Equazioni differenziali ordinarie

Lucia Gastaldi

Dipartimento di Matematica,
http://dm.ing.unibs.it/gastaldi/

9 marzo 2008

Outline

Il problema di Cauchy

Problema di Cauchy

Sia I un intervallo di $\mathbb{R}$, data $f : I \times \mathbb{R} \to \mathbb{R}$, trovare $y : I \to \mathbb{R}$ derivabile tale che

$$(\mathbf{C}) \begin{cases} y'(t) = f(t, y(t)) & t \in I \\ y(t_0) = y_0. \end{cases}$$

Teorema

Se la funzione f è **continua** in $I \times \mathbb{R}$ e **lipschitziana** rispetto a y, cioè esiste $L > 0$ tale che

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2| \quad \forall t \in I, \quad \forall y_1,\ y_2 \in \mathbb{R},$$

allora esiste una ed una sola $y : I \to \mathbb{R}$ soluzione del **problema di Cauchy**.

Dipendenza continua dai dati

Consideriamo il seguente problema:

$$\begin{cases} z'(t) = f(t, z(t)) + \delta(t) & \text{per } t \in I \\ z(t_0) = y_0 + \delta_0. \end{cases}$$

Definizione

Sia I un insieme limitato. Il problema di Cauchy si dice **stabile** se per ogni perturbazione $(\delta_0, \delta(t))$ tale che:

$$|\delta_0| < \varepsilon, \quad \max_{t \in I} |\delta(t)| < \varepsilon, \quad \text{con } \varepsilon > 0,$$

$$\text{vale:} \quad \exists C > 0 : \quad \max_{t \in I} |y(t) - z(t)| < C\varepsilon.$$

Dipendenza continua dai dati

Proposizione

Sia $I = [0, T]$, $f : I \times \mathbb{R} \to \mathbb{R}$ continua su $I \times \mathbb{R}$ e sia L la costante di Lipschitz di f rispetto a y, cioè vale

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2| \quad \forall (t, y_1), (t, y_2) \in I \times \mathbb{R}.$$

Allora il problema di Cauchy (**C**) è stabile e vale la seguente maggiorazione

$$\max_{t \in I} |y(t) - z(t)| \leq \frac{e^{LT} - 1}{L} \max_{t \in I} |\delta(t)| + e^{LT} |\delta_0|.$$

Il metodo di Eulero esplicito

Consideriamo:

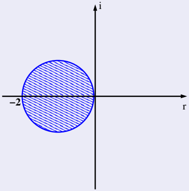
- una partizione dell'intervallo $I = [t_0, T]$ in un numero finito di intervalli N;
- $h = (T - t_0)/N$ **passo di discretizzazione**;
- $t_n = t_0 + hn$

In ogni punto t_n si cerca un valore u_n che *approssimi* il valore di $y_n = y(t_n)$.

Metodo di Eulero in avanti

$$u_{n+1} = u_n + hf(t_n, u_n) \quad n = 0, 1, \ldots, N_h.$$

- Metodo esplicito.
- Metodo del primo ordine: $\tau_n(h) = \frac{h}{2} y''(\xi_n)$.
- **Regione di stabilità assoluta**: $|1 + h\lambda| < 1$.



Function Eulero esplicito

Dato $N \in \mathbb{R}$. Si consideri passo costante $h = (T - t_0)/N$.
La function `euleroesp` fornisce la soluzione di un problema di Cauchy:

```
[t,u]=euleroesp(f,t0,T,y0,N)
```

dove

- `f` è il nome di una function che contiene l'espressione di $f(t, y)$ in funzione di `t, y`.
- `t, u` sono i vettori che contengono i valori di t_n e u_n rispettivamente per $n = 0, \ldots, N$.
- `t0, T` sono gli estremi dell'intervallo I.
- `N` il numero dei passi da effettuare a passo costante.
- `y0` è il valore iniziale.

