jaxFlowSim

Diego Renner

January 16, 2024

Introduction

What is jaxFlowSim?

- 1D-haemodynamics solver
- written in JAX
- differentiable

Motivation

Use and Novelty

- towards personalised medicine
- parameter inference
- sensitivity analysis

1D-Navier Stokes Equations

$$
\begin{aligned}
\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial z} &= \mathbf{S}(\mathbf{U}), \;\; t > 0, \;\; z \in [0, l], \\
\mathbf{U}(z; 0) &= \mathbf{U}_0(z), \;\; z \in [0, l], \\
\mathbf{U}(0; t) &= \mathbf{U}_L(t), \;\; t > 0, \\
\mathbf{U}(l; t) &= \mathbf{U}_R(t), \;\; t > 0,
\end{aligned}
\tag{1}
$$

$$
\mathbf{U} := \begin{bmatrix} A \\ Q \end{bmatrix}, \; \mathbf{F}(\mathbf{U}) := \begin{bmatrix} Q \\ \frac{Q^2}{A} + \frac{\beta A^{\frac{3}{2}}}{3\rho\sqrt{A_0}} \end{bmatrix}, \; \mathbf{S}(\mathbf{U}) := \begin{bmatrix} 0 \\ -22\frac{\mu}{\rho}\frac{Q}{A} \end{bmatrix}. \tag{2}
$$

$\mathbf{U_0} \widehat{=}$ initial condition, $\mathbf{U_L} \widehat{=}$ left boundary values, $\mathbf{U_R} \widehat{=}$ right boundary values, $A \widehat{=}$ cross-section, $A_0 \widehat{=}$ reference cross-section, $Q \widehat{=}$ volumetric flow-rate, $\beta \widehat{=}$ elasticity coefficient, $\rho \widehat{=}$ blood density, $\mu \widehat{=}$ blood dynamic viscosity.

Tube Law

$$P(z;t) := P_{ext}(z;t) + \beta\left(\sqrt{\frac{A(z;t)}{A_0(z)}} - 1\right), \tag{3}$$

$$\beta(z) := \frac{\sqrt{\pi} E h_0(z)}{(1-\nu^2)\sqrt{A_0(z)}}. \tag{4}$$

$P \hat{=}$ pressure, $P_{ext} \hat{=}$ external pressure,
$E \hat{=}$ Young's modulus, $h_0 \hat{=}$ reference vessel wall thickness, $\nu \hat{=}$ Poisson's ratio (elasticity parameter).

Initial Conditions

$$u(z;0) \equiv 0, \qquad z \in [0, l], \tag{5}$$

$$A(z;0) = A_0(z), \qquad z \in [0, l], \tag{6}$$

$$Q(z;0) = u(z;0)A(z;0) \equiv 0, \qquad z \in [0, l]. \tag{7}$$

Inlets, Juntions, Outlets

Inlets

set P from data $\rightarrow$ set u, Q, A, c through linear extrapolation of charateristics

set Q from data $\rightarrow$ set u, A, c, P through linear extrapolation of charateristics

Junctions

solve linear system of equations consisting of:

- conservation of mass,
- conservation of pressure,
- extrapolation of charateristics

Outlets

0D-/ lumped parameter model: three element Windkessel (RCR) model

Numerical Methods

1D-Model

FV method: MUSCL scheme with Lax-Friedrichs Flux

Junctions & Outlets

Newton method

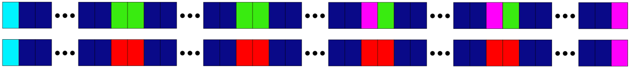
Code Structure

```
def runSimulation(config_filename, J)
  config = loadConfig(config_filename)
  simulation_data = buildArterialNetwork(config)

  P_t = [0]

  converged = False
  while not converged:
    t = 0
    i = 0
    P_t_temp = P_t
    while t < T:
      dt = computeDt(simulation_data)
      simulation_data = setBoundaryValues(simulation_data, dt)
      simulation_data = muscl(simulation_data, dt)
      P_t[i,:] = savePressure(simulation_data)
      t = t + dt
    i = i + 1
    if i >= J:
      break
    converged = checkConv(P_t, P_t_temp)
```

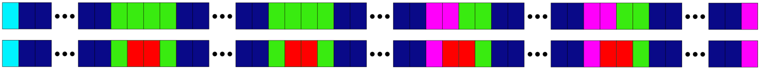
Listing: $dt \widehat{=}$timestep(CFL), setBoundaryValues$\widehat{=}$inlet(from data), outlet(Windkessel),junctions(conservation laws), muscl$\widehat{=}$Monotonic Upstream-centered Scheme for Conservation Laws(Finite Volume)

