Implementation of the Non-linear $\mathrm{uw}p$-Form into FEATFLOW2 (Code Debugging)

Abdulrahman Obaid
(Dated: November 6, 2013)

1. ANALYTICAL SIMULATION

Ignoring the body load , the set of PDEs of the $\mathbf{uw}p$-form are written below

$$\left[\rho^S + \rho^F\right] (\mathbf{v}_S)'_S + \rho^{FR}(\mathbf{w})'_S - \operatorname{div} \mathbf{T}^S_E + \operatorname{grad} p + \rho^{FR}\mathbf{w} \cdot \operatorname{div} \mathbf{v}_S = \mathbf{f_u} \tag{1}$$

$$\rho^{FR}(\mathbf{v}_S)'_S + \frac{\rho^{FR}}{n^F}(\mathbf{w})'_S + \frac{\rho^{FR}}{n^F}\mathbf{w} \cdot \operatorname{div} \mathbf{v}_S + \operatorname{grad} p + \frac{\gamma^{FR}}{k^F}\mathbf{w} = \mathbf{f_w} \tag{2}$$

$$\operatorname{div} \mathbf{w} + \operatorname{div} \mathbf{v}_S = 0 \tag{3}$$

$$(\mathbf{u}_S)'_S = \mathbf{v}_S\,, \tag{4}$$

where

$$\begin{aligned}
n^S &= n^S_0\Big[1 - \operatorname{div} \mathbf{u}_S + \big|\operatorname{grad} \mathbf{u}_S\big|\Big] \\
n^F &= 1 - n^S \\
\rho^F &= n^F\,\rho^{FR} \\
\rho^S &= n^S\,\rho^{SR} \\
k^F &= k^F_0\left[\frac{n^F}{n^F_0}\right]^{\kappa}
\end{aligned} \tag{5}$$

and n^S_0, n^F_0, ρ^{FR} , ρ^{SR} and k^F_0 are constants given in the input data file. Remark that, we must pick analytical solutions that produce bounded n^F (and $1/n^F$) so that our bilinear forms are bounded. In addition, the resulting RHS functions must be bounded. This is necessary to have a well-posed problem which further requires that the constants are selected carefully so that the coercivity is fulfilled. Observe that n^F is not constant . Thus, the differential operator generating these partial differential equations is not self-adjoint operator.

A working candidate for the analytical simulation is the following set of functions and constants:

$$\begin{aligned}
u_{S1} &= 0.05x^2 \\
u_{S2} &= 0.05y^2 - 0.1y \\
w_1 &= x \\
w_2 &= -y \\
p &= 0.5 - x \\
\\
\lambda &= 1000 \\
\mu &= 1000 \\
\rho^{FR} &= 1 \\
\rho^{SR} &= 1 \\
k_0^F &= 0.0625 \\
n_0^F &= 0.25 \\
n_0^S &= 0.75 \\
\gamma^{FR} &= 10.0
\end{aligned}$$

which produce the following RHS functions:

$$\begin{aligned}
f_{u1} &= -301 \\
f_{u2} &= -300 \\
f_{w1} &= \frac{+16000x}{33x + 30y - 3xy + 70} - 1 \\
f_{w1} &= \frac{-16000y}{33x + 30y - 3xy + 70}
\end{aligned}$$

and

$$n^F = \frac{33}{400}x + \frac{3}{40}y - \frac{3}{400}xy + \frac{7}{40} \; .$$

Since the simulation was performed on the unit square domain (i.e., $[0\ 1] \times [0\ 1]$), $\mathbf{f_u}$, $\mathbf{f_w}$ and the bilinear forms are bounded.

