Stabilità : radici secondo grado

Dato $x^2 + 2px - q$, con $\sqrt{p^2+q} \geq 0$ eseguiamo un primo algoritmo Matlab che valuta la radice via:

$$y = -p + \sqrt{p^2+q}. \tag{1}$$

- $\sqrt{p^2+q} \geq 0$ implica radici reali.

Valutiamo radice con un secondo algoritmo stabile via razionalizzazione di (1):

$$\begin{aligned} y &= -p + \sqrt{p^2+q} = \frac{(-p+\sqrt{p^2+q})(p+\sqrt{p^2+q})}{(p+\sqrt{p^2+q})} \\ &= \frac{q}{(p+\sqrt{p^2+q})} \end{aligned} \tag{2}$$

Stabilità : radici secondo grado

Dato $x^2 + 2\,px - q$, con $\sqrt{p^2+q} \geq 0$ eseguiamo un primo algoritmo Matlab che valuta la radice via:

$$y = -p + \sqrt{p^2+q}. \tag{1}$$

- $\sqrt{p^2+q} \geq 0$ implica radici reali.
- Potenzialmente instabile per $p \gg q$ a causa della sottrazione tra p e $\sqrt{p^2+q}$ (cancellazione).

Valutiamo radice con un secondo algoritmo stabile via razionalizzazione di (1):

$$\begin{aligned} y &= -p + \sqrt{p^2+q} = \frac{(-p+\sqrt{p^2+q})(p+\sqrt{p^2+q})}{(p+\sqrt{p^2+q})} \\ &= \frac{q}{(p+\sqrt{p^2+q})} \end{aligned} \tag{2}$$

Codice complessita': algoritmo 1

Salviamo il seguente codice in `radicesecgrado.m`.

```
p=1000; q=0.018000000081; sol=0.9*10^(-5);

% ALGORITMO 1
s=p^2;
t=s+q;
if t>=0
    u=sqrt(t);
else
    fprintf('\n \t [RADICI COMPLESSE]');
end
s1=-p+u;
```

Codice complessita': algoritmo 2

```
% ALGORITMO 2
s=p^2;
t=s+q;
if t >=0
    u=sqrt(t);
else
    fprintf('\n \t [RADICI COMPLESSE]');
end
v=p+u;
t1=q/v;
```

Codice complessita': stampa risultati

```
fprintf('\n \t [ALG.1]: %10.19f',s1);
fprintf('\n \t [ALG.2]: %10.19f',t1);
if length(sol) > 0 & (sol ~= 0)
    rerr1 =abs(s1-sol)/abs(sol);
    rerr2=abs(t1-sol)/abs(sol);
    fprintf('\n \t [REL.ERR.ALG.1]: %2.2e',rerr1);
    fprintf('\n \t [REL.ERR.ALG.2]: %2.2e',rerr2);
end
```

Test.

Come previsto, il secondo algoritmo si comporta notevolmente meglio del primo, che compie un errore relativo dell'ordine di circa 10^{-9}. Infatti:

```
>> radicesecgrado

    [ALG.1]  [1]:  0.000008999999772772
    [ALG.2]  [1]:  0.000009000000000000
    [REL.ERR.][ALG.1]:  2.52e-009
    [REL.ERR.][ALG.2]:  0.00e+000

>>
```

Calcolo π

Eseguiamo un codice Matlab che valuti le successioni $\{u_n\}$, $\{z_n\}$, definite rispettivamente come

$$\begin{cases} s_1 = 1, \ s_2 = 1 + \frac{1}{4} \\ u_1 = 1, \ u_2 = 1 + \frac{1}{4} \\ s_{n+1} = s_n + \frac{1}{(n+1)^2} \\ u_{n+1} = \sqrt{6\, s_{n+1}} \end{cases}$$

e

$$\begin{cases} z_1 = 1, \ z_2 = 2 \\ z_{n+1} = 2^{n-\frac{1}{2}} \sqrt{1 - \sqrt{1 - 4^{1-n} \cdot z_n^2}} \end{cases} \tag{3}$$

che *teoricamente* convergono a π.

