# SVILUPPO DEI CODICI DI SIMULAZIONE PER IL TRASPORTO DEI FRAMMENTI DI FISSIONE E PER IL COMPUTO DEGLI EFFETTI RADIATIVI

## SINTESI

In questo rapporto si sviluppano i codici ab initio che permettono di risolvere le problematiche del rilascio energetico dei frammenti di fissione e del bilancio della radiazione nel gas. Questo lavoro può essere portato a compimento solo se si accoppiano i codici coi risultati derivanti dall'analisi fluidodinamica.

In particolare è stato necessario affiancare ad un codice di fluidodinamica nativo del CRS4 (KARALIS), due nuovi codici che permettono di studiare in geometrie tridimensionali i meccanismi di cessione di calore al gas da parte dei frammenti di fissione (FF_HEAT) e la ridistribuzione dell'energia così assorbita a causa dei processi radiativi innescati dalle alte temperature raggiunte in camera di combustione (PHOTON).

Il rapporto illustra i dettagli sul funzionamento di codici e le tecniche di sviluppo di FF_HEAT e PHOTON e gli algoritmi Montecarlo sui quali sono basati. Inoltre viene descritta l'architettura ed il funzionamento del codice completo che simula la fisica completa del gas nella camera di combustione e le interazioni tra i singoli codici.

# INDICE

INTRODUZIONE ..... 3

STRUTTURA DEL CODICE ..... 5

*STRUTTURA DEL CODICE MONTECARLO PER IL CALCOLO DELLA DISTRIBUZIONE DEI FRAMMENTI DI FISSIONE* ..... *14*

*RISULTATI DEL CODICE FF_HEAT* ..... *16*

CALCOLO DELLA DISTRIBUZIONE DI ENERGIA NEL GAS PROPELLENTE AD ALTA TEMPERATURA: IL CODICE PHOTON ..... 21

*LA TECNICA MONTE CARLO IMPLEMENTATA* ..... *21*

*STRUTTURA DEL CODICE MONTECARLO PER IL CALCOLO DELLA DISTRIBUZIONE DI ENERGIA MEDIANTE FOTONI* ..... *24*

*INPUT e OUTPUT DI UNA SIMULAZIONE* ..... *26*

ALGORITMO PATHFINDER: ATTRAVERSAMENTO DELLA GRIGLIA REGOLARE (LATTICE) ... 28

RIFERIMENTI ..... 31

## INTRODUZIONE

In questo rapporto verranno descritti i risultati dello sviluppo delle proposte presentate nel rapporto 5 del progetto ASI 242 dal titolo "Sviluppo dei Codici di Simulazione per il Trasporto dei Frammenti di Fissione e per il Computo degli Effetti Radiativi".

Vengono quindi presentati gli algoritmi ed i codici sviluppati appositamente nell'ambito del progetto 242 che simulano il trasporto dei frammenti di fissione ed il relativo riscaldamento del gas all'interno della camera di combustione del motore spaziale ampiamente descritto nel rapporto [1].

Per poter simulare il comportamento del gas propellente (idrogeno in questa prima fase di studio) è sorta la necessità di dover accoppiare i risultati delle simulazioni della fluidodinamica all'interno della camera di combustione, ottenuti attraverso il codice KARALIS, con un nuovo codice implementato ad hoc che simula il trasporto ed il rilascio di calore dei frammenti di fissione generati dallo strato di materiale fissile distribuito sulle pareti della camera stessa. Questo nuovo codice denominato FF_HEAT è stato sviluppato appositamente per risolvere il problema del riscaldamento del gas in geometrie tridimensionali complesse in modo tale da poter confermare i risultati ottenuti nel modello uni dimensionale presentato nel rapporto [1]

Come descritto di seguito, per poter interfacciare i codici si è reso necessario sviluppare uno strumento computazionale in grado di trasformare e convertire le strutture dati caratteristiche di KARALIS e FF_HEAT che risultano essere incompatibili data la natura intrinsecamente diversa delle metodologie di base dei due codici. Il software sviluppato è stato denominato MAPPING, ampiamente descritto nell'allegato [2], in quanto permette di costruire le matrici di mappatura tra gli elementi costitutivi delle griglie computazionali su cui sono basati i due codici. Una volta definite ed immagazzinata le matrici così costruite il trasferimento di considerevoli quantità di dati (centinaia di milioni) da un codice all'altro risulta più agevole.

Da considerazioni qualitative [1] e dalle simulazioni preliminari è emerso che una volta definiti i parametri di regime all'interno del motore, quali pressione, flusso di idrogeno in ingresso e raggio della camera di combustione, i frammenti di fissione ionizzati nel loro tragitto all'interno del gas provocano i seguenti fenomeni:

- Cessione di energia con conseguente forte riscaldamento
- Dissociazione delle molecole di $H_2$ e conseguente
- Ionizzazione degli atomi H

Data l'alta temperatura (>9000 K) raggiunta dal gas, si è reso necessario studiare gli effetti del bilancio di radiazione da parte del gas medesimo, che per temperature superiori a 8000 K tende ad essere importante. Infatti il gas a quelle temperature emette grandi quantità di energia sotto forma di radiazione elettromagnetica che può essere sia assorbita dagli strati di gas adiacenti sia dispersa sulle pareti. Perciò nonostante le temperature del gas non sono tali da poterlo considerare nello stato di plasma, questi effetti e la loro interazione con

l’energia rilasciata dai frammenti di fissione diventano di fondamentale importanza per le prestazioni del motore.

Quindi in aggiunta a quanto definito nel documento programmatico si è reso necessario lo sviluppo un ulteriore codice denominato PHOTON basato su una struttura simile a quella utilizzata per il codice FF_HEAT per studiare gli effetti di radiazione e la loro interazioni con gli altri fenomeni (fluidodinamica e riscaldamento da frammenti di fissione) all’interno della camera. PHOTON è basato su una struttura geometrica del tutto simile a quella utilizzata per il codice FF_HEAT in maniera tale da poter utilizzare le stesse strutture dati e le matrici di trasformazione generate dal codice MAPPING.

In riferimento ai risultati ottenuti in [1] per avere una comprensione completa della complessa fenomenologia dovuta agli effetti di riscaldamento e di irraggiamento del gas all’interno della camera di combustione si rende necessaria una simulazione tridimensionale completa nella quale gli effetti fisici di riscaldamento del gas vengano accoppiati in tempo reale con le simulazioni fluido dinamiche.

Come già evidenziato nel modello uni dimensionale del motore spaziale riscaldato dai frammenti di fissione una simulazione realistica deve tenere in conto sia la geometria esatta sia le reali condizioni fluido dinamiche(in equilibrio termodinamico) del propellente.

Per poter effettuare una simulazione completa si è quindi reso necessario lo sviluppo di un nuovo strumento di simulazione basato su tecniche Montecarlo (MC) accoppiate con un solutore di Fluido Dinamica Computazionale (CFD) preesistente.

La maniera più immediata ci e' sembrata quella di mutuare gli schemi classici utilizzati in fisica delle particelle.

In particolare si e' cercato, in maniera differente da quanto avviene per i codici CFD di utilizzare un tipo di rappresentazione geometrica il più semplice possibile, in modo tale da poter gestire un grande numero di eventi (ordine centinaia di milioni) da cui poter estrarre risultati statistici con la possibilità di tenere sotto controllo gli errori introdotti tramite l'uso del metodo statistico.

Questo e' l'approccio classico delle metodologie Montecarlo, largamente utilizzate nelle simulazioni di fisica delle particelle. La scelta di utilizzare queste tecniche, in contrapposizione a quelle dei volumi finiti utilizzate nei codici fluido dinamici e' dovuta alla necessita' di trattare griglie cartesiane regolari che ci permettono di utilizzare algoritmi di propagazione dei raggi (ray tracing) in maniera algoritmicamente semplice con conseguente abbassamento dei costi computazionali.

In particolare un codice che sia in grado di simulare la fisica all’interno della camera di combustione richiede una serie di strumenti che descrivano:

L’attivazione e l’evoluzione temporale dello strato di materiale fissile (Neutronica: rapporto [3])

Il trasporto dei frammenti di fissione generati dallo strato fissile all’interno della camera di combustione

Il trasporto di energia di radiazione dovuta ai fotoni generati ad alta temperatura

L’evoluzione degli effetti termodinamici dovuti al riscaldamento del gas e degli effetti di radiazione

Non essendo disponibile codici commerciali che siano in grado di risolvere contemporaneamente problematiche relative ai primi tre punti, abbiamo deciso di sviluppare un nostro codice ab initio.

Il sistema integrato permette di produrre risultati quantitativi relativamente ai processi globali di riscaldamento del propellente all'interno della camera di combustione.

Il punto 1) relativo alla perdita di energia dei frammenti di fissione all'interno del sottile strato di materiale fissile, qualche mg/cm2 di Americio 242, viene ampiamente descritto nel Cap. 3 della referenza [Bibbia].

I codici atti a simulare il trasporto dei neutroni e il loro rilascio all'interno dello strato di materiale fissile sono descritti ampiamente nel rapporto [3] a cui rimandiamo.

I dati provenienti dalle simulazioni di neutronica sono da considerarsi come un dato in ingresso per quanto riguarda il codice integrato. Difatti una volta definiti le geometrie ed i materiali costitutivi del motore, la distribuzione di energia rilasciata dalle pareti di materiale fissile risulta univocamente determinata.

Un attenzione particolare è stata riposta nello sviluppo dei codici Montecarlo che simulano il trasporto dei frammenti di fissione (rapporto [1], cap. 5) e per la simulazione degli effetti del trasporto dell'energia di radiazione (rapporto [1], cap. 6).

Fondamentalmente questi codici estendono al caso tridimensionale i modelli fisici applicati al caso uni dimensionale per il riscaldamento e l'irraggiamento nel gas, e descritti nel rapporto [1] che risulta essere la base teorica per il nostro lavoro.

Inoltre tutti i codici sono stati sviluppati secondo una procedura che risulta essere indipendente dall'architettura della macchina parallela utilizzata (nel nostro caso specifico risulta essere una Convex V-Class con 16 processori in parallelo) ed in maniera tale da fornire un ciclo completo relativo ad una simulazione nell'ordine di qualche giorno.

In questa sezione intendiamo presentare la struttura generale del programma integrato, le metodologie di accoppiamento tra le varie strutture del codice e fornire argomentazioni dettagliate circa gli algoritmi stocastici (Montecarlo) sviluppati all'interno di ciascun codice. Per i risultati dettagliati delle simulazioni effettuate con il codice completo si rimanda alla sezione al rapporto [3].