Padding

without padding

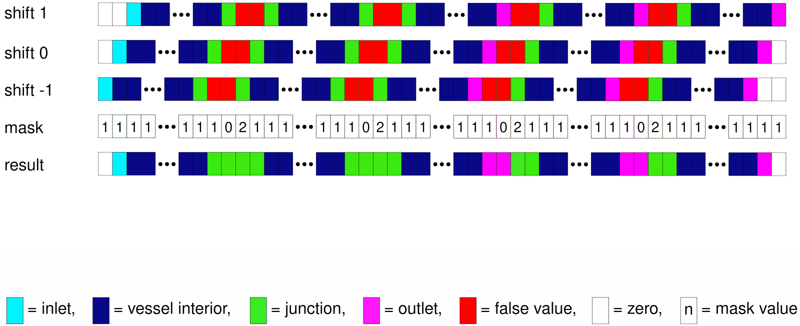


with padding

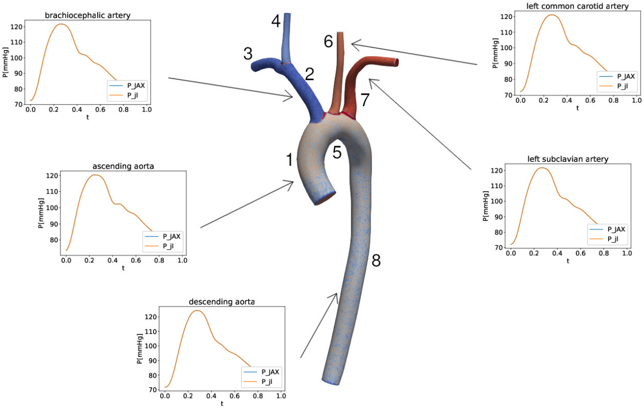


= inlet, = vessel interior, = junction, = outlet, = false value, = zero, n = mask value

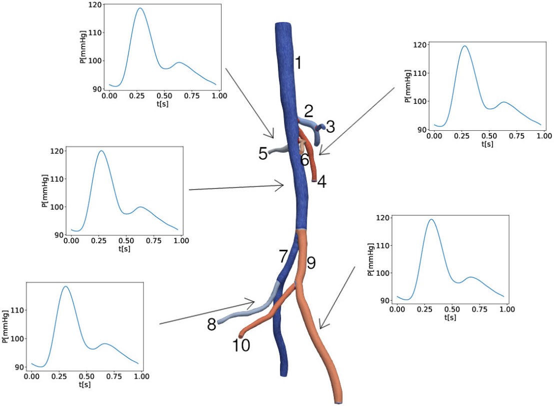
Masking


shift 1
shift 0
shift -1
mask
1 1 1 1 ••• 1 1 1 0 2 1 1 1 ••• 1 1 1 0 2 1 1 1 ••• 1 1 1 0 2 1 1 1 ••• 1 1 1 0 2 1 1 1 ••• 1 1 1 1
result
= inlet, = vessel interior, = junction, = outlet, = false value, = zero, n = mask value

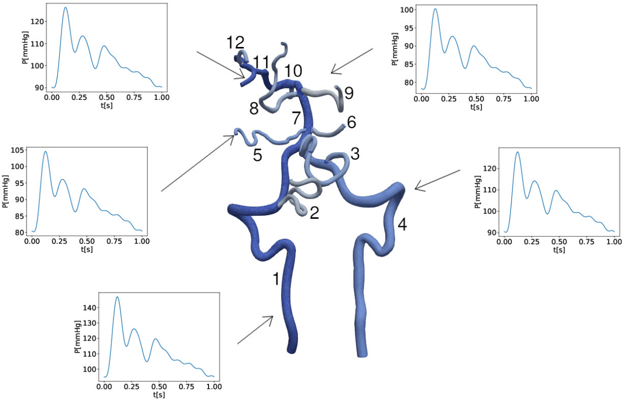
Model: Aorta (0007_H_AO_H)



