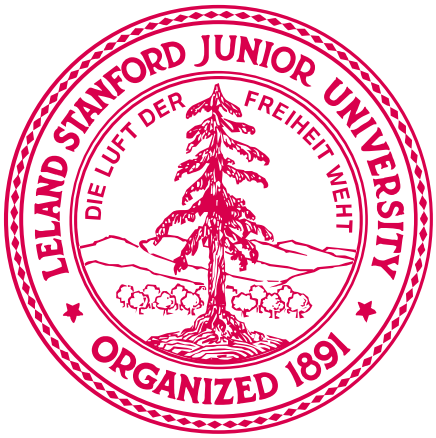
RANS simulation of a viscoelastic turbulent flow in a channel using the FENE-P model
ME269B Final Project



Vincent E. Terrapon
Stanford University

April 14, 2003

Abstract

The FENE-P model is used in a channel flow to investigate turbulent drag reduction. In opposite to previous works, a RANS calculation is performed. Different simple closure models and a wide range of parameters are tested. This investigation shows that these simple models are not accurate and cannot describe qualitatively the physics of drag reduction. Although these models fail to describe the polymer stress term, it is suspected that its accurate modelization is not enough and that the turbulence model needs also to be adapted since the action of the polymers is very localized and three dimensional.

1 Introduction

It has been experimentally shown for more than 50 years [?] that dilute solutions of long polymer chains can reduce the drag in a turbulent flow. However, an exact description of the mechanisms has still not been proposed. Many theories [?, ?, ?] have been developed. Some models have successfully predicted drag reduction [?, ?, ?, ?] and even reached the Virk [?] or maximum drag reduction asymptote. These simulations used the so-called FENE-P model in conjunction with DNS. The FENE-P model is an approximation of the FENE dumbbell, which represents the polymer as an entropic spring connecting two beads for the hydrodynamic forces. It envisages the polymers and the Newtonian fluid as a continuum. The equations are the usual Navier-Stokes equations slightly modified by an extra-term which accounts for the stress created by the polymers.

The goal of this project is to implement this system of equations in Fluent and perform a RANS simulation. It should investigate different closure models and the influence of the parameters.

2 Basic equations

The domain used is two-dimensional with two walls at the upper and lower boundaries. In the x-direction, the domain is periodic. All length are made dimensionless using the half height of the channel. The extent of the domain in the streamwise direction is then unity. Fig. 1 shows the mesh used.

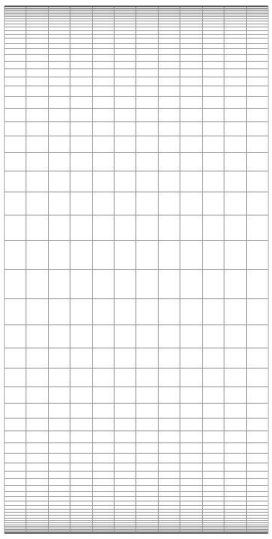


Figure 1: Mesh used for the calculation; streamwise direction from left to right, walls at the top and bottom

If the extra-stress created by the polymers is defined by T_{ij}, the modified Navier-Stokes

equations are

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\beta}{Re}\frac{\partial^2 U_i}{\partial x_j^2} + \frac{1-\beta}{Re}\frac{\partial T_{ij}}{\partial x_j} \tag{2}$$

where β represents the ratio of solvent η_s to the total viscosity η_t, and Re is the Reynolds number defined with the half height of the channel H, the total viscosity η_t and the bulk velocity U_{bulk} of the flow. The total viscosity is the sum of the solvant viscosity η_s and the polymer viscosity η_p. The polymeric stress tensor T_{ij} is related to a conformation tensor C_{ij}

$$T_{ij} = \frac{1}{We}\left(\frac{L^2}{L^2 - C_{ii}} C_{ij} - \delta_{ij}\right) \tag{3}$$

