

SAND2015-XXXXR

LDRD PROJECT NUMBER: 186870

LDRD PROJECT TITLE: A Galerkin/Least-Squares Approach to Viscoelastic Flow

PROJECT TEAM MEMBERS: Daniel S Hariprasad (UNM), Weston W Ortiz (UNM), Rekha R Rao (SNL), P Randall Schunk (SNL)

ABSTRACT:

A Galerkin/least-squares stabilization technique is applied to a discrete Elastic Viscous Stress Splitting formulation of for viscoelastic flow. From this, a possible viscoelastic stabilization method is proposed. This method is tested with the flow of an Oldroyd-B fluid past a rigid cylinder, where it is found to produce inaccurate drag coefficients. Furthermore, it fails for relatively low Weissenberg number indicating it is not suited for use as a general algorithm. In addition, a decoupled approach is used as a way separating the constitutive equation from the rest of the system. A Pressure Poisson equation is used when the velocity and pressure are sought to be decoupled, but this fails to produce a solution when inflow/outflow boundaries are considered. However, a coupled pressure-velocity equation with a decoupled constitutive equation is successful for the flow past a rigid cylinder and seems to be suitable as a general-use algorithm.

INTRODUCTION:

Viscoelastic flows have many applications from enhanced oil recovery to polymer processing and 3D printing. The simulation of these flows present difficult computational problems due to a constitutive equation describing the stress of the fluid [1]. The constitutive equation increases the number of unknowns in the resulting system of governing equations. In Newtonian fluids, the stress tensor is given directly by the velocity and pressure fields, but in viscoelastic flows, it must be solved as another tensor unknown. Moreover, the constitutive equation is hyperbolic, or “wave-like”, in nature, where advective terms dominate the equation. Viscoelastic stress constitutive equations are notoriously difficult to solve, especially as the amount of elasticity in the fluid is increased. This has been termed the High Weissenberg Number Problem. Thus, stabilization techniques are required to handle the numerical instabilities.

Multiple stabilization techniques have been explored for viscoelastic flows previously in the context of the finite element method. Marchal and Crochet [2] developed a high-order 4x4 stress element with Streamline-Upwind/Petrov-Galerkin (SUPG) upwinding in the context of an LBB element for the velocity and pressure space. Marchal and Crochet’s investigation found that SUPG stabilization led to spurious pressure oscillations. They proposed that the stabilization terms should only be applied to the advective terms, rather than the entire constitutive equation. They termed this stabilization Streamline-Upwind (SU) and showed its effectiveness. However, this stabilization technique is inconsistent and results in only first-order accuracy as shown by

Crochet and Legat [4]. Discontinuous-Galerkin (DG) methods have also been demonstrated by Fortin and Fortin [5] and Baijeens [5] where the stress is a discontinuous variable, and a continuous LBB element is used for the velocity and pressure. This method is appealing for the constitutive equation due to its hyperbolic nature. However, DG implementations are significantly more complex compared to other stabilization methods, since the number of unknowns is much larger and additional surface integrations must be carried out at all inflow element boundaries.

Galerkin/least-squares (GLS) is a stabilization technique proposed by Franca and Hughes [6] first applied to viscoelastic flows by Behr et al. [7]. This technique takes a least-squares form of the governing equations and adds it to the Galerkin finite element form. The resulting form of the finite element system includes SUPG terms, as well as pressure stabilizing terms. The pressure stabilizing terms circumvent the notorious LBB condition so that equal-order interpolation of all variables is possible, unlike in the SUPG technique. Implementation of this technique is also straightforward, as it relies on terms already present in the Galerkin method [7]. Furthermore, this method is consistent, so it is more appealing than an inconsistent method such as SU.

In the techniques described above, the governing system of equations is treated as fully-coupled, that is, all of the governing equations are solved at the same time. In a full three-dimensional (3D) viscoelastic flow, the constitutive equation adds 6 degrees of freedom per viscoelastic mode, and a shear-rate tensor adds 9 degrees of freedom. In full 3D, the resulting matrix system for a fully-coupled problem can get unmanageable. Decoupling equations is a useful practice to make the resulting matrix system manageable. This can lead to better conditioned matrices, but it also allows for the use of separate stabilization techniques on each equation. Given the nature of the governing equations, there are two possibilities for decoupling. One involves solving the stress separately and then inputting this into a coupled pressure-velocity equation, and the other is solving each equation separately.

For decoupling pressure from velocity, one popular class of methods are pressure projection methods [8]. A particular form of this method is the Characteristic Based Split (CBS) method [9]. This method has been shown to be successful for a wide class of problems and has been extended to viscoelastic flows [10]. What is most appealing of this method is that it can circumvent the LBB condition allowing for equal order interpolation of the velocity and pressure fields [12].

Matrix systems resulting from the finite element method applied to incompressible flows are usually sparse and large. This means direct solution of the systems is computationally expensive and requires direct Gaussian elimination, because the incompressibility constraint creates a zero on the diagonal of the matrix. Iterative or indirect, methods of solution are needed to solve problems as they scale in size. These iterative methods often rely on the condition number of the matrix and when that condition number is large preconditioning techniques are used to reduce the condition number of the matrix.

Robust matrix preconditioners exist, such as Incomplete LU Factorization (ILUT) [11], but they can often be expensive when scaling a problem and difficult to do in parallel. These robust preconditioners also tend to fail when the matrix system becomes difficult to solve. Decoupling equations leads to better conditioned and smaller matrices meaning computational requirements for solving matrix systems are reduced. The decoupling also allows for the preconditioners to

take advantage of the underlying behavior of the equations that form each matrix system, offering improved performance and convergence.

In this report, we explore a GLS implementation for viscoelastic flow in GOMA [12] following the work of Behr et al. [7]. This method is tested using two benchmarks for viscoelastic flows. In addition, a decoupled equation implementation is created in GOMA for both a CBS method and coupled velocity-pressure solve with a decoupled stress component.

DESCRIPTION OF NUMERICAL METHOD:

Governing equations

An incompressible, Oldroyd-B fluid is considered in a bounded domain Ω with boundary Γ and for all times $t \in [0, T]$. Conservation of momentum and mass result in:

$$\begin{aligned} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f} \right) - \nabla \cdot \boldsymbol{\sigma} &= 0 & \forall \mathbf{x} \in \Omega, \forall t \in [0, T] \\ \nabla \cdot \mathbf{u} &= 0 & \forall \mathbf{x} \in \Omega, \forall t \in [0, T] \end{aligned}$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the velocity field, $\boldsymbol{\sigma}$ is the fluid stress, ρ is the fluid density, and $\mathbf{f} = \mathbf{f}(\mathbf{x}, t)$ is an external body force. Both Dirichlet and Neumann-type boundary conditions are considered:

$$\begin{aligned} \mathbf{u} &= \mathbf{g} & \forall \mathbf{x} \in \Gamma_D \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \mathbf{h} & \forall \mathbf{x} \in \Gamma_N \end{aligned}$$

where $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Initial conditions are taken to be an incompressible flow field:

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \nabla \cdot \mathbf{u}_0 = 0 \forall \mathbf{x} \in \Omega$$

