Analisi Numerica

Debora Botturi

ALTAIR

http://metropolis.sci.univr.it

Introduzione

Argomenti

- Rappresentazione di sistemi con variabili di stato;
- Tecniche di integrazione numerica

Obiettivo: risolvere sistemi di equazioni differenziali con metodi numerici.

Osservazioni

- Non é sempre pratico analizzare le equazioni differenziali attraverso la loro forma esplicita
- In tali casi possono essere usati metodi numerici per trovare una soluzione per un’equazione differenziale
- La soluzione trovata é generalmente in forma di grafico od insieme di numeri
- Tali soluzioni possono quindi essere usate per analizzare casi di studio specifici
- La soluzione é data velocemente grazie alle tecniche usate (iterative che puntano all’azzeramento dell’errore)
- Ma senza una rappresentazione esplicita il sistema non puó essere capito e manipolato

Metodologia Generale

Rappresentazione a stati

- Il processo generale per analizzare un sistema prevede:
  - Mettere le equazioni in forma standard (variabili di stato)
  - Integrarle con un metodo numerico
- Ad ogni istante di tempo un sistema ha uno stato
- Per identificare le variabili di stato si usano i seguenti fattori
  - Le variabili dovrebbero descrivere gli elementi che immagazzinano energia
  - Le variabili devono essere indipendenti
  - Gli stati dovrebbero descrivere gli elementi del sistema

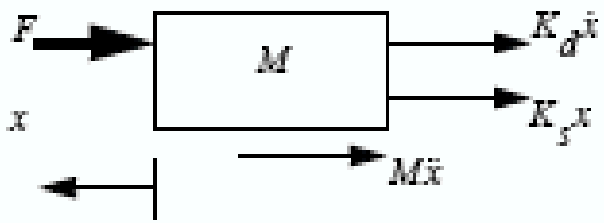
Rappresentazione a stati

- Gli stati del sistema vengono usati per scrivere un'equazione del primo ordine lineare:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

  - $x =$ vettore degli stati
  - $u =$ vettore di input
  - $A =$ matrice di transizione relativa agli stati
  - $B =$ matrice che relaziona l'input all'output
  - $y =$ valore che puó essere trovato direttamente
  - $C =$ matrice di transizione relativa agli stati
  - $D =$ matrice che relaziona l'input all'output
- L'equazione di output non é sempre richiesta, ma puó essere usata per calcolare nuovi valori di output

Esempio



$$\sum F_x = -F - K_d\dot{x} - K_s x = M\ddot{x}$$
$$-F - K_d\dot{x} - K_s x = M\ddot{x}$$

The equation is second-order, so two state variables will be needed. One obvious choice for a state variable in this equation is 'x'. The other choice can be the velocity, 'v'. Equation (1) defines the velocity variable. The velocity variable can then be substituted into the differential equation for the system to reduce it to first-order.

$$\dot{x} = v \qquad (1)$$
$$M\ddot{x} = -F - K_d\dot{x} - K_s x$$
$$M\dot{v} = -F - K_d v - K_s x$$
$$\dot{v} = x\left(\frac{-K_s}{M}\right) + v\left(\frac{-K_d}{M}\right) + \left(\frac{-F}{M}\right) \qquad (2)$$

Equations (1) and (2) can also be put into a matrix form similar to that given in Figure 4.1.

$$\frac{d}{dt}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-K_s}{M} & \frac{-K_d}{M} \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-F}{M} \end{bmatrix}$$

Nota: per avere un insieme risolvibile di equazioni differenziali dobbiamo avere lo stesso numero di equazioni e di variabili. Se abbiamo poche equazioni si deve sviluppare una equazione sfruttando relazioni non ancora usate. Se ci sono troppe equazioni la ridondanza deve essere eliminata.

Esercizio

Mettere l'equazione nella forma a stati:

$$F = M\ddot{x}$$

soluzione:

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= \frac{F}{M} \end{aligned}$$

Esempio

Introduzione

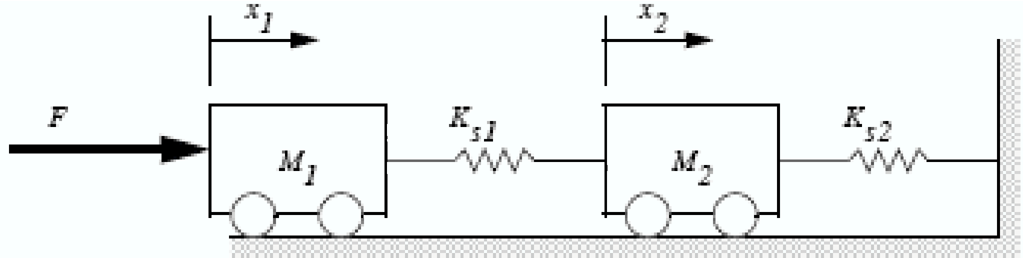
Metodologia Generale
- Rappresentazione a stati
- Rappresentazione a stati
- Esempio
- Esercizio
- Esempio
- Esercizio
- Esercizio
- Osservazioni
- Esempio
- Esempio

