Fisica nel 2000, ha conseguito il Dottorato in Ingegneria Energetica presso il Politecnico di Milano nel 2004, dove nel 2005 ha preso servizio come ricercatrice di ruolo nel settore scientifico Fisica tecnica ambientale e nel 2019 quale professore associato nello stesso SSD. Si dall'inizio ha svolto attività di ricerca nell'ambito dell'efficienza energetica del sistema edificio-impianto: prestazioni degli impianti di climatizzazione (geotermici, sistemi a bassa exergia) e delle tecnologie di involucro edilizio innovative (facciate traspiranti, facciate ventilate, inerzia termica, sistemi tessili per l'architettura); benessere termoigrometrico in ambienti interni ed esterni; ventilazione naturale alla scala urbana, con particolare riferimento alla loro modellazione ai fini della simulazione dinamica.

Ing. Giulia Filippini

Laureata in Building Engineering nell'Aprile 2018, Giulia Filippini è assegnista di ricerca presso il Dipartimento di Energia del Politecnico di Milano. Ha cominciato la sua collaborazione con il gruppo di ricerca RELAB grazie alla sua tesi sul progetto Europeo "Heat4Cool", riguardante la ristrutturazione di impianti di condizionamento. Continua a lavorare su progetti che richiedono la conoscenza di modellazione energetica degli edifici, in particolare l'uso di software per simulazioni dinamiche. Nel Gennaio 2020 ha ottenuto l'abilitazione all'esercizio della professione di Ingegnere.

Prof. Alessandro Gandelli

Alessandro Gandelli, ha conseguito la Laurea e il Dottorato in Ingegneria Elettrica presso il Politecnico di Milano. Dal 1990 fa parte della Facoltà di Ingegneria dello stesso ateneo, dove è Professore Associato di Ingegneria Elettrica. Dal 1998 al 2001 è stato membro del Senato Accademico. Le sue attività di ricerca e interessi scientifici riguardano la teoria di base dei sistemi elettrici e del mercato elettrico. Ha pubblicato più di 100 articoli e tre libri di testo nel campo dell'ingegneria elettrica ed elettronica. È membro di IEEE, ACM, SPIE, IMAPS e della New York Academy of Science.

Ing. Martina Pasini

Laureata con lode nel 2006 con tesi "Analisi tecnica comparativa di nuove tecnologie emergenti", riguardante principalmente la tecnologia di facciata ventilata trasparente (Double Skin Façades) e diplomatasi nello stesso anno all'Alta Scuola Politecnica, ha conseguito il titolo di dottore di ricerca, con lode, nel 2010 con la tesi "Towards an enriched modularity of building performance simulation's programs. Reconceptualisation and development of an Object-Oriented model for the Simulation of the Building System". Assegnista di ricerca per il Politecnico di Milano dal 2010 al 2017. Collaboratrice, come analista informatica e programmatrice, per un'industria produttrice di scambiatori di calore dal 2016 al 2019. Si occupa di simulazione dinamica dei fabbisogni del sistema Edificio-Impianto, sostenibilità ambientale, ottimizzazione di componente e di processo, algoritmi di automazione di processo, analisi dati.

12 Allegato A: Confronto tra le prestazioni di C#, C++ e FORTRAN

12.1 Sviluppo del codice: pro e contro di C# vs C++

C++ è in genere più veloce di C#, ma in C#:

- non è necessario gestire la memoria in .net,
- è più facile da imparare,
- è più leggibile,
- consente una scrittura più veloce del codice.

12.2 Tempo di esecuzione degli algoritmi sequenziali: influenza della natura dell'algoritmo

12.2.1 Complessità dell'algoritmo e/o dipendenza dal numero di linee del codice

Confrontando i tempi di esecuzione di codici scritti in C# e C++, è evidente che i due fattori principali che ne influenzano le prestazioni in esecuzione sono: il numero di linee di codice da eseguire e la complessità degli algoritmi.

La tabella seguente mostra il tempo di esecuzione di un codice che calcola la somma di una serie, scritto separatamente per ciascun linguaggio di programmazione (Saxena, 2009).

Tabella A- 1 – Confronto prestazioni C#, C++ rispetto alla lunghezza del codice

Processore	Intel Core 2 Duo							
Linguaggio di programmazione	C#				C++			
Linee di codice	10 ¹	10 ²	10 ³	10 ⁴	10 ¹	10 ²	10 ³	10 ⁴
Tempo medio di esecuzione (in μ s)	0.32	0.60	3.60	34.18	0.32	0.61	3.48	29.38

Come mostrato nella tabella, aumentando le righe di codice, la differenza tra i tempi di esecuzione tra il codice in C++ e in C# è più evidente, anche se resta limitata (al massimo C# risulta il 16% più lento).

Per indagare l'impatto della complessità, si sono analizzati i prodotti matriciali e la risoluzione di sistemi lineari con decomposizione LU di matrici sparse.

12.2.1.1 Prodotti matriciali

L'immagine seguente mostra il tempo di esecuzione di prodotti matriciali in C# e C++.

Nel benchmark di test, sono state moltiplicate due matrici in doppia precisione (double) di 1000 x 1000 elementi. L'immagine seguente mostra la versione C++ originale e il suo diretto equivalente in C#, oltre alla versione doppia [n][n] che è il metodo più appropriato in C#⁴.

