ANALISI IN AVANTI DEL I ORDINE DEGLI ERRORI DI ARROTONDAMENTO

Si basa sul teorema fondamentale sulle operazioni tra numeri finiti: se $a, b \in F(\beta, t, L, U)$,

$$fl(a \bullet b) = (a \bullet b)(1 + \epsilon) \quad |\epsilon| \leq u$$

ove u è la precisione di macchina. Si assume che i dati siano esatti, ossia che appartengano a F.

TECNICA IN AVANTI. Si calcola l'errore relativo sul risultato finale in termine degli errori introdotti dalle singole operazioni, trascurando i termini in cui compaiono prodotti di errori (ANALISI DEL I ORDINE).

ESEMPIO. $\varphi(a,b,c) = a+b+c$

- ALGORITMO 1.

$$\begin{aligned}
fl(a+b) &= (a+b)(1+\epsilon_1) = y \quad |\epsilon_1| \leq u \\
fl(y+c) &= (y+c)(1+\epsilon_2) = z \quad |\epsilon_2| \leq u \\
fl(a+b+c) &= z = (y+c)(1+\epsilon_2) = \\
&= ((a+b)(1+\epsilon_1)+c)(1+\epsilon_2) = \\
&= (a+b)(1+\epsilon_1)(1+\epsilon_2) + c(1+\epsilon_2) = \\
&\simeq (a+b)(1+\epsilon_1+\epsilon_2) + c(1+\epsilon_2) = \\
&= a+b+c+(a+b)\epsilon_1 + (a+b+c)\epsilon_2
\end{aligned}$$

$(a+b)\epsilon_1\epsilon_2$ trascurabile

$$\epsilon_{alg1} = \frac{fl(a+b+c)-(a+b+c)}{a+b+c} \simeq \frac{a+b}{a+b+c}\epsilon_1 + \epsilon_2$$

Fattori di amplificazione dell'errore: $\begin{cases} \frac{a+b}{a+b+c} & \text{per } \epsilon_1 \\ 1 & \text{per } \epsilon_2 \end{cases}$

Si definisce INDICE ALGORITMICO I_{alg} la somma dei valori assoluti dei fattori di amplificazione dei singoli errori introdotti da ciascuna operazione:

$$I_{alg1} = |\frac{a+b}{a+b+c}| + 1$$

Il fattore di amplificazione dell'ultima operazione eseguita è sempre 1.

- ALGORITMO 2.

$$\begin{aligned}
fl(b+c) &= (b+c)(1+\epsilon_3) = w \quad |\epsilon_3| \leq u \\
fl(a+w) &= (a+w)(1+\epsilon_4) = v \quad |\epsilon_4| \leq u \\
fl(a+b+c) &= v = (a+w)(1+\epsilon_4) = \\
&= (a+(b+c)(1+\epsilon_3))(1+\epsilon_4) = \\
&= (b+c)(1+\epsilon_3)(1+\epsilon_4) + a(1+\epsilon_4) = \\
&\simeq (b+c)(1+\epsilon_3+\epsilon_4) + a(1+\epsilon_4) = \\
&= a+b+c+(b+c)\epsilon_3 + (a+b+c)\epsilon_4
\end{aligned}$$

$(b+c)\epsilon_3\epsilon_4$ trascurabile

$$\epsilon_{alg2} = \frac{fl(a+b+c)-(a+b+c)}{a+b+c} \simeq \frac{b+c}{a+b+c}\epsilon_3 + \epsilon_4$$

Fattori di amplificazione dell'errore: $\begin{cases} \frac{b+c}{a+b+c} & \text{per } \epsilon_3 \\ 1 & \text{per } \epsilon_4 \end{cases}$

$$I_{alg2} = |\frac{b+c}{a+b+c}| + 1$$

E' numericamente più stabile l'algoritmo per cui l'errore algoritmico è più piccolo. Se $a = .2337126\ 10^{-4}, b = .3367843\ 10^{2}, c = -0.3367781\ 10^{2}$:

$$I_{alg1} = |\frac{a+b}{a+b+c}| + 1 = 0.510^{5} + 1$$

$$I_{alg2} = |\frac{b+c}{a+b+c}| + 1 = .96 + 1$$

Il secondo è più stabile per i valori assunti dai dati.

ESEMPIO. $\varphi(a,b) = a^2 - b^2$

- ALGORITMO 1.

$$\begin{aligned}
fl(a^2) &= a * a(1+\epsilon_1) = x \quad |\epsilon_1| \leq u \\
fl(b^2) &= b * b(1+\epsilon_2) = y \quad |\epsilon_2| \leq u \\
fl(x-y) &= (x-y)(1+\epsilon_3) \quad |\epsilon_3| \leq u \\
fl(a^2-b^2) &= (a^2(1+\epsilon_1) - b^2(1+\epsilon_2))(1+\epsilon_3) = \\
&= a^2(1+\epsilon_1)(1+\epsilon_3) - b^2(1+\epsilon_2)(1+\epsilon_3) = \\
&\simeq a^2 - b^2 + a^2\epsilon_1 - b^2\epsilon_2 + (a^2-b^2)\epsilon_3
\end{aligned}$$

Si trascura $(a^2\epsilon_1 - b^2\epsilon_2)\epsilon_3$.