Viscoelastic fluid stresses exhibit a dependence on the rate of strain history. For an Oldroyd-B fluid, the constitutive equation for stress is:

$$\begin{aligned} \boldsymbol{\sigma} &= -p\mathbf{I} + \boldsymbol{\sigma}_1 + \boldsymbol{\sigma}_2 \\ \boldsymbol{\sigma}_1 + \lambda \overset{\nabla}{\boldsymbol{\sigma}}_1 &= 2\mu_p \boldsymbol{\epsilon}(\mathbf{u}) \\ \boldsymbol{\sigma}_2 &= 2\mu_s \boldsymbol{\epsilon}(\mathbf{u}) \\ \mu &= \mu_s + \mu_p \end{aligned}$$

where $p = p(\mathbf{x}, t)$ is the pressure field, $\boldsymbol{\sigma}_1 = \boldsymbol{\sigma}_1(\mathbf{x}, t)$ is the extra stress tensor corresponding to an elastic Maxwell fluid with viscosity μ_p and polymer time constant λ , $\boldsymbol{\sigma}_2 = \boldsymbol{\sigma}_2(\mathbf{x}, t)$ is the

extra stress tensor corresponding to a Newtonian fluid with viscosity μ_s , $\overset{\nabla}{\sigma}_1$ is the upper convected derivative:

$$\overset{\nabla}{\sigma}_1 = \frac{\partial \sigma_1}{\partial t} + \mathbf{u} \cdot \nabla \sigma_1 - (\nabla \mathbf{u})^T \cdot \sigma_1 - \sigma_1 \cdot \nabla \mathbf{u}$$

and $\epsilon(\mathbf{u})$ is the rate of strain tensor:

$$\epsilon(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$$

Note that if $\lambda = 0$, a Newtonian fluid with viscosity μ is recovered from the constitutive equation, and if $\mu_s = 0, \lambda > 0$, a Maxwell fluid is recovered from the constitutive equation. In the case of steady-state calculations, all time derivative terms are dropped.

Weak formulation of the governing equations

The governing equations above are written in strong form above, that is, we seek functions that satisfy those operators pointwise throughout the domain. Given the strong form, a less strict weak or variational form of the governing equations is find the solution $\{\mathbf{u}, p, \sigma_1\}$ such that:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f, \mathbf{v} \right) + (\sigma, \nabla \mathbf{v}) = 0 \forall \mathbf{v}$$

$$(\nabla \cdot \mathbf{u}, q) = 0 \quad \forall q$$

$$(\sigma_1 + \lambda \overset{\nabla}{\sigma}_1 - 2\mu_p \epsilon(\mathbf{u}), \mathbf{S}) = 0 \forall \mathbf{S}$$

where $\{\mathbf{v}, q, \mathbf{S}\}$ are test functions chosen from an appropriate test function space, and (\mathbf{u}, \mathbf{v}) denotes the standard L^2 inner-product:

$$(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} d\Omega$$

Velocity gradient projection

The Elastic Viscous Stress Splitting (EVSS) formulation of stress [13] uses a change of variables, which results in a stabilizing elliptic contribution to the momentum equation. Guénette and Fortin [14] proposed a modification of the EVSS formulation, which greatly simplifies implementation, called the discrete EVSS method (DEVSS/DEVSS-G) where a projection of either the rate of strain tensor (DEVSS) or the velocity gradient is used (DEVSS-G) for the change of variables. Here, a DEVSS-G formulation is used, and the weak formulation is modified to find $\{\mathbf{u}, p, \mathbf{G}, \sigma_1\}$ such that:

$$\begin{aligned} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f, \mathbf{v} \right) + \left(\sigma + 2\mu_p \left(\epsilon(\mathbf{u}) - \frac{1}{2}(\mathbf{G} + \mathbf{G}^T) \right), \nabla \mathbf{v} \right) &= 0 \quad \forall \mathbf{v} \\ (\nabla \cdot \mathbf{u}, q) &= 0 \quad \forall q \\ (\mathbf{G} - (\nabla \mathbf{u})^T, \mathbf{E}) &= 0 \quad \forall \mathbf{E} \\ (\sigma_1 + \lambda \overset{\nabla}{\sigma}_1 - 2\mu_p \epsilon(\mathbf{u}), \mathbf{S}) &= 0 \quad \forall \mathbf{S} \end{aligned}$$

where \mathbf{G} is a continuous projection of the velocity gradient. The upper convected derivative is modified to:

$$\overset{\nabla}{\sigma}_1 = \frac{\partial \sigma_1}{\partial t} + \mathbf{u} \cdot \nabla \sigma_1 - \mathbf{G}^T \cdot \sigma_1 - \sigma_1 \cdot \mathbf{G}$$

This method requires an additional tensor equation.

Galerkin/Least-Squares

Following the work of Behr et al. [7] a three-field GLS (GLS3) formulation as find $\{\mathbf{u}, p, \sigma_1\}$ such that:

$$\begin{aligned} &\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f, \mathbf{v} \right) + (\sigma, \nabla \mathbf{v}) + (\nabla \cdot \mathbf{u}, q) \\ &\quad + (\sigma_1 + \lambda \overset{\nabla}{\sigma}_1 - 2\mu_p \epsilon(\mathbf{u}), \mathbf{S}) \\ &\quad + \tau_{MOM} \left(\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f \right) + \nabla \cdot \sigma, \rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{v} \right) \right) \\ &\quad + \tau_{MOM} \left(\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f \right) + \nabla \cdot \sigma, \nabla q - \nabla \cdot \mathbf{S} - 2\mu_p \nabla \cdot \epsilon(\mathbf{v}) \right) + \tau_{CONT} (\nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{v}) \\ &\quad + \tau_{CONST} (\sigma_1 + \lambda \overset{\nabla}{\sigma}_1 - 2\mu_p \epsilon(\mathbf{u}), \mathbf{S} + \lambda \overset{\nabla}{\mathbf{S}} - 2\mu_p \epsilon(\mathbf{v})) = 0 \quad \forall \mathbf{v}, q, \mathbf{S} \end{aligned}$$

where τ_{MOM} , τ_{CONT} , and τ_{CONST} are stability parameters. Previous work on the momentum term has shown that reducing it to pressure only provides adequate stabilization leading to the so called Pressure-Stabilized/Petrov-Galerkin method (PSPG) [15]. Furthermore, this formulation lacks the DEVSS-G contribution outlined above. Given both of these points, a new GLS form is proposed for the current work as find $\{\mathbf{u}, p, \mathbf{G}, \sigma_1\}$:

$$\begin{aligned} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f, \mathbf{v} \right) + \left(\sigma + 2\mu_p \left(\epsilon(\mathbf{u}) - \frac{1}{2}(\mathbf{G} + \mathbf{G}^T) \right), \nabla \mathbf{v} \right) + (\nabla \cdot \mathbf{u}, q) \\ (\mathbf{G} - (\nabla \mathbf{u})^T, \mathbf{E}) + (\sigma_1 + \lambda \overset{\nabla}{\sigma}_1 - 2\mu_p \epsilon(\mathbf{u}), \mathbf{S}) \end{aligned}$$

$$\begin{aligned}
& +\tau_{MOM}(\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f) + \nabla \cdot (\boldsymbol{\sigma} + 2\mu_p(\boldsymbol{\epsilon}(\mathbf{u}) - \frac{1}{2}(\mathbf{G} + \mathbf{G}^T)), \nabla q) \\
& \quad +\tau_{CONT}(\nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{v}) \\
& +\tau_{CONST}(\boldsymbol{\sigma}_1 + \lambda \overset{\nabla}{\boldsymbol{\sigma}}_1 - 2\mu_p \boldsymbol{\epsilon}(\mathbf{u}), \mathbf{S} + \lambda \overset{\nabla}{\mathbf{S}} - 2\mu_p \boldsymbol{\epsilon}(\mathbf{v})) = 0 \quad \forall \mathbf{v}, q, \mathbf{E}, \mathbf{S}
\end{aligned}$$

This formulation modifies the GLS3 formulation with the velocity projection typical of DEVSS-G and reduces the momentum GLS term to include only the test pressure. This formulation will be referred to as DEVSS-G/GLS.