Esercizio

Esercizio

Sia `N=[20 40 60 80 100 150 200 400]`.
Scrivere un programma di tipo script che per ogni valore di `N`:

- calcola, usando la function `euleroesp`, la soluzione del seguente problema di Cauchy

$$\begin{cases} y'(t) = t - 2y \quad 0 < t < 20 \\ y(0) = 0.75. \end{cases}$$

- riporta il grafico della soluzione esatta e della soluzione approssimata in una stessa figura, essendo la soluzione esatta $y(t) = e^{-2t} + \frac{1}{2}t - \frac{1}{4}$;
- calcola l'errore: $err(\mathtt{N}) = \max_{1 \le n \le N} |u_n - y(t_n)|$.

Fare il grafico in scala bilogaritmica dell'errore in funzione di `N`.
Calcolare l'ordine di convergenza.

Traccia dell'esercizio

- Assegnare `N=[20 40 60 80 100 150 200 400]`.
- Per ciascun valore di `N` (`for i=1:length(N)`):
  - Calcolare la soluzione dell'equazione differenziale con il comando `[t,u]=euleroesp(f,t0,T,y0,N)`.
  - Valutare la soluzione esatta `sol` nel vettore `t`.
  - Plottare la soluzione esatta e quella approssimata.
  - Calcolare l'errore relativo: `E(i)=norm(sol-u,inf)/norm(sol,inf)`.
- Plottare l'errore in scala bilogaritmica con il comando: `loglog(N,E)`.
- Calcolare l'ordine del metodo `p` con il comando

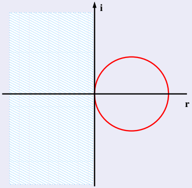
```
p=log(E(1:end-1)./E(2:end))./log(N(2:end)./N(1:end-1))
```

Il metodo di Eulero implicito

Metodo di Eulero all'indietro

$$u_{n+1} = u_n + hf(t_{n+1}, u_{n+1}) \quad n = 0, 1, \ldots, N_h.$$

- Metodo implicito.
- Metodo del primo ordine: $\tau_n(h) = \frac{h}{2} y''(\xi_n)$.
- **Regione di stabilità assoluta**: $|1 - h\lambda| > 1$.



Osservazione

ad ogni passo si deve risolvere l'**equazione non lineare**

$$\xi - u_n - hf(t_{n+1}, \xi) = 0,$$

usando uno dei metodi per il calcolo degli zeri di una funzione.

Il metodo di Crank-Nicolson o dei trapezi

Metodo di Crank-Nicolson o dei trapezi

$$u_{n+1} = u_n + \frac{h}{2}(f(t_n, u_n) + f(t_{n+1}, u_{n+1})).$$

- Metodo implicito.
- Metodo del secondo ordine: $\tau_n(h) = -\frac{h^2}{12} y'''(\xi_n)$.
- **Regione di stabilità assoluta**: $\forall h \mathrm{Re} \lambda < 0$.

Derivazione: dalla formula di quadratura dei trapezi si ottiene:

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + \int_{t_n}^{t_{n+1}} f(\tau, y(\tau)) d\tau \\ &\cong y(t_n) + \frac{h}{2}(f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))). \end{aligned}$$

Il metodo di Heun

Metodo di Heun

$$u_{n+1} = u_n + \frac{h}{2}\left(f(t_n, u_n) + f(t_{n+1}, u_n + hf(t_n, u_n))\right)$$

- Metodo esplicito.
- Metodo del secondo ordine: $\tau_n(h) = -Ch^2 y'''(\xi_n)$.
- **Regione di stabilità assoluta**: $-2 < h\lambda < 0$.