Sebbene questi risultati non siano molto precisi e possono variare da un processore ad un altro, si può dedurre che:

1. C++ vince in ogni caso;
2. il framework .Net4 ha prestazioni leggermente migliori rispetto al .Net3;
3. mono (l'implementazione Linux del framework .Net) ha le prestazioni peggiori.

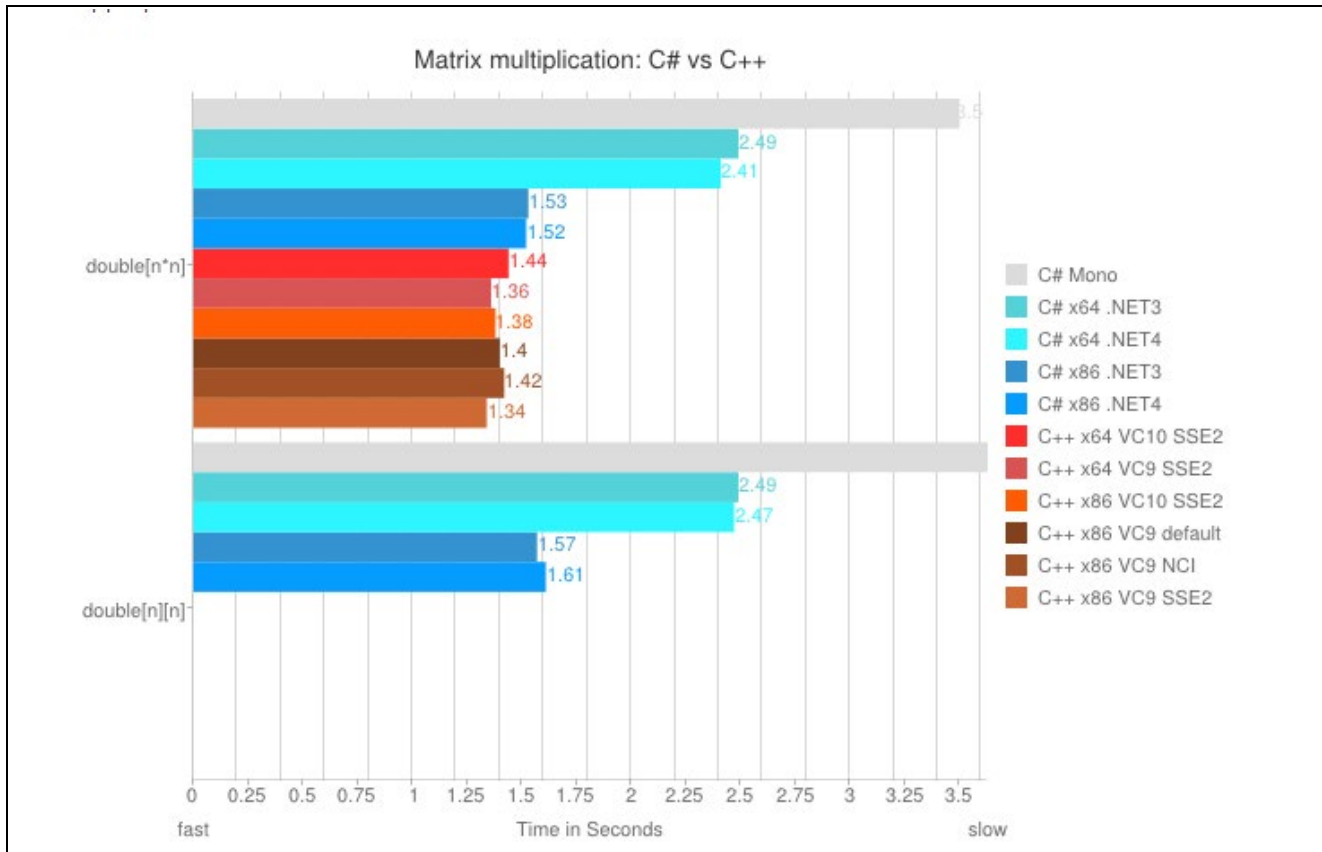


Figura A- 1 Prodotto matriciale in C++ e C#

Il doppio [] [], è il modo più ragionevole per eseguire la moltiplicazione di matrici di grandi dimensioni in C#, poiché uso dell'aritmetica dei puntatori è "disapprovato" in C#.

Come si vedeva, per algoritmi complessi come la moltiplicazione di matrici, C++ funziona meglio di C#, ma alcune precauzioni possono ridurre la distanza tra i due linguaggi di programmazione (vedi Figura A- 1)

12.2.1.2 Impiego delle librerie C++

Da questi risultati si evince che, se si vuole puntare alla massima prestazione possibile, ma sfruttare al contempo i vantaggi offerti dal C#, si può ricorrere ad una programmazione mista: utilizzare delle librerie scritte in C++ per i calcoli più complessi.

I codici di calcolo devono essere scritti in C++ e non in Visual C++ ed essere compilati in una libreria "linkabile" dinamicamente (DLL), che può essere chiamata utilizzando l'interoperabilità tra C# e DLL. In questo caso le prestazioni dovrebbero essere almeno le stesse della libreria C++.

⁴ <http://www.codeproject.com/Articles/212856/Head-to-head-benchmark-Csharp-vs-NET>

Per utilizzare la libreria C++ in C#, occorre ovviamente utilizzare costrutti di interfaccia compatibili.