where We is the Weissenberg number which is the ratio of the characteristic time scale of the flow to the characteristic time scale of the polymer given by its relaxation time. The FENE and FENE-P models are improved versions of the Oldroyd-B, where a maximum extensibility has been introduced to bound the extension of the spring. In the above equation, the trace of the conformation tensor C_{ii} cannot extend over the parameter L^2. The conformation tensor is obtained by solving an evolution equation

$$\frac{\partial C_{ij}}{\partial t} + U_k \frac{\partial C_{ij}}{\partial x_k} - \left(C_{ik}\frac{\partial U_j}{\partial x_k} + C_{jk}\frac{\partial U_i}{\partial x_k}\right) - \kappa \frac{\partial^2 C_{ij}}{\partial x_k^2} = -T_{ij} \tag{4}$$

where the last term on the lefthand side corresponds to an artificial diffusion characterized by the parameter κ to guarantee the numerical stability of the scheme.

We can now take the average of the previous equations to get the Reynolds average equations

$$\frac{\partial \overline{U}_i}{\partial x_i} = 0 \tag{5}$$

$$\frac{\partial \overline{U}_i}{\partial t} + \overline{U}_j \frac{\partial \overline{U}_i}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\beta}{Re}\frac{\partial^2 \overline{U}_i}{\partial x_j^2} + \frac{1-\beta}{Re}\frac{\partial \overline{T}_{ij}}{\partial x_j} - \frac{\partial \overline{u_i u_j}}{\partial x_j} \tag{6}$$

$$\overline{T_{ij}} = \frac{1}{We}\left(\overline{\frac{L^2 C_{ij}}{L^2 - C_{ii}}} - \delta_{ij}\right) \tag{7}$$

$$\begin{aligned}\frac{\partial \overline{C}_{ij}}{\partial t} + \overline{U}_k \frac{\partial \overline{C}_{ij}}{\partial x_k} - \left(\overline{C}_{ik}\frac{\partial \overline{U}_j}{\partial x_k} + \overline{C}_{jk}\frac{\partial \overline{U}_i}{\partial x_k}\right) - \kappa \frac{\partial^2 \overline{C}_{ij}}{\partial x_k^2} + \frac{1}{We}\left(\overline{\frac{L^2 C_{ij}}{L^2 - C_{ii}}} - \delta_{ij}\right) \\ = -\frac{\partial \overline{u_k c_{ij}}}{\partial x_k} + \left(\overline{c_{ik}\frac{\partial u_j}{\partial x_k}} + \overline{c_{jk}\frac{\partial u_i}{\partial x_k}}\right)\end{aligned} \tag{8}$$

where the different variables have been split into their mean and the fluctuations from this mean

$$U_i = \overline{U}_i + u_i \tag{9}$$

$$P = \overline{P} + p \tag{10}$$

$$C_{ij} = \overline{C_{ij}} + c_{ij} \tag{11}$$

$$T_{ij} = \overline{T_{ij}} + \tau_{ij} \tag{12}$$

As expected, these equations are not closed. A model for the different new terms is necessary, namely the two last terms in Eq. 8, the average of the polymeric stress in Eq. 6 and the term containing the mean value of a fraction in Eq. 7 and 8. It is also important to notice that the Reynolds stress is also influenced by the polymers and should be adapted accordingly. The modeling of turbulence itself is done by using the standard Spalart-Allmaras model.

Three strategies of closure are investigated in this project. Due to a lack of time, only simple models will be tested and presented. Moreover, a wide range of parameters have been used to investigate their influence on the flow. Each solution is then compared to the Newtonian case which serves as reference. The Newtonian calculation is first presented, then the different closure approximations are discussed.

