Erasmus Mundus Masters In computational Mechanics

Finite Element Methods In Fluid

Assignment

2017-18

Aditya Mangaonkar

1. TRANSPORT PROBLEM

The actin filaments and monomers densities (F and G) are modelled by the following coupled system of partial differential equations

$$\begin{cases} \dfrac{\partial F}{\partial t} = -\boldsymbol{u} \cdot \boldsymbol{\nabla} F + D_F \nabla^2 F - \sigma_F F & \text{in } (0,T) \times \Omega \text{ in} \\ \dfrac{\partial G}{\partial t} = D_G \nabla^2 G - \sigma_G G + \hat{\sigma}_{GF} F & (0,T) \times \Omega \end{cases}$$

where u is the fluid velocity and the following material parameters are used

$D_F = 5\ \mu\text{m/s}$ $\sigma_F = 0.25\ \text{s}^{-1}$

$D_G = 15\ \mu\text{m/s}$ $\sigma_G = 2\ \text{s}^{-1}$ $\hat{\sigma}_{GF} = 0.5\ \text{s}^{-1}$

The equations are completed with the following boundary conditions:

- The filament density is constant at the upper boundary: $F(r = 25) = 80\ \mu$M
- No flux boundary conditions are considered for F everywhere else and for G on the entire boundary

Consider a velocity field $u(x,y) = -1/1500(rx,ry)\ \mu$m/s, where (x,y) are the points coordinates and $r = \sqrt{x^2 + y^2}$

ANS.

The above problem explains space and time dependant partial differential equation involves a double discretization. The solution of the problem involves spatial and the temporal descritization. Space and time are linked through the characteristics. The discretization of one influences the discretization of the other. An accurate spatial discretization can be eroded when transported in time if the time integration method cannot propagate the information along the directions prescribed by the convection term. Space discretization:

1)Finite element method

2)Time discretization: two classes of methods

• Based on the characteristics

• Based on standard coordinate system and use time-stepping algorithms.

For the solution of the above problem, we are going to use, θ family method in which Crank Nicolson is the best option as it is second order method ($\theta = 1/2$) which is implicit and unconditionally stable.

$$\frac{\partial \mathrm{F}}{\partial t} = -\mathrm{u}.\, \nabla \mathrm{F} + \mathrm{D_f} \nabla^2 \mathrm{F} - \sigma \mathrm{F}$$

$$\frac{\Delta F}{\Delta t} - \theta \Delta F t = F_t$$

$$\frac{F^{n+1} - F^n}{\Delta t} - \theta[(-\mathbf{u}.\nabla F^{n+1} + \mathbf{D}_F \nabla^2 F^{n+1} - \sigma_F F^{n+1}) - (-\mathbf{u}.\nabla F^n + \mathbf{D}_F \nabla^2 F^n - \sigma_F F^n)] = (-u.\nabla F^n + D_F \nabla^2 F^n - \sigma_F F^n)$$

Implementing Crank-Nicolson ($\theta = 1/2$)

$$\frac{F^{n+1} - F^n}{\Delta t} - \frac{1}{2}[(-\mathbf{u}.\nabla F^{n+1} + \mathbf{D}_F \nabla^2 F^{n+1} - \sigma_F F^{n+1}) - (-\mathbf{u}.\nabla F^n + \mathbf{D}_F \nabla^2 F^n - \sigma_F F^n)] = (-u.\nabla F^n + D_F \nabla^2 F^n - \sigma_F F^n)$$

$$\frac{\Delta F}{\Delta t} - \frac{1}{2}(-\mathbf{u}.\nabla \Delta F + \mathbf{D}_F \nabla^2 \Delta F - \sigma_F \Delta F) = (-u.\nabla F^n + D_F \nabla^2 F^n - \sigma_F F^n)$$

This equation represents the strong form of the equation in Galerkin form, to convert it into weak form we multiply each term with weighted function '***w***' and applying boundary conditions, we get following results .

$$\left(w, \frac{\Delta F}{\Delta t}\right) + \frac{1}{2}[C(u; w, \Delta F) + a(w, \Delta F) + (w, \sigma_F \Delta F) = -[C(u; w, \nabla F^n) + a(w, F^n + (w, \sigma_F F^n)]$$

Similar formulation is used to solve the equation no. (2) as follows

$$\frac{\partial G}{\partial t} = \mathrm{D_G} \nabla^2 \mathrm{G} - \sigma_{\mathrm{G}} \mathrm{G} + \sigma_{\mathrm{GF}} \mathrm{F}$$

$$\frac{\mathbf{G}^{\mathbf{n+1}} - \mathbf{G}^{\mathbf{n+1}}}{\Delta t} - \theta\left[\left(-\sigma_G.\nabla \mathbf{G}^{\mathbf{n+1}} + \mathbf{D}_G \nabla^2 \mathbf{G}^{\mathbf{n+1}} + \sigma_{GF} \mathbf{G}^{\mathbf{n+1}}\right) - \left(-\sigma_G\, \nabla \mathbf{G}^{\mathbf{n}} + \mathbf{D}_G \nabla^2 \mathbf{G}^{\mathbf{n}} + \sigma_{GF} \mathbf{G}^{\mathbf{n}}\right)\right] = \left(-\sigma_G\, \nabla \mathbf{G}^{\mathbf{n+1}} + D_G \nabla^2 \mathbf{G}^{\mathbf{n+1}} + \sigma_{GF} \mathbf{G}^{\mathbf{n+1}}\right)$$