12.3 Sovraccosto per la parallelizzazione

Essendo oggi i processori dei computer multi-CPU, quando lo sforzo computazionale diventa notevole, come avviene nei codici di simulazione dinamica, diventa categorico sfruttare tutte le risorse disponibili, cioè tutte le CPU, fisiche e/o virtuali (multi-threads) per minimizzare i tempi di calcolo e quindi ricorrere al calcolo parallelo.

Al fine di parallelizzare il calcolo si possono impiegare specifiche librerie, quali Task Parallel Library (TPL) in C# o Parallel Patterns Library (PPL) in C++, che, però, quando le iterazioni sono piccole e le dimensioni del lavoro sono variabili, comportano un sovraccosto dovuto alla gestione del parallelismo, che potrebbe vanificare i vantaggi.

Il ciclo FOR parallelo, come predefinito dei linguaggi che lo gestiscono, inizia suddividendo il lavoro tra i core fisici e/o virtuali disponibili, lavoro che poi viene bilanciato dinamicamente tra questi in base al carico. Nella maggior parte dei casi, con un gran numero di iterazioni e quando il lavoro sottostante tra ogni iterazione non ha condizioni di esecuzione significativamente dissimili, funziona ottimamente, ma in alcuni casi potrebbe introdurre un sovraccarico inaccettabile, per cui la convenienza della parallelizzazione va indagata caso per caso.

Bisogna inoltre considerare che C# e C++ implementano per il calcolo parallelo algoritmi leggermente diversi nel modo in cui gestiscono le iterazioni⁵.

12.3.1 Casi test di sovraccarico di parallelizzazione: fattorizzazione LU di matrici sparse in C# e C++

In un recente lavoro (Angelo Kaldéren, 2013), sono stati confrontati modelli paralleli in C / C++ e C# / Java. Il codice di benchmark C++ è stato parallelizzato dalla libreria Threading Building Blocks (TBB) mentre in C# è stato utilizzato TPL. TBB ha il vantaggio di sottrarre attività alle varie CPU per bilanciare correttamente un carico di lavoro parallelo attraverso la piattaforma multi-core. Significa che, se un processore termina tutte le attività presenti nella sua coda di esecuzione, C++ TBB, che prende in considerazione tutte le attività che devono svolgere i vari core, dirotta su questo core scarico quelle attività che altri core hanno nella loro coda in attesa di essere eseguite. In tal modo si ottiene un carico di lavoro più equilibrato su tutti i processori e si crea parallelismo più efficiente. TPL semplifica la programmazione parallela in C# semplificando il processo di aggiunta di parallelismo e concorrenza alle applicazioni. Ridimensiona automaticamente il grado di concorrenza per utilizzare in modo più efficiente tutti i processori disponibili.

Per il test di C#, poiché non questo non consentiva il controllo diretto del numero di thread in esecuzione nel pool di thread, il numero di attività eseguite dal pool di thread è risultato limitato. Ciò comporta una perdita di prestazioni aumentando il sovraccarico.

Le immagini riportate in Figura A- 2 mostrano la velocità e il tempo di esecuzione di fattorizzazione LU di matrici sparse (SparseLU). Dall'esame dei grafici riportati, si può dedurre che la parallelizzazione in C# e C++ è simile fino a un certo numero di thread. Ma in C# dopo 36 thread non c'è più crescita della velocità di esecuzione che sembra essere correlata alla limitazione del numero di attività in esecuzione parallelo.

In (Miskovski, 2007) viene riportata l'accelerazione del calcolo π greco, PI, (Figura A- 3Figura A- 3), come algoritmo lineare di ordine $O(n)$ e della moltiplicazione tra matrici (Figura A- 4), come un algoritmo complesso di ordine $O(n^3)$, in funzione del numero di threads contemporaneamente in esecuzione parallela. L'algoritmo PI è un calcolo intensivo senza dati di input, mentre la moltiplicazione matriciale (1000x1000) è intensiva sia rispetto al calcolo che ai dati.

⁵ <http://social.msdn.microsoft.com/Forums/vstudio/en-US/954a3bd9-fc7a-41ff-afd1-72336c55b112/c-tpl-faster-than-c-ppl?forum=parallelcpnative>

