

POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale e dell'Informazione
Corso di Laurea in Ingegneria Aeronautica



**Simulazioni di gasdinamica
molecolare su GPU: applicazioni
ed analisi prestazionali**

Relatore: Prof. Aldo FREZZOTTI

Tesi di laurea di:
Federico GENTILE
Matr. 799504

Anno Accademico 2014 - 2015

Sommario

La continua necessità di ridurre i tempi di calcolo in ambito scientifico sta comportando sempre di più l'utilizzo di schede grafiche (GPU) come strumenti per l'abbattimento dei costi computazionali; tale pregio è da attribuirsi alla possibilità di sfruttare la tecnica del parallel-computing evitando perciò il calcolo sequenziale. Usufruento di tale vantaggio tecnologico, il seguente lavoro di tesi si pone come primo obiettivo il calcolo del coefficiente di viscosità dell'ossigeno molecolare nell'ambito delle simulazioni numeriche Monte Carlo. A seguito di una breve introduzione iniziale riguardante i concetti fondamentali circa l'equazione di Boltzmann e dei gas in condizione di rarefazione, segue una descrizione del modello utilizzato per la descrizione della dinamica collisionale tra le molecole di ossigeno; successivamente viene presentato il metodo numerico atto all'inizializzazione dello stato delle molecole di ossigeno secondo una distribuzione di tipo maxwelliana. Per quanto concerne i risultati numerici circa il calcolo del coefficiente di viscosità, essi sono stati validati confrontando i valori ottenuti mediante il codice CT-DSMC con i valori sperimentali presentati in letteratura.

Nella seconda ed ultima parte della tesi, si è voluto indagare il vantaggio offerto dall'utilizzo della GPU per ridurre i tempi di calcolo per mezzo del confronto con il medesimo codice CT-DSMC in versione seriale. Lo Speed-Up ottenuto ha permesso in oltre di determinare quali siano le condizioni ideali di utilizzo della scheda grafica per ottenere il massimo delle performance a disposizione.

Parole chiave: GPU, coefficiente di viscosità, CT-DSMC, Speed-Up.

Abstract

The increasing need to reduce computational time in scientific areas of research has led to the use of graphic cards (GPU) as instruments to lower numerical effort; the main reason of such a choice is given by the possibility of taking advantage of the parallel-computing method instead of using a more common serial approach. The primary goal of this thesis is to estimate the value of viscosity of molecular oxygen in the contest of direct simulation Monte Carlo by using the parallel technique previously mentioned. After a brief introduction about the basic concepts of the Boltzmann equation and the rarefaction condition of dilute gases, a description on the mathematical model for the simulation of collisional dynamics is given; later on, the numerical method for the initialization of the molecule velocities and orientations according to a Maxwellian distribution is presented. Lastly, the numerical results representing the estimated viscosity coefficient by using a CT-DSMC program have been compared with experimental data offered by the bibliography.

In the second part of the present work, we focused our attention on the advantages offered by the use of GPUs in terms of time reduction; more precisely a comparison between the calculation time of the CT-DSMC program in its parallel version and the serial one has been conducted. The Speed-Up ratio obtained from this analysis allowed to determine the ideal conditions of operation of the graphic card in order to get the best performances.

Key words: GPU, viscosity coefficient, CT-DSMC, Speed-Up.

Indice

Elenco delle figure	7
Elenco delle tabelle	9
1 Introduzione	11
1.1 Modello molecolare dei gas e condizioni di applicabilità	11
1.2 Gas semplice diluito	12
1.3 Equazione di Boltzmann per un gas monoatomico o poliatomico	13
1.3.1 Modelli molecolari per gas monoatomici e poliatomici	13
1.3.2 Il modello BGK	15
1.3.3 Il modello fenomenologico di Borgnakke-Larsen	16
1.4 Calcolo delle quantità macroscopiche del gas	17
1.4.1 Stima della densità, quantità di moto ed energia	17
1.4.2 Tensore degli sforzi $\underline{\underline{P}}$	18
2 Potenziale di interazione molecolare	21
2.1 Potenziale di interazione in coordinate Jacobiane	21
2.2 Espressione analitica del potenziale di interazione	23
2.3 Parametrizzazione dei termini radiali	24
2.4 Espressioni delle configurazioni di riferimento del potenziale molecolare	26
3 Distribuzione di Equilibrio per un set di molecole di ossigeno	29
3.1 Distribuzione di Maxwell	29
3.2 Implementazione della subroutine per la generazione della distribuzione di equilibrio	31
3.3 Confronto tra i risultati numerici ed analitici	32
4 Struttura e validazione del codice CT-DSMC per il calcolo della viscosità dell'ossigeno molecolare	37
4.1 Descrizione del codice DSMC	37
4.2 Uniform shear flow	39
4.3 Calcolo delle viscosità	40

4.4	Validazione del codice	42
5	Prova di Speed-Up del codice CT-DSMC e vantaggi dell'utilizzo della GPU	45
5.1	Linguaggio di programmazione delle schede grafiche (GPU)	45
5.2	CUDA (Compute Unified Device Architecture)	46
5.2.1	Kernel	47
5.2.2	Threads	48
5.2.3	Streaming Multiprocessors (SM)	48
5.2.4	Tipologie di memoria	48
5.3	Proprietà della scheda grafica GeForce GTX 780	50
5.4	Descrizione del programma-test per la stima dei tempi di esecuzione	51
5.4.1	Collisioni con time-step fisso e con adattività di passo	52
5.5	Tentativi di ottimizzazione e riduzione dei tempi di calcolo	53
5.6	Analisi dei tempi di calcolo	53
5.7	Interpretazione dei risultati	59
6	Conclusioni	61
6.1	Sviluppi futuri	62
	Bibliografia	63

Elenco delle figure

2.1	Orientazione degli angoli θ_a , θ_b e ϕ	22
2.2	Configurazioni di riferimento L , H , X e T	23
2.3	Parametri del potenziale di interazione molecolare	24
2.4	Potenziale in configurazione $X T L H$	27
3.1	Disposizione della velocità angolare ω^*	32
3.2	Istogrammi delle componenti x , y e z della velocità traslazionale.	33
3.3	Istogramma dell'energia interna.	35
3.4	Istogramma dell'energia interna in scala logaritmica.	36
4.1	Andamento dello uniform shear flow.	39
4.2	$N_p = 100000$, $\alpha = 0.001$, $T_{ref} = 300K$, $P_{ref} = 10^5 Pa$	42
4.3	Viscosità O_2 al variare della temperatura.	43
5.1	In <i>arancione/giallo</i> regione dedicata alla memoria; in <i>verde</i> regione dedicata al processamento dei dati.	46
5.2	Distribuzione di threads e blocchi in 2D.	47
5.3	Esempio di CUDA kernel.	47
5.4	Formula per l'identificazione dei thread.	48
5.5	Workflow del programma-test per l'analisi dei tempi di esecuzione.	52
5.6	Tempo di esecuzione del singolo thread.	54
5.7	Tempo di calcolo totale del kernel.	56
5.8	Confronto tra i tempi di esecuzione della CPU e della GPU.	58

Elenco delle tabelle

3.1	Parametri di riferimento	30
4.1	Pressione e tempo di riferimento	41
4.2	Confronto tra valori numerici CT-DSMC e valori sperimentali. . .	43
5.1	Tempi di esecuzione dei threads con passo fisso.	55
5.2	Tempi di esecuzione dei threads con passo adattivo.	55
5.3	Tempi medio della singola collisione via CPU.	57
5.4	Valori di Speed-Up sia con passo fisso che con passo variabile. . .	58

Capitolo 1

Introduzione

Lo studio di un gas in condizioni di non equilibrio è reso possibile attraverso la risoluzione dell'equazione di Boltzmann, la quale permette di descrivere la distribuzione statistica delle molecole in un fluido. Tra le principali applicazioni che coinvolgono l'uso di tale equazione vi è la descrizione del comportamento di un gas qualora il libero cammino medio non sia trascurabile rispetto alla lunghezza caratteristica della geometria del flusso; un ulteriore possibile utilizzo di questo approccio statistico, consiste nel dedurre il comportamento macroscopico di un gas a partire dalle sue proprietà microscopiche nel caso in cui il libero cammino medio sia molto minore rispetto alla lunghezza caratteristica del problema. Nel seguente capitolo si vogliono introdurre i concetti basilari per la comprensione dell'equazione di Boltzmann e la modalità con la quale quest'ultima verrà utilizzata per il calcolo della viscosità dell'ossigeno molecolare.

1.1 Modello molecolare dei gas e condizioni di applicabilità

Il modello matematico che permette di descrivere un fluido come un mezzo continuo è rappresentato dalle equazioni di Navier-Stokes; questa descrizione di tipo macroscopica permette di ottenere informazioni circa la velocità, densità, pressione e temperatura del fluido preso in considerazione al variare del tempo e delle coordinate spaziali. L'applicabilità di questo modello è tuttavia limitata qualora il gradiente delle variabili macroscopiche sia dello stesso ordine di grandezza del libero cammino medio delle molecole.

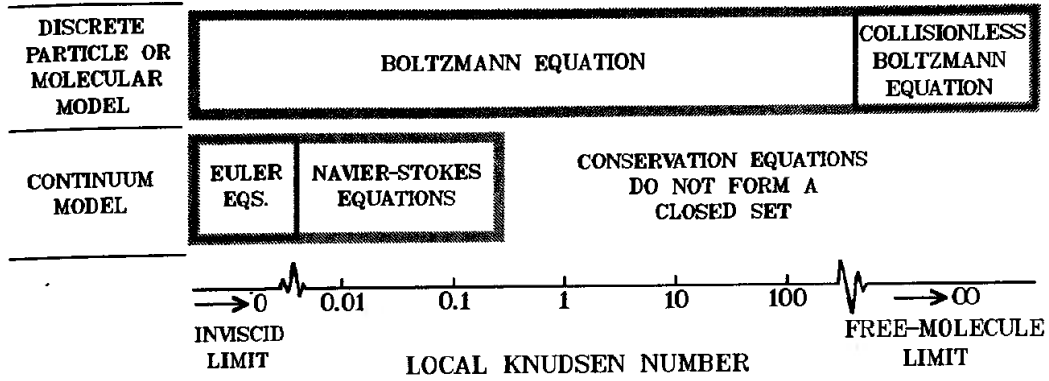
Per quanto riguarda invece il modello molecolare (oppure microscopico), il gas è inteso come un insieme di molecole distinte dotate ad ogni istante temporale di una propria posizione, velocità e stato. Per quanto riguarda l'aspetto matematico, l'equazione di riferimento dell'approccio microscopico è l'equazione di Boltzmann.

Dal punto di vista quantitativo, il parametro che permette di stabilire il grado

di applicabilità della rappresentazione continua di un fluido è dato dal *numero di Knudsen* (Kn); esso è definito come il rapporto tra il libero cammino medio¹ λ e la lunghezza caratteristica del flusso L :

$$Kn = \frac{\lambda}{L} \quad (1.1)$$

L'interpretazione fisica del coefficiente appena descritto consiste nel fornire il grado di rarefazione del gas. Qualora $Kn > 0.1$, il set di equazioni di Navier-Stokes non è più uno strumento adatto per la rappresentazione corretta della dinamica dei gas. A titolo esemplificativo, viene ora proposto un grafico riassuntivo che descrive i limiti dei vari modelli matematici per la rappresentazione dei fluidi al variare del numero di Knudsen:



1.2 Gas semplice diluito

Lo studio del comportamento della viscosità dell'ossigeno molecolare è stato effettuato sotto l'ipotesi di gas semplice e diluito; per capire meglio che cosa si intende con i termini *semplice* e *diluito* consideriamo un volume V contenente un gas formato da N molecole appartenenti alla stessa specie chimica ed aventi la stessa struttura (*gas semplice*); definiamo come densità numerica $n = \frac{N}{V}$ il numero di molecole per unità di volume; è possibile stimare l'ordine di grandezza della distanza che separa due molecole mediante la seguente relazione:

$$\delta = \left(\frac{1}{n}\right)^{\frac{1}{3}} \quad (1.2)$$

Introduciamo ora il diametro effettivo molecolare σ , ossia la distanza per la quale le forze di interazione molecolare sono non trascurabili; si può quindi definire

¹Distanza percorsa dalle molecole tra due collisioni successive.

1.3 Equazione di Boltzmann per un gas monoatomico o poliatomico 13

come *diluito* un gas tale per cui:

$$\frac{\delta}{\sigma} = \left(\frac{1}{n\sigma^3} \right)^{\frac{1}{3}} \gg 1 \quad (1.3)$$

Sotto queste condizioni, solamente una piccola parte del volume spaziale è occupata dalle molecole; di conseguenza esse trascorrono gran parte del loro moto senza cadere nel dominio di influenza delle particelle circostanti. Per quanto riguarda le eventuali collisioni, esse sono da considerarsi binarie ovvero coinvolgono solamente due molecole alla volta. È importante infine ricordare che nel caso di gas diluito, la durata dello scontro tra due molecole è da considerarsi infinitesimale, tanto da definire le collisioni come istantanee.

1.3 Equazione di Boltzmann per un gas monoatomico o poliatomico

Lo studio dell'equazione di Boltzmann è assai complicato a causa del termine di collisione Q ; esso rappresenta la variazione netta delle proprietà cinetiche delle molecole dovuta agli urti. L'espressione di questa quantità avviene per mezzo di modelli molecolari i quali simulano la legge di interazione tra le molecole che partecipano ad un urto. Tale procedimento è generalmente associato a gas formati da molecole monoatomiche; il motivo di tale scelta è da attribuirsi alla maggiore semplicità della trattazione matematica alla base del problema. Qualora si abbia a che fare con sistemi di molecole poliatomiche, l'espressione del termine di collisione diviene assai più complicata e si necessita di modelli ancora più articolati. Viene ora proposta una panoramica relativa ai vari modelli utilizzati per l'espressione della quantità Q sia nel caso di gas monoatomico e poliatomico. Infine viene introdotto il modello di Borgnakke-Larsen il quale si occupa di gestire il termine di collisione nell'ambito delle simulazione DSMC qualora le molecole del gas posseggano gradi di libertà interni.

1.3.1 Modelli molecolari per gas monoatomici e poliatomici

Uno degli obiettivi principali della teoria cinetica classica dei gas consiste nello stimarne le proprietà macroscopiche: pressione, temperatura, conducibilità termica, viscosità etc. Queste ultime devono tuttavia essere calcolate a partire dalle quantità microscopiche associate alle molecole che compongono il gas ossia la massa, la velocità, l'energia cinetica, le forze di interazione molecolare, i gradi di libertà interni etc.

L'equazione che sta alla base della teoria cinetica è l'equazione di Boltzmann; essa può assumere diverse forme in base al tipo di gas considerato. Nel caso di

un set di molecole monoatomiche in assenza di forzanti esterne, l'equazione che governa la dinamica del gas assume la seguente forma:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \frac{\partial f}{\partial \mathbf{x}} = Q(f, f) \quad (1.4)$$

dove f è la funzione di distribuzione associata ad una molecola che abbia velocità compresa tra $[\boldsymbol{\xi}, \boldsymbol{\xi} + d\boldsymbol{\xi}]$ e posizione tra $[\mathbf{x}, \mathbf{x} + d\mathbf{x}]$, $\boldsymbol{\xi}$ è il vettore velocità associato alle molecole, \mathbf{x} è il vettore associato alle coordinate spaziali e $Q(f, f)$ è il termine collisionale.

