

Calcolo intensivo

Annalisa Massini

Introduzione

- Gli strumenti tradizionali della **scienza** e **dell'ingegneria** sono:
 - ♦ sviluppare teorie o progetti
 - ♦ eseguire esperimenti o costruire sistemi
- Ma può essere:
 - ♦ troppo **difficile** ----- costruire gallerie del vento
 - ♦ troppo **costoso** ----- crash testing
 - ♦ troppo **lento** ----- evoluzione di una galassia
 - ♦ troppo **pericoloso** --- farmaci, diff. sost. tossiche



Introduzione

- Lo strumento della **scienza computazionale**:
 - ◆ sistemi di **calcolo ad alte prestazioni** per la simulazione di fenomeni
- Il **calcolatore** rappresenta lo strumento essenziale dell'investigazione scientifica e ha la funzione sia di **microscopio** che di **telescopio** rispetto allo **spazio** e al **tempo**
- **Esempi**:
 - ◆ modellare le molecole in tutti i dettagli
 - ◆ tornare agli albori dell'universo e studiare la sua evoluzione
 - ◆ produrre previsioni del tempo



Introduzione

Strumenti come **acceleratori di particelle**, **telescopi**, **scanner**, ecc., che producono grandi quantità di dati, devono essere associati ad un **calcolatore** che possa elaborare i dati ottenuti, che tipicamente devono essere

- ridotti e trasformati
- rappresentati e visualizzati

Tali elaborazioni servono a

- comprendere il significato dei dati prodotti
- elaborare teorie
- verificare fenomeni di varia natura

Scienza computazionale

La **scienza computazionale**:

- ♦ è un campo multidisciplinare in crescita
- ♦ si basa sull'uso di risorse di calcolo avanzate

Si fonda su:

- ♦ sviluppo di **algoritmi** numerici e non numerici, **modelli**, **simulazioni** per risolvere problemi scientifici, ingegneristici e sociali
- ♦ sviluppo di **sistemi avanzati** hw, sw, di rete, di gestione dati per soddisfare la richiesta computazionale
- ♦ sviluppo delle **infrastrutture** di supporto alla soluzione di problemi sia scientifici che ingegneristici

Quando e perché usare simulazioni

■ Quando il problema è:

- ♦ Complesso
- ♦ Grande/piccolo
- ♦ Costoso
- ♦ Pericoloso

■ Per problemi che:

- ♦ Manipolano **grandi quantità** di dati
- ♦ Hanno **scale di valori** molto grandi/piccole rispetto alle distanze e al tempo (*cosmologia, molecole*)
- ♦ Modellare previsioni (*tempo, clima, ecc.*)

Simulazioni anche perchè

Le simulazioni permettono di provare diverse **opzioni** variando **parametri** in modo più **veloce, economico e sicuro** rispetto ai metodi tradizionali:

- ♦ Esempio: Crash testing → si possono variare i parametri di progetto ed esplorare le possibili soluzioni per ottenere caratteristiche ottimizzate

Simulazioni





Grand challenge problems

- Alla fine degli anni '80 l'Ufficio per le politiche scientifiche e tecnologiche (USA) individuava gli obiettivi per il finanziamento della ricerca per 5 anni sulle tematiche riguardanti *high-performance computing* e *communications* definendo i **Grand challenges**

"A Research and Development Strategy for High Performance Computing",
Executive Office of the President, Office of Science and Technology Policy,
November 20, 1987

- Un *Grand challenge* è un problema fondamentale della scienza o dell'ingegneria, con ampie applicazioni, la cui soluzione può essere trovata utilizzando **risorse di calcolo ad alte prestazioni** che saranno disponibili nel prossimo futuro.



