Wykład 9

Numeryczne rozwiązywanie równań różniczkowych

Równania różniczkowe zwyczajne

Zagadnienie początkowe w numerycznym rozwiązywaniu równań różniczkowych zwyczajnych pierwszego rzędu z warunkiem początkowym można sformułować następująco:
znaleźć funkcję (zmienną stanu) y(t) zmiennej niezależnej t spełniającą równanie:

$$\frac{dy}{dt} = F(t, y) \quad oraz \quad y(t_0) = y_0$$

lub równania jak powyższe (w przypadku wielu takich równań).

I.Numeryczne metody rozwiązywania równań różniczkowych zwyczajnych

Wykorzystując możliwość programowania w Matlabie, użytkownik może zaprogramować swoje własne skrypty realizujące znane metody numeryczne rozwiązywania równań różniczkowych zwyczajnych. Do najprostszych należą metody Eulera, wyprowadzane wprost ze wzoru Taylora. Dokładniejsze są tzw. metody Runge-go-Kutty oraz metody wielokrokowe.
Metody Eulera dzielą się na trzy rodzaje: zwykła metoda Eulera, wsteczna oraz ulepszona metoda Eulera (metoda trapezowa).

1. Zwykła metoda Eulera

Metoda ta rozwiązuje zagadnienie początkowe postaci y'= F(t, y) z warunkiem początkowym $y(t_o)=y_o$ wykorzystując następujące równanie:

$$\frac{y_{n+1} - y_n}{h} \approx y_n^{'}$$

stąd po podstawieniu yn'= F(tn, yn) otrzymujemy:

$$y_{n+1}= y_n + h\,F(t_n,y_n)$$

Jest to tzw. metoda jawna (schemat otwarty).

2. Wsteczna metoda Eulera

Wsteczna metoda Eulera oparta jest na zależności:

$y_{n+1}= y_n + h\,F(t_n+1,y_{n+1})$

Jest to metoda niejawna (schemat zamknięty).

Dokładność wstecznej metody Eulera jest taka sama, jak zwykłej metody Eulera.

3. Ulepszona metoda Eulera (metoda Heuna, metoda trapezowa)

Ulepszona metoda Eulera (znana również pod nazwą metody trapezowej) jest bardziej dokładna i stabilna niż zwykła i wsteczna metoda Eulera. Jest ona połączeniem obu tych metod. Rozwiązanie równania $y' = F(t, y)$ oparte jest na zależności:

$$y_{n+1} = y_n + h/2\ [F(t_{n+1}, y_{n+1}) + F(t_n, y_n)]$$

Ulepszona metoda Eulera jest metodą niejawną (schemat uwikłany).

Przykład 1

Rozwiąż następujące równanie różniczkowe zwyczajne z podanym warunkiem początkowym:

$$\frac{dy}{dt} = \frac{t}{y} \qquad y(0)=1$$

za pomocą zwykłej, wstecznej i ulepszonej metody Eulera w przedziale <0, 0.5> z krokiem h=0.05 oraz porównać graficznie wyniki tych obliczeń z rozwiązaniem analitycznym.