Derivazione: dalla formula di quadratura dei trapezi

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + \int_{tn}^{t_{n+1}} f(t, y(t))\,dt \\ &\approx y(t_n) + \frac{h}{2}\left(f(t_n, u_n) + f(t_{n+1}, y(t_{n+1}))\right). \end{aligned}$$

Si valuta $y(t_{n+1})$ con Eulero esplicito $y(t_{n+1}) \approx u_n + hf(t_n, u_n)$

Il metodo di Runge-Kutta del 4^o ordine

Metodo di Runge-Kutta del 4^o ordine

$$
\begin{aligned}
u_{n+1} &= u_n + \frac{h}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right)\\
K_1 &= f(t_n, u_n)\\
K_2 &= f(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_1)\\
K_3 &= f(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_2)\\
K_4 &= f(t_n + h, u_n + hK_3)
\end{aligned}
$$

- Metodo esplicito.
- Metodo del quarto ordine: $\tau_n(h) = -Ch^4 y^{(5)}(\xi_n)$.
- **Regione di stabilità assoluta**: $|1 + q + q^2/2 + q^3/12 + q^4/24| < 1$ essendo $q = h\lambda$.

Convergenza dei metodi numerici

Esercizio

Analizzare il comportamento dell'errore in funzione di h per i metodi di Eulero implicito, Heun, di Crank-Nicolson e di Runge-Kutta. A tal scopo usare i comandi

```
[t,u]=euleroimp(f,t0,T,y0,N,df,p1,p2,...)
[t,u]=Heun(f,t0,T,y0,N,p1,p2,...)
[t,u]=cranknic(f,t0,T,y0,N,df,p1,p2,...).
[t,u]=RK4(f,t0,T,y0,N,p1,p2,...).
```

Valutare l'ordine di convergenza per ciascun metodo.

Nota bene

- `df` è il nome della function che contiene la derivata rispetto a y di `f`.
- `p1,p2,...` sono parametri che si possono passare alla funzione `f`.

Uso di parametri nelle funzioni

Sia $\lambda \in \mathbb{R}$, si vuole calcolare la soluzione del problema di Cauchy

$$y'(t) = t - \lambda y \quad t \in [0,4], \qquad y(0) = 1.$$

Due modi:

- Si assegna la funzione con il comando
  `f=inline('t-lam*y','t','y','lam')`
  per calcolare la soluzione dell'equazione differenziale corrispondente a $\lambda = 2$, usando ad esempio la function `RK4`, il comando è:
  `[t,u]=RK4(f,0,4,N,1,2)`
- nel caso in cui si voglia usare un solutore di Matlab ad esempio `ode45`, la funzione deve essere assegnata con una **function** come segue:

```
function df=f(t,y,lam)
df=t-lam*y
```

e poi il comando è:

`[t,u]=ode45(@f,[0,4],1,[],2)`

Propagazione degli errori di macchina

Esercizio

Plottare in scala bilogaritmica l'errore di approssimazione che si ottiene risolvendo con il metodo di Runge-Kutta del 4^o ordine la seguente equazione differenziale:

$$y' = \frac{y}{t+1} + 5\frac{t+1}{1+25t^2} \quad \text{per } t \in [0,1], \qquad y(0) = 0$$

la cui soluzione esatta è $y(t) = (t+1)\arctan(5t)$.
Usare i seguenti valori per il numero di passi

```
N=[1000 5000 7500 10000 10500 11000 11500 12000]
```

.

Il metodo del punto medio

Metodo del punto medio

$$u_{n+1} = u_{n-1} + 2hf(t_n, u_n)$$

- Metodo esplicito a due passi: u_{n+1} si ottiene da u_n e u_{n-1}.
- Metodo del secondo ordine: $\tau_n(h) = -Ch^2 y^{(3)}(\xi_n)$.
- **Regione di stabilità assoluta**: nessun $q = h\lambda$.

Derivazione: dalla formula del punto medio applicata all'integrale da t_{n-1} a t_{n+1}

$$y(t_{n+1}) = y(t_{n-1}) + \int_{t_{n-1}}^{t_{n+1}} f(\tau, y(\tau))d\tau \cong y(t_{n-1}) + 2hf(t_n, y(t_n)).$$

Inizializzazione

Si assegnano u_0 e u_1: $u_0 = y_0$, $u_1 = u_0 + hf(t_0, u_0)$.

