The Ghost Fluid Method for Viscous Flows[1]

Ronald P. Fedkiw

Computer Science Department
Stanford University, Stanford, California 94305
Email:fedkiw@cs.stanford.edu

Xu-Dong Liu

Department of Mathematics
University of California Santa Barbara, Santa Barbara, California, 93106
Email:xliu@math.ucsb.edu

Keywords: Navier Stokes Equations, Interfaces, Multiphase Flow, Ghost Fluid Method, Viscosity

Abstract: The level set method for compressible flows [13] is simple to implement, especially in the presence of topological changes. However, this method was shown to suffer from large spurious oscillations in [11]. In [4], a new Ghost Fluid Method (GFM) was shown to remove these spurious oscillations by minimizing the numerical smearing in the entropy field with the help of an Isobaric Fix [6] technique. The original GFM was designed for the inviscid Euler equations. In this paper, we extend the formulation of the GFM and apply the extended formulation to the viscous Navier-Stokes equations. The resulting numerical method is robust and easy to implement along the lines of [15].

1 Introduction

In [13], the authors applied the level set method to multiphase compressible flows. The level set function was used as an indicator function and each grid point was designated as one fluid or the other for evaluation of the equation of state. Then numerical fluxes were formed and differenced in the usual manner [15]. The usual virtues of ease of implementation, in particular as regards to topological changes, were apparent. However, in [11], it was shown that this technique produced large spurious oscillations in the pressure and velocity fields. This problem was rectified in [10], [3], and [2] with schemes that involved *explicit* treatment of the appropriate conditions at the interface. As a consequence, these schemes are intricate in one dimension and can only be extended to multiple dimensions with ill-advised dimensional splitting in time. In addition, multilevel time integrators, such as Runge Kutta methods, are difficult to implement for these schemes.

The Ghost Fluid Method [4] avoids these oscillations at multimaterial interfaces without *explicitly* using interface jump conditions. Instead, the GFM

[1]This paper was presented in "Solutions of PDE" Conference in honour of Prof. Roe on the occassion of his 60th birthday, July 1998, Arachaon, France

creates a artificial fluid which *implicitly* induces the proper conditions at the interface. In the flavor of the level set function which gives an *implicit* representation of the interface, the GFM gives an *implicit* representation of the Rankine-Hugoniot jump conditions at the interface. Since the jump conditions are handled *implicitly* by the construction of a ghost fluid, the overall scheme becomes easy to implement in multidimensions without time splitting. In addition, Runge Kutta methods are trivial to apply.

In [4], the GFM was implemented for contact discontinuities in the inviscid Euler equations. In this case, the pressure and normal velocity of the ghost fluid are just copied over from the real fluid in a node by node fashion while the entropy and tangential velocities are defined with a simple partial differential equation for one-sided extrapolation in the normal direction. See [4] for details.

In this paper, we will generalize the GFM and show how this new general technique can be used for material interfaces in the viscous Navier-Stokes equations. The GFM will *implicitly* enforce the jump conditions at the interface by the construction of a ghost fluid. The resulting numerical method is easy to implement in multidimensions (without time splitting) and extends trivially to Runge Kutta methods.

2 Equations

2.1 Navier-Stokes Equations

The basic equations for viscous compressible flow are the Navier-Stokes equations,

$$\vec{U}_t + \vec{F}(\vec{U})_x + \vec{G}(\vec{U})_y + \vec{H}(\vec{U})_z = Vis \tag{1}$$

which can be written in more detail as

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E+p)u \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E+p)v \end{pmatrix}_y + \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E+p)w \end{pmatrix}_z = Vis \tag{2}$$

where t is the time, (x, y, z) are the spatial coordinates, ρ is the density, $\vec{V} =< u, v, w >$ is the velocity field, E is the total energy per unit volume, and p is the pressure. The total energy is the sum of the internal energy and the kinetic energy,

$$E = \rho e + \frac{\rho(u^2 + v^2 + w^2)}{2} \tag{3}$$

where e is the internal energy per unit mass. The two-dimensional Navier-Stokes equations are obtained by omitting all terms involving w and z. The one-dimensional Navier-Stokes equations are obtained by omitting all terms involving v, w, y, and z. The inviscid Euler equations are obtained by setting $Vis = 0$.