Implementing Crank-Nicolson ($\theta = 1/2$)

$$\frac{\boldsymbol{G^{n+1} - G^{n+1}}}{\boldsymbol{\Delta t}} - \frac{\boldsymbol{1}}{\boldsymbol{2}}[(-\boldsymbol{\sigma_G} \, \boldsymbol{\nabla G^{n+1}} + \boldsymbol{D_G \nabla^2 G^{n+1}} - \boldsymbol{\sigma_{GF} G^{n+1}}) - (-\boldsymbol{\sigma_G} \, \boldsymbol{\nabla G^{n}} + \boldsymbol{D_G \nabla^2 G^{n}} + \boldsymbol{\sigma_{GF} G^{n}})]$$

$$= (-\boldsymbol{\sigma_G} \, \boldsymbol{\nabla G^{n}} + \boldsymbol{D_G \nabla^2 G^{n}} + \boldsymbol{\sigma_{GF} G^{n}})$$

$$\frac{\boldsymbol{\Delta G}}{\boldsymbol{\Delta t}} - \frac{1}{2}(-\boldsymbol{\sigma_G} \, \boldsymbol{\nabla \Delta G} + \mathbf{D}_{\boldsymbol{G}} \boldsymbol{\nabla^2 \Delta G} + \boldsymbol{\sigma_{GF} \Delta G}) = (-\boldsymbol{\sigma_G} \, \boldsymbol{\nabla G^{n}} + \mathbf{D}_{\boldsymbol{G}} \boldsymbol{\nabla^2 G^{n}} + \boldsymbol{\sigma_{GF} G^{n}})$$

This equation represents the strong form of the equation in Galerkin form, to convert it into weak form we multiply each term with weighted function '***w***' and applying boundary conditions, we get following results.

$$\left(\boldsymbol{w}, \frac{\boldsymbol{\Delta G}}{\boldsymbol{\Delta t}}\right) + \frac{\boldsymbol{1}}{\boldsymbol{2}}[\boldsymbol{a}(\boldsymbol{w}\,, \boldsymbol{\Delta G}) + (\boldsymbol{w}, \boldsymbol{\sigma_G \Delta G})] = -[(\boldsymbol{a}(\boldsymbol{w}, \boldsymbol{G^n}) + (\boldsymbol{w}, \boldsymbol{\sigma_G G^n}) + + \left(\boldsymbol{w}, \frac{\boldsymbol{1}}{\boldsymbol{2}} \boldsymbol{\sigma_{GF} F^n}\right) = \left(\boldsymbol{w}, \frac{\boldsymbol{1}}{\boldsymbol{2}} \boldsymbol{\sigma_{GF} F^{n+1}}\right)]$$

How it works at the background of the programme?

The input parameters are provided by the inputs as follows

```
D_F=5;
sigma_F=0.25;
D_G=15;
sigma_G=2;
sigma_GF=0.5;

% GEOMETRY
Nr=20;
Ntheta=20;
[X,T]= createMesh(Nr,Ntheta);
numnp = size(X,1);
```

The convection velocity is provided in the problem parameters. Spatial discretization is performed first using linear finite elements Then, transient response is computed using a time-stepping algorithm.

Crank Nicolson scheme is used for time descritization,4 Gause points method is used for time integration, the shape function and mass, convection matrix is called boundary value and mesh are given which are applied on boundary. Using Lagrange's multiplication the equations are solved and the values of ***F*** and ***G*** are plotted across the domain.

Variation of G value over the domain:

The output shows exact similar results as we have applied the boundary conditions below. As the mesh is courser is the first figure the graph also has edges on the other hand figure gets finer as the mesh is improved .

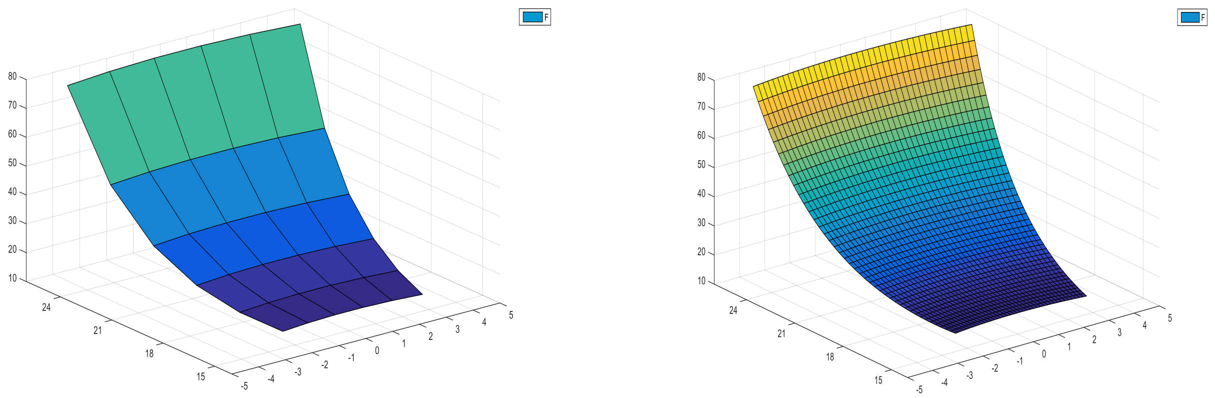

Figure 1 Mesh 50×50 And 40 ×40 F value over the domain

Variation of G value over the domain:

The following figure shows the variation of G value over the domain and the results are satisfactory as per the boundary conditions. It is denser in bottom side as compared to upper size as per results

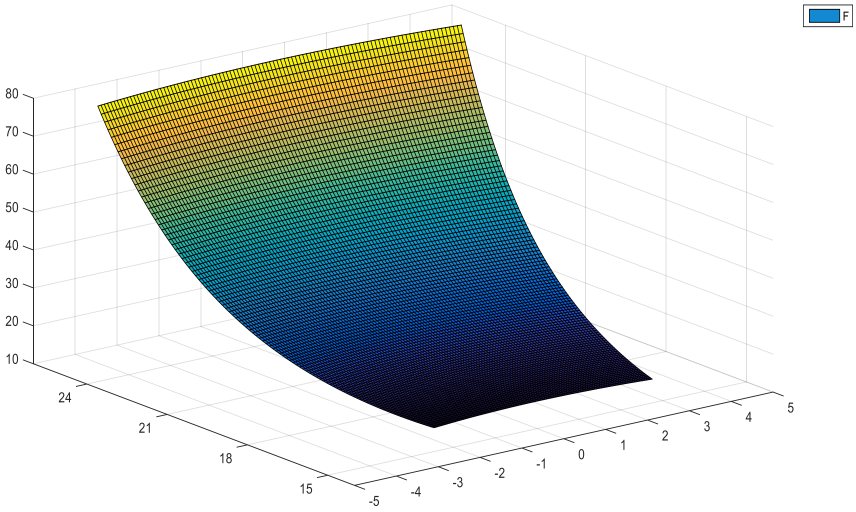

Figure 2 mesh 100 ×100 F value over the domain

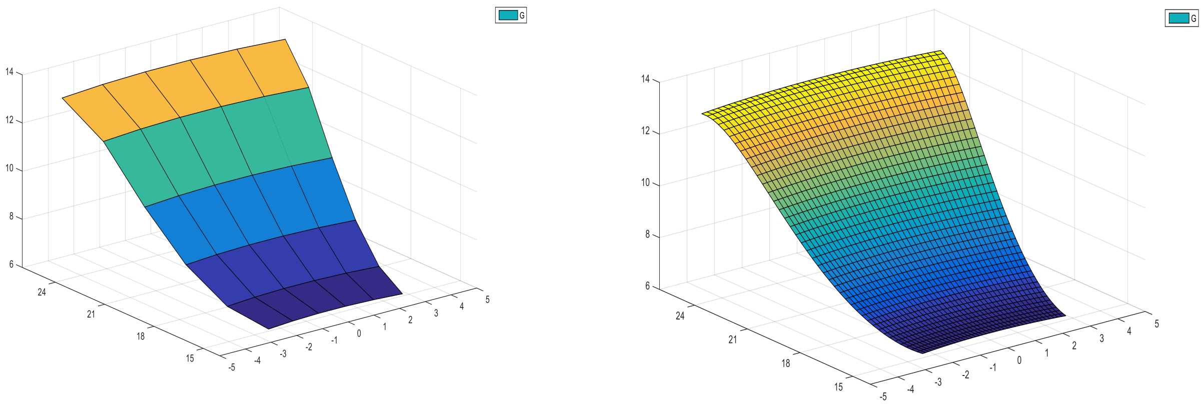

Figure 3 Mesh 5×5 And 400×40 G value over the domain

The Crank Nicolson method is unconditionally stable hence proves here, the results which we have got are satisfactory and there is no discontinuity. But the accuracy is very poor at lover mesh size proves here.

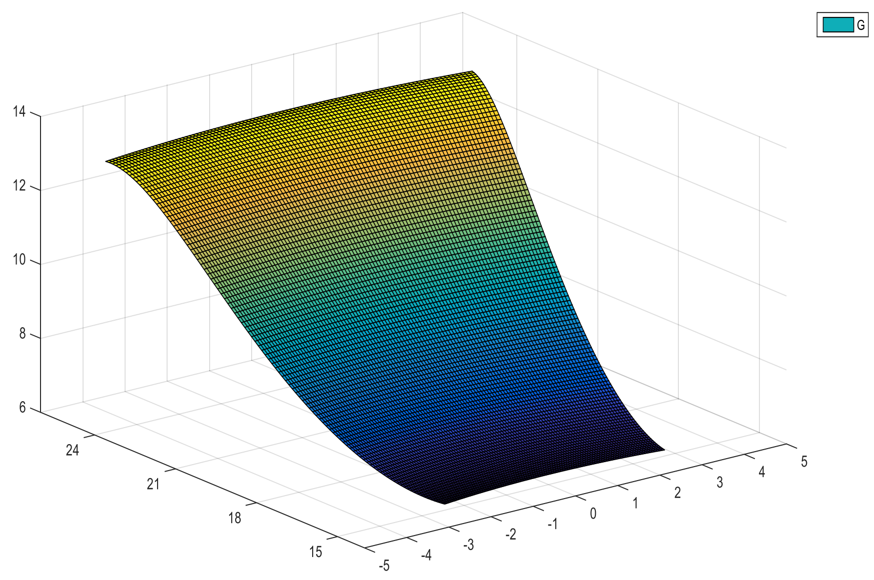

Figure 4 Mesh 100×100 G value over the domain

2. STOKES PROBLEM

Solve a Stokes problem

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \quad \text{in } \Omega$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega$$