## STRUTTURA DEL CODICE

L'architettura generale del codice da noi sviluppato è mostrata nella Figura 1.

I componenti principali sono rappresentati dai blocchi che singolarmente risolvono le tre situazioni fisiche dominanti all'interno del motore :

- La fluidodinamica del gas in equilibrio termodinamico (KARALIS)

- Il riscaldamento del gas da parte dei frammenti di fissione(FF_HEAT)
- Gli scambi di calore tramite irraggiamento (PHOTON).

I tipi di strategie ed algoritmi utilizzati per affrontare e risolvere fenomeni fisici così differenti tra di loro hanno richiesto approcci alle geometrie ed alle strutture dati diametralmente opposte.

Se dal punto di vista della Fluido Dinamica Computazionale le tecniche deterministiche usuali di soluzione tramite schemi a volumi finiti permettono un supporto geometrico basato su griglie esaedriche di forma e volume variabili in maniera tale da meglio seguire la dinamica fortemente localizzata e non lineare, nel caso dei codici FF_HEAT e PHOTON basati su metodologie Montecarlo la natura intrinsecamente statistica degli algoritmi richiede una geometria basata su celle cubiche di uguale volume.

Questo è un approccio largamente utilizzato nei codici nelle simulazioni di fisica delle particelle, nelle quali si trattano enormi quantità di eventi che vengono esaminati in seguito in maniera statistica.

Data la diversa costituzione delle geometrie e delle strutture dati sulle quali sono basati i codici si è reso necessaria l'implementazione di un ulteriore strumento di supporto per poter maneggiare gli scambi di dati tra i diversi codici:

Il codice MAPPING converte le informazioni tra le due strutture dati.

L'operazione di trasformazione dei dati svolta ad ogni iterazione potrebbe sembrare comunque pesante dal punto di vista computazionale ed allo stesso tempo poco elegante. Risulta essere invece necessaria dato l'approccio completamente diverso dei due tipi di codici (deterministico KARALIS, statistici FF_HEAT e PHOTON) che richiedono griglie completamente diverse. Nel primo caso infatti è necessario avere griglie di forme e dimensioni variabili che seguano le differenti situazioni dinamiche nelle diverse zone della camera, nel caso dei codici Montecarlo è invece di fondamentale importanza che le celle (voxels) siano tutte identiche in modo tale da poter replicare in ognuna di esse la medesima statistica. Infine il tempo utilizzato per convertire i dati da un formato all'altro è trascurabile (qualche minuto) rispetto ai tempi tipici delle simulazioni.

Informazioni più dettagliate sullo sviluppo del codice MAPPING sono fornite nell'allegato 2 .

**KARALIS** è un codice di Fluido Dinamica Computazionale interno del CRS4 che risolve le equazioni di Eulero e di Navier-Stokes anche in presenza di regimi turbolenti e di combustione.

Il metodo di discretizzazione usa uno schema a celle centrate (in cui l'informazione fisica in ogni elemento è definita nel centro cella) basato su un metodo ad Elementi Finiti ed avanzamento in tempo basato su algoritmi Runge-Kutta multi-stage. Tra le peculiarità del codice vi è la presenza del metodo di soluzione multi blocco che permette di risolvere geometrie complesse e l'utilizzo del metodo multi griglia che ne accelera la convergenza. Sulla base di queste caratteristiche una parallelizzazione del codice, con tecniche MPI, permette di ottenere ottimi risultati e lo rende altamente compatibile con tutte le architetture.

**FF_HEAT** è il codice sviluppato appositamente per il progetto e come detto precedentemente permette di simulare il processo di diffusione dei frammenti di fissione rilasciati dalle pareti della camera di combustione all'interno della camera stessa ed il relativo rilascio di energia nel gas. Il codice, che viene ampiamente descritto nel paragrafo successivo, è stato sviluppato in modo tale da poter sfruttare nella maniera migliore possibile le potenzialità delle macchine parallele. Difatti essendo gli eventi, in questo caso i frammenti di fissione liberati dalle pareti, tra loro completamente non correlati possono essere seguiti e valutati indipendentemente e sommati statisticamente i dati relativi ad ogni processore.

**PHOTON** è un codice che non risulta inserito nel documento programmatico, ma che è stato sviluppato in corso d'opera. Il problema dell'innalzamento della temperatura all'interno della camera di combustione ha richiesto infatti di sviluppare un codice che permettesse di studiare gli scambi termici dovuti all'emissione e all'assorbimento di radiazione all'interno del gas. La sviluppo del codice è avvenuto su basi simili a quelle utilizzate per il codice FF_HEAT utilizzando quindi geometrie, strutture dati ed algoritmi simili sui quali vengono implementate le routines relative l'irraggiamento ed all'emissione descritte nel paragrafi seguenti.

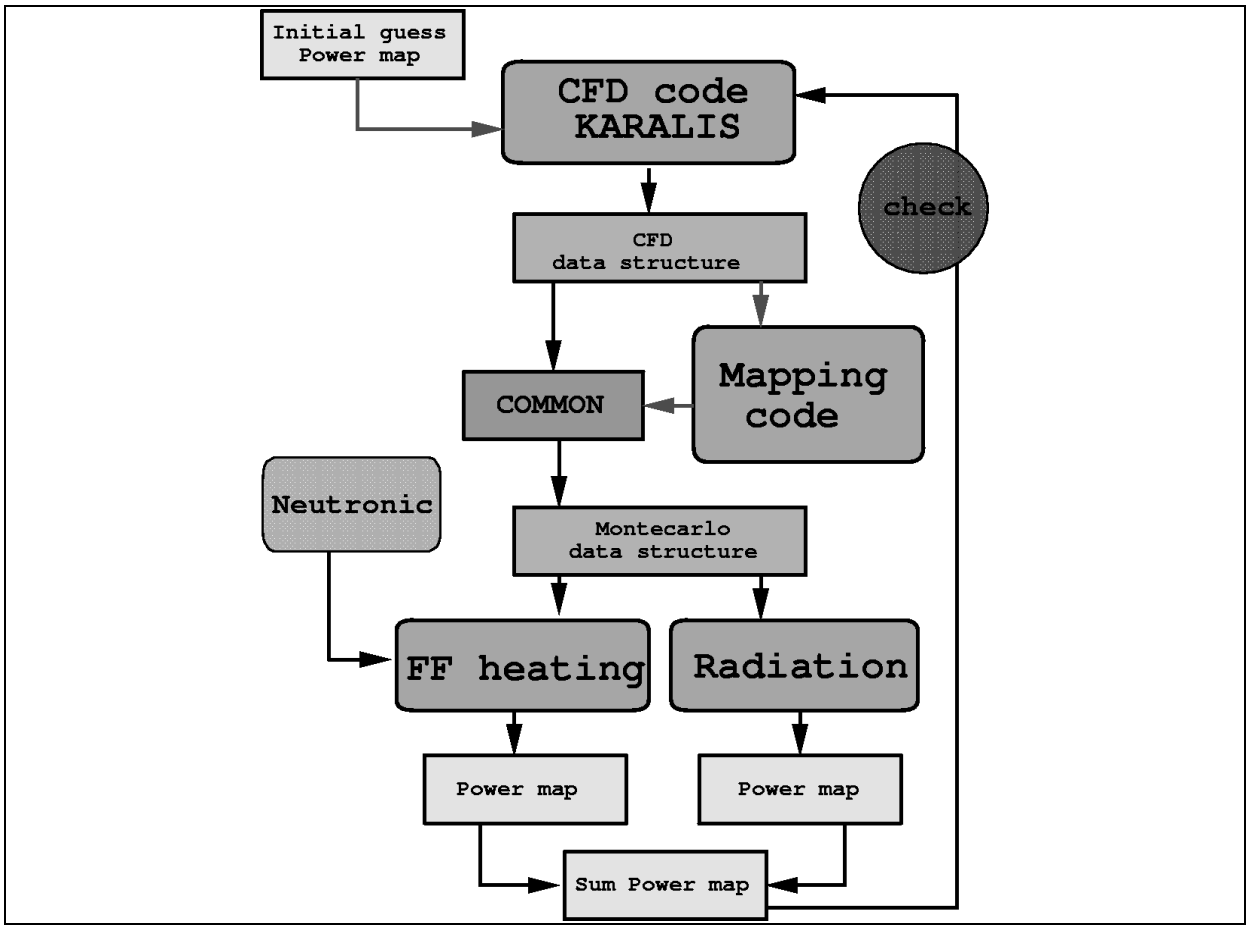


*Figura 1 : Architettura Generale del programma*

La figura 1 schematizza in un diagramma a blocchi il funzionamento del codice che simula la fisica completa del gas nella camera di combustione.

Il codice di fluidodinamica richiede come dati di ingresso la definizione delle variabili fisiche (pressione, temperatura e densità) definite sulla griglia esagonale costruita all'interno del cilindro.

Alla prima iterazione viene fornita in ingresso al codice CFD una serie di mappe che tentano di avvicinarsi a quella che si suppone essere la forma finale. Una volta raggiunto lo stato stazionario in seguito all'evoluzione dinamica sulle mappe iniziali i risultati escono dal codice CFD ancora definiti sulle griglie esagonali. A questo punto il codice MAPPING trasforma i dati in uscita dal codice CFD nel formato di input necessario ai codici Montecarlo.

Le mappe di densità e temperatura vengono quindi passate in input ai codici FF_HEAT e PHOTON. I due codici girano indipendentemente in parallelo e forniscono in uscita le nuove mappe di distribuzione dell'energia depositata nel gas dai frammenti di fissione e ridistribuita tramite l'irraggiamento e assorbimento.

Le mappe di energia così ottenute rappresentano il nuovo input per il codice di CFD avviando così una nuova iterazione del processo.

Ad ogni iterazione vengono perciò ripetuti cinque passi:

- Calcolo della CFD (codice KARALIS)
- Operazioni di pre-processing, identificazione delle matrici di trasformazione dati; mapping dei dati dal formato CFD al formato MC
- Calcolo del riscaldamento dovuto ai frammenti di fissione (codice FF_HEAT)
- Calcolo del trasporto di radiazione (codice PHOTON)
- Operazioni di post-processing somma e remapping dei dati dalle strutture MC a quelle CFD.

