Calcolo Numerico - Corso A: Laboratorio Lezione 2

Gianna Del Corso <gianna.delcorso@unipi.it>

5 Marzo 2021

Quantità di esercizi: in questa dispensa ci sono *più esercizi* di quanti uno studente medio riesce a farne durante una lezione di laboratorio, specialmente tenendo conto anche degli esercizi facoltativi. Questo è perché sono pensate per "tenere impegnati" per tutta la lezione anche quegli studenti che già hanno un solido background.
Quindi fate gli esercizi che riuscite, partendo da quelli *non* segnati come facoltativi, e non preoccupatevi se non li finite tutti! Questa dispensa è il risultato del contributo di varie persone che hanno proposto, elaborato e raffinato i vari esercizi.

1 Analisi dell'errore

Esercizio 1. Si studi il condizionamento del calcolo della seguente funzione $f(x) = \frac{x-1}{x}$. Si studi poi la stabilità dei due seguenti algoritmi per il calcolo di $f(x)$,

$$g_1(x) = \frac{x-1}{x}, \quad g_2(x) = 1 - 1/x.$$

Esercizio 2. Si consideri il calcolo di $f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i$ con $x_i > 0$ e se ne valuti il condizionamento. Si valuti poi la stabilità del calcolo di f secondo il seguente algoritmo:

```
s=x(1)
for k=2:n
s=s+x(k)
end
```

2 Funzioni e accumulatori

```
function f=fact(n);
% calcola il fattoriale di n
```

```
% n dev'essere un intero
f=1;
% f fa da accumulatore: parte da 1,
% a ogni passo, lo moltiplico per k
for k=1:n
  f=f*k;
end
% ora f vale n!
end
```

Va scritto in un file chiamato `fact.m` e messo *nella cartella da cui abbiamo lanciato Matlab*; poi possiamo lanciarlo:

```
>> fact(10)
ans = 3628800
```

Esercizio 3. Scrivete una funzione `pow(x,n)` che calcoli x^n, per $n \geq 1$.

3 Calcolo dell'esponenziale

La formula di Taylor permette di esprimere una funzione come un polinomio con infiniti termini, e troncando in modo opportuno questa serie è possibile approssimare una funzione con un polinomio.

In particolare, data una funzione $f : (a, b]) \to \mathbb{R}$, $f \in C^n[(a, b)]$, cioè derivabile n-volte con derivata n-esima continua Preso un punto $x_0 \in (a, b)$, abbiamo che

$$f(x) = f(x_0) + f'(x_0)(x - x_0)| + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + R_n(x)$$

dove $R_n(x)$ è detto resto di Taylor e nella sua forma di Lagrange è dato da

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1},$$

sotto l'ipotesi che $f^{(n+1)}(x)$ esista per ogni $x \in (a, b), x \neq x_0$, e con $|\xi - x_0| < |x - x_0|$.

Si può semplicemente vedere che il polinomio di Taylor per $x_0 = 0$, arrestato al termine n-esimo della funzione esponenziale è

$$1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

da cui possiamo scrivere la seguente funzione per approssimare il valore dell'esponenziale in un punto x.

```
function a=myexp(x,n)
% calcola exp(x) con la serie di Taylor troncata all'n-esimo termine
  a=1; %accumulatore
  for k=1:n
```

```
    a=a+pow(x,k)/fact(k);
  end
end
```

Qualche esperimento su quanti termini servono per approssimare bene — confrontiamo con la funzione `exp(x)` di Matlab, che calcola l'esponenziale meglio di noi.

```
>> myexp(1,5)
ans = 2.7167
>> myexp(10,50)
ans = 2.2026e+04
>> exp(10)
ans = 2.2026e+04
>> format long
>> myexp(10,50)
ans = 22026.4657948067
>> exp(10)
ans = 22026.4657948067
```

Due problemi:

1. Inaccurato: prova `myexp(-20,500)`
2. Lento: con n termini, il numero di operazioni da eseguire cresce come n^2 (perché?)