Decoupled equations

The DEVSS-G formulation has four equations resulting in multiple possibilities for decoupling equations. Two possible decoupling procedures will be explored. The first procedure is to decouple all equations. When this occurs, the CBS method will be used to decouple pressure and velocity followed by solving the equation for velocity gradient then stress.

Characteristic Based Split

The CBS methods rely on splitting the momentum operator and updating it with a correction from the mass operator. There are two possible ways to perform this split: Split A and Split B. Split A is reported to be better for steady-state problems, while Split B is better for transient problems [9]. Since transient problems are of vast interest for viscoelastic flows, Split B was chosen. The algorithm is as follows for an incompressible, Newtonian fluid:

$$\begin{aligned}
\frac{\rho(\mathbf{u}^* - \mathbf{u}^n)}{\Delta t} + \rho \mathbf{u}^n \cdot \nabla \mathbf{u}^n - \frac{\mu}{2} \nabla^2 \{\mathbf{u}^* + \mathbf{u}^n\} + \nabla p^n &= 0 \\
\nabla^2 p^* - \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}^* &= 0 \\
\frac{\rho(\mathbf{u}^{n+1} - \mathbf{u}^*)}{\Delta t} + \nabla p^* &= 0 \\
p^{n+1} - p^* - p^n + \frac{\mu}{2} \nabla \cdot \mathbf{u}^* &= 0
\end{aligned}$$

where \mathbf{u}^n is the velocity at the nth time step, p^n is pressure at the nth time step, \mathbf{u}^* is an intermediate velocity, and p^* is an intermediate pressure. Here, the intermediate pressure is used in a Pressure-Poisson equation in order to enforce incompressibility when the pressure and velocity are decoupled. A weak form of this algorithm can also be specified and used in the context of finite elements.

The second procedure involves leaving velocity and pressure coupled, then decoupling the velocity gradient and stress equations. In this decoupling procedure, velocity and pressure are coupled just as in a Newtonian flow, then the velocity gradient projection is solved using the result of the velocity-pressure system, and finally the stress tensor is solved using the results of the preceding two steps. The stress is then input into the first step, and this is repeated until steady state is reached. This is referred to as the decoupled DEVSS-G/SUPG method.

Goma had to be modified to accommodate and build multiple problem graphs and store multiple solutions for each step in a decoupled system. To create a decoupled solve in Goma, a separate procedure for solving the problem had to be created. Convergence at every time step had to be checked for in the result of each step of the CBS solve and for each group solve in the decoupled solve instead of once at each time step. The ability to pass information from previous steps also had to be implemented as each step and group solve relies on information from the previous solves and previous time steps.

To solve steady state problems for both CBS and decoupled methods we marched to steady state where we solve a problem from some initial state and continue to step through time until the problem stops changing. We checked this by computing the L^2 -norm of the difference between the current time step solution vector and the previous time step solution vectors for all steps and once all norms were under a specified tolerance (often 10^{-5}) we considered the problem to have reached steady state.

Linear solvers

We used both direct solvers and iterative solvers in solving our matrix systems. All solutions were obtained using the Trilinos [16] linear solver packages and associated third party libraries. Trilinos has both iterative and direct solver interfaces and we have used both within GOMA. For direct solves we used a Multifrontal Massively Parallel sparse direct Solver (MUMPS) [17] through the Amesos package in Trilinos which offered the best parallel performance for a direct solver on our problems. Multifrontal direct solvers are not scalable, and are only useful up to 16 processors.

We also used Trilinos's iterative solvers, namely the biconjugate gradient stabilized method (BiCGStab) [18] and the generalized minimal residual method (GMRES) [19] through the AztecOO package. In order to access additional preconditioning methods and solvers Stratimikos, a Trilinos package that provides access to many of its linear solver packages, was interfaced with in GOMA.

The Semi-Implicit Method for Pressure Linked Equations (SIMPLE) and its variant SIMPLEC offers a preconditioning method for solving incompressible Navier-Stokes equations through approximate block factorizations [20]. Trilinos and its package Tekos were used through Stratimikos in order to precondition matrices created by the Navier-Stokes equations with coupled momentum, u , and pressure, p , with SIMPLEC during decoupled solves. ILUT through the AztecOO package was also used for multiple matrix types.

RESULTS:

Galerkin/Least-Squares

Two standard test problems for viscoelastic flows are used to test the DEVSS-G/GLS method: a Maxwell fluid in a 4:1 contraction and flow of an Oldroyd-B fluid past a rigid cylinder.

4:1 Contraction

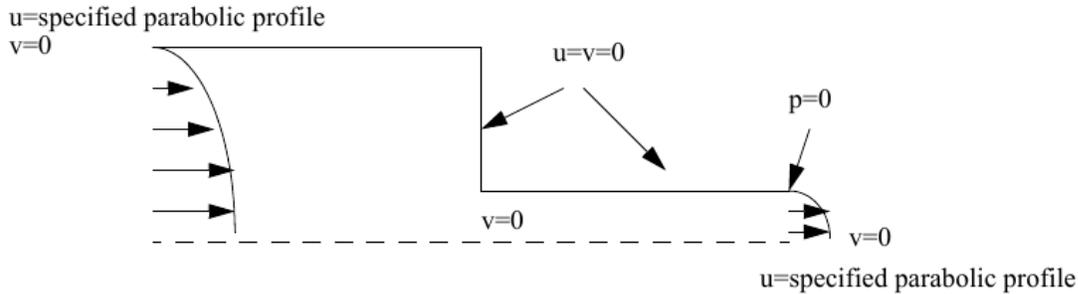


Figure 1: Computational domain and boundary conditions for a Maxwell fluid in a 4:1 contraction. Note $u_y=v$ in this diagram.

A computational domain with a sudden contraction from a height of 4 to a height of 1 is simulated with a Maxwell fluid, as shown in Figure 1. Recall that a Maxwell fluid is specified by taking $\mu_s = 0$ in the constitutive equation. Boundary conditions are taken to be no-slip on solid boundaries and symmetry about the centerline. At the inflow, velocity and stresses are specified:

$$\begin{aligned} u_x &= \frac{3}{2} \left(1 - \frac{y^2}{16}\right) \\ u_y &= 0 \\ \sigma_1^{xx} &= \frac{18}{256} We y^2 \\ \sigma_1^{xy} &= -\frac{3}{16} y \\ \sigma_1^{yy} &= 0 \end{aligned}$$

where We is the dimensionless Weissenberg number representing the ratio of elastic to viscous stresses. At the outflow, velocities are specified:

$$\begin{aligned} u_x &= 6(1 - y^2) \\ u_y &= 0 \end{aligned}$$

In addition, a pressure datum $p = 0$ is specified at a point on the corner of the domain at the outflow. The polymer viscosity is taken to be $\mu_p = 1$ and the polymer time constant is taken to be $\lambda = 0.02$ resulting in $We = 0.02$.

