Anna Delle Piane (5565836), Marco Zoratti (5562866).

Relazione Laboratorio 2 ALAN

Introduzione

Tutto il codice è contenuto nell'unico file *labo2.cpp* ed è suddiviso in 3 parti:

- Funzioni ausiliarie
- Esercizi
- Main

Per rappresentare le matrici è stata usata la struttura dati *Vector* per via della sua scalabilità, efficienza e semplicità d'uso.

Queste ultime vengono inizializzate nel main del programma e passate come argomento di funzione ad ogni esercizio.

Nota: All'inizio dell'esecuzione del programma verrà chiesto all'utente se si vuole stampare la matrice triangolare, questo perché la sua stampa è molto invasiva e di difficile comprensione per via della sua grande dimensione. Se la risposta è affermativa, si consiglia di visionare i dati su uno schermo sufficientemente grande e con il terminale a schermo intero.

Funzioni Ausiliarie

Al fine di migliorare la scalabilità e leggibilità del codice sono state definite 6 funzioni ausiliarie:

- ***defMatricePascal()***: funzione che ritorna la matrice di Pascal 10x10;
- ***defMatriceTriangolare()***: funzione che ritorna la matrice triangolare di dimensione variabile in base alla prima e alla seconda cifra della matricola utilizzata;
- ***fattoriale(double n)***: ritorna il fattoriale di n;
- ***printVettore(vector<float> v)***: stampa in console il vettore passato come argomento;
- ***printMatrice(vector<vector<float>> A)***: stampa in console la matrice passata come argomento;
- ***stampaMatriceTriangolare()***: funzione booleana che indica se la matrice triangolare dev'essere stampata o meno.

Esercizio 1

Per calcolare la norma infinito di una matrice è stata creata la funzione ***normaInfinito(vector<vector<float>> matrice)*** che calcola e restituisce la massima norma dei vettori che compongono la matrice quadrata passata come parametro.

La funzione ***es1(…)*** richiama *normaInfinito* sulle 4 matrici definite nel *main* e stampa il risultato.

Risultati Ottenuti

Norma ∞ della matrice 1	17
Norma ∞ della matrice 2	9
Norma ∞ della matrice di Pascal	24310.1
Norma ∞ della matrice triangolare	4

I risultati sono stati confrontati con i valori restituiti dalla funzione *norm(X, Inf)* di *MatLab* e risultano corretti.

Esercizio 2

Per la realizzazione dell'esercizio 2 sono state create diverse funzioni ausiliarie per aumentare la leggibilità del codice e il riuso dello stesso.

In particolare, la funzione ***es2_singolaMatrice(vector<vector<float>> A)*** calcola il termine noto, la riduzione di Gauss e la sostituzione all'indietro tramite le seguenti funzioni:

- Per il calcolo del termine noto: ***calcolaB(vector<vector<float>> A)***;
  Dato che x = {1, ..., 1}, per trovare il termine noto dell'equazione avente come coefficienti i valori alla riga i della matrice A, la funzione somma questi valori tra loro e restituisce il vettore b.
- Per la riduzione di Gauss: ***Gauss(vector<vector<float>> A, vector<float> b)***;
  La funzione calcola A ridotta e B ridotta tramite il pivoting parziale per poi ritornare la matrice formata dal merge delle due matrici calcolate.
- Per la sostituzione all'indietro: ***sostituzioneIndietro(vector<vector<float>> A)***;
  Questa funzione dev'essere chiamata successivamente a *Gauss(…)* in quanto la matrice A presa come input deve essere formata dal merge di due matrici ridotte.
  Successivamente esegue la sostituzione all'indietro (si noti il ciclo for che decrementa il parametro i) e ritorna il vettore X con le soluzioni del sistema.

Nota: tutte le funzioni utilizzate in questo esercizio contengono variabili numeriche per specificare la dimensione della matrice, questo perché si è voluto implementare il punto facoltativo dell'esercizio che permette di eseguire i calcoli su matrici di dimensione arbitraria.

Risultati ottenuti

Vettore b:

La seguente tabella contiene i valori del vettore b di ogni matrice, ottenuto tramite la risoluzione del sistema:

$$\mathrm{b} = \mathrm{Ax'} \ \ dove \ \ \mathrm{x'} = (1, \dots, 1)^{\mathrm{T}}$$

matrice 1	[3 5 9 -12]
matrice 2	[6 5 1 4]
matrice di Pascal	[4.71786 10 46 165.5 495.333 1287.25 3003.2 6435.17 12870.1 24310.1]
matrice triangolare	[1 0 0 … 0 0 1]

Vettore x:

La seguente tabella riporta i risultati della risoluzione del sistema $b = Ax$ ottenuti tramite l'eliminazione gaussiana e la tecnica del pivoting parziale.

