**Esame di Calcolo Numerico**

**21/02/2019**

*Portare alla propria postazione solo la penna.*

*Per accedere correttamente ai pc eseguire le istruzioni riportare nel foglio allegato.*

*Con l'avverbio "analiticamente" si richiede di effettuare i calcoli solo con carta e penna, senza comandi Matlab.*
*Per ogni esercizio che richiede esecuzione di comandi Matlab creare file script con tutte le istruzioni programmate per risolverlo (non utilizzare la Command Window).*
*Salvare i file contenenti le figure, corredate da tutti i dati necessari per la loro interpretazione (legende e/o axis label e/o titolo...).*
*Riportare e salvare eventuali tabelle in file di testo.*

*Tempo a disposizione per lo svolgimento: **2 ore**.*

**1.** Calcolare "analiticamente" la soluzione della seguente espressione

$$(3.1 \cdot 10^{200}) \cdot (2 \cdot 10^{150}) \cdot (1 \cdot 10^{-100}) \,.$$

Confrontare con il risultato ottenuto scrivendo l'espressione in Matlab.
Calcolare "analiticamente" la soluzione della seguente espressione

$$(3.1 \cdot 10^{200}) \cdot (1 \cdot 10^{-100}) \cdot (2 \cdot 10^{150}) \,.$$

Confrontare con il risultato ottenuto scrivendo l'espressione in Matlab.
Commentare i risultati ottenuti in relazione alla variabile Matlab **realmax**.

**2.** Risoluzione di sistemi lineari.
2a) Definizione di metodi diretti e metodi iterativi.
2b) Descrivere l'algoritmo diretto di risoluzione di un sistema linare di ordine $n$ con matrice diagonale, calcolando il costo computazionale.
2c) Risolvere "analiticamente" il sistema lineare $Ax = b$ con

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{pmatrix} \qquad b = \begin{pmatrix} 59/7 \\ 14 \\ 12 \end{pmatrix}$$

applicando il metodo di sostituzione all'indietro.

**3.** 3a) Data una funzione $f : (a, b) \to \mathbb{R}$ non lineare, descrivere l'algoritmo delle tangenti per la ricerca di una radice $\alpha \in (a, b)$, ossia tale che $f(\alpha) = 0$.
3b) Fare il grafico della funzione

$$f(s) = s^8 - 2 \quad \text{con} \quad s \in (0.5, 2) .$$

3c) Implementare l'algoritmo delle tangenti in Matlab e applicarlo alla funzione $f$ con dato iniziale $s_0 = 2$ e test di arresto basato sul controllo dell'incremento con tolleranza pari a $10^{-10}$.
3d) Calcolare un'approssimazione dell'unica radice reale di $f$ in $(0.5, 2)$ con funzioni della libreria Matlab. Rappresentarla sullo stesso grafico.
3e) Sovrapporre al grafico della funzione la successione di approssimanti della radice ottenuta con l'algoritmo delle tangenti.

Implementare il metodo di bisezione trascrivendo il codice sottostante:

```
===========================================================
function [xvect,nit,xdif]=bisect(a,b,toll,nmax,fun)
%metodo di bisezione
%INPUT
%a=estremo di sinistra intervallo di esistenza della radice
%b=estremo di destra intervallo di esistenza della radice
%toll=tolleranza per l'ampiezza del k-esimo sottointervallo
%nmax=numero massimo di iterazioni
%fun=funzione di cui si cerca uno zero
%OUTPUT
%xvect=successione di valori approssimanti la radice
%nit=numero di iterazioni
err=abs(b-a);
nit=1;
xvect=b;
fx=[];
xdif=[];
while nit<nmax && err>toll
    nit=nit+1;
    c=(a+b)/2;
    x=c;
    fc=fun(x);
    xvect=[xvect;x];
    fx=[fx;fc];
    x=a;
    if fc*fun(x)>0
        a=c;
    else
        b=c;
    end
    err=abs(xvect(nit)-xvect(nit-1));
    xdif=[xdif;err];
end
===========================================================
```

3f) Dopo quante iterazioni si arresta il metodo di Newton? E il metodo di bisezione? Confrontare l'approssimazione finale dei due metodi con l'approssimazione ottenuta al punto 3c) e commentare.
3g) Applicare il metodo di Newton con dato iniziale $s_0 = 0.5$. Cosa succede? Commentare.