The results of the DEVSS-G/GLS implementation were compared with a DEVSS-G/SUPG implementation in GOMA. The DEVSS-G/GLS results were found to be in good agreement with the DEVSS-G/SUPG results, with only minor differences appearing near the stress singularity near the contraction.

Flow past a rigid cylinder

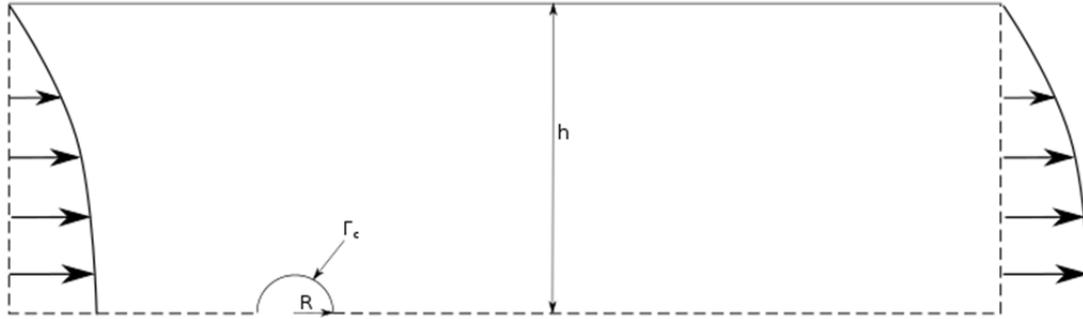


Figure 2: Computational domain for flow of an Oldroyd-B fluid past a rigid cylinder.

Flow of an Oldroyd-B fluid between two parallel plates past a rigid circular cylinder is considered, as shown in Figure 2. The cylinder radius, $R = 1$, is taken to be one-eighth of the slit half-width, $h = 8$. Boundary conditions are taken to be no-slip on solid boundaries and symmetry conditions about the centerline. At the inflow and outflow boundaries a parabolic flow profile is specified as follows:

$$\begin{aligned} u_x &= \frac{3Q}{2h} \left(1 - \frac{y^2}{h^2}\right) \\ u_y &= 0 \\ \sigma_1^{xx} &= 2\lambda\mu_p \left(\frac{-3y}{2h^2}\right)^2 \\ \sigma_1^{xy} &= -\mu_p \frac{3y}{2h^2} \\ \sigma_1^{yy} &= 0 \end{aligned}$$

where $Q = 8$ is the flow rate per unit thickness. The polymer viscosity is taken to be $\mu_p = 0.41$ and the solvent viscosity is taken to be $\mu_s = 0.59$. The polymer time constant, λ , is varied allowing for variations in the Weissenberg number:

$$We = \frac{Q\lambda}{hR} = \lambda$$

The drag on the cylinder, D , is a characteristic quantity of the flow field as it varies with Weissenberg number. It is computed as follows:

$$D = -2 \int_{\Gamma_c} e_1 \cdot \sigma \cdot \mathbf{n} d\Gamma$$

where Γ_c is the surface of the cylinder, e_1 is the unit horizontal vector, and \mathbf{n} is the unit normal vector on the cylinder surface.

Sun et al. [21] studied the flow past a rigid cylinder using discrete Adaptive Viscoelastic Stress Splitting and the discontinuous Galerkin method (DAVSS-G/DG).

Figure 3 shows a comparison of drag values for the DEVSS-G/GLS and DEVSS-G/SUPG implementations in GOMA and the DAVSS-G/DG of Sun et al. for values of $We \in [0.0, 2.0]$. The results of DEVSS-G/SUPG and DAVSS-G/DG are in good agreement, with only a slight deviation at higher We .

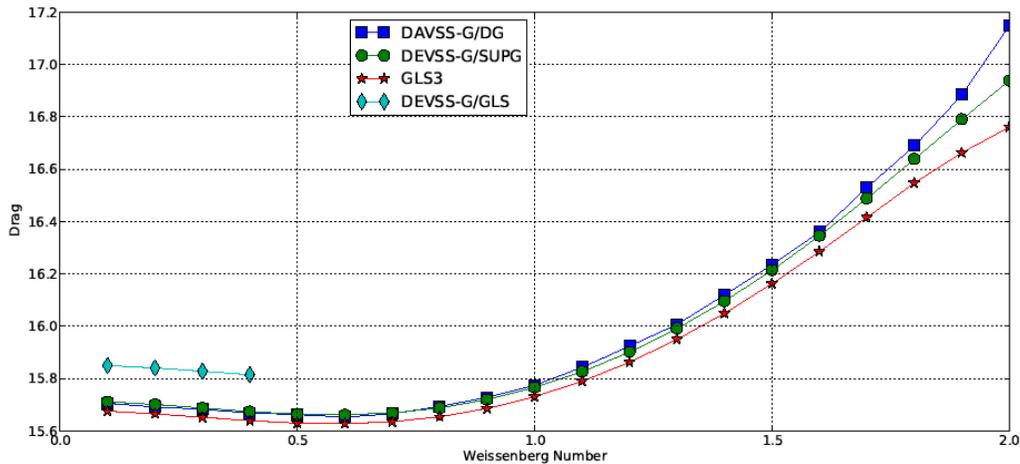


Figure 3: Comparison of drag forces on a rigid cylinder for DAVSS-G/DG [23], DEVSS-G/SUPG, GLS3 [8], and DEVSS-G/GLS.

However, the results of the DEVSS-G/GLS do not show good agreement at low We , as the DEVSS-G/GLS returned significantly higher drag value. Furthermore, the DEVSS-G/GLS method only reached a maximal $We = 0.4$ before it failed to converge.

Decoupled equations

With a DEVSS-G formulation for viscoelastic flows, there are multiple ways of decoupling the governing equations. All equations can be considered separate, or certain sets of equations can be coupled together, e.g., velocity and pressure are coupled but the velocity gradient projection and stress are solved separately. In the following, various forms of decoupling are explored, and when pressure and velocity are decoupled, the CBS method is used.

Lid driven cavity timing using CBS

Given the transient nature of the CBS Split B method, it is more appropriate to test it using a transient problem. With this consideration, the classic lid-driven cavity test problem [9] is modified to have a time-dependent moving lid, as shown in Figure 4. The fluid is taken to be Newtonian with Reynolds number, $Re = 100$.