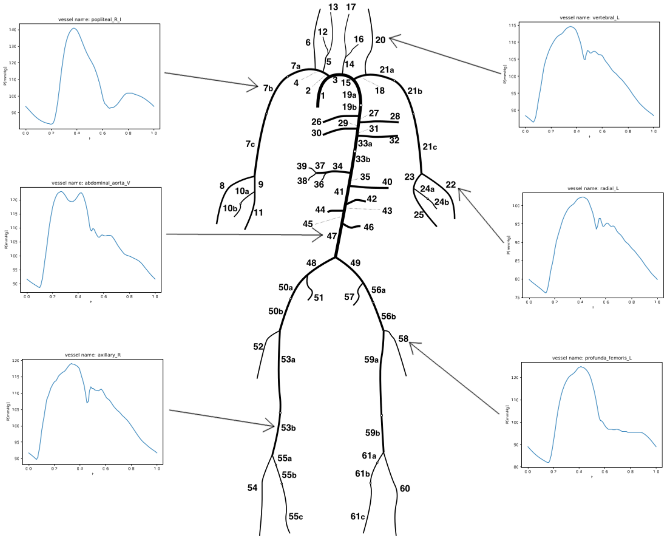
Model: Abdominal Arteries (0029_H_ABAO_H)



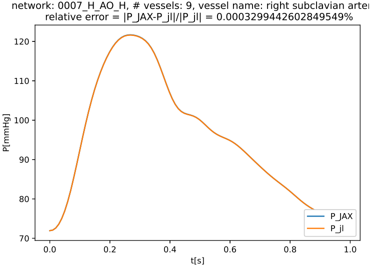
Model: Cerebellar Arteries (0053_H_CERE_H)

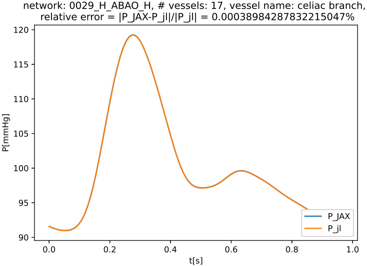


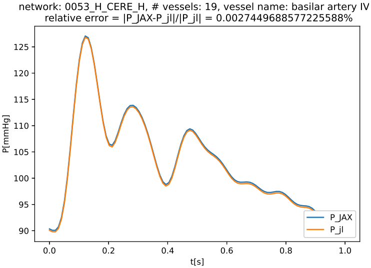
Model: ADAN56

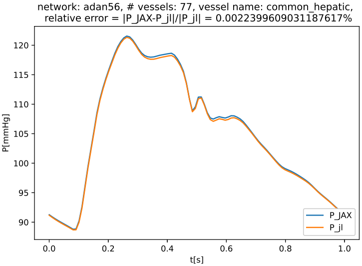


Validation

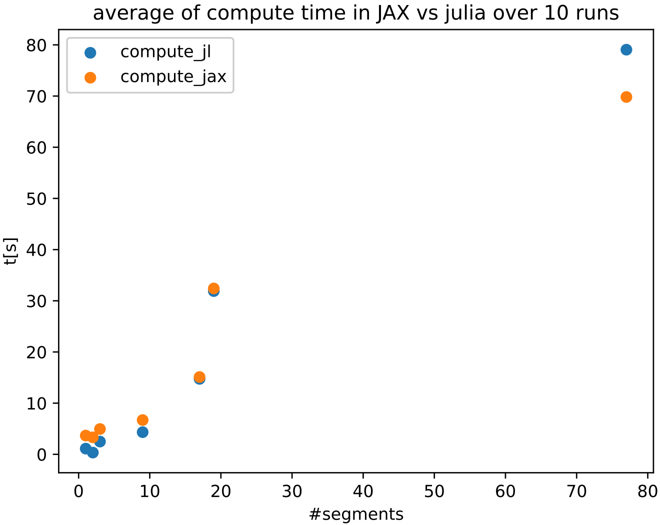

network: 0007_H_AO_H, # vessels: 9, vessel name: right subclavian arter
relative error = |P_JAX-P_jl|/|P_jl| = 0.0003299442602849549%
P[mmHg]
t[s]
P_JAX
P_jl


network: 0029_H_ABAO_H, # vessels: 17, vessel name: celiac branch,
relative error = |P_JAX-P_jl|/|P_jl| = 0.00038984287832215047%
P[mmHg]
t[s]
P_JAX
P_jl


network: 0053_H_CERE_H, # vessels: 19, vessel name: basilar artery IV
relative error = |P_JAX-P_jl|/|P_jl| = 0.0027449688577225588%
P[mmHg]
t[s]
P_JAX
P_jl


network: adan56, # vessels: 77, vessel name: common_hepatic,
relative error = |P_JAX-P_jl|/|P_jl| = 0.0022399609031187617%
P[mmHg]
t[s]
P_JAX
P_jl

Comparison


average of compute time in JAX vs julia over 10 runs
compute_jl
compute_jax
t[s]
#segments
0
10
20
30
40
50
60
70
80

Inference

Demo

inferring an outlet resistance parameter from precomputed data

Future Work

Main Points of Interest

- improving performance (GPU optimization)
- fine tuning parameter inference
- sensitivity analysis

Questions?

Thank you for your attention!