In general, the pressure can be written as a function of density and internal energy, $p = p(\rho, e)$, or as a function of density and temperature, $p = p(\rho, T)$. In order to complete the model, we need an expression for the internal energy per unit mass. Since $e = e(\rho, T)$ we write

$$de = \left(\frac{\partial e}{\partial \rho}\right)_T d\rho + \left(\frac{\partial e}{\partial T}\right)_\rho dT \tag{4}$$

which can be shown to be equivalent to

$$de = \left(\frac{p - Tp_T}{\rho^2}\right) d\rho + c_v dT \tag{5}$$

where c_v is the specific heat at constant volume. [6]

The sound speeds associated with the equations depend on the partial derivatives of the pressure, either p_ρ and p_e or p_ρ and p_T, where the change of variables from density and internal energy to density and temperature is governed by the following relations

$$p_\rho \to p_\rho - \left(\frac{p - Tp_T}{c_v \rho^2}\right) p_T \tag{6}$$

$$p_e \to p_\rho + \left(\frac{1}{c_v}\right) p_T \tag{7}$$

and the sound speed c is given by

$$c = \sqrt{p_\rho + \frac{p p_e}{\rho^2}} \tag{8}$$

for the case where $p = p(\rho, e)$ and

$$c = \sqrt{p_\rho + \frac{T(p_T)^2}{c_v \rho^2}} \tag{9}$$

for the case where $p = p(\rho, T)$.

2.1.1 Viscous Terms

We define the viscous stress tensor as

$$\tau = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{pmatrix} \tag{10}$$

where

$$\tau_{xx} = \frac{2}{3}\mu(2u_x - v_y - w_z), \qquad \tau_{xy} = \mu(u_y + v_x) \tag{11}$$

$$\tau_{yy} = \frac{2}{3}\mu(2v_y - u_x - w_z), \qquad \tau_{xz} = \mu(u_z + w_x) \tag{12}$$

$$\tau_{zz} = \frac{2}{3}\mu(2w_z - u_x - v_y), \qquad \tau_{yz} = \mu(v_z + w_y) \tag{13}$$

and μ is the viscosity. In addition, we define

$$\vec{\bigtriangledown} \cdot \tau = \begin{pmatrix} \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \end{pmatrix}_x + \begin{pmatrix} \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \end{pmatrix}_y + \begin{pmatrix} \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \end{pmatrix}_z \tag{14}$$

$$\vec{V}\tau = (u\tau_{xx} + v\tau_{xy} + w\tau_{xz}, u\tau_{xy} + v\tau_{yy} + w\tau_{yz}, u\tau_{xz} + v\tau_{yz} + w\tau_{zz}) \tag{15}$$

so that the viscosity terms in the Navier-Stokes equations can be represented as

$$Vis = \begin{pmatrix} 0 \\ \vec{\bigtriangledown} \cdot \tau \\ \vec{\bigtriangledown} \cdot (\vec{V}\tau) \end{pmatrix} \tag{16}$$

2.1.2 Eigensystem

The Navier-Stokes equations can be thought of as the inviscid Euler equations plus some viscosity terms. We discretize the spatial part of the inviscid Euler equations in the usual way, e.g. ENO [15]. These methods require an eigensystem which we list below. Note that we only list the two dimensional eigensystem, since there are no three dimensional examples in this paper. However, the method works well and it is straightforward to implement in three dimensions as we shall show in a future paper. Once the spatial part of the inviscid Euler equations is discretized, we discretize the viscous terms and use the combined discretization as the right hand side for a time integration method, e.g. we use 3rd order TVD Runge-Kutta [15].

The eigenvalues and eigenvectors for the Jacobian matrix of $\vec{F}(\vec{U})$ are obtained by setting $A = 1$ and $B = 0$ in the following formulas, while those for the Jacobian of $\vec{G}(\vec{U})$ are obtained with $A = 0$ and $B = 1$.