Risolviamo il problema 2. introducendo un altro accumulatore:

```
function a=myexp2(x,n)
  %calcola e^x con Taylor troncato
  %ma usa solo O(n) operazioni
  t=1; %accumulatore che contiene il termine generico della sommatoria
  a=1; %accumulatore che contiene le somme parziali
  for k=1:n
    t=t*x/k;
    a=a+t;
  end
end
```

Ora va meglio:

```
>> myexp(-20,500)
ans = NaN
>> myexp2(-20,500)
ans = 5.62188447213042e-09
```

Cosa succedeva?

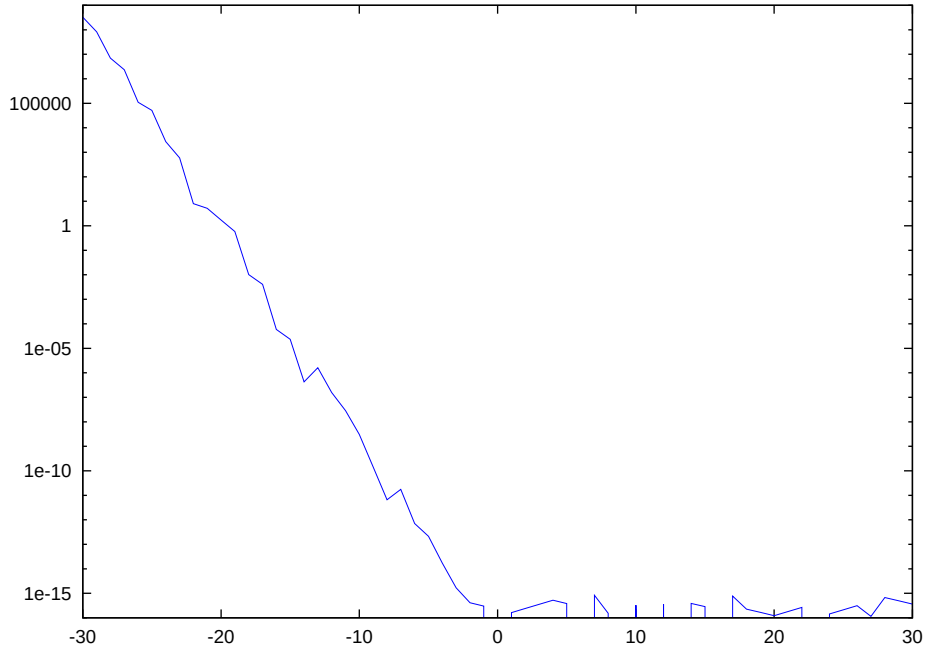
```
>> fact(500)
ans = Inf
```

```
>> pow(-20,500)
ans = Inf
>> Inf/Inf
ans = NaN
```

Ci sono ancora pesanti perdite di accuratezza sui numeri negativi:

```
>> myexp2(-30,500)
ans = -3.06681235635622e-05
```

Un esponenziale dovrebbe sempre essere positivo... Ci sono pesanti errori di cancellazione nella formula che abbiamo usato (perché?)



Errore relativo: $\frac{|\text{myexp2}(x)-e^x|}{e^x}$

La soluzione: cambiare algoritmo e sceglierne uno che non porti a errori di cancellazione che sono dovuti al fatto che per $x < 0$ la serie esponenziale è a segni alterni.

```
>> exp(-30)
ans = 9.3576e-14
>> myexp2(-30,500)
ans = -3.0668e-05
>> 1/myexp2(30,500)
ans = 9.3576e-14
>> format long
>> exp(-30)
```

```
ans = 9.35762296884017e-14
>> 1/myexp2(30,500)
ans = 9.35762296884017e-14
```

Esercizio 4. Scrivere una funzione `myexp(x)` che controlla se x è negativo o positivo, e a seconda del segno calcola l'esponenziale come $1/e^{-x}$ oppure e^x con la serie di Taylor troncata a $n = 500$.

Esercizio 5 (facoltativo). Si studi il condizionamento del problema del calcolo di $f(x) = x^n$ e si valuti la stabilità del calcolo di $f(x)$ secondo i due algoritmi:

$$z_1 = x, z_i = x * z_{i-1} \quad i = 2, \ldots, n$$

e $g(x) = \exp(n \log(x))$ supponendo che la funzione esponenziale e logaritmico siano implementate da due funzioni di libreria EXP e LOG che introducono un errore totale limitato dalla precisione di macchina.