FEM automation of non-Newtonian fluids

Andy R Terrel

Texas Advanced Computing Center
Univesity of Texas at Austin

European Seminar on Coupled Problems
June 28 – July 2, 2010

Collaborators

- L. Ridgway Scott, Computer Science, University of Chicago
- Matthew G. Knepley, Computation Institute, University of Chicago
- Garth N. Wells, Mechanical Engineering, Cambridge University
- Robert C. Kirby, Mathematics, Texas Tech University

Software

- The FEniCS Project
- PETSc

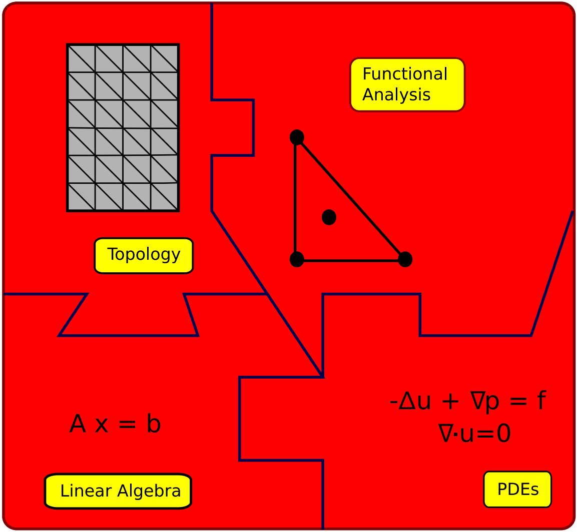
Outline

1. Fluid model
2. FEM Automation
3. Automated Solvers
4. Polynomial Solvers

The Rheology Drugstore

- Rivlin Ericksen Order Fluids
- Oldroyd-B, Maxwell, PTT, Giesekus, Grmela
- Jeffreys
- Bingham
- Burger

Mathematics Puzzle



High Level Goals of Research

- Automate writing non Newtonian fluid simulations.
- Test stability of automated simulations.
- Automatically rewritten to improve robustness.

Outline

Basic equations

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\nabla \cdot \boldsymbol{T} = \boldsymbol{f} \tag{2}$$

$$\boldsymbol{T} \equiv Ip + 2\eta \boldsymbol{D} + \boldsymbol{\tau}, \tag{3}$$

$$\boldsymbol{D} \equiv \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{T}) \tag{4}$$

Oldroyd-B type models

- Models polymers as Maxwell solids.
- Characterized by a relaxation time λ.
- $\lambda \to \infty$ corresponds to Hookean elastic solid
- $\lambda \to 0$ corresponds to Newtonian fluid ($\boldsymbol{\tau} = 0$, $\eta_p = 0$).

$$\lambda \overset{\nabla}{\boldsymbol{\tau}} + \boldsymbol{\tau} - 2\eta_p \boldsymbol{D} + \boldsymbol{g}(\boldsymbol{\tau}) = \boldsymbol{0}, \tag{5}$$

$$\overset{\nabla}{\boldsymbol{\tau}} \equiv \frac{\partial \boldsymbol{\tau}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{\tau} - (\nabla \boldsymbol{u})^T \cdot \boldsymbol{\tau} - \boldsymbol{\tau} \cdot \nabla \boldsymbol{u}, \tag{6}$$

Describes Oldroyd-B, UC Maxwell, Phan-Than Tanner, Giesekus models.

Addressing hyperbolicity

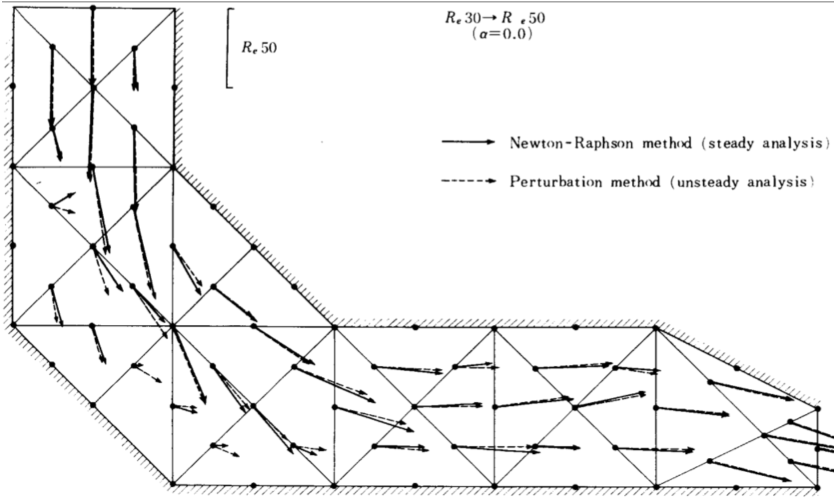


Fig. 4. Comparison of computed steady velocity between by steady analysis and by unsteady analysis.