I metodi di Adams espliciti

Metodo di Adams-Bashforth a due passi

$$u_{n+1} = u_n + \frac{h}{2}(3f(t_n, u_n) - f(t_{n-1}, u_{n-1}))$$

- Metodo del secondo ordine: $\tau_n(h) = -Ch^2 y^{(3)}(\xi_n)$.

Metodo di Adams-Bashforth a tre passi

$$u_{n+1} = u_n + \frac{h}{12}(23f(t_n, u_n) - 16f(t_{n-1}, u_{n-1}) + 5f(t_{n-2}, u_{n-2}))$$

- Metodo del terzo ordine: $\tau_n(h) = -Ch^3 y^{(4)}(\xi_n)$.

I metodi di Adams impliciti

Metodo di Adams-Moulton a due passi

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

- Metodo del secondo ordine: $\tau_n(h) = -Ch^2 y^{(3)}(\xi_n)$.

Metodo di Adams-Moulton a tre passi

$$\begin{aligned} u_{n+1} = u_n + \frac{h}{24}(9f(t_{n+1}, u_{n+1}) &+ 19f(t_n, u_n) \\ &- 5f(t_{n-1}, u_{n-1}) + f(t_{n-2}, u_{n-2}) \end{aligned}$$

- Metodo del terzo ordine: $\tau_n(h) = -Ch^3 y^{(4)}(\xi_n)$.

I metodi Backward Differentiation Formulae

BDF a due passi

$$u_{n+1} = \frac{4}{3}u_n - \frac{1}{3}u_{n-1} + \frac{2}{3}hf(t_{n+1}, u_{n+1})$$

- Metodo implicito del secondo ordine: $\tau_n(h) = -Ch^2y^{(3)}(\xi_n)$.

BDF a tre passi

$$u_{n+1} = \frac{18}{11}u_n - \frac{9}{11}u_{n-1} + \frac{2}{11}u_{n-2} + \frac{6}{11}hf(t_{n+1}, u_{n+1})$$

- Metodo implicito del terzo ordine: $\tau_n(h) = -Ch^3y^{(4)}(\xi_n)$.

Regione di assoluta stabilità'

Esercizio

Risolvere il problema modello

$$y' = \lambda y, t \in [0, 10], \qquad y(0) = 1$$

per $\lambda = -1, -5$, usando le function `euleroesp`, `euleroimp`, `Heun`, `RK4`, `cranknic` e `ptomedio` con $N = 10, 20, 40, 50, 60, 90, 100, 110, 200, 400$.

Risolutori di equazioni differenziali ordinarie

Problemi non stiff

`ode45`	Metodo di Runge-Kutta (4,5).
`ode23`	Metodo di Runge-Kutta (2,3).
`ode113`	Metodo di Adams-Bashforth-Moulton PECE.

Problemi stiff

`ode15s`	Metodo multistep basato su una formula di tipo BDF.
`ode23s`	Metodo ad un passo.

Altre opzioni

`odeset`	Crea o modifica le OPTIONS.
`odeplot`	Grafico della soluzione.
`odephas2`	Grafico del piano delle fasi in 2D.
`odephas3`	Grafico del piano delle fasi in 3D.

Come si risolve una equazione differenziale usando i solutori di Matlab

- Scrivere una function che accetta due argomenti `t` e `y` e restituisce il valore della funzione

```
function dy=F(t,y)
dy=(1-t*y-t^2*y^2)/t^2;

F=inline('1-t*y-t^2*y^2)/t^2','t','y');
```

- Applicare un solutore mediante i comandi
  ```
  [t,u] = ode23 (@F, [t0 T], y0)
  [t,u] = ode23 (F, [t0 T], y0)
  ```
- Usare il comando `plot` per vedere i risultati: `plot(t,u)`

Come usare i solutori di ODE in Matlab

Il comando più semplice per risolvere un'equazione differenziale è:

```
[t,u] = solver (odefun,tspan, y0)
```

Input

`odefun`	function in cui si valuta $f(t, y)$
`tspan`	vettore contenente gli estremi di integrazione
`y0`	dato iniziale (vettore colonna)

Output

`t`	vettore colonna degli istanti t in cui viene calcolata la soluzione approssimata
`u`	array contenente la soluzione, le righe sono le componenti di y ad un certo istante t.