A stationary problem was solved. Hence, the red terms were dropped out. Q2/P1 element was adopted and the results for level 1 is given below

```
-------------------------------------------------------------------------------
 IT  REL_US    REL_VS    REL_VF    RELP      DEF-US    DEF-VS    DEF-W     DEF-DIV  DEF-TOT  RHONL     OMEGNL    RHOMG
-------------------------------------------------------------------------------
  0                                          3.36E+00 0.00E+00 8.77E-01 2.37E-01 3.48E+00
-------------------------------------------------------------------------------
  1  7.55E-01 0.00E+00 5.00E-01 1.00E+00 2.77E-16 0.00E+00 9.08E-02 2.28E-17 9.08E-02 2.61E-02 1.00E+00 0.00E+00
  2  5.50E-03 0.00E+00 6.95E-17 6.98E+00 1.97E-16 0.00E+00 3.70E-04 2.83E-18 3.70E-04 1.03E-02 1.00E+00 0.00E+00
  3  2.18E-05 0.00E+00 7.79E-18 2.79E-02 6.04E-17 0.00E+00 1.32E-06 2.83E-18 1.32E-06 7.24E-03 1.00E+00 0.00E+00
  4  8.01E-08 0.00E+00 7.81E-18 1.03E-04 2.62E-17 0.00E+00 5.47E-09 2.83E-18 5.47E-09 6.30E-03 1.00E+00 0.00E+00
  5  3.21E-10 0.00E+00 7.81E-18 4.11E-07 2.14E-16 0.00E+00 1.93E-11 2.83E-18 1.93E-11 5.61E-03 1.00E+00 0.00E+00
  6  1.17E-12 0.00E+00 7.81E-18 1.50E-09 2.08E-16 0.00E+00 8.08E-14 2.83E-18 8.08E-14 5.34E-03 1.00E+00 0.00E+00
  7  4.68E-15 0.00E+00 7.81E-18 6.04E-12 1.44E-16 0.00E+00 2.93E-16 2.83E-18 3.27E-16 5.13E-03 1.00E+00 0.00E+00

Nonlinear solver statistics
---------------------------
Initial defect: 3.481557211095358E+00
Final defect:  3.268550894893278E-16
#Iterations:   7

Error Analysis
--------------
||uS-reference||_L2 =    1.289941E-17
||vS-reference||_L2 =    0.000000E+00
||w-reference||_L2  =    1.938438E-16
||p-reference||_L2  =    7.220392E-15
||uS-reference||_H1 =    3.281386E-17
```

```
||vS-reference||_H1 =    0.000000E+00
||w-reference||_H1  =    2.708191E-16
||p-reference||_H1  =    2.499474E-14

Remaining collection statistics:
--------------------------------

[]

Total time:                              0.0890000000
Time for initial mesh generation:        0.0040000001
Time for initial matrix assembly:        0.0000000000
Total time for grid generation:          0.0080000001
Total time for complete solver:          0.0129999999
Total time for nonlinear solver:         0.0120000000
Total time for defect calculation:       0.0059999999
Total time for optimal damping:          0.0000000000
Total time for RHS assembly:             0.0000000000
Total time for matrix assembly:          0.0020000000
Total time for linear solver:            0.0000000000
Total time for factorisation:            0.0040000001
Total time for postprocessing:           0.0110000001
Total #iterations nonlinear solver:      7
Total #iterations linear solver:         7
Total number of calculated timesteps:    0
```

furthermore, A non-stationary simulation (in which the red terms are multiplied with constant factors) converges to the same solution. However, BE reaches the stationary state much faster the CN. That is fine. What is the next step? Answer:

- including the multgrid results for uwp-solver in the report.
- implementing and validating the red terms into FEATFLOW2.
- implementing the updated lagrangian formulation (UL) into FEATFLOW2
- Asking Prof. Turek to get the expression for the rate-independent hyper-elastic material suitable for porous media from Prof. Bernd Markert. **I already asked Markert 2 weeks ago but he did reply. I would be thankful if you could kindly remind him? please inform him not to pass our report to third party before we publish it.**
- Compare with Markert
- Implementing the full newton method and compare it with the Picard iteration