[Kawahara Takeuchi 1977]

Addressing hyperbolicity

Option 1: Change the discretization

- Use Hermite elements [MarchalCrochet1986]
- Use Discontinuous Galerkin methods [FortinFortin1989]

Option 2: Stabilize the model

- Use a streamline/symmetric/regularized Galerkin approach. [GiraultScott2002, Amara et al 2005]
- Use SUPG on both velocity and stress. [BrookesHughes1982, MarchalCrochet1987]
- Streamline only the convected stress (SU). [MarchalCrochet1987]

Addressing hyperbolicity

Option 1: Change the discretization

- Use Hermite elements [MarchalCrochet1986]
- Use Discontinuous Galerkin methods [FortinFortin1989]

Option 2: Stabilize the model

- Use a streamline/symmetric/regularized Galerkin approach. [GiraultScott2002, Amara et al 2005]
- Use SUPG on both velocity and stress. [BrookesHughes1982, MarchalCrochet1987]
- Streamline only the convected stress (SU). [MarchalCrochet1987]

Preserving incompressibility

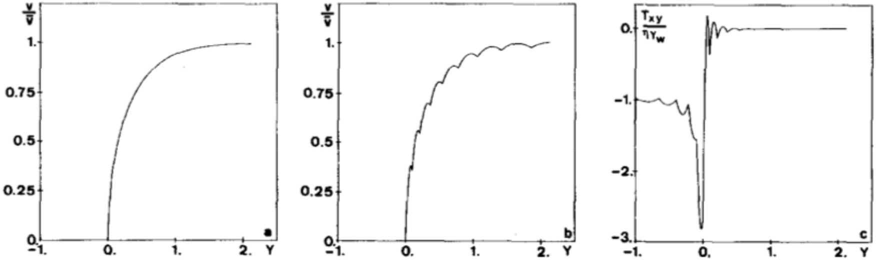


Fig. 2. Newtonian solution of the stick-slip problem along the $x = 1$ line. (a) velocity, velocity-pressure formulation. (b), (c) velocity and shear-stress, MIX1 formulation.

Preserving incompressibility

Early versions of coupling of stress with pressure and velocity violated incompressiblity condition.

Option 1: Change the discretization

- Stress subelement of velocity and pressure [MarchalCrochet1987]

Option 2: Change the model

$$\Sigma = \tau - 2\eta \boldsymbol{D} \quad (7)$$

- Elastic and viscous stress splitting [Ranjagopal et al 1990], consider $\boldsymbol{D}$ as separate unknown.
- Discrete elastic and viscous stress splitting [GuenetteFortin1995], use projection of $\boldsymbol{D}$

Preserving incompressibility

Early versions of coupling of stress with pressure and velocity violated incompressiblity condition.

Option 1: Change the discretization

- Stress subelement of velocity and pressure [MarchalCrochet1987]

Option 2: Change the model

$$\boldsymbol{\Sigma} = \boldsymbol{\tau} - 2\eta \boldsymbol{D} \tag{7}$$

- Elastic and viscous stress splitting [Ranjagopal et al 1990], consider $\boldsymbol{D}$ as separate unknown.
- Discrete elastic and viscous stress splitting [GuenetteFortin1995], use projection of $\boldsymbol{D}$

Numerics all together now

Each technique requires certain amount of flexibility in both rewriting the governing equations (M) and/or assembling the spatial discretization (S)

