Partial Differential Equations in OpenModelica

Jan Šilar

First Faculty of Medicine
Charles University
Prague, Czech Republic

OpenModelica Workshop 2016

PDE

- unknown – functions of >2 coordinates (time, space)
- e.g. advection transport, vibration of a string, heat transfers, …

- PDE extension proposed (based on Levon Saldamli)
- Partially implemented in omc

Only subset of extension supported

- one dimension
- first derivative

- focus on hyperbolic eq., conservation laws (advection, string, hydrodynamics ...)

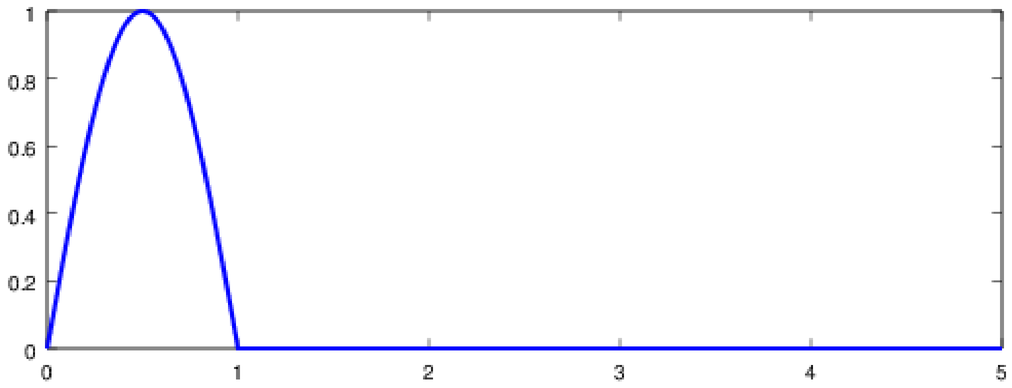
Advection equation

$$u(t,x):\mathbb{R}^2 \rightarrow \mathbb{R}$$

$$\frac{\partial u}{\partial t}+2\frac{\partial u}{\partial x}=0 \quad t\in[0,T], x\in[0,5]$$

$$u(t,0)=0$$

$$u(0,x)=\begin{cases}\sin(\pi x) & x\in[0,1]\\ 0 & x\in[1,5]\end{cases}$$



```
model advection "advection"
  parameter DomainLineSegment1D omega(L = 5, N=200);
  field Real u(domain = omega);
initial equation
  u = if omega.x < 1 then sin(3.14*omega.x) else 0 indomain omega;
equation
  der(u) + 2*pder(u,x) = 0                         indomain omega;
  u = 0                                            indomain omega.left;
  u = extrapolateField()                           indomain omega.right;
end advection;
```

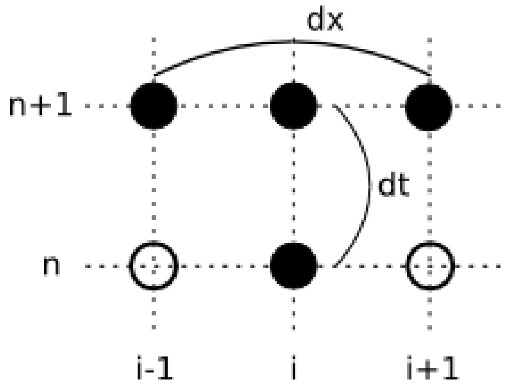
`extrapolateField()` .. temporary solution

Equations solved by method of lines

During translation (Front-end):

- field → array (domain.N = number of elements)
- spatial derivative → difference ($\frac{\partial u}{\partial x} \rightarrow \frac{u_{i+1} - u_{i-1}}{2\,dx}$)
- PDE → system of ODEs ($\frac{d\,u_i}{d\,t} + 2\frac{u_{i+1} - u_{i-1}}{2\,dx} = 0 \quad i = 2..N-1$)
- resulting systém processed by compiler solved by current simulation runtime
- implementation only in compiler

Boundaries



Space difference evaluation (→ PDE)

- inner points only
- outer points – BC or extrapolation

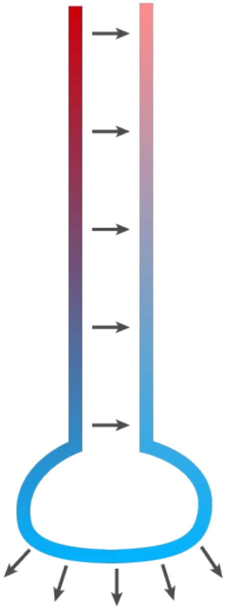
How many BC and where?

