Some simple models for Vesicles and RBCs in Fluid Flow.

Towards Realistic Blood Flow Simulation. A HPC FEEL++ Framework

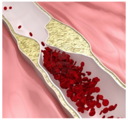
Workshop on PDE's Modelling and Theory
PALAIS DES SCIENCES DE MONASTIR-TUNISIE,
May. 9-10th 2018

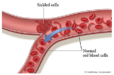
Mourad ISMAIL
Mourad.Ismail@univ-grenoble-alpes.fr
http://www-liphy.ujf-grenoble.fr/Mourad-Ismail
Grenoble Alpes University, France

Long term goals

Understand the physical mechanisms to some physio-cardiovascular diseases

- Atherosclerosis, Stenosis, Drepanocytosis (Sickle Cells Anemia), ...

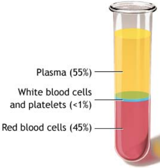


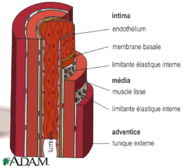


Blood characteristics

- RBCs are predominant from rheological point of view
- Blood $\simeq$ plasma (Newtonian fluid) + RBCs
- A dense suspension of micron sized RBCs

- Arterial Elasticity
  - Support the pressure of blood flux
  - Regulate the pulse of the cardiac flow (large arteries)



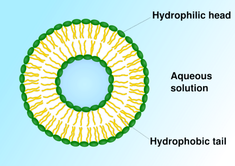


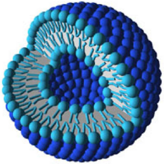
Challenges

- How to take into account the presence of RBCs? which models?
- Complex interactions at different levels :
  - Blood/Vessels
  - Plasma/RBCs
  - RBCs/RBCs
- Different kinds of deformations and different scales
  - Dynamic of RBCs in arteries
  - Deformation of RBCs in capillaries

Outline

Vesicle : A Simple RBC Model

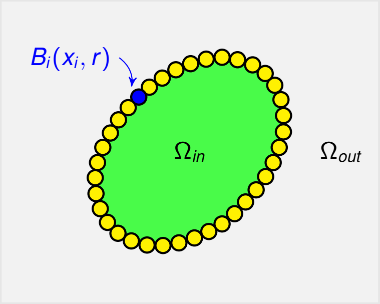




- Unilamellar vesicle : lipid bi-layer membrane
- Easy to produce in the laboratory
- Imitate some behaviors of red blood cells
  ➠ passive mechanical properties
- Properties of the membrane :
  - Nonporous : Conservation of inner fluid's Volume
  - Non extensible : Conservation of the membrane surface (perimeter in 2D)
  - Bending Energy (Helfrich energy)

From Fluid/Rigid Particles model to Vesicles [Coll. with A. Lefebvre] I

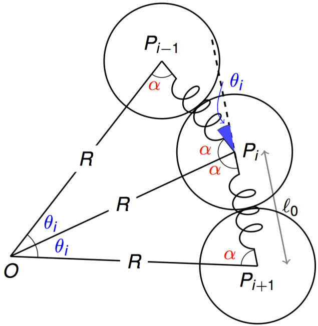
Vesicle membrane modeled by a “Necklace" of rigid particles



- Membrane modeled by a Necklace of small rigid particles
- Incompressibility and curvature forces modeled by interaction between rigid particles (springs)
- FEM and Penalty Methods

[IJNMF, 76, pp.835-854 (2014)]

From Fluid/Rigid Particles model to Vesicles [Coll. with A. Lefebvre] II



- Stretch/Compression springs f_i^s on B_i and bending springs f_i^b on B_i
- Classical Energy for strech/compression springs :

$$E_{st} = \sum_i k_{rp_i} \ell_i^2$$

with $k_{rp_i} = cst = k_{rp}$

- Bending spring energy :

$$E_b = \sum_i k_{a_i} (u_i \cdot v_i + 1)$$

with $k_{a_i} = cst = k_a$.

[IJNMF, 76, pp.835-854 (2014)]

Formulation I

- Let

$$K_B = \left\{ \begin{array}{l} v \in H_0^1(\Omega),\ \ \forall i,\ \ \exists (V_i, \omega_i) \in \mathbb{R}^2 \times \mathbb{R}, \\ v = V_i + \omega_i (x - x_i)^\perp \text{ a.e. in } B_i \end{array} \right\}$$
$$= \left\{ v \in H_0^1(\Omega),\ \ \mathrm{D}v = 0 \text{ a.e. in } B \right\}$$

- Find u in K_B and p in $L_0^2(\Omega)$ s.t.

$$(\mathcal{P}) \left\{ \begin{array}{l} 2\mu \displaystyle\int_\Omega \mathrm{D}u : \mathrm{D}\tilde{u} - \int_\Omega p \nabla \cdot \tilde{u} = \int_\Omega f \cdot \tilde{u},\ \ \forall \tilde{u} \in K_B, \\ \displaystyle\int_\Omega q \nabla \cdot u = 0,\ \ \forall q \in L_0^2(\Omega), \end{array} \right.$$

where $f = \displaystyle\sum_{i=1}^{N} f_i \mathbf{1}_{B_i}$.