Argomenti addizionali

```
[t,u] = solver (odefun,tspan, y0,options,p1,p2,...)
```

`options` struttura che contiene i parametri per cambiare le proprietà di default del solutore

`p1,p2,...` parametri che si possono passare alla `odefun`.

Per definire le `options` si usa il comando `odeset`.

```
>> odeset
```

fornisce i valori di default e il nome delle varibili che si possono definire.

odeset

Variabile	Valore	descrizione
RelTol	`1.e-3`	tolleranza per l'errore relativo
AbsTol	`1.e-6`	tolleranza per l'errore assoluto
MaxStep	`|tspan|/10`	valore massimo per il passo
InitialStep	calcolato	passo iniziale scelto
OutputFcn	`Function`	controlla l'output
OutputFcn	`odeplot`	grafico in funzione di t
OutputFcn	`odephas2`	plot del piano delle fasi
OutputFcn	`odephas3`	plot del piano delle fasi in 3D
OutputSel	vettore di interi	specifica le componenti del vettore soluzione che si vogliono come output

Oscillatori accoppiati

Consideriamo due oscillatori, ciascuno dei quali ha una frequenza naturale data dalle costanti ω_1 e ω_2. Lo stato degli oscillatori è descritto dall'angoli di fase θ_1 e θ_2. Le seguenti equazioni descrivono l'accoppiamento dei due oscillatori:

$$\theta_1' = \omega_1 + k_1 \sin(\theta_2 - \theta_1)$$
$$\theta_2' = \omega_2 + k_2 \sin(\theta_1 - \theta_2).$$

Le costanti k_1 e k_2 indicano la forza delle costanti di accoppiamento. Questo modello può essere usato come punto di partenza per lo studio dei processi di sincronizzazione di fase.

In particolare consideriamo il seguente sistema:

$$\theta_1' = 1 + \sin(\theta_2 - \theta_1)$$
$$\theta_2' = 1.5 + \sin(\theta_1 - \theta_2)$$
$$\theta_1(0) = 3, \quad \theta_2(0) = 0.$$

Calcolare la soluzione del sistema nell'intervallo $[0, 8]$.
Calcolare la soluzione del sistema nell'intervallo $[0, 200]$.

Risultati non attendibili di ode45

Esercizio

Si consideri la seguente equazione differenziale:

$$y'(t) = 100 - y(t) \text{ per } t \in [0, 200], \qquad y'(0) = 0,$$

la cui soluzione è $y(t) = 100(1 - e^{-t})$.

- Calcolare la soluzione con `ode45`.
- Confrontare la soluzione con la soluzione esatta calcolando l'errore relativo.
- Calcolare la soluzione ottenuta a passo costante con `RK4` con 250 passi e confrontarla con la soluzione esatta.
- Calcolare la soluzione con `ode15s`, confrontarla con la soluzione esatta e con le soluzioni precedentemente ottenute.
- Verificare il comportamento della function `ode45` plottando i valori dei passi.

Sistema di equazioni differenziali

Problema

Siano a, b, c e d numeri reali positivi. Cercare $y_1(t)$ e $y_2(t)$ tali che risolvano nell'intervallo $[0, 7]$ il seguente sistema di equazioni differenziali ordinarie:

$$\begin{cases} y_1' = (a - by_2)y_1 \\ y_2' = (-c + dy_1)y_2 \\ y_1 = \alpha \\ y_2 = \beta. \end{cases}$$

Porre: $a = 1,\ b = 1,\ c = 2,\ d = 3$
$\alpha = 0.5, 0.7, 1.4, 2.6$
$\beta = 1$

Come si risolve il sistema

- Scrivere una function che accetta due argomenti `t` e `y` e restituisce il valore della funzione a valori vettoriali

```
function dy=lotkavolterra(t,y)
a=1; b=1; c=2; d=3;
dy=[(a-b*y(2))*y(1); (-c+d*y(1))*y(2)];
```