Calcolo π

Implementiamo poi la successione, diciamo $\{y_n\}$, che si ottiene *razionalizzando* (3), cioè moltiplicando numeratore e denominatore di

$$z_{n+1} = 2^{n-\frac{1}{2}} \sqrt{1 - \sqrt{1 - 4^{1-n} \cdot z_n^2}}$$

per

$$\sqrt{1 + \sqrt{1 - 4^{1-n} \cdot z_n^2}}$$

e calcoliamo u_m, z_m e y_m per $m = 2, 3, \ldots, 40$ (che teoricamente dovrebbero approssimare π).
Infine disegniamo in un unico grafico l'andamento dell'errore relativo di u_n, z_n e y_n rispetto a π aiutandoci con l'help di Matlab relativo al comando `semilogy`.

Calcolo π: metodo 1

In seguito scriviamo un'implementazione di quanto richiesto commentando i risultati. Si salvi in un file `pigreco.m` il codice

```
% SEQUENZE CONVERGENTI "PI GRECO".

% METODO 1.
s(1)=1; u(1)=1;
s(2)=1.25; u(2)=s(2);
for n=2:40
    s(n+1)=s(n)+(n+1)^(-2);
    u(n+1)=sqrt(6*s(n+1));
    fprintf('\n \t [SEQ.1][INDEX]: %3.0f ', n);
    fprintf('[REL.ERR]: %2.2e',abs(u(n+1)-pi)/pi);
end
rel_err_u=abs(u-pi)/pi;

fprintf('\n');
```

Calcolo π: metodo 2

```
% METODO 2.
format long
z(1)=1;
z(2)=2;
for n=2:40
    c=(4^(1-n)) * (z(n))^2; inner_sqrt=sqrt(1-c);
    z(n+1)=(2^(n-0.5))*sqrt( 1-inner_sqrt );
    fprintf('\n \t [SEQ.2][N]: %3.0f ', n);
    fprintf('[REL.ERR]: %2.2e',abs(z(n+1)-pi)/pi);
end
rel_err_z=abs(z-pi)/pi;

fprintf('\n');
```

Calcolo π: metodo 3

```
% METODO 3.
y(1)=1;
y(2)=2;
for n=2:40
    num=(2^(1/2)) * abs(y(n));
    c=(4^(1-n)) * (z(n))^2;
    inner_sqrt=sqrt(1-c);
    den=sqrt( 1+inner_sqrt );
    y(n+1)=num/den;
    relerr3=
    fprintf('\n \t [SEQ.3][N]: %3.0f ',n);
    fprintf('[REL.ERR]:%2.2e',abs(y(n+1)-pi)/pi);
end
rel_err_y=abs(y-pi)/pi;
```

Calcolo π: plots

```
% SEMILOGY PLOT.
hold on;
semilogy(1:length(u),rel_err_u,'k.');
semilogy(1:length(z),rel_err_z,'m+');
semilogy(1:length(y),rel_err_y,'ro');
hold off;
```