Formulation II

- A penalty method is used to approximate the constraint problem $(\mathcal{P})$ by a sequence $(\mathcal{P}^{\varepsilon})$ of unconstrained problems:

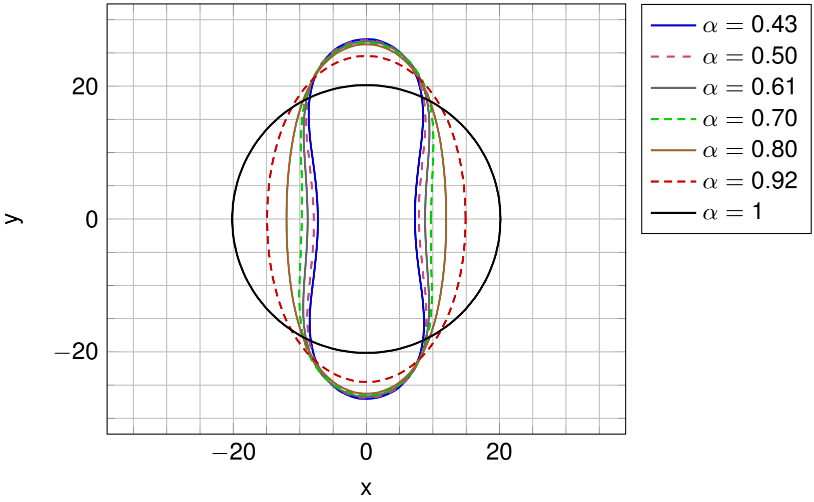
$$\begin{cases} \text{Find } u^{\varepsilon} \text{ in } H_0^1(\Omega) \text{ and } p^{\varepsilon} \text{ in } L_0^2(\Omega) \text{ such that} \\ 2\mu \displaystyle\int_{\Omega} \mathrm{D}u^{\varepsilon} : \mathrm{D}\tilde{u} + \frac{2}{\varepsilon} \int_{B} \mathrm{D}u^{\varepsilon} : \mathrm{D}\tilde{u} - \int_{\Omega} p^{\varepsilon} \nabla \cdot \tilde{u} \quad = \quad \int_{\Omega} f \cdot \tilde{u}, \\ \hfill \forall \quad \tilde{u} \in H_0^1(\Omega), \\ \hfill \displaystyle\int_{\Omega} q \nabla \cdot u^{\varepsilon} = 0, \quad \forall \quad q \in L_0^2(\Omega). \end{cases}$$

Algorithm

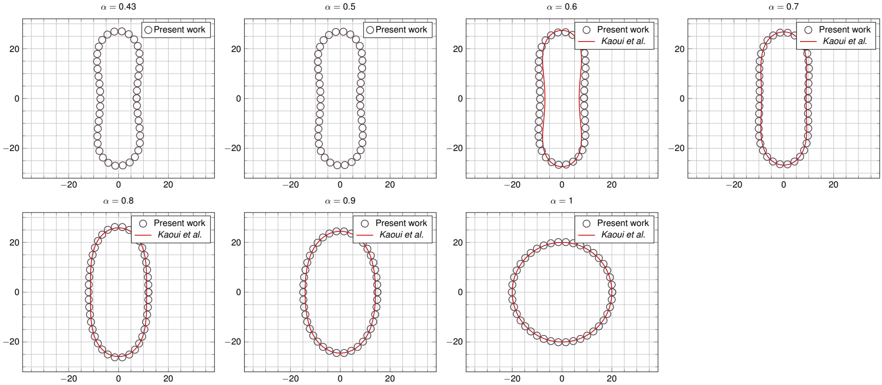
1. Compute (u^n, p^n) solution to $(\mathcal{P}^\varepsilon)$ with $B = B^n$ and $f = f^n$,
2. Compute the corresponding velocities of the particles: $\tilde{V}_i^n = \dfrac{1}{|B_i^n|}\displaystyle\int_{B_i^n} u^n$,
3. Deal with contacts: $\hat{V}^n = \Pi_{K_c^n}\tilde{V}^n$,
4. Deal with the volume constraint: $V^n = \Pi_{K_v^n}\hat{V}^n$,
5. Compute B^{n+1}: $x_i^{n+1} = x_i^n + hV_i^n$,

where Π_K denotes the projection onto K and is performed using a Uzawa algorithm.

Equilibrium Shapes I



Equilibrium Shapes II

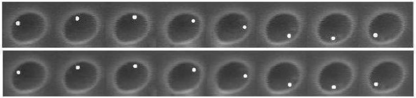


Comparison with results from [Kaoui et al. Phys. Rev. E, 83, 2011]

Tank Treading and Tumbling motions [Courtesy of T. Podgorski]

Small Viscosity Contrast ➠ Tank-Treading motion

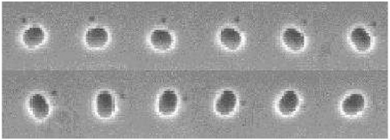
- vesicle reaches a stationary angle
- rotation of the membrane



Movie

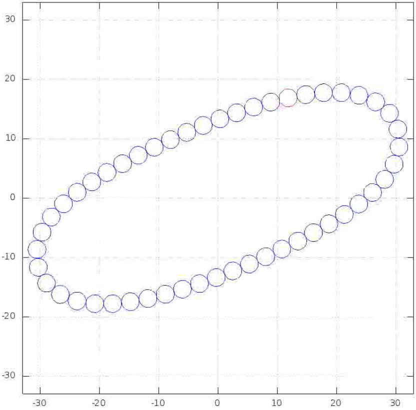
High viscosity ratio ➠ Tumbling motion

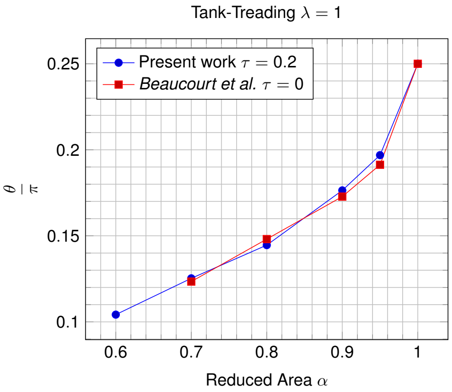
- vesicle in quasi solid rotation
- rotation velocity depends on the angle