- Applicare un solutore mediante il comando
  `[t,u] = ode45 (@lotkavolterra, [0 7], [`α`;`β`])`
- Usare il comando `plot` per vedere i risultati:
  `plot(t,u)` oppure `plot(u(:,1),u(:,2))`
- Per ottenere il piano delle fasi si può procedere come segue:
  `options=odeset('OutputFcn','odephas2')`
  `[t,u]=ode45(@lotkavolterra,[0 7],[`α`;`β`],options)`

Equazione differenziale di ordine n

Problema

Sia $f : I \times \mathbb{R}^n \to \mathbb{R}$. Consideriamo l'equazione differenziale:

$$\begin{cases} y^{(n)}(t) = f(t, y(t), y'(t), \dots, y^{(n-1)}(t)) \quad t \in I \\ y(t_0) = \alpha_1 \\ y'(t_0) = \alpha_2 \\ \dots \\ y^{(n-1)}(t_0) = \alpha_n \end{cases}$$

Equazione differenziale di ordine n

Il problema di Cauchy per l'equazione differenziale di ordine n è equivalente ad un sistema differenziale del primo ordine.
Si pone

$$y_1(t) = y(t),\ y_2(t) = y'(t),\ \ldots\ ,\ y_n(t) = y^{(n-1)}(t).$$

Osserviamo che

$$\begin{aligned} y_2(t) &= y'(t) = y_1'(t), \\ &\ldots \\ y_n(t) &= y^{(n-1)}(t) = y_{n-1}'(t). \end{aligned}$$

Riduzione di un'equzione differenziale d'ordine n

Quindi si ottiene il seguente sistema di equazioni differenziali

$$\begin{cases} y_1'(t) = y_2(t) \\ y_2'(t) = y_3(t) \\ \dots \\ y_n'(t) = f(t, y_1(t), y_2(t), \dots, y_n(t)) \\ \\ y_1(t_0) = \alpha_1 \\ y_2(t_0) = \alpha_2 \\ \dots \\ y_n(t_0) = \alpha_n \end{cases}$$

Esempio: equazione di van der Pol

$$y'' - \mu(1 - y^2)y' + y = 0 \quad \text{dove } \mu > 0.$$

L'equazione si riduce al seguente sistema:

$$\begin{cases} y_1'(t) = y_2(t) \\ y_2'(t) = \mu(1 - y_1^2)y_2 - y_1 \end{cases}$$

Si costruisce la function che definisce il sistema:

```
function dy = vdpol(t,y,mu)
dy = [y(2); mu*(1-y(1)^2)*y(2)-y(1)];
```

Risolvere il sistema per $\mu = 1$ e $\mu = 50$ usando `ode45` e `ode15s`

```
[t,u]=ode45(@vdpol,[0 20],[2;0],[],mu)
[t,u]=ode15s(@vdpol,[0 300],[2;0],[],mu);
```

Esercizio - molle e masse

Si consideri il sistema meccanico composto da due corpi di massa 1 fissati tra di loro e a due supporti fissi da tre molle con costante di elasticità k. Le incognite del sistema y_1 e y_2 rappresentano lo spostamento rispetto alla posizione di equilibrio. La dinamica del sistema è governata dalle seguenti equazioni differenziali:

$$
\begin{array}{ll}
y_1''(t) = -ky_1(t) + k(y_2(t) - y_1(t)) & t \in [0, 10] \\
y_2''(t) = -k(y_2(t) - y_1(t)) - ky_2(t) & t \in [0, 10] \\
y_1(0) = 1 \qquad y_1'(0) = \sqrt{3k} & \\
y_2(0) = 1 \qquad y_1'(0) = -\sqrt{3k} &
\end{array}
$$

Trasformare il sistema in un sistema del primo ordine.
Calcolare la soluzione numerica mediante la function `ode45` plottando solo le componenti che danno la soluzione $(y_1(t), y_2(t))$.