	Macro elements	EVSS	DEVSS
DG	S	M+S	M+S
SUPG	M+S	M	M
SU	M+S	M	M

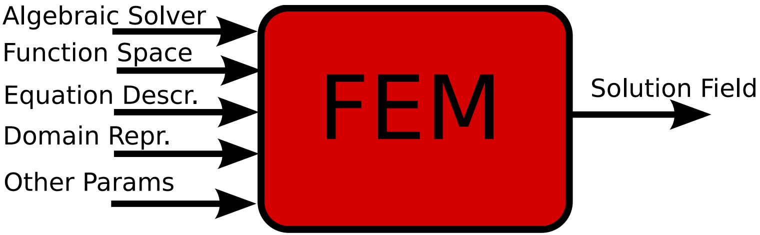
Outline

1. Fluid model
2. FEM Automation
3. Automated Solvers
4. Polynomial Solvers

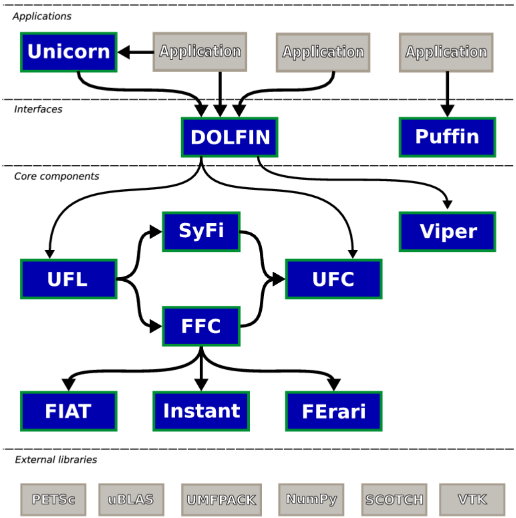
The FEniCS Project

- Started in 2003 as a collaboration between
  - Chalmers
  - University of Chicago
- Now spans
  - KTH
  - University of Oslo and Simula Research
  - University of Chicago
  - Cambridge University
  - TU Delft
- Focused on Automated Mathematical Modelling
- Allows researchers to easily and rapidly develop simulations

The FEniCS Project



The FEniCS Project



Equation input

```
from ufl import *
# Element Definitions
stress = TensorElement(dfamily, cell, order-1)
velocity = VectorElement(family, cell, order)
pressure = FiniteElement(family, cell, order-1)
mixed = MixedElement([velocity,pressure,stress])

# Test and Trial function definitions
mTest = TestFunction(mixed)
v, q, phi = split(mTest)
mTrial = TrialFunction(mixed)
u, p, sigma = split(mTrial)
```

Equation input

```
# Conservation equations (Stokes-like system)
a_stokes = (2*eta_e*inner(D(v), D(u)) - inner(p, div(v))
           + inner(q, div(u)) + inner(D(v), sigma))*dx
L_stokes = (inner(v,f))*dx

# Constitutive equations: Oldroyd-B
sigma_uc = dot(uF,grad(sF)) - dot(grad(uF), sF)
           - dot(sF, transpose(grad(uF)))
a_con = (inner(sF, lam*sigma_uc)
       + inner(sF, 2*eta*D(uF) - 2*eta * D(uF)))*dx
L_con = (2*eta_e*inner(sF, D_proj))*dx

# Full bilinear form and residual
f = a_con + a_stokes - L_con - L_stokes
F = derivative(f, mF, mTest)
J = derivative(F, mF, mTrial)
```

Outline

FEniCS as library generator

- Automation is not enough, simulation still requires expert knowledge.
- Libraries give simple interface for this expertise.
- FEniCS-Apps examples
  - Ascot – automated stability condition testing
  - CBC.Solve – biomedical solvers
  - DiffSim – coupled stochastic and deterministic problems
  - Rheagen –non-Newtonian fluid problems
  - DOLFWAVE – surface water waves problems
  - FEniCS Plasticity – standard plasticity
  - TriTetMesh – high quality DOLFIN meshes
  - Unicorn – unified continuum mechanics solver

Rheagen example

4-1 planar flow example.
First begin with a simple description of the fluid.

Rheagen example

```
Mesh mesh("../data/planarcontraction.xml.gz");
Inlet inlet; Outlet outlet;
TopWall top_wall; SymmetryLine sym_line;

Inflow in(mesh); Outflow out(mesh);
NoSlipBC ns_bc(mesh); Constant sym_bc(mesh, 0.0);

Array< Function* > vel_bc_funcs(&ns_bc, &sym_bc, &in, &out);
Array< int > vel_comps(-1, 1, -1, -1);

Fluid fluid(mesh, vel_subdomains, vel_bc_funcs, vel_comps);
```