Movie

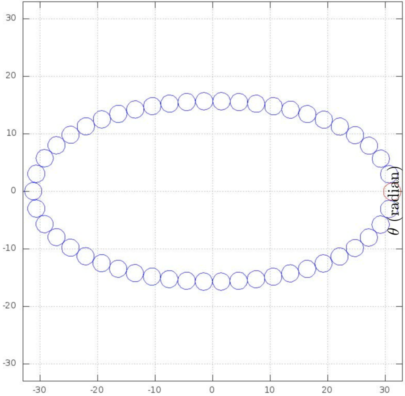
Tank-Treading angles

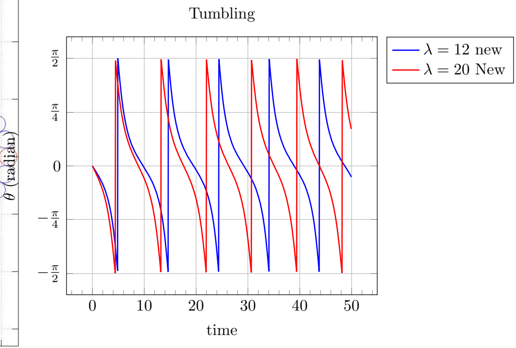




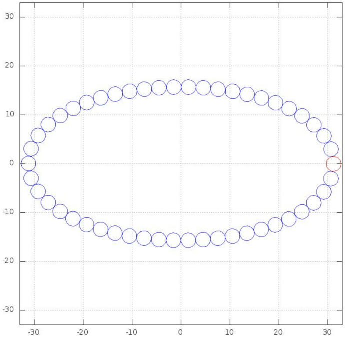
Comparison with results from [Beaucourt et al. Phys. Rev. E, 69, 2004]

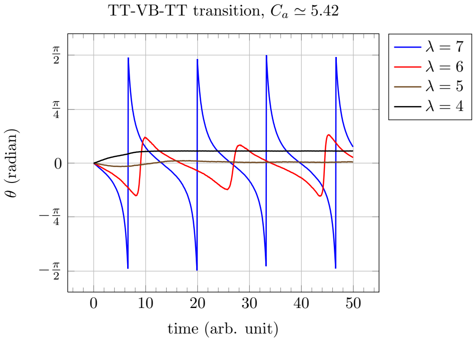
Tumbling


30
20
10
0
-10
-20
-30
-30
-20
-10
0
10
20
30


Tumbling
$\lambda = 12$ new
$\lambda = 20$ New
$\frac{\pi}{2}$
$\frac{\pi}{4}$
0
$-\frac{\pi}{4}$
$-\frac{\pi}{2}$
θ (radian)
0
10
20
30
40
50
time

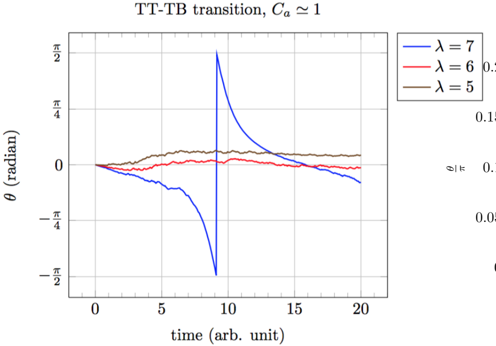
Transition Tank treading / Tumbling : Vacillating-Breathing

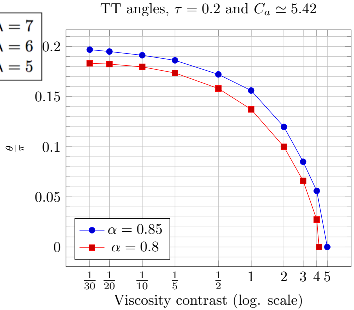




Effect of the Capillary Number

$$C_a = \frac{\mu_{out} R^3 \dot{\gamma}}{k_a 2r}$$





Outline

Collaborations and Funding

VivaBrain[1] and FEEL++[2] communities

- S. Bertoluzza (Italy)
- V. Chabannes **(former PhD) LIPhy/LJK**
- G. Coupier (LIPhy)
- V. Doyeux **(former PhD) LIPhy**
- Y. Guyot **(former M2 Student) LIPhy**
- T. Podgorski (LIPhy)
- S. Priem **(former PhD Student) LIPhy/LJK**
- C. Prud'homme (Strasbourg)
- T. Métivet **(PostDoc) LIPhy/LJK**
- V. Hubert (Strasbourg)
- G. Pena (Portugal)
- S. Salmon (Reims)
- M. Szopos (Strasbourg)
- R. Tarabay (PhD Student)
- Computer scientists
- Radiologists and Hospital Doctors
- ....

Funding

- ANR MOSICOB (2008–2012)
- ANR VivaBrain (2013–2017) **ANR-12-MONU-0010**

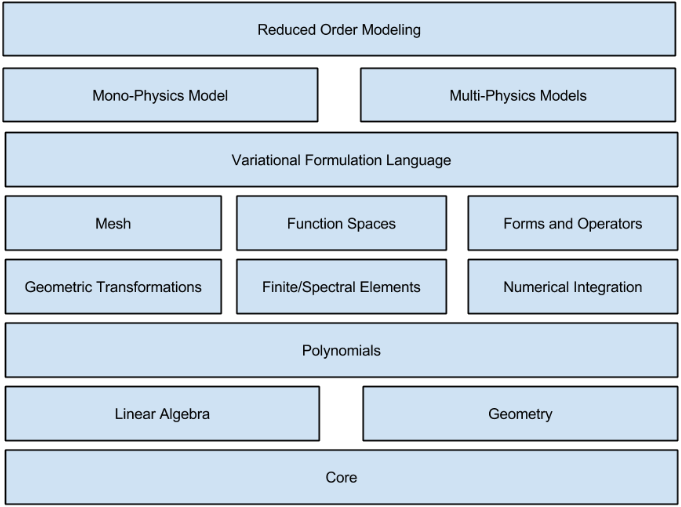
[1] http://vivabrain.fr
[2] http://www.feelpp.org