$$\begin{aligned}
\epsilon_{alg1} &= \frac{fl(a^2-b^2) - (a^2-b^2)}{a^2-b^2} \\
&\simeq \frac{a^2}{a^2-b^2}\epsilon_1 - \frac{b^2}{a^2-b^2}\epsilon_2 + \epsilon_3
\end{aligned}$$

$$I_{alg1} = \left|\frac{a^2}{a^2-b^2}\right| + \left|\frac{b^2}{a^2-b^2}\right| + 1$$

- ALGORITMO 2.

$$\begin{aligned}
fl(a+b) &= (a+b)(1+\epsilon_4) = z \quad |\epsilon_4| \leq u \\
fl(a-b) &= (a-b)(1+\epsilon_5) = v \quad |\epsilon_5| \leq u \\
fl(zv) &= zv(1+\epsilon_6) \quad |\epsilon_6| \leq u \\
fl(a^2-b^2) &= (a+b)(1+\epsilon_4)(a-b)(1+\epsilon_5)(1+\epsilon_6) = \\
&\simeq (a^2-b^2)(1+\epsilon_4+\epsilon_5+\epsilon_6)
\end{aligned}$$

$$\epsilon_{alg2} = \frac{fl(a^2 - b^2) - (a^2 - b^2)}{a^2 - b^2} \simeq \epsilon_4 + \epsilon_5 + \epsilon_6$$
$$I_{alg2} = 1 + 1 + 1 = 3$$

Si analizza quando l'Algoritmo 2 è numericamente più stabile dell'Algoritmo 1. Si controlla quando:

$$I_{alg1} \geq I_{alg2}$$

$$\frac{a^2}{|a^2 - b^2|} + \frac{b^2}{|a^2 - b^2|} + 1 \geq 3$$
$$\Downarrow$$
$$a^2 + b^2 \geq 2|a^2 - b^2|$$

Ciò si verifica se:

1. $a^2 - b^2 \leq \frac{a^2}{2} + \frac{b^2}{2} \Rightarrow \frac{a^2}{b^2} \leq 3$
2. $-(a^2 + b^2) \leq 2(a^2 - b^2) \Rightarrow \frac{1}{3} \leq \frac{a^2}{b^2}$

Dunque Algoritmo 2 è numericamente più stabile di algoritmo 1 se

$$\frac{1}{3} \leq \frac{a^2}{b^2} \leq 3$$

Prodotto di n numeri finiti

Siano $x_1, x_2, ..., x_n \in F$,

$$\varphi(x_1, x_2, ..., x_n) = x_1 x_2 ... x_n$$

ALGORITMO
$p = x_1$
for $j = 2$ to n do
  $p = p\ x_j$
$p = fl(x_1 x_2 ... x_n)$

$$\begin{aligned} fl(x_1 x_2 ... x_n) &= x_1 x_2 (1+\epsilon_2) x_3 (1+\epsilon_3) ... x_n (1+\epsilon_n) = \\ &= x_1 x_2 ... x_n (1+\epsilon_2)(1+\epsilon_3)...(1+\epsilon_n) = \\ &\simeq x_1 x_2 ... x_n (1+\epsilon_2+\epsilon_3+...+\epsilon_n) = \\ &= x_1 x_2 ... x_n + x_1 x_2 ... x_n \epsilon_2 + ... + x_1 x_2 ... x_n \epsilon_n \end{aligned}$$

$$\frac{fl(x_1 x_2 ... x_n) - (x_1 x_2 ... x_n)}{x_1 ... x_n} \simeq \epsilon_2 + ... + \epsilon_n$$

Se $|\epsilon_i| \leq u \quad i = 2, ..., n$, $|\epsilon_{alg}| \leq (n-1)u$.

$$I_{alg} = n - 1$$

L'algoritmo è numericamente stabile.

Somma di n numeri finiti

Siano $x_1, x_2, ..., x_n \in F$,

$$\varphi(x_1, x_2, ..., x_n) = x_1 + x_2 + ... + x_n = S$$

ALGORITMO
$s = x_1$
for $j = 2$ to n do
$\quad s = s + x_j$
$s = fl(x_1 + x_2 + ... + x_n)$

$$\begin{aligned}
fl(S) &= (...(((x_1 + x_2)(1 + \epsilon_2) + x_3)(1 + \epsilon_3) + x_4)(1 + \epsilon_4) + \\
&\quad ... + x_n)(1 + \epsilon_n) = \\
&= x_1(1 + \epsilon_2)(1 + \epsilon_3)...(1 + \epsilon_n) + x_2(1 + \epsilon_2)...(1 + \epsilon_n) + \\
&\quad + x_3(1 + \epsilon_3)...(1 + \epsilon_n) + ... + x_n(1 + \epsilon_n) \\
&\simeq x_1(1 + \epsilon_2 + \epsilon_3 + ... + \epsilon_n) + x_2(1 + \epsilon_2 + ... + \epsilon_n) + \\
&\quad + x_3(1 + \epsilon_3 + ... + \epsilon_n) + ... + x_n(1 + \epsilon_n) = \\
&= x_1 + x_2 + ... + x_n + (x_1 + x_2)\epsilon_2 + \\
&\quad + (x_1 + x_2 + x_3)\epsilon_3 + ... + (x_1 + x_2 + x_{n-1})\epsilon_{n-1} + \\
&\quad + (x_1 + x_2 + ... + x_n)\epsilon_n
\end{aligned}$$

$$\begin{aligned}
\frac{fl(S) - S}{S} &\simeq \frac{x_1 + x_2}{x_1 + ... + x_n}\epsilon_2 + \frac{x_1 + x_2 + x_3}{x_1 + ... + x_n}\epsilon_3 + \\
&\quad + ... + \frac{x_1 + ... + x_{n-1}}{x_1 + ... + x_n}\epsilon_{n-1} + \epsilon_n
\end{aligned}$$

$$I_{alg} = |\frac{x_1 + x_2}{S}| + |\frac{x_1 + x_2 + x_3}{S}| + ... + |\frac{x_1 + ... + x_{n-1}}{S}| + 1$$