- eigenvalue analysis

$$\frac{\partial \bar{u}}{\partial t} + \boldsymbol{A}(\bar{u}) \frac{\partial \bar{u}}{\partial x} = 0$$ (conservation law)

  - positive λ – left, negative – right BC
  - magnitude – speed of waves (time step)
  - not implemented extrapolateField() instead for now

Counter-current heat exchange



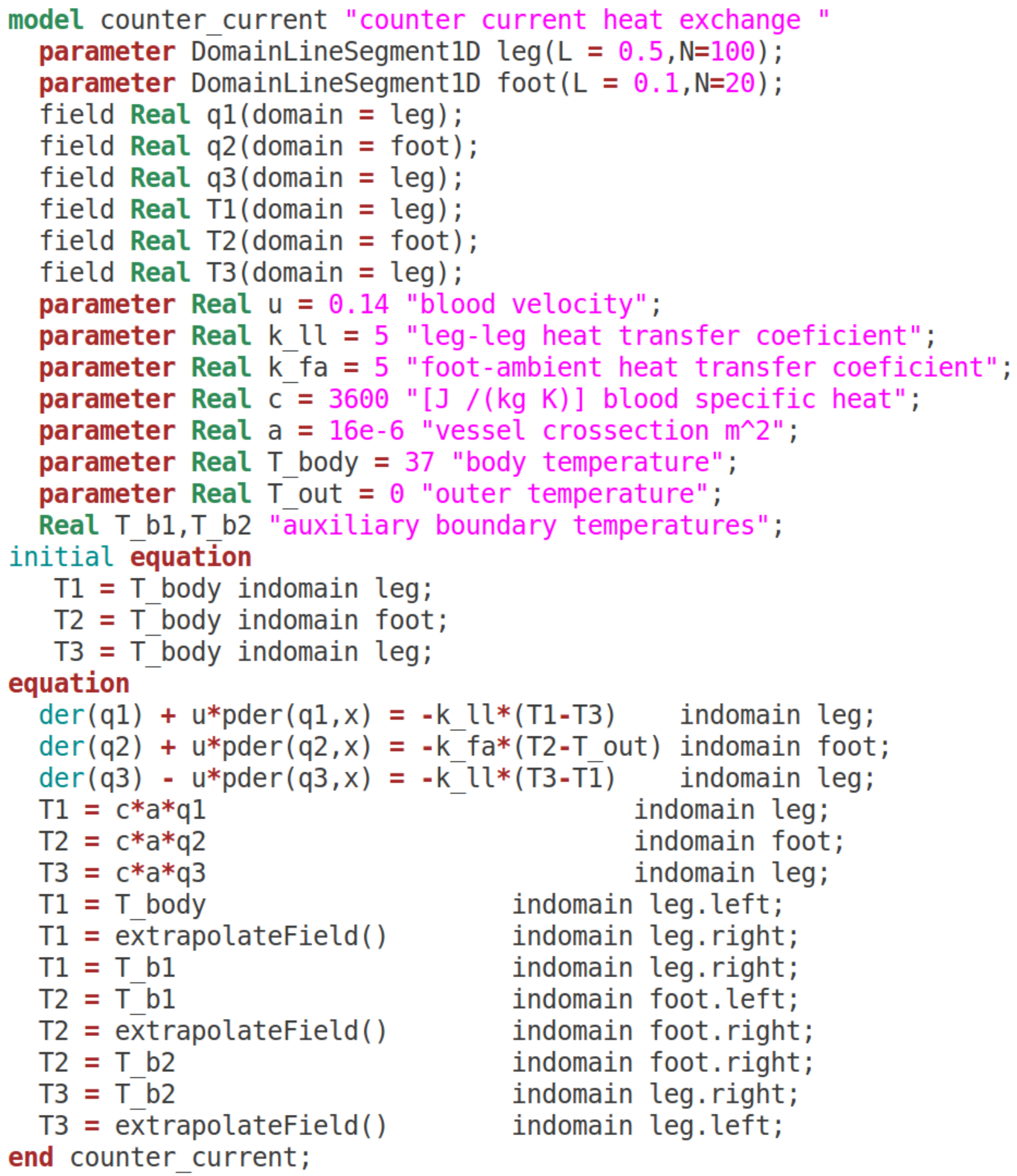
$$\frac{\partial q_1}{\partial t} + u\frac{\partial q_1}{\partial x} = -k_l(T_3 - T_1)$$

