

Wykład X

Rozwiązywanie zagadnień początkowych dla równań różniczkowych zwyczajnych

- Sformułowanie zadania i podstawowe idee jego rozwiązania
- Metody samostartujące (Eulera, Rungego-Kutty)
- Metody niesamostartujące (Adamsa-Bashforta-Moultona, Geara)
- Predykcja-korekcja, kontrola dokładności, adaptacja
- Symulacja systemów dynamicznych w Matlabie i Simulinku

Równania różniczkowe zwyczajne w technice

Równania różniczkowe zwyczajne są jednym ze sposobów opisu zmienności systemów. W naturalny sposób powstają w wyniku zastosowania praw fizyki do modelowania zjawisk dynamicznych zachodzących w analizowanych systemach. W takim ujęciu rozwiązanie polega na symulacji (obliczeniu) odpowiedzi czasowej systemu dynamicznego.

Przykłady:

- prawa Kirchoffa dla układu zmiennoprądowego RLC,
- opis dynamiki termicznej czujnika temperatury z obudową,
- opis elektromechaniczny silnika prądu stałego,
- dynamika systemów nietechnicznych – modele zachowań drapieżcy-ofiary, reakcja psychologiczna na pobudzenie, ruch fali samochodów w gęstym ruchu ulicznym – równanie różniczkowe z nieliniowościami
- Obwód z elementami nagrzewającymi się – równ. różn. z parametrami zmiennymi z czasem.

Równania różniczkowe wyższego stopnia możemy sprowadzić do układu równań pierwszego stopnia metodą podstawienia zmiennych uzyskując układ (stan $\mathbf{x}(t)$ jest wektorem !):

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), u(t)) \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad y(t) = g(\mathbf{x}(t), u(t)) \quad \begin{array}{l} u(t) = we \\ y(t) = wy \end{array}$$

Przypadek liniowy wielu wejść i wielu wyjść (MIMO) prowadzi do powszechnie używanego opisu w postaci macierzowych równań stanu.

Opis systemów/obiektów liniowych równaniami stanu

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

$\mathbf{u}(t)$ - wektor wejść

$\mathbf{x}(t)$ - wektor stanu wewn.

$\mathbf{y}(t)$ - wektor wyjść

\mathbf{A} – macierz kwadratowa $n \times n$,

\mathbf{B} – macierz $n \times m$,

\mathbf{C} – macierz $p \times n$,

\mathbf{D} – macierz $p \times m$,

n – ilość stanów, m – ilość wejść, p – ilość wyjść.

Macierzowe równanie stanu opisuje układ MIMO (wiele wejść, wiele wyjść).

Ogólne rozwiązanie równań stanu ma postać splotową:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$$

jednak w praktyce obliczeniowej wykorzystuje się do wyznaczania trajektorii stanu efektywniejsze czasowo algorytmy numerycznego całkowania równań stanu.

Opis obiektu pierwszego rzędu równaniami stanu

Równanie różniczkowe:

$$T \frac{dy(t)}{dt} + y(t) = Ku(t)$$

Równanie stanu:

$$\begin{aligned} x &= y, \\ A &= -\frac{1}{T}, & B &= \frac{K}{T}, \\ C &= 1, & D &= 0 \end{aligned}$$

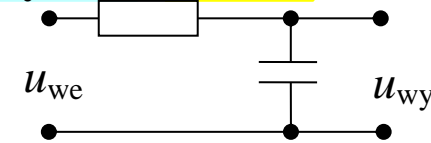
Odpowiedź impulsowa:

$$h(t) = \frac{K}{T} e^{-\frac{t}{T}} = y(t)$$

Odpowiedź skokowa:

$$k(t) = K \left(1 - e^{-\frac{t}{T}}\right) = y(t)$$

Przykład: filtr RC



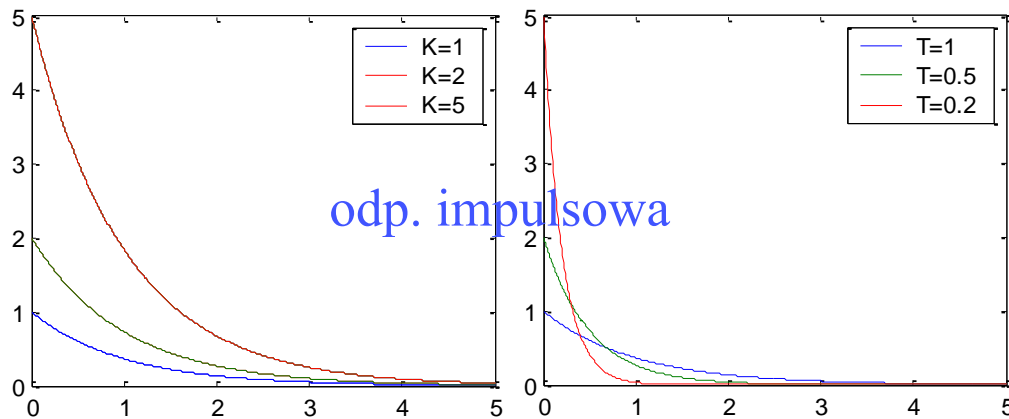
Założenia: brak wzajemnego obciążenia nast./poprz. elementu

$$R \left[\Omega = \frac{\text{kgm}^2}{\text{s}(\text{As})^2} \right], C \left[F = \frac{\text{s}^2}{\text{kgm}^2} (\text{As})^2 \right]$$