Supponiamo $|\epsilon_i| \leq u \;\; i = 2, ..., n.$

- Se gli $x_i \; i = 1, ..., n$ sono di segno concorde, $|\epsilon_{alg}| \leq (n-1)u$. Tuttavia l'algoritmo è più stabile numericamente se si sommano i numeri dal più piccolo al più grande.
- Se gli x_i sono di segno discorde, S può essere piccolo e si ha amplificazione degli errori. Se occorre avere la somma in semplice precisione, è conveniente fare la somma di tutti i positivi e poi di tutti i negativi separatamente in semplice precisione e poi sommare le due somme parziali in doppia precisione.

ESEMPIO. Sia $\beta = 10, t = 7$, arrotondamento. Sia $x_1 = 1, x_i = .1\ 10^{-6}\ i = 2, ..., 10.$

ALGORITMO 1.

$s \leftarrow x_1$
for $i \leftarrow 2$ to 10 do
  $s \leftarrow s + x_i$
$fl(1 + .1\ 10^{-6}) =$
$= fl(1.0000001) = 1$
$fl(1 + .1\ 10^{-6}) = 1$
...
$s \leftarrow 1$
$\epsilon_r = 8.999..\ 10^{-7}$

ALGORITMO 2.

$s \leftarrow x_{10}$
for $i \leftarrow 9$ to 1 step -1 do
  $s \leftarrow s + x_i$
$fl(.1\ 10^{-6} + .1\ 10^{-6}) = fl(.2\ 10^{-6})$
$= .2\ 10^{-6}$
...
$fl(.9\ 10^{-6} + 1) = fl(1.0000009)$
$s \leftarrow 1.000001$
$\epsilon_r = 9.999..\ 10^{-8}$

Prodotto scalare tra numeri finiti

Siano $x = (x_1, ..., x_n)^T, y = (y_1, ..., y_n)^T$ due vettori di $\mathbb{R}^n$. Il prodotto scalare è dato da

$$P = x^T y = \sum_{i=1}^{n} x_i y_i$$

ALGORITMO
$p = x_1\ y_1$
for $j = 2$ to n do
$\quad p = p + x_j\ y_j$
$p = fl(x_1\ y_1 + x_2\ y_2 + ... + x_n\ y_n)$

$$\begin{aligned} fl(P) &= (...(((x_1\ y_1(1+\epsilon_1) + x_2\ y_2(1+\epsilon_2))(1+\sigma_2) + \\ &\quad +x_3\ y_3(1+\epsilon_3))(1+\sigma_3) \\ &\quad +x_4\ y_4(1+\epsilon_4))(1+\sigma_4) + \\ &\quad ... + x_n\ y_n(1+\epsilon_n))(1+\sigma_n) = \\ &= x_1\ y_1(1+\epsilon_1)(1+\sigma_2)(1+\sigma_3)...(1+\sigma_n) + \\ &\quad +x_2\ y_2(1+\epsilon_2)(1+\sigma_2)...(1+\sigma_n) + \\ &\quad +x_3\ y_3(1+\epsilon_3)(1+\sigma_3)...(1+\sigma_n) + \\ &\quad +... + x_n\ y_n(1+\epsilon_n)(1+\sigma_n) \end{aligned}$$

$$
\begin{aligned}
fl(P) &\simeq x_1\ y_1(1+\epsilon_1+\sigma_2+\ldots+\sigma_n)+ \\
&\quad +x_2\ y_2(1+\epsilon_2+\sigma_2+\ldots+\sigma_n)+ \\
&\quad +x_3\ y_3(1+\epsilon_3+\sigma_3+\ldots+\sigma_n)+\ldots \\
&\quad +x_n\ y_n(1+\epsilon_n+\sigma_n)= \\
&= P+x_1\ y_1\epsilon_1+x_2\ y_2\epsilon_2+\ldots+x_n\ y_n\epsilon_n+ \\
&\quad +(x_1\ y_1+x_2\ y_2)\sigma_2+(x_1\ y_1+x_2\ y_2+x_3\ y_3)\sigma_3+ \\
&\quad +\ldots+(x_1\ y_1+x_2\ y_2+\ldots+x_{n-1}\ y_{n-1})\sigma_{n-1}+ \\
&\quad +(x_1\ y_1+x_2\ y_2+\ldots+x_n\ y_n)\sigma_n
\end{aligned}
$$

$$
\begin{aligned}
\frac{fl(P)-P}{P} &\simeq \frac{x_1\ y_1}{P}\epsilon_1+\ldots+\frac{x_n\ y_n}{P}\epsilon_n+ \\
&\quad +\frac{x_1\ y_1+x_2\ y_2}{P}\sigma_2+\frac{x_1\ y_1+x_2\ y_2+x_3\ y_3}{P}\sigma_3+ \\
&\quad +\ldots+\frac{x_1\ y_1+\ldots+x_{n-1}\ y_{n-1}}{P}\sigma_{n-1}+\sigma_n
\end{aligned}
$$

$$
I_{alg}=\sum_{i=1}^{n}\left|\frac{x_i\ y_i}{P}\right|+\sum_{j=2}^{n}\left|\frac{\sum_{i=1}^{j}x_i\ y_i}{P}\right|
$$