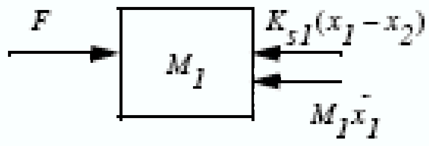
Integrazione Numerica

Serie di Taylor

Integrazione di Runge-Kutta

Risposta del Sistema





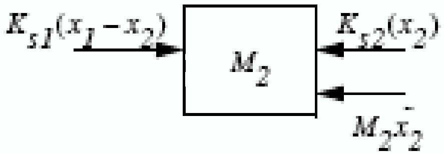
$$\overset{+}{\rightarrow}\sum F_x = F - K_{s1}(x_1 - x_2) = M_1\ddot{x}_1$$

$$M_1\ddot{x}_1 + K_{s1}x_1 - K_{s1}x_2 = F$$

$$\dot{x}_1 = v_1 \qquad (1)$$

$$M_1\dot{v}_1 + K_{s1}x_1 - K_{s1}x_2 = F$$

$$\dot{v}_1 = \frac{F}{M_1} - \frac{K_{s1}}{M_1}x_1 + \frac{K_{s1}}{M_1}x_2 \qquad (2)$$



$$\overset{+}{\rightarrow}\sum F_x = K_{s1}(x_1 - x_2) - K_{s2}(x_2) = M_2\ddot{x}_2$$

$$M_2\ddot{x}_2 + (K_{s1} + K_{s2})x_2 - K_{s1}x_1 = 0$$

$$\dot{x}_2 = v_2 \qquad (3)$$

$$M_2\dot{v}_2 + (K_{s1} + K_{s2})x_2 - K_{s1}x_1 = 0$$

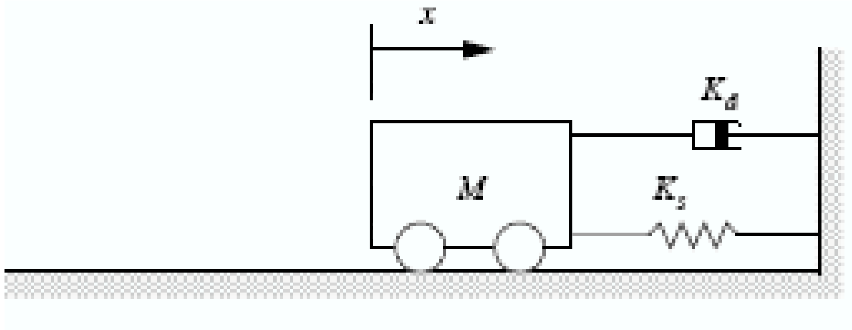
$$\dot{v}_2 = \frac{K_{s1}}{M_2}x_1 - \left(\frac{K_{s1} + K_{s2}}{M_2}\right)x_2 \qquad (4)$$

The state equations can now be combined in a matrix form.

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-K_{s1}}{M_1} & 0 & \frac{K_{s1}}{M_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{K_{s1}}{M_2} & 0 & \frac{-K_{s1} - K_{s2}}{M_2} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{F}{M_1} \\ 0 \\ 0 \end{bmatrix}$$

Esercizio

Sviluppare l'equazione a stati del seguente sistema

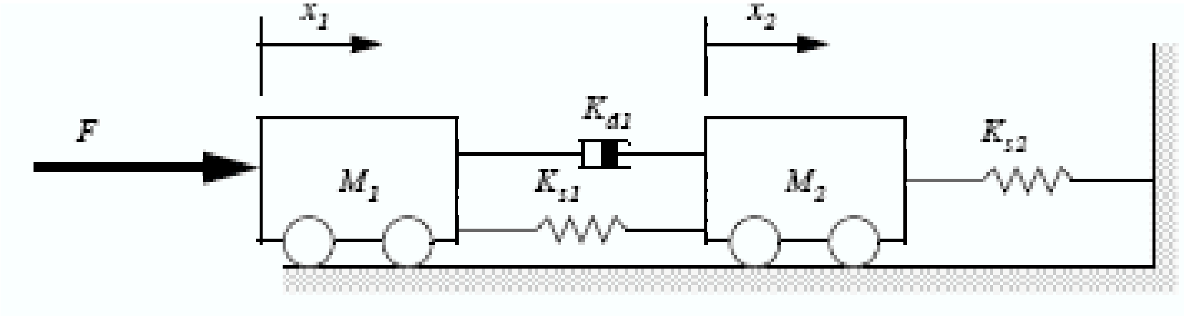


Soluzione:

$$
\begin{aligned}
\dot{x} &= v \\
\dot{v} &= v\frac{K_d}{M} + x\frac{K_s}{M}
\end{aligned}
$$