Apollo e luna

Il seguente sistema di equazioni differenziali descrive il moto di un corpo in orbita attorno a due altri corpi molto più pesanti.

$$
\begin{aligned}
x'' &= 2y' + x - \frac{\mu^*(x+\mu)}{r_1^3} - \frac{\mu(x-\mu^*)}{r_2^3} \\
y'' &= -2x' + y - \frac{\mu^* y}{r_1^3} - \frac{\mu y}{r_2^3} \\
\mu &= \frac{1}{82.45}, \quad \mu^* = 1-\mu \\
r_1 &= ((x+\mu)^2 + y^2)^{1/2} \quad r_2 = ((x-\mu^*)^2 + y^2)^{1/2}
\end{aligned}
\tag{1}
$$

Si studiano le soluzioni periodiche di questo problema.
È noto che le seguenti condizioni iniziali

$$
\begin{array}{ll}
x(0) = 1.2 & x'(0) = 0 \\
y(0) = 0 & y'(0) = -1.04935751
\end{array}
$$

danno luogo ad una soluzione periodica di periodo $T = 6.19216933$.

Apollo e luna (cont)

Il sistema di equazioni differenziali (1) potrebbe rappresentare, ad esempio, il moto della capsula Apollo in orbita intorno alla terra e alla luna. I tre corpi determinano un piano nello spazio e fissiamo su questo piano un sistema di coordinate come segue. L'asse x è la retta che congiunge i due corpi pesanti, l'origine viene posta nel loro baricentro e la loro distanza è presa come unità di misura. Quindi se μ è il rapporto fra la massa della luna e quella della terra, allora la luna e la terra si trovano nei punti di coordinate $(1-\mu, 0)$ e $(-\mu, 0)$, rispettivamente e il sistema di coordinate si muove in accordo con la luna che ruota intorno alla terra. Si suppone che il terzo corpo, l'Apollo, abbia massa trascurabile rispetto ai primi due e che la sua posizione sia una funzione del tempo $(x(t), y(t))$. Le equazioni che governano il moto dell'Apollo possono essere dedotte dalla legge del moto di Newton e dalla legge di gravitazione. I termini con le derivate prime nell'equazione provengono dal moto del sistema di coordinate ruotante.

Apollo e luna (cont)

Esercizio

- Ricondurre il sistema di 2 equazioni differenziali del secondo ordine ad un sistema di 4 equazioni differenziali del primo ordine.
- Usare le function di Matlab **ode45** oppure **ode23s** per calcolare la soluzione.
- Riportare l'orbita dell'Apollo in un grafico.
- Osservato che con i valori di default l'orbita non è perfettamente periodica, modificare la tolleranza relativa per l'errore ponendo nelle opzioni `RelTol=1.e-5`.

Suggerimenti Usare le seguenti options per produrre il grafico desiderato:
`options=odeset('OutputFcn','odephas2','OutputSel',[n m])`
essendo `n` e `m` gli indici delle incognite associate a $x(t)$ e a $y(t)$.

Equazione del calore

La temperatura di una sbarra omegenea, isotropa e di sezione costante, soggetta ad una sorgente di calore disposta lungo tutta la sbarra è governata dalla seguente equazione alle derivate parziali:

$$
\begin{array}{l}
u_t - u_{xx} = f(t,x) \quad t \in [0, T],\ x \in [a, b] \\
u(t,a) = \alpha(t), \quad u(t,b) = \beta(t) \qquad t \in [0, T] \\
u(0,x) = \varphi(x) \qquad x \in [a, b]
\end{array}
$$

essendo:

- $u(t,x)$ la temperatura della sbarra nel punto x all'istante t;
- u_t la derivata parziale del primo ordine rispetto a t;
- u_{xx} la derivata parziale del secondo ordine rispetto a x;
- $f(t,x)$ la sorgente di calore;
- $\alpha(t)$ e $\beta(t)$ la temperatura che viene assegnata agli estremi della sbarra;
- $\varphi(x)$ la temperatura all'istante iniziale.