Se $P \simeq 0$, l'algoritmo è instabile. Poichè il prodotto scalare è un'operazione molto usata in algebra lineare, si consiglia di convertire i dati dalla semplice precisione alla doppia precisione e di effettuare l'operazione in doppia precisione. Il risultato viene poi convertito alla semplice precisione. In tal modo l'errore di arrotondamento è dell'ordine della precisione di macchina.

Schema di Horner

$$p_n(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_n \quad a_i \in F,\ \alpha \in F$$

$$\begin{aligned}
fl(p_n(\alpha)) \quad &= \quad (\ldots(((a_0\alpha(1+\epsilon_0)+a_1)(1+\delta_1)\alpha(1+\epsilon_1)+ \\
&\quad +a_2)(1+\delta_2)\alpha(1+\epsilon_2)+ \\
&\quad +a_3)(1+\delta_3)\alpha(1+\epsilon_3)+\ldots \\
&\quad +a_{n-1})(1+\delta_{n-1})\alpha(1+\epsilon_{n-1})+ \\
&\quad +a_n)(1+\delta_n) = \\
&= \quad a_0\alpha^n(1+\epsilon_0)\ldots(1+\epsilon_{n-1})(1+\delta_1)\ldots(1+\delta_n)+ \\
&\quad +a_1\alpha^{n-1}(1+\epsilon_1)\ldots(1+\epsilon_{n-1})(1+\delta_1)\ldots(1+\delta_n)+ \\
&\quad +a_2\alpha^{n-2}(1+\epsilon_2)\ldots(1+\epsilon_{n-1})(1+\delta_2)\ldots(1+\delta_n)+ \\
&\quad +a_{n-1}\alpha(1+\epsilon_{n-1})(1+\delta_{n-1})(1+\delta_n)+ \\
&\quad +a_n(1+\delta_n) = \\
&\simeq \quad a_0\alpha^n(1+\epsilon_0+\ldots+\epsilon_n+\delta_1+\ldots+\delta_n)+ \\
&\quad +a_1\alpha^{n-1}(1+\epsilon_1+\ldots+\epsilon_{n-1}+\delta_1+\ldots+\delta_n)+ \\
&\quad +a_2\alpha^{n-2}(1+\epsilon_2+\ldots+\epsilon_{n-1}+\delta_2+\ldots+\delta_n)+ \\
&\quad +\ldots+ \\
&\quad +a_{n-1}\alpha(1+\epsilon_{n-1}+\delta_{n-1}+\delta_n)+ \\
&\quad +a_n(1+\delta_n) = \\
&= \quad p_n(\alpha)+a_0\alpha^n(\epsilon_0+\ldots+\epsilon_{n-1}+\delta_1+\ldots+\delta_n)+ \\
&\quad +a_1\alpha^{n-1}(\epsilon_1+\ldots+\epsilon_{n-1}+\delta_1+\ldots+\delta_n)+ \\
&\quad +a_2\alpha^{n-2}(\epsilon_2+\ldots+\epsilon_{n-1}+\delta_2+\ldots+\delta_n)+\ldots+ \\
&\quad +a_{n-1}\alpha(\epsilon_{n-1}+\delta_{n-1}+\delta_n)+a_n\delta_n
\end{aligned}$$

$$\begin{aligned}\frac{fl(p_n(\alpha)) - p_n(\alpha)}{p_n(\alpha)} \quad &\simeq \quad \frac{a_0\alpha^n}{p_n(\alpha)}\epsilon_0 + \\ &+\frac{a_0\alpha^n + a_1\alpha^{n-1}}{p_n(\alpha)}(\epsilon_1 + \delta_1) + \\ &+\frac{a_0\alpha^n + a_1\alpha^{n-1} + a_2\alpha^{n-2}}{p_n(\alpha)}(\epsilon_2 + \delta_2) + \\ &+\ldots + \\ &+\frac{a_0\alpha^n + a_1\alpha^{n-1} + \ldots + a_{n-1}\alpha}{p_n(\alpha)}. \\ &.(\epsilon_{n-1} + \delta_{n-1}) + \\ &+\delta_n\end{aligned}$$

Se $|\epsilon_i| \leq u, |\delta_i| \leq u$,

$$|fl(p_n(\alpha)) - p_n(\alpha)| \leq u(|a_0\alpha^n|2n + |a_1\alpha^{n-1}|(2n-1) + \ldots$$