$$\frac{\partial q_2}{\partial t} + u\frac{\partial q_2}{\partial x} = -k_f(T_2 - T_{out})$$

$$\frac{\partial q_3}{\partial t} + u\frac{\partial q_3}{\partial x} = -k_l(T_1 - T_3)$$

$$T_i = c\,q_i$$

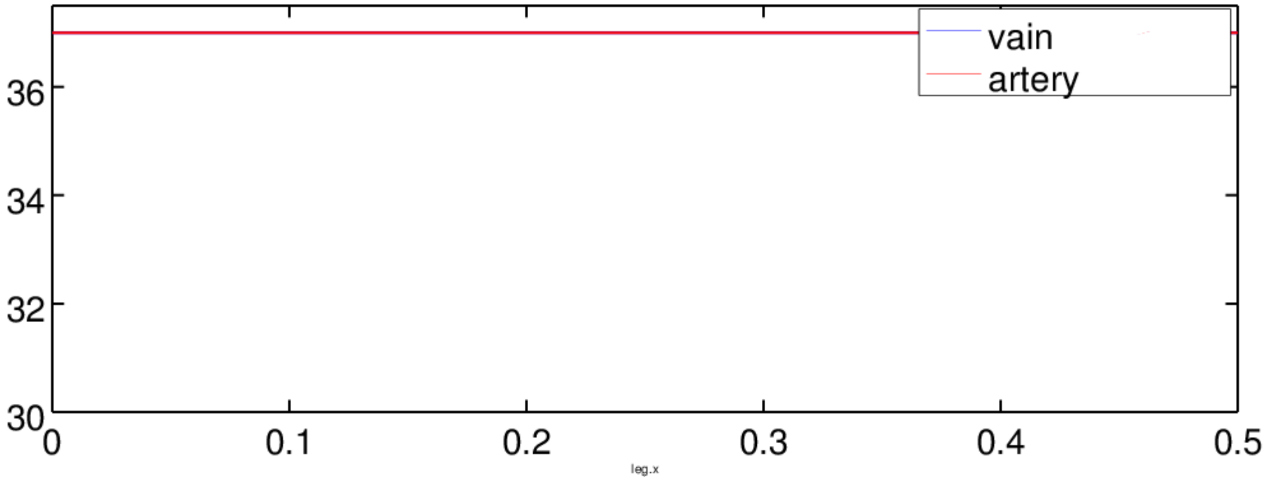
Counter-current – model

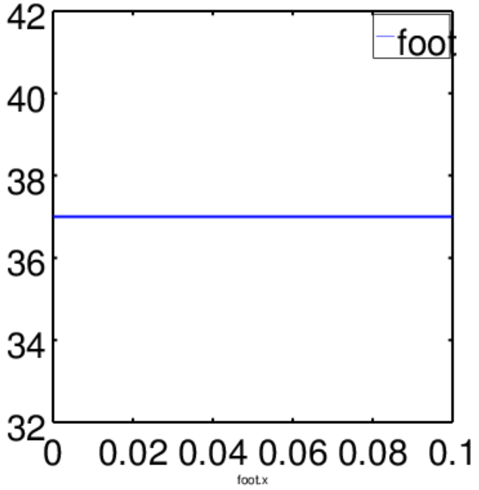


```
model counter_current "counter current heat exchange "
  parameter DomainLineSegment1D leg(L = 0.5,N=100);
  parameter DomainLineSegment1D foot(L = 0.1,N=20);
  field Real q1(domain = leg);
  field Real q2(domain = foot);
  field Real q3(domain = leg);
  field Real T1(domain = leg);
  field Real T2(domain = foot);
  field Real T3(domain = leg);
  parameter Real u = 0.14 "blood velocity";
  parameter Real k_ll = 5 "leg-leg heat transfer coeficient";
  parameter Real k_fa = 5 "foot-ambient heat transfer coeficient";
  parameter Real c = 3600 "[J /(kg K)] blood specific heat";
  parameter Real a = 16e-6 "vessel crossection m^2";
  parameter Real T_body = 37 "body temperature";
  parameter Real T_out = 0 "outer temperature";
  Real T_b1,T_b2 "auxiliary boundary temperatures";
initial equation
   T1 = T_body indomain leg;
   T2 = T_body indomain foot;
   T3 = T_body indomain leg;
equation
  der(q1) + u*pder(q1,x) = -k_ll*(T1-T3)    indomain leg;
  der(q2) + u*pder(q2,x) = -k_fa*(T2-T_out) indomain foot;
  der(q3) - u*pder(q3,x) = -k_ll*(T3-T1)    indomain leg;
  T1 = c*a*q1                            indomain leg;
  T2 = c*a*q2                            indomain foot;
  T3 = c*a*q3                            indomain leg;
  T1 = T_body                    indomain leg.left;
  T1 = extrapolateField()        indomain leg.right;
  T1 = T_b1                      indomain leg.right;
  T2 = T_b1                      indomain foot.left;
  T2 = extrapolateField()        indomain foot.right;
  T2 = T_b2                      indomain foot.right;
  T3 = T_b2                      indomain leg.right;
  T3 = extrapolateField()        indomain leg.left;
end counter_current;
```