The eigenvalues are

$$\lambda^1 = \hat{u} - c, \quad \lambda^2 = \lambda^3 = \hat{u}, \quad \lambda^4 = \hat{u} + c \tag{17}$$

and the eigenvectors are

$$\vec{L}^1 = \left(\frac{b_2}{2} + \frac{\hat{u}}{2c}, -\frac{b_1 u}{2} - \frac{A}{2c}, -\frac{b_1 v}{2} - \frac{B}{2c}, \frac{b_1}{2} \right) \tag{18}$$

$$\vec{L}^2 = (1 - b_2, b_1 u, b_1 v, -b_1) \tag{19}$$

$$\vec{L}^3 = (\hat{v}, B, -A, 0) \tag{20}$$

$$\vec{L}^4 = \left(\frac{b_2}{2} - \frac{\hat{u}}{2c}, -\frac{b_1 u}{2} + \frac{A}{2c}, -\frac{b_1 v}{2} + \frac{B}{2c}, \frac{b_1}{2} \right) \tag{21}$$

$$\vec{R}^1 = \begin{pmatrix} 1 \\ u - Ac \\ v - Bc \\ H - \hat{u}c \end{pmatrix}, \quad \vec{R}^2 = \begin{pmatrix} 1 \\ u \\ v \\ H - \frac{1}{b_1} \end{pmatrix} \tag{22}$$

$$\vec{R}^3 = \begin{pmatrix} 0 \\ B \\ -A \\ -\hat{v} \end{pmatrix}, \quad \vec{R}^4 = \begin{pmatrix} 1 \\ u + Ac \\ v + Bc \\ H + \hat{u}c \end{pmatrix} \tag{23}$$

where

$$q^2 = u^2 + v^2, \quad \hat{u} = Au + Bv, \quad \hat{v} = Av - Bu \tag{24}$$

$$\Gamma = \frac{p_e}{\rho}, \quad c = \sqrt{p_\rho + \tfrac{\Gamma p}{\rho}}, \quad H = \frac{E + p}{\rho} \tag{25}$$

$$b_1 = \frac{\Gamma}{c^2}, \qquad b_2 = 1 + b_1 q^2 - b_1 H \tag{26}$$

The eigensystem for the one-dimensional equations is obtained by setting $v = 0$.

2.2 Level Set Equation

We use the level set equation

$$\phi_t + \vec{V} \cdot \vec{\bigtriangledown} \phi = 0 \tag{27}$$

to keep track of the interface location as the zero level of ϕ. In this equation, the level set velocity, $\vec{V}$, is chosen to be the local fluid velocity which is a natural choice when the interface is a simple contact discontinuity or a non-reacting material interface. In general ϕ starts out as the signed distance function, is advected by solving equation 27 using the methods in [9], and then is reinitialized using

$$\phi_t + S(\phi_o)\left(|\vec{\bigtriangledown}\phi| - 1\right) = 0 \tag{28}$$

to keep ϕ approximately equal to the distance function, $|\vec{\bigtriangledown}\phi| = 1$, near the interface where we need additional information. We note that our method allows us to solve equation 27 independently of the Navier-Stokes equations. That is, equation 27 can be solved directly using the method in [9], and the eigensystem listed above does not depend on ϕ, since we will be solving only one phase problems with any given eigensystem (see the later sections). For more details on the level set function see [4, 13, 14, 16].

2.3 Equation of State

For an ideal gas $p = \rho RT$ where $R = \frac{R_u}{M}$ is the specific gas constant, with $R_u \approx 8.31451\frac{J}{molK}$ the universal gas constant and M the molecular weight of the gas. Also valid for an ideal gas is $c_p - c_v = R$ where c_p is the specific heat at constant pressure. Additionally, gamma as the ratio of specific heats $\gamma = \frac{c_p}{c_v}$. [8]

For an ideal gas, equation 5 becomes

$$de = c_v dT \tag{29}$$

and assuming that c_v does not depend on temperature (calorically perfect gas), we integrate to obtain

$$e = c_v T \tag{30}$$

where we have set e to be zero at $0K$. Note that e is not uniquely determined, and we could choose any value for e at $0K$ (although one needs to use caution when dealing with more than one material to be sure that integration constants are consistent with the heat release in any chemical reactions that occur).

Note that we may write

$$p = \rho RT = \frac{R}{c_v}\rho e = (\gamma - 1)\rho e \tag{31}$$

for use in the eigensystem.

3 The GFM for Inviscid Flow

We use the level set function to keep track of the interface. The zero level marks the location of the interface, while the positive values correspond to one fluid and the negative values correspond to the other fluid. Each fluid satisfies the inviscid Euler equations ($Vis = 0$) described in the last section with different equations of state for each fluid. Based on the work in [9], the discretization of the level set function can be done independent of the two sets of Euler equations. Besides discretizing equation 27 we need to discretize two sets of Euler equations. This will be done with the help of ghost cells.