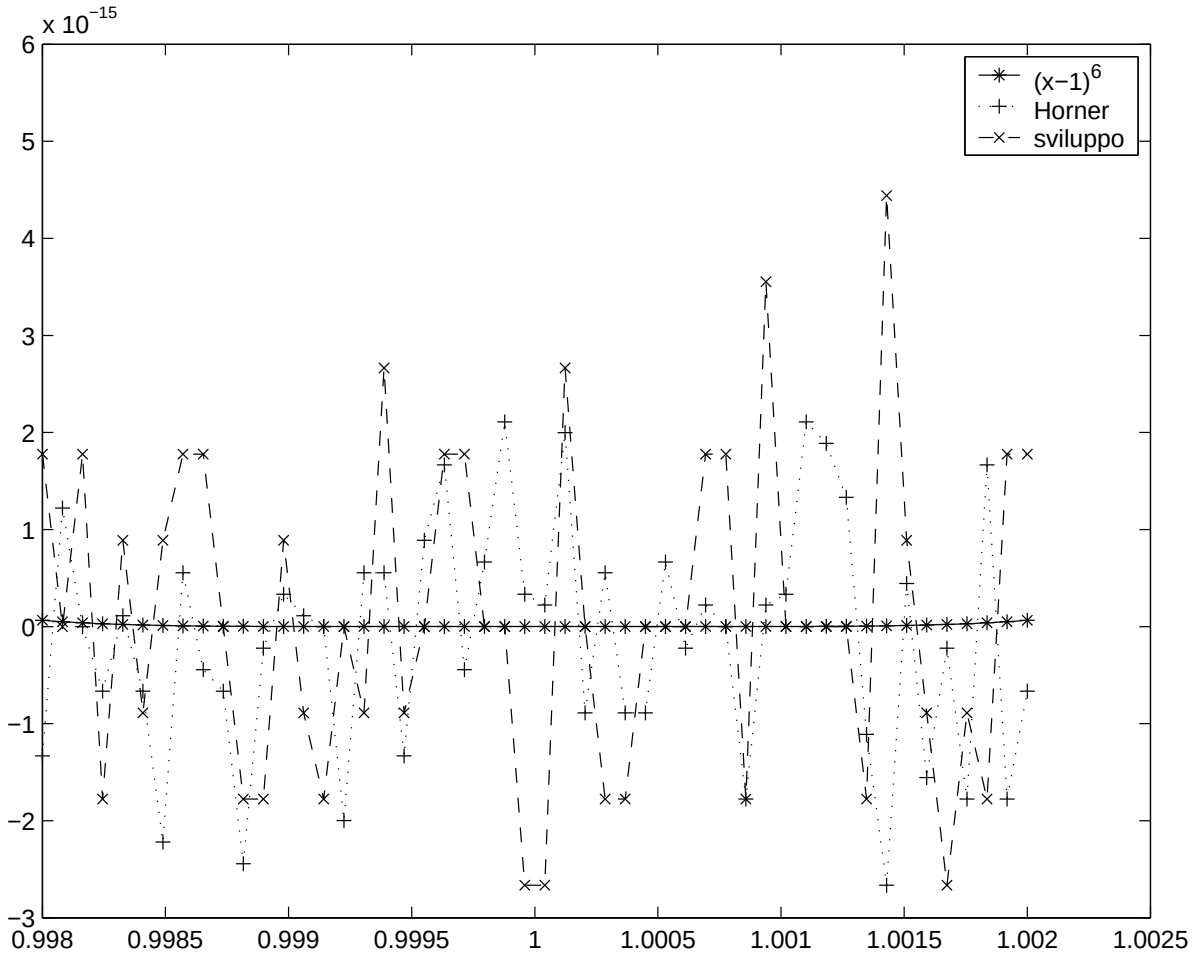
$$+|a_2\alpha^{n-2}|(2n-3) + \ldots + |a_{n-1}\alpha|3 + |a_n|)$$

Se $p_n(\alpha) \simeq 0$, l'algoritmo è instabile. Può essere che $fl(p_n(\alpha)) = 0$, pur non essendo α uno zero del polinomio.

Calcolo del polinomio $(x-1)^6$.

```
clear; x=linspace(0.998,1.002,50);
y1=(x-1).^6;
p=[1 -6 15 -20 15 -6 1];
y2=polyval(p,x);
y3=x.^6-6*x.^5+15*x.^4-20*x.^3+15*x.^2-6*x+1;
%
plot(x,y1,'-*',x,y2,':+',x,y3,'--x');
legend('(x-1)^6','Horner','sviluppo');
fprintf('x\t (x-1)^6\t  Horner \t sviluppo \n'); for
i=1:3:length(x);
fprintf('%7.4e\t %6.3e\t %6.3e\t...
   %6.3e  \n', x(i), y1(i),y2(i),y3(i));
end;
```

```
>> sviluppo
x              (x-1)^6      Horner       sviluppo
9.9800e-001  6.400e-017  -1.332e-015    1.776e-015
9.9824e-001  2.923e-017  -6.661e-016    -1.776e-015
9.9849e-001  1.186e-017  -2.220e-015    8.882e-016
9.9873e-001  4.104e-018  -6.661e-016    0.000e+000
9.9898e-001  1.129e-018  3.331e-016     8.882e-016
9.9922e-001  2.175e-019  -1.998e-015    0.000e+000
9.9947e-001  2.232e-020  -1.332e-015    -8.882e-016
9.9971e-001  5.440e-022  -4.441e-016    1.776e-015
9.9996e-001  4.624e-027  3.331e-016     -2.665e-015
1.0002e+000  7.225e-023  -8.882e-016    0.000e+000
1.0004e+000  8.191e-021  -8.882e-016    0.000e+000
1.0007e+000  1.116e-019  2.220e-016     1.776e-015
1.0009e+000  6.845e-019  2.220e-016     3.553e-015
1.0012e+000  2.750e-018  1.887e-015     0.000e+000
1.0014e+000  8.500e-018  -2.665e-015    4.441e-015
1.0017e+000  2.196e-017  -2.220e-016    -2.665e-015
```


x 10^−15
6
5
4
3
2
1
0
−1
−2
−3
0.998
0.9985
0.999
0.9995
1
1.0005
1.001
1.0015
1.002
1.0025
(x−1)^6
Horner
sviluppo