3 Newtonian flow

The RANS simulation of the channel flow is performed at two different Reynolds number to validate the turbuelent model. $Re = 5000$ and $Re = 10000$ are used in this case. Fig. 2 shows the velocity profile non-dimensionalized by u_τ as function of y^+ for both cases. It can be seen that both curves collapse on eachother. For all the calculations, the drag at the wall is calculated by using the pressure gradient, since all simulations are done at fixed mass flow. At $Re = 5000$, we get $\tau_w = 0.003542$ and $h^+ = 297.6$, whereas at $Re = 10000$, the stress at the wall is $\tau_w = 0.003025$ and $h^+ = 275.0$. The increase of the Reynolds number was performed by decreasing the viscosity by a factor of 2. These results are consistent with DNS data, showing that the turbulence model used is accurate enough for the Newtonian flow.

4 First model

As mentioned before, the turbulence is described by the Spalart-Allmaras model. The additional source term corresponding to the mean polymer stress (see Eq. 7) is approximated by

$$\overline{T_{ij}} = \frac{1}{We}\left(\frac{L^2}{L^2 - \overline{C_{ii}}}\overline{C_{ij}} - \delta_{ij}\right) \tag{13}$$

and $\overline{C_{ij}}$ is calculated from Eq. 8 neglecting the terms on the right-hand side and replacing again the mean of the fraction by the fraction of the means

$$\frac{\partial \overline{C}_{ij}}{\partial t} + \overline{U}_k\frac{\partial \overline{C}_{ij}}{\partial x_k} - \left(\overline{C}_{ik}\frac{\partial \overline{U}_j}{\partial x_k} + \overline{C}_{jk}\frac{\partial \overline{U}_i}{\partial x_k}\right) - \kappa\frac{\partial^2 \overline{C}_{ij}}{\partial x_k^2} + \frac{1}{We}\left(\frac{L^2}{L^2 - \overline{C}_{ii}}\overline{C}_{ij} - \delta_{ij}\right) = 0 \tag{14}$$

The artificial diffusivity κ was set to 0.0001 for all the runs in order to stabilize the numerical computation. Also the factor in front of the polymeric stress term in Eq. 6 is modified by adding the eddy viscosity μ_T to the total viscosity η_t giving $(1-\beta)\left(\frac{1}{Re} + \frac{\mu_T}{\rho}\right)$. This is motivated by the analogy with the Reynolds stress where a extra turbulent diffusion is added.

Different values of the concentration ratio β and of the Weissenberg number We have been simulated. In each case, the extensibility L was set to 60, which is a realistic value for relatively large polymers, and the Reynolds number to $Re = 5000$. Fig. 3 and 4 show

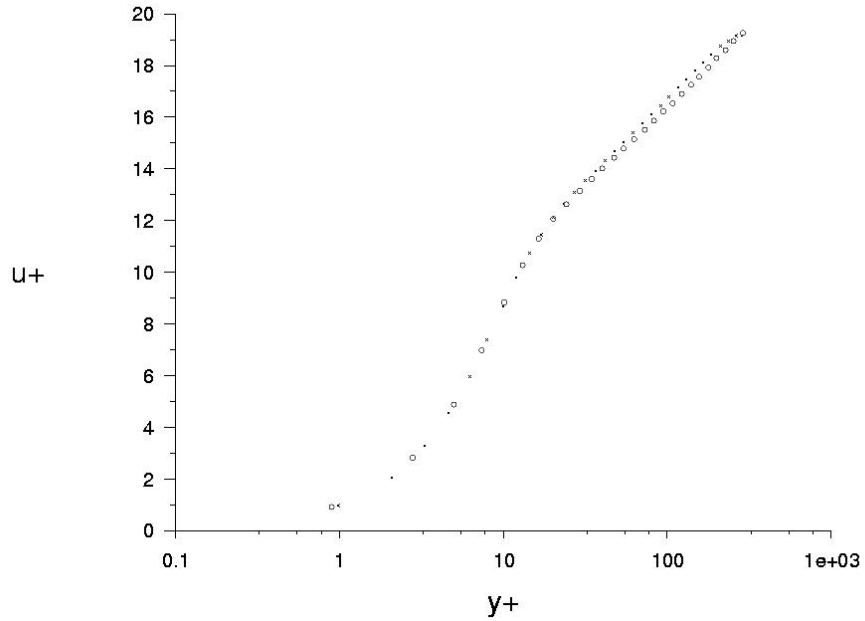


Figure 2: Velocity profile for the Newtonian case using the Spalart-Allmaras model; x: $Re = 5000$, o: $Re = 10000$