Any level set function defines two separate domains for the two separate fluids, i.e. each point corresponds to one fluid or the other. Our goal is to define a ghost cell at every point in the computational domain. In this way, each grid point will contain the mass, momentum, and energy for the real fluid that exists at that point (according to the sign of the level set function) and a ghost mass, momentum, and energy for the other fluid that does not really exist at that point (the fluid on the other side of the interface). Once the ghost cells are defined, we can use standard methods, e.g. see [15], to update the Euler equations at every grid point for both fluids. Then we advance the level set function to the next time step and use the sign of the level set function to determine which of the two updated fluid values should be used at each grid point.

Consider a general time integrator for the Euler equations. In general, we construct right hand sides of the ordinary differential equation for both fluids based on the methods in [15], then we advance the level set to the next time level and pick one of the two right hand sides to use for the Euler equations based on the sign of the level set function. This can be done for every step and every combination of steps in a multistep method. Since both fluids are solved for at every grid point, we just choose the appropriate fluid based on the sign of the level set function.

To summarize, use ghost cells to define each fluid at every point in the computational domain. Update each fluid separately in multidimensional space for one time step or one substep of a multistep time integrator with standard methods. Then update the level set function independently using the real fluid velocities and the sign of the level set function to decide which of the two answers is the valid answer at each grid point. Keep the valid answer and discard the other so that only one fluid is defined at each grid point.

Lastly, we note that only a band of 3 to 5 ghost cells on each side of the interface is actually needed by the computational method depending on the stencil and movement of the interface. One can optimize the code accordingly.

3.1 Defining Values at the Ghost Cells

In [4], the GFM was implemented for a contact discontinuity in the inviscid Euler equations. In that case, it was apparent that the pressure and normal velocity were continuous, while tangential velocity was continuous in the case of a no-slip interface but discontinuous for shear waves. It was also apparent that the entropy was discontinuous.

For variables that are continuous across the interface, we define the values of the ghost fluid to be equal to the values of the real fluid at each grid point. Since these variables are continuous, this node by node population will implicitly capture the interface values of the continuous variables.

Note that the discontinuous variables are governed by a linearly degenerate eigenvalue. Thus, they move with the speed of the interface and information in these variables should not cross the interface. In order to avoid numerical smearing of these variables, we use one sided extrapolation to populate the values in the ghost fluid. Note that the work in [6] shows that one does not have to deal directly with the entropy. There are a few options for the choice of the variable used in extrapolation, ranging from density to temperature.

The extrapolation of the discontinuous variables is carried out in the following fashion. Using the level set function, we can define the unit normal at every grid point as

$$\vec{N} = \frac{\vec{\bigtriangledown}\phi}{|\vec{\bigtriangledown}\phi|} =< n_1, n_2, n_3 > \tag{32}$$

and then we can solve the advection equation

$$I_t \pm \vec{N} \cdot \vec{\bigtriangledown} I = 0 \tag{33}$$

for each variable I that we wish to extrapolate. Note that the "$\pm$" sign is chosen to be of one sign to propagate one fluid's values into one ghost region and the opposite sign to propagate the other fluid's values into the other ghost region. Of course this partial differential equation is solved with a Dirichlet type boundary condition so that the real fluid values do not change. This equation only needs to be solved for a few time steps to populate a thin band of ghost cells needed for the numerical method.

In addition, we use the Isobaric Fix [6] to minimize "overheating" when necessary. In order to do this, we adjust our Dirichlet type boundary condition in the procedure above so that a band of real fluid values can change their entropy.

When the need arises to extrapolate the tangential velocity, we achieve this by first extrapolating the entire velocity field, $\vec{V} =< u, v, w >$. Then, at every cell in the ghost region we have two separate velocity fields, one from the real fluid and one from the extrapolated fluid. For each velocity field, the normal component of velocity, $V_N = \vec{V} \cdot \vec{N}$, is put into a vector of length three, $V_N\vec{N}$, and then we use a complementary projection idea [7] to define the two dimensional velocity field in the tangent plane by another vector of length three, $\vec{V} - V_N\vec{N}$. Finally, we take the normal component of velocity, $V_N\vec{N}$, from the real fluid and the tangential component of velocity, $\vec{V} - V_N\vec{N}$, from the extrapolated fluid and add them together to get our new velocity to occupy the ghost cell. This new velocity is our ghost fluid velocity.

Once the ghost fluid values are defined as outlined above, they can be use to assemble the conserved variables for the ghost fluid.