Esercizio

Sviluppare l'equazione a stati del seguente sistema



Soluzione:

$$
\begin{aligned}
\dot{x}_1 &= v_1 \\
\dot{x}_2 &= v_2 \\
\dot{v}_1 &= v_1 \frac{-K_{d1}}{M_1} + x_1 \frac{-K_{s1}}{M_1} + v_2 \frac{-K_{d1}}{M_1} + \frac{F}{M_1} \\
\dot{v}_2 &= v_2 \frac{-K_{d1}}{M_2} + x_2 \frac{-K_{s1} - K_{s2}}{M_2} + v_1 \frac{-K_{d1}}{M_2} + \frac{K_{s1}}{M_2}
\end{aligned}
$$

Osservazioni

- In alcuni casi le equazioni differenziali hanno piú di un termine di ordine piú elevato e quindi non possono essere ridotte (ad esempio una equazione del secondo ordine con due variabili derivate seconde non puó essere trasformata nella forma a stati)
- Si usa in questi casi una variabile fittizia che sostituisce le due variabili di piú alto grado
- Questo ad esempio succede nei sistemi meccanici quando le masse non sono considerate

Esempio

Data l'equazione:

$$3\dot{y} + 2y = 5\dot{u}$$

Passo 1: mettere entrambe le derivate del primo ordine dalla parte sinistra dell'equazione

$$3\dot{y} - 5\dot{u} = -2y$$

Passo 2: Sostituire la parte sinistra dell'equazione con una variabile fittizia

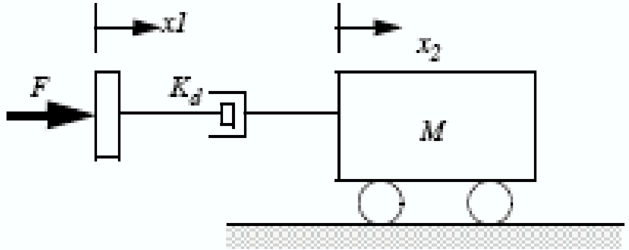
$$q = 3y - 5u \qquad \dot{q} = -2y$$

Passo 3: Risolvere l'equazione usando la variabile fittizia, quindi risolvere per y come fosse un'equazione di output

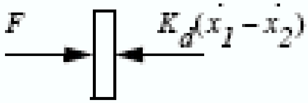
$$\dot{q} = -2y \qquad y = \frac{q + 5u}{3}$$

Esempio

In altri casi é possibile eliminare i termini ridondanti attraverso manipolazioni algebriche



The FBDs and equations are;



$$\xrightarrow{+} \sum F_x = F - K_d(\dot{x}_1 - \dot{x}_2) = 0$$

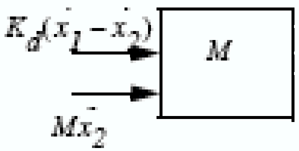
$$K_d(\dot{x}_1 - \dot{x}_2) = F \qquad (1)$$

$$q = x_1 - x_2$$

$$K_d(\dot{q}) = F$$

$$\dot{q} = \frac{F}{K_d} \qquad (2)$$

$$x_1 = x_2 + q \qquad (3)$$



$$\xrightarrow{+} \sum F_x = K_d(\dot{x}_1 - \dot{x}_2) = M\ddot{x}_2$$

$$F = M\ddot{x}_2$$

$$\dot{x}_2 = v_2 \qquad (4)$$

$$\dot{v}_2 = \frac{F}{M} \qquad (5)$$

The state equations (2, 4, 5) can be put in matrix form. The output equation (2) can also be put in matrix form.

$$\frac{d}{dt}\begin{bmatrix} q \\ x_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} q \\ x_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} \frac{F}{K_d} \\ 0 \\ \frac{F}{M} \end{bmatrix} \qquad \begin{bmatrix} x_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} q \\ x_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}$$