the results. It must be reminded that the pressure gradient in the Newtonian case for the same Reynolds number was $dP/dx = 0.003542$ (see section 3). Therefore this model does not give any drag reduction but in contrary a drag increase, since the pressure gradient has increased. Moreover, this drag increase is greater for small Weissenberg numbers and small viscosity ratios. For a really small concentration of polymers (large β) the drag is almost the same as for the Newtonian case.

Since the turbulence model used here is a standard Spalart-Allmaras model, the only contribution of the polymers come from T_{ij}. Fig. 5 shows the polymeric stress calculated with this model and Fig. 6 shows the velocity profile. It is interesting to notice that only one component of the stress, namely T_{xy} is non-zero in the streamwise direction. The stress has its maximum at the wall and decrease rapidly away from the wall. Results from DNS (see Fig. 10) show however that the maximum stress should be located at a small distance from the wall and not directly at the wall. This qualitative behaviour could explain the increase in drag found with this model. The velocity profile does not show either the expected shift up observed experimentally and numerically in drag reduced flow. All these results demonstrate that the model is not accurate.

5 Second model

The second approximation strategy is similar to the first model but replace the last term on the right-hand side of Eq. 8 by a conformation strain production term Λ_{ij} instead of

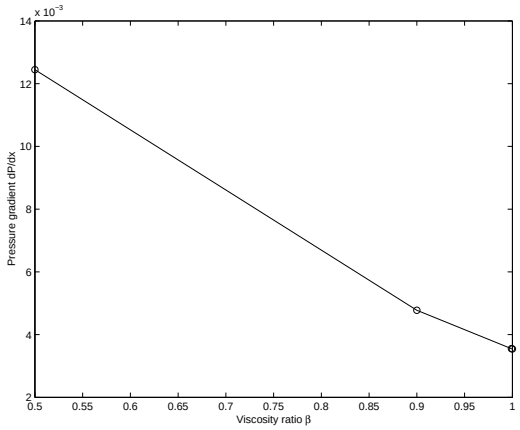


Figure 3: Variation of the drag as function of the viscosity parameter β for a fixed Weissenberg number $We = 2$

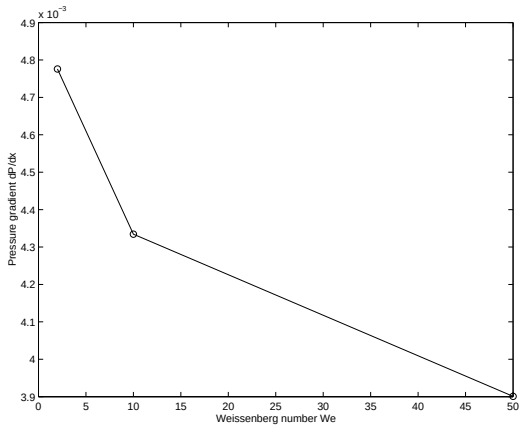


Figure 4: Variation of the drag as function of the Weissenberg number We for a fixed viscosity ratio $\beta = 0.9$