Esempi di Grand challenges

1. Computational fluid dynamics for

- the design of hypersonic aircraft, efficient automobile bodies, and extremely quiet submarines
- weather forecasting for short and long term effects
- efficient recovery of oil, and for many other applications

2. Electronic structure calculations for the design of new materials such as

- chemical catalysts
- immunological agents
- superconductors



Esempi di Grand challenges

3. **Plasma dynamics for fusion energy technology and for safe and efficient military technology**
4. **Calculations to understand the fundamental nature of matter, including quantum chromodynamics and condensed matter theory**
5. **Symbolic computations including speech recognition**
 - computer vision
 - natural language understanding
 - automated reasoning
 - tools for design, manufacturing, and simulation of complex systems



Esempi di Grand challenges

Previsioni del tempo, del clima e cambiamenti globali:
capire il sistema accoppiato atmosfera-biosfera oceanica
in dettaglio per produrre previsione a lungo termine.

Le **applicazioni** includono comprensione delle dinamiche del biossido di carbonio nell'atmosfera, andamento dell'ozono e perturbazioni climatiche dovute al rilascio di sostanze chimiche o di calore.



Esempi di problemi dal HPCP '89

Scienza dei materiali: per la comprensione della struttura atomica dei materiali

Esempi sono:

- ♦ **Semiconduttori** come silicio e arseniuro di gallio usati in microelettronica → capire a fondo come funzionano e come cambiare le loro caratteristiche
- ♦ **Superconduttività** la scoperta della superconduttività ad alte temperature ha permesso lo sviluppo di strumenti ultrasensibili, nuovi dispositivi e tecnologie per la trasmissione efficiente di energia → capire come ottenere, stabilizzare e usare materiali superconduttori



Esempi di problemi dal HPCP '89

Biologia strutturale: esplorare i meccanismi di catalisi enzimatica, capire il riconoscimento di acidi nucleici da parte delle proteine, individuare i collegamenti tra anticorpi e antigeni e molto altro → **simulazioni intensive** di dinamica molecolare e **visualizzazione 3D** del moto molecolare

Studio di farmaci: simulazione della formazione di proteine e di molecole di RNA → nuovi farmaci

Genoma umano: il confronto di sequenze normali e patologiche → capire il genoma e l'origine di molte malattie



Esempio: simulazione effetto serra

- Il **riscaldamento globale** è stato oggetto di grande attenzione internazionale.
- Questo problema è stato studiato per mezzo di **simulazioni** al computer con lo scopo di capire come i cambiamenti di concentrazione di biossido di carbonio nell'atmosfera contribuiscano al riscaldamento globale attraverso l'effetto serra.
- Uno studio di questo tipo richiede la **modellazione** del clima su un **lungo periodo** di tempo.



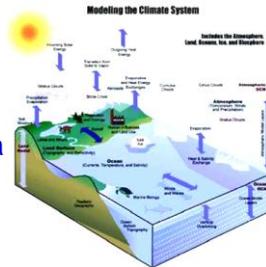
Esempio: simulazione effetto serra

- Il modello noto con il nome di **General Circulation Model, GCM**, è stato usato per studiare il riscaldamento causato dal raddoppiamento del biossido di carbonio su un periodo di 20 anni.
- Le simulazioni furono fatte (anni '90) su un **CRAY-1** con picchi di velocità di 200 Mflops ($2 \times 10^2 \times 10^6$ floating point operation/sec):
 - ◆ Ogni giorno simulato richiedeva 110 sec
 - ◆ Due simulazioni da 19 anni richiesero 400 ore di calcolo

Esempio: simulazione effetto serra

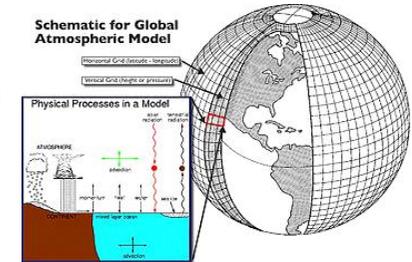
Gli effetti che il GCM cerca di modellare sono:

- L'atmosfera è un fluido → modellato da equazioni alle derivate parziali che governano il comportamento dei fluidi
- La soluzione al computer di queste equazioni viene ottenuta utilizzando il metodo delle differenze finite (le derivate rispetto allo spazio e al tempo sono approssimate da differenze)



Esempio: simulazione effetto serra

- Si utilizza una griglia 3D
- La griglia usata nella simulazione era composta di 2000 punti per coprire la superficie terrestre e 9 livelli di altitudine, per un totale di 18000 punti
- In ogni punto della griglia si hanno 9 variabili (velocità del vento, concentrazione di CO₂, temperatura, ecc).