Di seguito digitiamo sulla shell di Matlab/Octave

```
>> pigreco
```

Plot risultati

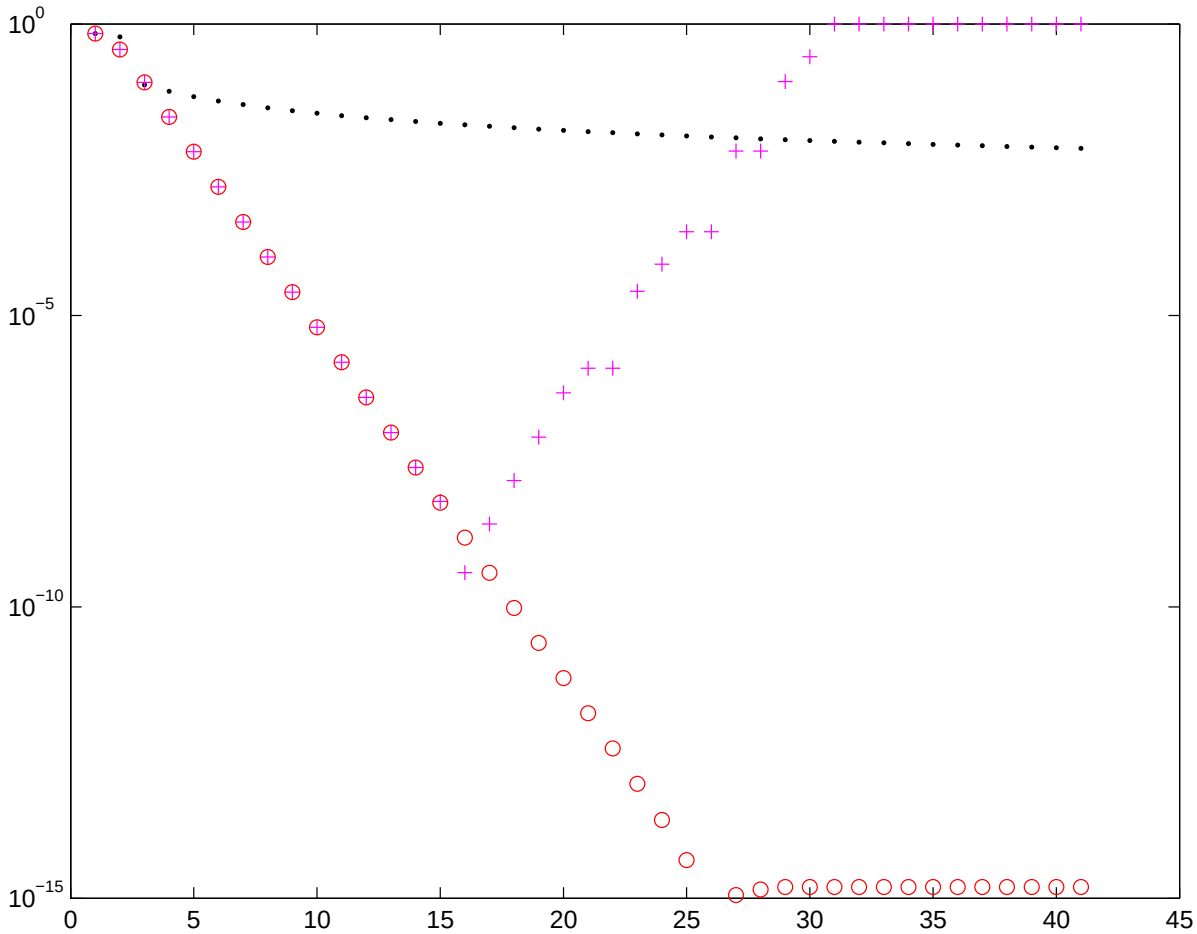

Figura: Grafico che illustra le 3 successioni, rappresentate rispettivamente da ., + e o.

Discussione risultati.

- La prima successione converge molto lentamente a π, la seconda diverge mentre la terza converge velocemente a π.

Discussione risultati.

- La prima successione converge molto lentamente a π, la seconda diverge mentre la terza converge velocemente a π.
- Per alcuni valori $\{z_n\}$ e $\{y_n\}$ coincidono per alcune iterazioni per poi rispettivamente divergere e convergere a π. Tutto ciò è naturale poichè le due sequenze sono analiticamente (ma non numericamente) equivalenti.

Discussione risultati.

- La prima successione converge molto lentamente a π, la seconda diverge mentre la terza converge velocemente a π.
- Per alcuni valori $\{z_n\}$ e $\{y_n\}$ coincidono per alcune iterazioni per poi rispettivamente divergere e convergere a π. Tutto ciò è naturale poichè le due sequenze sono analiticamente (ma non numericamente) equivalenti.
- Dal grafico dell'errore relativo, la terza successione, dopo aver raggiunto errori relativi prossimi alla precisione di macchina, si assesta ad un errore relativo di circa 10^{-15} (probabilmente per questioni di arrotondamento).

Una successione ricorrente.

Consideriamo la successione $\{I_n\}$ definita da

$$I_n = e^{-1} \int_0^1 x^n e^x \, dx \tag{4}$$

- $n = 0$: $I_0 = e^{-1} \int_0^1 e^x \, dx = e^{-1}(e^1 - 1)$.

Una successione ricorrente.

Consideriamo la successione $\{I_n\}$ definita da

$$I_n = e^{-1} \int_0^1 x^n \, e^x \, dx \tag{4}$$

- $n = 0$: $I_0 = e^{-1} \int_0^1 e^x \, dx = e^{-1}(e^1 - 1)$.
- integrando per parti

$$\begin{aligned} I_{n+1} &= e^{-1} \left(x^{n+1} \, e^x \, |_0^1 - (n+1) \int_0^1 x^n \, e^x \, dx \right) \\ &= 1 - (n+1) \, I_n. \end{aligned}$$

Una successione ricorrente.

