Esercizio

Sappiamo che

$$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n .$$

È sensato approssimare il valore del numero 'e' calcolando $\left(1+\frac{1}{n}\right)^n$ con n molto elevato?

Utilizzare ad esempio $n = 10^5, 10^{10}, 10^{20}$.

Motivare il comportamento "anomalo" (dal punto di vista di un analista matematico) di Matlab.

Svolgimento

Valutiamo e con la funzione esponenziale:

```
>> e=exp(1)
e =
     2.718281828459046e+00
```

Prendiamo $n = 10^5$:

```
>> n=1e5; ee=(1+1/n)^n
   ee=
      2.718268237192297e+00
```

poco accurato: ee differisce da e dalla quinta cifra decimale in poi.
Prendiamo $n = 10^{10}$

```
>> n=1e10; ee=(1+1/n)^n
   ee=
     2.718282053234788e+00
```

poco accurato: ee differisce da e dalla sesta cifra decimale in poi.
Prendiamo $n = 10^{20}$:

```
>> n=1.e20; ee=(1+1/n)^n
   ee=
     1
```

COME MAI?

Valutiamo i termini $a_n = \left(1+\frac{1}{n}\right)^n$ per diversi n

Scrivere una function Matlab che, dato in input un intero $M > 0$,

1. valuti la successione $a_n = \left(1+\frac{1}{n}\right)^n$ per $n = 1, 10, 10^2, 10^3, \ldots, 10^M$ e memorizzi i valori n nel vettore `nn` e i valori a_n nel vettore `an`;
2. rappresenti graficamente gli elementi calcolati della succesione

Suggerimenti:

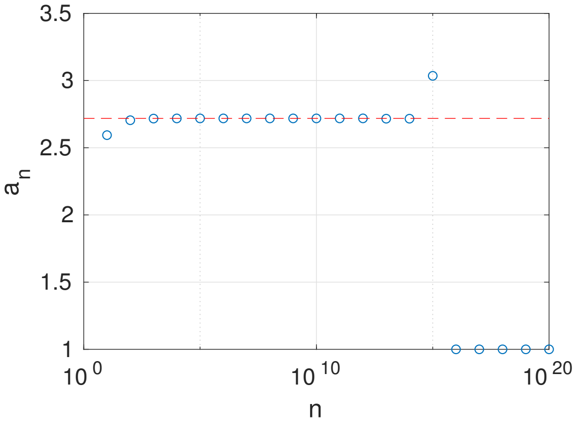
Per costruire il vettore `nn` si può utilizzare il comando `logspace`:

```
nn=logspace(1,M,M)
```

Per la rappresentazione grafica, poichè la scala delle ascisse è molto più estesa di quella delle ordinate, al posto del comando `plot` utilizzare il comando `semilogx`.

La sintassi di chiamata di `plot` e `semilogx` è identica.

Output grafico:



Perchè da un certo n in poi, $a_n = 1$??

Analisi dei risultati

Per $n \leq 10^{14}$ i valori di a_n sono prossimi a e, poi abbiamo un valore maggiore di 3 (in corrispondenza di $n = 10^{15}$) e poi per $n \geq 10^{16}$ si ha $a_n = 1$.
Se $n \geq 10^{16}$, allora

$$\frac{1}{n} \leq 10^{-16} < \epsilon_M$$

e

$$1 + \frac{1}{n} = 1 \quad \text{per la macchina.}$$

Di conseguenza anche

$$\left(1 + \frac{1}{n}\right)^n = 1 \quad \text{per la macchina.}$$

Esercizio (autovalori, rango, determinante)

Scaricare il file
`https://paola-gervasio.unibs.it/CS/MATLAB/matriceB.mat`
In esso è memorizzata una matrice B.
Scrivere uno script matlab che:
- carichi in memoria il contenuto del file `matriceB.mat`
- calcoli, con il comando `eig` di matlab, gli autovalori della matrice B memorizzata nel file
- calcoli, con il comando `det` di matlab, il determinante della matrice B
- calcoli, con il comando `rank` di matlab, il rango della matrice B.