Some words on FEEL++

Finite Element Embedded Library in C++

- A Domain Specific Language for PDEs embedded in C++ providing a syntax very close to the mathematical language.
- Supports generalized arbitrary order Galerkin methods (cG, dG) in 1D, 2D and 3D
- Supports simplex, hypercube, high order meshes and geometries
- Supports seamless parallel computing and seamless interpolation operator
- Supports seamless interpolation between grids/function spaces
- Supports symbolic calculus thanks to `GiNaC`
- Supports large scale parallel linear and non-linear solvers `(PETSc/SLEPc)`
- Supports in-situ visualization with `Paraview`

Architecture. Generic Programming (Boost) and C++11-14 Standards



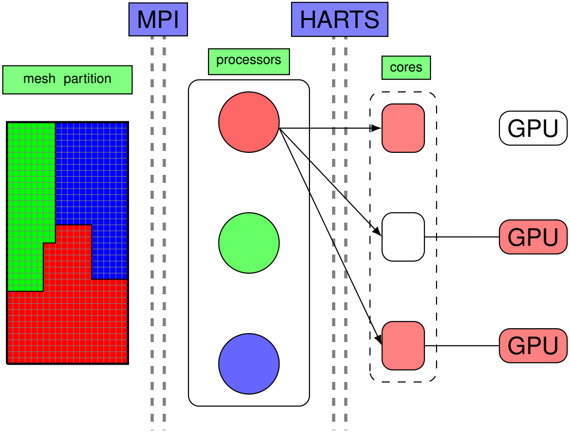
HPC FEEL++ Strategy. Seamless parallelization

Hybrid architectures

- many nodes, many cores, hybrid nodes
- MPI, multi-threads, GPU

MPI implementation :

- mesh partitioning
- dof table partitioning
- PETSc interface



- The parallelism is completely transparent (implicit use)
- Parallelism can be also made explicit (control communications)

A simple Example

$$\begin{cases} \text{Find } u \text{ s.t.} \\ -\Delta u = f \ \text{ in } \Omega \\ \quad u = 0 \ \text{ on } \partial\Omega \end{cases}$$

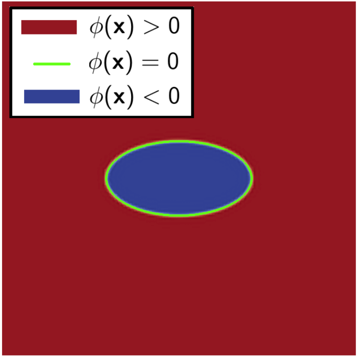
$$\Updownarrow$$

$$\begin{cases} \text{Find } u \in V \text{ s.t. } \forall v \in V \\ \int_\Omega \nabla u \cdot \nabla v = \int_\Omega fv \end{cases}$$

```
#include <feel/feel.hpp>
int main(int argc, char** argv)
{
  using namespace Feel;
  Environment env(_argc=argc,_argv=argv,
                  _desc=feel_options());
  auto mesh = loadMesh(_mesh=new Mesh<Simplex<3> >);
  auto Vh = Pch<2>( mesh );
  auto u = Vh->element();
  auto v = Vh->element();
  auto f = cst(1.0);

  // Linear Form : ℓ(v) = ∫_Ω fv
  auto l = form1(_test=Vh);
  l = integrate(_range=elements(mesh),
                _expr=f*id(v));

  // Bilinear Form : a(u,v) = ∫_Ω ∇u · ∇v
  auto a = form2(_trial=Vh,_test=Vh);
  a = integrate(_range=elements(mesh),
                _expr=inner(gradt(u),grad(v)) );

  // Boundary Conditions : u = 0 on ∂Ω
  a+=on(_range=boundaryfaces(mesh),
        _rhs=l,_element=u,
        _expr=cst(0.) );

  // Linear System Resolution : AU = F
  a.solve(_rhs=l,_solution=u);
  auto e = exporter( _mesh=mesh );
  e->add( "u", u ); e->save();
}
```

Level set method

[V. Doyeux PhD Thesis]



$\phi(\boldsymbol{x})$ used to track an interface

$$\phi(\boldsymbol{x}) = \begin{cases} dist(\boldsymbol{x}, \Gamma) & \boldsymbol{x} \in \Omega_1, \\ 0 & \boldsymbol{x} \in \Gamma, \\ -dist(\boldsymbol{x}, \Gamma) & \boldsymbol{x} \in \Omega_2, \end{cases}$$

➠ Advection by a divergence-free velocity $\boldsymbol{u}$

$$\frac{D\phi}{Dt} = 0 \qquad \frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0$$

Coupling with Navier Stokes equations

$$
\rho_\phi \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) - \nabla \cdot (2\mu_\phi D(\boldsymbol{u})) + \nabla p = \boldsymbol{F}_\phi
$$
$$
\nabla \cdot \boldsymbol{u} = 0
$$
$$
\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0
$$

- Level set advected by solution of Navier Stokes equations
- Fluid quantities depend on level set function ρ_ϕ, μ_ϕ, $\boldsymbol{F}_\phi$

$$
\begin{aligned}
\rho_\phi &= \rho^- + (\rho^+ - \rho^-) H_\epsilon(\phi) \\
\mu_\phi &= \mu^- + (\mu^+ - \rho^-) H_\epsilon(\phi)
\end{aligned}
$$
$$
\boldsymbol{F}_\phi = \int_\Gamma \boldsymbol{F}_s = \int_\Omega \boldsymbol{F}_s \delta_\epsilon(\phi)
$$