Nel caso di molecole monoatomiche, la f dipende esclusivamente dalla posizione, dalla velocità e dal tempo: $f = f(\mathbf{x}, \boldsymbol{\xi}, t)$. Risolvere l'equazione di Boltzmann per il calcolo delle quantità di trasporto del gas si traduce dal punto di vista pratico nel modellare opportunamente il termine collisionale; questa operazione risulta essere analiticamente gestibile qualora si abbia a che fare con un gas di tipo monoatomico. In tal caso infatti, un modello largamente utilizzato per la sua semplicità è il modello a sfere rigide. Secondo questa rappresentazione, le molecole che costituiscono il gas sono idealizzate come masse puntiformi aventi un raggio σ il quale determina la distanza minima per la quale due molecole interagiscono. In aggiunta, le collisioni sono puramente elastiche. A titolo esemplificativo, se si considerano le ipotesi appena elencate, il termine collisionale in presenza di una forza centrale assume la seguente espressione:

$$Q(f, f) = \frac{1}{m} \int (f' f'_* - f f_*) B(\theta, V) d\theta d\epsilon d\boldsymbol{\xi}_*, \quad (1.5)$$

dove termini V , n , θ , r ed ϵ sono rispettivamente la velocità relativa tra le due particelle che partecipano alla collisione, la normale diretta come la linea degli apsidi dell'orbita della molecola incidente rispetto a quella obiettivo, l'angolo compreso tra la velocità \mathbf{V} e la normale \mathbf{n} della sfera, e le coordinate polari definite sul piano ortogonale al vettore \mathbf{V} . Si intuisce facilmente che la trattazione matematica necessaria per calcolare le quantità di trasporto del gas sono piuttosto complicate. Questa complicazione è tanto più accentuata nel caso in cui si vogliono studiare le proprietà di un gas poliatomico; in tal caso infatti, il problema si accentua notevolmente poiché la funzione di distribuzione f risulta essere dipendente anche dai gradi di libertà interni; nel caso di una molecola rigida libera di traslare e ruotare nello spazio tridimensionale si individuano infatti 6 gradi di libertà: $f = f(\mathbf{x}, \boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\omega}, t)^2$. Ne consegue che l'equazione di Boltzmann assume un'espressione assai più complicata in cui si nota la comparsa di nuovi termini.

Anche in questo caso è necessario usare dei modelli appositamente costruiti per esprimere il termine $Q(f, f)$. Tra le varie possibilità [2], vengono presentati brevemente due possibili approcci che modellano in maniera opportuna il termine collisionale:

² $\boldsymbol{\theta}$ è l'orientazione della molecola, $\boldsymbol{\omega}$ è il vettore velocità angolare.

1.3 Equazione di Boltzmann per un gas monoatomico o poliatomico¹⁵

- *Rough Spheres*: il modello molecolare in questione per la descrizione di gas poliatomici è stato introdotto da G. Bryan; i coefficienti di trasporto sono stati calcolati successivamente da Pidduck. Secondo questo modello, le molecole sono rappresentate da sfere che interagiscono tra loro secondo un potenziale analogo al modello a sfere rigide; a differenza dell'approccio precedente, la sfera è considerata ruvida: questa caratteristica implica che durante una collisione elastica tra due corpi, non si verifichi dello scorrimento bensì un'adesione tra le superfici a contatto delle due sfere; in tal modo si ha una conversione totale dell'energia di deformazione in energia rotazionale e traslazionale. Come conseguenza di questa ipotesi, la velocità relativa delle sfere nel punto di impatto è invertita. In aggiunta, l'energia interna è caratterizzata esclusivamente dall'energia rotazionale ed il principale vantaggio di questo modello è che non c'è bisogno di specificare l'orientazione delle molecole.
- *Sphero-Cylindrical molecules*: Curtiss e Muckenfuss sono riuscite ad ottenere delle formule analitiche per calcolare i coefficienti di trasporto del gas modellando la singola molecola per mezzo di una forma detta sfero-cilindrica: essa consiste semplicemente nell'assimilare la molecola ad un cilindro avente due calotte semisferiche alle estremità. Le formule ottenute dai due autori hanno lo svantaggio di avere una forma estremamente complicata per poter essere riportate nonostante siano costituite da funzioni elementari.

Ulteriori possibili rappresentazioni sono il *Loaded Spheres* di Dahler e *Nearly Smooth Molecules* di Eucken.

1.3.2 Il modello BGK

Come ripetuto più volte, l'inconveniente maggiore associato all'equazione di Boltzmann è rappresentato dalla complicata forma assunta dal termine collisionale. Non sorprende perciò che siano state proposte espressioni semplificate di $Q(f, f)$ per agevolare la trattazione matematica [4]. Un possibile approccio che permetta di raggiungere tale scopo consiste nel supporre che una descrizione eccessivamente dettagliata della collisione tra due molecole sia superflua; l'operatore di collisione $Q(f, f)$ può quindi essere sostituito da un operatore semplificato $J(f, f)$ il quale contiene solamente indicazioni qualitative e valori medi circa il vero operatore di collisione. Il metodo più utilizzato appartenente a questa categoria di modelli collisionali è stato introdotto da Bhatnagar, Gross e Krook (BGK). Alla base del modello BGK si richiede che le caratteristiche principali dell'operatore collisionale siano le seguenti:

- La tendenza da parte del gas di raggiungere una distribuzione Maxwelliana:

$$\int \ln(f)Q(f, f) d\xi = 0 \quad e \quad \int \ln(f)J(f) d\xi = 0 \quad (1.6)$$

- Per qualsiasi invariante collisionale $\psi = [\rho, \rho\xi, \rho\xi^2]$, $Q(f, f)$ e $J(f, f)$ devono soddisfare la seguente condizione:

$$\int \psi Q(f, f) d\xi = 0 \quad e \quad \int \psi J(f) d\xi = 0 \quad (1.7)$$

Un possibile modo per rispettare le condizioni appena introdotte consiste nell'ipotizzare che gli effetti dovuti alle collisioni perturbino mediamente la funzione f di una quantità proporzionale allo scostamento che essa ha rispetto alla distribuzione Maxwelliana ϕ . Introducendo infine la frequenza di collisione ν , il modello collisionale è:

$$J(f, f) = \nu[\phi - f] \quad (1.8)$$

Quanto descritto è riferito ad un gas monoatomico ma può essere esteso anche al caso di molecole poliatomiche. L'utilizzo del modello BGK è vantaggioso in quanto permette di dedurre delle equazioni integrali per il calcolo delle variabili macroscopiche come ad esempio ρ, ξ, T ; le relazioni ottenute sono fortemente non lineari ma permettono comunque di semplificare notevolmente i conti consentendo lo studio di alcuni problemi di interesse per mezzo del calcolatore.

1.3.3 Il modello fenomenologico di Borgnakke-Larsen

Con l'avvento dei computer e lo sviluppo del metodo Monte Carlo, si è reso possibile un approccio di tipo numerico per quanto concerne la risoluzione dell'equazione di Boltzmann. Il metodo in questione prende il nome di Direct Simulation Monte Carlo (DSMC) ed è accostato ai modelli molecolari presentati in precedenza. I modelli tradizionali tuttavia risultano essere ragionevoli solamente in presenza di gas monoatomici. Qualora si abbia a che fare con gas poliatomici è necessario utilizzare modelli molecolari piuttosto complessi; di conseguenza la DSMC viene accostata al metodo introdotto da Larsen-Borgnakke; questo modello aggira le restrizioni e le complicazioni dei modelli meccanici (rough spheres, Sphero-Cylindrical molecules, loaded spheres etc) in cui si cercava di mantenere un accoppiamento tra l'energia traslazionale e quella rotazionale. I gradi di libertà interni j della molecola poliatomica sono contenuti all'interno dell'espressione dell'energia interna ϵ e quindi non compaiono nella trattazione ($f(\mathbf{x}, \xi, \epsilon, t)$). Consideriamo ora una generica collisione: la probabilità che essa sia di tipo inelastica è z ; in tal caso l'energia totale viene suddivisa in maniera casuale tra la componente traslazionale e quella rotazionale del moto; questo processo è il risultato del campionamento di una frazione E_{tr}/E dell'energia traslazionale per mezzo di

una funzione di densità di probabilità $P_1(E_{tr}/E|j)$; l'energia rotazionale rimanente E_{rot} , calcolabile come $E_{rot} = \epsilon_a + \epsilon_b = E - E_{tr}$, è distribuita casualmente tra le molecole partecipanti alla collisione campionando la frazione ϵ/E_{rot} da un'opportuna *p.d.f* $P_2(\epsilon/E_{rot}|j)$. Le collisioni elastiche aventi probabilità $1 - z$ sono caratterizzate dal fatto che l'energia rotazionale rimane inalterata durante l'urto. La dinamica dell'impatto è gestita mediante modelli di molecole monoatomiche come hard sphere, variable hard sphere o variable soft sphere.

Il modello di Borgnakke-Larsen risulta estremamente efficiente dal punto di vista computazionale, ma ignora i dettagli dinamici della collisione tra le due molecole; inoltre la presenza di alcuni parametri come z permette di regolare il grado di accoppiamento roto-traslazionale rendendo possibile una corrispondenza tra i risultati numerici e quelli sperimentali.

1.4 Calcolo delle quantità macroscopiche del gas

Come specificato all'inizio di questo capitolo, il primo obiettivo di questa tesi è quello di studiare i valori assunti dalla viscosità dell'ossigeno molecolare. Per poter affrontare questo problema, è necessario capire il legame tra la descrizione microscopica e macroscopica del gas preso in considerazione. A tal proposito, vengono inizialmente proposte le procedure per la stima della densità, quantità di moto ed energia; successivamente si è scelto di impostare il procedimento per il calcolo del tensore degli sforzi relativi al gas di Boltzmann per un set di molecole monoatomiche.

1.4.1 Stima della densità, quantità di moto ed energia

A partire dall'integrale della distribuzione f della singola particella rispetto alla variabile velocità, si ottiene la densità del gas:

$$\rho(\mathbf{x}, t) = \int f d\xi \quad (1.9)$$

essa può essere interpretata come la massa attesa per unità di volume nel punto (\mathbf{x}, t) oppure come il prodotto tra la massa molecolare m e la funzione densità di probabilità P di trovare una molecola nel punto (\mathbf{x}, t) ; ciò ci permette di introdurre una nuova quantità che prende il nome di densità di numero n :

$$n = \frac{\rho(\mathbf{x}, t)}{m} = \int P d\xi \quad (1.10)$$

Per quanto riguarda la velocità media del set di particelle si utilizza invece l'operazione di media effettuata sulla velocità di tutte le molecole:

$$\mathbf{u}(\mathbf{x}, t) = \frac{\int \xi f d\xi}{\int f d\xi} \quad (1.11)$$

Si nota che la velocità media \mathbf{u} è una quantità ottenibile a partire da osservazioni macroscopiche. Un'ulteriore quantità che può essere attribuita a ciascuna molecola è la velocità peculiare \mathbf{c} ; essa è definita come la differenza tra la velocità assunta dalla particella i -esima e la velocità media del gas: $\mathbf{c} = \boldsymbol{\xi} - \mathbf{u}$. Una proprietà importante che ottenibile dalla variabile appena definita è la seguente:

$$\int \mathbf{c} f \, d\boldsymbol{\xi} = \int \boldsymbol{\xi} f \, d\boldsymbol{\xi} - \mathbf{u} \int f \, d\boldsymbol{\xi} = \rho \mathbf{u} - \rho \mathbf{u} = 0 \quad (1.12)$$

La proprietà appena descritta verrà ripresa inseguito per giustificare alcuni passaggi matematici durante il calcolo dell'energia.

Osserviamo ora che combinando la definizione di densità e di velocità media del flusso, si può rielaborare l'equazione 1.11 in maniera tale da ricavare l'espressione della quantità di moto per unità di volume:

$$\rho \mathbf{u} = \int \boldsymbol{\xi} f \, d\boldsymbol{\xi} \quad (1.13)$$

Possiamo infine introdurre la definizione di energia per unità di volume:

$$\frac{1}{2} \int \|\boldsymbol{\xi}\|^2 f \, d\boldsymbol{\xi} = \frac{1}{2} \rho \|\mathbf{u}\|^2 + \frac{1}{2} \int \|\mathbf{c}\|^2 f \, d\boldsymbol{\xi} \quad (1.14)$$

dove la relazione 1.12 è stata sfruttata per eliminare il termine misto $\int \mathbf{u} \cdot \mathbf{c} f \, d\boldsymbol{\xi}$ generato dalla sostituzione di $\boldsymbol{\xi}$ con $\mathbf{u} + \mathbf{c}$; i due termini che compongono l'energia sono rispettivamente l'energia cinetica e l'energia interna del gas.

1.4.2 Tensore degli sforzi $\underline{\underline{P}}$

Prendiamo in considerazione l'equazione di Boltzmann:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}} = Q(f, f) \quad (1.15)$$

Quest'ultima può essere moltiplicata per l'invariante di collisione elementare $\boldsymbol{\xi}$ ed integrata nello spazio delle velocità:

$$\int \boldsymbol{\xi} \frac{\partial f}{\partial t} \, d\boldsymbol{\xi} + \int \boldsymbol{\xi} (\boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}}) \, d\boldsymbol{\xi} = \int \boldsymbol{\xi} Q(f, f) \, d\boldsymbol{\xi} \quad (1.16)$$

utilizzando la proprietà 1.7 presentata nel paragrafo precedente, si deduce subito che il r.h.s dell'equazione 1.16 è nullo; richiamando poi le definizioni di quantità di moto per unità di volume e sostituendo $\boldsymbol{\xi}$ con $\mathbf{c} + \mathbf{u}$, si ottiene la legge di conservazione della quantità di moto:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \int (\mathbf{c} + \mathbf{u})(\mathbf{c} + \mathbf{u}) \cdot \frac{\partial f}{\partial \mathbf{x}} \, d\boldsymbol{\xi} = 0 \quad (1.17)$$

Consideriamo ora il secondo termine dell'equazione; tramite opportuni passaggi matematici si dimostra che:

$$\int (\mathbf{c} + \mathbf{u})(\mathbf{c} + \mathbf{u}) \cdot \frac{\partial f}{\partial \mathbf{x}} d\xi = \frac{\partial [\rho \mathbf{u} \otimes \mathbf{u} + \int \mathbf{c} \otimes \mathbf{c} f d\xi]}{\partial \mathbf{x}} \quad (1.18)$$

Si osserva che il membro di destra dell'equazione è formato da due contributi: il primo termine rappresenta il flusso di quantità di moto di tipo macroscopico; il secondo contributo invece è il flusso di quantità di moto associato al moto casuale delle molecole. Quest'ultimo è associato al tensore degli sforzi $\underline{\underline{P}}$ ossia:

$$p_{ij} = \int c_i c_j f d\xi \quad (1.19)$$

La relazione 1.19 sarà di fondamentale importanza durante il resto della trattazione poiché permetterà di risalire al valore del coefficiente di viscosità.

Capitolo 2

Potenziale di interazione molecolare

Nel seguente capitolo viene presentato il modello matematico che permette di simulare il potenziale di interazione tra due molecole di ossigeno. L'articolo scientifico di riferimento nel quale viene proposto tale modello è *Molecular Beam Scattering of Aligned Oxygen Molecules. The Nature of the Bond in the $O_2 - O_2$ Dimer* [1].

L'espressione analitica del potenziale molecolare dipende dalle seguenti variabili:

- Posizione relativa delle molecole
- Orientazione nello spazio
- Interazione dello spin tra le due molecole

Nonostante nell'articolo di riferimento venga proposto una versione del potenziale di interazione molecolare che coinvolge tutti e tre i fattori appena elencati, si è optato per una versione semplificata che tenga conto solamente della posizione ed orientazione delle molecole; tale scelta è stata fatta con lo scopo di alleggerire la trattazione matematica. Infine si è proceduto ad uno studio del comportamento del potenziale molecolare al variare dei valori nominali dei coefficienti definiti all'interno della sua espressione analitica; i risultati raggiunti verranno utilizzati nel capitolo successivo come strumento per l'interpretazione dei dati ottenuti dalla simulazione CT-DSMC per il calcolo della viscosità dell'ossigeno.

