FEM automation of non-Newtonian fluids

Andy R Terrel

Texas Advanced Computing Center
University 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
The Rheology Drugstore

- Rivlin Ericksen
- Order Fluids
- Oldroyd-B,
- Maxwell, PTT,
- Giesekus, Grmela
- Jeffreys
- Bingham
- Burger
High Level Goals of Research

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

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

Basic equations

\[ \nabla \cdot \mathbf{u} = 0, \quad (1) \]

\[ \nabla \cdot \mathbf{T} = f \quad (2) \]

\[ \mathbf{T} \equiv l_p + 2\eta \mathbf{D} + \tau, \quad (3) \]

\[ \mathbf{D} \equiv \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (4) \]
Oldroyd-B type models

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

\[
\lambda \frac{\nabla}{\tau} + \tau - 2\eta_p D + g(\tau) = 0, \tag{5}
\]

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

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]
Fluid model

Addressing hyperbolicity

Option 1: Change the discretization
- Use Hermite elements [MarchalCrochet1986]
- Use Discontinuous Galerkin methods [FortinFortin1989]

Option 2: Stabilize the model
- 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 SUPG on both velocity and stress. [BrookesHughes1982, MarchalCrochet1987]
- Streamline only the convected stress (SU). [MarchalCrochet1987]
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 incompressibility condition.

**Option 1:** Change the discretization
- Stress subelement of velocity and pressure [MarchalCrochet1987]

**Option 2:** Change the model

\[
\Sigma = \tau - 2\eta D \tag{7}
\]

- Elastic and viscous stress splitting [Ranjagopal et al 1990], consider \( D \) as separate unknown.
- Discrete elastic and viscous stress splitting [GuenetteFortin1995], use projection of \( D \)
Preserving incompressibility

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

Option 1: Change the discretization

- Stress subelement of velocity and pressure [MarchalCrochet1987]

Option 2: Change the model

\[ \Sigma = \tau - 2\eta D \] (7)

- Elastic and viscous stress splitting [Ranjagopal et al 1990], consider \( D \) as separate unknown.
- Discrete elastic and viscous stress splitting [GuenetteFortin1995], use projection of \( D \)
Each technique requires certain amount of flexibility in both rewriting the governing equations (M) and/or assembling the spatial discretization (S).

<table>
<thead>
<tr>
<th></th>
<th>Macro elements</th>
<th>EVSS</th>
<th>DEVSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG</td>
<td>S</td>
<td>M+S</td>
<td>M+S</td>
</tr>
<tr>
<td>SUPG</td>
<td>M+S</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>SU</td>
<td>M+S</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>
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
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)
\[ \text{a}_{\text{stokes}} = (2\eta_E \text{inner}(D(v), D(u)) - \text{inner}(p, \text{div}(v)) \]
\[ + \text{inner}(q, \text{div}(u)) + \text{inner}(D(v), \sigma))*dx \]
\[ \text{L}_{\text{stokes}} = (\text{inner}(v,f))*dx \]

# Constitutive equations: Oldroyd-B
\[ \sigma_{uc} = \text{dot}(uF, \text{grad}(sF)) - \text{dot}(\text{grad}(uF), sF) \]
\[ - \text{dot}(sF, \text{transpose} \text{grad}(uF)) \]
\[ \text{a}_{\text{con}} = (\text{inner}(sF, \lambda \sigma_{uc}) \]
\[ + \text{inner}(sF, 2\eta_E \text{D}(uF) - 2\eta_E \text{D}(uF)))*dx \]
\[ \text{L}_{\text{con}} = (2\eta_E \text{inner}(sF, \text{D}_\text{proj}))*dx \]

# Full bilinear form and residual
\[ f = \text{a}_{\text{con}} + \text{a}_{\text{stokes}} - \text{L}_{\text{con}} - \text{L}_{\text{stokes}} \]
\[ \text{F} = \text{derivative}(f, mF, mTest) \]
\[ \text{J} = \text{derivative}(F, mF, mTrial) \]
Outline

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

Velocity Magnitude vs $y$ for $x=0.0$

Graph showing velocity magnitude $|u|$ vs $y$ for $x=0.0$.
## Common Function Spaces