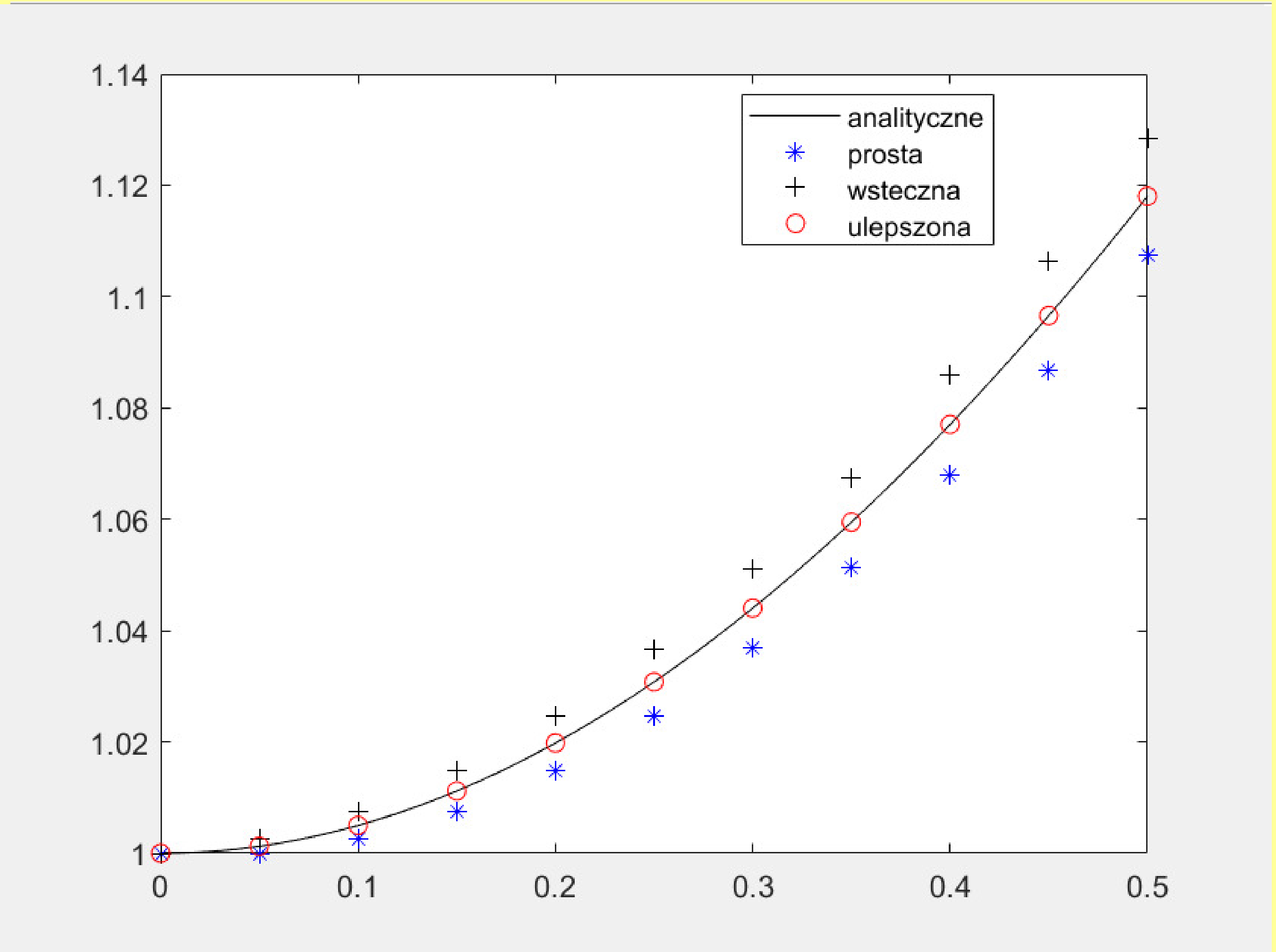
Rozwiązanie analityczne: $y = \sqrt{x^2 + 1}$

Eulera

zwykła

wsteczna

ulepszona



II. Metody ode

W Matlabie można rozwiązywać układy równań różniczkowych zwyczajnych, korzystając z kilku rodzajów funkcji ode (akronim ang. ***o**rdinary **d**ifferential **e**quations*). Funkcje te można podzielić na dwie grupy.

Pierwsza z nich przeznaczona jest do rozwiązywania równań i układów równań tzw. dobrze uwarunkowanych. Należą do niej funkcje:

a) ode45 (z użyciem jednokrokowej metody Rungego-Kutty rzędu 4 i 5)

b) ode23 (z użyciem jednokrokowej metody Rungego-Kutty rzędu 2 i 3)

c) ode113 (z użyciem wielokrokowej metody Adamsa-Bashfortha-Moultona, najlepsza z tej grupy)

Druga przeznaczona jest do rozwiązywania równań i układów równań źle uwarunkowanych, czyli sztywnych. Należą do niej takie funkcje jak: ode l5s, ode 23s, ode23t, ode23tb. W rozwiązaniach układów źle uwarunkowanych występują bardzo duże oraz bardzo małe stałe czasowe, w związku z tym układy takie są znacznie mniej stabilne niż układy dobrze uwarunkowane.

Składnia wszystkich w/w funkcji jest jednakowa:

[t,y] =funkcja ode (plik ode, przedział,y_0)

Parametry wyjściowe:

t - wektor kolumnowy wartości argumentów (np. chwil czasu) dla których obliczane było rozwiązanie

y - macierz rozwiązań. Każda kolumna jest wektorem reprezentującym wartości jednej ze zmiennych stanu w punktach określonym wektorem t.