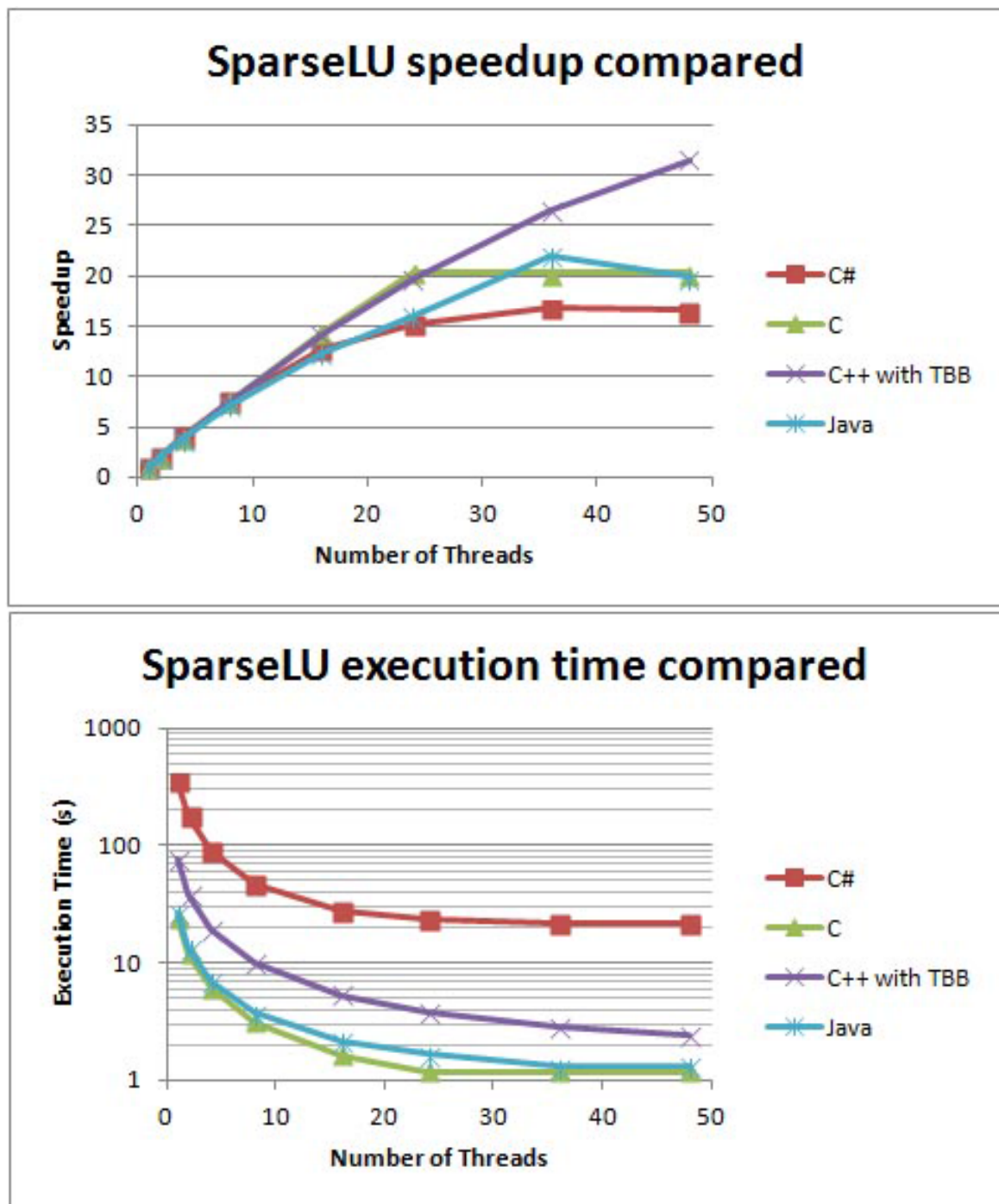


Figura A- 2 Fattorizzazione LU di matrici sparse in C++ e C#

Come si può vedere in Figura A- 3, quando le applicazioni sono in esecuzione con più di un thread, l'accelerazione ottenuta per il calcolo PI si stabilizza nell'intorno di 2 sia per C++ che per C#. Per la moltiplicazione matriciale le prestazioni un p' diverse (Figura A- 4). In questo caso, la moltiplicazione matriciale risulta in C# più lenta che in C++, ma per entrambi dopo due thread non c'è più incremento di velocità.

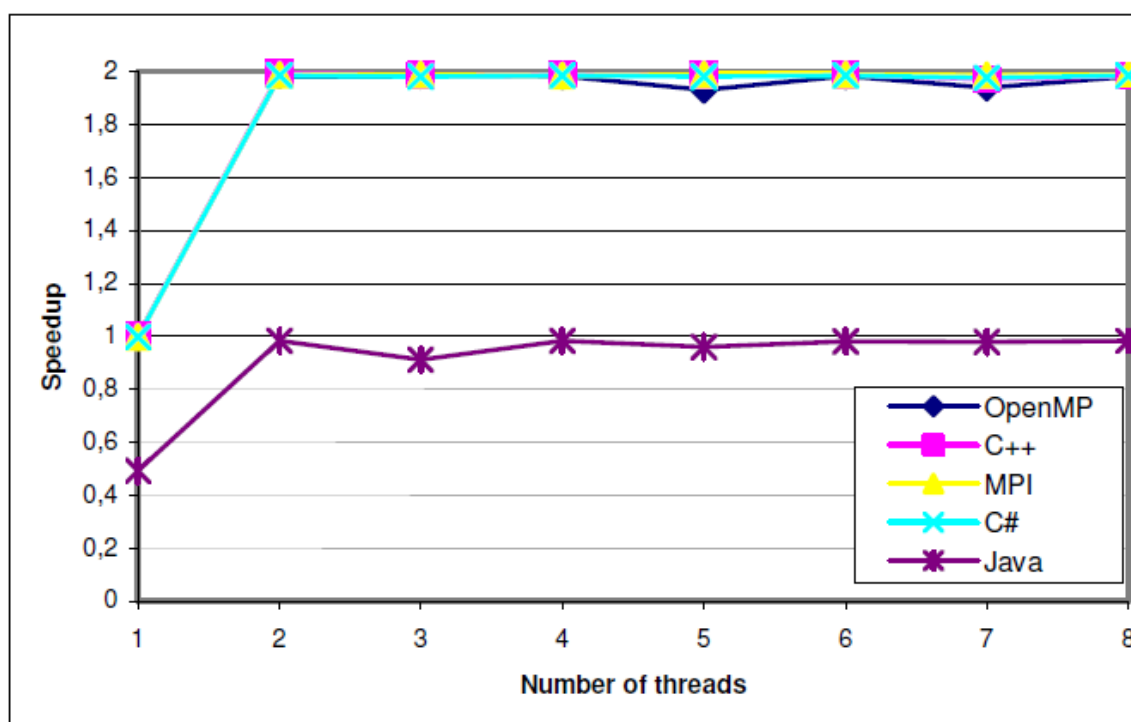


Figura A- 3 Accelerazione del calcolo PI rispetto all'applicazione sequenziale in base al numero di thread per Intel Pentium D Dual Core

