Corso di Laurea in Informatica – Università di Ferrara
Calcolo Numerico
Prova scritta e pratica dell'11/07/2018 – A.A. 2017–2018

Esercizio 1

1) (T) (2 punti) Determinare la rappresentazione del numero decimale $\alpha = -145.07 \cdot 10^{-2}$ secondo il formato ANSI standard IEEE in semplice precisione.

2) (M) (2 punti) Realizzare un M-script file Matlab che acquisisca dall'utente un numero reale positivo x_0 e, usando la M-function `ruffiniHorner` in allegato, calcoli e stampi i valori $p(x_0)$, $p'(x_0)$ e $p''(x_0)$ per il polinomio $p(x) = 7x^9 - 4x^6 + (x-2)(x+1)^3 - 5$. Lo script preveda un opportuno controllo sull'input e controlli i risultati anche con la M-function `polyval`. Provare lo script con il valore $x_0 = 1.7$ e riportare **nel foglio delle risposte** i risultati con almeno due cifre decimali.

Esercizio 2

1) (T) (2 punti) Valutare l'errore inerente, l'errore algoritmico, l'indice di condizionamento e l'indice algoritmico nel calcolo dell'espressione:
$$\varphi(x) = \ln(x+1)\sqrt{\cos(x+1)+4}$$
dove essa è definita. Esistono valori di x per i quali il problema tende ad essere mal condizionato? Motivare la risposta.

2) (M) (3 punti) Realizzare una M-function che accetti in input un parametro scalare $\mathtt{k} \in [1, 10]$, due parametri scalari $\mathtt{a}, \mathtt{b} \in \mathbb{R}$ con $\mathtt{a} < \mathtt{b}$ ed un numero naturale $\mathtt{N} > 0$.

   La function generi un vettore `xx` che suddivida in modo uniforme l'intervallo $[\mathtt{a}, \mathtt{b}]$ in `N` parti uguali e calcoli poi il valore della funzione
   $$f_k(x) = k(1-x)^2 \mathrm{e}^{-x^2}$$
   sui punti della griglia `xx` senza usare cicli, restituendo tali valori di f nel parametro `y` di uscita. Prima di effettuare i calcoli, la function esegua il controllo dei parametri in input, fornendo all'utente opportuni messaggi di errore in caso di valori non conformi alle specifiche.

   Scrivere infine un M-script di test della function che la provi nell'intervallo $[\mathtt{a}, \mathtt{b}] = [-3, 3]$, suddiviso in 50 parti uguali, all'interno di un ciclo sul parametro `k` che assume valori fra 1 e 10 con passo 10^{-1}. Ad ogni passo del ciclo, dopo la chiamata della funzione, lo script disegni la funzione $f_k(x)$ in $[\mathtt{a}, \mathtt{b}]$ limitando l'asse y del grafico all'intervallo $[-1, 20]$ e attenda 5 centesimi di secondo prima di passare alla successiva iterazione. Dotare il grafico di opportune etichette degli assi e di un titolo che di volta in volta indichi anche il valore k in uso.

Appendice: codici forniti

```matlab
function [r, q] = ruffiniHorner(p, a)
% RUFFINIHORNER - Schema di Ruffini-Horner
%  Calcola il valore di un polinomio p(x) nel punto x = a e i coefficienti
%  del polinomio quoziente q(x) = p(x) / (x - a)
% SYNOPSIS
%   [r, q] = ruffiniHorner(p, a)
% INPUT
%   p (double array) - Vettore dei coefficienti del polinomio, ordinati
%                      da quello di grado piu' alto a quello di grado zero
%   a (double)       - Punto (numero reale) in cui calcolare il polinomio
% OUTPUT
%   r (double)       - Valore del polinomio nel punto x = a
%   q (double array) - Vettore dei coefficienti del polinomio quoziente
%                      q(x) = p(x) / (x - a)
```

```
%
  r = [];
  if ( isempty(p) )
      q = [];
      warning('Il vettore p dei coefficienti e'' vuoto');
      return
  elseif ( isempty(a) )
      q = p;
      warning('Il punto ''a'' in cui valutare il polinomio e'' vuoto');
      return
  else
      n = numel(p) - 1;  % grado del polinomio
      q = zeros(n, 1);
      q(1) = p(1);
      for i = 2 : n+1
          q(i) = q(i-1)*a + p(i);
      end
      r = q(n+1);
      q = q(1:n);
  end
end % fine della function ruffiniHorner
```

Soluzioni e commenti

Nel seguito sono riportate le soluzioni degli esercizi del compito, includendo in generale solo quei commenti che risultino fondamentali per la comprensione della soluzione. Qualora uno stesso esercizio possa eventualmente essere risolto in più modi, in alcuni casi si riportano le descrizioni anche di qualche modo alternativo. Si invitano gli studenti a contattare il docente per eventuali dubbi o chiarimenti.

