Numeryczne rozwiązywanie równań różniczkowych zwyczajnych o stałych współczynnikach przy użyciu MATLAB-a

Ireneusz Czajka

Kraków listopad 2000

1 Wstęp

Równania różniczkowe zwyczajne o stałych współczynnikach znajdują zastosowanie do opisu układów dynamicznych. Aby móc zastosować taką formę opisu matematycznego należy:

1. dokonać dyskretyzacji układu rzeczywistego,
2. pominąć mało istotne wpływy,
3. założyć stałość parametrów układu.

Model fizyczny otrzymany w wyniku zastosowania powyższych uproszczeń możemy opisać już równaniem różniczkowym zwyczajnym o stałych współczynnikach.

Z kolei matematyczny opis modelu fizycznego nazywamy **modelem matematycznym**. Najczęściej ma on postać układu równań różniczkowych drugiego rzędu (każda masa bezwładna generuje jedno równanie) z wzajemnymi sprzężeniami.

2 Układy równań różniczkowych

Równanie różniczkowe zwyczajne o stałych współczynnikach rzędu n da się przedstawić jako układ n równań różniczkowych rzędu 1.

Załóżmy, że mamy równanie różniczkowe:

$$\frac{d^n x}{dt^n} = f(t, x, \dot{x}, \ddot{x}, \ldots, \frac{d^{n-1}x}{dt^{n-1}}) \tag{1}$$

Wykonamy podstawienie:

$$\begin{aligned} z_1 &= x, \\ z_2 &= \dot{z_1}, \\ &\vdots \\ z_n &= \dot{z_{n-1}} = \frac{d^{n-1}x}{dt^{n-1}} \end{aligned} \tag{2}$$

Dzięki temu podstawieniu otrzymamy układ równań różniczkowych pierwszego rzędu:

$$\begin{aligned} \dot{z_1} &= z_2, \\ \dot{z_2} &= z_3, \\ &\vdots \\ \dot{z_n} &= f(t, x, \dot{x}, \ddot{x}, \ldots, \frac{d^{n-1}x}{dt^{n-1}}) \end{aligned} \tag{3}$$

2.1 Przykład

Równanie opisujące ruch masy m zawieszonej sprężyście na sprężystości k, z tłumieniem b opisuje równanie:

$$m\ddot{x} + b\dot{x} + kx = f(t,x) \tag{4}$$

Podstawienia wyglądają następująco:

$$\begin{aligned} z_1 &= x \\ z_2 &= \dot{x} \end{aligned}$$

układ równań, który otrzymujemy wygląda tak:

$$\begin{aligned} \dot{z_1} &= z_2 \\ \dot{z_2} &= -\frac{b}{m}z_2 - \frac{k}{m}z_1 + \frac{f(t,z_1)}{m} \end{aligned}$$

Możemy zapisać go w postaci macierzowej jako:

$$\begin{bmatrix} \dot{z_1} \\ \dot{z_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{f(t,z_1)}{m} \end{bmatrix}$$

3 Metody rozwiązywania układów równań różniczkowych o stałych współczynnikach pierwszego rzędu

Najbardziej prymitywną metodą jest metoda Eulera. Polega ona na przybliżeniu szukanego rozwiązania odcinkami prostych stycznych do rozwiązania w dyskretnych punktach oddalonych od siebie o długość kroku Δx

Udoskonaleniem tej metody zajmowało się wielu badaczy. Obecnie stosuje się rozwinięcie w postaci metody Rungego–Kutty.

Równanie w postaci:

$$\frac{dy}{dx} = f(x,y)$$

rozwiązuje się korzystając z następujących wzorów:

$$\begin{aligned} x_{i+1} &= x_i + \Delta x \\ y_{i+1} &= y_i + \Delta y \end{aligned}$$

Gdzie: Δx jest znane (czasami stałe), zaś Δy obliczamy w oparciu o wzory:

$$\begin{aligned} k_1 &= \Delta x \cdot f(x_i, y_i) \\ k_2 &= \Delta x \cdot f(x_i + \frac{\Delta x}{2}, y_i + \frac{k_1}{2}) \\ k_3 &= \Delta x \cdot f(x_i + \frac{\Delta x}{2}, y_i + \frac{k_2}{2}) \\ k_4 &= \Delta x \cdot f(x_i + \Delta x, y_i + k_3) \\ \Delta y &= \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4] \end{aligned}$$

W zaprezentowanym przypadku mamy do czynienia z metodą Rungego–Kutty rzędu czwartego.