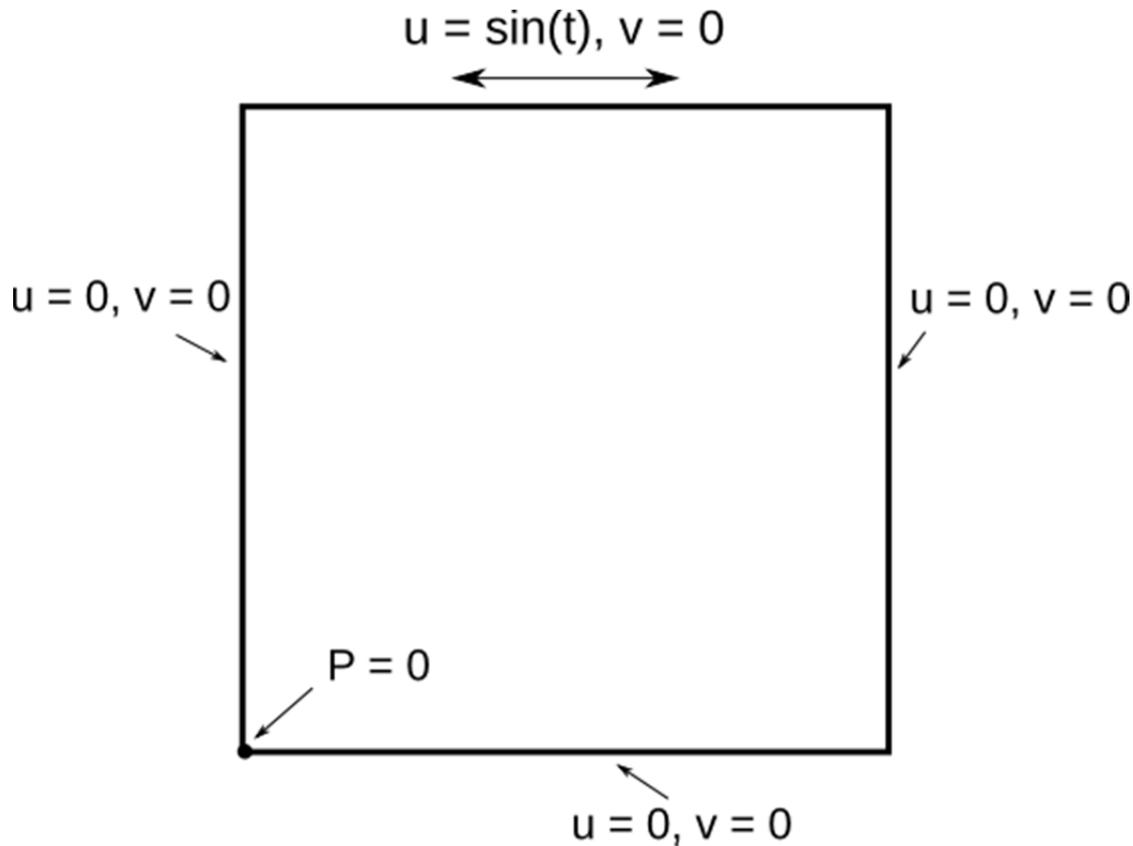


Figure 4: Time-dependent sinusoidal motion of a Lid Driven Cavity

With the CBS method, we were able to achieve results that matched the fully-coupled solution. For the fully-coupled solution, PSPG with quadratic elements (Q2) were used in order to take advantage of using the iterative solver GMRES with the ILUT preconditioner. For the CBS method, mixed order elements were used as required by the LBB condition, and BiCGStab without preconditioning was used for the solution of the matrix system. We ran both the fully-coupled and decoupled problems on a coarse mesh, a fine mesh, and finally a 3D mesh (each with increasing number of elements).

Table 1 show that CBS was faster to solve for our three mesh types and seems to scale well with the increase in mesh size.

Table 2 shows the representative assembly and solve times on the 3D mesh and we can see that for the fully coupled solution the solve takes more than half the time while the CBS method the majority of the time is spent in assembly.

Table 1: Timing results for lid driven cavity.

	Fully Coupled (min)	Segregated (min)
Coarse	31.59	25.5
Fine	280.00	151.83
3D	884.10	458.46

Table 2: Assembly and solve times for 3D mesh for Lid Driven Cavity

	Assembly (sec)	Solve (sec)
CBS step 1	5.3	0.14
CBS step 2	0.59	0.01
CBS step 3	2.6	0.21
CBS step 4	0.56	0.01
CBS Total	9.1	0.37
Fully Coupled Total	14	16

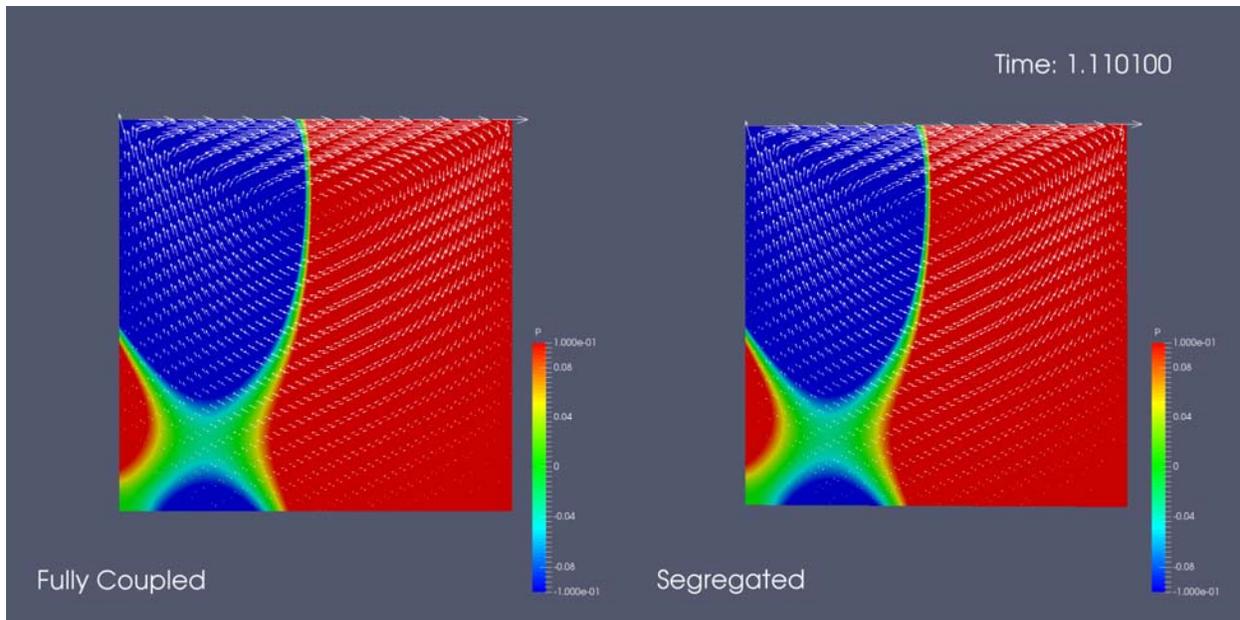


Figure 5: Snapshot at a single time step for the CBS (Segregated) and fully coupled simulations results of the lid driven cavity.

Poiseuille flow using CBS

Flow through a channel, Poiseuille flow, is used to test boundary condition implementations in the CBS method. For Poiseuille flow, a Newtonian fluid under pressure-driven flow is considered. No-slip conditions are taken on solid boundaries, and a parabolic profile is taken at the inflow/outflow:

$$u_x = -\frac{\Delta p}{2\mu} \left(1 - \frac{y^2}{h^2}\right)$$

$$u_y = 0$$

where Δp is a specified pressure drop in the channel, and h is the height of the channel.