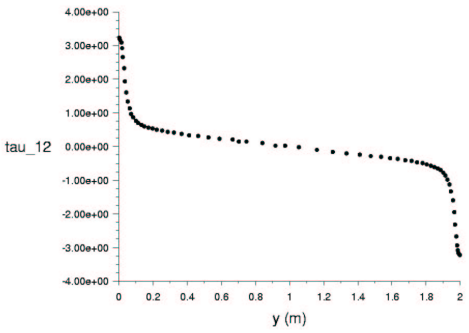


Figure 5: Polymer stress τ_{xy} for $\beta = 0.999$ and $We = 50$

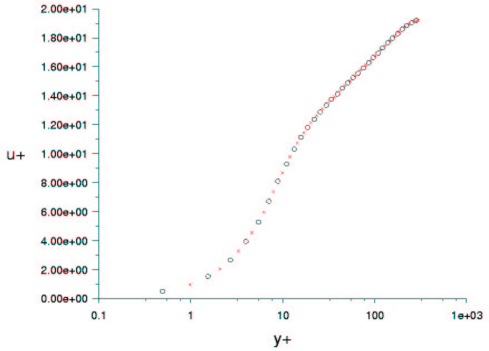


Figure 6: Velocity profile; o: $\beta = 0.999$ and $We = 50$, x: Newtonian ($\beta = 1$)

neglecting it

$$\frac{\partial \overline{C}_{ij}}{\partial t} + \overline{U}_k \frac{\partial \overline{C}_{ij}}{\partial x_k} - \left(\overline{C}_{ik} \frac{\partial \overline{U}_j}{\partial x_k} + \overline{C}_{jk} \frac{\partial \overline{U}_i}{\partial x_k} \right) - \kappa \frac{\partial^2 \overline{C}_{ij}}{\partial x_k^2} + \frac{1}{We} \left(\frac{L^2}{L^2 - \overline{C}_{ii}} \overline{C}_{ij} - \delta_{ij} \right) = \Lambda_{ij} \quad (15)$$

This production term is approximated by

$$\Lambda_{ij} = C_{r0} We \mu_T \frac{\delta_{ij}}{3} \quad (16)$$

The value of C_{r0} is however unknown. Different values have been tested but all simulations gave even higher drag than the previous model. For instance at a $\beta = 0.9$ and $We = 10$, a value $C_{r0} = 15$ leads to a pressure gradient $dP/dx = 0.004905$, whereas $C_{r0} = 0$ gives $dP/dx = 0.004334$. This seems to indicate that it is not the best way to model the conformation tensor C_{ij}. Fig. 7 shows the same qualitative behaviour as the previous

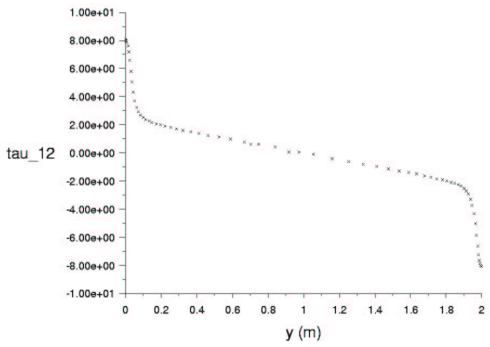


Figure 7: Polymer stress τ_{xy} for $\beta = 0.9$, $We = 10$ and $C_{r0} = 15$

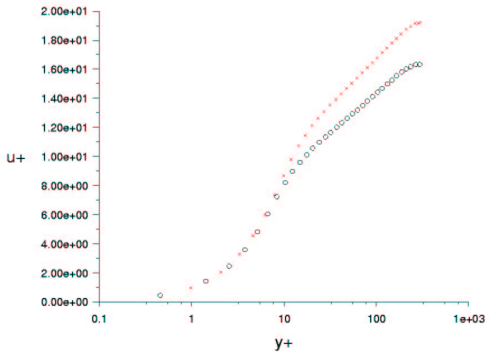


Figure 8: Velocity profile; o: $\beta = 0.9$, $We = 10$ and $C_{r0} = 15$, x: Newtonian ($\beta = 1$)

model. It tends to show that the addition of the conformation strain production term does not improve the results. Moreover, the shift of the velocity profile is down and not up.

6 No closure

Finally, the simplest way to close the problem is to neglect all the unknown terms. Again, this method is similar to the first model but without the modification of the factor in front of the polymer stress term. For $\beta = 0.9$ and $We = 50$ the pressure gradient is $dP/dx = 0.003509$ which is smaller than the Newtonian case. However, the velocity profile is still not shifted up as shown in Fig. 9. This prooves than the physics is still not described accurately, even if a small drag reduction is observed. The stress τ_{xy} is similar to that shown previously.