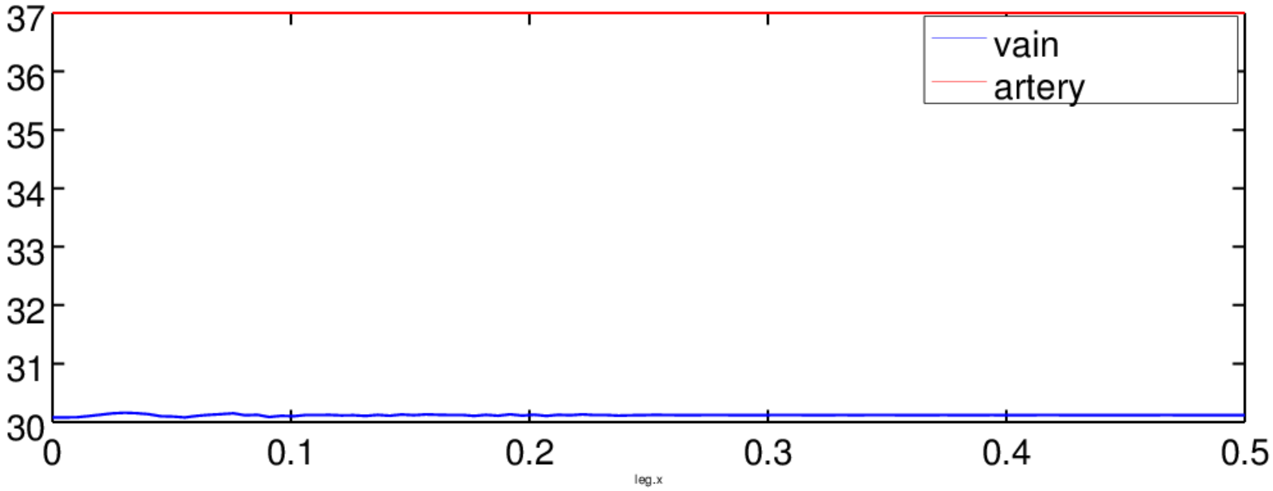
Results – no exchange

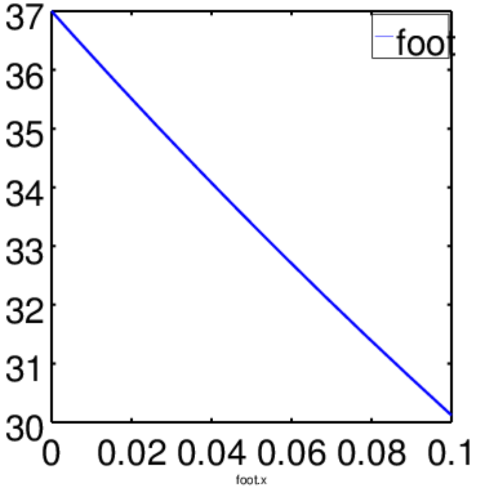
Initially:





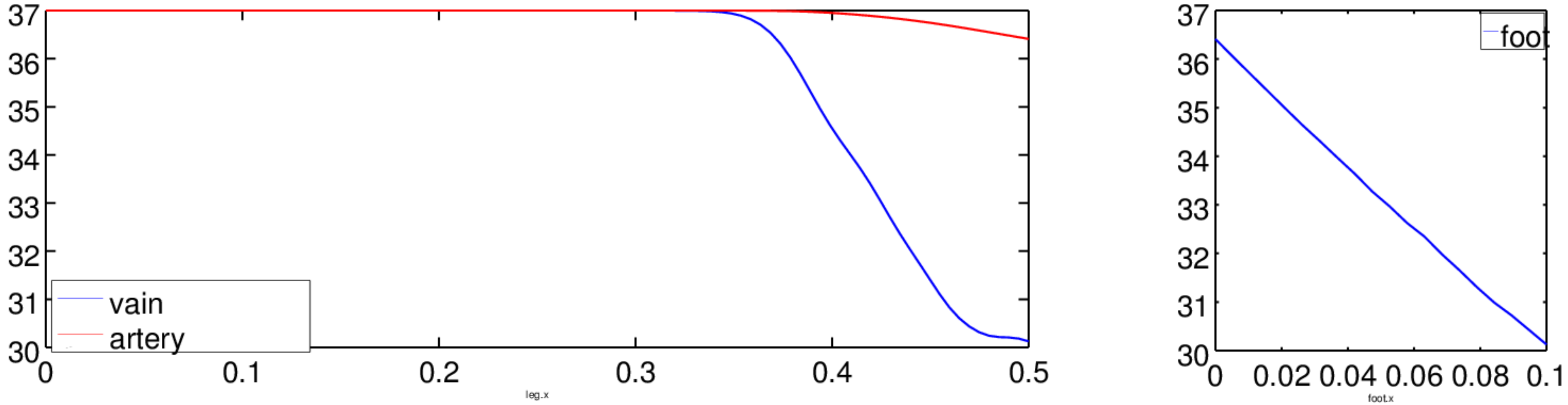
t = 5s



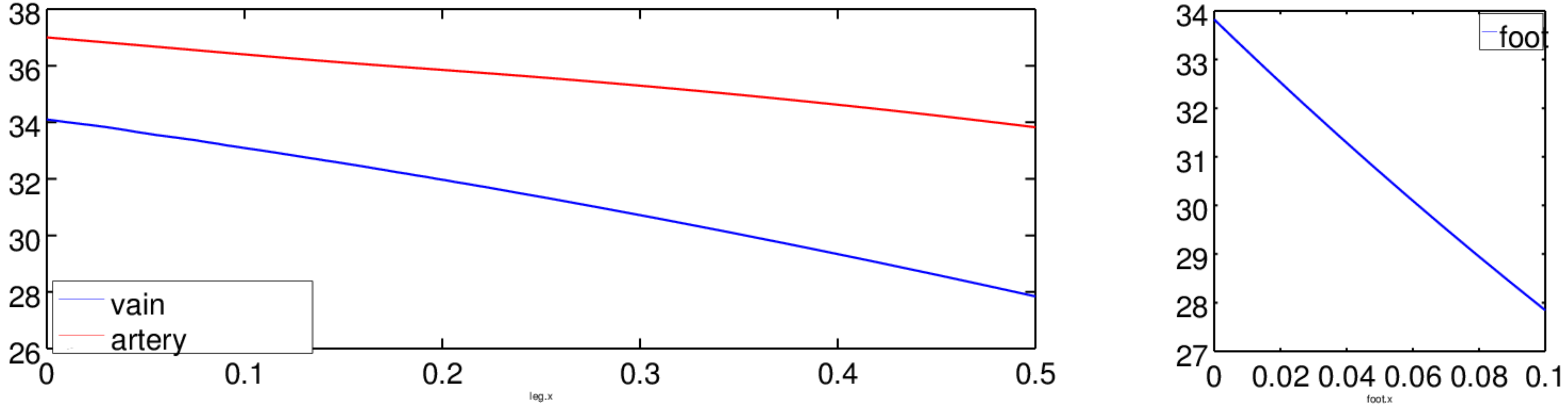


Results – with exchange

t = 1s



t = 5s



Practical notes

- compiler flag --grammar=“PDEModelica“
- time solver
  - BTCS → Radau1 (Implicit Euler)
  - Lax-Friedrichs (not implemented yet)→ Euler
- set time step ~ space step
  - cfl condition (explicit methods)

$$C=\frac{u\Delta t}{\Delta x}\leq C_{max}$$

- not merged yet
- not suppurted in OMEdit – edit models in external editor, simulate manually

Future work

- integrate in public repo. (soon)
- annotations for solver setup (method, time step)
- eigenvalue analysis
  - to determine time step
  - to determine BCs required – check model, add extrapolation
- add support in OMEdit
  - switch to enable extension
  - field plotting

Projects in physiology

- Kidney – Loop of Henle – counter-current exchange – urine filtration
- Breathing in snow (avalanche)
  - advection and diffusion of oxygen and CO_2

End

Thank you.