Calcolo delle radici dell'equazione di II grado:

$$x^2 - 6.433\ x + .009474 = 0$$

usando base 10 e 4 cifre significative (con arrotondamento).

$$\rho = \frac{6.433 - \sqrt{6.433^2 - 4(.009474)}}{2}$$

- $fl(6.433^2) = .4130\ 10^2$
- $fl(4\ .009474) = .3789\ 10^{-1}$
- $fl(.4138\ 10^2 - .3789\ 10^{-1}) = .4134\ 10^2$
- $fl(\sqrt{.4344\ 10^2}) = .6429\ \ 10$; $E_r(6.433) = 0$, $E_r(\sqrt{b^2 - 4ac}) = 1.6\ 10^{-4}$
- $fl(6.433 - 6.429) = .4\ 10^{-2}$)
- $fl(\rho) = .2\ 10^{-2}$

Ma $\rho = .0014731....$ Quindi l'errore relativo vale $E_r \simeq .36$.
L'errore è prodotto dall'errore di incolonnamento e dalla conseguente perdita di cifre significative nel calcolo di $(6.433)^2 - 4\ .009474$. $4ac$ è trascurabile rispetto b^2. Pertanto la successiva cancellazione provoca una amplificazione dell'errore commesso.
ALTERNATIVE

$$ax^2 + bx + c = 0 \qquad x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

1. usare $x_1 = \frac{-b - \text{segno}(b)\sqrt{b^2-4ac}}{2a}$, $x_2 = \frac{c}{ax_1}$
2. usare $x_1 = \frac{-b - \text{segno}(b)\sqrt{b^2-4ac}}{2a}$, $x_2 = \frac{2c}{-b - \text{segno}(b)\sqrt{b^2-4ac}}$

- Normalizzare i dati $(10^{30}x^2 + 10^{30}x + 10^{30})$
- Controllare che c non sia trascurabile rispetto ad a e a b e che a non sia trascurabile rispetto a b e a c.

Calcolo dell'integrale $E_n = \int_0^1 x^n e^{x-1} dx$, $n = 1, 2, \ldots$ mediante la formula di ricorrenza

$$E_n = 1 - nE_{n-1} \qquad E_1 = 1/e$$

Si assume aritmetica in base 10 e 6 cifre significative (con arrotondamento).

- $E_1 = 0.367879$
- $E_2 = 0.264242$
- $E_3 = 0.207274$
- $E_4 = 0.170904$
- $E_5 = 0.145480$
- $E_6 = 0.127120$
- $E_7 = 0.110160$
- $E_8 = 0.118720$
- $E_9 = -0.06848$ assurdo!!!

L'errore commesso al passo i viene amplificato di $(i+1)(i+2)(i+3)\ldots$.
L'errore commesso per il calcolo di $1/e$ viene amplificato di $9!$: $4.412\ 10^{-7}\ 9! = 0.1601$.
Se si usa

$$E_{n-1} = \frac{1 - E_n}{n} \qquad n = \ldots, 3, 2$$

l'errore iniziale è diviso per n ad ogni passo. Se si pone $E_{20} = 0$, al passo 15 l'errore è già ridotto a $4.8\ 10^{-8}$. Il secondo algoritmo è più stabile del primo. La stabilità dipende dall'ordine delle operazioni.

Calcolo di $e^{-5.5}$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$$

Se si calcola $e^{-5.5}$ con lo sviluppo in serie usando aritmetica in base 10 con 5 cifre significative e arrotondamento, ci si arresta dopo 25 termini (il termine successivo non altera la somma).

$$e^{-5.5} = 1 - 5.5 + 15.125 - 27.730 + 38.129 - 41.942 + 38.446 -$$

$$-30.208 + 20.768 - 12.692 + 1.9803 - 3.4902 +$$
$$+1.5997 - ... = 0.0026363$$
$$e^{-5.5} = 0.00408677 \quad \text{errore del 36\%}$$

Si tratta di termini con ordini di grandezza differenti. Si perdono le cifre che influiscono sul risultato.
ALTERNATIVA

$$e^{-5.5} = \frac{1}{e^{5.5}} \simeq .0040865 \quad \text{errore dello 0.007\%}$$

