

## LABORATORIO DI PROGRAMMAZIONE E CALCOLO

Canale 1 (A-K) A.A. 2024/2025

Foglio di esercizi n. 11

1. Scrivere un programma C++ strutturato in funzioni che acquisisce da tastiera il numero reale  $x \in [-1, 1]$ , stampa il valore  $e^x$  e calcola - con il criterio di arresto a posteriori discusso a lezione - il grado necessario a garantire che l'approssimazione fornita dal polinomio di Taylor di  $e^{|x|}$ , con  $x_0 = 0$ , abbia almeno precisione  $10^{-10}$ ; successivamente stampa sul video l'approssimazione di  $e^x$  ottenuta facendo uso del polinomio di Taylor costruito per l'approssimazione di  $e^{|x|}$  (se  $x < 0$ , stampandone quindi il reciproco) e il suo grado effettivo. Testare il programma con  $x = -0.9$  e  $x = 0.1$ .
2. Scrivere un programma C++ strutturato in funzioni che, definita la costante  $T$ ,
  - a) calcola il grado necessario a garantire che l'approssimazione fornita dal polinomio di Taylor di  $\cos(T\pi)$ , con  $x_0 = 0$ , sia accurata entro la tolleranza  $\tau$  acquisita da tastiera facendo uso del criterio di arresto a posteriori discusso a lezione; successivamente stampa sul video il valore  $\cos(T\pi)$ , il grado effettivo  $n_0$  del polinomio costruito e l'approssimazione calcolata.
  - b) stampa su video sia l'approssimazione di  $\cos(T\pi)$  realizzata facendo uso del polinomio interpolatore determinato dalla tavola della funzione coseno relativa agli  $n_0 + 1$  nodi (\*) che dividono  $[0, \pi/2]$  in intervalli di uguale ampiezza che l'errore.Confrontare l'output del programma nei casi  $T = 0.1$  e  $T = 0.4$ , avendo inserito in entrambi i casi prima  $\tau = 10^{-3}$  e poi  $\tau = 10^{-10}$ .

```
void nodi_equidistanti(double a, double b, double x[], int n)
{
    double h=(b-a)/n;
    x[0]=a;
    x[n]=b;
    for(int i=1;i<n;i++)
        x[i]=x[i-1]+h;
    return;
}
```

```
double polinomio_interpolatore (double x, double nodi[], int n)
{
    double elle_i, pol_x=0;
    for (int i=0;i<=n;i++)
    {
        elle_i=1;
        for(int k=0;k<=n;k++)
            if(i!=k)
                elle_i*=(x-nodi[k])/(nodi[i]-nodi[k]);
        pol_x+=funzione(nodi[i])*elle_i;
    }
    return pol_x;
}
```

3. Scrivere un programma C++ strutturato in funzioni che, definita la costante  $T$ , stampa su video l'approssimazione di  $\cos(T\pi)$  realizzata facendo uso del polinomio interpolatore determinato dalla tavola della funzione coseno relativa agli  $n_0 + 1$  nodi di Chebyshev (\*) associati all'intervallo  $[0, \pi/2]$ . Il grado  $n_0$ , che dipende da  $T$  ed è stato calcolato nell'esercizio precedente per due diverse tolleranze, deve essere inserito da tastiera.

Confrontare l'output del programma con l'output del programma precedente nei casi  $T = 0.1$  e  $T = 0.4$ , avendo inserito in entrambi i casi i due gradi  $n_0$  effettivi.

(\*) Ricordiamo che, per  $n > 0$ , nell'intervallo  $[a, b]$ , gli  $n + 1$  nodi equidistanti associati all'intervallo sono i seguenti:

$$x_i^{(eq)} = a + i \frac{b - a}{n}, \quad i = 0 : n,$$

e gli  $n + 1$  nodi di Chebyshev associati all'intervallo sono i seguenti:

$$x_i^{(C)} = \frac{a + b}{2} + \frac{b - a}{2} \cos \left( \frac{2i + 1}{2n + 2} \pi \right), \quad i = 0 : n.$$