2.1 Potenziale di interazione in coordinate Jacobiane

La geometria del sistema formato da due molecole di ossigeno (dimero $O_2 - O_2$) e il relativo potenziale di interazione sono riferiti ad un sistema di coordinate Jacobiane. Dal punto di vista grafico, la disposizione nello spazio delle due particelle è riportata in figura 2.1

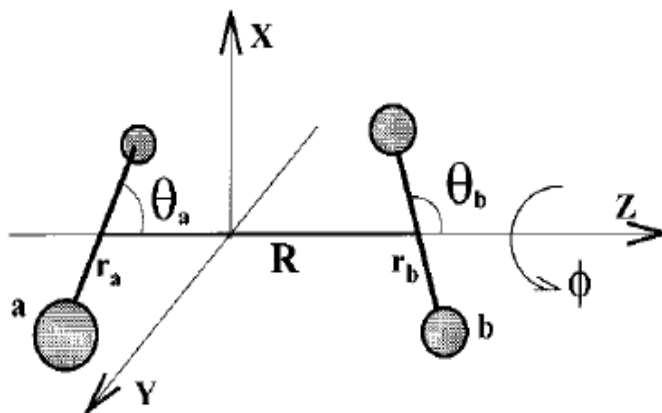


Figura 2.1: Orientazione degli angoli θ_a , θ_b e ϕ

Come si evince dall'immagine, le molecole di ossigeno possono essere assimilate a dei *manubri* dotati in generale di sei gradi di libertà: tre traslazionali e tre rotazionali; tuttavia osservando che l'inerzia lungo l'asse che collega i due atomi di ossigeno è trascurabile, si può eliminare dal conteggio dei *g.d.l.*¹ la rotazione ad essa associata.

Per quanto riguarda le variabili utilizzate, si indica con \mathbf{R}_{ab} il vettore che collega il centro di massa della molecola a con quello della molecola b . I vettori che invece legano due atomi della stessa molecola sono rispettivamente \mathbf{r}_a e \mathbf{r}_b ; quest'ultimi permettono anche di definire l'orientazione nello spazio. Infatti, ipotizzando di fissare un sistema di riferimento avente l'asse z diretta come \mathbf{R} , siamo in grado di definire il set di coordinate da cui dipende il potenziale di interazione:

- $R_{ab} = \|\mathbf{R}_{ab}\|$: distanza che separa le molecole $a-b$
- θ_i : angolo compreso tra il vettore \mathbf{R}_{ab} e \mathbf{r}_i (con $i = a, b$)
- $\phi = \phi_a - \phi_b$: angolo formato dall'intersezione di due piani (Π_a e Π_b) contenenti ciascuno l'asse della molecola i -esima ed il vettore \mathbf{R} ; l'angolo ϕ può essere anche inteso come la differenza tra ϕ_a e ϕ_b : in tal caso ϕ_i è l'angolo formato tra ognuno dei due piani con un piano di riferimento arbitrario; l'arbitrarietà di quest'ultimo è dovuto al fatto che il potenziale non dipende dall'orientazione assoluta nello spazio ma da quella relativa tra le molecole.

¹*g.d.l.*: gradi di libertà

2.2 Espressione analitica del potenziale di interazione

L'espressione che descrive il potenziale di interazione tra due molecole di ossigeno in assenza di spin è dato dalla seguente espansione troncata di armoniche sferiche:

$$V(R, \theta_a, \theta_b, \phi) = 4\pi \sum_{L_a, L_b, L} V^{L_a L_b L}(R_{ab}) Y_{L_a L_b}^{L_0}(\theta_a, \theta_b, \phi) \quad (2.1)$$

in cui sono stati messi in evidenza il contributo radiale e quello angolare; per quanto riguarda i vari coefficienti presenti 2.1, si ha che $L_a, L_b = 0, 1, 2, \dots$ e $|L_a - L_b| \leq L \leq L_a + L_b$. Le funzioni angolari $Y_{L_a L_b}^{L_0}$ sono invece armoniche sferiche bipolari:

$$Y_{L_a L_b}^{L_0}(\theta_a, \theta_b, \phi) = (2L + 1)^{\frac{1}{2}} \sum_m (-)^{L_a - L_b} \begin{pmatrix} L_a & L_b & L \\ m & -m & 0 \end{pmatrix} \cdot Y_{L_a m}(\theta_a, \phi_a) Y_{L_b - m}(\theta_b, \phi_b) \quad (2.2)$$

Dove $Y_{L_a m}$ e $Y_{L_b - m}$ sono armoniche sferiche, $(:::)$ è il simbolo $3 - j$ di Wigner, mentre $-\min(L_a, L_b) \leq m \leq \min(L_a, L_b)$.

Per quanto riguarda invece il termine radiale $V^{L_a L_b L}(R_{ab})$, esso è formato in generale da diversi tipologie di contributi: elettrostatico, dispersione, induzione, accoppiamento spin-spin etc. Si deduce quindi che l'equazione 2.1 è un'espansione completa che tiene conto di più fattori; gli autori dell'articolo [1] hanno tuttavia preferito troncare l'espansione al quarto termine in modo tale da avere 4 configurazioni base di riferimento:

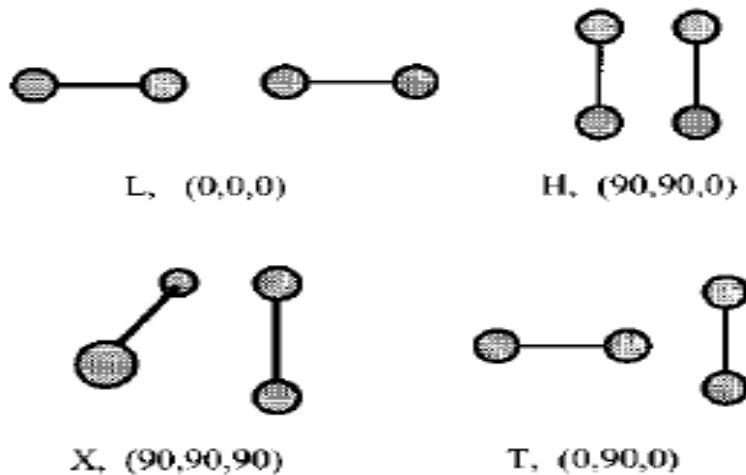


Figura 2.2: Configurazioni di riferimento L , H , X e T

A valle di quanto detto, qualsiasi configurazione potrà essere ottenuta combinando opportunamente le configurazioni di riferimento L , H , X e T .

Il modello di interazione molecolare assume perciò la seguente forma:

$$V(R, \theta_a, \theta b, \phi) = 4\pi \{ V^{000}(R_{ab}) + V^{202}(R_{ab}) [Y_{20}^{20}(\theta_a, \theta b, \phi) + Y_{02}^{20}(\theta_a, \theta b, \phi)] + V^{220}(R_{ab}) Y_{22}^{00}(\theta_a, \theta b, \phi) + V^{222}(R_{ab}) Y_{22}^{20}(\theta_a, \theta b, \phi) \} \quad (2.3)$$

dove il primo termine dell'espansione $V^{000}(R_{ab})$ rappresenta la componente isotropa dell'interazione, mentre $V^{202}(R_{ab})$, $V^{220}(R_{ab})$ e $V^{222}(R_{ab})$ descrivono il contributo anisotropo complessivo.

2.3 Parametrizzazione dei termini radiali

I termini radiali $V^{xyz}(R_{ab})$ contenuti all'interno dell'equazione 2.3 sono stati parametrizzati opportunamente per mezzo delle costanti riportate in figura:

V^{000}	
ϵ (meV) ^b	11.7 (± 0.6)
R_m (Å)	3.90 (± 0.08)
β	6.7
C^{000} (meV Å ⁶)	$5.3 (\pm 0.5) \times 10^4$
x_1	1.12
x_2	1.55
V^{202}	
A^{202} (meV)	$7.7 (\pm 1.5) \times 10^6$
c^{202} (Å ⁻¹)	4.0 (2%)
d^{202} (Å ⁻²)	-0.060 (2%)
C^{202} (meV Å ⁶)	6.12×10^3
V^{220}	
A^{220} (meV)	$5.0 (\pm 1.5) \times 10^6$
c^{220} (Å ⁻¹)	3.83 (2%)
V^{222}	
A^{222} (meV)	$-4.0 (\pm 1.5) \times 10^6$
c^{222} (Å ⁻¹)	3.83 (2%)

^a Estimated uncertainties are in parentheses. ^b 1 meV = 9.648×10^{-2} kJ·mol⁻¹.

Figura 2.3: Parametri del potenziale di interazione molecolare

Particolare importanza ricopre il trattamento del termine isotropo $V^{000}(R)$ il quale è stato parametrizzato attraverso un metodo denominato *Morse-spline-van Der Walls*; quest'ultimo permette di modellare in modo appropriato la forza intermolecolare in funzione della distanza. Adimensionalizziamo la distanza R

attraverso una lunghezza di riferimento pari a R_m tale che:

$$x = \frac{R}{R_m} \quad (2.4)$$

e definiamo la seguente funzione:

$$f(x) = \frac{V^{000}(R)}{\epsilon} \quad (2.5)$$

dove ϵ è il valore assunto dalla buca di potenziale. Poniamo ora

- Contributo *Morse*: $x \leq x_1$

$$f(x) = e^{-2\beta(x-1)} - 2e^{-\beta(x-1)} \quad (2.6)$$

- Contributo *Spline*: $x_1 < x < x_2$

$$f(x) = b_1 + (x - x_1)\{b_2 + (x - x_2)[b_3 + (x - x_1)b_4]\} \quad (2.7)$$

- Contributo *van Der Walls*: $x \geq x_2$

$$f(x) = - \left(\frac{C^{000}}{\epsilon R_m^6} \right) x^{-6} \quad (2.8)$$

Il parametro β , il quale influisce sulla forma della buca del potenziale, è caratteristico delle forze di van Der Walls ed è stato fissato pari a 6.7; le costanti b_1 , b_2 , b_3 e b_4 , sono state calcolate in modo tale da garantire lo stesso valore e la stessa derivata nei punti x_1 e x_2 .

Per quanto riguarda gli altri termini radiali, si hanno le seguenti espressioni:

$$V^{202}(R) = A^{202} e^{-c^{202}R - d^{202}R^2} - \frac{C^{202}}{R^6} \quad (2.9)$$

$$V^{220}(R) = A^{220e} e^{-c^{220}R} \quad (2.10)$$

$$V^{222}(R) = A^{222} e^{-c^{222}R} \quad (2.11)$$

Dove V^{202} descrive l'effetto dell'anisotropia delle forze repulsive a corto raggio e quelle dispersive a lungo raggio.

2.4 Espressioni delle configurazioni di riferimento del potenziale molecolare

Inserendo i valori opportuni di θ_a , θ_b e ϕ all'interno dell'equazione 2.3, si risale all'espressione dei quattro contributi base relativi alle configurazioni di riferimento riportate in figura 2.2.

Configurazione H ($\theta_a = 90^\circ$, $\theta_b = 90^\circ$ e $\phi = 0^\circ$)

$$V^H(R) = V^{000}(R) - \sqrt{5}V^{202}(R) + \sqrt{5}V^{220}(R) + \frac{5}{\sqrt{14}}V^{222}(R) \quad (2.12)$$

Configurazione H ($\theta_a = 90^\circ$, $\theta_b = 90^\circ$ e $\phi = 90^\circ$)

$$V^X(R) = V^{000}(R) - \sqrt{5}V^{202}(R) - \frac{\sqrt{5}}{2}V^{220}(R) - 2\frac{5}{\sqrt{14}}V^{222}(R) \quad (2.13)$$

Configurazione T ($\theta_a = 0^\circ$, $\theta_b = 90^\circ$ e $\phi = 0^\circ$):

$$V^T(R) = V^{000}(R) + \frac{\sqrt{5}}{2}V^{202}(R) - \frac{\sqrt{5}}{2}V^{220}(R) + \frac{5}{\sqrt{14}}V^{222}(R) \quad (2.14)$$

Configurazione L ($\theta_a = 0^\circ$, $\theta_b = 0^\circ$ e $\phi = 0^\circ$):

$$V^L(R) = V^{000}(R) + 2\sqrt{5}V^{202}(R) + \sqrt{5}V^{220}(R) - 2\frac{5}{\sqrt{14}}V^{222}(R) \quad (2.15)$$

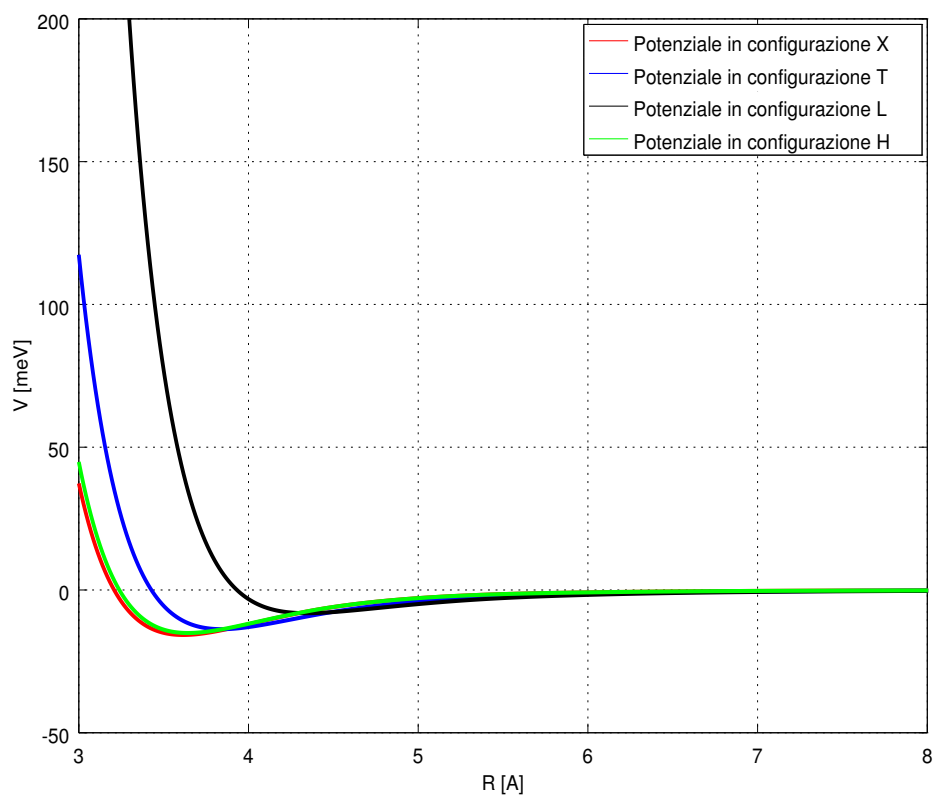


Figura 2.4: Potenziale in configurazione $X T L H$.

Capitolo 3

Distribuzione di Equilibrio per un set di molecole di ossigeno

Nel seguente capitolo viene descritta la procedura di inizializzazione delle velocità traslazionali e rotazionali del sistema di particelle utilizzato per simulare un set di molecole di ossigeno O_2 .

Dopo una breve introduzione teorica circa le equazioni utilizzate per definire le condizioni iniziali secondo la distribuzione di Maxwell, viene proposta la procedura utilizzata per riportare le nozioni teoriche nel codice DSMC. Segue infine la presentazione dei risultati che permettono di dedurre la corretta implementazione della subroutine di inizializzazione tramite un confronto con i risultati teorici.

3.1 Distribuzione di Maxwell

Si consideri una regione di spazio contenente un gas formato da un numero N di atomi o molecole; si ipotizzi poi che le particelle che compongono il gas siano libere di muoversi nello spazio ed interagiscano tra loro in corrispondenza di collisioni di durata istantanea; l'equazione che permette di descrivere la dinamica del gas in condizione di non equilibrio è data dall'equazione di Boltzmann descritta nel capitolo precedente:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}} = Q(f, f) \quad (3.1)$$

dove f è la funzione di distribuzione che dipende dalla posizione e dai gradi di libertà delle particelle, mentre Q è l'integrale di collisione.