Consideriamo la successione $\{I_n\}$ definita da

$$I_n = e^{-1} \int_0^1 x^n \, e^x \, dx \tag{4}$$

- $n = 0$: $I_0 = e^{-1} \int_0^1 e^x \, dx = e^{-1}(e^1 - 1)$.
- integrando per parti

$$\begin{aligned} I_{n+1} &= e^{-1} \left(x^{n+1} \, e^x \, |_0^1 - (n+1) \int_0^1 x^n \, e^x \, dx \right) \\ &= 1 - (n+1) \, I_n. \end{aligned}$$

- $I_n > 0$, decrescente e si prova che $I_n \longrightarrow 0$ come $1/n$.

Problema.

Calcoliamo I_n per $n = 1, \ldots, 99$:

- mediante la successione *in avanti*

$$I_n = e^{-1} \int_0^1 x^n e^x \, dx, I_0 = e^{-1}(e^1 - 1). \tag{5}$$

Si noti che se $I_{n+1} = 1 - (n+1) I_n$ allora $I_n = (1 - I_{n+1})/(n+1)$ e quindi $I_{n-1} = (1 - I_n)/n$.

Problema.

Calcoliamo I_n per $n = 1, \ldots, 99$:

- mediante la successione *in avanti*

$$I_n = e^{-1} \int_0^1 x^n e^x \, dx, I_0 = e^{-1}(e^1 - 1). \tag{5}$$

- mediante la successione *all'indietro*

$$\begin{cases} t_{1000} = 0 \\ t_{n-1} = \frac{1-t_n}{n} \end{cases}$$

Si noti che se $I_{n+1} = 1 - (n+1) I_n$ allora $I_n = (1 - I_{n+1})/(n+1)$ e quindi $I_{n-1} = (1 - I_n)/n$.

Successioni ricorrente in Matlab

Scriviamo il codice in un file `succricorrente.m`.

```
% SUCCESSIONE RICORRENTE.

% SUCCESSIONE "s_n".
s(1)=exp(-1);
for n=1:99
    s(n+1)=1-(n+1)*s(n);
end

% SUCCESSIONE "t_n".
m=500; M=2*m;
t=zeros(M,1); % INIZIALIZZAZIONE "t".
for n=M:-1:2
    j=n-1;
    t(j)=(1-t(n))/n;
end
```

Successioni ricorrente in Matlab

```
% PLOT SEMI-LOGARITMICO.
subplot(2,1,1)
semilogy(1:length(s),abs(s),'k-'); hold on;
semilogy(1:length(s),abs(t(1:length(s))),'m-');
hold off;
```