Le iterazioni vengono ripetute sino a che il calcolo non arriva a convergenza, cioè tutti i processi raggiungono uno stato stazionario.

# CALCOLO DELLA DISTRIBUZIONE INDOTTA DEI FRAMMENTI DI FISSIONE: IL CODICE FF_HEAT

Il calcolo della distribuzione indotta dai frammenti di fissione all'interno della camera di combustione deve tenere in conto oltre che di un accurato modello fisico, della la reale geometria del motore e delle condizioni del gas, in equilibrio termodinamico presenti all'interno della camera di combustione. Queste ultime sono ricavate dalle simulazioni effettuate tramite il programma KARALIS.

Il modello di calcolo usa un approccio Monte Carlo per implementare, in tre dimensioni, il modello mono-dimensionale descritto nel rapporto [1] (paragrafo 5.7). I suoi dati di input sono:

- i dati relativi ai frammenti di fissione prodotti all'interno del N-Hohlraum, e simulati dal programma di neutronica;
- i valori delle mappe delle variabili fisiche che entrano in gioco nell'interazione tra frammenti di fissione e gas (es. densità del gas);
- la geometria della camera di combustione il sistema.

Partendo da tali dati il codice e' in grado di simulare completamente il processo di distribuzione di energia indotta dai frammenti di fissione all'interno della camera di combustione.

Questo rapporto presenta la descrizione della tecnica senza entrare in dettaglio sulla fisica del processo di interazione tra FF e gas (ampiamente descritti nel capitolo 5 del rapporto [1] a cui si rimanda il lettore per ogni chiarimento). Più specificatamente presenta la struttura complessiva dell'algoritmo Monte Carlo sviluppato ed i risultati su un caso test significativo.

Questo lavoro e' stato realizzato dal CRS4 sotto la supervisione scientifica diretta del Prof. Rubbia a cui si deve parte dell'implementazione del codice stesso.

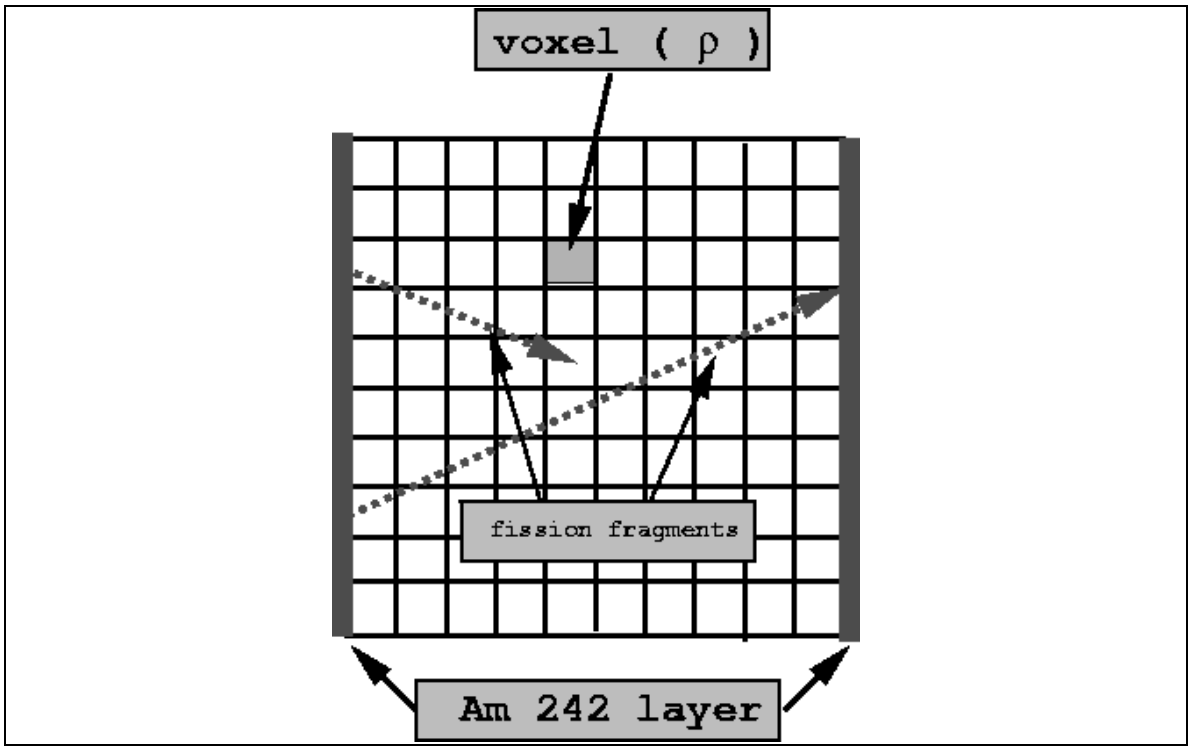


*Figura 2: Campionamento Monte Carlo della struttura a griglia regolare per la simulazione del processo di riscaldamento del gas: i frammenti di fissione (frecce in rosso) uscendo dalle pareti ricoperte di Americio attraversano il gas (la griglia regolare) rilasciando la loro energia cinetica sino a quando raggiungono una energia minima e si fermano all'interno del gas stesso (freccia in alto) o raggiungono la parete ed escono dalla camera di combustione (freccia in basso).*

La Figura 2 illustra il principio generale della tecnica Monte Carlo utilizzata. L'algoritmo sviluppato si basa sull'idea di seguire il comportamento del frammento di fissione (indicato successivamente con la sigla FF) durante la sua vita accumulando, su strutture dati opportune (array a tre dimensioni), le informazioni che consentono di risalire alla mappa di energia del processo e di riscaldamento del gas complessivo. In questo contesto vengono quindi modellate ed implementate le interazioni che ogni singolo frammento di fissione, uscente dallo strato di Americio, ha con il gas e le pareti della camera di combustione. Infatti il ff uscendo dallo strato di Americio depositato sulla parete della camera, dopo aver perso parte dell'energia cinetica interagendo con lo strato stesso (il trasporto dei FF all'interno dello strato di Americio e' calcolato dal modulo UniGamba illustrato in [3]) entra all'interno della regione cilindrica riempita di gas. Questa regione e' la camera di combustione. Nella camera il FF perde un'ulteriore parte (quella efficace) di energia cinetica interagendo con l'idrogeno con l'effetto di riscaldarlo. Se il gas ha una densità sufficientemente elevata (vedere Figura 5.10 del rapporto [1]) tutta l'energia viene trasferita al gas prima di raggiungere la parete opposta ottenendo la massima efficienza del processo di riscaldamento del gas. Se invece il gas e' poco denso o la direzione del FF e' poco efficace (ad esempio, il diametro del cilindro risulta troppo piccolo e vengono a generarsi molti FF con una direzione quasi parallela alla superficie della camera per cui percorrono solo un piccolo tratto del loro cammino all'interno del cilindro), il FF perde solo una piccola parte della sua energia nel gas mentre la restante parte viene trasferita alle pareti.

Quindi per simulare adeguatamente il comportamento dei FF dobbiamo essere in grado di calcolare il percorso geometrico dei ff, di calcolare l'energia persa dai FF (che dipende dalla loro energia cinetica, della distanza percorsa all'interno della camera e della densità del gas).

L'idea e' quella di suddividere lo spazio geometrico che rappresenta la camera di combustione (il singolo tubo senza la parte convergente viene preso per il momento in considerazione) in celle (parallelepipedi rettangolari) non sovrapposte. A ciascuna di queste celle e' assegnato il valore (di media) delle variabili fisiche (in equilibrio termodinamico) che entrano in gioco nel processo di interazione tra frammento di fissione gas propellente (idrogeno nel nostro caso). La quantità fisica per il momento considerata nella simulazione e' esclusivamente la densità del gas.

La struttura dati che ne deriva e' una griglia regolare (una enumerazione spaziale per volumi chiamata lattice) che necessita di un algoritmo per l'attraversamento delle celle che abbiamo chiamato PathFinder che simula il percorso dei FF nella griglia. Le celle o volumi base chiamati voxel memorizzano tutti i valori delle variabili fisiche scalari (densità, temperatura, pressione) e vettoriali (velocità) del gas prodotte dal codice di fluido-dinamica[1].

Il percorso di un FF nel gas e' rappresentato geometricamente con un raggio rettilineo che ha un'origine, il punto in cui il FF nasce, ed un vettore direzione, i coseni direttori.

Dato un FF, la procedura PathFinder determina l'insieme delle celle attraversate dal FF lungo il suo cammino e, incrementalmente, per ogni cella, la distanza che il FF percorre all'interno dei confini della cella stessa.

Gli FF hanno sempre origine sulla superficie del fine strato di Americio (la superficie interna del cilindro a contatto con il gas) in una posizione casuale della superficie della camera. Ogni FF[2] attraversa la griglia di celle fino a che o esce dalle pareti del cilindro o perde completamente la sua energia interagendo con il gas.

Dal punto di vista dell'organizzazione del codice che implementa la tecnica Monte Carlo possiamo considerare che la perdita di energia di un FF all'interno di una singola cella dipende sostanzialmente da tre fattori:

1) dall'energia cinetica del ff;
2) dalle condizioni termodinamiche del gas[3], in particolare dalla densita';
3) dalla lunghezza dell'intervallo percorso dal FF all'interno della cella.

Conoscendo i dati di cui sopra possiamo calcolare, per ogni cella attraversata dal FF, l'energia persa, che viene quindi accumulata nel gas.

Poiché ciascuna cella somma i singoli contributi dei vari FF che la attraversano, lanciando un numero di FF statisticamente significativo[4] possiamo calcolare la distribuzione dell'energia depositata dai frammenti di fissione nel gas.

---

[1] I dati memorizzati nella griglia sono prodotti in un passo preliminari chiamato Mapping [referenza Mapping].

[2] sono tenuti in considerazione solo i FF che hanno una energia cinetica sufficiente a farli sopravvivere all'interazione con l'Americio e si propagano all'interno della camera di combustione; questi frammenti costituiscono circa il 30 per cento dei totali e vengono propagati verso la superficie dello strato in un passo di pre-elaborazione prodotto dal programma UniGamba

[3] all'interno del processo di simulazione Monte Carlo noi supponiamo essere sempre in condizioni di equilibrio termodinamico

Il processo di accumulazione dell'energia eseguito dal singolo frammento di fissione può essere schematizzato dalle seguenti fasi di programma:

**Fase 1**) si lancia un FF in una posizione casuale della superficie cilindrica (strato di Americio di spessore fissato) con una direzione casuale prelevando i dati che lo caratterizzano (A, Z, Energia cinetica, etc.) dall'archivio dei FF generato dalla neutronica.