Discretizzazione nella variabile x

Suddividiamo l'intervallo $[a, b]$ in $N + 1$ parti uguali ($h_x = \dfrac{b-a}{N+1}$), indichiamo con $x_i = a + ih_x$ i punti di suddivisione e poniamo $\mathbf{u}_i(t) \approx u(t, x_i)$.

Applicando le differenze finite per approssimare la derivata seconda rispetto a x si ottiene:

$$\mathbf{u}_i'(t) - \frac{\mathbf{u}_{i-1}(t) - 2\mathbf{u}_i(t) + \mathbf{u}_{i+1}(t)}{h_x^2} = f(t, x_i) \quad i = 1, \dots, N \;\; t \in [0, T]$$

$$\mathbf{u}_0(t) = \alpha(t), \quad \mathbf{u}_{N+1}(t) = \beta(t) \qquad t \in [0, T]$$

$$\mathbf{u}_i(0) = \varphi(x_i) \quad i = 1, \dots, N.$$

Discretizzazione nella variabile x (cont)

Sostituendo $\mathbf{u}_0(t)$ e $\mathbf{u}_{N+1}(t)$ nelle equazioni per $i = 1$ e $i = N$ rispettivamente, si ottiene il sistema di equazioni differenziali ordinarie

$$\begin{array}{l} \mathbf{U}'(t) = -A\mathbf{U}(t) + b(t) \quad t \in [0, T] \\ \mathbf{U}(0) = \Phi \end{array} \qquad (2)$$

essendo $\mathbf{U}(t) = (\mathbf{u}_1(t), \mathbf{u}_2(t), \dots, \mathbf{u}_N(t))^T$, $\Phi = (\varphi(x_1), \varphi(x_2), \dots, \varphi(x_N))^T$, e

$$\begin{array}{ll} b_1(t) = f(t, x_1) + \alpha(t)/h_x^2 & \\ b_i(t) = f(t, x_i) & i = 2, \dots, N-1 \\ b_N(t) = f(t, x_N) + \beta(t)/h_x^2. & \end{array}$$

Problema stiff

La matrice A è simmetrica e definita positiva con autovalore massimo

$$\lambda_{max} = \frac{2}{h_x^2}\left(1 - \cos\left(\frac{N}{N+1}\pi\right)\right).$$

Il sistema (2) è un tipico esempio di **problema stiff**.

Applicazione dei metodi per equazioni differenziali ordinarie

La function `calore` fornisce la soluzione di (2) mediante i metodi di Eulero esplicito, di Eulero implicito e di Crank-Nicolson.

```
[t,u]=calore(f,phi,alfa,beta,t0,T,a,b,N,M,metodo)
```

`t`	vettore dei tempi
`u`	soluzione calcolata
`f(t,x)`	sorgente di calore
`phi`	dato iniziale $\varphi(x)$
`alfa, beta`	condizioni al bordo funzioni di t
`t0, T`	estremi dell'intervallo temporale
`a,b`	estremi dell'intervallo spaziale
`N`	numero di punti interni nella x
`M`	numero di intervalli di tempo
`metodo=1`	Eulero esplicito
`metodo=2`	Eulero implicito
`metodo=3`	Crank-Nicolson

Esercizio

Porre:

$$
\begin{aligned}
&f(t,x) = 0 \quad \forall t \in [0,1],\ x \in [0,1]\\
&\alpha(t) = \beta(t) = 0 \quad \forall t \in [0,1]\\
&\varphi(x) = x(1-x) \quad \forall x \in [0,1]
\end{aligned}
$$

- Porre $N = 10$ e risolvere con il metodo di Eulero esplicito. Osservare che per M piccolo si producono oscillazioni della soluzione. Per soddifare la condizione di assoluta stabilità bisogna prendere $M = 2 * (N+1)^2$.
- Porre $N = 10, 50, 100$. Calcolare la soluzione con i metodi di Eulero implicito e di Crank-Nicolson con alcuni valori di M. Osservare che la scelta $M = N$ fornisce una buona soluzione.