Svolgimento

0. Pulire workspace con il comando `clear`

1. Un file .mat contiene dati prodotti in una sessione precedente di Matlab e opportunamente salvati. Il comando per caricare in memoria il contenuto del file `nomefile.mat` è:

```
load nomefile
```

In workspace si ha ora la matrice `B`.

2. Per determinare le dimensioni della matrice:

```
[n,m]=size(B)
```

3. Per calcolare gli autovalori della matrice:

```
[v]=eig(B)
```

4. Per calcolare il determinante della matrice:

```
d=det(B)
```

5. Per calcolare il rango della matrice:

```
r=rank(B)
```

Si osserva che, pur essendo tutti gli autovalori di B inclusi nell'intervallo $[10^{-6}, 10^{-2}]$, ed essendo massimo il rango della matrice (pari alla dimensione della matrice), il determinante calcolato risulta nullo.

Spiegare il perchè di questa contraddizione.

Risposta

Sappiamo che $det(B) = \lambda_1 \cdot \lambda_2 \ldots \lambda_{100}$.
Abbiamo $\lambda_i \neq 0$ e $det(B) = 0$

Per $n = 1, \ldots, 100$, calcolo e stampo $p_n = \prod_{k=1}^{n} \lambda_k$.

```
v=eig(B);
p=v(1);
  for n=2:100
  p=p*v(n);
  fprintf('prodotto dei primi  %d eig = %13.6e \n',n, p)
end
....
product of first  2 eigenval =  9.111628e-05
product of first  10 eigenval =  1.519911e-22
product of first  70 eigenval = 2.656088e-238
product of first  86 eigenval = 2.104720e-320
product of first  87 eigenval =  0.000000e+00
```

$p_{86} <$ `realmin`??
$p_{87} = 0$??

Propagazione degli errori di arrotondamento

Esercizio (calcolo π)

Scrivere una function matlab che, dato in input un intero positivo N,

1. calcoli i valori f_n con $n = 2, \ldots, N$ della successione così definita:

$$\begin{cases} f_2 = 2 \\ f_{n+1} = 2^{n-0.5}\sqrt{1-\sqrt{1-4^{1-n}f_n^2}}, \quad n = 2, 3, \ldots. \end{cases}$$

2. calcoli gli errori relativi $e_n = |f_n - \pi|/\pi$, per $n = 2, \ldots, N$;
3. disegni su un grafico i punti (n, f_n) per $n = 2, \ldots, N$;
4. disegni su un altro grafico i punti (n, e_n) per $n = 2, \ldots, N$;

Commentare i risultati ottenuti, sapendo che è stato dimostrato che $\lim_{n\to\infty} f_n = \pi$.

Propagazione degli errori di arrotondamento

Esercizio (calcolo di un integrale)

Si vuole calcolare $I_n = \displaystyle\int_0^1 \frac{x^n}{x+5}dx$ con l'algoritmo:

$$\begin{cases} I_0 = \log(6/5) = \displaystyle\int_0^1 \frac{x^0}{x+5}dx & \\ I_n = \frac{1}{n} - 5I_{n-1} & n \geq 1 \end{cases}$$

Si scriva una function matlab che generi la successione I_n per $n \leq N$ con N dato in input e si studi la stabilità dell'algoritmo rispetto alla propagazione degli errori di arrotondamento.

Svolgimento.

Studio teorico

La successione degli integrali $I_0,\ I_1,\ ....,I_n,....$ ha valori reali positivi ed è infinitesima, cioè $I_n \longrightarrow 0$ per $n \to \infty$. Infatti, poiché

$$\frac{1}{6} < \frac{1}{x+5} < \frac{1}{5} \qquad \forall x \in [0,1],$$

abbiamo

$$\begin{array}{lccccc} & \displaystyle\frac{1}{6}\int_0^1 x^n dx & < I_n = & \displaystyle\int_0^1 \frac{x^n}{x+5}dx & < & \displaystyle\frac{1}{5}\int_0^1 x^n dx \\ n \to \infty & \downarrow & & & & \downarrow \\ & 0 & & & & 0 \end{array}$$

quindi, per il secondo teorema del confronto, anche $I_n \to 0$ per $n \to \infty$.