**Fase 2**) si identifica la cella attraversata dal FF lungo la direzione di propagazione e si calcola la distanza percorsa dal FF nella cella.

**Fase 3)** si calcola l'energia depositata e la si accumula nel lattice;

**Fase 4)** si valuta se l'energia va al di sotto di un prefissato valore di soglia; se il test e' positivo si va alla Fase 7);

**Fase 5)** si valuta se FF esce dalla camera di combustione; se il test e' positivo si va alla Fase 7);

**Fase 6)** si torna allo Step 2);

**Fase 7)** si esce dal loop

*Codice 1: schema dell'algoritmo Monte Carlo per il calcolo del processo di riscaldamento del gas mediante frammenti di fissione*

Da notare che il calcolo del processo di accumulazione e' indipendente per ogni ff. Il calcolo e' facilmente implementabile in parallelo su una macchina multiprocessore assegnando un seed point di partenza diverso nell'inizializzazione della libreria di calcolo dei numeri casuali. Abbiamo operato questa nell'implementazione del codice Monte Carlo e ci permette di ottenere una velocizzazione pressoché lineare dell'algoritmo con il numero dei processori.

Nell'implementazione parallela, realizzata mediante la libreria standard MPI, le griglie regolari (replicate per ogni processore) che memorizzano la mappa delle variabili fisiche e la geometria della camera vengono distribuite sui vari processori in una fase iniziale del programma ovvero prima che il calcolo su frammenti di fissione inizi.

Ogni processo di accumulazione di dati viene quindi fatto partire con un "seed point" differente. A questo punto ogni processore accumula i dati (statististicamente

[4] poche centinaia di milioni di FF sono sufficienti per non avere una variazione significativa della mappa di energia depositata nel gas.

indipendenti dagli altri processi) sulle proprie matrici  locali durante il loop che propaga i FF.

Alla fine dei loop, ovvero quando la statistica accumulata e' ritenuta essere adeguata, le matrici di tutte le energie memorizzate nei vari processori vengono rinviate e sommate assieme sul processore master.

Possiamo quindi normalizzare i valori accumulati[5] della griglia prendendo in considerazione la potenza totale generata dallo strato di Americio e la massa di gas presente in ogni cella per ottenere l'energia depositata in ciascuna cella.

A questo punto, mediante l'algoritmo Mapping, siamo in grado di fornire la mappa di energia rilasciata nel gas per la prossima iterazione del codice di fluido-dinamica.

La prossima sezione illustra in maggior dettaglio la struttura dell'algoritmo parallelo implementato.

[5] un grosso sforzo implementativo e' stato fatto per evitare errori di arrotondamento e di overflow (superamento del piu' grande valore numerico rappresentabile dal calcolatore) molto insidiosi in questo genere di simulazioni

**STRUTTURA DEL CODICE MONTECARLO PER IL CALCOLO DELLA DISTRIBUZIONE DEI FRAMMENTI DI FISSIONE**

Nei COMMON del programma Fortran il programma **MAPPING** ha memorizzato, su degli array a tre dimensioni, le mappe delle variabili fisiche generate dal codice di fluidodinamica. La struttura dell'algoritmo parallelo implementato nel codice Monte Carlo e' il seguente:

- **Processore 0**:

> ***Step 0)*** carica le variabili in common (geometria, mappa densità, energia dei ff nell'Am242 generati dal codice Mapping) dal file da file memorizzato su disco
>
> ***Step 1)*** legge da input il numero di ff da campionare (variabile *NSAMPLE)* e distribuisce i dati presenti nel common sui *NUMPROCS* processori paralleli

- **Ogni processore in parallelo**:

> ***Step 2)*** riceve i dati dal Processore 0 e genera una sequenza indipendente (dagli altri processori) di numeri casuali;
>
> ***Step 3)*** *lancia il numero prefissato* ***NSAMPLE/NUMPROCS*** *di frammenti di fissione accumulando l'energia da loro persa nella propria mappa (lattice): esegue il seguente loop per* ***NSAMPLE/NUMPROCS*** *volte*:
>
> > ***Step 3.A)*** se il numero dei campioni lanciato e' inferiore a (NSAMPLE/NUMPROCS) genera un nuovo FF associando un'energia cinetica, un'origine casuale (all'interno del layer di Am242) ed un direzione di propagazione casuale altrimenti va allo Step 4)
> >
> > **Step 3.B)** se il FF **non esce** dal layer torna allo Step 3.A)
> >
> > **Step 3.C)** se il FF e' orientato nella direzione opposta rispetto alla posizione della camera di combustione torna allo Step 3.A)
> >
> > **Step 3.D)** *accumula l'energia persa dal FF (e diminuisce l'energia cinetica associata al FF stesso) nelle celle attraversate lungo il suo cammino finche' l'energia cinetica va al di sotto di un valore di soglia o il FF esce dalla camera di H2*:

**Step 3.D.a)** determina la cella corrente (gli indici I,J,K) attraversata dal FF

**Step 3.D.b)** calcola la distanza **DWALL** percorsa dal FF nella cella (I,J,K) corrente

**Step 3.D.c)** calcola l'energia depositata nella cella (I,J,K) che dipende dalla densità di H2 RO, da DWALL, e dall'energia cinetica T_ION

**Step 3.D.d)** accumula l'energia depositata nella cella (I,J,K)

**Step 3.D.e)** decrementa l'energia residua T_ION associata al FF

**Step 3.D.f**) se l'energia residua T_ION e' al di sotto del valore di soglia esci dal loop e lanciare un nuovo FF; vai allo Step 3.A)

**Step 3.D.g)** se il FF esce dalla camera di idrogeno esci dal loop e lancia un nuovo FF; vai allo step 3.A)

**Step 3.D.h)** ritorna allo step 3.D.a)

**Step 3.E)** fine propagazione FF; ritorna allo Step 3.A)

- **Processore 0:**

***Step 4)*** raccoglie le mappe di energia dai vari processori e calcola la media

***Step 5)*** esegue il remapping (il processo di conversione verso il codice CFD) della mappa di energia e le memorizza su disco.

*Codice 2: Pseudo Codice del programma parallelo FF_HEAT*

**RISULTATI DEL CODICE FF_HEAT**

La tabella 1. mostra i parametri di input per il programma di FF heating assumendo che solo un singolo cilindro della configurazione del motore sia riempito di idrogeno ad una pressione di 6 atm e collocato all'interno del N-Hohlraum.

*Tabella 1. Parametri di Input per il codice FF heating*

| | |
|---|---|
| Pressione del gas: | 6.0 atm |
| Spessore dello strato di Americio (Am242*): | 0.3E-05 m |
| Raggio del cilindro che supporta lo strato di combustibile: (Am242*): | 0.2 m |
| Altezza del cilindro che supporta lo strato di combustibile: (Am242*): | 2.5 m |
| Risoluzione griglia (x): | 202 |
| (y): | 202 |
| (z): | 202 |
| Dimensione cella (x): | .2E-02 m |
| (y): | .2E-02 m |
| (z): | 1.25E-02 m |
| Superficie Attiva di Am242*: | 3.1416 $m^2$ |
| Densità di potenza dell'Am242*: | 200.00 $W/cm^2$ |
| Potenza Totale del motore (Gross Engine thermal power): | 6.2832 Mwatt |
| Numero di FF lanciati: | 200E+6 |

La Fig. 3 mostra la mappa di densità prodotta dal programma di fluidodinamica numerica (CFD) che costituisce l'input del programma. Queste mappe dovrebbero rappresentare le reali condizioni del gas propellente (idrogeno) all'interno della camera di combustione. Si noti che il gas e' sempre iniettato attraverso le pareti della camera.