Integrazione Numerica

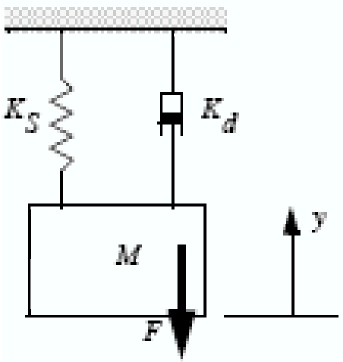
Osservazioni

- Calcoli ripetitivi possono essere usati per trovare una soluzione approssimativa di un insieme di equazioni differenziali
- Partendo da condizioni iniziali date l'equazione é risolta con piccoli passi temporali.
- Piú piccoli si prendono gli intervalli di tempo maggiore sará l'accuratezza del risultato ottenuto
- Il processo di analisi segue i seguenti passi:
  - Generare l'equazione generale che modella il sistema
  - Selezionare la variabile di stato
  - Riarrangiare l'equazione nella forma a stati
  - Aggiungere equazioni se necessario per renderla risolvibile
  - Calcolare e risolvere il sistema di equazioni

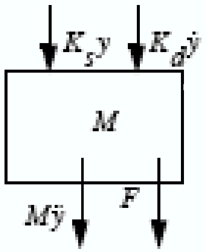
  Per quest'ultimo passo di analisi diversi tool sono stati implementati per risolvere equazioni differenziali in forma di stato.

Esempio

In altri casi é possibile eliminare i termini ridondanti attraverso manipolazioni algebriche



Step 1: Develop equations



$$\sum F_y = -F - K_d\left(\frac{d}{dt}\right)y - K_s y = M\left(\frac{d}{dt}\right)^2 y$$

$$F + K_d\left(\frac{d}{dt}\right)y + K_s y + M\left(\frac{d}{dt}\right)^2 y = 0$$

Step 2: We need to identify state variables. In this case the height is clearly a defining variable. We will also need to use the vertical velocity, because the acceleration is a second derivative (we can only have first derivatives). Using the height, y, and velocity, v, as state variables we may now proceed to rewriting the equations. (Note: this is just an algebraic trick, but essential when setting up these matrices.)

Step 3:

$$\left(\frac{d}{dt}\right)y = v$$

$$\left(\frac{d}{dt}\right)v = \frac{-F - K_d v - K_s y}{M}$$

Step 4: We put the equations into a state variable matrix form.

$$\begin{bmatrix} \left(\frac{d}{dt}\right)y \\ \left(\frac{d}{dt}\right)v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-K_S}{M} & \frac{-K_d}{M} \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \frac{F}{M}$$

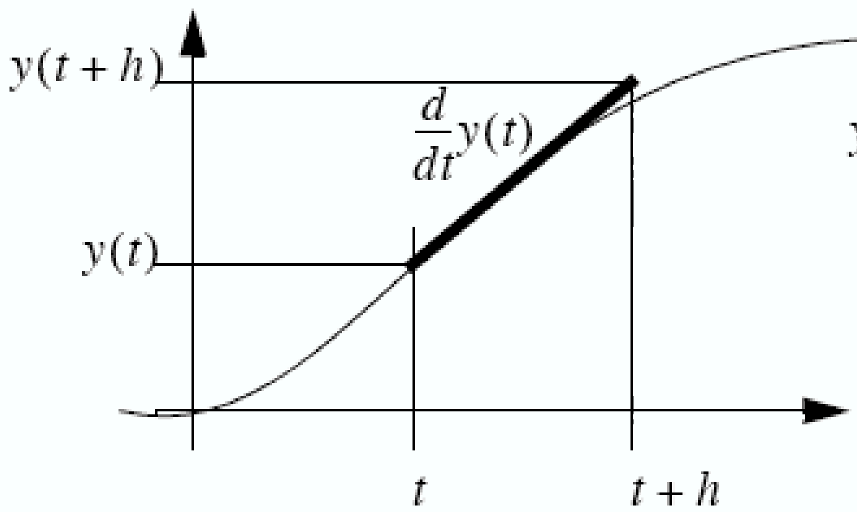
Selezioniamo alcuni valori per i parametri dell’equazione e la funzione di input

$$\left(\frac{d}{dt}\right)y = v_y$$

$$\left(\frac{d}{dt}\right)v_y = \frac{-F - K_d v_y - K_s y}{M} = -4e^{-0.5t}\sin(t) - 2v_y - 5y$$

Metodo di Eulero

- La forma piú semplice di integrazione numerica é il metodo di Eulero del primo ordine.
- Dato il valore corrente della funzione e la prima derivata, possiamo stimare il valore della funzione dopo poco tempo
- Conoscendo la posizione e la derivata prima possiamo calcolare un valore approssimativo dopo un breve tempo, h



$$y(t+h) \approx y(t) + h\frac{d}{dt}y(t)$$

Note: here the h value is the time step between integrations points. A smaller time step will increase the accuracy.