4 Generalization of the Ghost Fluid Method

For a simple contact discontinuity in the inviscid Euler equations, we were able to separate the variables into two sets based on their continuity at the interface. The continuous variables were copied into the ghost fluid in a node by node fashion in order to capture the correct interface values. The discontinuous variables were extrapolated in a one sided fashion to avoid numerical dissipation. We wish to apply this idea to a general interface, moving at speed D in the normal direction, separating two general materials. That is, we need to know which variables are continuous for this general interface.

Conservation of mass, momentum, and energy can be applied to the interface in order to abstract some continuous variables. One can place a flux on the interface oriented tangent to the interface so that material that passes through this flux passes through the interface. Then this flux will move with speed D (the interface speed in the normal direction), and the mass, momentum, and energy which flows into this flux from one side of the interface must flow back out the other side of the interface. That is, the mass, momentum, and energy flux in this moving reference frame are continuous variables. Otherwise, there would be a mass, momentum, or energy sink at this interface and conservation would be violated. We will denote mass, momentum, and energy flux in this moving reference frame as F_ρ, $\vec{F}_{\rho\vec{V}}$, and F_E respectively. The statement that these variables are continuous is essentially the Rankine-Hugoniot jump conditions for an interface moving with speed D in the normal direction. Instead of applying the Rankine-Hugoniot jump conditions explicitly to the interface, we will use the fact that F_ρ, $\vec{F}_{\rho\vec{V}}$, and F_E are continuous to define a ghost fluid that captures the interface values of these variables.

Remark: Note that numerically F_ρ, $\vec{F}_{\rho\vec{V}}$, and F_E may not be continuous. This could occur from initial data or wave interactions. However, since we treat F_ρ, $\vec{F}_{\rho\vec{V}}$, and F_E as though they were continuous in the numerical method, numerical dissipation will smooth them out. In fact, this numerical dissipation will help to guarantee the correct numerical solution.

Remark: The level set function is only designed to represent interfaces where the interface crosses the material at most once [14]. Simple material interfaces that move with the local material velocity never cross over material. If one material is being converted into another then the interface may include a regression rate for this conversion. If the regression rate for this conversion of one material into another is based on some sort of chemical reaction, then the interface can pass over a material exactly once changing it into another material. The same chemical reaction cannot occur to a material more than once. The GFM for deflagration and detonation discontinuities is presented in [5].

Remark: Shocks may be seen as a conversion of an uncompressed material to a compressed material. In this case, D would be the shock speed. This method could be used to follow a lead shock, but since shocks can pass over a material more than once, all subsequent shocks must be captured. In fact, this turns out to be useful and we will examine the use of this method for shocks in a future paper [1].

Remark: In the general case, $\vec{F}_{\rho\vec{V}}$ and F_E will include general mechanical stress terms on the interface besides viscosity and pressure, e.g. surface tension, and material models. We will consider these general mechanical stress terms in future papers. In this paper, pressure and viscosity will be the only mechanical stress terms on the interface.

Remark: In the general case, F_E will include general thermal stress on the interface, e.g. thermal conductivity. We will consider thermal stress in a future paper.

To define F_ρ, $\vec{F}_{\rho\vec{V}}$, and F_E, one takes the equations and writes them in conservation form for mass, momentum, and energy. The fluxes for these variables are then rewritten in the reference frame of a flux which is tangent to the interface by simply taking the dot product with the normal direction

$$\left[\left\langle \vec{F}(\vec{U}), \vec{G}(\vec{U}), \vec{H}(\vec{U}) \right\rangle - \begin{pmatrix} \vec{0} \\ \tau \\ \vec{V}\tau \end{pmatrix}\right] \cdot \vec{N} \tag{34}$$

where $\vec{0} =< 0, 0, 0 >$. Note that the dot product with $\vec{N}$ is equivalent to multiplication by the transpose of $\vec{N}$. We use the superscript T to designate the transpose and rewrite the above expression as

$$\begin{pmatrix} \rho \\ \rho\vec{V}^T \\ E \end{pmatrix} V_N + \begin{pmatrix} 0 \\ (pI - \tau)\vec{N}^T \\ \vec{V}(pI - \tau)\vec{N}^T \end{pmatrix} \tag{35}$$

where $V_N = \vec{V} \cdot \vec{N}$ is the local fluid velocity normal to the interface and I is the three by three identity matrix. Then the measurements are taken in the moving reference frame (speed D) to get