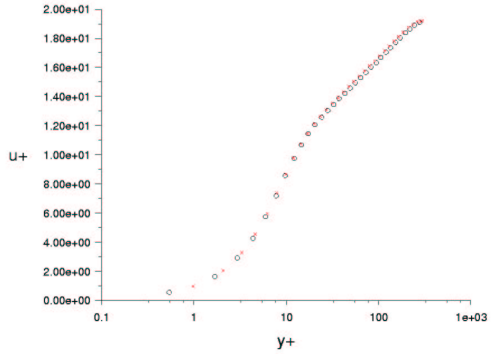


Figure 9: Velocity profile; o:$\beta = 0.9$, $We = 50$ and $C_{r0} = 0$, x:Newtonian ($\beta = 1$)

7 Discussion

The results shown in the previous sections demonstrate that the chosen strategy for the closure of the equations is not satisfactory. For almost all cases a drag increase has been observed instead of the expected drag reduction. The shape of the velocity profile and especially the log-law region are not accurately described. None of the calculations has shown the shift of the log-law region and its slope change. Different explanations can be attempted. The Reynolds average Navier-Stokes equation (Eq. 6) contains two terms which need closure, namely the Reynolds stress and the polymeric stress. Depending on their relative importance, one or the other or both are the source of the discrepancies. So far only the polymeric term has been investigated but the Reynolds stress is also influenced by the polymers. Especially since the polymers act locally in changing the turbulence. Therefore the Spalart-Allmaras model or any other model used for turbulence needs to be adapted accordingly. As shown in the previous figures, the polymeric stress itself is not either well described so that improvement is also needed in the closure of the polymeric stress term.

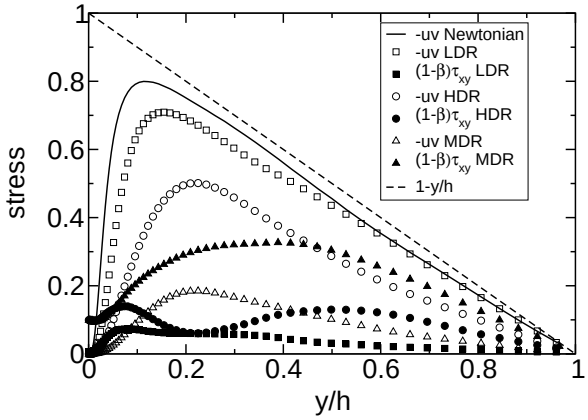


Figure 10: Polymeric and Reynolds stresses as function of the distance to the wall for different drag regimes (DNS simulation)

A simple way to investigate this problem is to use DNS results to validate the approximations. For instance, the polymeric stress given by a DNS simulation (see Fig. 10) could be used, so that the only term which is not exact is the Reynolds stress. It allows so a good quantification of the relative importance of the two terms. Similarly, the Reynolds stress given by the DNS could be used to check the accuracy of the polymeric stress.

8 Conclusion

The RANS simulation of a dilute polymer solution in a channel flow has been investigated. The polymers were modeled by the FENE-P equations. Different simple approximation have been tested as well as different parameters. It was shown that none of the models could describe quantitatively the physics and the drag reduction. A protocole has been proposed in order to investigate in a more systematic manner the influence of the different

terms. This approach was based on DNS results in order to localize and simplify the different closure problems. The subject is however complex and much more time would be required in order to develop a clean and accurate closure.