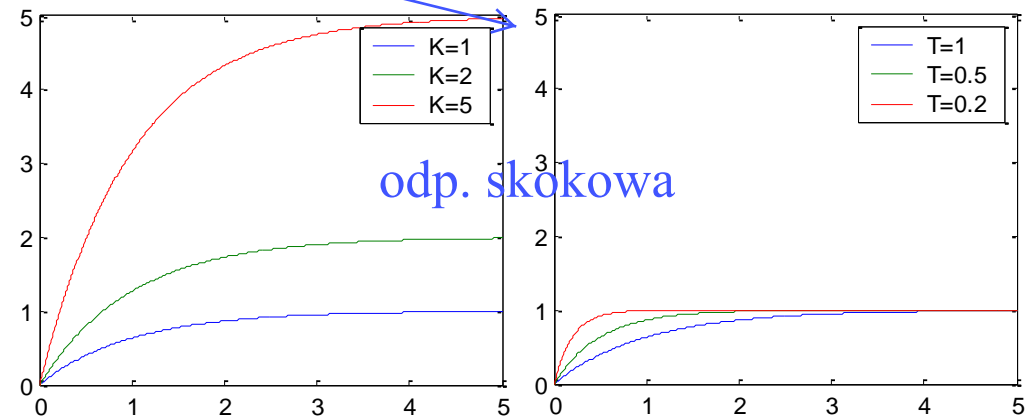
$$RC \frac{du_{wy}}{dt} + u_{wy} = u_{we}$$

$$T \frac{dy}{dt} + y = u$$

$$T = RC[\text{s}], \quad y = u_{wy}, \quad u = u_{we}$$



odp. impulsowa



odp. skokowa

Opis obiektu drugiego rzędu równaniami stanu

także układ RLC

Równanie różniczkowe:

$$\frac{d^2 y(t)}{dt^2} + 2\xi\omega_0 \frac{dy(t)}{dt} + \omega_0^2 y(t) = K\omega_0^2 u(t)$$

Równanie stanu:

$$\begin{aligned} x_1 &= y, \quad x_2 = \frac{dy}{dt}, \\ \mathbf{A} &= \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\xi\omega_0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ K\omega_0^2 \end{bmatrix}, \\ \mathbf{C} &= [1 \quad 0], \quad \mathbf{D} = 0 \end{aligned}$$

Odpowiedź impulsowa: ($\eta = \sqrt{1-\xi^2}$)

$$\xi < 1: h(t) = K \frac{\omega_0}{\eta} e^{-\xi\omega_0 t} \sin(\omega_0 \eta t)$$

$$\xi = 1: h(t) = K\omega_0^2 t e^{-\omega_0 t}$$

Odpowiedź skokowa: ($\eta = \sqrt{1-\xi^2}$)

$$\xi < 1: k(t) = K \left[1 - \frac{1}{\eta} e^{-\xi\omega_0 t} \sin(\omega_0 \eta t + \arcsin \eta) \right]$$

$$\xi = 1: k(t) = K \left[1 - (1 + \omega_0 t) e^{-\omega_0 t} \right]$$

arcsin()

Przykład: czujnik drgań sejsmicznych

$u(t) = F$ - siła wymuszająca

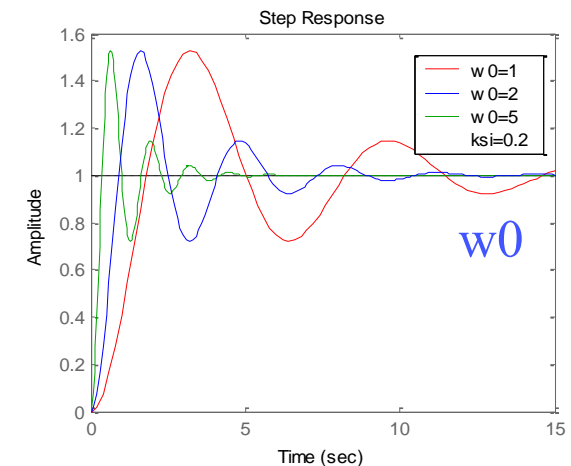
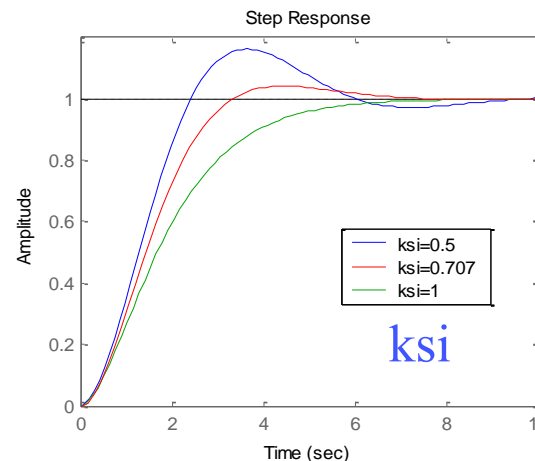
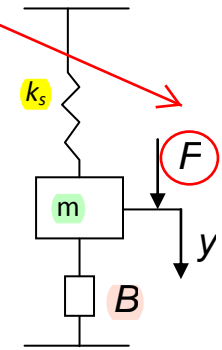
$F_b = m \frac{d^2 y}{dt^2}$ - siła bezwładności