Variational formulation

find $(\boldsymbol{u}, p, \phi) \in H^1(\Omega)^2 \times L^2(\Omega) \times H^1(\Omega)$ which verify $\forall(\boldsymbol{v}, q, \psi) \in H_0^1(\Omega)^2 \times L_0^2(\Omega) \times H^1(\Omega)$:

$$
\begin{aligned}
\rho_\phi \int_\Omega \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right) \cdot \boldsymbol{v} + \mu_\phi \int_{\Omega_f} D(\boldsymbol{u}) : D(\boldsymbol{v}) &- \int_\Omega p \nabla \cdot \boldsymbol{v} \\
&= \int_\Omega \boldsymbol{F}_\phi \cdot \boldsymbol{v}, \\
\int_{\Omega_f} q \nabla \cdot \boldsymbol{u} &= 0, \\
\int_\Omega \partial_t \phi \psi + \int_\Omega (\boldsymbol{u} \cdot \nabla \phi) \psi + \int_{\{\Omega \text{ or } \Omega_i^f\}} S(\phi, \psi) &= 0.
\end{aligned}
$$

with $S(\phi, \psi)$ a stabilization term (SUPG, GLS, SGS, CIP).

➠ Solved by FEM and using FEEL++ library

Rising of a bubble in a viscous fluids. Benchmark from : [Hysing et al. (2009)]

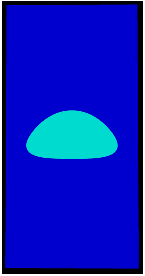
External forces added to Navier Stokes equation

- Gravity : $\boldsymbol{F}_g = \rho_\phi \boldsymbol{g}$
- Surface tension : $\boldsymbol{F}_{st} = \int_\Gamma \sigma\kappa\boldsymbol{n}$

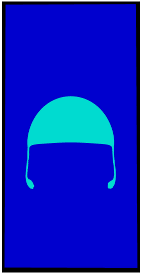
Two different cases tested

[JCAM 246, pp. 251-259 (2013)]

High surface tension ($\sigma = 24.5$)



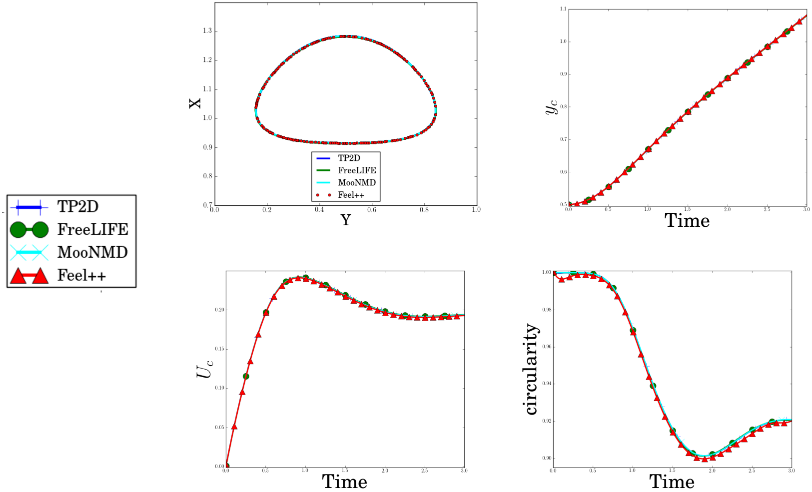
Low surface tension ($\sigma = 1.96$)



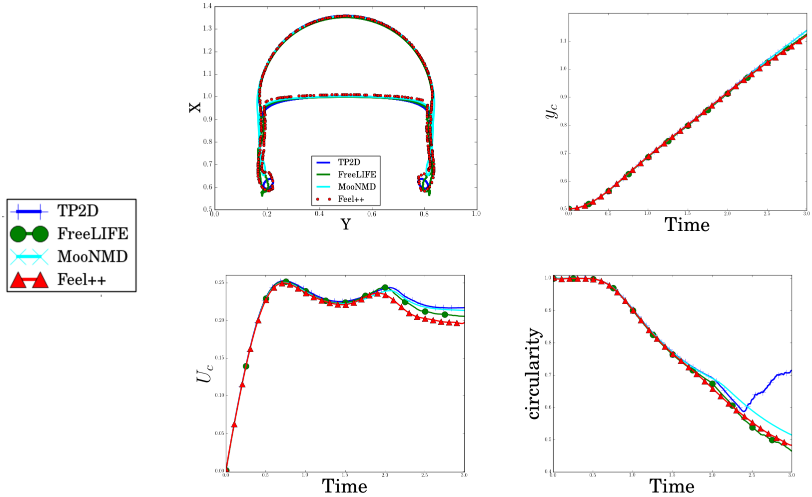
Groups having done the benchmark

Group and affiliation	Code/Method
TU Dortmount, Inst. of Applied Math. *S. Turek, D. Kuzmin, S. Hysing*	**TP2D** FEM-Level Set $\mathbb{Q}_1$-$\mathbb{Q}_0$ $\mathbb{Q}_1$
EPFL Lausanne, Inst. of Analysis ans Sci. Comp. *E.Burman, N.Parolini*	**FreeLIFE** FEM-Level Set $\mathbb{P}_2$-$\mathbb{P}_1$ $\mathbb{P}_2$
Uni. Magdeburg, inst. of Analysis and Num. Math. *L. Tobiska, S. Ganesan*	**MooNMD** FEM-ALE
Univ. Joseph Fourier, LIPhy. *V.Doyeux, Y.Guyot, V.Chabannes* *C.Prud'homme, M.Ismail*	**Feel++** FEM-Level Set $\mathbb{P}_2$-$\mathbb{P}_1$ $\mathbb{P}_1$