Nella trascrizione delle soluzioni è stata posta la massima cura. Tuttavia, nel caso si rilevino sviste e/o errori di battitura, si invitano gli studenti a comunicarle via e-mail al docente.

Soluzione esercizio 1

Teoria

$$
\begin{array}{c|c}
1 & /2 \\
\hline
0 & 1
\end{array}
\qquad
\begin{array}{r|l}
0.4507 & \times 2 \\
\hline
0.9014 & 0 \\
0.8028 & 1 \\
0.6056 & 1 \\
0.2112 & 1 \\
0.4224 & 0 \\
0.8448 & 0 \\
0.6896 & 1 \\
0.3792 & 1 \\
0.7584 & 0 \\
0.5168 & 1 \\
0.0336 & 1 \\
0.0672 & 0 \\
0.1344 & 0 \\
0.2688 & 0 \\
0.5376 & 0 \\
0.0752 & 1 \\
0.1504 & 0 \\
0.3008 & 0 \\
0.6016 & 0 \\
0.2032 & 1 \\
0.4064 & 0 \\
0.8128 & 0 \\
0.6256 & 1 \\
\vdots & \vdots
\end{array}
$$

$$
\begin{aligned}
(1)_{10} &= (1)_2 \\
(0.4507)_{10} &= (0.01110011011000010001001\ldots)_2 \\
\alpha &= (-1.4507)_{10} \\
&= (-1.01110011011000010001001\ldots)_2 \\
&= (-1.01110011011000010001001\ldots)_2 \cdot 2^0
\end{aligned}
$$

$$
\begin{aligned}
\alpha < 0 &\Rightarrow s = 1 \\
p &= (0)_2 = (0)_{10} \\
&\Rightarrow \widetilde{p} = p + bias = 0 + 127 = 127 \\
&\qquad = (01111111)_2 \\
m &= (1.01110011011000010001001)_2 \\
&\Rightarrow \widetilde{m} = 01110011011000010001001
\end{aligned}
$$

(attenzione: <u>troncamento</u> della mantissa alla 24-esima cifra binaria)

$$
\mathrm{IEEE}_{32}(\alpha) = 1\,\underbrace{01111111}_{\widetilde{p}}\,\underbrace{01110011011000010001001}_{\widetilde{m}}
$$

Matlab

Contenuto dell'M-script file `esercizio1.m`:

```
% Cognome Nome
% Matricola
close all
clear all
%----------------------------
%  esercizio 1 - 11/07/2018
%----------------------------
disp('Esecizio 1');
p = [7 0 0 -4 0 1 1 -3 -5 -7];
x0 = input('\nInserire un numero reale (x0 > 0): x0 = ');
while (x0 <= 0)
  x0 = input('\nValore non positivo. Reinserire: x0 = ');
end
[r1, q1] = ruffiniHorner( p,  x0 );
[r2, q2] = ruffiniHorner( q1, x0 );
[r3, q3] = ruffiniHorner( q2, x0 );
fprintf('\nValutazioni nel punto x0 = %g: ', x0);
fprintf('\n\tpolinomio: p(%g) = %g  (con polyval: %g)', ...
        x0, r1, polyval(p,x0));
fprintf('\n\tderivata prima: p''(%g) = %g (con polyval: %g)', ...
        x0, r2, polyval(q1,x0));
fprintf('\n\tderivata seconda: p''''(%g) = %g (con polyval: %g)\n\n', ...
        x0, 2*r3, 2*polyval(q2,x0));
```

L'esecuzione dello script con l'inserimento del valore 1.7 per `x0` fornisce i seguenti risultati:

$$
p(1.7) = 722.66 \qquad p'(1.7) = 4067.08 \qquad p''(1.7) = 19717.7
$$

confermati dal confronto con i rispettivi output di `polyval`.