Per il calcolo di e^x, con $x = [x] + f$, conviene

1. $e^x = e^{[x]}e^f = (e.e....e)(1 + f + f^2/2! + ...)$
2. $e^x = (e^{1+\frac{f}{[x]}})^{[x]} = (\sum_{i=0}^{\infty}(1 + \frac{f}{[x]})^i/i!)^{[x]}$, $1 \leq 1 + f/[x] < 2$

```
x=input('x=');
yes=exp(x);
fprintf('exp(x)=%g \n',yes);
y=expmio(x);
err=abs(y-yes)/abs(yes);
fprintf('valore=%g errore(sviluppo)=%g \n',y,err);
% modo alternativo
ind=0;
if x<0
   ind=1;
   x=-x;
end;
if x>=1
   xint=fix(x);
   xfrac=x-xint;
   y1=expmio(1+xfrac/xint);
   y1=y1^xint;
else
   y1=expmio(x);
end;
if ind==1
y1=1/y1;
end;
err1=abs(yes-y1)/abs(yes);
fprintf('valore=%g errore=%g \n',y1,err1);

>> esponenziale
x=-5.5
exp(x)=0.00408677
valore=0.00408677 errore(sviluppo)=7.08445e-013
valore=0.00408677 errore=1.06118e-015
```

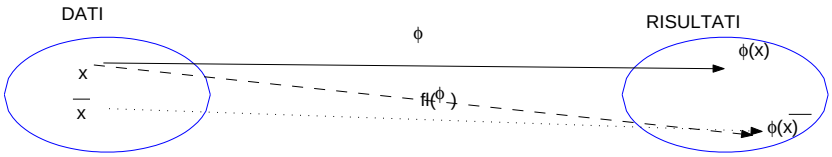
```
>> esponenziale
x=-50.5
exp(x)=1.16985e-022
valore=122132 errore(sviluppo)=1.044e+027
valore=1.16985e-022 errore=1.1857e-014

function y=expmiov(x);
% uso della funzione scalare
p=length(x);
y=ones(p,1);
for i=1:p
   y(i)=expmio(x(i));
end;

function y=expvett(x);
% funzione vettoriale
y=zeros(size(x));
term=ones(size(x));
k=0;
while any(abs(term)>eps*abs(y))
   y=y+term;
   k=k+1;
   term=term.*x/k;
end;
```

Altre tecniche per l'analisi degli errori

- ANALISI ALL'INDIETRO. Si considera il risultato approssimato come risultato esatto di un problema perturbato. Se la perturbazione calcolata è grande, l'algoritmo è instabile.



ANALISI ALL'INDIETRO
$\bar{x} - x$

ANALISI IN AVANTI
$\varphi(\bar{x}) - \varphi(x) =$
$= fl(\varphi(x)) - \varphi(x)$

ESEMPIO.

$$\begin{aligned} fl(x_1 + x_2) &= (x_1 + x_2)(1 + \epsilon) = \\ &= (x_1 + x_1\epsilon) + (x_2 + x_2\epsilon) \end{aligned}$$

E' somma esatta di due dati affetti da un errore assoluto pari a $x_1\epsilon$ e $x_2\epsilon$.

$$\begin{aligned} fl(x_1 \cdot x_2) &= x_1 \cdot x_2(1 + \epsilon) = \\ &= x_1(x_2 + x_2\epsilon) \\ &= (x_1 + x_1\epsilon)x_2 \end{aligned}$$

- USO DELLA DOPPIA PRECISIONE
- ARITMETICA DELL'INTERVALLO. Si associa ad ogni dato il suo intervallo di variabilità, definito da numeri di macchina:

$$x \rightarrow [x', x''] \quad x', x'' \in F$$

  Si ridefiniscono le operazioni come operazioni su un intervallo.
  Per esempio, $[a', a''] + [b', b''] = [c', c'']$, ove $c' = \max\{t' \in F, t' \leq a' + b'\}$, $c'' = \min t'' \in F, a'' + b'' \leq t''$.
- METODI STATISTICI

ANALISI DELL'ERRORE SUI DATI INIZIALI

ESEMPIO. $x^2 - 4x + \alpha = 0$

Per $\alpha = 4$, le soluzioni sono $x_{1,2} = 2$.

Per $\alpha = 4 - 10^{-6}$, le soluzioni esatte sono due, date da $x_{1,2} = 2 \pm 10^{-3}$.

Piccole variazioni nei dati iniziali (10^{-6}) comportano grosse variazioni nei risultati finali (10^{-3}).

Questo è un esempio di problema MAL CONDIZIONATO.

$$\begin{aligned} \epsilon_{dati} &= \frac{x(\alpha+\delta) - x(\alpha)}{x(\alpha)} = \\ &\simeq \frac{x(\alpha) + x'(\alpha)\delta - x(\alpha)}{x(\alpha)} = \\ &= \frac{x'(\alpha)\alpha}{x(\alpha)}\frac{\delta}{\alpha} \end{aligned}$$

δ/α è l'errore relativo sul dato iniziale.

$\frac{x'(\alpha)\alpha}{x(\alpha)}$ è il fattore di amplificazione dell'errore sui dati iniziali. Viene detto indice di condizionamento del problema.

$$\begin{aligned} P(x(\alpha),\alpha) = 0 \quad &\Rightarrow \quad P'_x(x,\alpha)x'(\alpha) + P'_\alpha(x,\alpha) = 0 \\ &\Rightarrow \quad x'(\alpha) = -\frac{P'_\alpha(x,\alpha)}{P'_x(x,\alpha)} = \\ &= \frac{-1}{2x(\alpha) - 4} \end{aligned}$$

ESEMPIO.