Argumenty (parametry wejściowe):

przedział - wektor określający przedział całkowania. W przypadku wektora dwuelementowego [t_o, t_k] całkowanie będzie wykonywane od chwili to do chwili t_k, zaś w przypadku wektora o większej liczbie elementów rozwiązania będą wykonywane wyłącznie w chwilach określonych poprzez ten wektor.

y_o - wektor kolumnowy warunków początkowych

plik ode - łańcuch znaków, określający nazwę funkcji zdefiniowanej w m-pliku lub zdefiniowanej inline. Funkcja ta zawiera definicję rozwiązywanego układu równań różniczkowych. Nazwa tej funkcji jest dowolna. Funkcja jest zawsze dwuargumentowa. Argumentami są: **t** (skalar) oraz **y** (kolumnowy wektor stanu).

Ogólna postać tej funkcji jest następująca:
function F = plik ode(t,y)

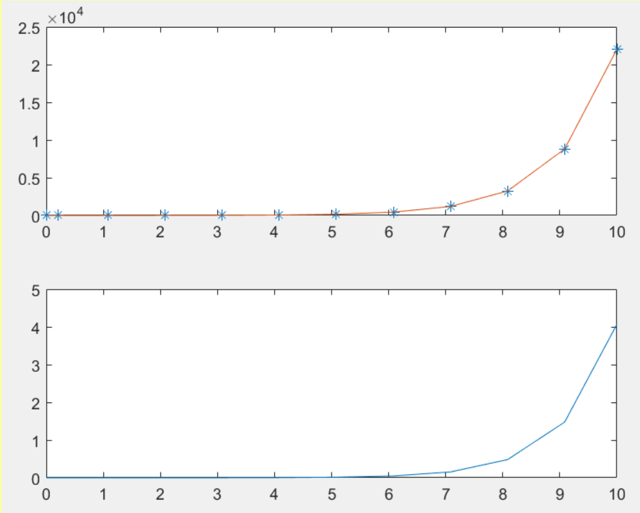
Rozwiązywanie równań pierwszego rzędu

Należy zauważyć, że rozwiązując numerycznie równanie różniczkowe, otrzymujemy na ogół tylko tablicę przybliżeń y, dokładnego rozwiązania y(t) w punktach t. Wykorzystując tę tablicę, można albo narysować wykres szukanej funkcji, albo przybliżyć ją odpowiednią funkcją za pomocą interpolacji.

Przykład.
Rozwiąż następujące zagadnienie początkowe w przedziale <0, T>:

$$\frac{dx}{dt} = x \quad oraz \quad x(0) = 1$$

```
function ex2
x0=1;
T = 10;
s=ode45(@fun,[0 T],x0);
subplot(2,1,1);
plot(s.x,s.y,'*');
yt=exp(s.x);
hold on;
plot(s.x,yt,'-');
subplot(2,1,2);
dy=s.y-yt;
plot(s.x,dy);
    function dx=fun(t,x)
        dx=x;
    end
end
```



Rozwiązywanie równań wyższego rzędu

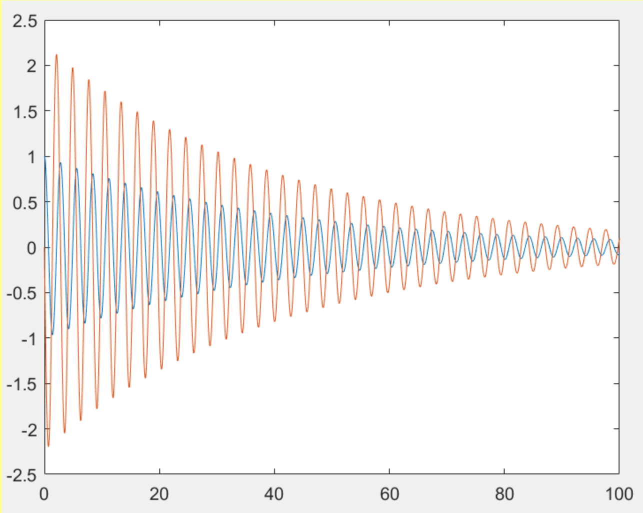
Podane wyżej funkcje ode rozwiązują wyłącznie równania lub układy równań różniczkowych zwyczajnych pierwszego rzędu (tzn. tylko z pierwszymi pochodnymi). Dlatego w przypadku równań różniczkowych wyższego rzędu musimy je najpierw zamienić na układ równań pierwszego rzędu, stosując odpowiednie zmienne (zmienne stanu).