Esempio

Soluzione della equazione di Newton usando il metodo di Eulero. In questo esempio calcoliamo la velocitá integrando l'accelerazione causata dalla forza, L'accelerazione é usata direttamente nell'equazione di Eulero.

$$F = M\left(\frac{d}{dt}\right)v$$

we can create difference equations using simple methods.

$$\left(\frac{d}{dt}\right)v = \frac{F}{M}$$ *first rearrange equation*

$$v(t+h) = v(t) + h\left(\frac{d}{dt}\right)v(t)$$ *put this in the Euler equation*

$$v(t+h) = v(t) + h\left(\frac{F(t)}{M}\right)$$ *finally substitute in known terms*

We can now use the equation to estimate the system response. We will assume that the system is initially at rest and that a force of 1N will be applied to the 1kg mass for 4 seconds. After this time the force will rise to 2N. A time step of 2 seconds will be used.

i	t (sec)	F (N)	$d/dt\ v_i$	v_i
-1	-2	0	0	0
0	0	1	1	0
1	2	1	1	2
2	4	2	2	4
3	6	2	2	8
4	8	2	2	12
5	10	2	2	16
6	12	2	2	20
7	14	2	etc	etc
8	16	2		

Nota: Se il sistema é di secondo ordine sono necessari due valori per il calcolo

Introduzione

Metodologia Generale

Integrazione Numerica
- Osservazioni
- Esempio
- Metodo di Eulero
- Esempio
- Esercizio
- Soluzione - I
- Soluzione - II
- Soluzione - III(a)
- Soluzione - III(b)

Serie di Taylor

Integrazione di Runge-Kutta

Risposta del Sistema

Esercizio

Usare l'integrazione del primo ordine per risolvere l'equazione differenziale da 0 a 10 secondi con un intervallo di 1 secondo.

$$\dot{x} + 0.1x = 5$$

Soluzione - I

Risoluzione di un'equazione di stato con un programma in linguaggio C

```
double step(double, double, double);

int main(){
        double          h = 0.1,
                        M = 1.0,
                        F;
        FILE    *fp;
        double  v,
                t;
        if( ( fp = fopen("out.txt", "w")) != NULL){
                v = 0.0;
                for( t = 0.0; t < 10.0; t += h ){
                        if((t >= 0.0) && (t < 4.0)) F = 1.0;
                        if(t > 4.0) F = 2.0;
                        v = step(v, h, F/M);
                        fprintf(fp, "%f, %f, %f\n", t, v, F, M);
                }
        }
        fclose(fp);
}

double step(double v, double h, double slope){
        double          v_new;
        v_new = v + h * slope;
        return v_new;
}
```

Soluzione - II

Risoluzione di un'equazione di stato con un programma in linguaggio Java

```
double step(double, double, double);

public class Integrate extends Object
        public void main() {
                double          h = 0.1,
                                M = 1.0,
                                F;
                FileOut fp = new FileOut("out.txt");
                if(fp.writeStatus != fp.IO_EXCEPTION){
                        double v = 0.0;
                        for( double t = 0.0; t < 10.0; t += h ){
                                if((t >= 0.0) && (t < 4.0)) F = 1.0;
                                if(t > 4.0) F = 2.0;
                                v = step(v, h, F/M);
                                fp.printf(fp, "%f, %f, %f\n", t, v, F, M);
                        }
                        fp.close();
                }
                fclose(fp);
        }

        public double step(double v, double h, double slope){
                double          v_new;
                v_new = v + h * slope;
                return v_new;
        }
}
```

Soluzione - III(a)

Risoluzione di un’equazione di stato con un programma in Scilab

```
//
// first_order.sce
//
// A first order integration of an accelerating mass
//
// To run this in Scilab use 'File' then 'Exec'.
//
// by: H. Jack Sept., 16, 2002
//


// System component values
mass = 10;
force = 100;

x0 = 8;        // initial conditions
v0 = 12;
X=[x0, v0];

// define the state matrix function
// the values returned are [x, v]
function foo=f(state,t)
       foo = [ state($, 2), force/mass]; // d/dt x = v, d/dt v = F/M
endfunction


// Set the time length and step size for the integration
steps = 100;
t_start = 1;
t_end = 100;
h = (t_end - t_start) / steps;
```

