Modifications done to ParStokes for Anisotropic Viscosity

September 2, 2021

The goal would be to implement the usage of anisotropic in rheology in the ParStokes solver of Elmer. For it the constitutive equation of the material is written under the form:

$$s = \mu\epsilon \tag{1}$$

where s and ϵ are the deviatoric stress and strain rate tensors, and μ is the fourth order viscosity tensor tensor. In Voigt notation, s and ϵ are 1x6 vectors and μ is a 6x6 matrix.

1 Important Convention

In standard Voigt notation the deviatoric stress is represented as $s = [s_{11}, s_{22}, s_{33}, 2 * s_{23}, 2 * s_{13}, 2 * s_{12}]$. Based on the `RotateMatrix3D()` function of Elmer (in the *Stress.F90* file), I understand that in Elmer the deviatoric stress is represented as $s = [s_{11}, s_{22}, s_{33}, s_{12}, s_{23}, s_{31}]$. I'm following this convention.

2 Structure of the Code

I modified the ParStokes to prepare the way for the implementation of tensorial viscosities. The modifications done to *ParStokes.F90* are:

- Removal of the `EffectiveViscosity()` function. It was replaced by the `AnisotropicEffectiveViscosity()` function that returns a 6x6 viscosity matrix.
- If `SkipPowerLaw` is true, the `AnisotropicEffectiveViscosity()` is skipped and an isotropic viscosity tensor is constructed on the fly.
- Once the viscosity tensor is created, the isotropic part is computed. For that, I took the first invariant of the fourth order tensor following Eq 3.10 of Betten (1982). That is to say `mu_iso` $= \mu_{ijij}/9.0$, with μ_{ijkl} the (3x3x3x3) representation of the viscosity. the normalization by 9.0 is used so that `mu_iso` matches the standard viscosity in the case of an isotropic viscosity tensor.
- For now the assembly of the viscosity term in the stiffness matrix is done with `mu_iso`. It should be changed to use the full viscosity matrix.
- The Velocity block preconditioning is deactivated. I guess it could be implemented if necessary.
- The Schur complement is computed as `M/mu_iso`.

The `AnisotropicEffectiveViscosity()` function is stored in a Module file called *AnisotropicMaterialModels.F90*. The idea is to (i) stick to the formalism and structure used in the standard ParStokes solver, and (ii) to be able to easily implemented new anisotropic viscosity models if needed. The *AnisotropicMaterialModels* module has four functions/subroutines:

- `AnisotropicEffectiveViscosity()`: Returns the matrix form (6x6) of the anisotropic viscosity. Similarly to `EffectiveViscosity()`, it uses a ViscosityFlag to distinguish between different viscosity models. For now only a non-linear transverse orthotropic material is implemented, following Gagliardini and Meyssonnier (1999).
- `SecondInvariant()`: returns the second invariant of the strain rate tensor, to be used in the case of a non-linear viscosity model. Directly copied from the existing *MaterialModels.F90* file.
- `RotateMatrix3D()`: rotates the viscosity matrix based on a transformation matrix. Directly copied from Stress.F90. The transformation matrix is based on Material Unit Vectors, similarly to the Stress Solver.
- `Rotate4IndexTensor()`: helper function for `RotateMatrix3D()`. Directly copied from Stress.F90.

3 References

- J. Betten, Integrity basis for a second-order and a fourth-order tensor, Int. J. Math. Math. Sci., 5, 1, 87-96, 1982
- O. Gagliardini, and J. Meyssonnier, Analytical derivations for the behavior and fabric evolution of a linear orthotropic ice polycrystal, J. Geophys. Res. Solid Earth, 104, B8, 17797-17809, 1999