Si noti che quasi tutti i valori sono "vicini" alla soluzione attesa x = (1, 1, … 1)^T:

matrice 1	[1 1 1 1]
matrice 2	[1 0.999999 1 1]
matrice di Pascal	[1.00905 0.995903 1.00393 0.952716 1.04771 1.04867 0.86985 1.10644 0.959522 1.0061]
matrice triangolare	[1 1 1 1 1 1 1 1 0.999999 0.999999 … 0.999999 0.999998 … 0.999998 0.999999 0.999999 0.999999 0.999999 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Il fatto che i risultati siano *vicini* e non *uguali* alla soluzione attesa è dovuto alla precisione applicata dalla macchina: la consegna impone che i numeri siano rappresentati in singola precisione e quindi con un'accuratezza minore rispetto alla doppia precisione.

Infatti sostituendo nel codice tutti i valori *float* in *double* (in VSCode si può fare molto velocemente premendo *ctrl+shift+f*) si può notare che risultati ottenuti siano molto più vicini se non uguali alla soluzione data.

Questo comportamento probabilmente è dovuto ad approssimazioni eseguite dal computer durante i calcoli che aumentano l'errore sui dati e la perdita d'informazione.

Esercizio 3

Come per gli esercizi precedenti, anche per il terzo esercizio del laboratorio sono state sviluppate delle funzioni ausiliarie richiamate dalla funzione principale ***es3_singolaMatrice(vector<vector<float>> A)***:

- ***termNoti_BP(vector<vector<float>> A)***;
  Questa funzione *perturba* il vettore dei termini noti della matrice A.
  Per farlo, esegue *calcolaB(A)* ed *normaInfinitoVettore(b)* e modifica i valori dentro il vettore ottenuto dalla prima funzione come da consegna, ovvero:
  $$\delta b = ||b||_\infty \cdot (-0.01, 0.01, -0.01, \dots, 0.01)^t$$
  dove δb è il vettore perturbato.
- ***Gauss(…)*** e ***sostituzioneIndietro(…)***:
  entrambe le funzioni sono gia state specificate nell'esercizio 2 e vengono riutilizzate all'interno di questa funzione per calcolare la riduzione Gaussiana e il suo vettore delle soluzioni con il vettore dei termini noti perturbato.

Risultati ottenuti

La seguente tabella riporta i risultati perturbati come da consegna:

matrice 1	[0.96 1.02017 0.993719 0.989256]
matrice 2	[0.778 1.24 1.048 0.724]
matrice di Pascal	[-14994.2 -3146.64 -45543.5 517577 -1.401e+06 1.97694e+06 -1.66486e+06 847328 -241985 29917.7]
matrice triangolare	[0.995065 1.00013 0.995195 1.00026 0.995324 1.00039 0.995454 1.00052 0.995584 1.00065 0.995714 1.00078 0.995844 1.00091 0.995973 1.00104 0.996103 1.00117 0.996233 1.0013 0.996363 1.00143 0.996493 1.00156 0.996622 1.00169 0.996752 1.00182 0.996882 1.00195 0.997012 1.00208 0.997142 1.00221 0.997272 1.00234 0.997401 1.00247 0.997531 1.0026 0.99766 1.00273 0.99779 1.00285 0.99792 1.00298 0.998049 1.00311 0.998179 1.00324 0.99831 1.00337 0.99844 1.0035 0.99857 1.00364 0.9987 1.00377 0.998831 1.0039 0.998961 1.00403 0.999091 1.00416 0.999221 1.00429 0.999351 1.00442 0.99948 1.00455 0.99961 1.00468 0.99974 1.00481 0.99987 1.00494]

La maggiorazione dell'errore in output è dovuto al condizionamento della matrice che si basa sul valore della norma infinito della stessa.

Si noti che la matrice di Pascal è quella con i risultati che si discostano maggiormente dalla soluzione attesa: questo è dovuto alla sua norma che è nettamente maggiore delle norme delle altre matrici come dimostrato nell'esercizio 1.

Ambiente di sviluppo, compilazione e altro

L'intero laboratorio è stato sviluppato tramite *Visual Studio Code* su macchina virtuale *Linux* (*WSL2, Ubuntu 22.04*) utilizzando il compilatore *g++* aggiornato alla versione *11.4.0* .

Per compilare, utilizzare il comando:

g++ -Wall labo2.cpp