Metodi a più passi - Esempi

- Metodo di Adams-Bashforth a 3 passi

$$u_{n+1} = u_n + \frac{h}{12}[23f_n - 16f_{n-1} + 5f_{n-2}] \,.$$

- Metodo di Adams-Moulton a 3 passi

$$u_{n+1} = u_n + \frac{h}{24}[9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}] \,.$$

- Metodo del punto medio

$$u_{n+1} = u_{n-1} + 2hf_n \,.$$

- Metodo BDF a 3 passi

$$u_{n+1} = \frac{18}{11}u_n - \frac{9}{11}u_{n-1} + \frac{2}{11}u_{n-2} + h\frac{6}{11}f_{n+1} \,.$$

$$f_k := f(t_k, u_k)$$

Metodi a più passi

Per $n = k, k+1, \ldots, N-1$

$$u_{n+1} = \sum_{j=0}^{k} a_j u_{n-j} + h \sum_{j=-1}^{k} b_j f_{n-j} \,.$$

- Metodo a $k+1$ passi se $a_k \neq 0$ o $b_k \neq 0$.
- Per innescare un metodo a $k+1$ passi servono $k+1$ condizione iniziali $u_0, \ldots, u_k$.
- Si prende $u_0 = y_I$ e i valori $u_1, \ldots, u_k$ si calcolano usando un metodo ad un passo di ordine elevato.
- Se $b_{-1} \neq 0$ il metodo è implicito e se $b_{-1} = 0$ il metodo è esplicito.

Esercizio

- Scrivere una funzione di Matlab per approssimare la soluzione del problema di Cauchy

$$\begin{cases} y'(t) = f(t, y(t)) & t \in [t_0, t_0 + T] \\ y(t_0) = y_0 \end{cases}$$

usando il metodo di Adams-Bashforth a quattro passi:

$$\begin{cases} u_{n+1} = u_n + \dfrac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) & n = 3, \ldots, N-1 \\ u_0 = y_0 \end{cases}$$

dove $h = T/N$ e $t_n = t_0 + nh$ per $n = 0, \ldots, N$.

- Per calcolare u_1, u_2 e u_3 usare
  - il metodo di Eulero;
  - il metodo di Heun;
  - il metodo di Runge-Kutta 4.
- Approssimare la soluzione del problema di Cauchy

$$\begin{cases} y' = -ty + \frac{4t}{y} & t \in [0, 1] \\ y(0) = 1 \end{cases}$$

e stimare per i tre casi l'ordine di convergenza del metodo di Adams-Bashforth a quattro passi.
(Soluzione esatta $y(t) = \sqrt{4 - 3e^{-t^2}}$.)

Assoluta stabilità - Esercizio

Approssimare la soluzione del problema di Cauchy

$$\begin{aligned} &y' = -5y \quad t > 0 \\ &y(0) = 1 \end{aligned}$$

usando

- Il metodo di Eulero in avanti con passo $h = 0.41$, $h = 0.4$, $h = 0.39$, $h = 0.2$.
- Il metodo di Crank-Nicolson con passo $h = 1$, $h = 0.5$, $h = 0.25$.
- Il metodo del punto medio con $h = 0.2$, $h = 0.1$, $h = 0.05$, $h = 0.01$.

Regione di assoluta stabilità

Il seguente script disegna la regione di assoluta stabilità del metodo a piu passi

$$u_{n+1} = \sum_{j=0}^{k} a_j u_{n-j} + h \sum_{j=-1}^{k} b_j f_{n-j} \qquad n = k, k+1, \ldots, N-1\,.$$

Chiede un vettore **a** coi coefficienti $a_0, \ldots,$ a_k e un vettore **b** coi coefficienti $b_{=1},$ $b_0, \ldots,$ b_k.

```
clear
a=input('Vettore a: ');
a=[-1 a];
b=input('Vettore b: ');
[X,Y]=meshgrid(-5:.05:5);
[m,n]=size(X);
r=zeros(m,n);
q=r;
for i=1:m
    for j=1:n
        p=a+(X(i,j)+1i*Y(i,j))*b;
        r(i,j)=max(abs(roots(p)));
        if r(i,j)<1
            q(i,j)=r(i,j);
        else
            q(i,j)=2;
        end
    end
end
contour(X,Y,q,1)
if q(1,1) < 1
   legend('Esterno')
end
```

Metodi impliciti

$$u_{n+1} = u_n + \frac{h}{12}\left[5f(t_{n+1}, u_{n+1}) + 8f(t_n, u_n) - f(t_{n-1}, u_{n-1})\right]$$

Ad ogni passo dobbiamo risolver un'equazione non lineare:

$$u_{n+1} = \Phi(u_{n+1}) \tag{1}$$

Per risolvere equazioni non lineari di questo tipo (problema di punto fisso) l'idea è costruire una succession nel modo seguente:

$$\begin{aligned} & u_{n+1}^0 \text{ assegnato} \\ & \texttt{for } \ k \geq 0 \\ & \quad u_{n+1}^{k+1} = \Phi(u_{n+1}^k) \end{aligned}$$

Se questa successione converge, il limite è soluzione di (1).

Metodi impliciti

- Per scegliere u_{n+1}^0 si usa un metodo esplicito.
- Se $|u_{n+1}^{k+1} - u_{n+1}^k| < \epsilon$ STOP $\rightsquigarrow u_{n+1} = u_{n+1}^{k+1}$.

$$u_{n+1}^0 = u_n + \frac{h}{2}\left[3f(t_n, u_n) - f(t_{n-1}, u_{n-1})\right]$$

$$\texttt{while} \;\; |u_{n+1}^{k+1} - u_{n+1}^k| \geq \epsilon$$

$$\quad u_{n+1}^{k+1} = u_n + \frac{h}{12}\left[5f(t_{n+1}, u_{n+1}^k) + 8f(t_n, u_n) - f(t_{n-1}, u_{n-1})\right]$$

$$\texttt{end}$$

$$u_{n+1} = u_{n+1}^{k+1}$$

Metodi predictor corrector

L'iterazione di punto fisso si ferma dopo un numero fisso di passi:

$$
\begin{array}{l}
u_{n+1}^{0} \text{ assegnato} \\
\texttt{for } \; k = 1 : K \\
\quad u_{n+1}^{k} = \Phi(u_{n+1}^{k-1}) \\
\texttt{end} \\
u_{n+1} = u_{n+1}^{K}
\end{array}
$$

Per esempio se $K = 1$

$$
\begin{aligned}
u_{n+1}^{0} &= u_n + \tfrac{h}{2}\,[3f(t_n, u_n) - f(t_{n-1}, u_{n-1})] \\
u_{n+1} &= u_n + \tfrac{h}{12}\left[5f(t_{n+1}, u_{n+1}^{0}) + 8f(t_n, u_n) - f(t_{n-1}, u_{n-1})\right]
\end{aligned}
$$