Rozwiążmy równanie oscylatora harmonicznego z tłumieniem:

$$\frac{d^2x}{dt^2} + \frac{b}{m}\frac{dx}{dt} + \frac{k}{m}x = 0 \quad \text{z warunkami: } x(0) = 1, v(0) = 0$$

Zamieniamy równanie drugiego rzędu na układ dwóch równań rzędu pierwszego, wprowadzając nowe zmienne (zmienne stanu): x1 =x oraz x2=x'.

```
function main
clear; clc; m=12; k=2;b=0.1;
options=odeset('MaxStep',0.01);
s=ode45(@ff,[0 100],[1 0],options);
plot(s.x,s.y);%s.x -czas, s.y(1,:)-wychylenie,s.y(2,:)-prędkość
function dx=ff(t,x); %definicja równania
dx(2)=-k/m*x(1)-b/m*x(2);
dx(1)=x(2);%x-funkcja,dx(1)-jej pierwsza poch.,dx(2)druga poch.
end
end
```



Rozwiązywanie zagadnień brzegowych

dla równań różniczkowych cząstkowych

Równania różniczkowe cząstkowe w technice

Równania różniczkowe cząstkowe opisują zmienność systemów z kilkoma zmiennymi, w zastosowaniach technicznych z czasami zmiennymi przestrzennymi. Wynikają z zastosowania polowych praw fizyki do modelowania zjawisk dynamicznych zachodzących w systemach z parametrami rozłożonymi. Rozwiązanie polega na wyznaczeniu czasowo-przestrzennej odpowiedzi systemu dynamicznego.

Przykłady:

- natężenie pola elektromagnetycznego pod linią przesyłową wysokiego napięcia,
- opis dyfuzji ciepła w czujniku temperatury z obudową,
- rozkład pola magnetycznego w pracującym silniku (problem 2D/3D + czas),
- propagacja sygnału w linii telekomunikacyjnej (1D + czas),
- zmienna z czasem prędkość wzdłuż kolumny samochodów w ruchu miejskim (1D+czas),
- zmienny z czasem rozkład prędkości wiatru i ciśnienia atmosferycznego w układzie współrzędnych biegunowych (2D+czas).

Klasyfikacja problemów brzegowych

Dla ogólnego równania różniczkowego cząstkowego (czas może być wybraną zmienną *x* lub *y*):

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} = F\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)$$

charakter rozwiązania i klasa opisywanych problemów zależy od współczynników *A*, *B*, *C*.

Równanie eliptyczne	Równanie paraboliczne	Równanie hiperboliczne
$B^2 - 4AC < 0$ Np. równanie Poissona $\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$ Opisujące pole statyczne (temperatury, potencjału) ze źródłem *f*. Dla *f*=0 jest to równanie Laplace'a.	$B^2 - 4AC = 0$ Przykład: Równanie dyfuzji (ciepła, masy): $\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$ *D* – współczynnik dyfuzji	$B^2 - 4AC > 0$ Przykład: Równanie falowe: $\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2}$ *v* – prędkość propagacji fali

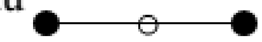
Warunki brzegowe (definiujące warunki na granicy analizowanego obszaru):

Dirichleta (zadana wartość pola na brzegu) $u = U$	**Neumanna** (zadana pochodna w kierunku normalnym do brzegu) $\frac{\partial u}{\partial n} = U'$

Podstawowe idee rozwiązania problemu metodą różnic skończonych

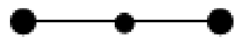
Zależność różniczkową zamienić na zależność różnicową na węzłach i rozwiązać układ równań

Pierwsza pochodna → dwupunktowy iloraz różnicowy centralny pierwszego rzędu

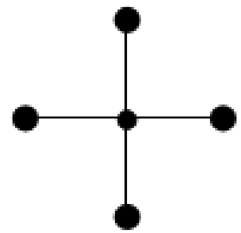


$$\left.\frac{\partial u(x,y)}{\partial x}\right|_{x=x_i,\,y=y_j} \approx \Delta_{2c}u_i = \frac{u(x_{i+1},y_j)-u(x_{i-1},y_j)}{x_{i+1}-x_{i-1}} = \frac{u_{i+1,j}-u_{i-1,j}}{2h}$$