Di seguito digitiamo sulla shell di Matlab/Octave

```
>> succricorrente
```

Plot risultati

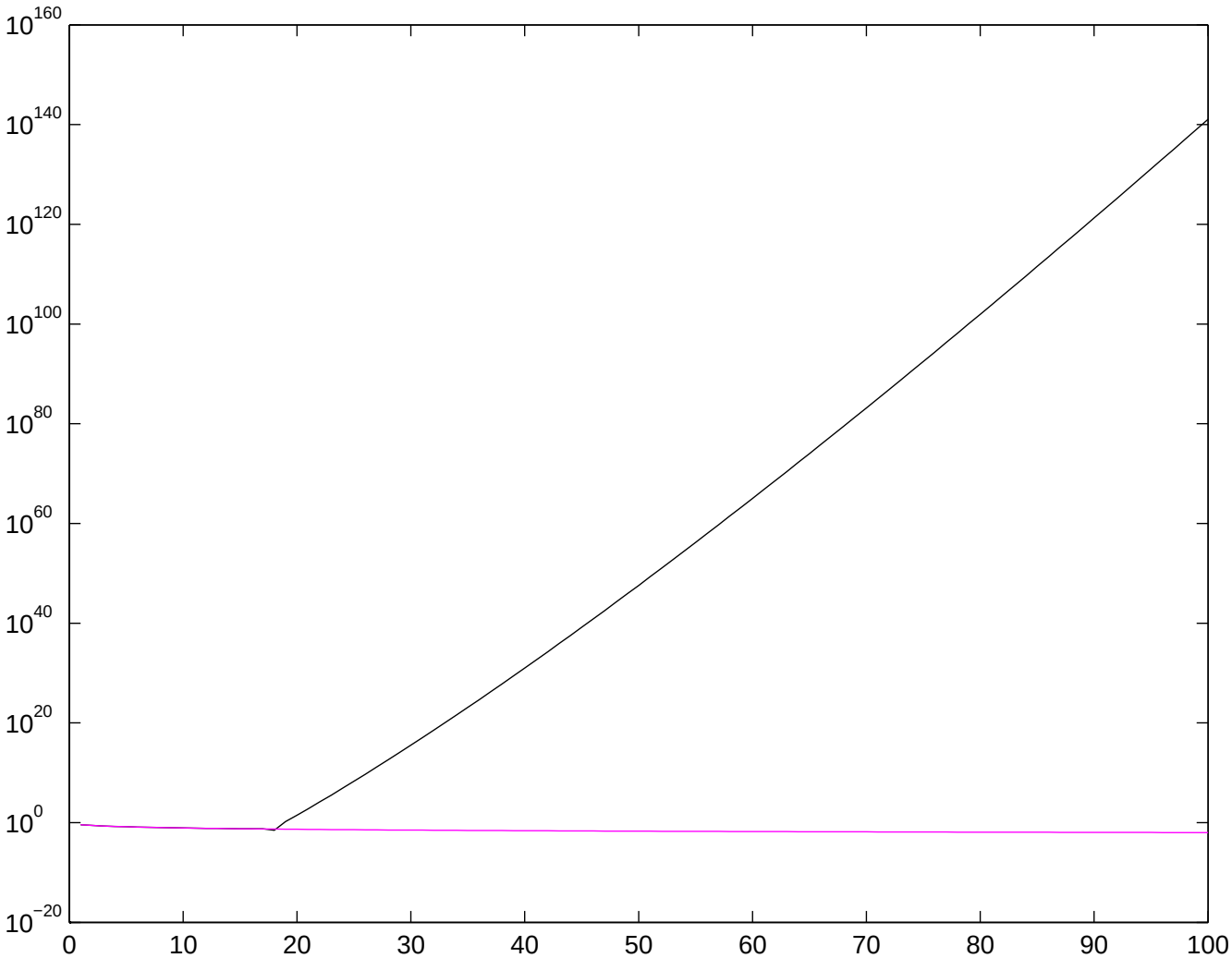

Figura: Grafico che illustra i valori assoluti assunti dalla successione in avanti (in nero) e all'indietro (in rosa magenta).

Commento successione ricorrente.

La successione in avanti amplifica gli errori. Infatti se

$$\overline{I_0} = I_0 + \epsilon$$

$$\overline{I}_{n+1} = 1 - (n+1)\overline{I}_n$$

allora

$$\begin{aligned}
\overline{I}_1 &= 1 - \overline{I}_0 = 1 - (I_0 + \epsilon) = I_1 - \epsilon \\
\overline{I}_2 &= 1 - 2\overline{I}_1 = 1 - 2(I_1 - \epsilon) = I_2 + 2 \cdot 1 \cdot \epsilon \\
\overline{I}_3 &= 1 - 3\overline{I}_2 = 1 - 3(I_2 + 2\epsilon) = I_3 - 3 \cdot 2 \cdot 1 \cdot \epsilon \\
\overline{I}_4 &= 1 - 4\overline{I}_3 = 1 - 4(I_3 + 3\epsilon) = I_4 - 4 \cdot 3 \cdot 2 \cdot 1 \cdot \epsilon
\end{aligned}$$

e in generale

$$\overline{I}_n = I_n + (-1)^n n! \cdot \epsilon.$$

con il termine $n! \cdot \epsilon$ che tende velocemente a $+\infty$ al crescere di n.

Commento successione ricorrente.

La successione all'indietro smorza gli errori. Infatti, se

$$\overline{I}_m = I_m + \epsilon$$

e

$$\overline{I}_{n-1} = (1 - \overline{I}_n)/n$$

allora

$$\begin{aligned}
\overline{I}_{m-1} &= (1 - \overline{I}_m)/m = (1 - (I_m + \epsilon))/m = I_m - \epsilon/m \\
\overline{I}_{m-2} &= (1 - (I_{m-1} + \epsilon/m))/(m-1) = I_{m-2} - \epsilon/((m-1)\cdot m) \\
&\dots \\
\overline{I}_{m-k} &= \dots = I_{m-k} - \epsilon/\prod_{s=0}^{k}(m-s)
\end{aligned}$$

con il termine $\epsilon/\prod_{s=0}^{k}(m-s)$ che tende velocemente a 0 al crescere di k.