with prescribed velocity at r = 15 and r = 25

$$u_r(r = 15) = -0.15, \;\; u_\theta(r = 15) = 0$$

$$u_r(r = 25) = -0.30, \;\; u_\theta(r = 25) = 0$$

and zero traction on the straight sides of the boundary. Consider a viscosity $\mathbf{v = 10^3\ pN \cdot s/\mu m.}$

ANS.

After neglecting the convective term and time dependency term from Navier stokes equation, we get Stokes problem which is the above question. There are two formulations of stokes problem, in first the the formulation is expressed in terms of pressure and velocity.

But here the other formulation is used where the formulation uses Cauchy tress. The advantage of this approach is that it can readily treat problems with fluid constitutive equations more general than the linear Stokes' law.

In differential form, a steady Stokes problem is stated as follows in terms of Cauchy stress: given the body force **0**.

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \quad \text{in } \Omega \qquad \text{(equilibrium)},$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \qquad \text{(incompressibility)}$$

The weak form obtained multiplying by the velocity test function ***w*** and integrating by parts the stress term. Similarly, the incompressibility condition is multiplied by the pressure test function ***q*** and the result integrated over the computational domain.

$$\int_{\Omega} \nabla \boldsymbol{w} : \boldsymbol{\sigma} \, d\Omega = \mathbf{0}$$

$$\int_{\Omega} q \nabla . \boldsymbol{v} d\Omega = \mathbf{0}$$

$$a(\boldsymbol{w}, \boldsymbol{v}) = \int_{\Omega} \boldsymbol{w}_{ij,} \boldsymbol{S}_{ij} \; d\Omega$$

$$b(v, q) = \int_{\Omega} q\nabla . v d\Omega$$

The mesh is created using provided code which is of Q1Q1 element mesh. The size of the mesh is taken 3 times over the domain. The velocity and pressure profile obtained over the domain is explained below. The X and Y component of velocity is also plotted in code and is shown below.

For stabilization of the equation, the Galerkin Least Square method is used. Which has stabilized the equation and can be seen in the results below.

Figure 5 Velocity Streamline profile of mesh 5× 5 and 40× 40

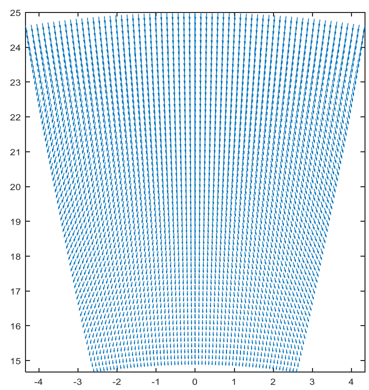

Figure 6 Velocity Streamline profile of mesh 60× 60

The streamline profile is totally disturbing in lower mesh but it is very much perfect at higher grid but also it takes lot of time to solve so we can say that the higher grid size may be expensive.