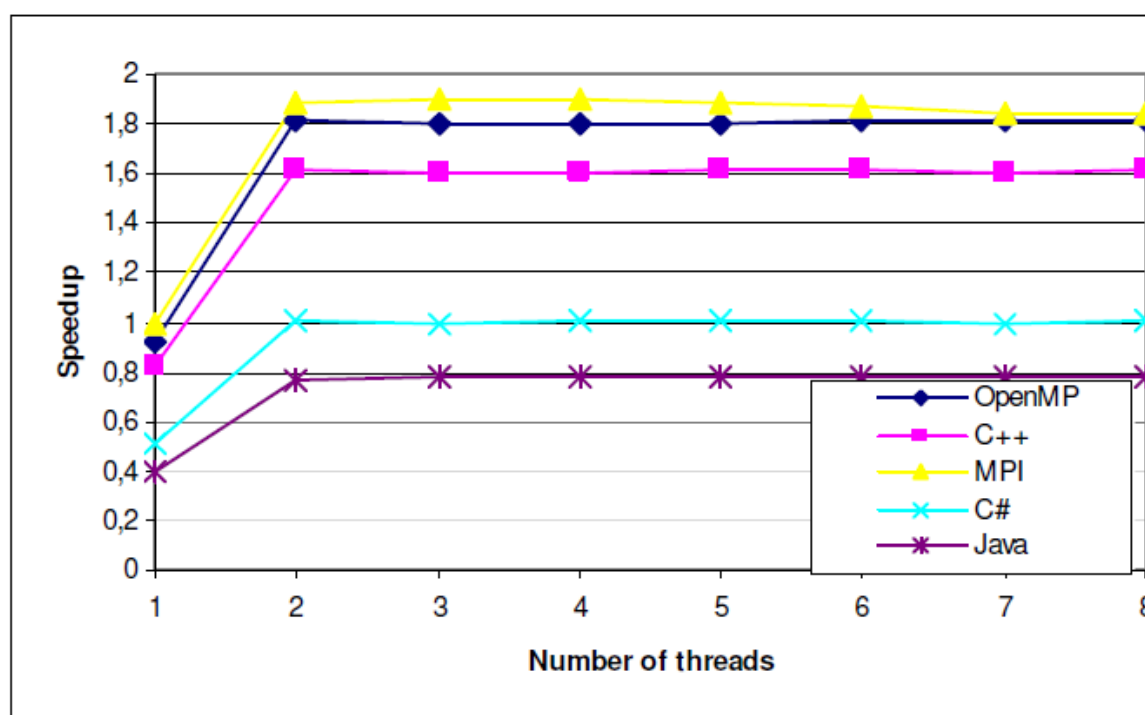


Figura A- 4 Accelerazione della moltiplicazione della matrice relativa all'applicazione sequenziale in base al numero di thread per Intel Pentium D Dual Core

12.4 Parallelizzazione con memoria condivisa o distribuita

Nell'implementazione di sistemi di calcolo multi-core, vi sono due diverse architetture:

- sistemi con memoria condivisa,
- sistemi con memoria distribuita.

Nel primo caso, ogni core⁶ accede ad una parte di una unica memoria RAM che viene condivisa tra le varie unità. Questo di solito è il caso di workstation con multiprocessore simmetrico (SMP), cioè una CPU con più core o più CPU che condividono in modo paritario un'unica memoria (vedasi Figura A- 5). Ciò implica un indirizzamento globale dello spazio di memoria detto accesso uniforme della memoria (UMA),

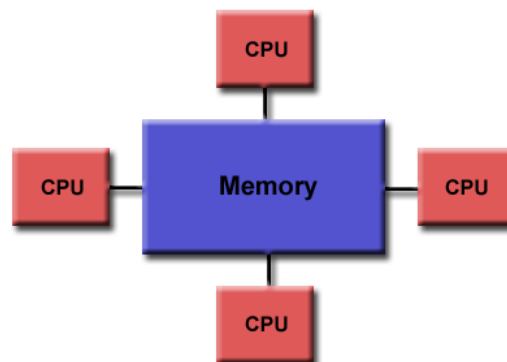


Figura A- 5 Architettura multiprocessore (SMP) con accesso uniforme alla memoria (UMA)

Esiste anche un'altra architettura di memoria condivisa, detta ad accesso non uniforme (NUMA), dove più multiprocessori simmetrici (SMPs), che possiedono ognuno una propria memoria (come nella configurazione SMP-UMA) possono condividere questa memoria "personale" con gli altri SMP, in modo subordinato. In questo caso si ha ancora un indirizzamento globale della memoria attraverso tutti i processori.