$$\begin{pmatrix} \rho \\ \rho\left(\vec{V}^T - D\vec{N}^T\right) \\ \rho e + \frac{\rho|\vec{V} - D\vec{N}|^2}{2} \end{pmatrix} (V_N - D) + \begin{pmatrix} 0 \\ (pI - \tau)\vec{N}^T \\ (\vec{V} - D\vec{N})(pI - \tau)\vec{N}^T \end{pmatrix} \tag{36}$$

from which we define

$$F_\rho = \rho(V_N - D) \tag{37}$$

$$\vec{F}_{\rho\vec{V}} = \rho\left(\vec{V}^T - D\vec{N}^T\right)(V_N - D) + (pI - \tau)\vec{N}^T \tag{38}$$

$$F_E = \left(\rho e + \frac{\rho|\vec{V} - D\vec{N}|^2}{2}\right)(V_N - D) + (\vec{V} - D\vec{N})(pI - \tau)\vec{N}^T \tag{39}$$

as continuous variables for use in the GFM. That is, we will define the ghost fluid in a node by node fashion by solving the system of equations

$$F_\rho^G = F_\rho^R \tag{40}$$

$$\vec{F}_{\rho\vec{V}}^G = \vec{F}_{\rho\vec{V}}^R \tag{41}$$

$$F_E^G = F_E^R \tag{42}$$

at each grid point. Note that the superscript "R" stands for a real fluid value at a grid point, while the superscript "G" stands for a ghost fluid value at a grid point. Since F_ρ^R, $\vec{F}_{\rho\vec{V}}^R$, F_E^R, $\vec{N}$, and D are known at each grid point, these can be substituted into equations 40, 41, and 42, leaving ρ^G, $\vec{V}^G$, p^G, e^G, and τ^G undetermined. The appropriate equation of state can be used to eliminate one of these variables.

4.1 Non-reacting Material Interfaces

In this paper, we are interested in simple material interfaces where the interface moves with the fluid velocity only. That is, we use the local fluid velocity, $\vec{V}$, in the level set equation as the level set velocity.

The level set velocity, $\vec{V}$, is defined everywhere by a continuous function where the value of $\vec{V}$ at the interface is equal to the value that moves the interface at the correct interface velocity. This global definition of $\vec{V}$ is the one we use to find D for use in solving equations 40, 41, and 42. In a node by node fashion, we define $D = \vec{V} \cdot \vec{N}$ as the velocity of the interface in the normal direction. In this way, we capture the correct value of D at the interface.

Using $D = \vec{V}^R \cdot \vec{N} = V_N^R$ in equation 40 yields

$$\rho^G (V_N^G - V_N^R) = 0 \tag{43}$$

implying that $V_N^G = V_N^R$. That is, the normal component of the ghost fluid velocity should be equal to the normal component of the real fluid velocity at each point. This allows us to simplify equations 38 and 39 to

$$\vec{F}_{\rho\vec{V}} = (pI - \tau)\vec{N}^T \tag{44}$$

$$F_E = (\vec{V} - V_N\vec{N})(pI - \tau)\vec{N}^T = -(\vec{V} - V_N\vec{N})\tau\vec{N}^T \tag{45}$$

If the flow were inviscid, then $\tau = 0$ and equation 41 becomes $p^G = p^R$ implying that the pressure of the ghost fluid should be equal to the pressure of the real fluid at each point. Equation 42 is trivially satisfied and we are left with some freedom. As shown earlier, the entropy should be extrapolated in the normal direction along with an Isobaric Fix [6] to minimize "overheating". The tangential velocities may be extrapolated for a shear wave or copied over node by node to enforce continuity of the tangential velocities for a "no-slip" boundary condition.

5 The Viscous Stress Tensor

Before we define τ^G, we compute τ at each grid point by first computing the spatial derivatives of the velocities at each grid point. For a "no-slip" boundary condition, we use standard second order central differencing in order to compute the derivatives of the velocities. Note that this will consist of differencing across the interface, and discontinuous velocity profiles will become continuous due to numerical dissipation. For a "slip" boundary condition, we avoid differencing across the interface in order to allow jumps in the velocity field. If a grid point and both of its neighbors have the same sign of ϕ (meaning that they are all on the same side of the interface), then we use a standard second order central difference. If a grid point only has one neighbor with the same sign of ϕ, then we only use that neighbor and a first order one-sided difference. If a grid point has a different sign of ϕ than both of its neighbors, we are in an underresolved region and cannot compute the derivative. Our only choice is to set the derivative to zero. Once all the spatial derivatives are computed, we combine them with the appropriate value of μ depending on the sign of ϕ.