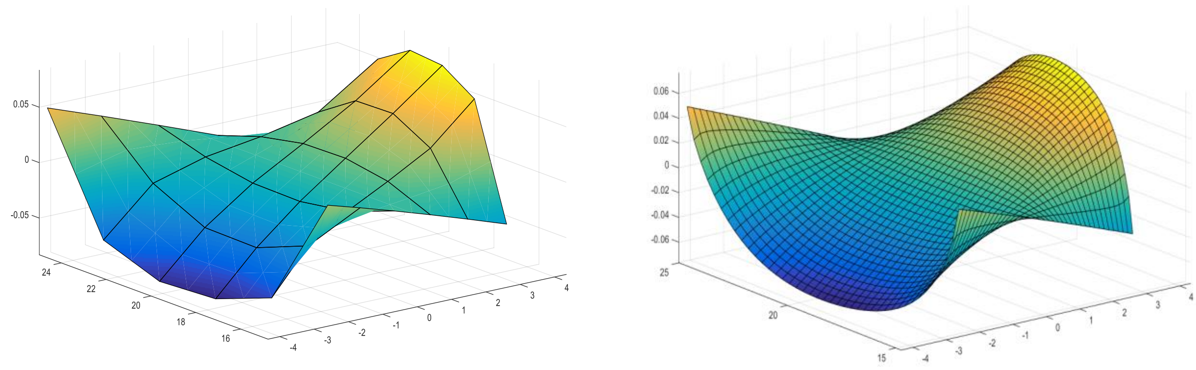

Figure 7 Velocity X component profile of mesh 5× 5 and 40× 40

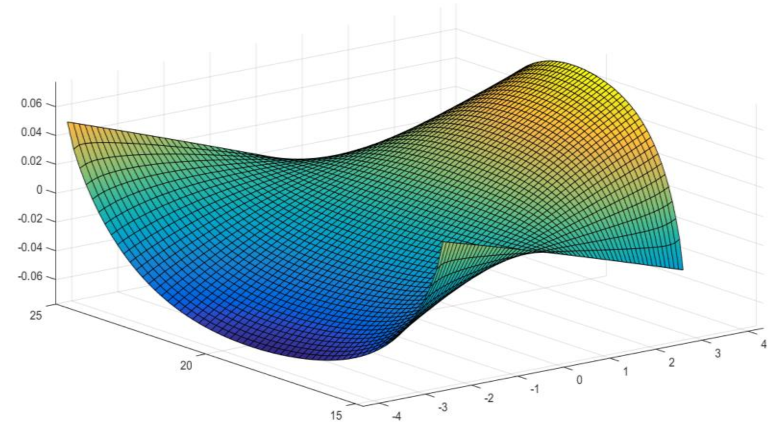

Figure 8 Velocity X component profile of mesh 60× 60

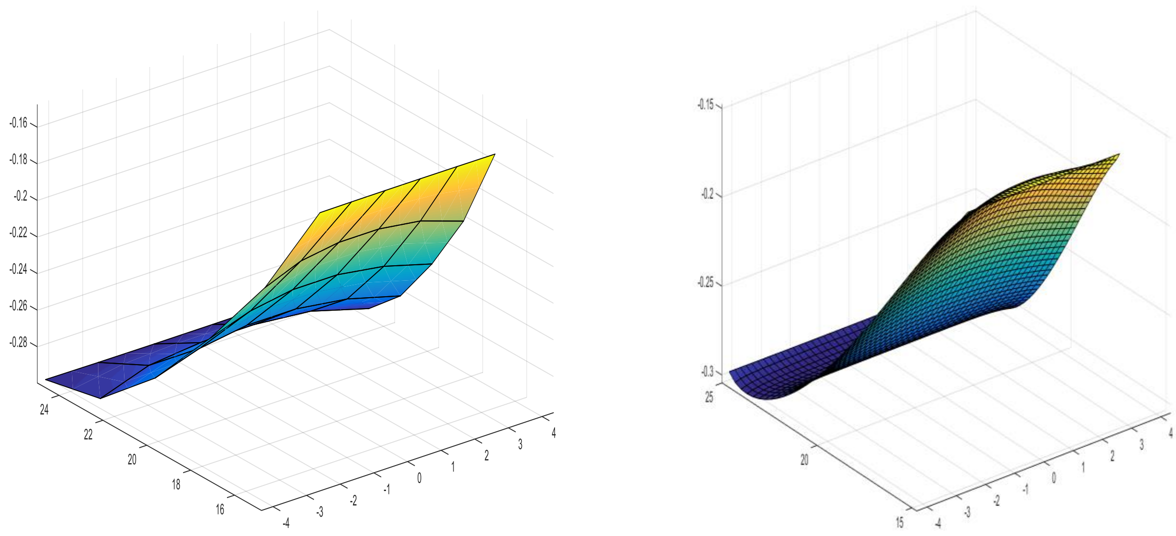

Figure 9 Velocity Y component profile of mesh 5× 5 and 40× 40

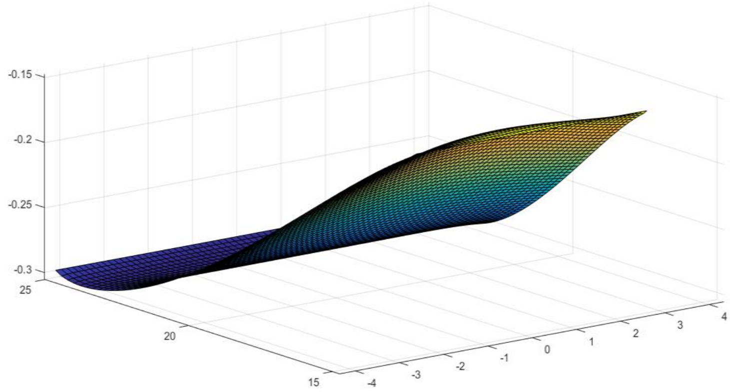

Figure 10 Velocity Y component profile of mesh 60× 60

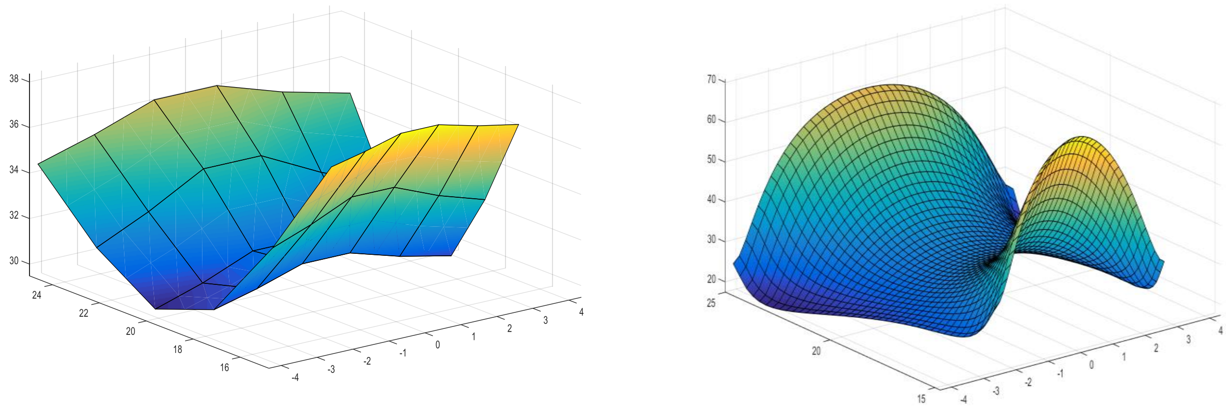

Figure 11 Pressure profile of mesh 5× 5 and 40× 40

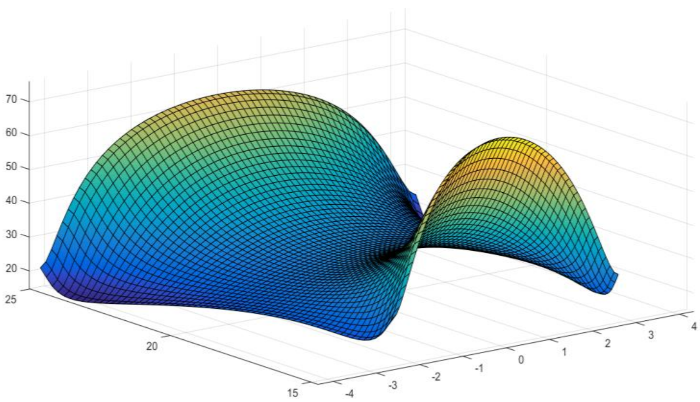

Figure 12 Pressure profile of mesh 60× 60

The pressure profile is pretty much similar at both the higher grid but it is very disappointing at lower grid though we have used G.L.S. method. But fine grid results are satisfactory so we can say the results are up to the mark

3. COUPLED PROBLEM

The equation describing the evolution of monomers densities G does not involve any convective transport and, therefore, only the fluid around the fibers has to be considered. This fluid is modelled using the equations of a quasi-steady viscous fluid. Moreover, due to the presence of actin fibers, the incompressibility constrain is dropped and pressure is neglected. Then, the equations governing the coupled problem can be written as

$$\begin{cases} \nu\nabla\cdot(\nabla^s \boldsymbol{u}) + \nabla\cdot\boldsymbol{\sigma}_m(F) + \mathbf{T}_m(\boldsymbol{u}) = \mathbf{0} & \text{in } (0,T)\times\Omega \\ \dfrac{\partial F}{\partial t} = -\boldsymbol{u}\cdot\nabla F + D_F\nabla^2 F - \sigma_F F & \text{in } (0,T)\times\Omega \\ \dfrac{\partial G}{\partial t} = D_G\nabla^2 G - \sigma_G G + \hat{\sigma}_{GF} F & \text{in } (0,T)\times\Omega \end{cases}$$