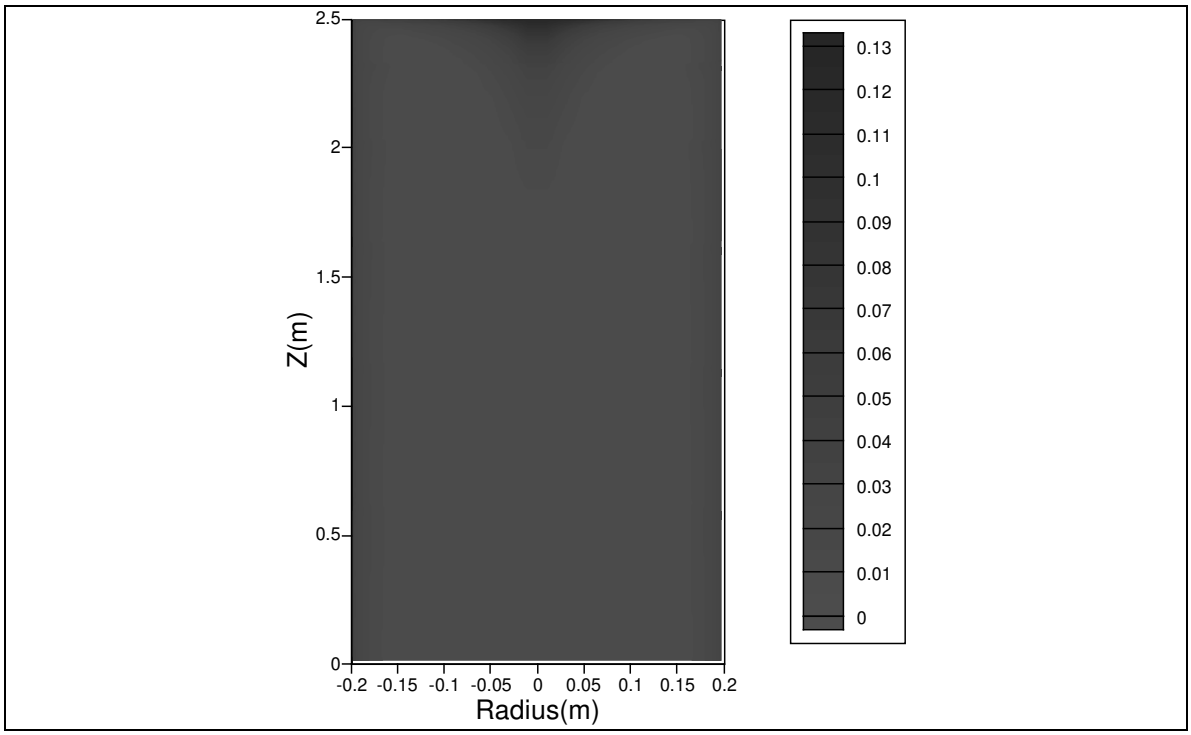


*Figura 3: Input del Programma FF_HEAT: Mappa di densità RO del gas idrogeno*

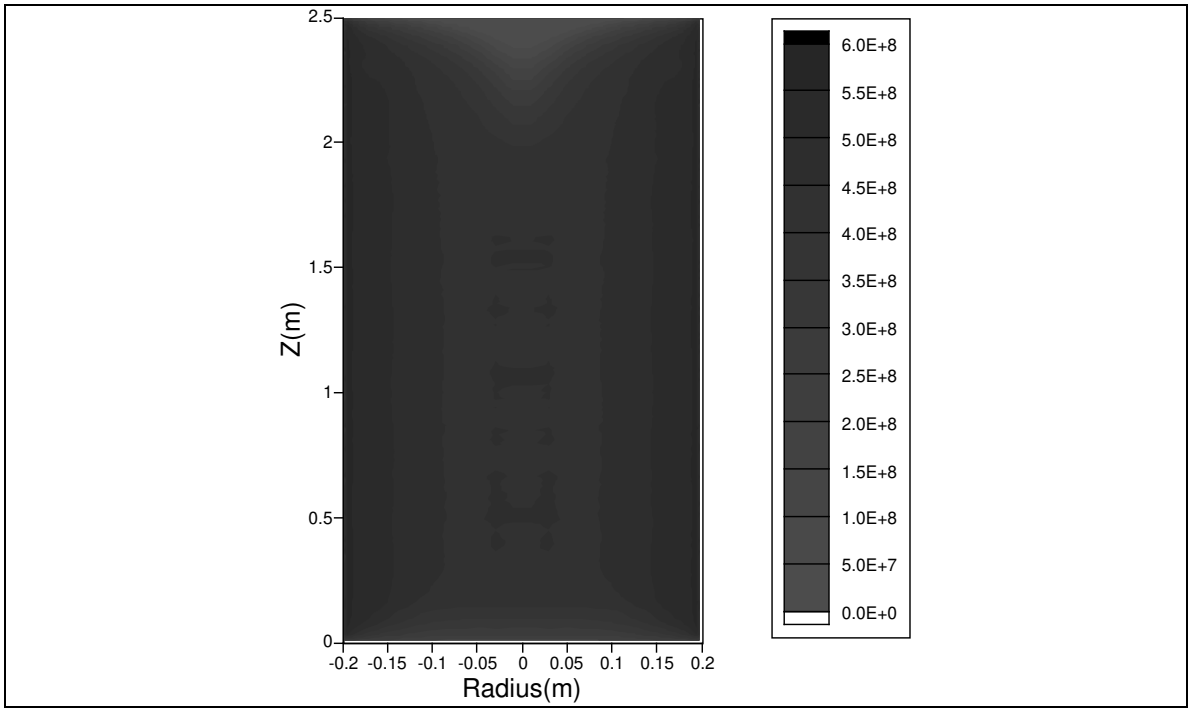


*Figura 4: Output dell'algoritmo Monte Carlo per il riscaldamento del gas mediante frammenti di fissione*

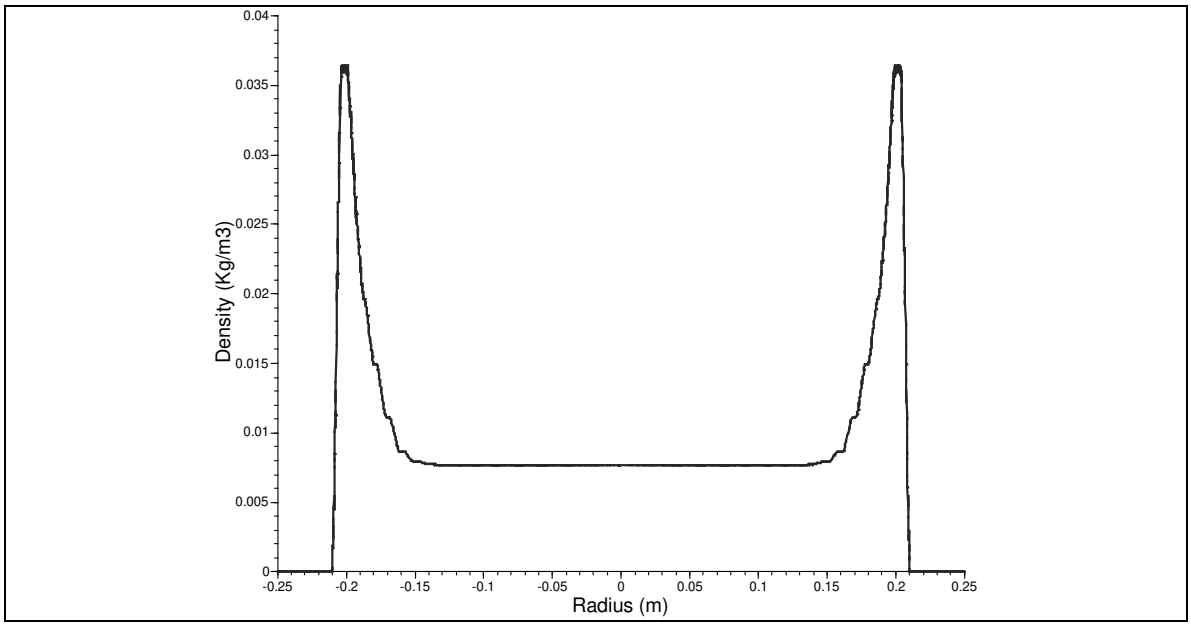


*Figura 5: Input dell'algoritmo Monte Carlo: mappa delle densita' su una sezione trasversale ad una altezza di 0.44 dalla base del cilindro*

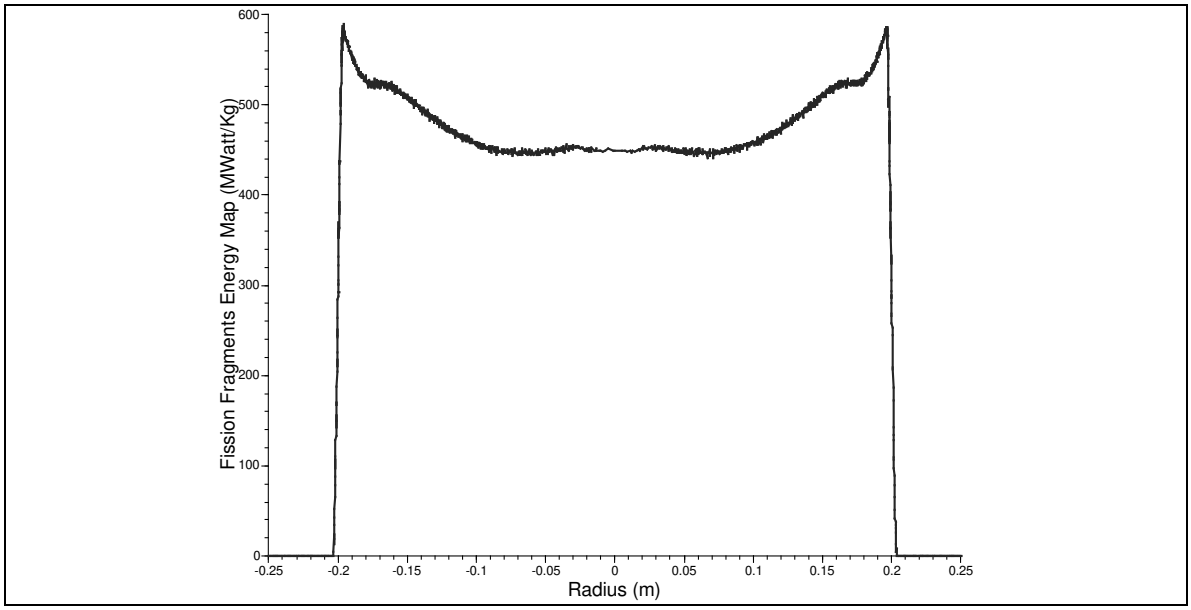


*Figura 6: Output del programma Monte Carlo: sezione trasversale della mappa di densità di potenza sulla griglia regolare ad una altezza di 0.55 m dal basso. Si possono notare le fluttuazioni statistiche dovute al processo di campionamento*

Possiamo osservare una regione con densità costante vicino al cuore (la regione centrale) e un più grande variazione vicino alle pareti più fredde. Questo comportamento può essere meglio osservato in Fig. 6 la quale mostra una sezione orizzontale del campo di densità del gas ad un'altezza di 0.55 metri (dalla base del cilindro). La reale densità del gas (determinata dagli effetti di termofluido-dinamica) e l'uso della geometria cilindrica rappresentano le principali differenze rispetto al modello mono-dimensionale (descritto

in [1]) dove erano stati assunti una densità costante del gas e due piani infiniti paralleli.

*Tabella .2 Dati di output del codice FF_HEAT (solo i frammenti di fissione che vanno entrano nella camera di combustione, sono la meta' di quelli che escono dallo strato di Americio sono tenuti in considerazione)*

| | |
|---|---|
| Energia Tot. rilasciata dai FF: | 3.1416 MWatt |
| Energia Tot. rilasciata dallo stato di combustibile: | 1.918 MWatt |
| Energia totale depositata nell'idrogeno: | 1.8463 MWatt |
| Energia Tot. che rientra nelle pareti: | 0.0717 MWatt |
| Frazione di FF energia che lascia lo strato di Americio: | 61.052 |
| Frazione di energia depositata nell'idrogeno: | 58.770 |
| Frazione di energia che riesce dalle pareti: | 2.2817 |
| Frazione del numero di frammenti di fissione che riescono dalle parete: | 9.7902 |
| Massa totale del gas: | 3.9818 g |
| Densità di potenza nel gas: | 470.27 W/mg |

L'output del codice di riscaldamento mediante frammenti di fissione sono illustrati nella Tabella 2. Questi risultati sono puramente indicativi nel senso che considerano un caso test. Si rimanda al rapporto [4] finale sulla fluidodinamica per le valutazioni complessive del processo di riscaldamento. Il nostro obiettivo in questa discussione e' illustrare le funzionalità e i risultati del programma FF heating a confronto con il modello mono-dimensionale.