Vediamo se le stesse proprietà valgono anche numericamente

Scrivere una function MATLAB che costruisca e rappresenti graficamente la successione I_n, con $n = 0, ..., N$.

Input: N

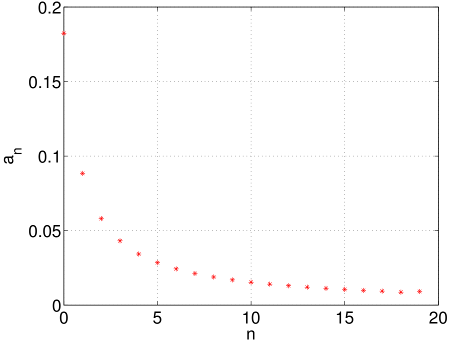
Ouput: un vettore contenente la successione $\{I_n\}$ degli integrali, con $n = 0, ..., N$.

N.B. In matlab, l'indice di vettore deve essere strettamente positivo.

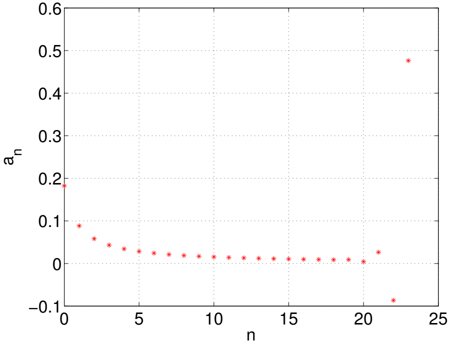
Se `an` è il vettore che contiene gli integrali calcolati, si può definire:

`an(1)`$=I_0$, ... `an(n)`$=I_{n-1}$

L'output grafico con `N=19` è:



Poi con `N=23`



La successione calcolata numericamente esplode per $n \to \infty$.

Facciamo i conti:

Considero l'errore di arrotondamento sul dato iniziale $I_0 = \log(6/5)$ e suppongo, per semplicità, che ci sia una propagazione solo di questo errore.

Si può dimostrare che dalla disuguaglianza $\dfrac{|x - fl_t(x)|}{|x|} \leq \dfrac{1}{2}\epsilon_M$ discende la seguente proprietà:

$$\boxed{fl_t(x) = x(1+\delta_0) \qquad \forall \delta_0 \leq \epsilon_M/2}$$

$$\tilde{I}_0 = fl_t(I_0) = I_0(1+\delta_0) \qquad \forall \delta_0 \leq \epsilon_M/2$$

$$I_1 = 1 - 5I_0, \qquad \tilde{I}_1 = 1 - 5\tilde{I}_0 = 1 - 5I_0(1+\delta_0) = (1 - 5I_0) - 5I_0\delta_0 = I_1 - 5I_0\delta_0$$

$$I_2 = 1/2 - 5I_1, \qquad \tilde{I}_2 = 1/2 - 5\tilde{I}_1 = 1/2 - 5(I_1 - 5I_0\delta_0) = (1/2 - 5I_1) + 25I_0\delta_0 = I_2 + (-5)^2 I_0\delta_0$$

$$\ldots.$$

Al passo generico n abbiamo:

$$\tilde{I}_n = I_n + (-5)^n I_0 \delta_0$$

$\delta_0 \leq \mathrm{u} = 1.11 \cdot 10^{-16}$,

per $n = 23$, $(-5)^{23} \simeq -1.1921\mathrm{e}+16$

quindi $(-5)^{23} I_0 \delta_0 \simeq -1.3232$

ovvero dopo soli 23 passi l'errore iniziale di arrotondamento si è amplificato fino ad arrivare all'ordine dell'unità.