where $\nabla\cdot\sigma_m$ and $\mathbf{T}_m$ are surface forces on the leading edge.

Solve the proposed problem with the parameters and boundary conditions described in the previous points.

In order to treat the boundary terms, you can use the function boundaryMatrices.m, that builds the matrices arising from the discretisation of these two terms. Note that this function assumes that the same interpolation degree is used for u and F.

Remark. The first equation in the system does not include any time derivatives and it can be evaluated at any time. Note, however, that our goal is to find the solution of the system at t^{n+1} and, therefore, it is convenient to evaluate the equation at this instant of time. Otherwise, we need to make sure that the initial condition verifies the equation.

ANS.

Given equation

$$\vartheta\,\nabla\cdot(\nabla^s u) + \nabla\cdot\sigma m\,(F) + Tm(u) = 0 \qquad (1)$$

$$\frac{\partial F}{\partial t} = -u.\ \nabla F + D_f\nabla^2 F - \sigma F \qquad (2)$$

$$\frac{\partial G}{\partial t} = D_G\nabla^2 G - \sigma_G G + \sigma_{GF} F \qquad (3)$$

In the above equations explains form of Navier stokes equation in which first equation contains 2 variables which are explained in the partial differential form in other two equations.

The first equation in the system does not include any time derivatives and it can be evaluated at any time. Our goal is to find the solution of the system at t^{n+1} and, therefore, it is convenient to evaluate the equation at this instant of time. Otherwise, we need to make sure that the initial condition verifies the equation.Therfore considering equation at time t^{n+1} yields equation;

$$\vartheta \nabla \cdot (\nabla^s u^{n+1}) + \nabla \cdot \sigma m\, (F^{n+1}) + Tm(u^{n+1}) = 0$$

