Elaborato di

Calcolo Numerico

Anno Accademico 2016/2017

Gabriele Puliti - 5300140 - *gabriele.puliti@stud.unifi.it*
Luca Passaretta - -

August 30, 2017

Capitoli

1 Capitolo 1

1.1 Esercizio 1

Sapendo che il metodo iterativo è convergente a x^* allora per definizione si ha:

$$\lim_{k \to +\infty} x_k = x^*$$

inoltre per definizione di Φ si calcola il limite:

$$\lim_{k \to +\infty} \Phi(x_k) = \lim_{k \to +\infty} x_{k+1} = x^*$$

infine ipotizzando che la funzione Φ sia uniformemente continua è possibile calcolare il limite:

$$\lim_{k \to +\infty} \Phi(x_k) = \Phi(\lim_{k \to +\infty} x_k) = \Phi(x^*)$$

dai due limiti si ha la tesi:

$$\Phi(x^*) = x^*$$

1.2 Esercizio 2

Dal momento che le variabili intere di 2 byte in Fortran vengono gestite in Modulo e Segno, la variabile `numero`, inizializzata con

```
integer*2 numero
```

varia tra $-32768 \leq$ `numero` ≤ 32767 ($-2^{15} \leq$ `numero` $\leq 2^{15} - 1$). Durante la terza iterazione del primo ciclo for si arriva al valore massimo rappresentabile tramite gli interi a 2 byte, alla quarta iterazione si avrà quindi la somma del `numero` in modulo e segno:

$$(32767)_{10} + (1)_{10} = (0111111111111111)_{2,MS} + (0000000000000001)_{2,MS} =$$

$$= (1000000000000000)_{2,MS} = (-32768)_{10}$$

Nel secondo ciclo for durante la quinta iterazione, al `numero` viene sottratto 1:

$$(-32768)_{10} - (1)_{10} = (1000000000000000)_{2,MS} - (0000000000000001)_{2,MS} =$$

$$= (0111111111111111)_{2,MS} = (32767)_{10}$$

Da cui si spiega l'output del codice.

1.3 Esercizio 3

Per definizione si ha che la precisione di macchina u per arrotondamento e' data da:

$$u = \frac{1}{2} b^{1-m}$$

Se $b = 8, m = 5$ si ha:

$$u = \frac{1}{2} \cdot 8^{1-5} = \frac{1}{2} \cdot 8^{-4} = 1,2 \cdot 10^{-4}$$

1.4 Esercizio 4

Il codice seguente:

```
format long e;

h=zeros(12,1);
f=zeros(12,1);

for i=1:12
    h(i)= power(10,-i);
end

for j=1:12
    f(j)=lim(0,v(j));
end

f

function p=lim(x,y)
    p=(exp(x+y) - exp(x))/y;
end
```

restituisce questo risultato (assumendo che $f(x) = e^x$ e $x_0 = 0$):

h	$\Psi_h(0)$
10^{-1}	1.051709180756477e+00
10^{-2}	1.005016708416795e+00
10^{-3}	1.000500166708385e+00
10^{-4}	1.000050001667141e+00
10^{-5}	1.000005000006965e+00
10^{-6}	1.000000499962184e+00
10^{-7}	1.000000049433680e+00
10^{-8}	9.999999939225290e-01
10^{-9}	1.000000082740371e+00
10^{-10}	1.000000082740371e+00
10^{-11}	1.000000082740371e+00
10^{-12}	1.000088900582341e+00

si può notare che al diminuire del valore h, la funzione $\Psi_h(0)$ approssima sempre meglio il valore $f'(0)$, come si può vedere dal plot 1.

1.5 Esercizio 5

Per dimostrare le due uguaglianze è necessario sviluppare in serie di taylor $f(x)$ fino al secondo ordine:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2 f''(x_0)}{2} + O((x - x_0)^2)$$

Da cui possiamo sostituire con i valori di $x = x + h$ e $x = x - h$:

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2 f''(x_0)}{2} + O(h^2)$$

$$f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2 f''(x_0)}{2} + O(h^2)$$