High surface tension : ellipsoidal bubble



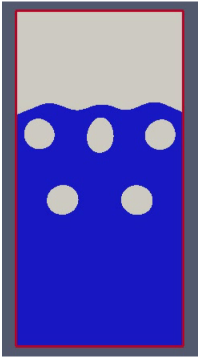
Low surface tension : squirted bubble



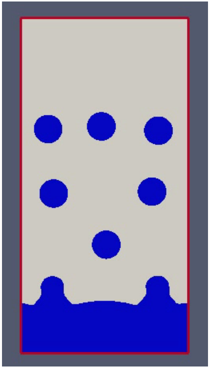
Handling many interfaces

Implicit handeling of many interfaces and topology changes

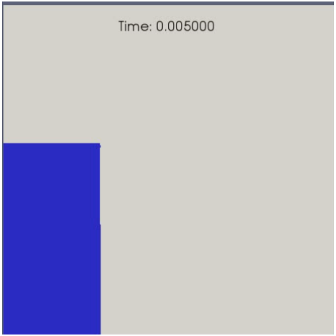
Many rising bubbles



Many falling drops

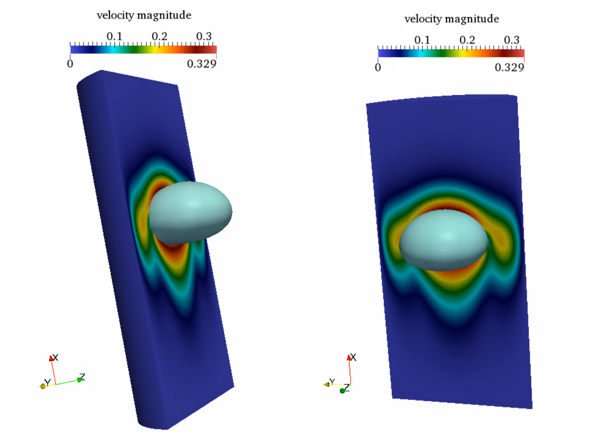


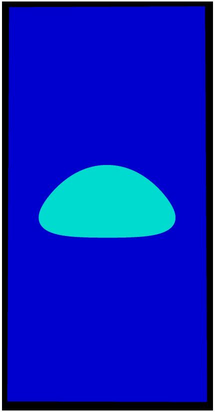
Dam Break



From 2D to 3D

[S. Priem PhD Thesis]





The case of vesicles [V. Doyeux PhD thesis]

[JCAM 246, pp. 251-259 (2013)]

- Curvature force derived from Helfrich energy

$$E_h = \int_\Gamma \frac{k_B}{2}\kappa^2$$

$$\boldsymbol{F_b} = \int_\Gamma \frac{k_B}{2}\left[\frac{\kappa^3}{2} + \boldsymbol{t}\cdot\nabla(\boldsymbol{t}\cdot\nabla\kappa)\right]\boldsymbol{n}$$

- Membrane inextensibility

Add a Lagrange multiplier constraint

$$\nabla_s \cdot \boldsymbol{u} = \nabla\cdot\boldsymbol{u} - (\nabla\boldsymbol{u}\cdot\boldsymbol{n})\cdot\boldsymbol{n} = 0.$$ [Laadhari PhD thesis (2011)]

Variational formulation on Stokes equation

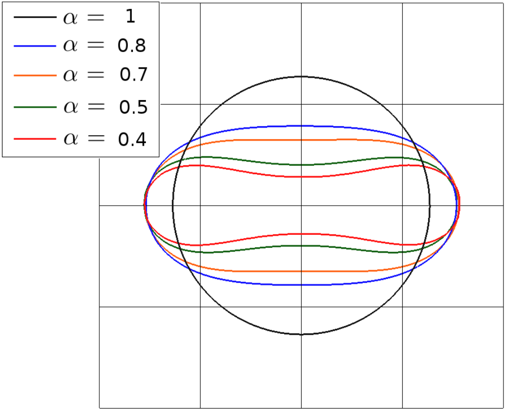
$$
\begin{aligned}
-\nabla \cdot (2\mu_\phi D(\boldsymbol{u})) + \nabla p &= \boldsymbol{F}_\phi \text{ in } \Omega \\
\nabla \cdot \boldsymbol{u} &= 0 \text{ in } \Omega \\
\nabla_s \cdot \boldsymbol{u} &= 0 \text{ on } \Gamma
\end{aligned}
$$

find $(\boldsymbol{u}, p, \lambda) \in H^1(\Omega)^2 \times L^2(\Omega) \times H^{\frac{1}{2}}(\Gamma)$ which verify
$\forall (\boldsymbol{v}, q, \nu) \in H^1_0(\Omega)^2 \times L^2_0(\Omega) \times H^{\frac{1}{2}}(\Gamma)$:

$$
\begin{aligned}
\mu_\phi \int_{\Omega_f} D(\boldsymbol{u}) : D(\boldsymbol{v}) - \int_\Omega p \nabla \cdot \boldsymbol{v} + \int_\Gamma \lambda \nabla_s \cdot \boldsymbol{v} &= \int_\Omega \boldsymbol{F}_\phi \cdot \boldsymbol{v} \\
\int_\Omega q \nabla \cdot \boldsymbol{u} &= 0 \\
\int_\Gamma \nu \nabla_s \cdot \boldsymbol{u} &= 0
\end{aligned}
$$