Multiplying the equation by weighting function w and integrating by parts ,

$$(\mathbf{w}, \vartheta\nabla \cdot (\nabla^s \mathbf{u}^{n+1})) + (\mathbf{w}, \nabla \cdot \sigma_m (F^{n+1})) + (\mathbf{w}, T_m(u^{n+1})) = 0$$

$$(\mathbf{w}, \vartheta\nabla \cdot (\nabla^s \mathbf{u}^{n+1})) = \qquad (\mathbf{K_s}\, \mathbf{u}^{n+1}))$$

$$(\mathbf{w}, \nabla \cdot \sigma_m\ (F^{n+1})) = \qquad (\mathbf{T_F}\ (F^{n+1}))$$

$$(\mathbf{w}, T_m\ (u^{n+1})) = \qquad (\mathbf{T_m}\ (u^{n+1}))$$

The equation becomes;

$$(\mathbf{K_s}\ \mathbf{un} + \mathbf{1})) + (\mathbf{T_F}\ (F^{n+1})) + (\mathbf{T_m}\ (u^{n+1})) = 0 \qquad \textbf{(i)}$$

For solving equation 2 and 3, we use Crank Nicolson scheme as follows,

$$\frac{\partial \mathbf{F}}{\partial t} = \text{-u. } \nabla F + D_f \nabla^2 F - \boldsymbol{\sigma} F$$

$$\frac{\Delta \mathbf{F}}{\Delta \mathbf{t}} - \theta \boldsymbol{\Delta} \mathbf{Ft} = \mathbf{F_t}$$

$$\frac{\mathbf{F^{n+1}} - \mathbf{F^n}}{\Delta \mathbf{t}} - \theta[(-\mathbf{u}.\nabla \mathbf{F^{n+1}} + \mathbf{D_F}\nabla^2\mathbf{F^{n+1}} - \sigma_\mathbf{F}\mathbf{F^{n+1}}) - (-\mathbf{u}.\nabla\mathbf{F^n} + \mathbf{D_F}\nabla^2\mathbf{F^n} - \sigma_\mathbf{F}\mathbf{F^n})]$$
$$= (-\mathbf{u}.\nabla\mathbf{F^n} + \mathbf{D_F}\nabla^2\mathbf{F^n} - \sigma_\mathbf{F}\mathbf{F^n})$$

Implementing Crank-Nicolson ($\theta = 1/2$)

$$\frac{\Delta \mathbf{F}}{\Delta \mathbf{t}} - \frac{1}{2}\left(-\mathbf{u^{n+1}}.\nabla\Delta\mathbf{F} + \mathbf{D_F}\nabla^2\Delta\mathbf{F} - \sigma_\mathbf{F}\Delta\mathbf{F}\right) = (-\mathbf{u}.\nabla\mathbf{F^n} + \mathbf{D_F}\nabla^2\mathbf{F^n} - \sigma_\mathbf{F}\mathbf{F^n})$$

$$\frac{\mathbf{F^{n+1}} - \mathbf{F^n}}{\Delta \mathbf{t}} - \frac{\mathbf{1}}{\mathbf{2}}[(-\mathbf{u^{n+1}}.\nabla \mathbf{F^{n+1}} + \mathbf{D_F}\nabla^2\mathbf{F^{n+1}} - \sigma_\mathbf{F}\mathbf{F^{n+1}}) - (-\mathbf{u}.\nabla\mathbf{F^n} + \mathbf{D_F}\nabla^2\mathbf{F^n} - \sigma_\mathbf{F}\mathbf{F^n})]$$
$$= (-\mathbf{u^{n+1}}\mathbf{u}.\nabla\mathbf{F^n} + \mathbf{D_F}\nabla^2\mathbf{F^n} - \sigma_\mathbf{F}\mathbf{F^n})$$

$$\frac{\mathbf{F^{n+1}}}{\mathbf{\Delta t}} + \frac{1}{2}[(\mathbf{u^{n+1}}.\nabla\mathbf{F^{n+1}} - \mathbf{D_F}\nabla^2\mathbf{F^{n+1}} + \sigma_F\mathbf{F^{n+1}})] = \frac{1}{2}(-\mathbf{u}.\nabla\mathbf{F^n} + \mathbf{D_F}\nabla^2\mathbf{F^n} - \sigma_F\mathbf{F^n}) + \frac{\mathbf{F^n}}{\mathbf{\Delta t}}$$

$\frac{\mathbf{F^{n+1}}}{\mathbf{\Delta t}}$	$\frac{\mathbf{M}}{\mathbf{\Delta t}}(\mathbf{F^{n+1}})$
$\frac{1}{2}[(\mathbf{u^{n+1}}.\nabla\mathbf{F^{n+1}} - \mathbf{D_F}\nabla^2\mathbf{F^{n+1}} + \sigma_F\mathbf{F^{n+1}})]$	$\frac{1}{2}[\mathbf{C}(\mathbf{u^{n+1}},\mathbf{F^{n+1}}) - \mathbf{K_D}(\mathbf{F^{n+1}}) + \mathbf{M}(\mathbf{F^{n+1}})]$
$\frac{1}{2}(-\mathbf{u}.\nabla\mathbf{F^n} + \mathbf{D_F}\nabla^2\mathbf{F^n} - \sigma_F\mathbf{F^n})$	$-\frac{1}{2}[(-\mathbf{u},\mathbf{F^n}) + \mathbf{K_D}\mathbf{F^n} - \mathbf{M}(\mathbf{F^n})]$
$\frac{\mathbf{F^n}}{\mathbf{\Delta t}}$	$\frac{\mathbf{M}}{\mathbf{\Delta t}}(\mathbf{F^n})$

