Numerical Methods
for heat and flow phenomena

PDE 4 - Matlab's built-in solvers – `bvp4c`

Thomas Weinhart

December 4, 2017

Boundary value problems (BVP)

The initial boundary value problem:

Find $T(t,x)$ on $x_l < x < x_r$, $0 < t < t_{\max}$, satisfying

$$\frac{\partial \rho C T}{\partial t} = \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) + s,$$

with initial and boundary conditions

$$T(0,x) = T_0, \quad T(t,x_l) = T_l, \quad T(t,x_r) = T_r.$$

Boundary value problems (BVP)

The initial boundary value problem:

Find $T(t,x)$ on $x_l < x < x_r$, $0 < t < t_{\max}$, satisfying

$$\cancel{\frac{\partial \rho C T}{\partial t}} = \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) + s,$$

with initial and boundary conditions

$$T(0,x) = T_0, \quad T(t,x_l) = T_l, \quad T(t,x_r) = T_r.$$

Often we are only interested in the final steady state.

Boundary value problems (BVP)

The initial boundary value problem:

Find $T(t,x)$ on $x_l < x < x_r$, $0 < t < t_{\max}$, satisfying

$$\cancel{\frac{\partial \rho C T}{\partial t}} = \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) + s,$$

with initial and boundary conditions

$$T(0,x) = T_0, \quad T(t,x_l) = T_l, \quad T(t,x_r) = T_r.$$

Often we are only interested in the final steady state.

Can we solve the steady-state heat equation directly?

Boundary value problems (BVP)

The steady-state heat equation

Find temperature $T(x)$ on $x_l < x < x_r$ satisfying

$$\frac{d}{dx}\left(\lambda \frac{dT}{dx}\right) = -s$$

with $T(x_l) = T_l,\ T(x_r) = T_r.$

Boundary value problems (BVP)

The steady-state heat equation

Find temperature $T(x)$ on $x_l < x < x_r$ satisfying

$$\frac{d}{dx}\left(\lambda \frac{dT}{dx}\right) = -s$$

with $T(x_l) = T_l$, $T(x_r) = T_r$.

This is a 2nd order ODE.

Boundary value problems (BVP)

The steady-state heat equation

Find temperature $T(x)$ on $x_l < x < x_r$ satisfying

$$\frac{d}{dx}\left(\lambda\frac{dT}{dx}\right) = -s$$

with $T(x_l) = T_l,\ T(x_r) = T_r.$

This is a 2nd order ODE.
We rewrite it as a 1st order ODE:

$$\text{Find } \begin{pmatrix} T(x) \\ f(x) \end{pmatrix} \text{ satisfying } \frac{d}{dx}\begin{pmatrix} T \\ f \end{pmatrix} = \begin{pmatrix} f/\lambda \\ -s \end{pmatrix}$$

$$\text{with } \begin{pmatrix} T(x_l) \\ T(x_r) \end{pmatrix} = \begin{pmatrix} T_l \\ T_r \end{pmatrix}.$$

Solving BVP's with Matlab

MATLAB has its own solver for BVP's:

$$\text{Find } y(x) \text{ satisfying } \frac{d}{dx}y(x) = f(x, y)$$
$$\text{with } y(x_a) = y_a,\ y(x_b) = x_b.$$

Syntax:

```
sol = bvp4c(odefun,bcfun,solinit)
```

Solving BVP's with Matlab

MATLAB has its own solver for BVP's:

Find $y(x)$ satisfying $\frac{d}{dx}y(x) = f(x, y)$
with $y(x_a) = y_a$, $y(x_b) = x_b$.

Syntax:

```
sol = bvp4c(odefun,bcfun,solinit)
```

Input arguments:

- function handle `dy = odefun(x,y)`
- function handle `res = bcfun(ya,yb)`
- struct solinit (`solinit.x`, `solinit.y`)

Output arguments:

- struct `sol` (`sol.x`, `sol.y`)

Solving BVP's with Matlab

MATLAB has its own solver for BVP's:

Find $y(x)$ satisfying $\frac{d}{dx}y(x) = f(x,y)$
with $y(x_a) = y_a$, $y(x_b) = x_b$.

Syntax:

```
sol = bvp4c(odefun,bcfun,solinit)
```

Input arguments:

- function handle `dy = odefun(x,y)`
- function handle `res = bcfun(ya,yb)`
- struct solinit (`solinit.x`, `solinit.y`)

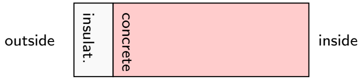
Output arguments:

- struct `sol` (`sol.x`, `sol.y`)

`solinit` is created using solinit=`bvpinit(xmesh,y0)`.

Example: Heat equation

Consider the problem from the last worksheet:



First, phrase it as an BVP:

Find $T(t, x)$ on $x_l < x < x_r$ satisfying

$$\frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) = -s,$$

with boundary conditions $T(t, x_l) = T_l, \quad T(t, x_r) = T_r.$

with $x_l = 0$, $x_r = 0.4\,\mathrm{m}$, $T_l = \text{-}10^\circ\mathrm{C}$, $T_r = 25^\circ\mathrm{C}$, $s = 0$,

$$\lambda = \begin{cases} 0.1 & x < 0.1 \\ 1.2 & \text{else.} \end{cases}$$

Example: Steady-state heat equation

Rewrite it as a 1st order ODE:

$$\text{Find } \begin{pmatrix} T(x) \\ f(x) \end{pmatrix} \text{ satisfying } \frac{d}{dx}\begin{pmatrix} T \\ f \end{pmatrix} = \begin{pmatrix} f/\lambda \\ -s \end{pmatrix}$$

$$\text{with } \begin{pmatrix} T(x_l) \\ T(x_r) \end{pmatrix} = \begin{pmatrix} T_l \\ T_r \end{pmatrix}.$$

Matlab implementation

```
% heat transfer coeff
l1 = 0.1; l2 = 1.2;
l = @(x) l1 + (l2-l1) * heaviside(x-0.1);
% heat source
s = 0;

%ode function
odefun=@(x,y)[y(2)/l(x); -s];
...
```

Example: Steady-state heat equation

```
% boundary conditions, given as res=0.
bcfun=@(yl,yr) [yl(1)-(-10);yr(1)-20];

% spatial discretisation/initial guess
x = linspace(0,0.4);
solinit=bvpinit(x,[0 0]);

% solve the boundary value problem
sol=bvp4c(odefun,bcfun,solinit);
subplot(1,2,1); plot(sol.x,sol.y(1,:),'.-')
subplot(1,2,2); plot(sol.x,sol.y(2,:),'.-')
```