One Dimensional Blood Flow Model in LifeV

Vincent MARTIN (Post-Doc, MOX, Politecnico di Milano)

Outline

- continuous/discrete problem.
- implementation
- boundary/compatibility conditions
- structure

1D Blood Flow Models

1D problem: $\varphi(\alpha, \beta_0, \beta_1, K_r, A_0) \mapsto (A, Q)$.

$$\begin{aligned}
\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial z} &= 0, \\
\frac{\partial Q}{\partial t} + \frac{\partial}{\partial z}\left(\frac{\alpha Q^2}{A}\right) + \frac{A}{\rho}\frac{\partial P}{\partial z} + K_r \frac{Q}{A} &= 0,
\end{aligned}$$

$$P = \psi(A, \mathbf{p}) = \beta_0 \left(\left(\frac{A}{A_0} \right)^{\beta_1} - 1 \right).$$

1D Linear problem: $\varphi(R', L', C') \mapsto (\hat{P}, \hat{Q})$.

$$\begin{aligned}
C' \frac{\partial \hat{P}}{\partial t} + \frac{\partial \hat{Q}}{\partial z} &= 0, \\
L' \frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{Q}}{\partial z} &= -R' \hat{Q}.
\end{aligned}$$

1D Blood Flow Model

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial z} = 0,$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial z}\left(\frac{\alpha Q^2}{A}\right) + \frac{A}{\rho}\frac{\partial P}{\partial z} + K_r \frac{Q}{A} = 0,$$

$$P = \psi(A, \mathbf{p}) = \beta_0 \left(\left(\frac{A}{A_0} \right)^{\beta_1} - 1 \right).$$

Rewritten in a conservative form with $\mathbf{U} = [A, Q]^\top$ and $\mathbf{p} = (\alpha, \beta_0, \beta_1, K_r, A_0)(z)$:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial}{\partial z}[\mathbf{F}(\mathbf{U}, \mathbf{p})] + \mathbf{S}(\mathbf{U}, \mathbf{p}) = \mathbf{0},$$

$$\phi_0(\mathbf{U}(t, 0), \mathbf{p}) = q_0(t), \quad \forall t \in [0, T],$$

$$\phi_L(\mathbf{U}(t, L), \mathbf{p}) = q_L(t), \quad \forall t \in [0, T] \,.$$

Discrete Model: Order 2 Taylor-Galerkin

With $\mathbf{F}_{h,LW}(\mathbf{U}) = \mathbf{F}_h(\mathbf{U}) - \frac{\Delta t}{2}\mathbf{H}_h(\mathbf{U})\mathbf{S}_h(\mathbf{U})$

and $\mathbf{S}_{h,LW}(\mathbf{U}) = \mathbf{S}_h(\mathbf{U}) - \frac{\Delta t}{2}(\mathbf{S}_{\mathbf{U}})_h(\mathbf{U})\mathbf{S}_h(\mathbf{U})$,

search $\mathbf{U}_h^{n+1} \in \mathbf{V}_h = (P_1)^2$ for $n = 0, 1, ...$ such that $\forall \boldsymbol{\psi}_h \in \mathbf{V}_{h,0}$:

$$(\mathbf{U}_h^{n+1}, \boldsymbol{\psi}_h)_\Omega = (\mathbf{U}_h^n, \boldsymbol{\psi}_h)_\Omega + \Delta t \left(\mathbf{F}_{h,LW}(\mathbf{U}_h^n), \frac{d\boldsymbol{\psi}_h}{dz}\right)_\Omega - \Delta t \left(\mathbf{S}_{h,LW}(\mathbf{U}_h^n), \boldsymbol{\psi}_h\right)_\Omega$$

$$+ \frac{\Delta t^2}{2}\left((\mathbf{S}_{\mathbf{U}})_h(\mathbf{U}_h^n)\frac{\partial \mathbf{F}_h(\mathbf{U}_h^n)}{\partial z}, \boldsymbol{\psi}_h\right)_\Omega - \frac{\Delta t^2}{2}\left(\mathbf{H}_h(\mathbf{U}_h^n)\frac{\partial \mathbf{F}_h}{\partial z}(\mathbf{U}_h^n), \frac{d\boldsymbol{\psi}_h}{dz}\right)_\Omega$$

$$(\mathbf{U}_h^{n+1}, \boldsymbol{\psi}_h)_\Omega = (\mathbf{U}_h^n, \boldsymbol{\psi}_h)_\Omega + a_h(\mathbf{U}_h^n, \boldsymbol{\psi}_h; \mathbf{p}) \quad \forall \boldsymbol{\psi}_h \in \mathbf{V}_{h,0}$$

$$\Theta_0(\mathbf{U}^n, \mathbf{p})\mathbf{U}_0^{n+1} - \mathbf{T_0}(\mathbf{U^n}, \mathbf{p}) = \mathbf{0}$$

$$\Theta_L(\mathbf{U}^n, \mathbf{p})\mathbf{U}_{N+1}^{n+1} - \mathbf{T_L}(\mathbf{U^n}, \mathbf{p}) = \mathbf{0} \quad .$$

Discretizing the NL functions

$\mathbf{U}_h$ is in P_1. The discrete functions of $\mathbf{U}$, $\mathbf{F}_h(\mathbf{U})$ and $\mathbf{S}_h(\mathbf{U})$, are taken in P_1 and their derivatives, $\mathbf{H}_h(\mathbf{U})$ and $(\mathbf{S}_\mathbf{U})_h(\mathbf{U})$, in P_0.