<table>
<thead>
<tr>
<th>Element</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enriched $(P_1 + B_3) \times P_1$</td>
<td>Cheap</td>
<td>$\text{div}(u) \neq 0$, very poor error</td>
</tr>
<tr>
<td>Stabilized $P_1 \times P_1$</td>
<td>Cheap</td>
<td>$\text{div}(u) \neq 0$, poor error</td>
</tr>
<tr>
<td>Taylor-Hood $P_2 \times P_1$</td>
<td>Cheap</td>
<td>$\text{div}(u) \neq 0$</td>
</tr>
<tr>
<td>Crouziex-Raviert</td>
<td>div$(u) = 0$</td>
<td>low order, poor matrix conditioning</td>
</tr>
<tr>
<td>Scott-Vogelius</td>
<td>div$(u) = 0$</td>
<td>high order only</td>
</tr>
</tbody>
</table>
### Element wise operations

<table>
<thead>
<tr>
<th></th>
<th>MIX</th>
<th>MIX/SUPG</th>
<th>MIX/SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oldroyd-B</td>
<td>29543</td>
<td>90870</td>
<td>67230</td>
</tr>
<tr>
<td>UCM</td>
<td>29495</td>
<td>90794</td>
<td>67154</td>
</tr>
<tr>
<td>PTT</td>
<td>53750</td>
<td>91603</td>
<td>60197</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DEVSS</th>
<th>DEVSS/SUPG</th>
<th>DEVSS/SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oldroyd-B</td>
<td>29553</td>
<td>90880</td>
<td>58904</td>
</tr>
<tr>
<td>UCM</td>
<td>29505</td>
<td>90804</td>
<td>58828</td>
</tr>
<tr>
<td>PTT</td>
<td>53750</td>
<td>91603</td>
<td>60197</td>
</tr>
</tbody>
</table>
### Results for lid driven cavity

**Table:** UCM method data from lid driven cavity

<table>
<thead>
<tr>
<th>discretization</th>
<th>stabilization</th>
<th>eta</th>
<th>lam</th>
<th>Newton iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIX</td>
<td>SU</td>
<td>100.0</td>
<td>1.0</td>
<td>18</td>
</tr>
<tr>
<td>MIX</td>
<td>SUPG</td>
<td>100.0</td>
<td>1.0</td>
<td>X</td>
</tr>
<tr>
<td>MIX</td>
<td>None</td>
<td>100.0</td>
<td>1.0</td>
<td>18</td>
</tr>
<tr>
<td>DEVSS</td>
<td>SU</td>
<td>100.0</td>
<td>1.0</td>
<td>20</td>
</tr>
<tr>
<td>DEVSS</td>
<td>SUPG</td>
<td>100.0</td>
<td>1.0</td>
<td>X</td>
</tr>
<tr>
<td>DEVSS</td>
<td>None</td>
<td>100.0</td>
<td>1.0</td>
<td>20</td>
</tr>
</tbody>
</table>
## Results for lid driven cavity

**Table:** Oldroyd-B method data from lid driven cavity

<table>
<thead>
<tr>
<th>discretization</th>
<th>stabilization</th>
<th>eta</th>
<th>eta_e</th>
<th>lam</th>
<th>Newton iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIX</td>
<td>SU</td>
<td>100.0</td>
<td>0.1</td>
<td>1.0</td>
<td>21</td>
</tr>
<tr>
<td>MIX</td>
<td>SUPG</td>
<td>100.0</td>
<td>0.1</td>
<td>1.0</td>
<td>22</td>
</tr>
<tr>
<td>MIX</td>
<td>None</td>
<td>100.0</td>
<td>0.1</td>
<td>1.0</td>
<td>X</td>
</tr>
<tr>
<td>DEVSS</td>
<td>SU</td>
<td>100.0</td>
<td>0.1</td>
<td>1.0</td>
<td>X</td>
</tr>
<tr>
<td>DEVSS</td>
<td>SUPG</td>
<td>100.0</td>
<td>0.1</td>
<td>1.0</td>
<td>X</td>
</tr>
<tr>
<td>DEVSS</td>
<td>None</td>
<td>100.0</td>
<td>0.1</td>
<td>1.0</td>
<td>X</td>
</tr>
</tbody>
</table>
### Results for lid driven cavity