Andando a sostituire questi valori si ottiene, nel primo caso:

$$\frac{f(x_0 + h) - f(x_0 + h)}{2h} =$$

$$= \frac{(f(x_0) + hf'(x_0) + \frac{h^2 f''(x_0)}{2} + O(h^2)) - (f(x_0) - hf'(x_0) + \frac{h^2 f''(x_0)}{2} + O(h^2))}{2h} =$$

$$= \frac{2hf'(x_0) + O(h^2)}{2h} = f'(x_0) + O(h^2)$$

nel secondo caso:

$$\frac{f(x_0+h)-2f(x_0)-f(x_0+h)}{h^2}=$$

$$=\frac{f(x_0)+hf'(x_0)+\frac{h^2f''(x_0)}{2}+O(h^2)-2f(x_0)+f(x_0)-hf'(x_0)+\frac{h^2f''(x_0)}{2}+O(h^2)}{h^2}=$$

$$=\frac{h^2f''(x_0)+O(h^2)}{h^2}=f''(x_0)+O(h^2)$$

Abbiamo quindi dimostrato che:

$$\frac{f(x_0+h)-f(x_0+h)}{2h}=f'(x_0)+O(h^2)$$

$$\frac{f(x_0+h)-2f(x_0)-f(x_0+h)}{h^2}=f''(x_0)+O(h^2)$$

1.6 Esercizio 6

Il codice MatLab, indicando con x=x_n e r=ϵ:

```
format longEng
format compact

conv=sqrt(2);
x=[2,1.5];
r=[x(1)-conv,x(2)-conv];

for i= 2:7
    x(i+1) = (x(i)*x(i-1)+2)/(x(i)+x(i-1))
end

for i=3:8
    r(i)=x(i)-conv;
end

x
r
```

restituisce i valori:

n	x_n	ϵ
0	2.00000000000000e+000	585.786437626905e-003
1	1.50000000000000e+000	85.7864376269049e-003
2	1.42857142857143e+000	14.3578661983335e-003
3	1.41463414634146e+000	420.583968367971e-006
4	1.41421568627451e+000	2.12390141496321e-006
5	1.41421356268887e+000	315.774073555986e-012
6	1.41421356237310e+000	0.00000000000000e+000
7	1.41421356237310e+000	0.00000000000000e+000

I valori indicano che per valori di n superiori a 5 l'errore, indicato con ϵ, è dell'ordine di 10^{-12}.

1.7 Esercizio 7

Sapendo che la rappresentazione del numero è stata fatta usando l'arrotondamento allora la precisione di macchina si calcola:

$$u = \frac{b^{1-m}}{2}$$

che sappiamo essere pari a:

$$u \approx 4.66 \cdot 10^{-10}$$

dato che stiamo cercando il numero di cifre binarie allora si deve avere $b = 2$, è possibile quindi ricavarsi m:

$$m = 1 - log_2(4.66 \cdot 10^{-10}) \approx 31.99$$

possiamo quindi affermare che servono 32 cifre dedicate alla mantissa per rappresentare il numero con precisione macchina $4.66 \cdot 10^{-10}$.

1.8 Esercizio 8

Sapendo che la mantissa in decimale è calcolabile tramite la funzione:

- $m = 1 - log_{10}u$ (troncamento)
- $m = 1 - log_{10}(2 \cdot u)$ (arrotondamento)

e che la precisione di macchina assuma un valore accettabilemente piccolo in modo tale che il $log_{10}u \approx 1$, cioè $0 <= u < 1$, allora è possibile scrivere:

- $m = 1 - log_{10}u \approx -log_{10}u$ (troncamento)
- $m = 1 - log_{10}2u = 1 - log_{10}2 - log_{10}u \approx -log_{10}u$ (arrotondamento)

1.9 Esercizio 9

```
x=0;
delta = 1/10;
while x~=1,x=x+delta , end
```

Il valore di delta è uguale a $\frac{1}{10}$, la rappresentazione binaria di questo numero però non è esatta. Si tratta di una rappresentazione periodica e quindi, in decimale, sarà circa 0.0999, prendendo quindi $delta \approx 0.9$ vedremo che per $i = 10$ $x \approx 0.999$, mentre per $i = 11$ $x \approx 1,0989$. Per questo motivo la condizione x=1 non si avvererà mai e il programma non terminerà.

1.10 Esercizio10

All'interno della radice può presentarsi un problema di overflow dato che la somma dei due quadrati potrebbe essere molto grande, tanto grande da poter superare il limite massimo rappresentabile dalla macchina:

$$realmax = (1 - b^{-m}) \cdot b^{b^s - \nu}$$

Per risolvere questo problema è necessario prendere il massimo valore tra le due variabili:

$$m = max\{|x|, |y|\}$$

E moltiplicare e dividere per questo valore:

$$\sqrt{x^2 + y^2} = m \cdot \frac{\sqrt{x^2 + y^2}}{m} = m \cdot \sqrt{\frac{x^2 + y^2}{m^2}} = m \cdot \sqrt{\left(\frac{x}{m}\right)^2 + \left(\frac{y}{m}\right)^2}$$

In questo modo si eviterà il problema di overflow, il problema è ben condizionato dato che potenza e radice sono ben condizionate e grazie alla modifica proposta.

1.11 Esercizio11

2 Grafici

2.1 Esercizio 4

Figure 1: Esercizio 1.4