Druga pochodna → trzypunktowy iloraz różnicowy w przód drugiego rzędu



$$\left.\frac{\partial^2 u(x,y)}{\partial x^2}\right|_{x=x_i,\,y=y_j} \approx \Delta_3^2 u = \frac{\Delta_2 u_i - \Delta_2 u_{i-1}}{h} = \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}$$



Laplasjan → różnica pięciopunktowa (ten sam przyrost h na zmiennych x i y)

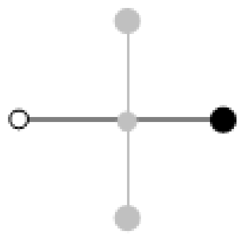
$$\nabla^2 u(x,y) \approx \nabla_5^2 u = \nabla_{3,x}u + \nabla_{3,y}u = \frac{1}{h^2}\left(u_{i+1,j}+u_{i,j+1}+u_{i-1,j}+u_{i,j-1}-4u_{i,j}\right)$$

Warunek brzegowy (zakładamy dla prostoty zapisu brzeg wzdłuż $x=x_0$):

Dirichleta: $u(x_0,y)=g(y) \rightarrow u_{0,j}=g(y_j)$ (zadana wartość węzłów brzegowych, bez rysunku)

Neumanna: $\dfrac{\partial u(x_0,y)}{\partial n}=f(y) \rightarrow \dfrac{u_{1,j}-u_{-1,j}}{2h}=f(y_j) \rightarrow u_{-1,j}=u_{1,j}-2hf(y_j)$

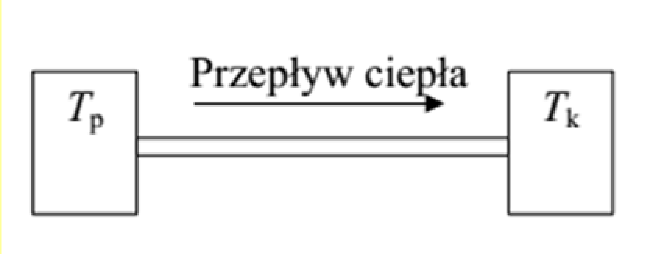
czyli tworzymy sztuczny punkt poza obszarem dla schematu $\nabla_5^2 u$



Przykład: Jednowymiarowy ustalony przepływ ciepła (równanie Laplace'a).

Rozwiążmy problem ustalonego przepływu ciepła w cienkim metalowym pręcie izolowanym z boku, z jednej strony (x=0) nagrzewanym źródłem o stałej temperaturze T_p, a z drugiej strony (x=L) chłodzonym odbiornikiem ciepła wymuszającym temperaturę.

Problem ten opisuje następujące równanie Laplace'a:



$$K\frac{\partial^2 T(x)}{\partial x^2} = \frac{\partial T}{\partial t}$$

Z warunkami początkowymi: $T(t,0) = T_p = 0$ i $T(t,L) = T_k = 100$

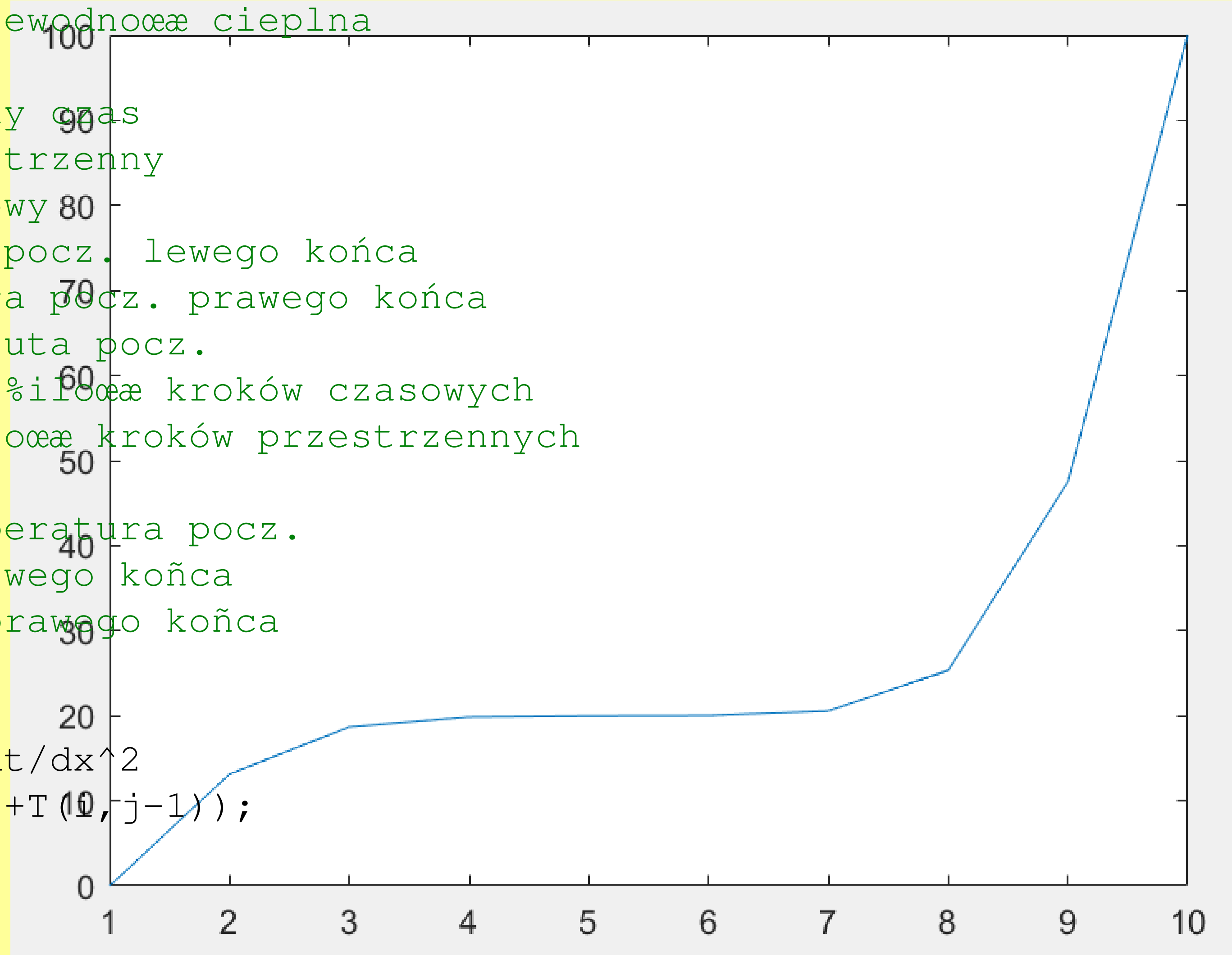
Dokonujemy dyskretyzacji równania:

$$\frac{\partial^2 T(x)}{\partial x^2} \rightarrow \frac{T_{n+1}^i - T_n^i}{dt} = K\frac{T_n^{i+1} - 2T_n^i + T_n^{i-1}}{(dx)^2}$$

Po przekształceniu otrzymujemy:

$$T_{n+1}^i = T_n^i + K\frac{dt}{(dx)^2}(T_n^{i+1} - 2T_n^i + T_n^{i-1})$$

```
clc;
D=0.01;%lub K - przewodnoœæ cieplna
L=1;%d³ugosc prêta
tmax=100;%maksymalny czas
dx=0.1; %krok przestrzenny
dt=0.1; %krok czasowy
TL=0; %temperatura pocz. lewego końca
TP=100; %temperatura pocz. prawego końca
Tpocz=20; %temperatuta pocz.
NT=round(tmax/dt); %iloœæ kroków czasowych
NX=round(L/dx); %iloœæ kroków przestrzennych
T=zeros(NT,NX);
T(1,:)=Tpocz; %temperatura pocz.
T(:,1)=TL;% temp lewego koñca
T(:,NX)=TP; %temp prawego koñca
for i=1:NT
    for j=2:NX-1
T(i+1,j)=T(i,j)+D*dt/dx^2
*(T(i,j+1)-2*T(i,j)+T(i,j-1));
    end;
    plot(T(i,:));
    %input('');
    pause(0.1);
  end;
```



Funkcja pdepe

Polecenie **pdepe** służy do rozwiązywania 1-wymiarowych równań parabolicznych i eliptycznych.

Before you can code the equation, you need to rewrite it in a form that the `pdepe` solver expects. The standard form that `pdepe` expects is

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\left(x,t,u,\frac{\partial u}{\partial x}\right).$$

Written in this form, the PDE becomes

$$\frac{\partial u}{\partial t}=x^{-1}\frac{\partial}{\partial x}\left(x^1\frac{\partial u}{\partial x}\right)$$

With the equation in the proper form you can read off the relevant terms:

- $m=1$
- $c\left(x,t,u,\frac{\partial u}{\partial x}\right)=1$
- $f\left(x,t,u,\frac{\partial u}{\partial x}\right)=\frac{\partial u}{\partial x}$
- $s\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$

Składnia:

sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan)

pdeModeler

Create complex 2-D geometries by drawing, overlapping, and rotating basic shapes

The PDE Modeler app provides an interactive interface for solving 2-D geometry problems. Using the app, you can create complex geometries by drawing, overlapping, and rotating basic shapes, such as circles, polygons and so on. The app also includes preset modes for applications, such as electrostatics, magnetostatics, heat transfer, and so on.