Nella Tabella 2 possiamo inoltre vedere che in questa particolare configurazione geometrica della camera e nelle condizioni di pressione (6 atm) solamente il 2% dell'energia trasportata dai frammenti di fissione esce dalle pareti della camera di combustione. La Fig. 4 mostra la mappa di energia depositata dai ff. Come previsto dal modello 1D il valore massimo di energia (dell'ordine di 600 Mwatt/Kg) lo si ottiene vicino alle pareti (più fredde) del cilindro dove la densità del gas risulta più alta ($\rho$ e' circa 0.478 Kg/m3). La Fig. 6 invece mostra una sezione trasversale della mappa di energia depositata ad un'altezza di 0.55 m dalla base del cilindro.

Esaminando le figure e le tabelle in cui sono riportati gli output delle simulazioni si evidenzia come questi risultati siano in ottimo accordo con quanto previsto dai modelli unidimensionali presentati nel capitolo 5 del rapporto [1]. Per questa particolare configurazione della camera di combustione (raggio 0.2 m, pressione 6 atm) e con una quantità di materiali fissile distribuita sulle pareti pari a 3 mg-cm$^2$, le mappe di densità di potenza (fig. 5 e fig.7) sono in ottimo accordo con quanto previsto dai modelli preliminari introdotti (fig. 5.13 rapporto [1]). Questa particolare configurazione ha quindi la caratteristica di far in modo che la gran parte dell'energia dei FF venga effettivamente ceduta all'idrogeno in quanto la frazione di energia dei FF dispersa sulle pareti è dell'ordine del 2-3%. Logicamente i risultati del codice FF_HEAT vanno letti

nell'ottica del suo accoppiamento con il codice fluidodinamica. Infatti i risultati discussi sono relativi allo stato stazionario, al momento in cui le iterazioni del codice integrato(KARALIS+FF_HEAT+PHOTON) sono arrivate a convergenza. Maggiori dettagli sui risultati delle simulazioni del codice integrato, in varie configurazioni di pressione e raggio della camera, sono riportate nel rapporto [4]. L'implementazione del cuore dell'algoritmo Monte Carlo, quella computazionalmente più costosa, risulta oltre che performante estremamente semplice ed elegante. Il modello fisico implementato dal Prof. Rubbia e' richiamato tramite pochissime routine di interfaccia dall'algoritmo di campionamento Monte Carlo.

L'algoritmo parallelo scala linearmente con il numero di processori. Inoltre l'algoritmo per l'attraversamento della griglia regolare utilizzato intensivamente nel calcolo dell'energia depositata nelle celle e' molto efficiente.

## CALCOLO DELLA DISTRIBUZIONE DI ENERGIA NEL GAS PROPELLENTE AD ALTA TEMPERATURA: IL CODICE PHOTON

Come descritto nel capitolo 6 del rapporto [1], il gas propellente ad alta temperatura emette energia radiante, che viene sia riassorbita dal gas vicino sia dissipata sulle pareti della camera di combustione. Questo processo può essere simulato nelle reali condizioni di utilizzo del motore quando si hanno a disposizione la geometria del motore stesso (la configurazione della camera di combustione) che le condizioni termodinamiche gas propellente (in equilibrio termodinamico) generate dal programma di fluidodinamica numerica (KARALIS3).

Il modello per il calcolo sviluppato in questo contesto utilizza un approccio Monte Carlo, in tre dimensioni, per implementare il modello fisico uni dimensionale illustrato in [1]. Più specificatamente, partendo dalle mappe quantitative sulle quantità fisiche (che entrano in gioco nel processo fisico di irraggiamento come, ad esempio, densità del gas, temperatura) e dalla geometria della camera di combustione, il sistema e' in grado di calcolare il processo globale di trasferimento dell'energia dovuta a radiazione. Senza entrare nel merito della fisica del processo di interazione tra FF e gas (descritti nei paragrafi 6.2, 6.3, 6.4, 6.5 del rapporto [1]) questo documento presenta la struttura dell'algoritmo Monte Carlo.

## LA TECNICA MONTE CARLO IMPLEMENTATA

L'elemento fondamentale del processo di simulazione Monte Carlo e' il fotone. I fotoni hanno origine nelle regioni del gas che presentano un'alta temperatura (un valore di soglia di 8000 K e' assegnato per individuare le regioni del gas che emettono energia). Al fotone viene associato uno spettro di emissione calcolato mediante il modello descritto in 6.2 di [1]. Lo spettro di emissione dipende fondamentalmente dal tipo di gas propellente utilizzato (idrogeno nel nostro caso) e dalla temperatura della regione emettente. I fotoni si propagano nel gas perdendo immediatamente la maggior parte della loro energia nelle strette vicinanze della zona di emissione. Il resto dell'energia viene persa successivamente nel gas fino ad arrivare ad una valore di soglia per cui il gas diventa di fatto trasparente (figura 6.7 di [1]) al fotone e per cui l'energia esce dalle pareti della camera di combustione. Lanciando un numero statisticamente significativo di fotoni e seguendo il loro comportamento dalla nascita fino alla morte (ovvero quando lo spettro di energia raggiunge un minimo valore di soglia o il fotone raggiunge le pareti della camera) e' possibile calcolare il processo globale di trasferimento di energia. Per questioni di simulazione al computer definiamo due elementi che sono importanti nel processo di trasferimento di energia mediante fotoni: le regioni del gas che emettono energia che chiamiamo ***regioni emettenti*** e le regioni del gas che invece assorbono l'energia dai fotoni stessi che chiamiamo ***regioni assorbenti***.

Discretizzando la camera di combustione in volumi (celle) che dividono lo spazio geometrico cilindrico racchiudente il motore in parallelepipedi rettangolari all'interno di una griglia regolare, chiamata lattice, e calcolando, mediante il codice Mapping, per ogni cella le caratteristiche termo-dinamiche del gas (ovvero la sua temperatura, la sua densità RO), siamo in grado di simulare, applicando le formule descritte in [1.6] l'intero processo.

Classifichiamo come celle emettenti le celle che hanno una temperatura superiore al valore di soglia cioè le celle per cui e' significativo considerare gli effetti dovuti all'emissione. Le assorbenti sono di fatto tutte le celle della griglia che contengono il gas. Le celle di bordo, quelle che cavalcano la superficie del cilindro, vengono consirate completamente riempite di gas.

La Fig. 8 descrive l'idea generale dell'algoritmo Monte Carlo. I fotoni partono, in una direzione casuale e con una energia determinata dal modello di emissione proposto da Rubbia, all'interno di una cella emettente. L'energia trasportata da ciascun fotone e' data dalle formule descritte in 6.4 e 6.5 del rapporto [1]. Ciascuna cella attraversata dal fotone lungo il suo cammino viene considerata come una cella che assorbe energia. Abbiamo quindi considerato nell'implementazione del programma di calcolo prima la fisica per l'emissione (che vale solo per la prima cella quella da cui parte il fotone) e poi la fisica per l'assorbimento (che vale sia per la prima cella che per quelle successive). Inizialmente sia il processo si emissione che di assorbimento erano simulati nel codice PHOTON.

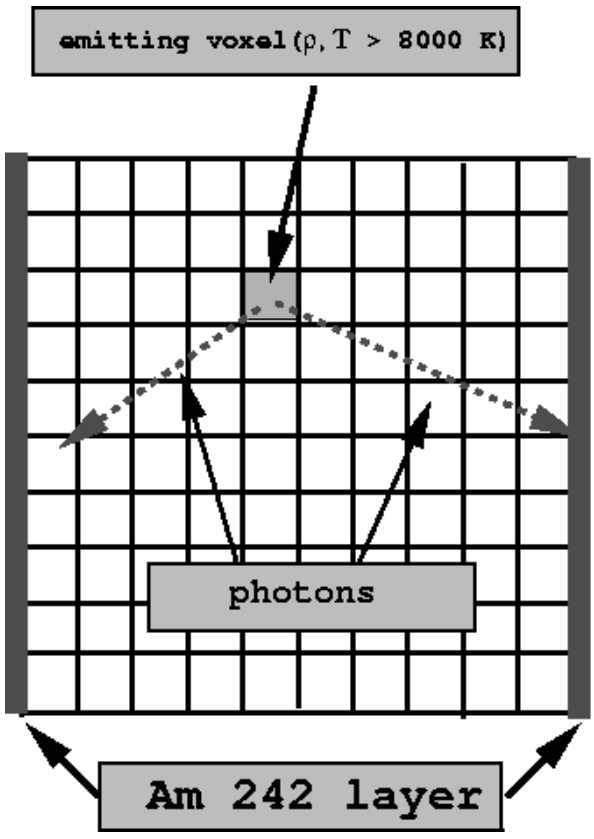


*Figura 7: Algoritmo Monte Carlo per il calcolo dell'energia trasportata mediante fotoni: due fotoni sono generato dalla cella emettendo. Uno di essi (quello di sinistra rappresentato dalla freccia segmentata) viene riassorbito prima di raggiungere le pareti della camera a differenza di quello di destra che scarica parte della sua energia sulle pareti*

In un secondo tempo, analizzando i risultati e la lenta convergenza della dinamica allo stato stazionario, si è deciso di implementare il processo fisico dell'emissione all'interno del codice di fluidodinamica e lasciare al codice PHOTON la prerogativa del processo di ridistribuzione dell'energia tramite il processo di riassorbiento. I dettagli relativi a questi cambiamenti ed i relativi risultati sono discusi dettagliatamente nel rapporto [5]. Campionando poche migliaia di fotoni per cella emettente riusciamo a produrre abbastanza statistica per descrivere il comportamento dell'intero sistema e quindi simulare il trasporto di energia tra celle che e' il nostro obiettivo.

Il codice sviluppato per la simulazione Monte Carlo del trasporto dell'energia radiata usa sostanzialmente la stessa organizzazione del programma utilizzato per il codice FF_HEAT descritto sopra. In particolare condivide con il codice FF_HEAT lo stesso COMMON memorizzato dal programma MAPPING (descritto nell'allegato [2]) le matrici che memorizzano le variabili fisiche che entrano in gioco nel processo di trasporto e l'algoritmo Path Finder per l'attraversamento delle celle da parte dei fotoni. Per quanto riguarda le variabili fisiche memorizzate nelle matrici troviamo oltre alla densità del gas, la temperatura e la potenza irraggiata della cella. Poiché la potenza irraggiata dalla cella e' esclusivamente funzione della temperatura che non cambia durante il processo Monte Carlo viene precalcolata nel codice di MAPPING.