With an initially zero velocity and pressure field, the CBS method fails to achieve a fully developed Poiseuille flow profile. The velocity profile oscillates at the inflow and outflow, which indicates an issue with boundary conditions at these types of boundaries.

Decoupled 4:1 contraction

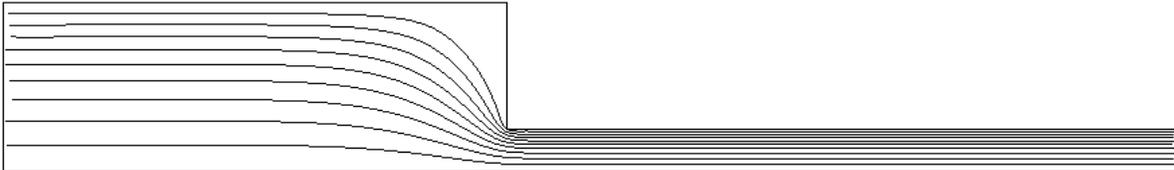


Figure 6: Streamlines of the decoupled solution to the 4:1

Given the inflow/outflow issue with Poiseuille flow in the CBS method, the 4:1 contraction is simulated in a decoupled fashion but with a coupled velocity and pressure equation, or the decoupled DEVSS-G/SUPG method.

To decouple the 4:1 contraction multiple methods were tried. Three main methods of decoupling were tried: first 3 groups $\{u, p\}$, $\{G\}$, and $\{\sigma\}$, then two sets of two groups $\{u, p, G\}$, and $\{\sigma\}$, and finally $\{u, p\}$, and $\{G, \sigma\}$. We found that the first decoupling of three groups, $\{u, p\}$, $\{G\}$, and $\{\sigma\}$, performed best and also matched the steady state fully coupled solution. This decoupling also had the advantage of allowing iterative solvers on all matrix systems.

Table 3 shows how solution time can change based on the solution method for one group of equations. MUMPS has a consistent solve time and was able to solve the problem until steady state. GMRES and ILUT seem to have poor performance because after some time steps are taken the matrix system becomes harder to solve and ILUT is not able to properly precondition the system and eventually GMRES fails to converge. GMRES and SIMPLEC were able to converge and solution times were kept low at the start but increased as the system became more complex but this was able to solve the problem and reach the same solution as the MUMPS solve. This was a fairly small, 2D, steady state problem. For larger, transient problems, we feel that the GMRES/SIMPLEC combination shows great promise.

We are also working on a decoupling procedure that will solve the steady problem directly, without any time marching. This should also help decrease solution times.

Table 3: 4 to 1 contraction {u,p} matrix solve times, note GMRES and ILUT failed to converge after some time.

Solver	Min Time (s)	Max Time (s)
MUMPS	4	4.1
GMRES and ILUT	4.5	23
GMRES and SIMPLEC	0.8	6.8

Flow past a rigid cylinder

The flow past a rigid cylinder problem is simulated using a decoupled DEVSS-G/SUPG method. The results of the drag force are compared with the fully-coupled DEVSS-G/SUPG and DAVSS-G/DG [21]. The results for the drag as a function of We are shown in Figure 7: Comparison of drag forces *on a rigid cylinder for DAVSS-G/DG, fully-coupled DEVSS-G/SUPG, and decoupled DEVSS-G/SUPG*. There is a small deviation between the decoupled DEVSS-G/SUPG and the fully-coupled DEVSS-G/SUPG.

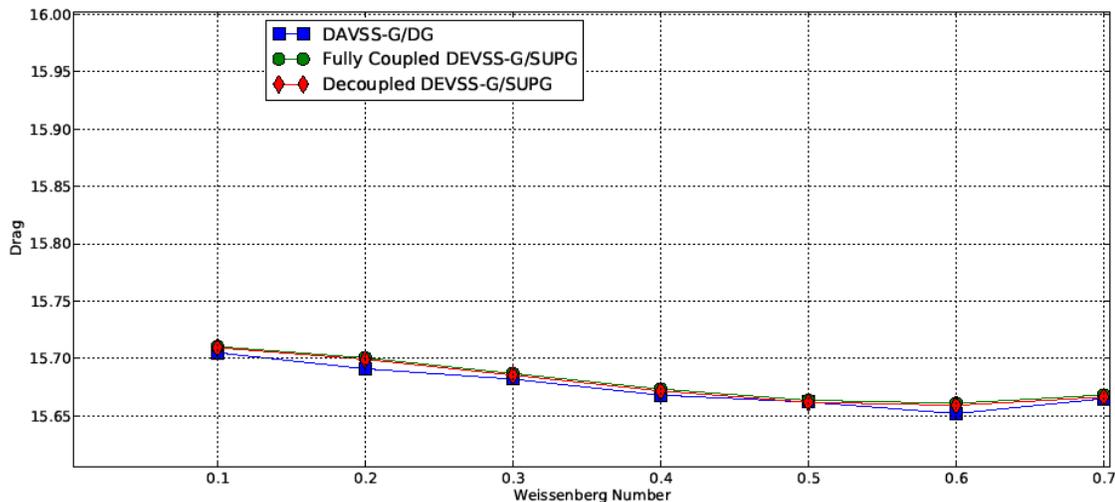


Figure 7: Comparison of drag forces on a rigid cylinder for DAVSS-G/DG [23], fully-coupled DEVSS-G/SUPG, and decoupled DEVSS-G/SUPG

DISCUSSION:

Galerkin/Least-Squares

The results for DEVSS-G/GLS showed that it would not be suitable as a general use algorithm in GOMA. While the 4:1 contraction problem appeared to be in good agreement with the DEVSS-G/SUPG method, the flow past a rigid cylinder failed to converge at a relatively low Weissenberg number. Not only did it fail, but the drag force was significantly different than those reported in the literature [7] [21], as well as those produced by GOMA's DEVSS-G/SUPG implementation. This indicates there were significant flaws in deriving the DEVSS-G/GLS algorithm.

Compared to the GLS3 method, some differences exist. The first difference is the lack of a GLS term for the velocity gradient projection. The next noticeable difference is that the stability parameter used was not determined for use in the DEVSS-G form of the system. These stability parameters are key to the performance of the GLS algorithms. Moreover, they must be fine-tuned so that the GLS form of the system to satisfy the LBB theorem. The other difference is the lack of a diffusive term in the GLS momentum contribution. The momentum GLS term as it appeared in Behr et al. is:

$$\begin{aligned} & \tau_{MOM}(\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f) + \nabla \cdot \sigma, \rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{v})) \\ & + \tau_{MOM}(\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f) + \nabla \cdot \sigma, \nabla q - \nabla \cdot \mathbf{S} - 2\mu_p \nabla \cdot \epsilon(\mathbf{v})) \end{aligned}$$

compared to the PSPG term that appears in the DEVSS-G/GLS:

$$\tau_{MOM}(\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f) + \nabla \cdot (\sigma + 2\mu_p(\epsilon(\mathbf{u}) - \frac{1}{2}(\mathbf{G} + \mathbf{G}^T))), \nabla q)$$

One noticeable difference is the GLS3 term contains $(\nabla \cdot \sigma, \nabla \cdot \mathbf{S})$ which adds diffusion to the system, namely in the stress equation. This was likely the key difference between the GLS3 and the DEVSS-G/GLS.