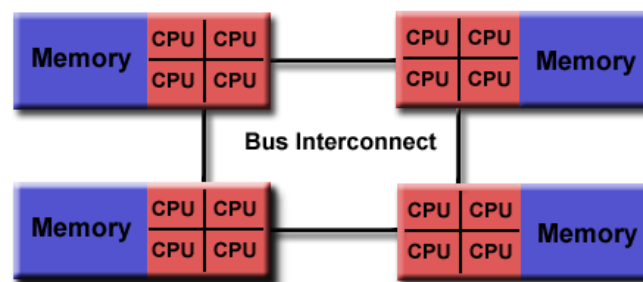


Figura A- 6 Architettura con più multiprocessori (SMPs) con accesso non uniforme alla memoria (NUMA)

I vantaggi della memoria condivisa sono:

- lo spazio di indirizzi globale offre una prospettiva di programmazione intuitiva per la memoria;
- la condivisione dei dati tra le attività è rapida e uniforme a causa della vicinanza della memoria alle CPU.

Nel caso di memoria distribuita, ogni core ha una sua memoria RAM dedicata che può essere utilizzata solo da tale CPU; non si ha in questo caso una condivisione degli spazi di memoria indirizzabili. Questo è di solito il caso di cluster di computer single core. Quando un processore ha bisogno di accedere ai dati in un altro processore, di solito è compito del programmatore definire esplicitamente come e quando i dati vengono

⁶ l'unità di elaborazione che riceve istruzioni ed esegue calcoli o azioni sulla base di tali istruzioni

comunicati. Anche la sincronizzazione tra compiti è responsabilità del programmatore. Tutto ciò presuppone l'esistenza di una rete di comunicazione per collegare tra loro i processori e quindi i dati contenuti nelle memorie proprietarie.

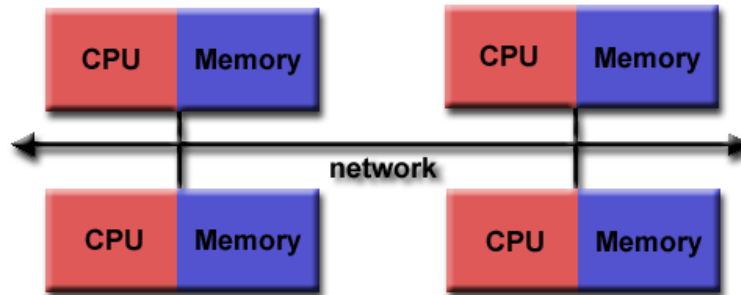


Figura A- 7 Architettura con più processori con memorie proprietarie interagenti tramite network di comunicazione

I vantaggi della memoria distribuita sono:

- la memoria è scalabile con il numero di processori: aumenta il numero di processori e la dimensione della memoria aumenta proporzionalmente;
- ogni processore può accedere rapidamente alla propria memoria senza interferenze e senza il sovraccarico dovuto al tentativo di mantenere la coerenza della cache globale;
- convenienza: può utilizzare prodotti di base, processori standard e networking.

12.4.1 Implementazione in C++

L'implementazione del parallelismo avviene tramite l'impiego di opportune librerie che sono diverse per l'utilizzo di memoria condivisa o distribuita.

- **memoria distribuita:**
 - MPI (Message Passing Interface):
è uno protocollo standardizzato e portatile per la trasmissione di messaggi progettato da un gruppo di ricercatori del mondo accademico e industriale per funzionare su un'ampia varietà di architetture di calcolo parallelo;
- **memoria condivisa:**
 - OpenMP
è un'API (Application Program Interface), definita congiuntamente da un gruppo di importanti fornitori di hardware e software per computer. OpenMP offre un modello portatile e scalabile per gli sviluppatori di applicazioni parallele con memoria condivisa. L'API supporta C/C ++ e Fortran su un'ampia varietà di architetture.
 - TBB (Threading Building Blocks)
è una libreria di modelli C++ sviluppata da Intel per la programmazione parallela su processori multi-core. Utilizzando TBB, un calcolo viene suddiviso in attività che possono essere eseguite in parallelo. La libreria gestisce e pianifica i thread per eseguire queste attività.

12.4.2 Implementazione in C#

12.4.2.1 Memoria distribuita

Il risultato di un'analisi delle prestazioni di un algoritmo parallelo scalabile di clustering di dati per sistemi multicore, che confronta C++ con MPI e una libreria C# di runtime di messaggistica, CCR, sono molto incoraggianti riguardo all'utilizzo di C# per applicazioni importanti.

Per C# l'equivalente della libreria MPI di C++ è MPI.net, che si basa sulla libreria MPI esistente come **MS-MPI**. Il supporto per la programmazione ibrida con MPI.net è limitato dalla libreria MPI sottostante, che non supporta il livello di threading MPI_THREAD_MULTIPLE.

McMPI, un'altra libreria MPI per C#, non è pronta per l'uso commerciale e presenta alcune limitazioni, come l'assenza di un meccanismo di tipo di dati, che attualmente supporta solo array di byte per i buffer dei dati dei messaggi.

CCR (Concurrency and Coordination Runtime) è una libreria di codici gestiti, una dll, accessibile da qualsiasi target .Net Common Language Runtime (**CLR**). CCR risponde all'esigenza di un'applicazione orientata ai servizi per gestire operazioni asincrone, gestire la concorrenza, sfruttare l'hardware parallelo e gestire i guasti parziali. Consente all'utente di progettare applicazioni in modo che i moduli software o i componenti possano essere accoppiati liberamente. Significa che possono essere sviluppati indipendentemente e fare ipotesi minime sul loro ambiente di runtime e altri componenti.