When solving a PDE problem in the app, follow these steps:

Create a 2-D geometry.

Specify boundary conditions.

Specify equation coefficients.

Generate a mesh.

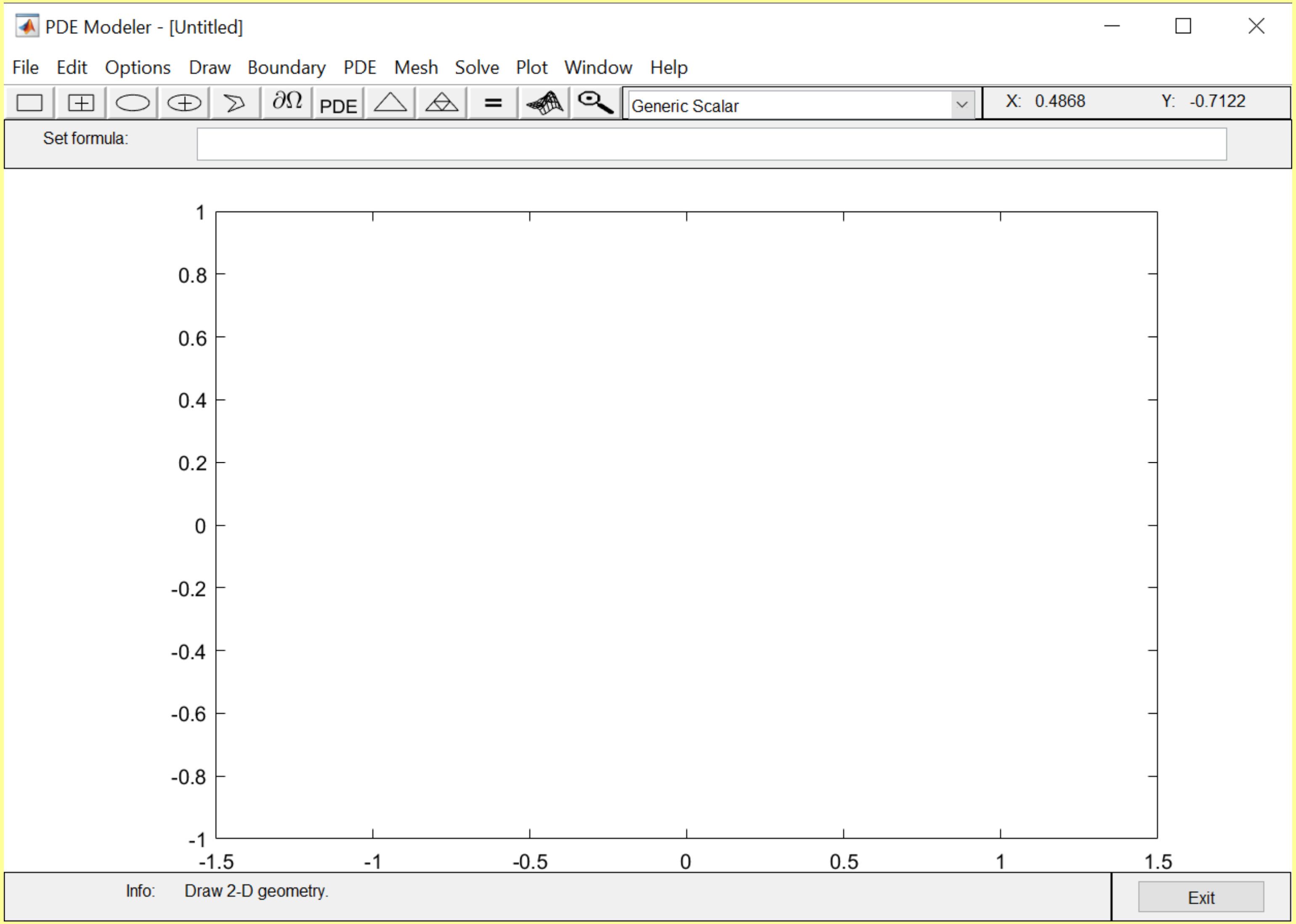
Specify parameters for solving a PDE. The set of parameters depends on the type of PDE. For parabolic and hyperbolic PDEs, these parameters include initial conditions.