$F_s = k_s y$ - siła reakcji sprężyny

$F_t = B \frac{dy}{dt}$ - siła tłumiąca

$$m \frac{d^2 y(t)}{dt^2} + B \frac{dy(t)}{dt} + k_s y(t) = u(t)$$

$$K = \frac{1}{k_s}, \quad \omega_0 = \sqrt{k_s/m}, \quad \xi = \frac{B}{2\sqrt{k_s/m}}$$



Podstawowe idee rozwiązań problemu

Informacja dostępna w każdym kroku:

- bieżąca wartość stanu (bieżący punkt),
- wartość pochodnej stanu w bieżącym punkcie (zależna od czasu, wejścia, stanu).

Ewentualnie do wykorzystania w metodach wielokrokowych:

- poprzednie wartości stanu,
- poprzednie wartości pochodnej stanu.

Na tej podstawie należy wyznaczyć wartość stanu w następnym punkcie.

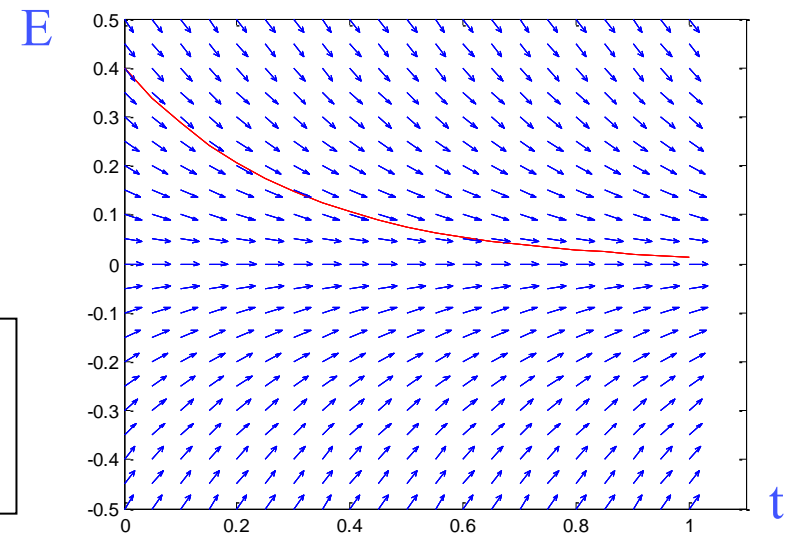
Przykład Inercyjne wytracanie energii początkowej (elektrycznej, cieplnej, mechanicznej)

Równanie różniczkowe jednorodne i warunek początkowy, opisujące taki proces, mają postać:

$$\frac{dE}{dt} = -\frac{1}{T} E \quad E(t_0) = E_0$$

Tempo utraty energii jest więc proporcjonalne do poziomu tej energii, co determinuje trajektorie energii z dowolnego warunku początkowego. Na rysunku przykład pola czas-energia z tendencją zmian (strzałki) i trajektorią dla $E_0=0.4$.

```
T=0.3; t=0:0.05:1; E=-0.5:0.05:.5;
plot(t, 0.4*exp(-t/T), 'r'), hold on;
[tt, EE]=meshgrid(t, E); sc=30*sqrt(1+1/T^2*EE.^2);
quiver(tt, EE, ones(size(tt))./sc, -1/T*EE./sc, 0);
```



Najprostszy **algorytm Eulera** (o znaczeniu bardziej dydaktycznym niż praktycznym) wykorzystuje tylko informację dostępną w bieżącym punkcie (np. w punkcie startowym danym przez warunek początkowy).

Wynika stąd algorytm Eulera (**ekstrapolacyjny, predykcyjny**):

$$x_{k+1} = x_k + \Delta t \cdot f(x_k, t_k)$$



zmiana proporc. do pochodnej zmiennej stanu

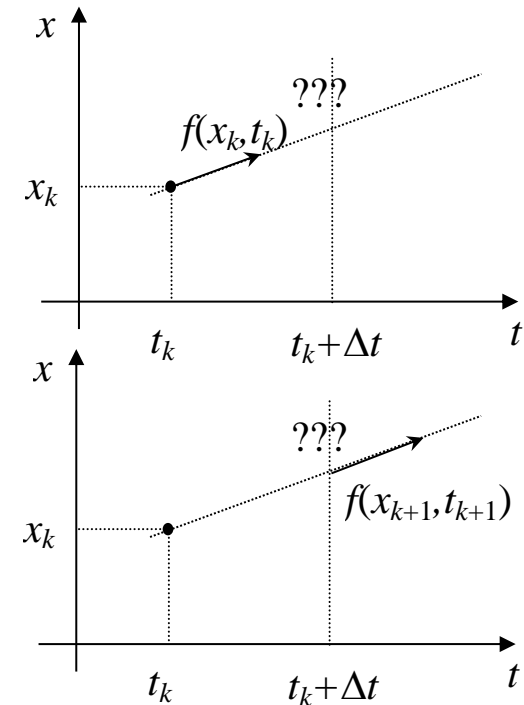
Jego odmianą jest algorytm Eulera w postaci uwikłanej (**interpolacyjny, korekcyjny**):

$$x_{k+1} = x_k + \Delta t \cdot f(x_{k+1}, t_{k+1})$$



zmiana

Wzory te można też wyprowadzić z rozwinięcia $x(t)$ wokół punktu $t = t_k$ w szereg Taylora (co można dalej uogólnić dla wyższego rzędu).