W MATLAB-ie metoda ta jest zaimplementowana w postaci funkcji `ode23` i `ode45` obliczających rozwiązanie równania odpowiednio metodą R-K rzędu 2-3 i 4-5. Poniżej zamieszczono opis tych komend zaczerpnięty z MATLAB-a.

```
help ode23


 ODE23 Solve differential equations, low order method.
  ODE23 integrates a system of ordinary differential equations using
  2nd and 3rd order Runge-Kutta formulas.
  [T,Y] = ODE23('yprime', T0, Tfinal, Y0) integrates the system of
  ordinary differential equations described by the M-file YPRIME.M,
  over the interval T0 to Tfinal, with initial conditions Y0.
  [T, Y] = ODE23(F, T0, Tfinal, Y0, TOL, 1) uses tolerance TOL
  and displays status while the integration proceeds.

  INPUT:
  F     - String containing name of user-supplied problem description.
          Call: yprime = fun(t,y) where F = 'fun'.
          t      - Time (scalar).
          y      - Solution column-vector.
          yprime - Returned derivative column-vector; yprime(i) = dy(i)/dt.
  t0    - Initial value of t.
  tfinal- Final value of t.
  y0    - Initial value column-vector.
  tol   - The desired accuracy. (Default: tol = 1.e-3).
  trace - If nonzero, each step is printed. (Default: trace = 0).

  OUTPUT:
  T  - Returned integration time points (column-vector).
  Y  - Returned solution, one solution column-vector per tout-value.

  The result can be displayed by: plot(tout, yout).

  See also ODE45, ODEDEMO.

help ode45


 ODE45 Solve differential equations, higher order method.
  ODE45 integrates a system of ordinary differential equations using
  4th and 5th order Runge-Kutta formulas.
  [T,Y] = ODE45('yprime', T0, Tfinal, Y0) integrates the system of
  ordinary differential equations described by the M-file YPRIME.M,
  over the interval T0 to Tfinal, with initial conditions Y0.
  [T, Y] = ODE45(F, T0, Tfinal, Y0, TOL, 1) uses tolerance TOL
  and displays status while the integration proceeds.

  INPUT:
  F     - String containing name of user-supplied problem description.
          Call: yprime = fun(t,y) where F = 'fun'.
          t      - Time (scalar).
          y      - Solution column-vector.
          yprime - Returned derivative column-vector; yprime(i) = dy(i)/dt.
  t0    - Initial value of t.
  tfinal- Final value of t.
  y0    - Initial value column-vector.
  tol   - The desired accuracy. (Default: tol = 1.e-6).
  trace - If nonzero, each step is printed. (Default: trace = 0).
```

```
OUTPUT:
T  - Returned integration time points (column-vector).
Y  - Returned solution, one solution column-vector per tout-value.

The result can be displayed by: plot(tout, yout).

See also ODE23, ODEDEMO.
```

4 Wykorzystanie

Aby móc rozwiązać układ równań różniczkowych rzędu metodami `ode23` lub `ode45` należy:

1. zapisać układ równań w postaci układu równań pierwszego rzędu,
2. zapisać układ równań w postaci m-pliku funkcyjnego,
3. wywołać funkcję `ode23` lub `ode45`

M-plik funkcyjny opisujący układ r. różniczkowych musi przyjmować dwa argumenty: (t,x), t – czas, zmienna niezależna (x), zaś x – zmienna zależna (y). Zwracać zaś kolumnowy wektor pochodnych $\begin{bmatrix} \dot{z_1} \\ \dot{z_2} \end{bmatrix}$

4.1 Przykład m-pliku funkcyjnego

M-plik funkcyjny o nazwie `uklad1.m` może mieć następującą postać:

```
function dy=uklad1(t,y)
global m,k,b
dy=[0 1;-k/m -b/m]*y+f(t,y)/m
```

gdzie $f(t,y)$ oznacza funkcję wymuszającą i może to być np: $f(t,y) = 100\sin(314 * t)$

Interpretacja wyników zależy od przyjętych podstawień. W naszym przykładzie z_1 oznacza przemieszczenie masy, zaś z_2 oznacza jej prędkość.