La programmazione ibrida (**MPI + X**) di solito è solo un tentativo per applicazioni su larga scala, oltre 10.000 core.

12.4.2.2 Memoria condivisa

La libreria **TPL** (Task Parallel Library) ha semplificato il processo di aggiunta di parallelismo e concorrenza alle applicazioni. Il TPL ridimensiona il grado di concorrenza in modo dinamico per l'uso più efficiente di tutti i processori disponibili. Inoltre, gestisce il partizionamento del lavoro, la pianificazione dei thread nel threadpool e alcuni altri dettagli di basso livello. Usando TPL, le prestazioni del codice possono essere massimizzate mentre ci si concentra sul lavoro che il programma è progettato per realizzare.

12.4.2.3 Progetti C# scalabili

Un codice di calcolo scalabile è un codice che a seconda della bisogna può essere eseguito su un numero variabile a scelta di CPU in parallelo.

Per scalare un codice C# nella memoria distribuita, è meglio prendere in considerazione le librerie di .Net Remoting o WCF (Windows Communication Foundation) che sono più lente di MPI ma sono più facili / veloci per far funzionare il codice come desiderato.

12.5 Librerie numeriche per il framework .NET

In questa sezione vengono esaminate le diverse librerie matematiche disponibili scritte in c/c++ e c#.

Le librerie matematiche per .Net possono essere implementate in diversi modi. Esistono diversi modi per implementare algoritmi numerici per il framework .Net⁷, quali:

1. chiamata a un codice nativo ottimizzato per il processore tramite P/Invoke,
2. chiamata a un codice nativo non ottimizzato tramite P/Invoke,
3. chiamata a un codice nativo ottimizzato da linguaggi dinamici,
4. impiego di codice gestito tramite C++/CLI ().
5. utilizzo di codice C# non sicuro;
6. impiego di codice C# gestito;

⁷ [http://msdn.microsoft.com/en-us/library/vstudio/304368\(v=vs.100\).aspx#gh](http://msdn.microsoft.com/en-us/library/vstudio/304368(v=vs.100).aspx#gh)

7. impiego di codice F # gestito,
8. impiego di codice IronPython gestito.

Tabella A- 2 – Confronto prestazioni delle varie librerie su uno stesso caso di prova

Metodo	Tempo medio (s)
P/Invoke optimized	0.41
P/Invoke native	3.96
IronPython/NumPy	5.11
Managed C++/CLI	5.50
Unsafe C#	6.28
Managed C#	13.16
Managed F#	13.08
Managed IronPython	non terminata!

12.5.1 Codice nativo

Si definisce codice “nativo” un codice eseguibile compilato, come applicazione o libreria dinamica (dll), per essere direttamente eseguito in uno specifico sistema operativo e uno specifico tipo di processore (SO/HW). È quindi un codice “macchina” non portabile, cioè non utilizzabile su altro sistema operativo e/o hardware.

12.5.1.1 Librerie sequenziali gratuite

- Eigen (sviluppato per C/C++)

12.5.1.2 Librerie parallele gratuite

- Trilinos (sviluppato per C/C++)
- Microsoft Sho (sviluppato per .Net)
- Microsoft Solver Foundation
- Math.NET Numerics (sviluppato per .Net)

12.5.1.3 Librerie sequenziali commerciali

- Intel MKL (sviluppato per C, C++ e Fortran)

12.5.1.4 Librerie parallele commerciali

- NMath (sviluppato per .Net)

Nmath è una libreria numerica che fornisce componenti orientati agli oggetti per applicazioni matematiche, ingegneristiche, scientifiche e finanziarie nella piattaforma .Net che contiene classi vettoriali, classi di matrici, classi di numeri complessi, generatori di numeri casuali e funzioni ad alte prestazioni per valori numerici orientati agli oggetti. Nelle seguenti tabelle sono riportati i risultati ottenuti nella moltiplicazione matrice/matrice per thread singolo e multithread per a C++ e C#.

Tabella A- 3 – Confronto prestazioni single-thread di NMath per C++ e C#

Dimensione	Ripetizioni	C++(sec)	NMath(sec)	C#(sec)
2	500,000	0.0748	0.0671	0.0272
5	100,000	0.0250	0.0515	0.0538
10	50,000	0.0437	0.0468	0.1806
100	1000	0.1919	0.2044	3.4780
1000	1	0.1810	0.1825	7.9482
Single-thread				

Tabella A- 4 – Confronto prestazioni multi-thread di NMath per C++ e C#

Dimensione	Ripetizioni	C++(sec)	NMath(sec)	C#(sec)
2	500,000	0.0780	0.0671	1.9484
5	100,000	0.0280	0.0515	0.4976
10	50,000	0.0422	0.0515	0.5850
100	1000	0.0702	0.0858	1.1638
1000	1	0.0577	0.0671	2.0436
Mutithread				

12.5.2 Codice gestito

Si definisce codice “gestito” un codice precompilato che viene successivamente elaborato dal compilatore Just In Time in codice nativo in fase di runtime. Il codice gestito ha l'interessante effetto collaterale di avere il potenziale di essere eseguibile su diversi sistemi operativi, poiché il codice macchina non viene creato fino a quando la VM (in questo caso il framework .NET) non lo utilizza effettivamente. In questo modo, è possibile eseguire lo stesso applicativo .NET su Windows e su Linux o Mac, su cui è installato il runtime Mono (i framework .NET per tali sistemi operativi).