After computing τ at all the grid points, we use use equation 33 to extrapolate each term in τ across the interface. Then, at every grid point in the ghost region we have two separate values for τ, one from the real fluid and one from the extrapolated fluid. Equations 41 and 42 tell us that part of the viscous stress tensor is continuous and part is discontinuous. We write

$$\tau = \tau_C + \tau_D \tag{46}$$

where τ_C is the continuous part of τ, and τ_D is the discontinuous part of τ. Then we take the real fluid value of τ_C and the extrapolated fluid value of τ_D and add them together to get τ^G as our ghost fluid value at each grid point.

Consider a specific grid point where the unit normal is $\vec{N}$. We could choose a basis for the the two dimensional tangent plane with two orthogonal unit tangent vectors $\vec{T}_1$ and $\vec{T}_2$. Then the viscous stress tensor can be written in this new coordinate system with a simple transformation

$$\begin{pmatrix} \tau_{nn} & \tau_{nt_1} & \tau_{nt_2} \\ \tau_{nt_1} & \tau_{t_1t_1} & \tau_{t_1t_2} \\ \tau_{nt_2} & \tau_{t_1t_2} & \tau_{t_2t_2} \end{pmatrix} = \begin{pmatrix} \vec{N} \\ \vec{T}_1 \\ \vec{T}_2 \end{pmatrix} \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{pmatrix} \begin{pmatrix} \vec{N} \\ \vec{T}_1 \\ \vec{T}_2 \end{pmatrix}^T \tag{47}$$

where we have defined our transformation matrix Q by

$$Q = \begin{pmatrix} \vec{N} \\ \vec{T}_1 \\ \vec{T}_2 \end{pmatrix} \tag{48}$$

with $Q^T Q = QQ^T = I$.

5.1 Momentum Terms

We use the transformation Q to put equation 44 into a better coordinate system

$$Q\vec{F}_{\rho\vec{V}} = \begin{pmatrix} p - \tau_{nn} \\ -\tau_{nt_1} \\ -\tau_{nt_2} \end{pmatrix} \tag{49}$$

illustrating that $p - \tau_{nn}$, τ_{nt_1} and τ_{nt_2} are continuous variables across the interface. Thus τ_C should contain τ_{nt_1} and τ_{nt_2} and the jump in τ_{nn} can be used to find the pressure of the ghost fluid.

We define the continuous part of τ as

$$\tau_C = Q^T \begin{pmatrix} 0 & \tau_{nt_1} & \tau_{nt_2} \\ \tau_{nt_1} & 0 & 0 \\ \tau_{nt_2} & 0 & 0 \end{pmatrix} Q \tag{50}$$

and compute τ_C through the equation

$$\tau_C = \hat{\tau}_C + \hat{\tau}_C^T \tag{51}$$

where

$$\hat{\tau}_C = Q^T \begin{pmatrix} 0 & 0 & 0 \\ \tau_{nt_1} & 0 & 0 \\ \tau_{nt_2} & 0 & 0 \end{pmatrix} Q = Q^T \begin{pmatrix} 0 \\ \tau_{nt_1} \\ \tau_{nt_2} \end{pmatrix} \vec{N} \tag{52}$$

is easy to find. Note that equation 47 shows that

$$\begin{pmatrix} \tau_{nn} \\ \tau_{nt_1} \\ \tau_{nt_2} \end{pmatrix} = Q\tau\vec{N}^T \tag{53}$$

and using this in equation 52 gives

$$\hat{\tau}_C = Q^T \left[Q\tau\vec{N}^T - \begin{pmatrix} \tau_{nn} \\ 0 \\ 0 \end{pmatrix} \right] \vec{N} = \tau\vec{N}^T\vec{N} - \vec{N}^T\tau_{nn}\vec{N} \tag{54}$$

which can be further simplified to

$$\hat{\tau}_C = (\tau - \tau_{nn}I)\vec{N}^T\vec{N} \tag{55}$$

finally allowing us to define

$$\tau_C = (\tau - \tau_{nn}I)\vec{N}^T\vec{N} + \vec{N}^T\vec{N}(\tau - \tau_{nn}I) \tag{56}$$

From equation 47 we have $\tau_{nn} = \vec{N}\tau\vec{N}^T$ and we use this in equation 56 to compute τ_C. We define τ^G by adding the real fluid value of τ_C to the extrapolated fluid value of $\tau_D = \tau - \tau_C$.