- La velocità del computer è fondamentale!!

Esempio: simulazione effetto serra

- La soluzione del problema richiede un insieme di **condizioni iniziali** per avviare la simulazione → servono ad assegnare valori alle variabili in ogni punto della griglia e procedere all'aggiornamento delle variabili ad ogni intervallo di tempo, all'evolvere del tempo

Osservazione La griglia considerata è molto rada, infatti considerando che la terra ha un'estensione di $5,1 \times 10^8$ km² si ha un **punto di mesh** ogni $2,6 \times 10^5$ km², cioè sulla regione Spagna-Portogallo cadono 2 punti

Esempio: simulazione effetto serra

- Chiaramente quello che vogliamo è una **maggiore accuratezza**, cioè più punti sulla griglia
- Se **raddoppiamo** la densità di punti sulla superficie terrestre:
 - ◆ i punti di griglia nelle tre dimensioni aumentano di un fattore 8
 - ◆ da 400 ore si passa a più di **3000 ore** di calcolo, avendo sempre pochi punti su Spagna e Portogallo



Soluzioni e simulazioni

Il processo di risoluzione di un problema per mezzo di **simulazioni computazionali** richiede diversi passi:

1. Sviluppare un **modello matematico** espresso di solito tramite equazioni del fenomeno o del sistema fisico che ci interessa
2. Sviluppare gli **algoritmi** per risolvere numericamente le equazioni
3. **Implementare** gli algoritmi con opportuno linguaggio o ambiente sw



Soluzioni e simulazioni

4. Far girare i programmi su un **calcolatore** appropriato per simulare il processo
 5. Rappresentare i risultati calcolati usando una **visualizzazione** grafica opportuna che li renda comprensibili
 6. **Interpretare** e **validare** i risultati e ripetere i passi necessari.
- E' importante tenere presente che in questo procedimento **ogni passo** influenza ed è influenzato dagli altri.

Soluzioni e simulazioni

- I principi e i metodi della simulazione computazionale e del calcolo scientifico devono:
 - mantenere una adeguata **generalità**
 - tenere conto del **contesto** del problema
 - tenere conto dell'uso che si deve fare **dei risultati**
- Un problema matematico si dice **ben posto** se esiste una soluzione, è unica e dipende in modo continuo dai dati
- Questo vuol dire che un piccolo **cambiamento** dei dati non deve causare un cambiamento enorme o comunque sproporzionato della soluzione

Soluzioni e simulazioni

- Anche se per modelli matematici di sistemi fisici è fondamentale che i problemi siano ben posti, non sempre questa richiesta è **soddisfacibile**.
- Per **esempio** la deduzione della struttura interna di un sistema fisico dalla sola osservazione esterna, come nella **tomografia** o nella **sismologia**, porta spesso a problemi matematici mal posti perchè configurazioni interne molto diverse possono avere un aspetto esterno indistinguibile
- Anche quando un problema è ben posto la soluzione può essere molto sensibile alle perturbazioni dei dati, bisogna allora definire **misure di sensibilità** ai dati e progettare **algoritmi stabili**

Strategia generale

Nel definire e ricercare la **soluzione di un problema computazionale**, una strategia di base è:

- **rimpiazzare un problema difficile con uno più facile che abbia la stessa soluzione o una soluzione sufficientemente vicina**

Per fare ciò è necessario:

- Sostituire **spazi** di dimensione **infinita** con spazi di dimensione **finita**
- Sostituire **processi infiniti** con processi **finiti**, ad esempio sostituire integrali o serie infinite con somme finite oppure sostituire derivate con differenze finite

Strategia generale

- Sostituire equazioni **differenziali** con equazioni **algebriche**
- Sostituire **problemi non lineari** con problemi **lineari**
- Sostituire **problemi** di grado **alto** con problemi di grado più **basso**
- Sostituire **funzioni complicate** con funzioni **semplici**, come ad esempio polinomi
- Sostituire **matrici** generali con matrici più semplici

Ad ogni passo si deve verificare che la soluzione non cambi o cambi entro un certo **limite di tolleranza** rispetto alla soluzione vera

Esempio

- Per risolvere un sistema di **equazioni differenziali non lineari** possiamo:
 - ◆ sostituire il sistema di equazioni differenziali con un sistema di **equazioni algebriche**
 - ◆ sostituire il sistema algebrico non lineare con un **sistema lineare**
 - ◆ sostituire la matrice del sistema lineare con una **matrice** che abbia una soluzione facile da calcolare

Strategia generale

- Affinché questa strategia sia applicabile si deve avere
 - un problema o classe di problemi più semplici da risolvere
 - una trasformazione dal problema dato al problema alternativo che preservi la soluzione
- Se la soluzione del problema trasformato è un'approssimazione della soluzione vera, va stimata l'**accuratezza** e calcolata la **convergenza** alla soluzione vera
- Spesso l'accuratezza **può essere resa arbitrariamente buona** a costo di **computazioni** e memoria

High performance computing

- Negli ultimi anni abbiamo assistito a uno sviluppo eccezionale dei calcolatori.
- I microprocessori sono diventati più piccoli e più potenti.
- Il risultato è che il calcolo ad alta efficienza basato su microprocessori sta diventando la tecnologia di preferenza nell'affrontare importanti problemi nelle scienze e nell'ingegneria.

High Performance Computers

- **25 anni fa**
 - 1×10^6 Floating Point Ops/sec (Mflop/s)
 - » processori scalari
- **15 anni fa**
 - 1×10^9 Floating Point Ops/sec (Gflop/s)
 - » processori vettoriali, memoria condivisa
- **Pochi anni fa**
 - 1×10^{12} Floating Point Ops/sec (Tflop/s)
 - » parallelismo massivo, calcolo distribuito, message passing
- **oggi**
 - 1×10^{15} Floating Point Ops/sec (Pflop/s)
 - » Diffusione processori multicore (SMP node), estensione della precisione maggiore adattività e fault tolerance.

Strumenti: GpGpu

- **GPU, Graphics Processing Unit**, è il **microprocessore** per le schede video (per computer o console) ed è principalmente usato per il rendering di immagini 3D.
- **GPGPU** (General Purpose computation using GPU) è un settore emergente della ricerca che studia l'utilizzo dei processori grafici per accelerare applicazioni che si prestano ad essere fortemente parallelizzate.

31

Strumenti: GpGpu

Esempi di applicazioni su cui le Gpu sono utilizzate:

- sequenziamento del genoma e allineamento di proteine
- dinamica molecolare
- simulazioni finanziarie

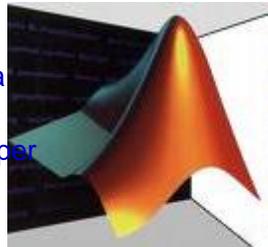


32



Strumenti: MATLAB

- **MATLAB** (abbreviazione di **Matrix Laboratory**) consente di manipolare **matrici**, visualizzare **funzioni e dati**, implementare **algoritmi**, creare **interfacce utente** e interfacciarsi con altri programmi
 - specializzato nel calcolo numerico
 - **funzioni grafiche** in 2D e 3D per la rappresentazione grafica dei dati
 - moltissime **funzioni matematiche** per algebra lineare, statistica, analisi di Fourier, filtraggio, ottimizzazione e integrazione numerica



33