9 Appendix: User-defined function

The following user-defined function was used for the different calculations. Depending on the model, C_{r0} and/or the eddy-viscosity were set to zero.

```
#include "udf.h"

/*-----------------*/
/* model constants */
/*-----------------*/

#define fenep_L 60.0
#define fenep_e 0.9
#define fenep_W 10.0
#define rho     1.0
#define mu      0.0002
#define C_r0    15.0

/*-----------------------------------------------------*/
/* User-defined scalars to store the C_ij, T_ij and F */
/*-----------------------------------------------------*/

enum
{
  C11,
  C22,
  C12,
  FF,
  T11,
  T22,
  T12,
  N_REQUIRED_UDS
};

/*-------------------------*/
/* Properties of the Fluid */
/*-------------------------*/

DEFINE_PROPERTY(density, c, t)
{
  return  rho;
}
```

```
DEFINE_PROPERTY(viscosity, c, t)
{
  return  (mu/rho)*fenep_e;
}

/*-----------------------------------------------------------*/
/* Source terms in the C_ij equations and in the momentum */
/*-----------------------------------------------------------*/

DEFINE_SOURCE(c11_source, c, t, dS, eqn)
{
  dS[eqn] = -(1./fenep_W)*C_UDSI(c,t,FF);
  return 2.*C_UDSI(c,t,C12)*C_DUDY(c,t)-(1./fenep_W)*
         (C_UDSI(c,t,FF)*C_UDSI(c,t,C11)-1.)+C_r0*fenep_W*C_MU_T(c,t)/3.0;
}

DEFINE_SOURCE(c22_source, c, t, dS, eqn)
{
  dS[eqn] = -(1./fenep_W)*C_UDSI(c,t,FF);
  return -(1./fenep_W)*(C_UDSI(c,t,FF)*C_UDSI(c,t,C22)-1.)+
 C_r0*fenep_W*C_MU_T(c,t)/3.0;
}

DEFINE_SOURCE(c12_source, c, t, dS, eqn)
{
  dS[eqn] = -(1./fenep_W)*C_UDSI(c,t,FF);
  return C_UDSI(c,t,C22)*C_DUDY(c,t)-(1./fenep_W)*
 C_UDSI(c,t,FF)*C_UDSI(c,t,C12);
}

DEFINE_SOURCE(u_source, c, t, dS, eqn)
{
  real us;
  dS[eqn] = 0.;
  us = (1.-fenep_e)*((mu+C_MU_T(c,t))/rho)*(C_UDSI_G(c,t,T11)[0]+
       C_UDSI_G(c,t,T12)[1]);
  return us;
}

DEFINE_SOURCE(v_source, c, t, dS, eqn)
{
  real vs;
  dS[eqn] = 0.;
  vs = (1.-fenep_e)*((mu+C_MU_T(c,t))/rho)*(C_UDSI_G(c,t,T12)[0]+
       C_UDSI_G(c,t,T22)[1]);
  return vs;
```

```
}

DEFINE_ADJUST(adjust, domain)
{
 /*-----------------------------------------------------------*/
 /* Called at the begining and at the end of each iteration */
 /*-----------------------------------------------------------*/

  Thread *t;
  Thread *tf;
  cell_t c;
  face_t f;

  real l2=fenep_L*fenep_L;

 /*-------------------------------------------*/
 /* clip the C_ij  and compute the FF and T_ij */
 /*-------------------------------------------*/

  thread_loop_c (t, domain)
    begin_c_loop(c,t)
      {
        C_UDSI(c,t,C11) = MAX(C_UDSI(c,t,C11),1.0e-6);
        C_UDSI(c,t,C22) = MAX(C_UDSI(c,t,C22),1.0e-6);
        C_UDSI(c,t,C11) = MIN(C_UDSI(c,t,C11),0.99*l2-C_UDSI(c,t,C22));

        C_UDSI(c,t,FF) = l2/(l2-C_UDSI(c,t,C11)-C_UDSI(c,t,C22));

        C_UDSI(c,t,T11) = (1./fenep_W)*(C_UDSI(c,t,FF)*C_UDSI(c,t,C11)-1.);
        C_UDSI(c,t,T22) = (1./fenep_W)*(C_UDSI(c,t,FF)*C_UDSI(c,t,C22)-1.);
        C_UDSI(c,t,T12) = (1./fenep_W)*(C_UDSI(c,t,FF)*C_UDSI(c,t,C12)   );
      }
    end_c_loop(c,t)

}
```

References