Wadą algorytmu Eulera jest kumulacja błędów w kolejnych krokach i konieczność stosowania małych kroków symulacji Δt z powodu zgrubnego oszacowania kierunku z pierwszej pochodnej.

Przykład: Dla problemu inercyjnej utraty energii możemy zapisać wzór iteracyjny:

ekstrapolacja: $E_{k+1} = E_k \frac{T - \Delta t}{T}$

interpolacja: $E_{k+1} = E_k \frac{T}{T + \Delta t}$

Grupa algorytmów **Rungego-Kutty** wykorzystuje dodatkową informację w pośrednich punktach próbnych. Przybliżenie rozwiązania jest uśrednieniem wartości wynikających ze wszystkich punktów próbnych.

Zależności zostały opracowane na zasadzie najlepszego przybliżenia schematu wynikającego z rozwinięcia Taylora odpowiedniego rzędu (ale bez użycia pochodnych) – patrz np. Mathews „Numerical Methods ...”, Buchanan “Numerical Methods and Analysis”.

Liczba punktów próbnych decyduje o rzędzie metody. Przykładowo metoda RK rzędu czwartego

ma postać:
$$x_{k+1} = x_k + \frac{1}{6}(d_1 + 2d_2 + 2d_3 + d_4)$$

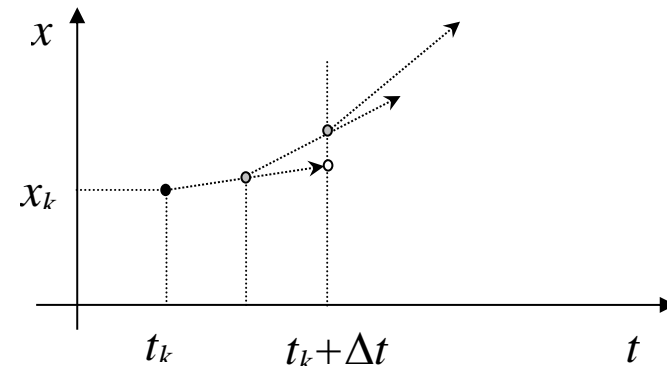
gdzie:

$$d_1 = \Delta t \cdot f(x_k, t_k)$$

$$d_2 = \Delta t \cdot f(x_k + d_1/2, t_k + \Delta t/2)$$

$$d_3 = \Delta t \cdot f(x_k + d_2/2, t_k + \Delta t/2)$$

$$d_4 = \Delta t \cdot f(x_k + d_3, t_k + \Delta t)$$



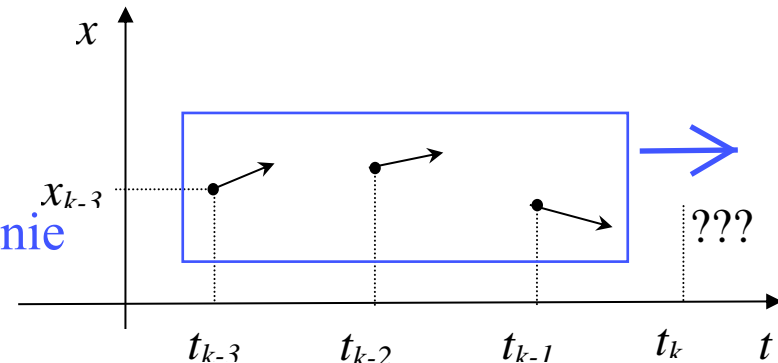
Algorytmy z tej grupy są najczęściej wykorzystywane w symulacjach. Tak jak metoda Eulera są to metody samostartujące ponieważ nie wykorzystują informacji z historii trajektorii stanu, w przeciwieństwie do dalej opisanych.

Dopuszczając wykorzystanie informacji dostępnej w poprzednio wyznaczonych punktach trajektorii, tzn. informacji o stanie i informacji o pochodnej, możemy zdefiniować całą rodzinę algorytmów o ogólnej postaci:

$$x_{k+1} = \sum_{i=1}^I a_i x_{k+1-i} + \Delta t \sum_{j=0}^J b_j f(x_{k+1-j}, t_{k+1-j})$$

poprzednie
wart. zmiennej
stanu

obecne/poprzednie
wart.
pochodnej



Dobór współczynników wynika z wielomianowej aproksymacji trajektorii stanu. Tak więc metoda będzie dawała dokładne wyniki, gdy trajektorię dokładnie opisuje wielomian zmiennej czasowej.

Będą to algorytmy niesamostartujące, ponieważ potrzebują do działania historii trajektorii dłuższej niż ostatnio wyznaczony punkt.

Jak zobaczymy dalej, wybierając rodzaj wykorzystywanej informacji (stan czy pochodne) i używane punkty (z bieżącym lub bez) uzyskujemy algorytmy predykcyjne (ekstrapolacyjne) Adamsa-Bashfortha, korekcyjne (interpolacyjne) Adamsa-Moultona, predykcyjny Geara.