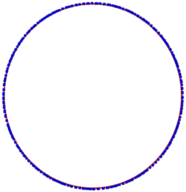
Equilibrium shapes

$$\text{reduce area } \alpha = \frac{\text{area vesicle}}{\text{area circle with same perimeter}}$$

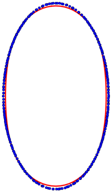


Comparison with another model

Comparison with boundary integral method from [Kaoui et al. 2011]



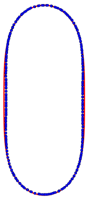
– Reduce area = 1
•• Reduce area = 1



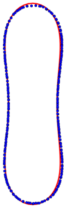
– Reduce area = 0.90
•• Reduce area = 0.93

Comparison with another model

Comparison with boundary integral method from [Kaoui et al. 2011]

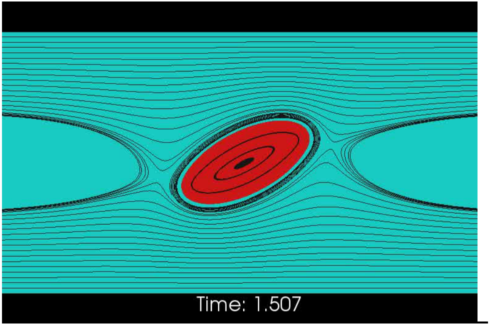


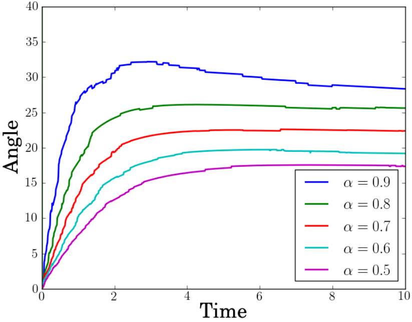
– Reduce area = 0.80
·· Reduce area = 0.78



– Reduce area = 0.60
·· Reduce area = 0.61

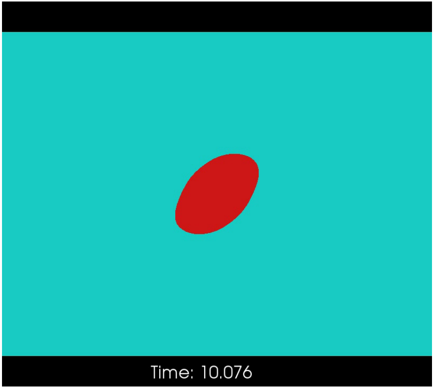
Tank treading motion

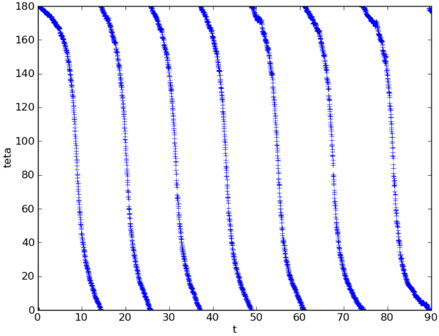




Angle vs. time for different α

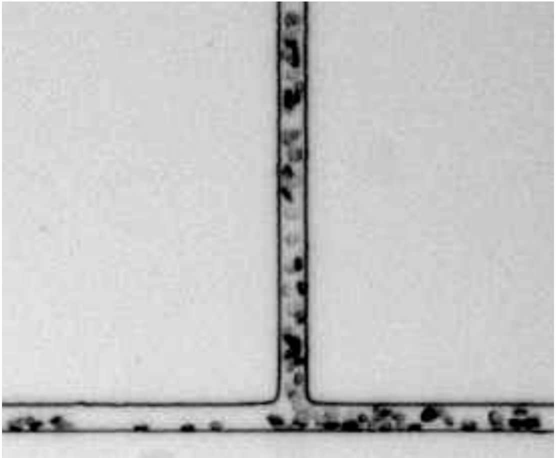
Tumbling motion



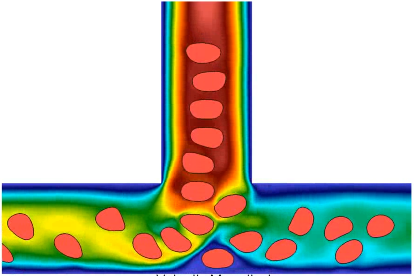


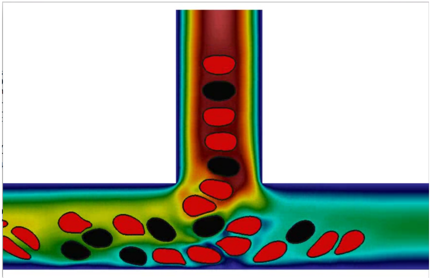
Angle of the vesicle during time

Suspensions of RBCs in Bifurcations [Courtesy of T. Podgorski]



Homogeneous and Heterogeneous Suspensions of Vesicles in a Bifurcation [V. Doyeux PhD Thesis]



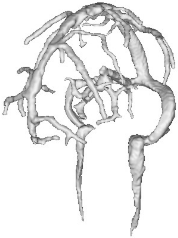


[JFM, 674, pp. 359-388 (2011)] (For the rigid particles case)

Outline

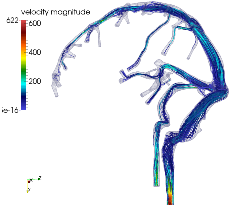
ANR VivaBrain

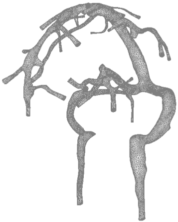
http://vivabrain.fr



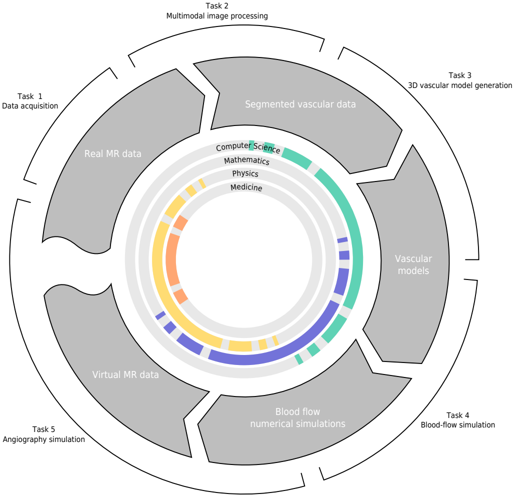
Virtual MRA

Goal : Closing the Loop!