Soluzione - III(b)

```
//
// Loop for integration
//
for i=1:steps,
        X = [X ; X($,:) + h*f(X, i*h)];
end
printf("The value at the end of first order integration is (x, v) = (%f, %f)\n", ...
        X($,1), ...
        X($,2));



//
// Explicit equation
//
function x=position(x0, v0, a0, t)
        x = (0.5 * a0 * t^2) + (v0 * t) + x0;
endfunction

function v=velocity(v0, a0, t)
        v = (a0 * t) + v0;
endfunction

printf("The value with integration is (x, v) = (%f, %f)\n", ...
                position(x0, v0, force/mass, t_end), ...
                velocity(v0, force/mass, t_end));



//
// The results should be
//      first order integration = (49710, 1002)
//      explicit                = (51208, 1012)
//
// The difference is 1498 for position and 10 for velocity. This is relatively small, but
// shows a clear case of the innacuracy of the numerical solutions.
//
// Note: increasing the number of steps increases the accuracy
//
```

Serie di Taylor

Osservazioni

- L'integrazione numerica funziona bene con funzioni smorzate
- Quando incontriamo funzioni con andamento ondulatorio possiamo usare equazioni di integrazione di piú alto ordine come l'equazione della serie di Taylor
- La prima parte dell'equazione della serie di Taylor é uguale alla equazione di Eulero, ma poi i termini di grado piú elevato aumentano l'accuratezza
- L'equazione alla variabili di stato di un sistema non é adatta allo sviluppo in serie di Taylor propio perché quest'ultima richiede termini di ordine superiore al primo.

$$x(t+h) = x(t) + h\left(\frac{d}{dt}\right)x(t) + \frac{1}{2!}h^2\left(\frac{d}{dt}\right)^2 x(t) + \frac{1}{3!}h^3\left(\frac{d}{dt}\right)^3 x(t) + \frac{1}{4!}h^4\left(\frac{d}{dt}\right)^4 x(t) + \ldots$$

Esempio

Esempio di applicazione della serie di Taylor. Data l'equazione differenziale dobbiamo calcolare le derivate e poi sostituirle nell'equazione di Taylor. Il risultato é usato per calcolarei valori iterativamente

Given $\dot{x} - x = 1 + e^{-20t} + t^3$

We can write,

$$\left(\frac{d}{dt}\right)x = 1 + e^{-20t} + t^3 + x$$

$$\left(\frac{d}{dt}\right)^2 x = -e^{-20t} + 3t^2$$

$$\left(\frac{d}{dt}\right)^3 x = e^{-20t} + 6t$$

In the Taylor series this becomes,

$$x(t+h) = x(t) + h(1 + e^{-20t} + t^3 + x) + \frac{1}{2!}h^2(-e^{-t} + 3t^2) + \frac{1}{3!}h^3(e^{-t} + 6$$

Thus $x_0 = 0$, $h = 0.1$

$t\,(s)$	$x(t)$
0	0
0.1	0
0.2	
0.3	
0.4	
0.5	
0.6	
0.7	
0.8	
0.9	

e.g., for t=0.1

$$x(0+0.1) = 0 + 0.1(2) + \frac{1}{2!}(0.1)^2(-1) + \frac{1}{3!}(0.1)^3(1) =$$

Integrazione di Runge-Kutta

Osservazioni

- L’integrazione del primo ordine dá una ragionevole soluzione all’equazione differenziale
- L’accuratezza puó migliorare usando derivate di ordine piú elevato che compensano per la curvatura della funzione
- La tecnica di Runge-Kutta usa equazioni del primo ordine (e quindi puó venir usata la rappresentazione di stato) per stimare le derivate di ordine superiore, quindi si ottiene accuratezza elevata senza richiedere piú del primo ordine dell’equazione differenziale

Esempio

Esempio di integrazione di Runge-Kutta del quarto ordine. La funzione $f(t)$ é l'equazione di stato. Per ogni istante di tempo i valori da $F1$ a $F4$ sono calcolati in sequenza e poi usati nell'equazione finale per trovare il valore successivo.

$$F_1 = hf(t, x)$$

$$F_2 = hf\left(t + \frac{h}{2}, x + \frac{F_1}{2}\right)$$

$$F_3 = hf\left(t + \frac{h}{2}, x + \frac{F_2}{2}\right)$$

$$F_4 = hf(t + h, x + F_3)$$

$$x(t+h) = x(t) + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$

where,

x = *the state variables*
f = *the differential function or* $(d/dt)\,x$
t = *current point in time*
h = *the time step to the next integration point*

Note: in this case the state equation function f(t, **x**) includes the state variables, **x**, and time, t. However, in simpler systems the state equations may not include time and it could be replaced with f(**x**).