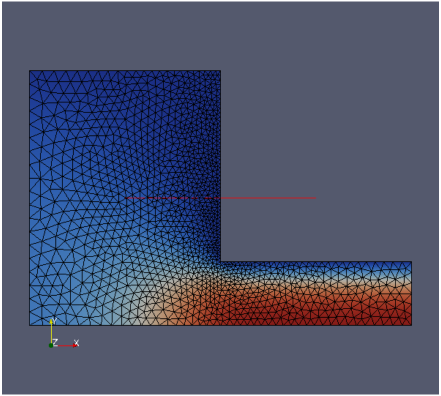
Rheagen example

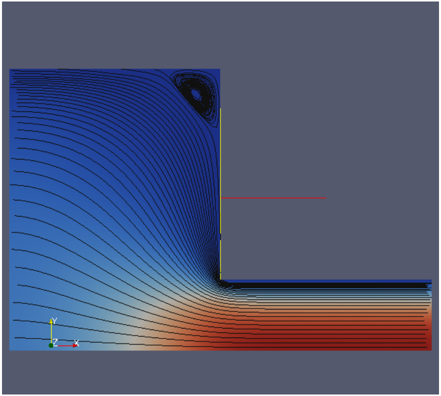
Then pass to generated library.

Rheagen example

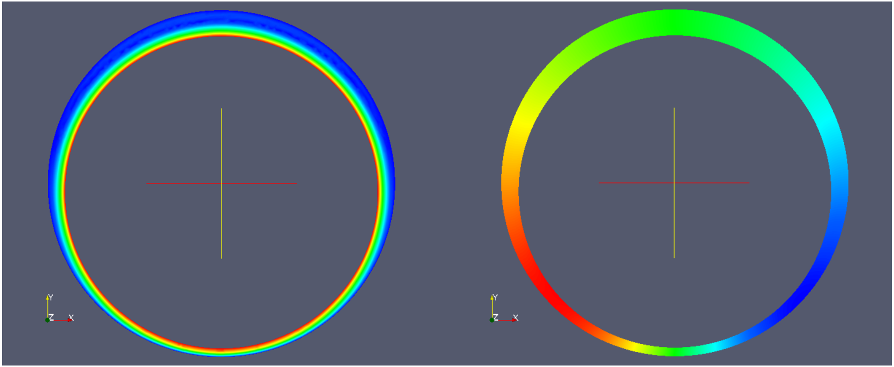
```
Grade2Solver grade2;
grade2.solve(fluid, zero);

StokesSolver stokes;
stokes.solve(fluid, zero);

OldroydBSolver oldroydb;
oldroydb.set("lam", 10);
NewtonSolver& ns = oldroydb.newton_solver();
ns.set("Newton maximum iterations", 20);
oldroydb.solve(fluid, zero);
```

Rheagen example

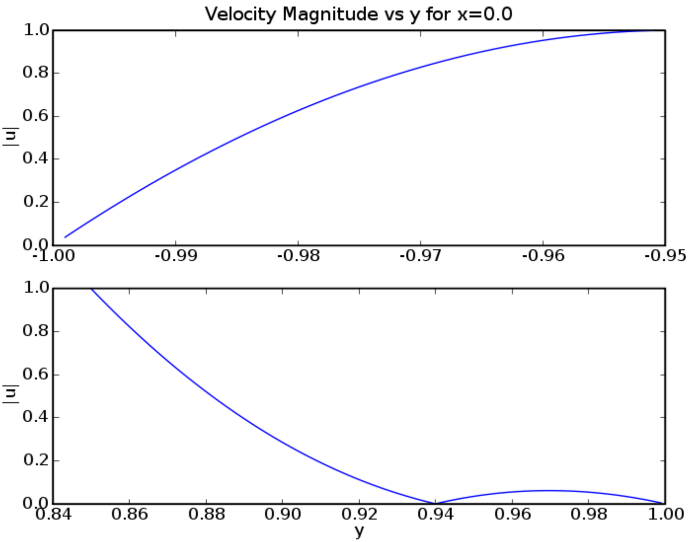




Journal Bearing Results



Journal Bearing Results



Function Space Discretization

Common Function Spaces

Element	Pros	Cons
Enriched $(P_1+B_3) \times P_1$	Cheap	div(u)!=0, very poor error
Stabilized $P_1 \times P_1$	Cheap	div(u)!=0, poor error
Taylor-Hood $P_2 \times P_1$	Cheap	div(u)!=0
Crouziex-Raviert	div(u)=0	low order, poor matrix conditioning
Scott-Vogelius	div(u)=0	high order only