Il processo di trasporto dei fotoni e' intrinsicamente parallelo. Sfruttando il fatto che ogni fotone fa' storia a se, esso può essere considerato come evento indipendente. Le statistiche del Monte Carlo possono essere quindi raccolte da N programmi paralleli (dove N e' il numero di processori che si ha a disposizione sulla macchina parallela).

Questa caratteristica e' stata sfruttata, come del resto avviene per tutti I codici Monte Carlo di Neutronica, e FF_HEAT, per l'implementazione parallela del codice PHOTON.

Sfruttando la libreria MPI ed il linguaggio Fortran 77 abbiamo realizzato un codice che offre una velocizzazione pressoché lineare con il numero dei processori sulla architettura parallela (la Convex SP5000).

Il paragrafo successivo descrive (in pseudo-codice) l'algoritmo parallelo implementato.

**STRUTTURA DEL CODICE MONTECARLO PER IL CALCOLO DELLA DISTRIBUZIONE DI ENERGIA MEDIANTE FOTONI**

- **Processore 0**:

***Step 1)*** legge da input il numero di fotoni da campionare (variabile *ICURR_MAXTRIALS)* e distribuisce i dati presenti nel proprio common (geometria, mappa densità in Kg, mappa di temperatura in K, mappa di energia di emissione in *Watt/microgrammo* generati dal codice Mapping) sui *NUMPROCS* processori paralleli

- **Ogni processore in parallelo**:

***Step 2)*** riceve i dati dal Processore 0 e genera una sequenza indipendente (dagli altri processori) di numeri casuali;

***Step 3)*** se la cella e' emettente (la sua temperatura e' maggiore del valore di soglia) *lancia* ***ICURR_MAXTRIALS/NUMPROCS*** *fotoni calcolando l'energia di emissione ed accumulando l'energia da loro persa nella propria mappa (lattice): esegue il seguente loop*:

***Step 3.A)*** se il numero dei campioni lanciato e' inferiore a (***ICURR_MAXTRIALS***/**NUMPROCS**) genera un nuovo fotone associando un'energia di emissione casuale (che dipende dalla energia totale radiata e dalla temperatura della cella), un'origine casuale (all'interno della cella emettente) ed un direzione di propagazione casuale; altrimenti va allo Step 4)

**Step 3.D)** *accumula, mediante il modello di assorbimento l'energia persa dal fotone (e diminuisce l'energia associata al fotone stesso) nelle celle attraversate lungo il suo cammino finche' la sua energia va al di sotto di un valore di soglia o il fotone esce dalla camera di H2 esegue i seguenti passi*:

**Step 3.D.a)** determina la cella corrente (gli indici IX,IY,IZ) attraversata dal fotone

**Step 3.D.b)** Calcola la distanza **DWALL** percorsa dal fotone nella cella (IX,IY,IZ) corrente

**Step 3.D.c)** calcola l'energia assorbita dalla cella (IX,IY,IZ) che dipende dalla densità di H2 RO, da DWALL, e dall'energia del fotone

**Step 3.D.d)** accumula l'energia assorbita dalla cella (IX,IY,IZ)

**Step 3.D.e)** decrementa l'energia residua associata al fotone

**Step 3.D.f**) Se l'energia residua del fotone e' al di sotto del valore di soglia esci dal loop e lanciare un nuovo fotone; vai allo Step 3.A)

**Step 3.D.g)** se il fotone esce dalla camera di idrogeno esci dal loop e lancia un nuovo fotone; vai allo step 3.A)

**Step 3.D.h)** ritorna allo step 3.D.a)

**Step 3.E)** fine propagazione fotone; ritorna allo Step 3.A)

- **Processore 0:**

***Step 4)*** raccoglie le mappe di energia assorbita dai vari processori e calcola la media

***Step 5)*** calcola il remapping (il processo di conversione verso il codice CFD) della mappa di energia e le memorizza su disco.

**FINE ALGORITMO MONTECARLO**

*Codice 3: Pseudo Codice del programma parallelo PHOTON*

Da notare che il rimapping dei dati dalla griglia regolare utilizzata dal codice PHOTON alla griglia non regolare utilizzata dal codice KARALIS utilizzando le informazioni calcolate dal codice Mapping (e memorrizzate nel COMMON) allo stesso modo del codice FF_HEAT.

## INPUT E OUTPUT DI UNA SIMULAZIONE

Vengono di seguito presentati alcuni risultati esemplificativi ottenuti dal codice PHOTON prendendo come input i dati ottenuti dalla simulazione a convergenza (quindi in equilibrio termodinamico) del codice di fluidodinamica KARALIS e riorganizzate nella struttura a griglia regolare dal codice MAPPING.

La Fig 11. Mostra la mappa di emissione relativa al campo di temperatura mostrato in Fig. 8. La Fig. 10 mostra invece la mappa di energia riassorbita dalle celle che costituisce l'output effettivo del programma PHOTON. Sommando queste due mappe calcoliamo gli effetti dovuti al complessivo trasferimento di energia dovuta a effetti radianti. La somma delle due precedenti mappe e della mappa di potenza ottenuta dal codice FF_HEAT costituisce la nuova mappa di input per il codice di fluidodinamica KARALIS.

*Tabella 3 Dati di output del programma PHOTON*

| | |
|---|---|
| Pressione del gas: | 6.0 atm |
| Spessore dello strato di Americio (Am242*): | 0.3E-05 m |
| Raggio del cilindro che supporta lo strato di combustibile: | 0.2 m |
| Altezza del cilindro che supporta lo strato di (Am242*): | 2.5 m |
| Risoluzione griglia (x): | 202 |
| (y): | 202 |
| (z): | 202 |
| Dimensione cella (x): | .2E-02 m |
| (y): | .2E-02 m |
| (z): | 1.25E-02 m |
| Energia depositata sulle pareti: | .50E06 Watt |
| Frazione (%) di energia depositata sulle pareti: | 18.235 |
| Frazione di celle emettenti (T > 8000 K) | 47.94 |
| Frazione (%) di fotoni che raggiungono le pareti: | 8.941 |
| Numero medio di passi di integrazioni per fotone: | 11.879 |
| Tot. potenza irraggiata: | .277E07 Watt |
| Numero di fotoni lanciati per secondo su un singolo processore: | 2241 |
| Lunghezza media del percorso del fotone: | 0.0358 m |

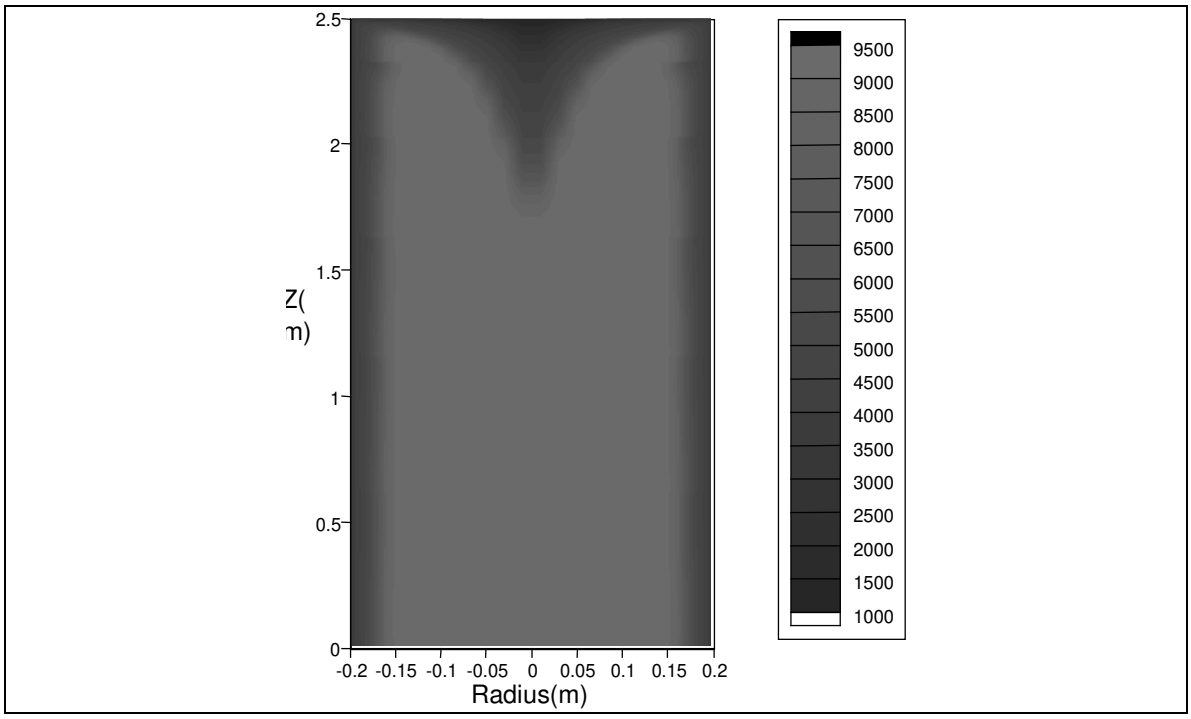


*Figura 8: Dati di input del codice PHOTON: la mappa di temperatura in K in condizioni di equilibrio termodinamico prodotta dal codice di fluidodinamica*