È possibile dimostrare tramite opportuni passaggi matematici che l'equazione 3.1 è soddisfatta dalla funzione di equilibrio di Maxwell; quest'ultima, permette di determinare la probabilità che una particella abbia una velocità compresa all'interno di un intervallo prestabilito. La funzione f che per un insieme di molecole aventi 3 componenti di velocità traslazionali e 2 rotazionali è data dalla

seguente espressione:

$$f = N \frac{(mI)^{\frac{3}{2}}}{(2\pi K_B T)^3} e^{-\frac{m\|\xi\|^2 + I\|\omega\|^2}{2K_B T}} \quad (3.2)$$

La distribuzione di Maxwell appena presentata può essere rielaborata in maniera tale da poter distinguere il contributo rotazionale da quello traslazionale:

$$f_{tras} = \left(\frac{m}{2\pi K_B T} \right)^{\frac{3}{2}} e^{-\frac{m\|\xi\|^2}{2K_B T}} \quad f_{rot} = \left(\frac{I}{2\pi K_B T} \right)^{\frac{3}{2}} e^{-\frac{I\|\omega\|^2}{2K_B T}} \quad (3.3)$$

L'interpretazione matematica associata alle *p.d.f* di cui sopra corrisponde alla percentuale di molecole che possiedono il modulo della velocità traslazionale e rotazionale all'interno degli intervalli $[c, c + dc]$ e $[\omega, \omega + d\omega]$.

Per quanto riguarda il problema in esame, è stato ritenuto necessario adimensionalizzare le quantità finora presentate attraverso dei parametri di riferimento; questa procedura permetterà in seguito di poter confrontare le espressioni analitiche della distribuzione di Maxwell con i valori ottenuti dall'implementazione della subroutine per la generazione della condizione di equilibrio.

Parametro	Espressione
T_{ref}	T_0
P_{ref}	P_0
M_{ref}	m_{O_2}
L_{ref}	$\frac{K_B T_{ref}}{\sqrt{2\pi R c^2 P_0}}$
V_{ref}	$\sqrt{\frac{K_B T_{ref}}{m_{ref}}}$
t_{ref}	$\frac{L_{ref}}{V_{ref}}$

Tabella 3.1: Parametri di riferimento

Le funzioni di distribuzione in forma adimensionale (indicate attraverso il simbolo $*$) assumono perciò il seguente andamento:

$$f_{tras}^* = \left(\frac{m^*}{2\pi T^*} \right)^{\frac{3}{2}} e^{-\frac{\|\xi^*\|^2}{2T^*}} \quad f_{rot}^* = \frac{1}{\pi^{3/2} \omega_m^{*3}} e^{-\frac{\|\omega^*\|^2}{\omega_m^{*2}}} \quad (3.4)$$

dove la quantità $\omega_m^* = \sqrt{\frac{2T^* m_{ref} L_{ref}^2}{I}}$; per quanto riguarda infine il momento di inerzia I , esso è stato posto (dopo le opportune adimensionalizzazioni) pari a $\left(\frac{1}{2} \frac{d}{L_{ref}} \right)^2$ dove d è la lunghezza del manubrio che modella la molecola di ossigeno molecolare.

3.2 Implementazione della subroutine per la generazione della distribuzione di equilibrio

Per inizializzare le velocità iniziali dei centri di massa delle molecole di ossigeno è stato utilizzato il metodo Box-Muller; questa tecnica permette di ricostruire la distribuzione di Maxwell per quanto riguarda le componenti della velocità traslazionale utilizzando un algoritmo per la generazione casuale di numeri r compresi nell'intervallo $[0, 1]$ secondo distribuzione uniforme. Dal punto di vista implementativo, una volta imposta la temperatura globale adimensionale del sistema $T_{gl}^* = \frac{3T_{tr}^* + 2T_{rt}^*}{5}$, le equazioni utilizzate nella subroutine sono state le seguenti:

$$\begin{cases} \xi_{x_i}^* = \sqrt{-2\frac{T_{gl}^*}{m^*} \ln(r_a)} \cos(2\pi r_b) \\ \xi_{y_i}^* = \sqrt{-2\frac{T_{gl}^*}{m^*} \ln(r_c)} \cos(2\pi r_d) \\ \xi_{z_i}^* = \sqrt{-2\frac{T_{gl}^*}{m^*} \ln(r_c)} \sin(2\pi r_d) \end{cases} \quad (3.5)$$

Per quanto riguarda l'orientazione della molecola nello spazio, si è usufruito di un generatore casuale di vettori giacenti sulla generica sfera di raggio 1 e di componenti Kx, Ky, Kz ; questi ultimi infatti permettono di determinare in maniera univoca per ciascuna molecola gli angoli θ e φ .

Le componenti della velocità angolare invece sono state ricavate a partire dal contributo rotazionale dell'energia: sfruttando il generatore di numeri casuali utilizzato precedentemente è stato possibile assegnare un valore dell'energia di rotazione ad ognuna delle particelle contenute nel sistema:

$$E_{rot_i} = \frac{1}{2} I \|\boldsymbol{\omega}_i^*\|^2 = -T_{rot_i}^* \ln(r_i) \quad (3.6)$$

Si è poi proceduto a ricavare il modulo del vettore velocità angolare semplicemente invertendo la 3.6:

$$\|\boldsymbol{\omega}_i^*\| = \sqrt{\frac{2E_{rot_i}}{I}} \quad (3.7)$$

Il vettore $\boldsymbol{\omega}_i^*$ è stato orientato casualmente nello spazio definendo l'angolo $\alpha_i = 2\pi r_i$ come riportato in figura

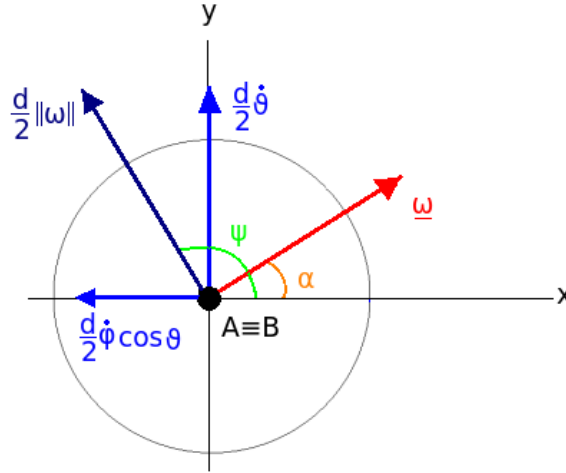


Figura 3.1: Disposizione della velocità angolare ω^* .

Dal prodotto vettoriale $\mathbf{R} \wedge \omega^*$ (\mathbf{R} è il versore associato alla i -esima molecola) si ottiene il vettore velocità tangenziale il quale è identificabile nel sistema di riferimento tramite l'angolo ψ (sfasato di 90° rispetto ad α). Sfruttando perciò le seguenti relazioni è possibile ottenere le componenti della velocità angolare:

$$\frac{d}{2} \dot{\varphi} \cos(\theta) = \frac{d}{2} \|\omega\| \cos(\psi) \quad \frac{d}{2} \dot{\theta} = \frac{d}{2} \|\omega\| \sin(\psi) \quad (3.8)$$

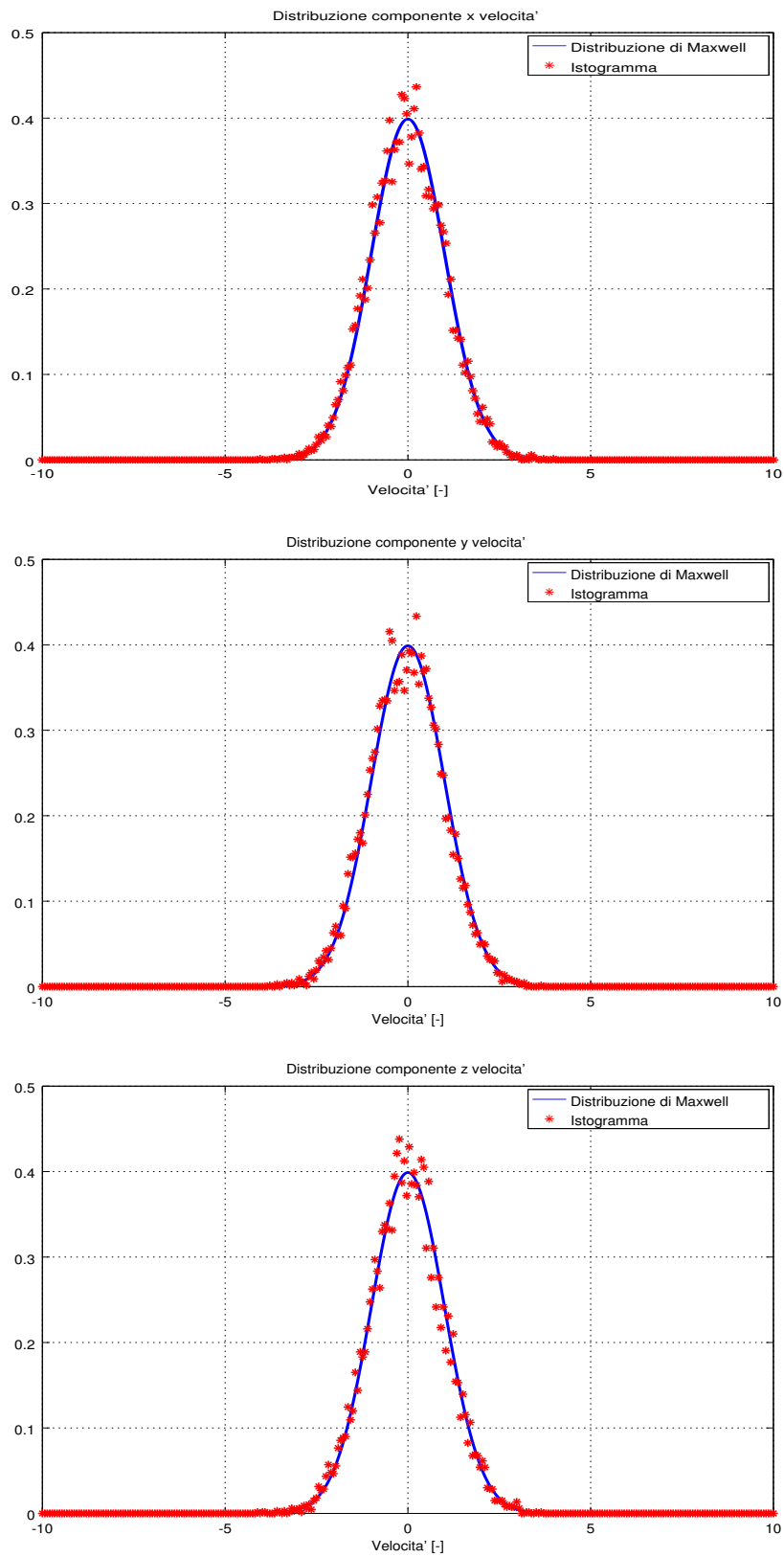
3.3 Confronto tra i risultati numerici ed analitici

I risultati ottenuti dall'implementazione numerica dei passaggi presentati nel paragrafo precedente sono stati confrontati con l'espressione analitica della distribuzione di equilibrio di Maxwell introdotta all'inizio di questo capitolo.

Per quanto riguarda le componenti traslazionali della velocità si è riscontrata una buona corrispondenza tra i valori forniti dalla subroutine e l'andamento della seguente funzione di distribuzione:

$$f_{tras_{x,y,z}}^* = \frac{1}{\sqrt{\frac{2\pi T^*}{m^*}}} e^{-\frac{1}{2} \frac{1}{T^*} \xi_{x,y,z}^{*2}} \quad (3.9)$$

Dal punto di vista grafico si evince che l'istogramma ricavato dal metodo numerico corrisponde in maniera coerente alla curva ottenuta a partire dall'equazione 3.9 a meno di rumore dovuto alla sparsità del campione considerato:

Figura 3.2: Istogrammi delle componenti x , y e z della velocità traslazionale.

Per quanto riguarda la distribuzione assunta dalla velocità angolare ω^* , si è deciso di sfruttare un approccio energetico come suggerito da Hinshelwood [6]: i gradi di libertà interni θ e φ sono infatti le variabili da cui dipende l'energia interna di ogni molecola secondo la seguente relazione:

$$E_{int} = \frac{1}{2}m \left(\frac{d}{2} \right)^2 (\dot{\theta}^2 + \dot{\varphi}^2 \cos^2 \theta) \quad (3.10)$$

Consideriamo ad esempio una molecola complessa: la sua energia totale può essere interpretata come la somma di n contributi quadratici. La probabilità che ciascun contributo energetico associato ad un particolare grado di libertà sia compreso tra $[Q, Q + dQ]$ è perciò:

$$\frac{Q^{-1/2}}{(\pi RT)^{1/2}} e^{-\frac{Q}{RT}} dQ \quad (3.11)$$

Ne consegue che la possibilità che due termini energetici (Q_1 e Q_2 siano simultaneamente compresi nei rispettivi intervalli $[Q_1, Q_1 + dQ_1]$ e $[Q_2, Q_2 + dQ_2]$ è definita dal prodotto delle rispettive probabilità:

$$\frac{1}{\pi RT} Q_1^{-1/2} e^{-\frac{Q_1}{RT}} dQ_1 \cdot Q_2^{-1/2} e^{-\frac{Q_2}{RT}} dQ_2 \quad (3.12)$$

tale per cui $Q_1 + Q_2 = E_{tot}$; ci chiediamo ora quale sia la probabilità che l'energia totale E_{tot} sia compresa tra i valori $[E_{tot}, E_{tot} + dE_{tot}]$ senza prendere in considerazione quali siano i contributi che la costituiscono. Per svolgere tale operazione è necessario esprimere $Q_2 = E_{tot} - Q_1$ ed inserire tale relazione nella 3.12; successivamente si procede all'integrazione rispetto a Q_1 :

$$\begin{aligned} \frac{dN}{N} &= \frac{1}{\pi RT} \int_0^{E_{tot}} Q_1^{-1/2} e^{-\frac{Q_1}{RT}} dQ_1 (E_{tot} - Q_1)^{-1/2} e^{-\frac{(E_{tot}-Q_1)}{RT}} dE_{tot} \\ &= \frac{e^{-\frac{E_{tot}}{RT}} dE_{tot}}{\pi RT} \int_0^{E_{int}} Q_1^{-1/2} (E_{tot} - Q_1)^{-1/2} dQ_1 \end{aligned} \quad (3.13)$$

dal momento che il valore assunto dalla quantità integrata è pari a π , l'equazione 3.13 diviene $\frac{dN}{N} = \frac{e^{-E_{tot}/RT} dE_{tot}}{RT}$; si deduce di conseguenza che la funzione di distribuzione di probabilità dell'energia totale formata dai due contributi Q_1 e Q_2 è:

$$f = \frac{1}{RT} e^{-\frac{E_{tot}}{RT}} \quad (3.14)$$

La procedura appena presentata può essere estesa al caso in cui i contributi energetici siano superiori a due; in tal caso, la funzione di densità di probabilità sarà ottenuta a partire dal prodotto delle n *p.d.f* dei contributi.

Dal momento che nel problema in esame siamo interessati esclusivamente all'energia interna associata a ciascuna molecola di ossigeno, si ha che l'espressione

3.10 può essere riscritta evidenziando i contributi associati ai due gradi di libertà $E_{int} = Q_1 + Q_2$ dove:

$$Q_1 = \frac{1}{2}m \left(\frac{d}{2}\right)^2 \dot{\theta}^2 \quad Q_2 = \frac{1}{2}m \left(\frac{d}{2}\right)^2 \dot{\varphi}^2 \cos^2\theta \quad (3.15)$$

La funzione di distribuzione di probabilità dell'energia interna diviene:

$$f_{E_{int}} = \frac{1}{RT} e^{-\frac{E_{int}}{RT}} \quad (3.16)$$

Utilizzando le quantità di riferimento introdotte nella tabella 3.1 è possibile adimensionalizzare la 3.16

$$f_{E_{int}}^* = \frac{m^*}{T^*} e^{-\frac{m^*}{T^*} E_{int}^*} \quad (3.17)$$

Nelle figure successive viene presentato il confronto tra i risultati ottenuti dall'implementazione della subroutine ed i valori analitici ottenuti a partire dall'equazione 3.17

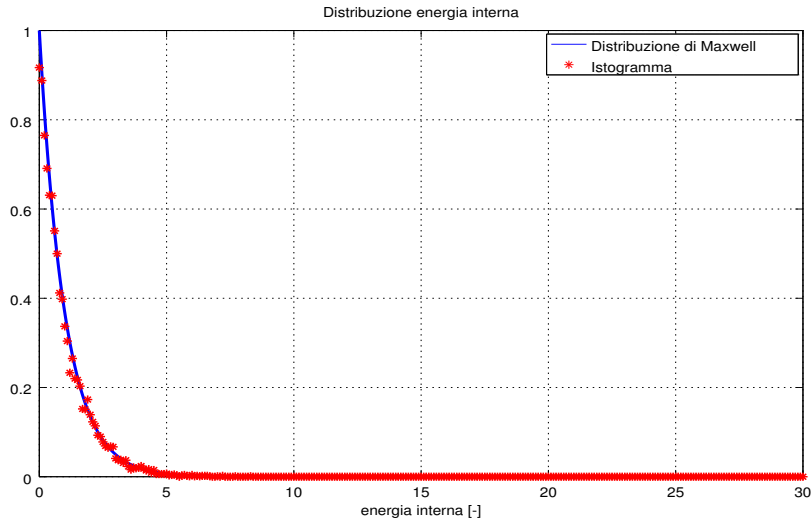


Figura 3.3: Istogramma dell'energia interna.