Esempio

La soluzione inizia mettendo l'equazione di stato in forma matriciale e definendo le condizioni iniziali. Poi vengono calcolati i quattro fattori di integrazione, che poi vengono combinati per dare il valore finale dopo un passo.

$$\frac{d}{dt}x = v \qquad y = 2 \text{ (assumed input)}$$

$$\frac{d}{dt}v = 3 + 4v + 5y \qquad v_0 = 1$$

$$x_0 = 3$$

$$\frac{d}{dt}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 5 & 3 \end{bmatrix}\begin{bmatrix} y \\ 1 \end{bmatrix} \qquad h = 0.1$$

For the first time step,

$$F_1 = 0.1\left(\begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 5 & 3 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = 0.1\left(\begin{bmatrix} 1 \\ 4 \end{bmatrix} + \begin{bmatrix} 0 \\ 13 \end{bmatrix}\right) = \begin{bmatrix} 0.1 \\ 1.7 \end{bmatrix}$$

$$F_2 = 0.1\left(\begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} 3 + \frac{0.1}{2} \\ 1 + \frac{1.7}{2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 5 & 3 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0.185 \\ 2.04 \end{bmatrix}$$

$$F_3 = 0.1\left(\begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} 3 + \frac{0.185}{2} \\ 1 + \frac{2.04}{2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 5 & 3 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0.202 \\ 2.108 \end{bmatrix}$$

$$F_4 = 0.1\left(\begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} 3 + 0.202 \\ 1 + 2.108 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 5 & 3 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0.3108 \\ 2.5432 \end{bmatrix}$$

$$\begin{bmatrix} x_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ v_i \end{bmatrix} + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$

$$\begin{bmatrix} x_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \frac{1}{6}\left(\begin{bmatrix} 0.1 \\ 1.7 \end{bmatrix} + 2\begin{bmatrix} 0.185 \\ 2.04 \end{bmatrix} + 2\begin{bmatrix} 0.202 \\ 2.108 \end{bmatrix} + \begin{bmatrix} 0.3108 \\ 2.5432 \end{bmatrix}\right) = \begin{bmatrix} 3.1974667 \\ 3.0898667 \end{bmatrix}$$

Esercizio

$$F = M(\frac{d}{dt})_2 x$$

usando:

$$\begin{aligned} x(0) &= 1 \\ \dot{x}(0) &= 2 \\ h &= 0.5s \\ F &= 10 \\ M &= 1 \end{aligned}$$

Soluzione I - parte I

Metodo di integrazione di Runge-Kutta per un sistema massa-moll-smorzatore con un programma in linguaggio C

```c
#include <stdio.h>

void    multiply(double, double[], double[]);
void    add(double[], double[], double[]);
void    step(double, double, double[]);
void    derivative(double, double[], double[]);

#define SIZE           2 /* the length of the state vector */
#define Ks             1000 /* the spring coefficient */
#define Kd             10000 /* the damping coefficient */
#define Mass           10 /* the mass coefficient */
#define Force          100 /* the applied force */

int main(){
        FILE    *fp;

        double  h = 0.001;
        double  t;
        int             j = 0;

        double  X[SIZE];// create state variable list
        X[0] = 0;// set initial condition for x
        X[1] = 0;// set initial condition for v

        if( ( fp = fopen("out.txt", "w")) != NULL){
                fprintf(fp, "   t(s)       x          v \n\n");
                for( t = 0.0; t < 50.0; t += h ){
                        step(t, h, X);
                        if(j == 0) fprintf(fp, "%9.5f %9.5f %9.5f\n", t, X[0], X[1]);
                        j++; if(j >= 10) j = 0;
                }
        }
        fclose(fp);
}
```