Pierwsza grupa, tj. metody **Adamsa-Bashfortha** mają charakter **ekstrapolacyjny**, tzn. współczynnik $b_0=0$ (nie dają równania uwikłanego).

Przykładowo, metoda Adamsa-Bashfortha rzędu trzeciego ma wzór:

$$x_{k+1} = x_k + \frac{\Delta t}{12} (23f_k - 16f_{k-1} + 5f_{k-2})$$

Metody **Adamsa-Moultona** to metody **interpolacyjne**, wykorzystujące informację o pochodnej w szacowanym punkcie stanu, wymagają więc **rozwiązania równania uwikłanego**.

Przykładowo, metoda Adamsa-Moultona rzędu trzeciego ma wzór:

$$x_{k+1} = x_k + \frac{\Delta t}{12} (5f_{k+1} + 8f_k - f_{k-1})$$

Do wystartowania tych metod kilka pierwszych wartości stanu wyznacza się metodą **samostartującą** (np. Euler albo RK).

Szczególnie **efektywna i stabilna** dla rozwiązywania układów sztywnych (źle uwarunkowane równania stanu) okazała się metoda **Geara** (rodzina algorytmów nazywana również **Backward Differentiation Formula BDF**). Jest to algorytm uwikłany, który **korzysta tylko z poprzednich wartości stanu i z wartości pochodnej stanu w szacowanym punkcie**.

Przykładowo metoda Geara rzędu czwartego ma wzór:

$$x_{k+1} = \frac{1}{25} (48x_k - 36x_{k-1} + 16x_{k-2} - 3x_{k-3} + 12\Delta t \cdot f_{k+1})$$

Oczywiście jest to również metoda **niesamostartująca**.

wart. pochodnej

Przykład Obliczenia odpowiedzi skokowej dla obiektu pierwszego i drugiego rzędu

Porównajmy obliczenia dla metod Rungego-Kutty i Adamsa-Bashfortha stosując zapis równań stanu dla obiektu pierwszego i drugiego rzędu.

```
% Adams-Bashforth for first order dynamics
ref=inline('K*(1-exp(-t/T))','t','K','T');
K=1.5;
T=0.2;
dt=0.001;
tv=0:dt:1;
A=-1/T;
B=K/T;
x=0;
u=1;
xv=ref(tv(1:3),K,T); historia, 3 punkty
for k=3:length(tv)
    x=xv(k);
    F0=A*xv(k)+B*u;
    F1=A*xv(k-1)+B*u;
    F2=A*xv(k-2)+B*u;
    x=x+dt*(23*F0-16*F1+5*F2)/12;
    xv=[xv, x];
end
plot(tv,xv(1:end-1),'r',tv,ref(tv,K,T),'g--');
```

```
% Runge-Kutta for second order dynamics
ref=inline('K*(1-1/sqrt(1-ksi^2))*exp(-ksi*w0*t).*sin(w0*sqrt(1-ksi^2)*t+asin(sqrt(1-ksi^2)))','t','K','ksi','w0');
K=1.2;
ksi=0.3;
w0=10;
dt=0.001;
tv=0:dt:1;
A=[0, 1; -w0^2, -2*ksi*w0];
B=[0; K*w0^2];
x=[0; 0];
u=1;
xv=[];
for t=tv
    xv=[xv, x];
    D1=dt*(A*x+B*u);
    D2=dt*(A*(x+D1/2)+B*u);
    D3=dt*(A*(x+D2/2)+B*u);
    D4=dt*(A*(x+D3)+B*u);
    x=x+(D1+2*D2+2*D3+D4)/6;
end
plot(tv,xv(1,:), 'r',tv,ref(tv,K,ksi,w0),'g--');
```

Algorytmy predykcyjno-korekcyjne

Algorytmy korekcyjne (interpolacyjne, z zależnością uwikłaną) dają się czasami, dla konkretnych symulowanych systemów, przekształcić do postaci jawnej, jak w przykładzie z rozpraszaniem energii. Nie jest to jednak metoda ogólna. Numeryczne rozwiązanie problemu uwikłanego można uzyskać metodami poszukiwania zera funkcji nieliniowej np. iteracją Newtona. Inne rozwiązanie to iteracja prosta wg zależności $x_{k+1}^{(i+1)} = F(x_{k+1}^{(i)}, t_{k+1})$. Iteracja jednak kosztuje dużo czasu.

Efektywne czasowo jest stosowanie par algorytmów predykcyjno-korekcyjnych. Algorytm predykcyjny oszacowuje następny punkt trajektorii, a algorytm korekcyjny dokonuje poprawki z wykorzystaniem cennej (bo najnowszej) informacji o pochodnej w oszacowanym punkcie. Często stosowana para to Adams-Bashforth + Adams-Moulton czwartego rzędu.

Kontrola dokładności i algorytmy adaptacyjne

Wszystkie wymienione algorytmy mają określone własności redukcji błędów ze zmianą kroku symulacji (co nie było podawane a jest w każdej książce nt. obliczeń numerycznych). Korzystając z tej informacji można poprzez porównanie wyników metody dla kroku bieżącego i dwukrotnie mniejszego oszacować bieżącą dokładność (dokładnie tak samo jak w algorytmach całkowania). Porównanie może równie dobrze dotyczyć wyników metod przy zmienianym rzędzie (jak w algorytmie adaptacyjnym Runge-Kutta-Fehlberg RK45). Celem kontroli dokładności jest adaptacyjny, w zależności od lokalnej dynamiki trajektorii, dobór długości kroku symulacji.