For a vectorial function $v(\mathbf{U}) = [v_1, v_2]^\top$ and a matrix $M(\mathbf{U}) = (M_{\alpha\beta})_{\alpha,\beta=1,2}$, we use the functions

$$v_h(\mathbf{U}) = \begin{bmatrix} \sum_{i=0}^N v_1(\mathbf{U}_i)\psi_i \\ \sum_{i=0}^N v_2(\mathbf{U}_i)\psi_i \end{bmatrix}, \qquad \frac{\partial v_h}{\partial z}(\mathbf{U}) = \begin{bmatrix} \sum_{i=0}^N v_1(\mathbf{U}_i)\frac{d\psi_i}{dz} \\ \sum_{i=0}^N v_2(\mathbf{U}_i)\frac{d\psi_i}{dz} \end{bmatrix},$$

and

$$M_h(\mathbf{U}) = \left(M_{h,\alpha\beta}(\mathbf{U})\right)_{\alpha,\beta=1,2}, \quad M_{h,\alpha\beta}(\mathbf{U}) = \sum_{i=0}^{N-1} \tilde{M}_{h,\alpha\beta,\,i+1/2} 1_{i+1/2}$$

with the mean value $\tilde{M}_{h,\alpha\beta,\,i+1/2} = \frac{1}{2}\left(M_{\alpha\beta}(\mathbf{U}_i) + M_{\alpha\beta}(\mathbf{U}_{i+1})\right)$ over the element $[z_i, z_{i+1}]$, for $i = 0, \ldots, N-1$.

Implementation

- 2D vectorial, Linear / Non Linear.
- Order 2 explicit Taylor Galerkin scheme (NO slope limitors).
- Boundary + Compatibility cond. are treated explicitly after linearization.
- clarity and generality of the code rather than efficiency (1D...).
- separated the scheme part from the model description.
  ($\longrightarrow$ switch from one model to another)
- fully tested for Linear, seems OK for Non Linear.

Implementation

- initialization

1. build the 1D mesh and 1D dof. (new and ad'hoc)
2. use feSegP1 and quadRuleSeg3pt.
3. assemble constant mass and gradient matrices (standard LifeV procedure).
4. BC treatment for the mass matrix (non symmetrical so far).
5. factorize the mass matrix.

- iteration

1. update the Flux, Source: nodal vectors.
2. update the diffFlux, diffSource element matrices.
3. *assemble* the 16 tridiagonal matrices depending on current $\mathbf{U}_h^n$:
   `_M_massMatrixDiffSrcIJ, _M_stiffMatrixDiffFluxIJ,`
   `_M_gradMatrixDiffFluxIJ, _M_divMatrixDiffSrcIJ`
   .
4. 22 matrix – vector products to compute the rhs$(\mathbf{U}_h^n)$.
5. BC treatment (compute BC values and set them to rhs *(first dof / last dof)*).
6. 2 systems solves to find the new $(\mathbf{U}_h^{n+1})$.

$$
\begin{aligned}
(\mathbf{U}_h^{n+1}, \boldsymbol{\psi}_h)_\Omega \quad = \quad & \\
& (\mathbf{U}_h^n, \boldsymbol{\psi}_h)_\Omega \\
& +\Delta t \left(\mathbf{F}_h(\mathbf{U}_h^n), \frac{d\boldsymbol{\psi}_h}{dz} \right)_\Omega \\
& -\frac{\Delta t^2}{2} \left(\mathbf{H}_h(\mathbf{U}) \mathbf{S}_h(\mathbf{U}_h^n), \frac{d\boldsymbol{\psi}_h}{dz} \right)_\Omega \\
& +\frac{\Delta t^2}{2} \left((\mathbf{S}_\mathbf{U})_h(\mathbf{U}_h^n) \frac{\partial \mathbf{F}_h(\mathbf{U}_h^n)}{\partial z}, \boldsymbol{\psi}_h \right)_\Omega \\
& -\frac{\Delta t^2}{2} \left(\mathbf{H}_h(\mathbf{U}_h^n) \frac{\partial \mathbf{F}_h}{\partial z}(\mathbf{U}_h^n), \frac{d\boldsymbol{\psi}_h}{dz} \right)_\Omega \\
& -\Delta t \left(\mathbf{S}_h(\mathbf{U}_h^n), \boldsymbol{\psi}_h \right)_\Omega \\
& +\frac{\Delta t^2}{2} \left((\mathbf{S}_\mathbf{U})_h(\mathbf{U}) \mathbf{S}_h(\mathbf{U}_h^n), \boldsymbol{\psi}_h \right)_\Omega
\end{aligned}
$$