Solve the problem.

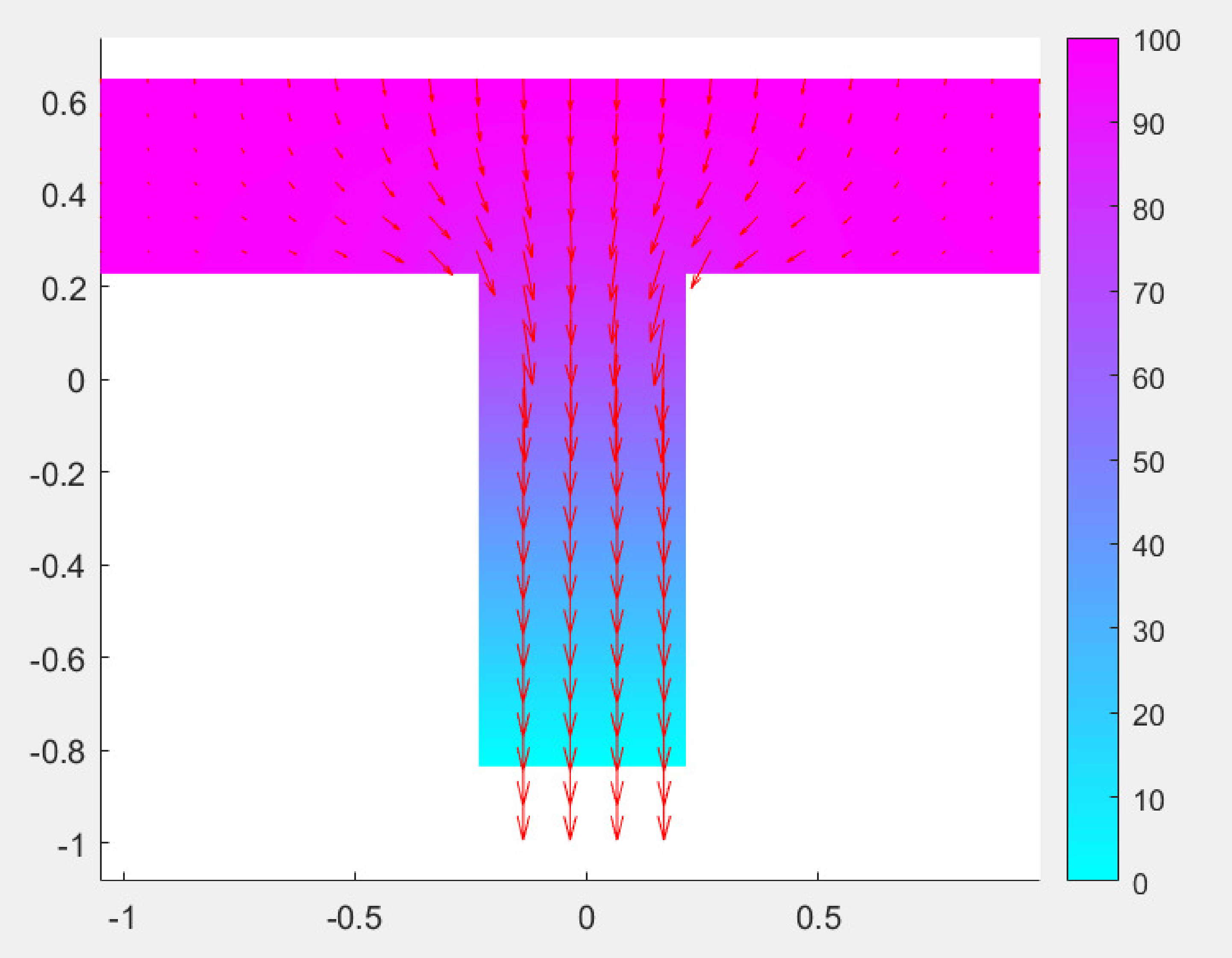
Specify plotting parameters and plot the results.

You can choose to export data to the MATLAB® workspace from any step in the app and continue your work outside the app.

Narzędzie pdeModeler



Przykład: przepływ ciepła



DZIĘKUJĘ ZA UWAGĘ