The GLS3 method was extended to a DEVSS-G formulation and a so called four-field GLS (GLS4) form by Coronado et al. [22]. In their work, they claim that the GLS term for the velocity gradient is not necessary. But, the GLS4's momentum term is as follows:

$$\tau_{MOM}(\nabla \cdot \sigma, \nabla q - \nabla \cdot \mathbf{S} - \mu_p \nabla \cdot (\mathbf{G} + \mathbf{G}^T))$$

This further indicates the importance of GLS terms associated with the stress contribution in the momentum operator.

Decoupling and Linear Solvers

The CBS method showed promising improvements in solve time for the lid-driven cavity due to the simplicity of the resulting matrices. Because BiCGStab was able to converge quickly without preconditioners much less work was needed and allowed for speed up from the fully coupled solution.

From

Table 2 the major bottleneck of the problem was from assembly this could be reduced if LBB elements were not required as in 3D Q2 elements require 27 nodes at which evaluations occur increasing computational time dramatically.

However, boundary conditions for the CBS method's Pressure Poisson step at inflow/outflow boundaries are difficult to handle. The lid-driven cavity has Dirichlet conditions everywhere on the boundaries, but most problems of interest are not in this form. Boundary conditions for Pressure Poisson equations are notoriously difficult to handle as there are no pressure specific conditions that can easily be specified from given velocity or traction conditions. There are strategies that have been applied to other Pressure Poisson equations [23] [24] that may be useful for the CBS method's form of the Pressure Poisson equation. The difficulty with these formulations is that they typically require second-order derivatives of the stress tensor, which is not typically evaluated in any of the weak forms of the governing equations from Section 2.

In the 4:1 contraction the decoupling of solutions of equations allowed for iterative solvers to be used on a problem which required direct solution when fully coupled. An already built equation specific preconditioner, SIMPLEC, was also used which is not available when fully coupled and had better performance than the robust ILUT method which failed to converge when the problem became difficult. These matrices however did not turn out to be as nice as those created by the CBS method and as shown with ILUT failing the matrix solution can still be tricky.

These decoupling results suggest that the smaller and more specific decoupled matrices allow for techniques which are not available when solving the fully coupled problems, though some difficulties in validating the decoupling may exist as for the 4:1 contraction only one method of three decouplings was accurate and also performed well.

ANTICIPATED IMPACT:

This research will open doors for modeling soft solids, whose behavior is neither fluid nor solid and will lead to publishable results. Currently, no scalable viscoelastic flow solver is available to understand manufacturing flows of polymers, especially for flows involving free surfaces such as injection molding and extrusion. With the ongoing life extension programs and Sandia's own manufacturing efforts, production of many new polymeric components including foam parts, encapsulants, and pads is underway. A scalable computational modeling approach for viscoelastic polymer flows will allow us to improve the design of new processes and troubleshoot existing processes, thereby increasing yields and reducing costs and production time.

In addition to manufacturing, viscoelastic flow is important for enhanced recovery in the oil and gas industry where guar solutions and foams are used to sweep out residual oil deposits and sand-filled guar "proppant" is used to prop formations to increase yield of both shale-gas and shale-oil. We are currently supporting a Geoscience LDRD "The Effect of Proppant Placement on Closure of Fractured Shale Gas Wells" using suspension models based on Newtonian

suspending fluids. Adding the elastic stresses to the carrier fluid could make our solutions both more realistic and more impactful.

We also have a collaboration with University of Texas Petroleum Department (Prof. Matthew Balhoff, Dr. Peixi Zhu) on modeling an oil droplet surrounded by viscoelastic fluid passing through a constriction. These problems are transient and require fine meshes and frequent remesh-remapping steps as the moving mesh gets sheared or distorted with the motion of the drop. One constriction with one set of fluid properties can take 24 hours to run on 6 processors using a parallel direct solver. With our new method, we hope to significantly reduce run times so that we can carry out parameter studies of constriction geometries and fluid properties, allowing upscaling of the results to a network model for porous flow in the reservoir. We were contacted by Prof. Ernesto Di Maio (University of Naples, Italy) to collaborate on bubble expansion in a viscoelastic fluid to produce an open cell foam. This work builds on our viscoelastic flow expertise and on our current vapor-liquid equilibrium modeling of bubble growth in polymerizing fluids for polyurethanes. Because of our viscoelastic free surface capability, we were asked to collaborate with the University of Vermont (Profs. Yves Dubief and Patrick Lee) on coextrusion modeling of polymers using a 3D printing technique. This a particularly appealing collaboration since Prof. Dubief is a code developer with experience in viscoelastic flow and Prof. Lee is an experimentalist with experience in polymer processing. They would also like to set up a collaboration using Goma as the computational platform with Husky, the largest manufacturer of injection molding equipment in the world. We have also begun a collaboration with Prof. Jae Wook Lee (Sogang University, South Korea), who is an expert in viscoelastic flow modeling, code development, and GLS methods. He has taken the Goma training and plans to use Goma for viscoelastic flow modeling and as a platform for code development and research.

Because of our LDRD work, we successfully obtained follow-on funding to complete our viscoelastic flow speedup from Gillette through a two-year CRADA. Through a CRADA with P&G, we are working on speeding up 3D flows of viscoelastic flows in complex geometries with nontrivial inflow conditions, which do not have an analytical representation. We are also working on a new CRADA with 3M on polymer processing and flow instabilities based on this project. In summary, many new and exciting viscoelastic flow projects are underway.

For next steps, since we have received follow-on funding, we plan to modify our GLS to be more like the published versions of GLS3 and GLS4 and see if this improves performance and accuracy. We will also investigate other formulations for the pressure-projection equations since this showed such promise for enabling iterative solvers. Possibly other formulations are available that provide both scalability and accuracy for the pressure field. For 3D viscoelastic flows, we may add a mini-element to Goma. The mini is a stabilized LBB element based on a linear tetrahedral with a cubic bubble. It has proved stable and accurate for viscoelastic flow and performs better than traditional LBB elements once the bubble is condensed out.

In addition to the direct impact we have had in this project by improving the performance of our viscoelastic solvers, we have had an indirect impact on our open source software product, Goma 6.0. We have made it easier to run Goma in parallel by adding decomposition and recomposition to Goma, instead of as stand-alone programs. We have had a new software release, Goma 6.1,

based on this improvement. We now can solve any problem either fully-coupled or decoupled and apply different solvers to the different matrices. We have also added new solver/preconditioner options through the Stratimikos interface to Trilinos, which have great potential for reducing solve times. As part of this project, we have begun a port of Goma to GPU and improved its parallel performance.

CONCLUSION:

A possible stabilization method, DEVSS-G/GLS, was proposed for use on viscoelastic flows. The proposed method was applied to two test problems. While it did not appear to show significant deviations from the DEVSS-G/SUPG method in the 4:1 contraction problem, its inaccuracy appeared in the flow past a rigid cylinder problem. This indicates the method is not suitable for general use in GOMA. However, the method of Coronado et al. [22] showed possible reasons for this failure. Their GLS4 implementation relies on the same DEVSS-G formulation, but the momentum stabilizing term used in their work contains more than just the PSPG term. Their method is relatively close to the DEVSS-G/GLS implementation, so it presents a possible next step for GOMA.