Soluzione I - parte II

```
void step(double t, double h, double X[]){
        double  tmp[SIZE],
                        dX[SIZE],
                        F1[SIZE],
                        F2[SIZE],
                        F3[SIZE],
                        F4[SIZE];
        /* Calculate F1 */
        derivative(t, X, dX);
        multiply(h, dX, F1);

        /* Calculate F2 */
        multiply(0.5, F1, tmp);
        add(X, tmp, tmp);
        derivative(t+h/2.0, tmp, dX);
        multiply(h, dX, F2);

        /* Calculate F3 */
        multiply(0.5, F2, tmp);
        add(X, tmp, tmp);
        derivative(t+h/2.0, tmp, dX);
        multiply(h, dX, F3);

        /* Calculate F4 */
        add(X, F3, tmp);
        derivative(t+h, tmp, dX);
        multiply(h, dX, F4);

        /* Calculate the weighted sum */
        add(F2, F3, tmp);
        multiply(2.0, tmp, tmp);
        add(F1, tmp, tmp);
        add(F4, tmp, tmp);
        multiply(1.0/6.0, tmp, tmp);
        add(tmp, X, X);
}
```

```
/* State Equations Calculated Here */
void derivative(double t, double X[], double dX[]){
        dX[0] = X[1];
        dX[1] = (-Ks/Mass)*X[0] + (-Kd/Mass)*X[1] + (Force/Mass);
}

/* A subroutine to add vectors to simplify other equations */
void add(double X1[], double X2[], double R[]){
        for(int i = 0; i < SIZE; i++) R[i] = X1[i] + X2[i];
}

/* A subroutine to multiply a vector by a scalar to simplify other equations*/
void multiply(double X, double V[], double R[]){
        for(int i = 0; i < SIZE; i++) R[i] = X*V[i];
}
```

Soluzione II

Metodo di integrazione di Runge-Kutta per un sistema massa-moll-smorzatore con un programma in Scilab

```
// runge_kutta.sce
// A first order integration of an accelerating mass
// To run this in Scilab use 'File' then 'Exec'.
// by: H. Jack Sept., 15, 2003

// System component values
mass = 10;
force = 100;

x0 = 8;         // initial conditions
v0 = 12;
X=[x0, v0];

// define the state matrix function
// the values returned are [x, v]
function foo=f(state,t)
        foo = [ state($, 2), force/mass]; // d/dt x = v, d/dt v = F/M
endfunction

// Set the time length and step size for the integration
steps = 1000;
t_start = 0;
t_end = 100;
h = (t_end - t_start) / steps;
t = [t_start];

// Loop for integration
for i=1:steps,
        t = [t ; t($,:) + h];
        F1 = h * f(X($,:), t($,:));
        F2 = h * f(X($,:) + F1/2.0, t($,:) + h/2.0);
        F3 = h * f(X($,:) + F2/2.0, t($,:) + h/2.0);
        F4 = h * f(X($,:) + F3, t($,:) + h);
        X = [X ; X($,:) + (F1 + 2.0*F2 + 2.0*F3 + F4)/6.0];
end

// print some results to compare
printf("The value at the end of first order integration is (x, v) = (%f, %f)\n", ...
        X($,1), ...
        X($,2));
printf("The position (using an equation) should be %f\n", 0.5*force/mass*(t_end-
t_start)^2.0 + v0*(t_end - t_start) + x0);

// Graph the values
plot2d(t, X, [-2, -5], leg="position@velocity");
                // leg - the legend titles
                // style - draw lines with marks
                // nax - grid lines for the graph
xtitle('Time (s)');
```

Risposta del Sistema

Osservazioni

- In molti casi il risultato di un'analisi numerica é un grafico od una tabella
- Dettagli come costanti di tempo e frequenze di smorzamento possono essere ottenute con metodi di analisi sperimentale
- Per determinare la risposta a regime del sistema si usa l'equazione di stato in cui le derivate vengono settate al valore di zero e poi viene risolta l'equazione (vedi esempio)

Esempio

Given the state variable form:

$$\frac{d}{dt}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K_s}{M} & -\frac{K_d}{M} \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{F}{M} \end{bmatrix}$$

Set the derivatives to zero

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K_s}{M} & -\frac{K_d}{M} \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{F}{M} \end{bmatrix}$$

Solve for x and v

$$\begin{bmatrix} 0 & 1 \\ -\frac{K_s}{M} & -\frac{K_d}{M} \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{F}{M} \end{bmatrix}$$

$$x = \frac{\left|\begin{bmatrix} 0 & 1 \\ -\frac{F}{M} & -\frac{K_d}{M} \end{bmatrix}\right|}{\left|\begin{bmatrix} 0 & 1 \\ -\frac{K_s}{M} & -\frac{K_d}{M} \end{bmatrix}\right|} = \frac{\left(\frac{F}{M}\right)}{\left(\frac{K_s}{M}\right)} = \frac{F}{K_S} \qquad v = \frac{\left|\begin{bmatrix} 0 & 0 \\ -\frac{K_s}{M} & -\frac{F}{M} \end{bmatrix}\right|}{\left|\begin{bmatrix} 0 & 1 \\ -\frac{K_s}{M} & -\frac{K_d}{M} \end{bmatrix}\right|} = \frac{0}{\left(\frac{K_s}{M}\right)} = 0$$