Symulacja systemów dynamicznych w Matlabie i Simulinku

W Matlabie symulacje można prowadzić przy opisie transmitancyjnym lub w postaci liniowych równań stanu funkcjami wyznaczającymi odpowiedź na standardowe wymuszenie skokowe (*step*) i impulsowe (*impulse*). Obydwie opierają swoje działanie na transformacji modelu do postaci dyskretnej i iteracyjne wyznaczanie odpowiedzi układu dyskretnego. Dobór kroku czasowego i zakresu czasu symulacji jest dokonywany przez analizę części rzeczywistych wartości własnych macierzy dynamiki **A**. Podobnie działa funkcja wyznaczania odpowiedzi na zadane wejście *lsim*.

W przypadku opisu systemu nieliniowego można wykorzystać funkcje Matlabu *odeset*, *ode23*, *ode45* z generowaniem wejścia wewnątrz funkcji.

W przypadku układów złożonych, które trudno opisać jedną transmitancją/równaniem stanu trzeba skorzystać ze środowiska dedykowanego – Simulinka.

Simulink oferuje do symulacji metody stało i zmiennokrokowe:

- Euler
- Rungego-Kutty
- Adamsa-Bashforth-Moultona
- NDF (podobna do metody Geara, dla układów sztywnych)

Symulację można uruchamiać w środowisku Simulink lub „offline” z Matlabu z użyciem funkcji *simset* (określenie metody, dokładności, itp.) i funkcji *sim* (symulacja). Modele symulacyjne można budować z modeli bibliotecznych (błoczków) lub programowo w postaci s-funkcji.

Przykład Obliczenia trajektorii balistycznej w Matlabie

Układ równań różniczkowych trajektorii balistycznej ma postać:

$$\begin{cases} m \frac{dv_x}{dt} = -kv_x \\ m \frac{dv_y}{dt} = -kv_y - mg \end{cases}$$

Zadanie: obliczyć trajektorię pocisku (s_x, s_y) w zakresie czasu od 0 do 10 sekund dla danych:

$$m=5[\text{kg}]$$

$$k=1.0[\text{kg/s}],$$

$$v=50[\text{m/s}],$$

$$\alpha=30^\circ$$

Do obliczeń użyj funkcji Matlab `ode45`.

Pytania:

Jak wyznaczyć początkowe v_x, v_y ?

Jak wyliczyć trajektorię skoro w równaniu występuje prędkość?

Jak uwzględnić współczynnik tarcia zależny nieliniowo od czasu, wysokości i prędkości?

Przykład Symulacja obiektu drugiego rzędu w Simulinku

Transmitancja obiektu drugiego rzędu ma postać:

$$G(s) = \frac{K\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0^2}$$

Zadanie: obliczyć odpowiedź skokową obiektu dla:

$K=2.0$,

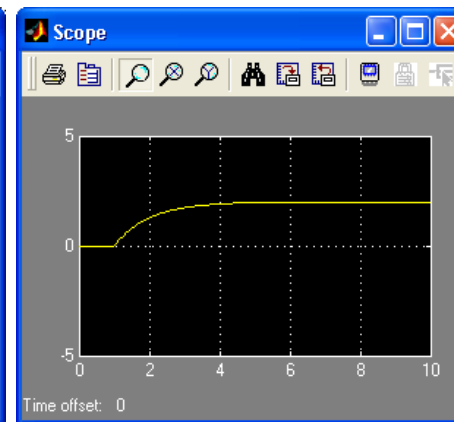
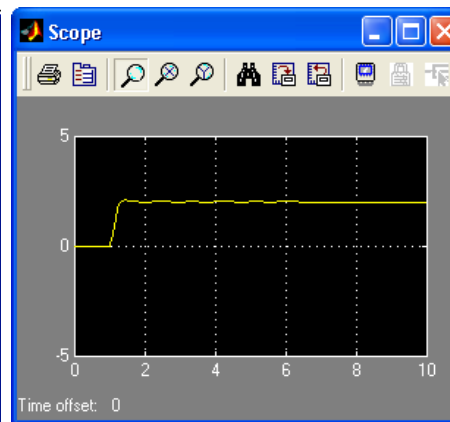
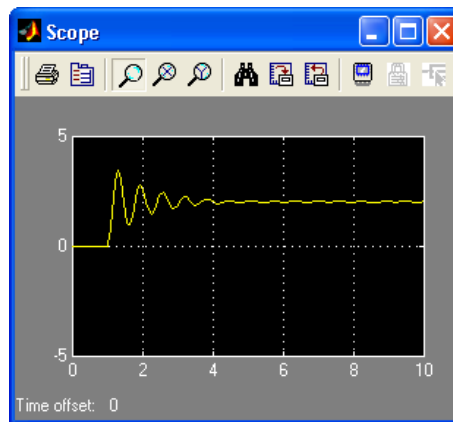
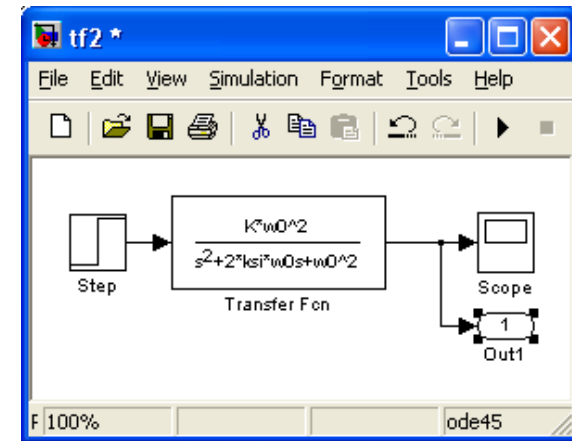
$\xi=0.1, 0.707, 5.0$,

$\omega_0=10.0$

W zakresie czasu od 0 do 10 sekund.