$$\frac{\mathbf{M}}{\mathbf{\Delta t}}(\mathbf{F^{n+1}}) + \frac{1}{2}[\mathbf{C}(\mathbf{u^{n+1}},\mathbf{F^{n+1}}) - \mathbf{K_D}(\mathbf{F^{n+1}}) + \mathbf{M}_F(\mathbf{F^{n+1}})] = \frac{1}{2}[C(-\mathbf{u^n},\mathbf{F^n}) + \mathbf{K_D}\mathbf{F^n} - \mathbf{M}(\mathbf{F^n})] + \frac{\mathbf{M}}{\mathbf{\Delta t}}(\mathbf{F^n}) \quad \text{...(ii)}$$

$$(\mathbf{K_s}\ \mathbf{un+1})) + (\mathbf{T_F}\ (\mathbf{F^{n+1}})) + (\mathbf{T_m}\ (\mathbf{u^{n+1}})) = 0 \qquad \text{(i)}$$

From equation (i) and (ii) we get following matrix

$$\begin{bmatrix} \mathbf{K_s} + \mathbf{T_m} & \mathbf{T_F} \\ 0 & \frac{1}{2}[C + \mathbf{K_D} + \mathbf{M}_F] + \frac{\mathbf{M}}{\mathbf{\Delta t}} \end{bmatrix} + \begin{bmatrix} u^{n+1} \\ F^{n+1} \end{bmatrix} = \begin{bmatrix} O \\ \frac{1}{2}[C(-u^n, F^n) + K_D F^n - M(F^n)] + \frac{M}{\Delta t}(F^n) \end{bmatrix}$$

The above equation has three unknowns which are $\boldsymbol{C}, \boldsymbol{u^{n+1}}, \boldsymbol{F^{n+1}}$ and we have only 2 equations. So to solve this equations we are going to use Picard method in which we are going to approximate the initial value till we get convergence and then after solving it we will get value of $\boldsymbol{F^{n+1}}$ **.**

Similar formulation as equation (1) is used to solve the equation no. (2) as follows

$$\frac{\partial G}{\partial t} = D_G \nabla^2 G - \sigma_G G + \sigma_{GF} F$$

$$\frac{\mathbf{G^{n+1} - G^{n+1}}}{\boldsymbol{\Delta t}} - \theta\left[\left(-\boldsymbol{\sigma_G}.\, \nabla \mathbf{G^{n+1}} + \mathbf{D}_{\boldsymbol{G}} \boldsymbol{\nabla}^2 \mathbf{G^{n+1}} + \boldsymbol{\sigma_{GF}} \mathbf{F^{n+1}}\right) - \left(-\boldsymbol{\sigma_G}\, \nabla \mathbf{G^n} + \mathbf{D}_{\boldsymbol{G}} \boldsymbol{\nabla}^2 \mathbf{G^n} + \boldsymbol{\sigma_{GF}} \mathbf{F^n}\right)\right]$$

$$= \left(-\boldsymbol{\sigma_G}\, \boldsymbol{\nabla} \mathbf{G^{n+1}} + \boldsymbol{D_G} \boldsymbol{\nabla}^2 \mathbf{G^{n+1}} + \boldsymbol{\sigma_{GF}} \mathbf{F^{n+1}}\right)$$

Implementing Crank-Nicolson ($\theta = 1/2$)

$$\frac{\boldsymbol{\Delta G}}{\boldsymbol{\Delta t}} = \frac{1}{2}\left(-\boldsymbol{\sigma_G}\, \boldsymbol{\nabla G^n} + \mathbf{D}_{\boldsymbol{G}} \boldsymbol{\nabla}^2 \boldsymbol{G^n} + \boldsymbol{\sigma_{GF} F^n}\right) + \frac{1}{2}\left(-\boldsymbol{\sigma_G}\, \nabla \boldsymbol{\Delta G} + \mathbf{D}_{\boldsymbol{G}} \boldsymbol{\nabla}^2 \boldsymbol{\Delta G} + \boldsymbol{\sigma_{GF} \Delta F}\right)$$

$$\frac{\boldsymbol{G^{n+1} - G^n}}{\boldsymbol{\Delta t}} - \frac{\mathbf{1}}{\mathbf{2}}\left[\left(-\boldsymbol{\sigma_G}\, \boldsymbol{\nabla G^{n+1}} + \boldsymbol{D_G} \boldsymbol{\nabla}^2 \boldsymbol{G^{n+1}} - \boldsymbol{\sigma_{GF} F^{n+1}}\right)\right] = \frac{\mathbf{1}}{\mathbf{2}}\left(-\boldsymbol{\sigma_G}\, \boldsymbol{\nabla G^n} + \boldsymbol{D_G} \boldsymbol{\nabla}^2 \boldsymbol{G^n} + \boldsymbol{\sigma_{GF} F^n}\right)$$

The only unknown which is not given in above equation is $\boldsymbol{F^{n+1}}$ which we found using Picard method. We found it from the above matrix above and we will solve the above equation using Crank-Nicolson method, for the value of $\boldsymbol{G^{n+1}}$

(As I come to the theoretical solution at last minute I cannot implement it in the code)