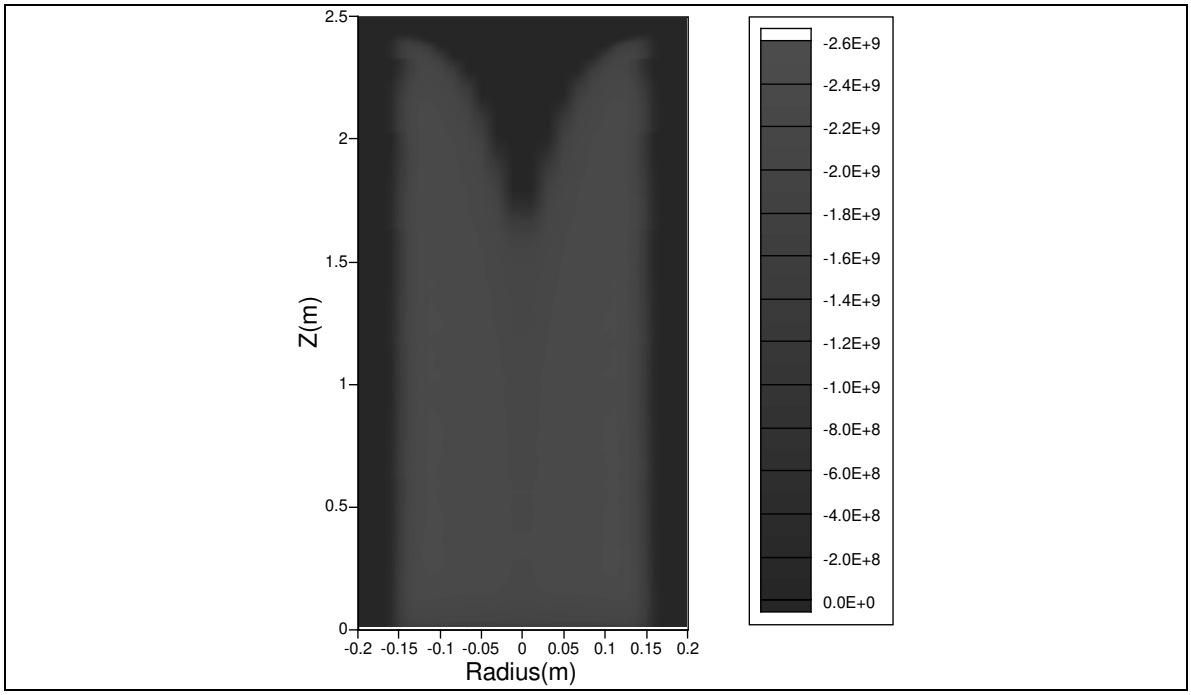


*Figura 9: Dati di output del codice PHOTON: Mappa potenza di emissione in W/Kg prodotta*

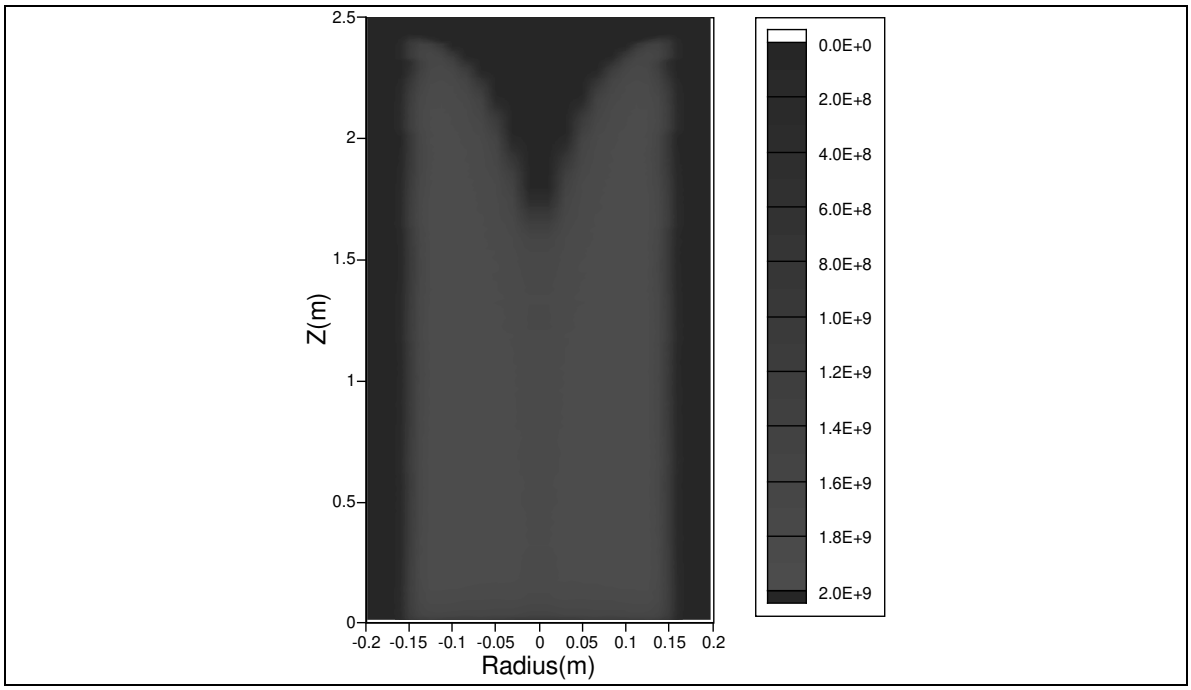


*Figura 10: Dati di Output dal codice PHOTON: mappa di potenza riassorbita dal gas in W/Kg prodotta*

## ALGORITMO PATHFINDER: ATTRAVERSAMENTO DELLA GRIGLIA REGOLARE (LATTICE)

Questa sezione illustra l'algoritmo utilizzato per il campionamento della griglia regolare nel calcolo del percorso dei frammenti di fissione e dei fotoni. Questa tecnica partiziona il volume inglobante tutta la camera di combustione in elementi di volume cubici che memorizzano i valori (in questo caso sono costanti all'interno della cella) del campo di densità, pressione e temperatura all'interno delle singole celle chiamati voxel (elemento atomico di volume: Volume Element).

Come detto sopra una fase di pre-elaborazione (Codice mapping) e' responsabile della determinazione dei valori da assegnare ai singoli voxels.

Un raggio (che come detto rappresenta un FF o un fotone) viene verificato esclusivamente con i voxel che sono vicini al cammino da esso percorso propagandosi all'interno della griglia. Poiché i voxel non si sovrappongono e dato ché il raggio impone un ordinamento globale dei voxel attraversati, l'interazione del con il materiale (il gas di idrogeno) può venire considerata in sequenza.

L'interazione FF (fotone) con la materia (gas) avviene temporalmente prima all'interno del voxel sotto esame e poi nei voxel successivamente attraversati lungo la direzione in cui si propaga il frammento di fissione (fotone). Questo fatto velocizza l'intero algoritmo. Infatti se il frammento di fissione (fotone) non ha una energia cinetica sufficiente (lo spettro di energia nel caso del fotone) ad attraversare il volume di gas rappresentato dal voxel non occorre effettuare successivi calcoli, in quanto i contributi agli altri voxel che seguono nel cammino sono ininfluenti e il calcolo termina. Se il frammento di

fissione non perde tutta l'energia cinetica la valutazione passa al voxel successivo. Questo avviene quindi fino a quando o il ff (fotone) raggiunge il valore di soglia minimo e muore o il ff (fotone) esce dalla camera di combustione ed il suo contributo va sommato all'energia che esce dalla parete.

Una volta memorizzata la struttura dati a griglia regolare è necessario disporre di un veloce strumento che determini le celle attraversate dal raggio. Per l'implementazione ci siamo rifatti ad un algoritmo ideato da Amanatides [ref]. L'algoritmo opera in due fasi: una fase di inizializzazione ed una fase di attraversamento incrementale.

La fase di inizializzazione inizia identificando il voxel che racchiude l'origine del raggio che rappresenta il frammento di fissione e inizializzando alcune variabili illustrate in tabella:

| | |
|---|---|
| $\mathbf{T_{ion}}$ | Energia cinetica del ff |
| $\mathbf{X_{orig}, Y_{orig}, Z_{orig}}$ | Coordinate dell'origine del raggio |
| $\mathbf{X_{direz}, Y_{direz}, Z_{direz}}$ | Coordinate della direzione del raggio |
| **IX, IY,IZ** | Coordinate a valori interi del voxel in esame |
| $\mathbf{X_{axis}, Y_{axis}, Z_{axis}}$ | Vettori Unitari del sistema di riferimento |
| $\mathbf{X_{step}, Y_{step}, Y_{step}}$ | Valori interi incrementali di IX ,IY e IZ ( vale -1 se la direzione e' negativa lungo l'asse; 1 viceversa ) |
| $\mathbf{T_{xmax}}$ | Valore del parametro t intersezione tra il raggio ed il piano di confine del voxel lungo la direzione $\mathbf{X_{step}}$* $\mathbf{X_{axis}}$ |
| $\mathbf{T_{ymax}}$ | Valore del parametro t intersezione tra il raggio ed il piano di confine del voxel lungo la direzione $\mathbf{Y_{step}}$* $\mathbf{Y_{axis}}$ |
| $\mathbf{T_{zmax}}$ | Valore del parametro t intersezione tra il raggio ed il piano di confine del voxel lungo la direzione $\mathbf{Z_{step}}$* $\mathbf{Z_{axis}}$ |
| **Delta_t**$_{\mathbf{x}}$ | Proiezione sull'asse x della direzione del raggio (movimento lungo $\mathbf{X_{axis}}$) |
| **Delta_t**$_{\mathbf{y}}$ | Proiezione sull'asse x della direzione del raggio (movimento lungo $\mathbf{Y_{axis}}$) |
| **Delta_t**$_{\mathbf{z}}$ | Proiezione sull'asse x della direzione del raggio (movimento lungo $\mathbf{Z_{axis}}$) |

La fase di attraversamento della griglia incrementale, descritta in pseudo-fortran, è la seguente:

```
WHILE ( T_ion .GT. valore soglia .OR. Ray_orig esce dal cilindro ) DO

CalcolaEnergiaDepositata (IX,IY, IZ)

    IF (T_xmax .LT. T_ymax) THEN

        IF ( T_xmax  .LT. T_zmax) THEN
            IX = IX + IX_step;
            T_xmax= T_xmax+ Delta_t_x ;
        ELSE
            T_zmax= T_zmax+ Delta_t_z ;
            IZ = IZ + IZ_step;
        ENDIF

    ELSE
        IF (T_ymax .LT. T_zmax) THEN
            IY = IY + IY_step;
            T_ymax= T_ymax+ Delta_t_y ;
        ELSE
            T_zmax= T_zmax+ Delta_t_z ;
            IZ = IZ + IZ_step;
        END IF
ProssimoVoxel(IX,IY, IZ);
END DO
```

**Codice Path Finder: Attraversamento incrementale della griglia regolare;**

Il calcolo dell'attraversamento richiede per ogni voxel attraversato quindi 2 confronti e 1 addizione in floating point,e 2 confronti e 1 addizione in aritmetica intera.

## RIFERIMENTI


1. C. Rubbia, *Fission Fragment Heating for Space Propulsion* 30/04/2000
2. MAPPING Code: Interfacing the Fluid Dynamics solver (KARALIS3) and FF_HEAT and PHOTON codes, Technical Report, CRS4
3. I problemi di geometrizzazione e di interfacciamento dei diversi codici per il calcolo dei parameteri del razzo. Allegato: Il codice MAPPING Allegato : GeoView : an iteractive 2D/3D viewer for EA-MC format
4. Final Report P242: Fluid dynamics.