Il medesimo grafico è spesso riportato in letteratura in scala logaritmica:

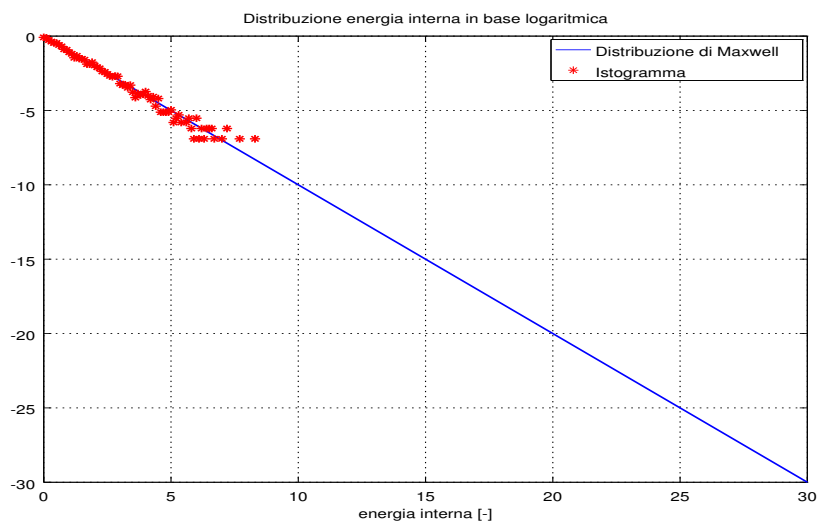


Figura 3.4: Istogramma dell'energia interna in scala logaritmica.

Il fatto che i valori assunti dall'istogramma non si estendono lungo tutto il dominio della funzione di densità di probabilità è giustificabile nella seguente maniera: dal momento che i valori assunti dalle estremità dell'istogramma sono nulli, non è possibile effettuare il logaritmo; di conseguenza tali quantità non vengono riportati in figura.

Capitolo 4

Struttura e validazione del codice CT-DSMC per il calcolo della viscosità dell'ossigeno molecolare

L'obiettivo di questo capitolo è quello di descrivere la struttura del codice CT-DSMC che ha il compito di simulare la dinamica collisionale tra le molecole di ossigeno. A seguito di una presentazione circa la tecnica di integrazione delle equazioni del moto del sistema di particelle preso in esame, viene proposta la validazione del programma tramite un confronto tra i risultati numerici ottenuti ed i dati offerti dalla letteratura.

4.1 Descrizione del codice DSMC

Prima di cominciare a descrivere il codice che si occupa della simulazione della viscosità dell'ossigeno molecolare O_2 , è importante ribadire le caratteristiche del gas presentate nel capitolo 1:

- Gas diluito $\frac{d}{\sigma} \gg 1$
- Collisioni istantanee tra le particelle $\frac{\tau}{t_c} \gg 1$ ¹
- Collisioni binarie: solo due molecole coinvolte per collisione

Solamente sotto le ipotesi appena elencate è possibile giustificare l'algoritmo impiegato per la risoluzione delle equazioni cinetiche tramite il metodo DSMC (Direct Simulation Monte Carlo). La funzione (*kernel*) che si occupa di simulare la dinamica collisionale tra le molecole di ossigeno è stata implementata da D. Ferrigni nella tesi da lui sviluppata ed intitolata “*Simulazione su GPU del rilassamento roto-traslazionale in un gas biatomico tramite calcolo di traiettorie*”

¹ τ è il libero tempo medio; t_c è il tempo di collisione medio.

classiche".

Con il termine DSMC si intende una tecnica probabilistica basata sul metodo Monte Carlo per risolvere l'equazione di Boltzmann presentata nel capitolo 1; tale metodo è stato proposto da G. Bird [5] come strumento per modellare la dinamica dei gas rarefatti; questa tipologia di approccio è tuttavia estendibile anche al regime continuo ovvero per $Kn < 0.1$. Il metodo DSMC modella i fluidi usando delle particelle rappresentative di un numero assai più elevato di molecole. Le collisioni tra le particelle possono essere simulate mediante diverse tipologie di modelli come ad esempio il Variable Hard Sphere (*VHS*) oppure il Variable Soft Sphere (*VSS*); nel caso in esame si è scelto il modello a sfere rigide (*HS*).

La simulazione DSMC è stata accoppiata al calcolo di traiettorie classiche (CT) le quali permettono di fornire una rappresentazione più realistica della dinamica collisionale ed una maggiore accuratezza nella capacità di predire i risultati. Dal punto di vista analitico, la meccanica classica consente di ricavare le equazioni del moto associate a due molecole in rotta di collisione in maniera deterministica; tale procedura richiede che venga definito un potenziale di interazione molecolare V che dipenda dalle posizioni relative e dall'orientazione delle molecole come presentato nel capitolo 3.

Per poter scrivere il sistema di equazioni differenziali di Lagrange, è necessario specificare le variabili indipendenti adottate nella descrizione del moto complessivo:

- (x_{CM}, y_{CM}, z_{CM}) : coordinate del centro di massa del sistema di particelle
- (R_x, R_y, R_z) : componenti del vettore \mathbf{R} che separa i centri delle molecole
- θ_a, φ_a : rotazione degli assi principali di inerzia della molecola A
- θ_b, φ_b : rotazione degli assi principali di inerzia della molecola B

Ne consegue che l'energia cinetica T e potenziale V del sistema sono funzione delle quantità² appena introdotte:

$$T(\dot{x}_{CM}, \dot{y}_{CM}, \dot{z}_{CM}, \dot{R}_x, \dot{R}_y, \dot{R}_z, \dot{\theta}_a, \dot{\varphi}_a, \dot{\theta}_b, \dot{\varphi}_b)$$

$$V(R_x, R_y, R_z, \theta_a, \varphi_a, \theta_b, \varphi_b)$$

Dato che il sistema è isolato da forze esterne, è possibile dimostrare che la quantità di moto totale si conserva; ciò implica che il centro di massa del sistema si muova di moto rettilineo uniforme, riducendo a 7 il numero di equazioni non banali. Definendo infine la lagrangiana del sistema $L = T - V$ si ottengono le equazioni del moto:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} = 0 \tag{4.1}$$

²Le variabili indipendenti sono definite nel sistema di riferimento assoluto; $\boldsymbol{\theta}$ è il vettore degli angoli θ e $\boldsymbol{\varphi}$; $\boldsymbol{\omega}$ è il vettore velocità angolare.

dove q_k sono le variabili $(R_x, R_y, R_z, \theta_a, \varphi_a, \theta_b, \varphi_b)$.

Il metodo di integrazione utilizzato per la risoluzione delle equazioni di Lagrange è Runge Kutta del quarto ordine (RK_4); nonostante l'elevato onere computazionale causato dalla necessità di valutare le derivate 4 volte per ogni time-step di integrazione, le collisioni sono risolte con un passo temporale sufficientemente elevato ed un numero di iterazione relativamente ridotto.

4.2 Uniform shear flow

Mediante l'utilizzo della subroutine descritta nel capitolo precedente, è stato possibile inizializzare le orientazioni delle particelle e la loro velocità traslazionale e rotazionale stabilendo di conseguenza una condizione di equilibrio termodinamico. Si ipotizzi ora di perturbare il set di molecole imponendo un gradiente di velocità costante α (*uniform shear rate*) lungo la direzione y ; di conseguenza la velocità media \mathbf{u} del flusso di molecole di O_2 risulta essere:

$$u_x = 0 \quad u_y = \alpha x \quad u_z = 0 \quad (4.2)$$

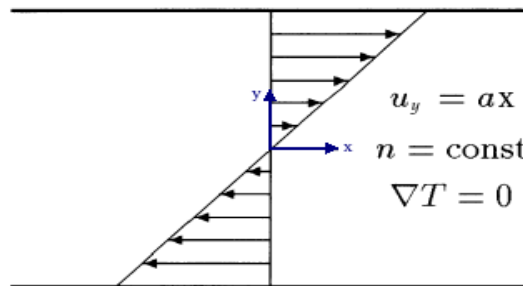


Figura 4.1: Andamento dello uniform shear flow.

Dal momento che le variazioni spaziali sono non nulle solamente lungo la direzione x , è ragionevole studiare l'equazione di Boltzmann nel caso monodimensionale; per motivi pratici si fa riferimento al caso monoatomico:

$$\frac{\partial f}{\partial t} + \xi_x \frac{\partial f}{\partial x} = Q(f, f) \quad (4.3)$$

Osservando che le soluzioni dell'equazione 4.3 sono invarianti rispetto ad una traslazione del sistema di riferimento, la funzione di distribuzione di probabilità assume la seguente forma:

$$f(x, \xi, t) = f(\xi - \alpha x \hat{y}, t) \quad (4.4)$$

Applicando il ragionamento appena proposto all'equazione 4.3, lo shear rate assume le proprietà di una forza dissipativa dipendente dalla velocità che agisce uniformemente nello spazio; dal punto di vista analitico si ottiene la seguente formulazione:

$$\frac{\partial f}{\partial t} - \alpha c_x \frac{\partial f}{\partial c_y} = Q(f, f) \quad (4.5)$$

dove $\mathbf{c} = \boldsymbol{\xi} - \alpha x \hat{\mathbf{y}}$. Dal punto di vista implementativo, lo uniform shear flow è stato raggiunto sottraendo alla componente y della velocità di ciascuna particella la quantità αx tale per cui:

$$\xi_y \rightarrow \xi_y - \alpha \xi_x dt \quad (4.6)$$

L'operazione appena introdotta deve essere ripetuta ad ogni istante temporale. È importante sottolineare che la presenza dello shear provoca della dissipazione di energia sotto forma di calore ed un conseguente aumento di temperatura del gas. Per questo motivo è stato implementato un termostato Gaussiano il cui compito è quello di mantenere la temperatura traslazionale del sistema costante. Questa condizione è raggiungibile aggiungendo alla 4.5 un termine forzante:

$$\frac{\partial f}{\partial t} - \alpha c_x \frac{\partial f}{\partial c_y} - \beta \frac{\partial}{\partial \mathbf{c}} \cdot (f\mathbf{c}) = Q(f, f) \quad (4.7)$$

Il valore del coefficiente β deve essere scelto opportunamente affinché l'energia dissipata sotto forma di calore non provochi un aumento della temperatura. Dal punto di vista implementativo, il termostato è rappresentato da una scalatura delle componenti della velocità traslazionale le quali devono essere moltiplicate per un coefficiente fsv calcolato ad ogni time-step come:

$$fsv = \frac{T_R}{T_{rt}} \quad (4.8)$$

Dove T_R è la temperatura traslazionale di riferimento, mentre T_{rt} è la temperatura traslazionale istantanea calcolata dopo aver effettuato il processo di collisione tra le particelle.

In generale, imporre un gradiente di velocità in presenza di gas poliatomico provoca la separazione della temperatura traslazionale da quella rotazionale. Lo scostamento tra T_{tr} e T_{rot} è tanto più piccolo quanto minore è lo shear α desiderato.

Come verrà mostrato nel paragrafo seguente, l'introduzione dello uniform shear flow nel sistema di particelle è di vitale importanza per il calcolo delle quantità di trasporto come ad esempio la viscosità.

4.3 Calcolo delle viscosità

Come descritto nei paragrafi precedenti, il metodo CT-DSMC ha permesso di risolvere numericamente l'equazione 1.15 in presenza di gradiente di velocità indotto. Gli effetti dell'incremento della temperatura sono stati gestiti riscaldando

le velocità delle particelle per mezzo di un termostato.

A questo punto è possibile calcolare lo sforzo di taglio p_{xy} replicando dal punto di vista numerico l'integrale 1.19

$$p_{xy} = \int c_x c_y f d\xi \quad (4.9)$$

ossia la componente xy del tensore degli sforzi $\underline{\underline{P}}$.

La viscosità η è stata calcolata utilizzando uno shear rate molto piccolo tale da soddisfare il seguente limite:

$$\lim_{\alpha \rightarrow 0} \frac{p_{xy}}{\alpha} \quad (4.10)$$

Per poter confrontare i risultati ottenuti con i valori sperimentali contenuti in letteratura, si richiede che lo sforzo di taglio p_{xy} e lo shear rate α vengano riportati in forma dimensionale; ciò può essere fatto moltiplicando rispettivamente per la pressione di riferimento P_{ref} ed il tempo di riferimento t_{ref} :

Parametro	Espressione
$P_{ref} [Pa \cdot s]$	P_0
$t_{ref} [s]$	$\frac{\lambda}{\sqrt{\frac{k_b T_{ref}}{m_{O_2}}}}$

Tabella 4.1: Pressione e tempo di riferimento

Le quantità appena introdotte permettono di definire una viscosità di riferimento $\eta_{ref} = P_{ref} t_{ref}$ che moltiplicata per la viscosità adimensionale conduce al valore di interesse:

$$\eta = \eta^* \eta_{ref} \quad (4.11)$$

dove η^* è la viscosità adimensionale ottenuta a partire dal limite 4.10.

Nella seguente figura viene proposto l'andamento assunto dallo sforzo di taglio e dalla temperatura traslazionale/rotazionale

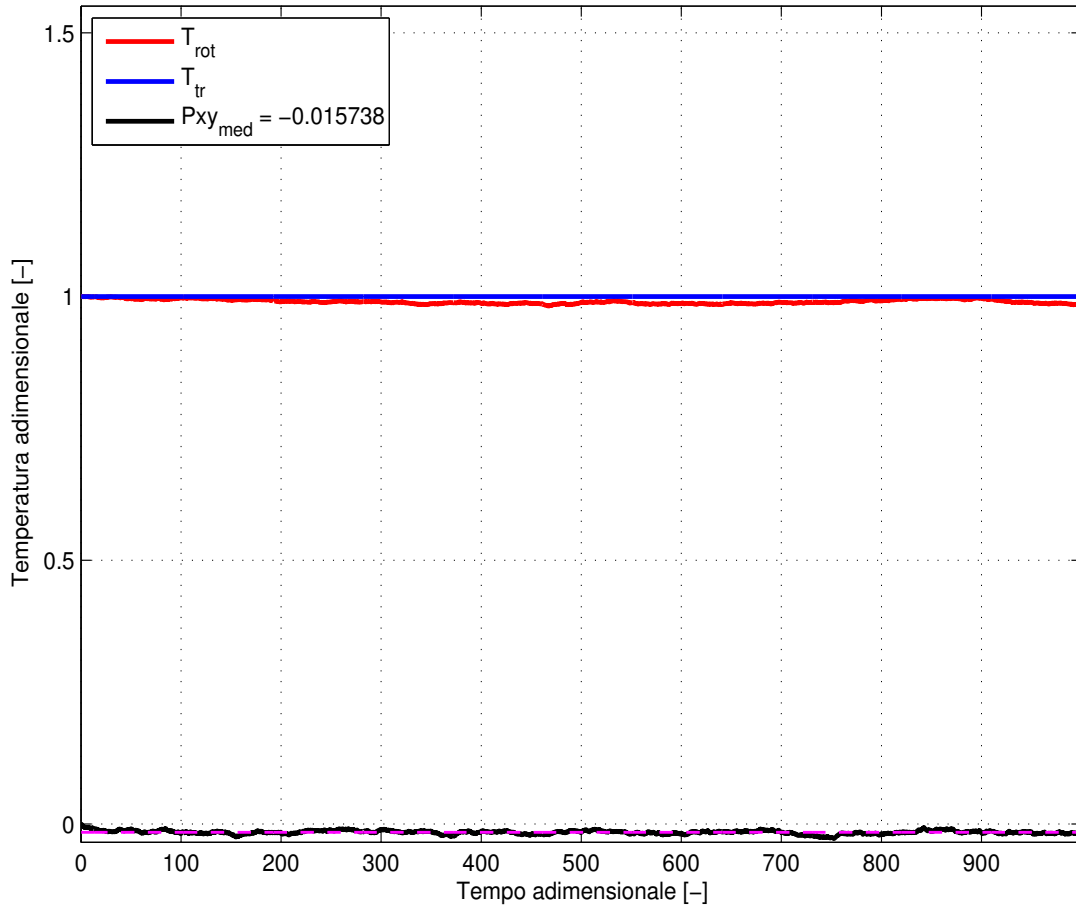


Figura 4.2: $N_p = 100000$, $\alpha = 0.001$, $T_{ref} = 300K$, $P_{ref} = 10^5 Pa$.