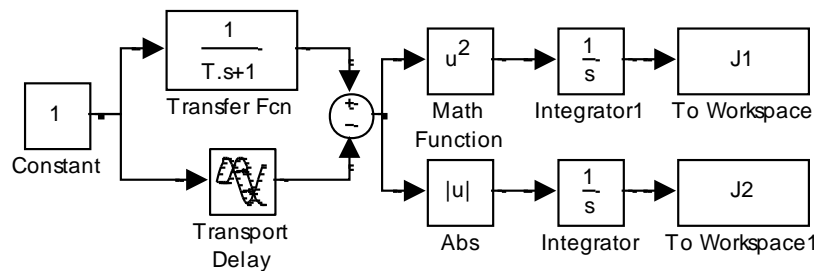
Uruchomienie z linii poleceń:

```
op=simget('tf2'),
op=simset(op,'solver','ode5'),
[t,s,y]=sim('tf2',10,op);
```

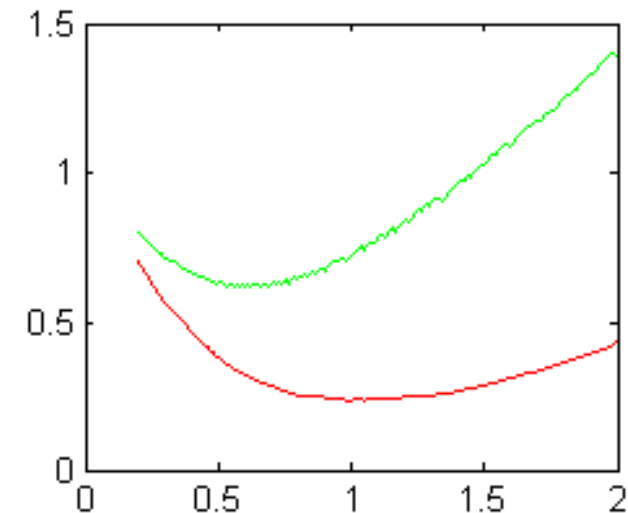


Przykład Symulacja systemu z obiektem pierwszego rzędu

W tym przykładzie symulowany jest wielokrotnie, ze zmianą parametru, system złożony z kilku elementów dynamicznych. Celem jest określenie odległości odpowiedzi inercyjnej od opóźnienia czasowego w funkcji stałej czasowej wg kryterium kwadratowego i modułowego.

