

Introduzione al Calcolo Scientifico

Corso di Calcolo Numerico, a.a. 2010/2011

Francesca Mazzia

Dipartimento di Matematica
Università di Bari

Calcolo Scientifico

Insieme degli strumenti, delle tecniche e delle teorie necessarie per risolvere con il computer problemi della scienza e della tecnica.

- nasce negli anni '40 (modellizzazione di problemi balistici e di armi nucleari nella Seconda Guerra Mondiale);
- si sviluppa negli anni '70 e '80 (problemi dell'industria, es. progettazione di aerei);
- si consolida negli anni '90 (problemi di Informatica, Fisica, Chimica, Ingegneria, Economia, ...).

CALCOLO NUMERICO

Risoluzione di problemi matematici al calcolatore.

Richiede:

- una conoscenza dello strumento computazionale da usare.
- una conoscenza del problema da risolvere
- la costruzione di un algoritmo che risolva il problema con una accuratezza desiderata ed entro i limiti delle risorse (tempo, memoria, ...) disponibili.

Ambiente computazionale

- hardware: PC o supercomputer (computer vettoriali, computer paralleli);
- sistemi operativi e linguaggi:
 - UNIX, LINUX, WINDOWS, ...
 - C, C++, Java, Fortran95, ...
- data management: costruzione di database contenenti tutte le informazioni rilevanti per un particolare progetto applicativo;
- visualizzazione: rappresentazione grafica finalizzata ad una veloce comprensione del calcolo;

Esempi di Applicazioni Informatiche

- Grafica
- Elaborazione di segnali ed immagini
- Motori di ricerca
- Data mining
- Machine learning

Linguaggi per il Calcolo Scientifico

Il linguaggio di programmazione per la comunità scientifica è stato per molti anni il Fortran (introdotto negli anni '50). Ora anche altri linguaggi, specialmente il C e il C++ e Java, sono usati per il Calcolo Scientifico. la disponibilità di un linguaggio che può essere usato su una molteplicità di macchine determina la portabilità del software da macchina a macchina con pochissimi cambiamenti nelle performance del codice.

Problem Solving Environments

Un PSE è un ambiente di alto livello che contiene tutto il software necessario per risolvere una determinata classe di problemi, fra cui

- metodi avanzati per la soluzione;
- selezione automatica o semiautomatica dei metodi di soluzione;
- modi per incorporare facilmente nuovi metodi di soluzione.

Gli utenti possono usarlo senza avere una conoscenza specialistica del computer.

PSE per il calcolo numerico

- Matlab;
- Scilab;
- Octave;
- R;
- Python;

MATLAB

MATLAB (MATrix LABoratory) : PSE per il calcolo scientifico ad alte prestazioni e la visualizzazione, integra analisi numerica, calcolo con matrici, grafica. È usato in una varietà di applicazioni e rappresenta un potente strumento di calcolo, visualizzazione e programmazione:

- elaborazione di immagini e segnali;
- progettazione di sistemi di controllo;
- ingegneria finanziaria;
- ricerca in medicina;

<http://www.mathworks.com/>

SCILAB

È un ambiente di calcolo scientifico simile al Matlab, sviluppato dallo Scilab group (INRIA-Rocquencourt Metalau Project, Cergrene ENPC). Scilab è un software free scaricabile dal sito:

<http://www.scilab.org/>

Scilab è disponibile per la maggior parte delle piattaforme Unix, per PC con Linux e Windows.

OCTAVE

GNU OCTAVE è un ambiente di calcolo scientifico simile al Matlab, sviluppato da John W. Eaton e molti altri.

OCTAVE è un software free scaricabile dal sito

<http://www.octave.org/>

Octave è disponibile per la maggior parte delle piattaforme Unix, per PC con Linux e Windows.

R

R (R development Core Team, 2007) è una versione open-source del linguaggio S.

È più conosciuto come un linguaggio per analisi statistica dei dati. Di recente si sta evolvendo come software per il calcolo scientifico.

R è un software free scaricabile dal sito

<http://www.r-project.org>

R è disponibile per la maggior parte delle piattaforme Unix, per PC con Linux e Windows.

Python

Python è un linguaggio di programmazione ad alto livello orientato agli oggetti, che utilizzato insieme a un IDE può essere considerato come un PSE. L'implementazione di Python è stata iniziata nel 1989 da Guido van Rossum del CWI (Netherlands) e il linguaggio è in continua evoluzione. L'ultima release la 3.2 (febbraio 2011), ma le versioni 2.x continuano ad essere utilizzati perchè le versioni 3.x non sono compatibili con le precedenti versioni.

Il linguaggio può essere utilizzato per sviluppare applicazioni distribuite, scripting, e per il calcolo scientifico. Python può essere paragonato a Tcl, Perl, Java, JavaScript e Visual Basic.

Python è un software free scaricabile dal sito

<http://www.python.org/>

ed è disponibile per la maggior parte delle piattaforme, Unix, per PC con Linux e Windows.

Scopo del corso

Costruire un algoritmo efficiente per il calcolo scientifico può essere una sfida complessa e difficile.

Lo scopo di questo corso è farvi conoscere metodi e strumenti di analisi di base da poter utilizzare per risolvere problemi più complessi che potreste incontrare in futuro.

Molti degli algoritmi che faremo sono già implementati nei PSE.

Il loro studio ci fornisce le basi per risolvere problemi più complessi che non sono stati ancora risolti.

Sorgenti di errori

- Problema reale \rightarrow Modello Matematico
semplificazione del modello:
si suppongono trascurabili alcune grandezze fisiche
- Modello Matematico \rightarrow Metodo Numerico
approssimazione nel metodo risolutivo (es. un procedimento infinito approssimato mediante un procedimento finito: errore di troncamento);
- Metodo Numerico \rightarrow programma
errori di arrotondamento: i dati numerici elaborati e i risultati delle operazioni eseguite vengono arrotondati (si opera con aritmetica finita, errore di round-off)

Il processo di risoluzione numerica

- sviluppo di metodi numerici che tengano conto:
 - ▶ della natura del problema;
 - ▶ delle risorse hardware e software;
- sviluppo di un algoritmo o un set di istruzioni che descrivono come risolvere un problema con un calcolatore;
- esame dei diversi aspetti che riguardano la scrittura, la comprensione e la valutazione di algoritmi ed implementazione in un determinato ambiente di calcolo;
- formulazione di problemi test;
- fase di testing e misura dell'efficienza.

ottimizzazione della complessità di tempo e di spazio
studio accurato dell'algoritmo e della fase di implementazione attraverso:

- la ricerca di tecniche che consentono l'effettiva minimizzazione dei calcoli
- una corretta pianificazione delle aree di memoria da utilizzare

Alcuni criteri per un buon codice

- affidabilità
- robustezza
- portabilità
- leggibilità
- buona documentazione
- ampia fase di testing

Libri di testo

- F. Mazzia, D. Trigiante, Laboratorio di Programmazione e Calcolo, Pitagora Editrice, Bologna, 1992.
- P. Amodio, D. Trigiante, Elementi di Calcolo Numerico, Pitagora Editrice, Bologna, 1993.
- James F. Epperson, Introduzione all'analisi numerica, teoria, metodi, algoritmi. McGraw-Hill, Milano, 2003
- Uri. M. Asher and Chen Grief, A first course on Numerical Methods.