Come si evince dal grafico, lo scostamento tra T_{tr} e T_{rot} è minimo grazie all'utilizzo di un α basso. Lo sforzo di taglio è caratterizzato invece da un breve transitorio che porta p_{xy} da un valore nullo ad oscillare intorno al valore medio \bar{p}_{xy} .

Dal punto di vista quantitativo, il valore della viscosità ottenuto per un set di $N_p = 100000$ particelle alla temperatura di $T = 300K$ ed a $P = 10^5 Pa$ di pressione è:

$$\eta = \left(\frac{\bar{p}_{xy}}{\alpha} \right) \eta_{ref} = 2.1556 \cdot 10^{-5} Pa s$$

4.4 Validazione del codice

Al fine di avere un maggior numero di valori per poter effettuare un confronto con i dati sperimentali offerti dalla letteratura, si è deciso di ripetere la prova per un range di temperature comprese tra $200K$ e $500K$. I valori sperimentali

utilizzati sono disponibili in [8] e rispettano l'ipotesi di gas diluito; essi inoltre sono stati campionati ad una pressione pari a quella definita nel codice CT-DSMC. In tabella vengono ora riportati i valori numerici ottenuti e l'errore percentuale rispetto alle viscosità di riferimento:

Temp [K]	η CT-DSMC [Pa s]	η Sperimentale [Pa s]	Errore [%]
200	$1.578 \cdot 10^{-5}$	$1.475 \cdot 10^{-5}$	7.353
300	$2.156 \cdot 10^{-5}$	$2.081 \cdot 10^{-5}$	3.585
400	$2.698 \cdot 10^{-5}$	$2.573 \cdot 10^{-5}$	3.781
500	$3.177 \cdot 10^{-5}$	$3.080 \cdot 10^{-5}$	3.169

Tabella 4.2: Confronto tra valori numerici CT-DSMC e valori sperimentali.

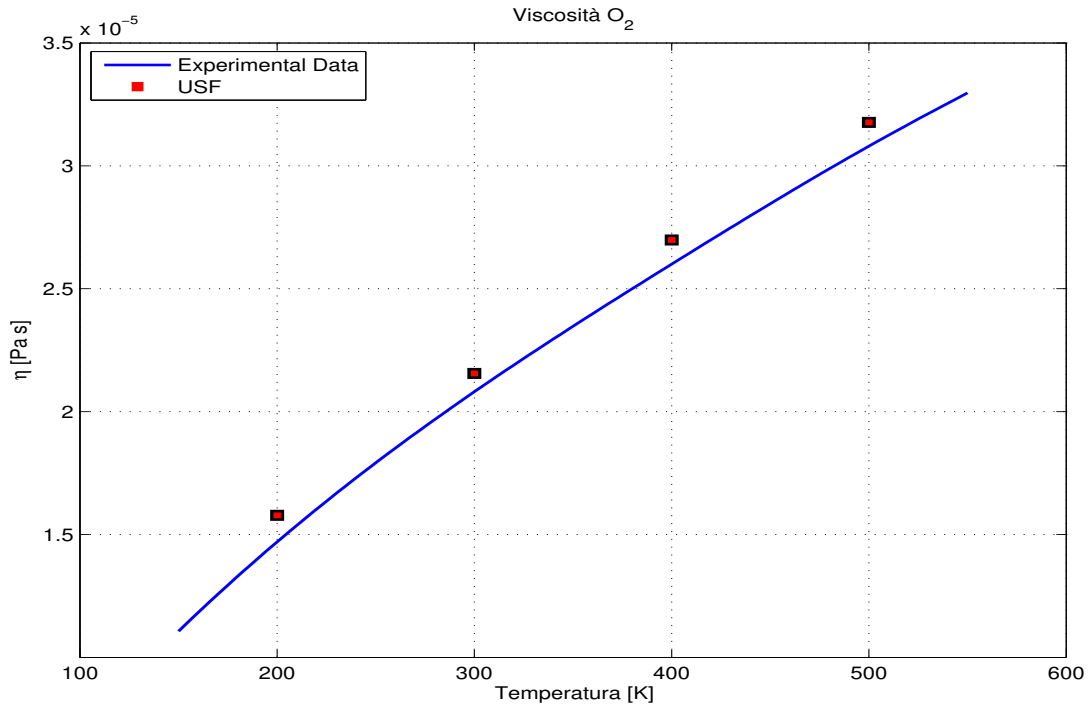


Figura 4.3: Viscosità O_2 al variare della temperatura.

L'errore percentuale risulta essere in media minore del 5%; si deduce quindi che la viscosità η calcolata assume valori realistici e lo scostamento rispetto ai valori sperimentali può essere imputato ai seguenti fattori:

- Durata della simulazione non eccessivamente elevata
- Shear rate α troppo elevato

- Incertezza statistica
- Utilizzo dei valori nominali dei parametri del potenziale V

Capitolo 5

Prova di Speed-Up del codice CT-DSMC e vantaggi dell'utilizzo della GPU

Nel seguente capitolo viene presentato il linguaggio di programmazione impiegato nel codice di dinamica molecolare per la stima della viscosità dell'ossigeno O_2 . A seguito di una breve descrizione circa la struttura del device (GPU) utilizzato per eseguire la parallelizzazione del programma, viene proposto un primo tentativo di ottimizzazione del codice per la riduzione dei tempi di calcolo. Per concludere viene effettuata l'analisi di Speed-Up del il codice parallelizzato rispetto all'analogica versione seriale; il confronto è stato eseguito per due versioni del codice di dinamica collisionale in maniera tale da verificare per via indiretta il corretto utilizzo delle performance della scheda grafica.

5.1 Linguaggio di programmazione delle schede grafiche (GPU)

La richiesta da parte del mercato di schede grafiche ad alte prestazioni ha portato nel corso degli anni alla creazione di dispositivi altamente performanti caratterizzati dalla possibilità di effettuare un considerevole numero di calcoli in parallelo. Dal punto di vista componentistico, le GPU sono dotate di un'elevata quantità di transistors dedicati esclusivamente al processamento dei dati piuttosto che alla loro allocazione in memoria o al controllo del loro flusso:

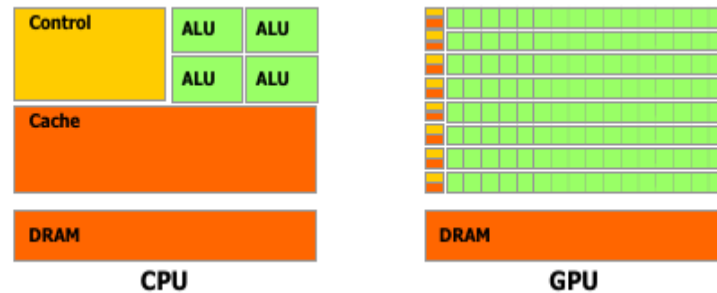


Figura 5.1: In *arancione/giallo* regione dedicata alla memoria; in *verde* regione dedicata al processamento dei dati.

Come detto, le GPU risultano essere ideali nella risoluzione di problemi in cui è richiesta la parallelizzazione: essa consiste nella suddivisione del problema di riferimento in più sottoprocessi che prendono il nome di *threads* i quali vengono eseguiti contemporaneamente.

Verranno ora presentati il linguaggio di programmazione tipicamente utilizzato per l'elaborazione dati tramite scheda grafica e le varie tipologie di memorie che ne fanno parte.

5.2 CUDA (Compute Unified Device Architecture)

Il linguaggio di programmazione CUDA è stato rilasciato dalla casa produttrice di schede grafiche NVIDIA con lo scopo di fornire uno strumento in grado di sfruttare l'elevata velocità di calcolo in parallelo delle GPU (*device*). Per la precisione CUDA è un'architettura costituita da un set di istruzioni e comandi che sfrutta altri linguaggi di programmazione come C, C++ e Fortran per la parallelizzazione di codici da eseguire su gruppi di sottoprocessi (*threads*) indipendenti; essi vengono organizzati in blocchi i quali sono a loro volta disposti secondo una griglia per essere poi elaborati dai processori che costituiscono il device.

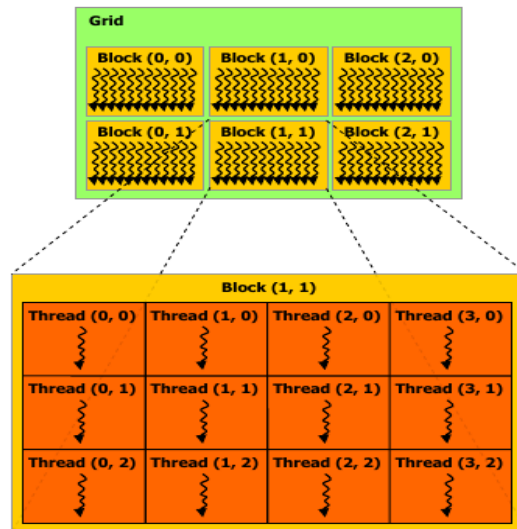


Figura 5.2: Distribuzione di threads e blocchi in 2D.

Threads e Blocchi possono essere caratterizzati da 1, 2 o 3 indici in maniera tale da adattarsi alle dimensioni del dominio del problema in questione (1D, 2D, 3D).

5.2.1 Kernel

Per mezzo di CUDA, è possibile accostare al linguaggio seriale eseguito dalla CPU (*host*) parti di codice dedicate al calcolo parallelo. Le funzioni che vengono eseguite sulla GPU prendono il nome di *kernel* ed sono identificate all'interno del codice tramite lo specifier `__global__`. Ogni kernel deve essere caratterizzato dal numero di threads e blocchi in cui viene partizionato il problema. Qualora vi siano delle subroutines che vengono eseguite esclusivamente dal codice parallelo, esse devono essere accompagnate dallo specifier `__device__`.

```

__global__ void kernelFunction(int param){
    // Parallel code
}

int main(){
    // Serial code
    ...
    // Kernel invocation
    kernelFunction<<< gridDim, blockDim >>>(param);
}

```

Figura 5.3: Esempio di CUDA kernel.

5.2.2 Threads

Quando il kernel viene richiamato, ogni thread deve eseguire in parallelo le istruzioni contenute nella funzione. Ogni sottoprocesso è caratterizzato da un codice di identificazione (ID) che è calcolabile a partire dalla posizione del generico thread all'interno del blocco (*threadIdx*), dalla posizione del blocco nella griglia (*blockIdx*), dal numero di threads per blocco (*blockDim*) e dal numero di blocchi per griglia (*gridDim*). L'operazione necessaria per identificare ciascun sottoprocesso è la seguente:

```

__device__ threadIdxUniqueId() {
    return threadIdx.x + blockDim.x*threadIdx.y + blockDim.x
        *blockDim.y*threadIdx.z;
}

```

Figura 5.4: Formula per l'identificazione dei thread.

5.2.3 Streaming Multiprocessors (SM)

L'hardware della GPU copre un ruolo fondamentale per un suo corretto ed efficiente utilizzo; l'unità fondamentale che impone dei vincoli alle performance dei threads è lo *streaming multiprocessor (SM)*; esso ha il compito di processare un elevato numero di threads i quali sono organizzati in gruppi di 32 (*warps*) tramite un'architettura denominata SIMT (*Single-Instruction Multiple Thread*). Qualora uno o più blocchi siano assegnati ad un SM per essere processato, essi vengono suddivisi in warps organizzati dal *warp scheduler*; nonostante il multiprocessore possa ospitare *n* warps ($24 < n < 64$), esso può eseguirne solamente uno alla volta. Un possibile motivo di questa scelta è dovuto al fatto che mentre un warp viene eseguito, un altro set da 32 threads sfrutta il tempo di calcolo per effettuare alcune istruzioni caratterizzate da grande latenza come ad esempio gli accessi in memoria; in questo modo è possibile mantenere il multiprocessore sempre occupato.

5.2.4 Tipologie di memoria

La struttura della scheda grafica è tale da disporre diverse tipologie di memoria ciascuna delle quali ha una dimensione ed uno scopo distinto. Conoscere le differenze tra le varie memorie a disposizione è di fondamentale importanza per un corretto utilizzo del device e per usufruire al massimo delle possibilità di calcolo. Qualora esse non vengano utilizzate in maniera appropriata, si possono generare dei "colli di bottiglia" durante l'esecuzione dei threads diminuendo di conseguenza l'efficienza del calcolo parallelo.

- **Global Memory:** la memoria globale risiede sul device e può essere utilizzata da ogni thread; nonostante l'elevato tempo di accesso, essa ha la capacità di immagazzinare una grande quantità di informazioni. La memoria globale, prima di essere utilizzata, deve essere allocata utilizzando l'apposita funzione *cudaMalloc*. Una volta allocato lo spazio in memoria, è necessario copiare in GPU i dati presenti sulla CPU. Questa operazione può essere fatta tramite il comando *cudaMemcpy*. Una volta terminata l'operazione di processamento dati, questi vengono trasferiti nuovamente dalla memoria globale al host.
Per sfruttare al meglio la bandwidth, tutti gli accessi alla memoria globale devono essere coalescenti, i.e gli accessi in memoria da thread differenti devono essere raggruppati e eseguiti con un'unica transizione di memoria.
- **Constant Memory:** La memoria costante è ideale per ospitare coefficienti e altri dati a cui si accede in modo uniforme e in sola lettura. Per tutti i dati che nei kernel verranno solo letti, questa memoria è preferibile rispetto alla memoria globale in quanto permette di ottenere prestazioni grazie ai meccanismi di caching che essa mette a disposizione. I limiti di questa memoria, oltre al fatto di essere di sola lettura, stanno nelle sue piccole dimensioni. Lo specifier da utilizzare in questo caso è semplicemente `__constant__`.
- **Shared Memory:** La memoria condivisa è una memoria veloce residente su ciascun Streaming Multiprocessor; essa viene condivisa da tutti i threads appartenenti ad un blocco e può essere usata per comunicazione e sincronizzazione tra i threads stessi. La shared memory è accessibile sia in lettura e scrittura ed è ha una durata pari alla vita del blocco. Il principale vantaggio della shared memory è la sua elevata rapidità di accesso; tuttavia la dimensione è limitata.
- **Local Memory:** Ogni thread ha dedicata una memoria locale la quale viene generalmente utilizzata meno sia perché ha dei tempi di accesso paragonabili alla global memory, sia perché essendo locale al thread non ha lo scopo di contenere i risultati globali delle operazioni eseguite dal kernel. Le variabili che risiedono in memoria locale sono variabili che hanno visibilità ristretta al singolo thread e che quindi hanno un ciclo di vita pari alla durata del kernel.
- **Texture Memory:** è una tipologia di memoria alternativa alla constant memory ottimizzata per accessi bidimensionali e quindi utilizzata nel caso di domini 2D.
- **Registri:** essi sono una regione di memoria da 32bit e presentano i migliori tempi di accesso in assoluto. Ogni streaming multiprocessor ha a disposizione in base al tipo di scheda grafica 16K, 32K o 64K di registri. Sebbene

queste cifre possano sembrare elevate, è importante tenere presente che ogni SM gestisce 768, 1024 o 2048 threads in base alla compute capability del device; di conseguenza si richiede di trovare un bilancio corretto circa il numero di registri da assegnare ad ogni thread. Se il numero di registri richiesti è eccessivo allora vengono limitate le performance della GPU.