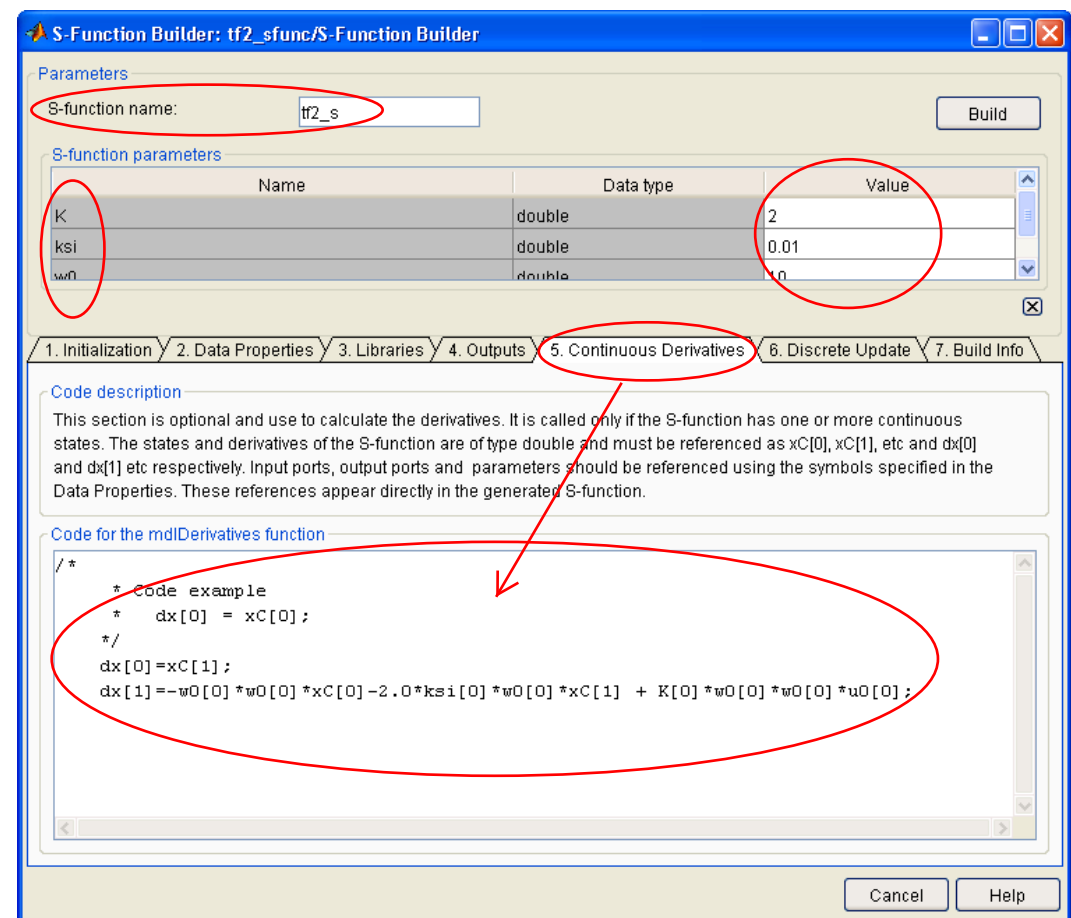
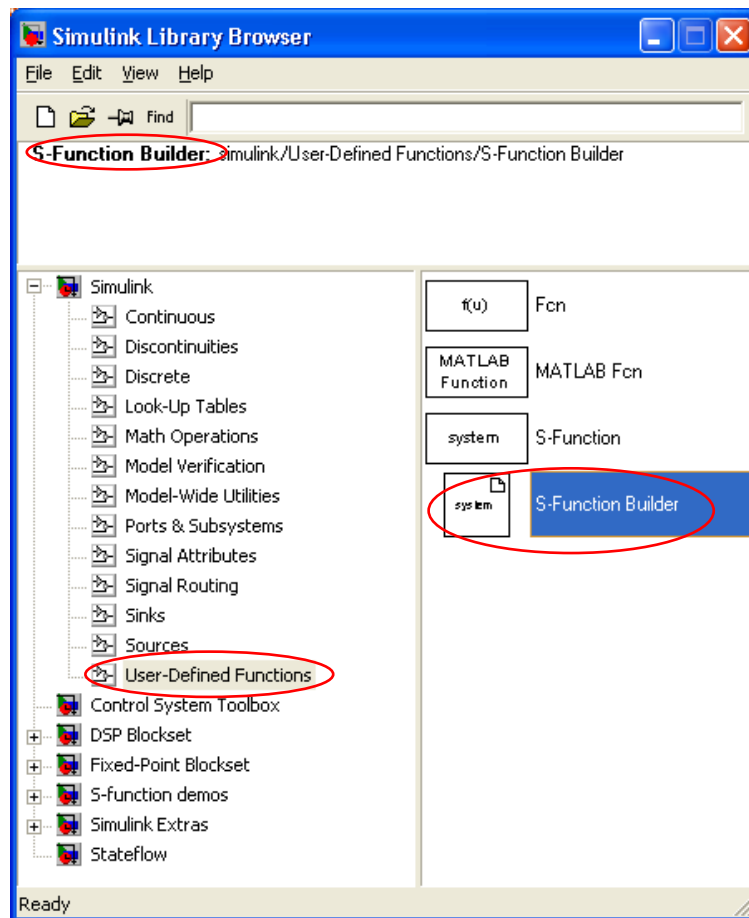


```
% delay set in model is 1
% simulation over 5 time constants
Tv=0.2:0.01:2;
J1v=[], J2v=[];
for T=Tv
    % simulate with model's options
    sim('ex_opt_T',5*T);
    J1v=[J1v, J1], J2v=[J2v, J2];
end
plot(Tv,J1v,'r',Tv,J2v,'g');
```



Przykład: Konstrukcja własnego modelu w postaci S-funkcji (kod w C)

Poprzez zapis modelu (bloczka) w postaci kodu w C możliwe jest **szybkie** zaprogramowanie skomplikowanego modelu, np. z parametrami zmiennymi z czasem i z nieliniowościami.



Podsumowanie, kluczowe elementy tego wykładu:

- Równania różniczkowe zwyczajne (RRZ) opisują dynamikę systemów (zmiennosc w czasie)
- W ogólnej postaci są to nieliniowe równania macierzowe podające zależność pochodnej stanu obiektu po czasie od bieżącego stanu i czasu (a pośrednio od wielkości wejściowych)
- W szczególnym przypadku zależności liniowych równania te są liniowe i noszą nazwę równań stanu opisanych czterema macierzami **A**, **B**, **C**, **D** i stanem początkowym $\mathbf{x}(0)$
- Rozwiązywanie problemów RRZ polega na całkowaniu wyrażenia na pochodną po czasie
- Informacja do wykorzystania to wartość stanu i pochodnej:
 - Pochodna w bieżącym/następnym punkcie – metoda Eulera, Rungego-Kutty
 - Pochodna w poprzednich punktach – metoda Adamsa-Bashfortha, Adamsa-Moultona
 - Stan w poprzednich punktach – metoda Geara
- Jeśli metoda korzysta z informacji w następnym punkcie w sposób uwikłany, to wymaga rozwikłania metodami analitycznymi lub iteracyjnymi
- Narzędziem dedykowanym do symulacji dynamiki systemów jest Simulink