A decoupling procedure was also tested in order to decouple the stress from the pressure and velocity variables. This opened the possibility for multiple formulations, as it was not clear whether the velocity and pressure variables should still be coupled.

When the velocity and pressure variables were decoupled, a CBS method was used. The CBS method relies on the use of Pressure Poisson equation. This method showed promise for the lid-driven cavity problem, which contains all Dirichlet boundary conditions. However, the CBS method failed to produce a solution when Neumann boundary conditions were considered, such as in a Poiseuille flow. This is due to the fact that when velocities or stresses are specified at boundaries, a pressure boundary condition is not necessarily specified at those boundaries. Since many problems of interest have inflow/outflow boundaries, the CBS method is not seen as suitable for GOMA until the issue of boundary conditions is resolved.

When the velocity and pressure remained coupled, little to no difference was observed compared to the fully-coupled solutions for both the 4:1 contraction and flow past a rigid cylinder problem. This method is intriguing as it allows for the use of linear solvers and matrix preconditioners previously unusable in the fully-coupled approach, e.g., SIMPLEC and GMRES. However, the decoupling of the stress and velocity gradient equations must be carefully considered, as the 4:1 contraction was only viable with both equations kept separate. Despite this, this approach to decoupling the stress and velocity gradient with a coupled pressure and velocity solve does seem suited for use as a general algorithm in GOMA.

Aside from the technical progress, we have also developed several new collaborations based on this work and also negotiated two new CRADAs and are working on another one. This has been a very successful LDRD project since was very small (50k) and has led to follow-on funding and collaborations with four universities.



References

- [1] R. G. Owens and T. N. Phillips, *Computational rheology*, vol. 2, World Scientific, 2002.
- [2] J. M. Marchal and M. J. Crochet, "Hermitian finite elements for calculating viscoelastic flow," *Journal of Non-Newtonian Fluid Mechanics*, vol. 20, pp. 187-207, 1986.
- [3] A. N. Brooks and T. J. Hughes, "Streamline-upwind /Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Computer methods in applied mechanics and engineering*, vol. 32, no. 1, pp. 199-259, 1982.
- [4] M. J. Crochet and V. Legat, "The consistent streamline-upwind /Petrov-Galerkin method for viscoelastic flow revisited," *Journal of non-newtonian fluid mechanics*, vol. 42, no. 3, pp. 283-299, 1992.
- [5] M. Fortin and A. Fortin, "A new approach for the FEM simulation of viscoelastic flows," *Journal of Non-Newtonian Fluid Mechanics*, vol. 32, no. 3, pp. 295-310, 1989.
- [6] L. P. Franca and T. J. Hughes, "Two classes of mixed finite element methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 69, no. 1, pp. 89-129, 1988.
- [7] M. Behr, D. Arora, O. M. Coronado and M. Pasquali, "GLS-type finite element methods for viscoelastic fluid flow simulation," in *Proc of the Third MIT Conference on Computational Fluid and Solid Mechanics, Massachusetts Institute of Technology, Cambridge, USA, 2005*.
- [8] J.-L. Guermond and L. Quartapelle, "On stability and convergence of projection methods based on pressure Poisson equation," *International Journal for Numerical Methods in Fluids*, vol. 26, no. 9, pp. 1039-1053, 1998.
- [9] O. C. Zienkiewicz, R. L. Taylor and P. Nithiarasu, *The finite element method for fluid dynamics*, Butterworth-heinemann, 2005.
- [10] P. Nithiarasu, "A fully explicit characteristic based split (CBS) scheme for viscoelastic flow calculations," *International journal for numerical methods in engineering*, vol. 60, no. 5, pp. 949-978, 2004.
- [11] Y. Saad, "ILUT: A dual threshold incomplete LU factorization," *Numerical Linear Algebra with Applications*, vol. 1, no. 4, pp. 387-402, 1994.
- [12] P. R. Schunk, R. R. Rao, K. S. Chen, D. A. Labreche, A. C. Sun, M. M. Hopkins, H. K. Moffat, R. A. Roach, P. L. Hopkins, P. K. Notz, S. A. Roberts, P. A. Sackinger, S. R. Subia, E. D. Wilkes, T. A. Baer, D. R. Noble and R. B. Secor, "GOMA 6.0 - A Full-Newton Finite Element Program for Free and Moving Boundary Problems with Coupled Fluid/Solid Momentum, Energy, Mass, and Chemical Species Transport: User's Guide," Sandia National Labs., Albuquerque, NM (United States), 2013.
- [13] A. N. Beris, R. C. Armstrong and R. A. Brown, "Finite element calculation of viscoelastic flow in a journal bearing: I. Small eccentricities," *Journal of Non-Newtonian Fluid Mechanics*, vol. 16, no. 1, pp. 141-172, 1984.
- [14] R. Guénette and M. Fortin, "A new mixed finite element method for computing viscoelastic flows," *Journal of non-newtonian fluid mechanics*, vol. 60, no. 1, pp. 27-52, 1995.
- [15] T. J. Hughes, L. P. Franca and M. Balestra, "A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of

- the Stokes problem accommodating equal-order interpolations," *Computer Methods in Applied Mechanics and Engineering*, vol. 59, no. 1, pp. 85-99, 1986.
- [16] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams and K. S. Stanley, "An overview of the Trilinos project," *ACM Trans. Math. Softw.*, vol. 31, no. 3, pp. 397-423, 2005.
- [17] P. R. Amestoy, I. S. Duff, J. Koster and J.-Y. L'Excellent, "A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 1, pp. 15-41, 2001.
- [18] H. A. van der Vorst, "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631-644, 1992.
- [19] Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856-689, 1986.
- [20] E. C. Cyr, J. N. Shadid and R. S. Tuminaro, "Stabilization and scalable block preconditioning for the Navier Stokes equations," *Journal of Computational Physics*, vol. 231, no. 2, 2012.
- [21] J. Sun, M. D. Smith, R. C. Armstrong and R. A. Brown, "Finite element method for viscoelastic flows based on the discrete adaptive viscoelastic stress splitting and the discontinuous Galerkin method: DAVSS-G/DG," *Journal of Non-Newtonian Fluid Mechanics*, vol. 86, no. 3, pp. 281-307, 1999.
- [22] O. M. Coronado, D. Arora, M. Behr and M. Pasquali, "Four-field Galerkin/least-squares formulation for viscoelastic fluids," *Journal of non-newtonian fluid mechanics*, vol. 140, no. 1, pp. 132-144, 2006.
- [23] H. Johnston and J.-G. Liu, "Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term," *Journal of Computational Physics*, vol. 199, no. 1, pp. 221-259, 2004.
- [24] D. Shirokoff and R. R. Rosales, "An efficient method for the incompressible Navier–Stokes equations on irregular domains with no-slip boundary conditions, high order up to the boundary," *Journal of Computational Physics*, vol. 230, no. 23, pp. 8619-8646, 2011.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.