Element wise operations

	MIX	MIX/SUPG	MIX/SU
Oldroyd-B	29543	90870	67230
UCM	29495	90794	67154
PTT	53750	91603	60197

	DEVSS	DEVSS/SUPG	DEVSS/SU
Oldroyd-B	29553	90880	58904
UCM	29505	90804	58828
PTT	53750	91603	60197

Results for lid driven cavity

Table: UCM method data from lid driven cavity

discretization	stabilization	eta	lam	Newton iterations
MIX	SU	100.0	1.0	18
MIX	SUPG	100.0	1.0	X
MIX	None	100.0	1.0	18
DEVSS	SU	100.0	1.0	20
DEVSS	SUPG	100.0	1.0	X
DEVSS	None	100.0	1.0	20

Results for lid driven cavity

Table: Oldroyd-B method data from lid driven cavity

discretization	stabilization	eta	eta_e	lam	Newton iterations
MIX	SU	100.0	0.1	1.0	21
MIX	SUPG	100.0	0.1	1.0	22
MIX	None	100.0	0.1	1.0	X
DEVSS	SU	100.0	0.1	1.0	X
DEVSS	SUPG	100.0	0.1	1.0	X
DEVSS	None	100.0	0.1	1.0	X

Results for lid driven cavity

Table: PTT method data from lid driven cavity

discretization	stabilization	eta	eta_e	lam	Newton iterations
MIX	SU	1.0	0.0	1.0	18
MIX	SUPG	1.0	0.0	1.0	X
MIX	None	1.0	0.0	1.0	18
DEVSS	SU	1.0	0.0	1.0	20
DEVSS	SUPG	1.0	0.0	1.0	X
DEVSS	None	1.0	0.0	1.0	20

Results for lid driven cavity

Table: PTT method data from lid driven cavity

discretization	stabilization	eta	eta_e	lam	Newton iterations
MIX	SU	100.0	0.1	0.1	9
MIX	SUPG	100.0	0.1	0.1	X
MIX	None	100.0	0.1	0.1	7
DEVSS	SU	100.0	0.1	0.1	X
DEVSS	SUPG	100.0	0.1	0.1	X
DEVSS	None	100.0	0.1	0.1	X

Outline

Polynomial Solvers

Homotopy Continuation

Form a homotopy from a trivial start system to the solution

$$\mathcal{H}(x,t) = (1-t)Q(x) + tP(x) \tag{8}$$

Polynomial Solvers

A great opportunity exists for polynomial solvers

- Bézout's theorem – track num equations * polynomial order
- Bernshtein theorem – track num to volume of mixed space

Testing code

Our prototype has

- FEM assembly routines for full non-linear tensor
- several small test problems
- connection to several polynomial solvers
  - Sage
  - PHCpack
  - Bertini

Testing code

Our test cases

- Few linear forms
- $u^2 - \Delta(u)$, testing for multiple solutions
- Navier-Stokes, Re 10, 500 dofs (small but exact)

Testing code

	dofs	mixed volume	solutions found	time
TH P2XP1 5X5	122	1	1	70ms
TH P2XP1 6X6	197	1	1	95ms
TH P2XP1 8X8	401	1	1	930ms

Table: PHC data for stokes problem

	dofs	mixed volume	solutions found	time
P1 4X4	4	16	16	60ms
P1 6X6	16	65536	65536	5318s

Table: PHC data for nonlinear Laplacian problem

Testing code

	dofs	mixed volume	solutions found	time
TH P2XP1 3X3	116	2^{24}	–	–

Table: PHC data for Navier-Stokes problem

Conclusion

- FEM Automation enables flexiblity in simulation software
- Mathematics $\Leftrightarrow$ Software Abstractions
- Difficultl non Newtonian Fluid simulations

Future Directions

- Full Approximation Schemes using Polynomial Solvers
- Automatic rewriting model equations with stability testing
- Formal derivation of assembly algorithms

References

- **FEniCS Documentation**:
  http://www.fenics.org/wiki/FEniCS_Project
  - Project documentation
  - Users manuals
  - Repositories, bug tracking
  - Image gallery
- **Publications**:
  http://www.fenics.org/wiki/Related_presentations_and_publications
  - Research and publications that make use of FEniCS
- **PETSc Documentation**:
  http://www.mcs.anl.gov/petsc/docs
  - PETSc Users manual
  - Manual pages
  - Many hyperlinked examples
  - FAQ, Troubleshooting info, installation info, etc.
  - Publication using PETSc