5.3 Proprietà della scheda grafica GeForce GTX 780

Per lo sviluppo di questa tesi, si è utilizzata una scheda grafica GeForce GTX 780; le caratteristiche tecniche sono ottenibili sfruttando il pacchetto *queryDevice* offerto da NVIDIA:

```
CUDA Device Query Driver API statically linked version
Detected 2 CUDA Capable devices
```

```
Device 0: "GeForce GTX 780"
```

```
CUDA Driver Version:          6.5
CUDA Capability Major/Minor version number:  3.5
Total amount of global memory:  3072 MBytes 3221028864 bytes
12 Multiprocessors, 192 CUDA Cores/MP:      2304 CUDA Cores
GPU Clock rate:                902 MHz 0.90 GHz
Memory Clock rate:             3004 Mhz
Memory Bus Width:              384-bit
L2 Cache Size:                 1572864 bytes
Max Texture Dimension Sizes    1D=65536 2D=65536, 65536 3D=4096, 4096, 4096
Maximum Layered 1D Texture Size, num layers 1D=16384, 2048 layers
Maximum Layered 2D Texture Size, num layers 2D=16384, 16384, 2048 layers
Total amount of constant memory: 65536 bytes
Total amount of shared memory per block:    49152 bytes
Total number of registers available per block: 65536
Warp size:                       32
Maximum number of threads per multiprocessor: 2048
Maximum number of threads per block:       1024
Max dimension size of a thread block x,y,z: 1024, 1024, 64
Max dimension size of a grid size x,y,z:   2147483647, 65535, 65535
Texture alignment:               512 bytes
Maximum memory pitch:           2147483647 bytes
Concurrent copy and kernel execution:      Yes with 1 copy engines
Run time limit on kernels:         No
Integrated GPU sharing Host Memory:       No
Support host page-locked memory mapping:  Yes
Concurrent kernel execution:         Yes
Alignment requirement for Surfaces:      Yes
Device has ECC support:           Disabled
Device supports Unified Addressing UVA:    Yes
```

Device PCI Bus ID / PCI location ID: 4 / 0
Compute Mode:

Osservando i dati appena presentati, è possibile ricavare ulteriori dettagli quali il numero massimo di warps per multiprocessore; nel caso in esame risulta essere pari a $2048/32 = 64$. È infine considerata buona prassi settare il numero di threads per blocco pari ad un multiplo di 32 affinché venga favorito il riempimento dello streaming multiprocessor.

5.4 Descrizione del programma-test per la stima dei tempi di esecuzione

Come riportato nel capitolo precedente, il codice scritto in linguaggio CUDA ha il compito di stimare la viscosità dell'ossigeno O_2 sotto l'influenza di uno uniform shear flow. Dal momento che il secondo obiettivo di questa tesi riguarda l'analisi di Speed-Up del codice parallelizzato rispetto all'analogica versione seriale, si è deciso di modificare il programma per effettuare la stima dei tempi di esecuzione della funzione che simula la dinamica di collisione intermolecolare; per condurre tale analisi, sono state usate due versioni distinte del kernel ed il motivo di questa scelta sarà chiaro nel sotto-paragrafo successivo.

Per cominciare è importante descrivere il workflow del programma-test il quale è formato da una parte principale (main), eseguita dalla CPU, che stabilisce il numero di collisioni ed assegna a ciascuna di esse le condizioni iniziali in termini di velocità (traslazionali e rotazionali) e di orientazione. Tali informazioni sono contenute all'interno di una struttura la quale viene copiata sulla GPU mediante i comandi descritti nei paragrafi precedenti. Ad ogni collisione viene associato un thread il quale ha il compito di effettuare i conti relativi al generico urto. Poiché è consuetudine che il numero di threads per blocco sia un multiplo di 32, risulta conveniente aggiungere delle collisioni fittizie oltre alle collisioni reali calcolate dall'host. Di conseguenza la GPU dovrà eseguire ad ogni time-step un numero di collisioni pari ad N_{thread} ; questa quantità è costituita sia da collisioni reali e da collisioni fittizie. Il programma-test si conclude senza effettuare la stima della viscosità in quanto essa risulta essere superflua nel calcolo del tempo di esecuzione del kernel.

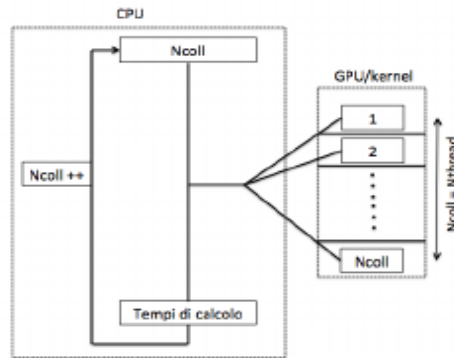


Figura 5.5: Workflow del programma-test per l'analisi dei tempi di esecuzione.

5.4.1 Collisioni con time-step fisso e con adattività di passo

La funzione che si occupa della dinamica collisionale intermolecolare è stata implementata dall'autore in due versioni distinte:

- Collisioni con time-step fisso: le collisioni sono risolte tutte con lo stesso passo temporale adimensionale pari a 10^{-4} e con il medesimo numero di iterazioni (40000). Quest'ultimo è stato scelto empiricamente poiché si è verificato che dopo 40000 time-step le molecole che partecipano ad una collisione hanno lasciato la regione di influenza del potenziale. In parole semplici, bisogna garantire che dopo 40000 iterazioni, la coppia di molecole urti e si separi.
- Collisioni con adattività di passo: a differenza del codice a passo fisso, l'integrazione temporale è caratterizzato da un time-step adattivo basato sul criterio di conservazione dell'energia totale tra due iterazioni successive del moto. Ad ogni iterazione, il kernel verifica che la differenza tra l'energia totale istantanea e quella iniziale sia minore di una tolleranza prefissata; in caso ciò non si verifichi, il time-step di integrazione (secondo RK4) si dimezza.

Dal momento che il potenziale V dipende dalla distanza R che separa le molecole, si osserva che le accelerazioni sono elevate qualora le molecole siano vicine; di conseguenza è necessario un passo di integrazione più piccolo rispetto al caso in cui R sia elevato. Inoltre, poiché V è definita a tratti si è reso necessario garantire la conservazione dell'energia entro una soglia di tolleranza anche a monte e a valle delle regioni di passaggio tra un'espressione del potenziale e l'altra.

Tra i vantaggi dell'utilizzo dell'adattività di passo vi sono la possibilità di risolvere le collisioni che richiedono un time-step ridotto per garantire la conservazione dell'energia e l'opportunità di utilizzare passi di integrazione

maggiori qualora le accelerazioni in gioco siano basse. Il codice con time-step variabile dispone di un numero massimo di iterazioni per il dimezzamento del passo di integrazione, ed un limite al numero di integrazioni delle equazioni; di conseguenza si evita che le collisioni che richiedono un time-step troppo piccolo appesantiscano eccessivamente i calcoli.

5.5 Tentativi di ottimizzazione e riduzione dei tempi di calcolo

Rispetto alla versione originale, il codice è stato leggermente modificato con l'intento di ottimizzare l'utilizzo delle memorie; nel problema in esame si è deciso infatti di collocare tutte le costanti presenti nel problema (come ad esempio quelle presentate in figura 2.3) nella constant memory anziché nella memoria globale. In termini prestazionali si è notato che i tempi di esecuzione non sono cambiati in maniera significativa.

Successivamente si è fatto ricorso all'opzione `-USE_FAST_MATH` la quale permette di risolvere le operazioni di elevazione a potenza in maniera più veloce rispetto alle funzioni contenute nella libreria `math.h`; con tale approccio, le operazioni vengono eseguite da funzioni definite direttamente sul device senza doverle richiamare dall'host.

Aggiungendo l'opzione `-ptxas-options=-v` al makefile del programma, è stato possibile ottenere i file assembly generati dal compilatore `nvcc` ed analizzare le informazioni relative alla fase di compilazione. In questo caso si è osservato che l'architettura usata da `nvcc` non era coerente con la *compute capability* della scheda grafica. Per poter sfruttare in maniera completa le caratteristiche della GPU è stato necessario esplicitare in fase di compilazione l'architettura desiderata tramite l'opzione `-arch=compute_35` e `-code=sm_35` permettendo di usufruire delle risorse offerte dal device. Applicando tale cambiamento il numero di registri è incrementato notevolmente passando da 64 a 232.

Dopo diversi tentativi, è emerso che la quantità di registri da assegnare a ciascun thread rappresenta un elemento fondamentale per l'abbattimento dei tempi di esecuzione. Nel problema in esame si è osservato che il numero ideale è 128 reg/thread. Per indicare al compilatore il valore desiderato è stato utilizzato il comando `-maxrregcount`.

5.6 Analisi dei tempi di calcolo

Prima di accingersi a confrontare i tempi di esecuzione del codice CUDA con l'analoga versione seriale scritta in linguaggio C, è importante determinare le migliori condizioni di operabilità della GPU. Le schede grafiche infatti hanno bisogno di eseguire un elevato numero di operazioni per poter sfruttare al meglio

le risorse di calcolo offerte dal device. L'intento è quello di assegnare più blocchi a ciascun SM in maniera tale da nascondere la latenza del multiprocessore permettendo ai warps di prepararsi prima di essere eseguiti. In generale le GPU raggiungono il massimo utilizzo delle loro risorse qualora si distribuiscono 3 – 4 blocchi per SM.

La condizione appena descritta è raggiungibile definendo un numero di collisioni sufficientemente elevato affinché si raggiunga la saturazione del device. Dal punto di vista grafico l'andamento ottenuto è il seguente:

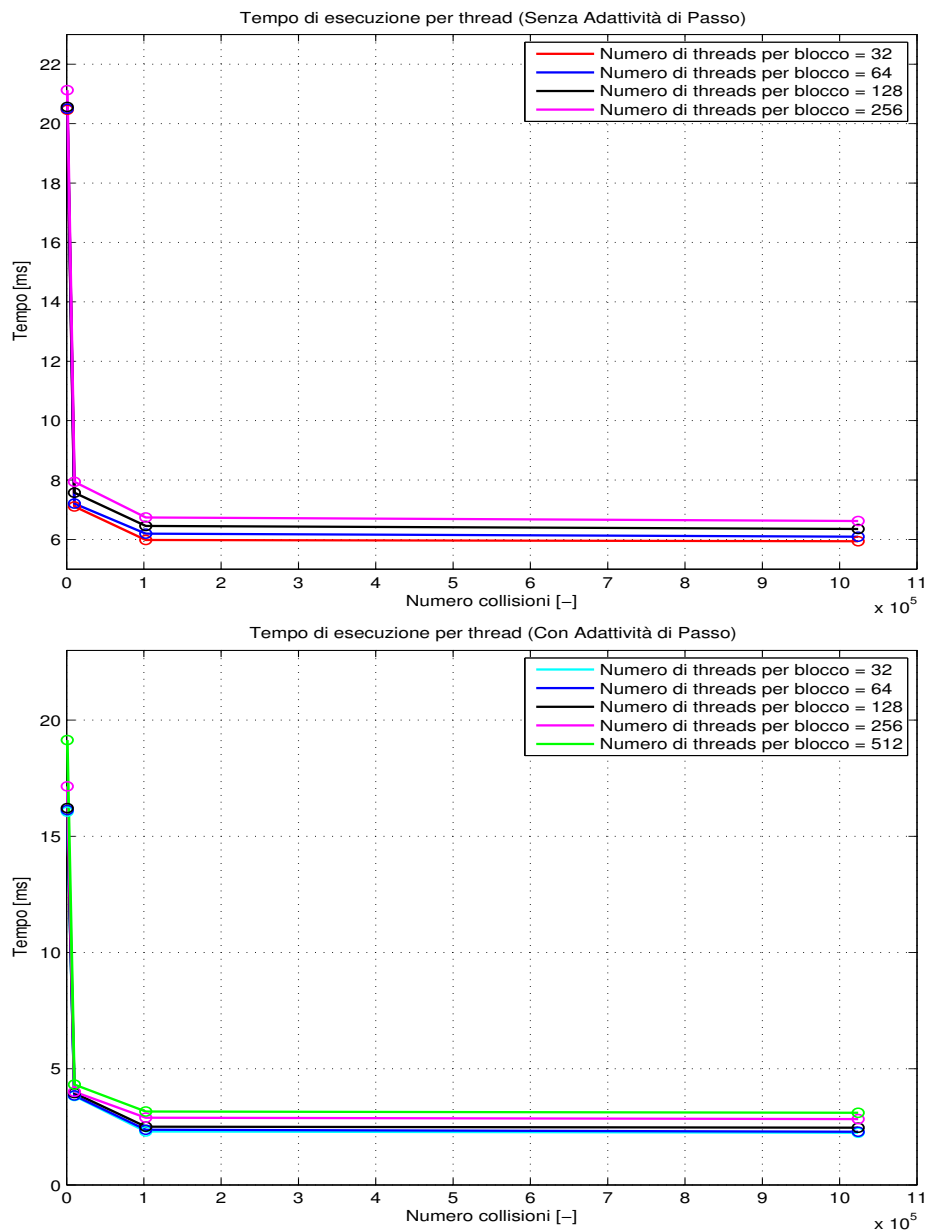


Figura 5.6: Tempo di esecuzione del singolo thread.

Come si evince da entrambe le figure, i tempi di esecuzione del singolo thread calano drasticamente all'aumentare delle collisioni fornite al device; in entrambi i casi infatti sono necessarie almeno 100000 collisioni affinché la GPU utilizzi tutte le risorse a sua disposizione. Si osserva inoltre che il tempo di esecuzione per thread nel caso in cui l'integrazione sia effettuata a passo fisso è maggiore rispetto ai valori ottenuti con kernel a time-step variabile; tale risultato è dovuto al fatto che tutte le collisioni associate alla funzione con passo temporale fissato impiegano lo stesso tempo di esecuzione indipendentemente dal tipo di urto il quale potrebbe essere risolto anche con un numero di iterazioni minori.

Dal punto di vista quantitativo, i valori ottenuti vengono ora riportati in tabella:

N_{coll}	t_{32TxB} [ms]	t_{64TxB} [ms]	t_{128TxB} [ms]	t_{256TxB} [ms]
1000	20.47	20.50	20.56	21.13
10000	7.11	7.21	7.57	7.94
100000	5.98	6.19	6.45	6.73
1000000	5.93	6.09	6.35	6.62

Tabella 5.1: Tempi di esecuzione dei threads con passo fisso.

N_{coll}	t_{32TxB} [ms]	t_{64TxB} [ms]	t_{128TxB} [ms]	t_{256TxB} [ms]	t_{512TxB} [ms]
1000	16.06	16.13	16.21	17.15	19.14
10000	3.83	3.85	3.96	4.00	4.32
100000	2.30	2.38	2.51	2.89	3.16
1000000	2.24	2.29	2.46	2.83	3.10

Tabella 5.2: Tempi di esecuzione dei threads con passo adattivo.

È importante sottolineare che il tempo di esecuzione del singolo thread con passo adattivo sia più che dimezzato rispetto al caso a passo fisso a parità di N_{coll} .