12.5.2.1 Librerie parallele commerciali

- IMSL parallela

IMSI è una libreria numerica per C# che utilizza Task Parallel Library (TPL) per abilitare il parallelismo nei sistemi di memoria condivisa, in particolare i sistemi multi-core. La tabella seguente mostra il risultato di alcuni benchmark:

Tabella A- 5 –Prestazioni di codice C# al variare dei numero di core utilizzati

Nome	Core	Tempo (s)
Matrix Multiplication	1	28.851
Matrix Multiplication	2	15.482
Matrix Multiplication	4	7.9
Matrix Multiplication	8	4.21
LU	1	10.876
LU	2	6.453
LU	4	6.82
LU	8	5.138

12.6 Test finali e scelta

Per pervenire ad una decisione sul linguaggio preferibile per lo sviluppo di un nuovo codice BPS, che fosse di facile utilizzo, ad oggetti, ma che non sacrificasse troppo le prestazioni computazionali e fosse in grado di usare in modo efficiente il calcolo parallelo, si sono eseguiti degli ulteriori test, rispetto a quanto già evidenziato e riportato in letteratura, impiegando tre diversi linguaggi di programmazione: C++, C# e il FORTRAN.

Per evidenziare le eventuali differenze rispetto all'efficienza nel calcolo numerico utilizzando le librerie base di linguaggi, si è implementato un ciclo di calcolo, ripetuto $2E+8$ volte, della seguente relazione:

$$\sum_{i=1}^{2 \cdot 10^8} \frac{\text{Exp}(\log i)}{i}$$

che coinvolge due espansioni in serie incastate una dentro l'altra (il calcolo del logaritmo e il calcolo dell'esponenziale) più una divisione, che è l'operazione aritmetica che richiede il maggior tempo di calcolo. In Tabella A- 6, sono riportati i risultati a seguito della compilazione ed esecuzione di tale test su un computer equipaggiato con processore multi-core Intel Core i7 Extreme 2960XM a 2,7 GHz, con 4 core e 8 threads con Windows 7 a 64 bit. I risultati migliori si ottengono in calcolo parallelo con il FORTRAN (0,98 s), che però non è un linguaggio di programmazione ad oggetti, e solo se nella compilazione si usa l'ottimizzazione specifica a 64 bit per il processore Intel. Segue il C++ (1.47) sempre in calcolo parallelo e con l'ottimizzazione specifica a 64 per il processore Intel, con una riduzione al 23% del tempo richiesto per il calcolo seriale. È interessante notare cosa succede con C#. Nel calcolo parallelo si posiziona al terzo posto, con un miglior tempo di 1.73 s e una riduzione al 22% del tempo richiesto per il calcolo seriale. Ma tale risultato si ottiene con una precompilazione non dedicata (any CPU), che consente poi al compilatore Just on Time di ottimizzare rispetto allo specifico OS e HW su cui si manda in esecuzione il codice. Infatti se si forza l'ottimizzazione per un sistema operativo a 32 bit i risultati peggiorano notevolmente, ma risultano ancora peggiori anche quando si usa l'ottimizzazione a 64 bit.

Tabella A- 6 –Prestazioni dello stesso algoritmo di calcolo con C++, C# e FORTRAN, con 1 o 4 core.

C#	N. Iterazioni	Tempo [s]
Sequenziale precompilato per any CPU	2.00E+08	7.75
Sequenziale precompilato per x86	2.00E+08	11.24
Sequenziale precompilato per x64	2.00E+08	6.44
Parallelo precompilato per any CPU	2.00E+08	1.73
Parallelo precompilato per x86	2.00E+08	2.53
Parallelo precompilato per x64	2.00E+08	2.11
C++	N. Iterazioni	Tempo [s]
Sequenziale compilato ottimizzato HW	2.00E+08	6.43
Parallelo compilato ottimizzato HW	2.00E+08	1.47
Intel Fortran	N. Iterazioni	Tempo [s]
Sequenziale compilato ottimizzato HW	2.00E+08	5.37
Parallelo compilato ottimizzato HW	2.00E+08	0.98

In sintesi, con la precompilazione generica, quindi senza vincolare l'eseguibile generato ad alcun sistema operativo e hardware, si ottiene con C# una prestazione in calcolo parallelo che è solo del 18% più lenta di quella ottenibile con C++, però ottimizzato per lo specifico SO e HW. Inoltre, se il codice in C++ non è ottimizzato questo risulta più lento del C#. Da cui, la scelta di adottare C# per lo sviluppo del nuovo codice.

12.7 Bibliografia

Angelo Kaldéren, G. (2013). A comparative analysis between parallel models in C/C++ and C#/Java: A quantitative comparison between different programming models on how they implement parallelism.

Mishkovski, Igor & Trajanov, Dimitar & Grnarov, Aksenti. (2007). PERFORMANCE COMPARISON OF MULTICORE PROCESSORS USING VARIOUS SOFTWARE PLATFORMS. Conference proceedings ETAI 2007. International Conference on Emerging Intelligent Technologies and Applications,

Saxena, V. &. (2009). Performance evaluation for object oriented software systems. ACM SIGSOFT Software Engineering Notes, (pp. 1-5).