<table>
<thead>
<tr>
<th>discretization</th>
<th>stabilization</th>
<th>eta</th>
<th>eta_e</th>
<th>lam</th>
<th>Newton iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIX</td>
<td>SU</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>18</td>
</tr>
<tr>
<td>MIX</td>
<td>SUPG</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>X</td>
</tr>
<tr>
<td>MIX</td>
<td>None</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>18</td>
</tr>
<tr>
<td>DEVSS</td>
<td>SU</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>20</td>
</tr>
<tr>
<td>DEVSS</td>
<td>SUPG</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>X</td>
</tr>
<tr>
<td>DEVSS</td>
<td>None</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>20</td>
</tr>
</tbody>
</table>

**Table:** PTT method data from lid driven cavity
### Table: PTT method data from lid driven cavity

<table>
<thead>
<tr>
<th>discretization</th>
<th>stabilization</th>
<th>eta</th>
<th>eta_e</th>
<th>lam</th>
<th>Newton iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIX</td>
<td>SU</td>
<td>100.0</td>
<td>0.1</td>
<td>0.1</td>
<td>9</td>
</tr>
<tr>
<td>MIX</td>
<td>SUPG</td>
<td>100.0</td>
<td>0.1</td>
<td>0.1</td>
<td>X</td>
</tr>
<tr>
<td>MIX</td>
<td>None</td>
<td>100.0</td>
<td>0.1</td>
<td>0.1</td>
<td>7</td>
</tr>
<tr>
<td>DEVSS</td>
<td>SU</td>
<td>100.0</td>
<td>0.1</td>
<td>0.1</td>
<td>X</td>
</tr>
<tr>
<td>DEVSS</td>
<td>SUPG</td>
<td>100.0</td>
<td>0.1</td>
<td>0.1</td>
<td>X</td>
</tr>
<tr>
<td>DEVSS</td>
<td>None</td>
<td>100.0</td>
<td>0.1</td>
<td>0.1</td>
<td>X</td>
</tr>
</tbody>
</table>
Outline

1. Fluid model
2. FEM Automation
3. Automated Solvers
4. Polynomial Solvers
Homotopy Continuation
Form a homotopy from a trivial start system to the solution

\[ \mathcal{H}(x, t) = (1 - t)Q(x) + tP(x) \]  

(8)
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)
### Polynomial Solvers

**Testing code**

<table>
<thead>
<tr>
<th></th>
<th>dofs</th>
<th>mixed volume</th>
<th>solutions found</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH P2XP1 5X5</td>
<td>122</td>
<td>1</td>
<td>1</td>
<td>70ms</td>
</tr>
<tr>
<td>TH P2XP1 6X6</td>
<td>197</td>
<td>1</td>
<td>1</td>
<td>95ms</td>
</tr>
<tr>
<td>TH P2XP1 8X8</td>
<td>401</td>
<td>1</td>
<td>1</td>
<td>930ms</td>
</tr>
</tbody>
</table>

**Table:** PHC data for stokes problem

<table>
<thead>
<tr>
<th></th>
<th>dofs</th>
<th>mixed volume</th>
<th>solutions found</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 4X4</td>
<td>4</td>
<td>16</td>
<td>16</td>
<td>60ms</td>
</tr>
<tr>
<td>P1 6X6</td>
<td>16</td>
<td>65536</td>
<td>65536</td>
<td>5318s</td>
</tr>
</tbody>
</table>

**Table:** PHC data for nonlinear Laplacian problem
Testing code

<table>
<thead>
<tr>
<th>TH P2XP1 3X3</th>
<th>dofs</th>
<th>mixed volume</th>
<th>solutions found</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>116</td>
<td>$2^{24}$</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

**Table:** PHC data for Navier-Stokes problem
Conclusion

- **FEM Automation** enables flexibility in simulation software.
- **Mathematics ↔ Software Abstractions**
- **Difficult** 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