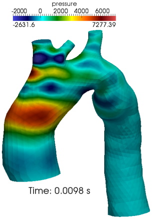
VivaBrain Tasks



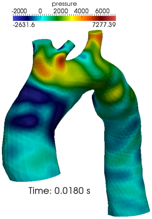
FSI applications and realistic geometries [Chabannes PhD]

Aorta

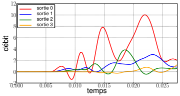
(a) t=0.0032 s



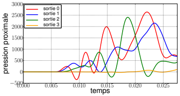
(b) t=0.0098 s



(c) t=0.0180 s

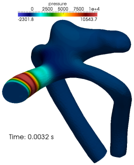


(d) Flow rate

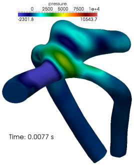


(e) Proximal pressure

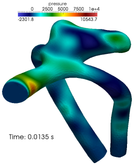
Cerebral Aneurysm



(f) t=0.0032 s

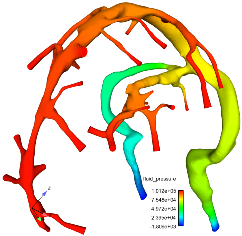


(g) t=0.0077 s

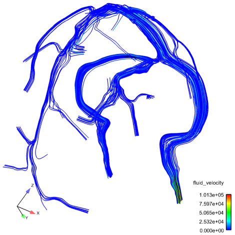


(h) t=0.0135 s

Flow simulation in cerebrovenous network [JCSMD, 3, pp.23-37 (2015)]



(a) Pressure



(b) Streamlines

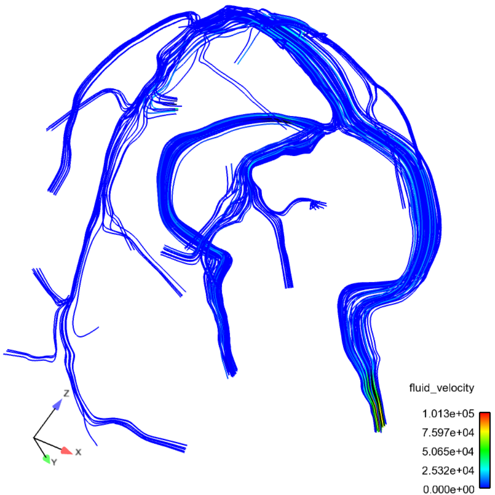
Stokes

- 29 inlets and 2 outlets.
- P2P1 Approximation
- 10 millions DOF
- Prace/SuperMuC Computers

nProc	Assembly (t_1)	Resolution (t_2)	total ($t_1 + t_2$)	nIterations
512	2.15219	45.4506	47.6028	295
1024	1.33274	17.9034	19.2361	377
2048	1.06893	14.6693	15.7382	379

[ESAIM: Proc (2013)]

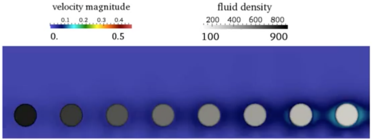
Flow simulation in cerebrovenous network [JCSMD, 3, pp.23-37 (2015)]

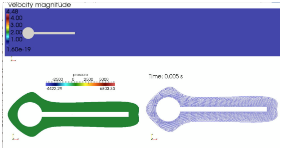


Outline

Conclusions and ongoing work

- ✔ Unified framework for two-fluid flows (2D and 3D using FEEL++). Application to bubbles, drops and vesicles
- ✔ Validation using benchmarks and physical experiments
- Still some works to handle real applications (e.g. Virtual MRI)
- Working on new scheme to model RBCs **(PhD starting this year)**. Collab. with E. Maitre (LJK Grenoble)
- Couple macroscopic and microscopic models **(PostDoc starting this year)**. Coll. Prud'homme (Strasbourg)





http://www.feelpp.org

Some success Stories

- Consortium (Strasbourg University, UJF, University of Coimbra, CNRS and CNR)
- About 20 developers. Project leader: C. Prud'homme.
- Start-up in progress (But the kernel of the library remains open-source)
- Mesochallenge equip@meso 2013 prize for the project "HPC for Blood Flow Simulations in Complex Geometries"
- An article in HPCMagazine was published about this project
- 60 millions of core-hours computing (PRACE European call 2012/2013)
- 30 millions of core-hours computing (PRACE European call 2013/2014)
- 10 millions of core-hours computing (PRACE European call 2014/2015)
- Best students' posters in various workshops

Feel++: Communication Tools

- Feel++ website:
  http://www.feelpp.org
- Feel++ Git Repo:
  https://github.com/feelpp/feelpp.git
- Feel++ @ Google+:
  https://plus.google.com/u/0/communities/104696212880173187475
- Feel++ Google Groups:
  http://groups.feelpp.org
- Feel++ Publications:
  http://www.feelpp.org/publications
- Feel++ @ YouTube Channel:
  http://www.youtube.com/channel/UCnLX6kyV8j644isqhMpUN4Q
- Feel++ @ Twitter (commits, youtube videos):
  https://twitter.com/feelpp