$$\begin{cases} x + \alpha y &= 1 \\ \alpha x + y &= 0 \end{cases}$$

Le soluzioni sono $x = \frac{1}{1-\alpha^2}$ e $y = \frac{-\alpha}{1-\alpha^2}$. Per α^2 prossimo a 1, il problema è mal condizionato, mentre è ben condizionato per α^2 distante da 1. Infatti,

$$\frac{x'(\alpha)\alpha}{x(\alpha)}\frac{\delta}{\alpha} = \frac{2\alpha^2}{1-\alpha^2}\frac{\delta}{\alpha} \simeq \frac{x(\alpha+\delta)-x(\alpha)}{x(\alpha)}$$

Calcolo di

$$z = x + y = \frac{1}{1-\alpha^2} - \frac{\alpha}{1-\alpha^2} = \frac{1}{1+\alpha}$$

Il calcolo è ben condizionato per $\alpha \simeq 1$ e mal condizionato per $\alpha \simeq -1$.

$$\frac{z(\alpha+\delta)-z(\alpha)}{z(\alpha)} \simeq \frac{-\alpha}{1+\alpha}\frac{\delta}{\alpha}$$

Tuttavia per $\alpha = .99$, con $\beta = 10, t = 4$ e arrotondamento:

$$fl(z) = fl(\frac{1}{.0199} - \frac{.99}{.0199}) = 50.25 - 49.75 = 0.5$$

mentre $z = .5025125$, $\epsilon_{tot} = 4.999\ 10^{-3}$. Ciò è dovuto all'amplificazione dell'errore fatta dalla cancellazione. L'algoritmo è instabile. Invece $fl(\frac{1}{1+\alpha}) = 0.5025$, $\epsilon_{tot} \simeq 2.49\ 10{-}5$.

Gli esempi mostrano che:

- un problema può essere ben condizionato per certi valori e mal condizionato per altri;
- un problema ben condizionato può dare risultati non buoni se risolto con algoritmi instabili;
- il condizionamento può essere espresso mediante la derivata della trasformazione;
- un algoritmo è stabile o meno a seconda del condizionamento delle trasformazioni che lo compongono.

Analisi dell'errore sui dati iniziali

Si vuole calcolare $y = \varphi(x)$.

Tuttavia x è affetto da un errore Δx, per cui si calcola

$$\varphi(x + \Delta x) = y + \Delta y$$

Se φ è derivabile, facendo un'analisi del I ordine,

$$y + \Delta y = \varphi(x + \Delta x) \simeq \varphi(x) + \varphi'(x)\Delta x$$

$$E_{dati} = \Delta y \simeq \varphi'(x)\Delta x$$

Se $\varphi(x) \neq 0,\ x \neq 0$, segue

$$\epsilon_{dati} = \frac{\Delta y}{y} \simeq \frac{\varphi'(x)x}{\varphi(x)}\frac{\Delta x}{x} = \frac{\varphi'(x)x}{\varphi(x)}\epsilon_x$$

L'errore $\epsilon_x = \frac{\Delta x}{x}$ produce un errore sui risultati finali $\epsilon_{dati} = \frac{\Delta y}{y}$ amplificato della quantità:

$$I_{cond} = K(x, \varphi) = |\frac{\varphi'(x)}{\varphi(x)}x|$$

detta INDICE DI CONDIZIONAMENTO.

Se $K(x, \varphi) >> 1$, il problema è mal condizionato; se $K(x, \varphi)$ è piccolo, il problema è ben condizionato.

ESEMPIO. $y = \varphi(x) = \log(x)$.

$$\epsilon_{dati} = \frac{1}{\log(x)}\epsilon_x$$

$$I_{cond} = |\frac{1}{\log(x)}|$$

Consideriamo il caso generale:

$$y = \varphi(x_1, ..., x_n)$$

Se $x_i,\ i = 1, ..., n$ è affetto da un errore Δx_i, con $\epsilon_{x_i} = \frac{\Delta x_i}{x_i}$, allora si calcola:

$$y + \Delta y = \varphi(x_1 + \Delta x_1, ..., x_n + \Delta x_n)$$

Pertanto se φ è sufficientemente regolare,

$$\begin{aligned} E_{dati} &= \Delta y = \varphi(x_1 + \Delta x_1, ..., x_n + \Delta x_n) - \varphi(x_1, ..., x_n) \\ &\simeq \sum_{i=1}^{n} \frac{\partial \varphi(x_1, ..., x_n)}{\partial x_i} \Delta x_i \end{aligned}$$

Inoltre se $\varphi(x_1, ..., x_n) \neq 0$ e $x_i \neq 0$,

$$\begin{aligned} \epsilon_{dati} &= \frac{\Delta y}{y} = \frac{\varphi(x_1 + \Delta x_1, ..., x_n + \Delta x_n) - \varphi(x_1, ..., x_n)}{\varphi(x_1, ..., x_n)} \\ &\simeq \sum_{i=1}^{n} \frac{\partial(\varphi(x_1, ..., x_n))}{\partial x_i} \frac{x_i}{\varphi(x_1, ..., x_n)} \epsilon_{x_i} \end{aligned}$$

In tal caso si dice indice di condizionamento del problema la somma dei valori assoluti dei coefficienti dei singoli errori sui dati iniziali:

$$I_{cond} = \sum_{i=1}^{n} K(x_i, \varphi) = \sum_{i=1}^{n} \left| \frac{\partial \varphi(x_1, ..., x_n)}{\partial x_i} \frac{x_i}{\varphi(x_1, ..., x_n)} \right|$$