A valle di quanto appena descritto, si deduce che i tempi di calcolo per risolvere un fissato numero di urti è minore qualora si utilizzi il kernel con time-step variabile:

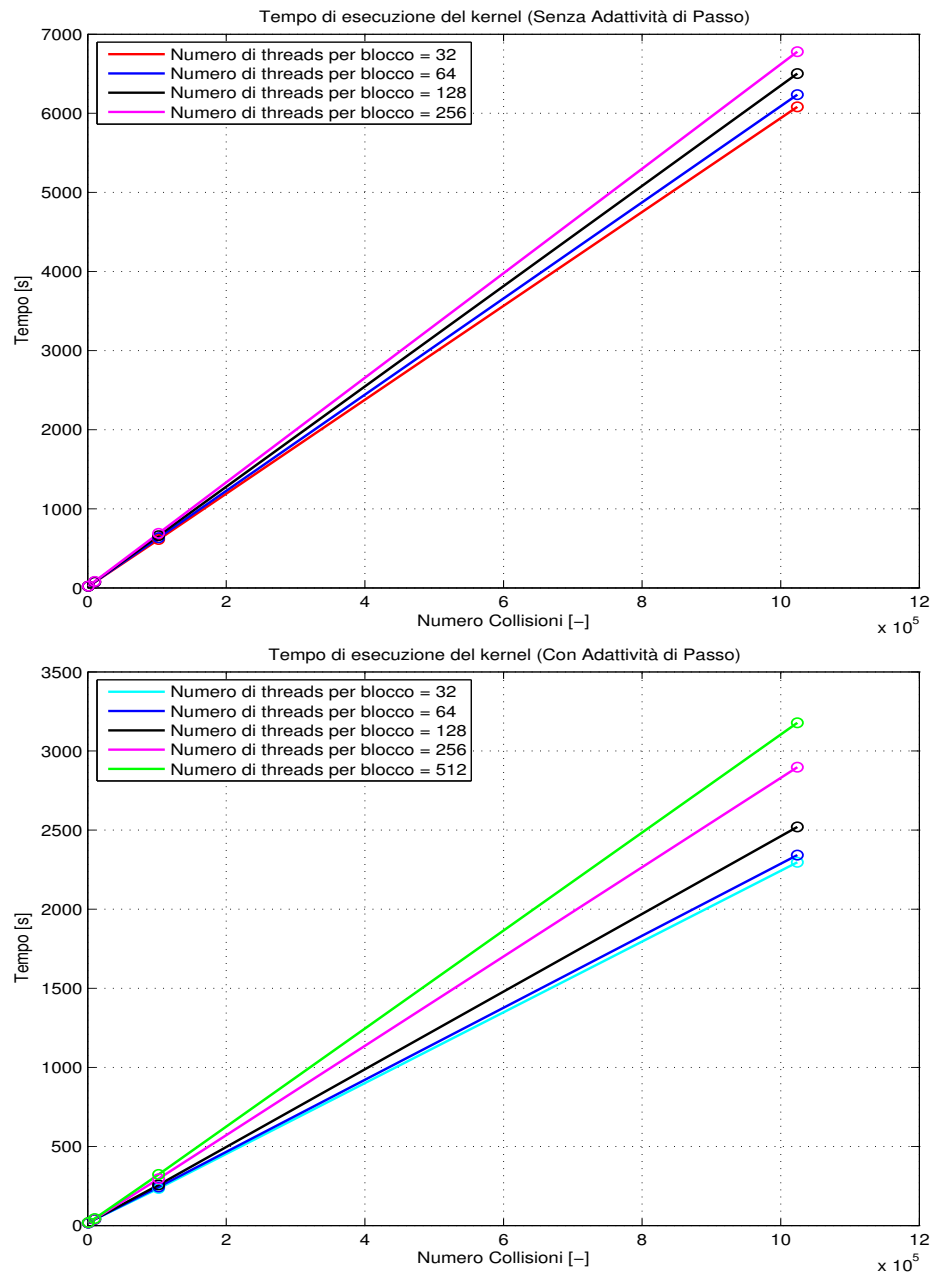


Figura 5.7: Tempo di calcolo totale del kernel.

Da entrambe le figure si osserva che la quantità di threads per blocco che minimizza i tempi di esecuzione del singolo thread corrisponde a 32 TxB . D'ora in avanti i valori di riferimento per quanto riguarda la GPU saranno associati a questa configurazione. Una volta raggiunta la saturazione, i tempi di calcolo sono lineari e possono quindi essere confrontati con il programma seriale. La CPU utilizzata per l'analisi è una Intel-Xeon avente le seguenti specifiche tecniche:

```

vendor_id: GenuineIntel
cpu family: 6
model: 45
model name: IntelR XeonR CPU E5-2630 0 @ 2.30GHz
stepping: 7
microcode: 0x70d
cpu MHz: 1200.000
cache size: 15360 KB
physical id: 1
siblings: 12
core id: 5
cpu cores: 6
apicid: 43
initial apicid: 43
fpu: yes
fpu_exception: yes
cpuid level: 13
wp: yes

```

Per misurare il tempo di esecuzione della singola collisione da parte della CPU, è stato sufficiente simulare un numero basso di urti poiché la durata della singola collisione è indipendente dal numero di urti prefissato. La stima dei tempi di calcolo per un valore di N_{coll} elevato è ottenibile quindi per mezzo di un'interpolazione lineare.

I valori ottenuti dalla simulazione sono riportati in tabella:

Tempo medio passo fisso [ms]	Tempo medio passo variabile [ms]
655.94	72.02

Tabella 5.3: Tempi medio della singola collisione via CPU.

Paragonando i risultati appena presentati con quelli riportati nelle tabelle 5.1 e 5.2 si nota immediatamente come i tempi di esecuzione da parte della CPU siano ben due ordini di grandezza più elevati rispetto a quelli ottenuti per mezzo della scheda grafica. A questo punto è possibile calcolare il valori di Speed-Up che determinano quanto la GPU sia più veloce rispetto all'approccio seriale; eseguendo il rapporto tra il tempo impiegato dalla CPU e dalla GPU per simulare la medesima quantità di collisioni si ottiene che:

Speed-Up passo fisso [-]	Speed-Up passo variabile [-]
106.42	31.82

Tabella 5.4: Valori di Speed-Up sia con passo fisso che con passo variabile.

L'andamento grafico dei risultati ottenuti è presentato in seguito:

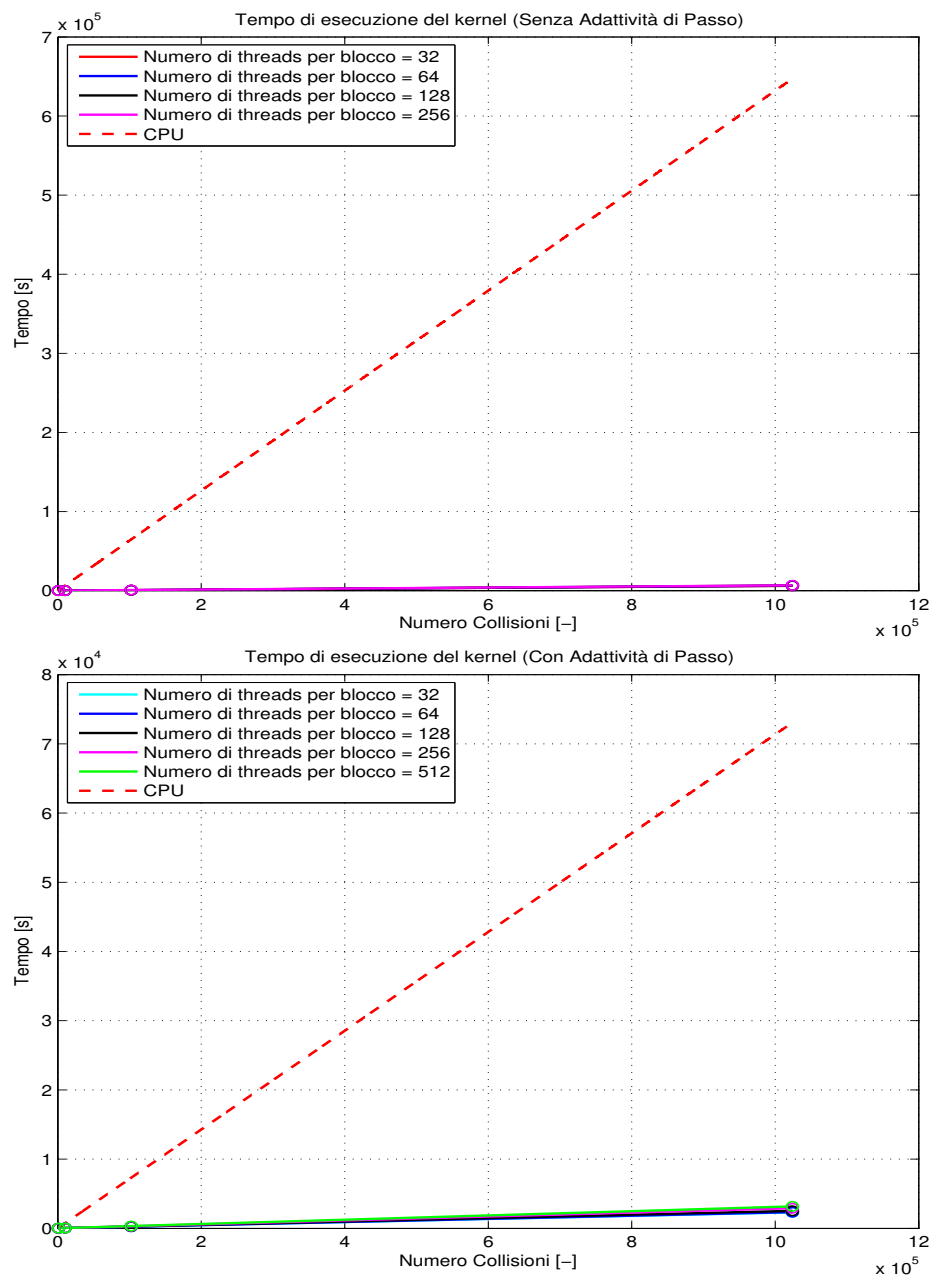


Figura 5.8: Confronto tra i tempi di esecuzione della CPU e della GPU.

5.7 Interpretazione dei risultati

I risultati presentati finora inducono a chiedersi come mai i valori di Speed-Up ottenuti nel caso di kernel con time-step fisso e con passo variabile siano distinti. L'utilizzo di un passo temporale fissato infatti risulta essere una scelta migliore in quanto è circa 3 volte più veloce rispetto alla versione dotata di adattività di passo. Di conseguenza si sta dimostrando per via indiretta che, nonostante il tempo di calcolo del singolo thread sia minore, il kernel adattivo non sta gestendo bene l'esecuzione dei thread i quali sono soggetti ad una condizione di divergenza; quest'ultima si verifica qualora i threads di un warp eseguano operazioni distinte tra loro limitando le performance del device. Nel problema in questione infatti i threads associati alle collisioni più veloci vengono eseguiti in maniera più rapida ma devono attendere che anche i conti relativi agli urti più lenti vengano terminati. Poiché gli streaming multiprocessor elaborano un solo warp alla volta, anche se un thread è stato processato, esso deve attendere che anche gli altri threads vengano eseguiti prima di lasciare il posto alle collisioni presenti nei warp successivi.

Diversamente, se il codice che simula la dinamica collisionale intermolecolare si basa sull'integrazione temporale a passo fisso, si ottiene uno Speed-Up maggiore. Tale risultato è giustificato dal fatto che i threads del generico warp compiono il medesimo numero di operazioni poiché esso è stato fissato a 40000. Ne consegue che il tempo di esecuzione del thread da parte dello SM è uguale per qualsiasi tipo di collisione. Non essendoci condizioni di divergenza, la GPU opera in condizioni ideali e la velocità di esecuzione è di gran lunga superiore rispetto ai risultati ottenuti con la CPU.

È importante infine sottolineare che nonostante lo Speed-Up ottenuto con la versione del codice a passo fisso sia maggiore rispetto alla versione a passo variabile, i tempi di esecuzione a parità di collisioni suggeriscono comunque l'utilizzo del codice dotato di time-step variabile.

Capitolo 6

Conclusioni

Il primo obiettivo di questo lavoro di tesi consisteva nel modificare un codice di dinamica molecolare¹ per il calcolo della viscosità dell'ossigeno O_2 . Per raggiungere tale scopo si è utilizzato un potenziale di interazione molecolare basato sui valori nominali dei coefficienti che lo costituiscono. Per quanto riguarda le condizioni iniziali del problema, si è implementata una subroutine che definisce secondo una distribuzione maxwelliana le velocità ed orientazione del sistema di molecole. Le simulazioni per il calcolo della viscosità η , effettuate per mezzo di un approccio di tipo DSMC, sono stati confrontati con i risultati sperimentali offerti dalla letteratura e si è potuta riscontrare un'accettabile corrispondenza. Le operazioni sono state eseguite per un range di temperature compreso tra $200K$ e $500K$.

Nella seconda parte della tesi si sono volute valutare le performance della funzione che si occupa di simulare la dinamica collisionale intermolecolare attuando un confronto tra i tempi di calcolo eseguiti su GPU e CPU. L'utilizzo della scheda grafica come strumento per la riduzione dei tempi di simulazione di un elevato numero di collisioni basandosi sul calcolo parallelo, ha confermato che l'utilizzo della GPU è vantaggioso rispetto all'approccio seriale. L'operazione è stata svolta per due diverse funzioni che si occupano di simulare la dinamica collisionale: uno caratterizzato da un time-step di integrazione temporale fissato, ed uno contraddistinto da un passo temporale di tipo adattivo; di conseguenza è stato possibile determinare due valori di Speed-Up (104 e 31). A partire da questi risultati si è identificata una condizione di divergenza dei threads ed una conseguente riduzione delle performance nel caso in cui il codice simuli le collisioni con un time-step variabile.

¹Codice precedentemente sviluppato e destinato alla simulazione del rilassamento rototraslazionale in un gas biatomico

6.1 Sviluppi futuri

Considerando che il potenziale di interazione molecolare è definito a partire dai valori nominali dei coefficienti che lo costituiscono, si potrebbe indagare come cambiano i valori della viscosità modificando i valori dei coefficienti per mezzo delle tolleranze riportate in figura 2.3. Inoltre, per ridurre il rumore statistico associato ai valori assunti da η bisognerebbe effettuare più volte le simulazioni per ogni valore di temperatura considerato.

Per quanto riguarda il miglioramento delle performance di calcolo per l'abbattimento di tempi di esecuzione, si propone una rielaborazione dei kernel che si occupano della simulazione della dinamica collisionale al fine di utilizzare al meglio le varie tipologie di memoria offerte dalla GPU. Sempre con l'intento di rendere il codice più efficiente sarebbe utile modificare la funzione dotata di passo di integrazione adattivo affinché ogni warp sia costituito da un set di collisioni simili tra loro; in questo modo si permetterebbe al device di processare lo stesso numero di calcoli per ogni thread appartenente ad un generico warp riducendo di conseguenza l'effetto di divergenza.

Bibliografia

- [1] Bartolomei M. Cappelletti D. Cavalli S. Vitores M.d.C Pirani F. Aquilanti V., Ascendi D. Molecular beam scattering of aligned oxygen molecules. the nature of the bond in the $o_2 - o_2$ dimer. *Journal of American Chemistry Society*, May 1999.
- [2] Chapman S. Cowling T.G. The Mathematical Theory of Non-Uniform Gases Cambridge Mathematical Library, Third Edition.
- [3] Ghiroldi G.P Bruno D., Frezzotti A. Oxygen transport properties estimation by ct-dsmc simulations. *AIP Conference Proceedings*, Volume 1628 Proceedings of the 29th International Symposium on Rarefied Gas Dynamics, January 2015.
- [4] Cercignani C. The Boltzmann Equation and Its Applications. Springer-Verlag, 1975.
- [5] Bird C.N. Molecular Gas Dynamics and the Direct Simulation of Gas Flows. Oxford at the Clarendon Press, 1994.
- [6] Hinshelwood C.N. The Kinetics of Chemical Change in Gaseous Systems. Oxford at the Clarendon Press, 1929.
- [7] Ferrigni D. Simulazione su GPU del rilassamento roto-traslazionale in un gas biatomico tramite calcolo di traiettorie classiche. Politecnico di Milano, 2012.
- [8] Smith E.B Maitland G.C. Critical reassessment of viscosity of 11 common gases. Journal of Chemical and Engineering Data, 1972.
- [9] NVIDIA. NVIDIA CUDA C Programming Guide. 2014.