- mass matrix factorized. 1
- mass matrix $*\mathbf{U}_h$. 1
- gradient matrix $*\mathbf{F}_h(\mathbf{U}_h)$. 1
- **updated** $\mathbf{H}_h(\mathbf{U}_h)$-gradient matrices $*\mathbf{S}_h(\mathbf{U}_h)$. 4
- **updated** $(\mathbf{S}_\mathbf{U})_h(\mathbf{U}_h)$-diverg. matrices $*\mathbf{F}_h(\mathbf{U}_h)$. 4
- **updated** $\mathbf{H}_h(\mathbf{U}_h)$-stiffness matrices $*\mathbf{F}_h(\mathbf{U}_h)$. 4
- mass matrix $*\mathbf{S}_h(U_h)$. 0
- **updated** $(\mathbf{S}_\mathbf{U})_h(\mathbf{U}_h)$-mass matrices $*\mathbf{S}_h(\mathbf{U}_h)$. 4

Compatibility Conditions: linear explicit

Linearise around a constant state $\mathbf{U}_\sharp^n$: $\mathbf{U} = \mathbf{U}_\sharp^n + \mathbf{U}'$. At first order:

$$\frac{\partial \mathbf{U}'}{\partial t} + \mathbf{H}(\mathbf{U}_\sharp^n, \mathbf{p}) \frac{\partial \mathbf{U}'}{\partial z} + \mathbf{B}(\mathbf{U}, \mathbf{p}) = \mathbf{0} \ .$$

With $\mathbf{LHL}^{-1} = \mathbf{C} = \text{diag}(c_1, c_2)$, the pseudo-characteristic variable are:

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{U}} = \mathbf{L}(\mathbf{U}_\sharp^n), \qquad \mathbf{Z}' = \mathbf{Z} - \mathbf{Z}_\sharp^n = \mathbf{L}(\mathbf{U}_\sharp^n)(\mathbf{U} - \mathbf{U}_\sharp^n) = \mathbf{L}_\sharp^n \mathbf{U}',$$

and the linear equation becomes

$$\frac{\partial \mathbf{Z}'}{\partial t} + \mathbf{C}_\sharp^n \frac{\partial \mathbf{Z}'}{\partial z} + \mathbf{G}_\sharp^n(\mathbf{Z}', \mathbf{p}) = \mathbf{0},$$

with

$$\mathbf{G}_\sharp^n(\mathbf{Z}', \mathbf{p}) = \mathbf{L}_\sharp^n \mathbf{B}(\mathbf{U}, \mathbf{p}) - \mathbf{L}_\sharp^n \left(\mathbf{C}_\sharp^n \frac{\partial \mathbf{L}_\sharp^n}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial z} \right) \mathbf{U}' \ .$$

Boundary Conditions

The **compatibility condition** at *right bc* ($z = L$) reads, with $\mathbf{U}^n_{1,\star} = \mathbf{U}(t^n, L - c_1\Delta t)$ and $\mathbf{U}^n_\sharp = \mathbf{U}^n_L$ (for instance),

$$(\mathbf{l}_1^\top)^n_\sharp \mathbf{U}^{n+1}_L = (\mathbf{l}_1^\top)^n_\sharp \mathbf{U}^n_{1,\star} - \Delta t \left\{ \mathbf{L}^n_\sharp \mathbf{B}(\mathbf{U}^n_{1,\star}, \mathbf{p}) - \mathbf{L}^n_\sharp \left(\mathbf{C}^n_\sharp \frac{\partial \mathbf{L}^n_\sharp}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial z} \right) \mathbf{U}^n_{1,\star} \right\}_1$$

The **boundary condition** at *right bc* ($z = L$) reads

- Or: flux imposed $Q^{n+1}_L = q^d_L(t^{n+1})$.
- Or: pressure imposed $A^{n+1}_L = \psi^{-1}(p^d_L(t^{n+1}))$ (with ψ: pressure function).
- Or: "both imposed" through the incoming pseudo-characteristics:

$$(\mathbf{l}_2^\top)^n_\sharp \mathbf{U}^{n+1}_L = (\mathbf{l}_2^\top)^n_\sharp [a^d_L(t^{n+1}); q^d_L(t^{n+1})]$$

At *right and left bc*, two 2x2 linear systems to be solved.
$\rightarrow$ `Vec2D = std::pair<Real, Real>` introduced

To be done...

- move (?):
  - ⋆ `basicOneDMesh` to lifemesh
  - ⋆ `gracePlot` to lifefilters
  - ⋆ `dofOneD` to lifefem
- `tridiagMatrix` already in lifearray.
  Add a (lapack) linear solver interface in lifealg.
- `Vec2D` to be added both in lifearray/alg?

Future work

- multitube: Interface conditions.
- multiscale: A. Moura and...
- 1D parameters: space dependant: $(\alpha, \beta_0, \beta_1, K_r, A_0)(z)$.

Questions:

- .emacs with the new LifeV indentation convention?
- templates pre-compiled?
- ...