We can find the pressure by solving the equation

$$p^G - \tau_{nn}^G = p^R - \tau_{nn}^R \tag{57}$$

for p^G using the extrapolated value of τ to define τ_{nn}^G.

We stress that equation 56 and the entire numerical algorithm for the viscous stress tensor only depends on the normal, $\vec{N}$, of Q. We have used a basis free projection technique to eliminate the dependence on $\vec{T}_1$ and $\vec{T}_2$ similar to the work in [7].

5.2 Energy Terms

We use equation 47 to rewrite equation 45 as

$$F_E = -(\vec{V} - V_N\vec{N})Q^T \begin{pmatrix} \tau_{nn} \\ \tau_{nt_1} \\ \tau_{nt_2} \end{pmatrix} = -(0, V_{t_1}, V_{t_2}) \begin{pmatrix} \tau_{nn} \\ \tau_{nt_1} \\ \tau_{nt_2} \end{pmatrix} \tag{58}$$

where V_{t_1} and V_{t_2} are the velocities in the tangent directions T_1 and T_2 respectively. Thus, we need

$$F_E = -(V_{t_1}\tau_{nt_1} + V_{t_2}\tau_{nt_2}) \tag{59}$$

to be continuous across the interface.

When the viscosity is zero or the flow is uniform, we have $\tau_{nt_1} = \tau_{nt_2} = 0$ and equation 59 is trivially satisfied. In this case the tangential velocities may or may not be continuous. That is, we may have a shear wave in which case we use a "slip" boundary condition.

When the viscosity is nonzero and the flow is not uniform, $\tau_{nt_1} \neq 0 \neq \tau_{nt_2}$ and we need to define V_{t_1} and V_{t_2} so that equation 59 is satisfied. This can be done with the "no-slip" boundary condition which requires V_{t_1} and V_{t_2} to be continuous across the interface. Since V_N is necessarily continuous, the "no-slip" boundary condition requires the entire velocity field, $\vec{V}$, to be continuous.

5.3 Constructing Fluxes

We will discuss the the numerical discretization in the x-direction, after which the y-direction and z-direction can be inferred by symmetry. We will need a spatial discretization for

$$\begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \end{pmatrix}_x \tag{60}$$

which will be achieved by differencing fluxes. In order to construct the appropriate flux function (which is just equation 60 without the x derivative) we need values of τ_{xx}, τ_{xy}, τ_{xz} and $\vec{V}$ at the cell walls which are centered between the grid points. We define these functions at the cell walls by the average of the values at the neighboring grid points.

Using the arithmetic average to define τ_{xx}, τ_{xy}, τ_{xz} and $\vec{V}$ gives the standard definition of μ, v_y, w_z, u_y, and u_z, but gives a rather wide stencil for u_x, v_x, and w_x. We wish to avoid this wide stencil in regions of the real fluid away from the interface. If a flux in the real fluid has both of its neighboring grid points in the real fluid, then we replace arithmetic average of u_x, v_x, and w_x with the centered difference computed using these two grid points. This gives the standard compact stencil which avoids "odd-even" decoupling.

6 Examples

In our numerical examples we use third order ENO-LLF with 3rd order TVD Runge Kutta methods [15].

6.1 Example 1

We use a $1m$ square domain with 20 grid points in each direction with linear extrapolation of all variables as a first order boundary condition. We use $\gamma = 1.4$, $M = .029\frac{kg}{mol}$, $\rho = 1\frac{kg}{m^3}$, $u = 0\frac{m}{s}$, and $p = 1 \times 10^5 Pa$ for both gases. The interface is defined by $\phi = x - .5$ which makes the interface a straight line defined by $x = .5$, i.e. $\phi = 0$. We initialize the velocity tangent to the interface by $v = 300\sin(2\pi\phi)$ as shown in figure 1. A cross-sectional view of the tangential velocity is shown in 2.

We set the viscosity in both fluids to $100\frac{kg}{ms}$ and show the solution at $t = .001s$ in figure 3. Since both viscosities are equal, the spatial profile of the tangential velocity is linear. We used the "no-slip" boundary condition.

In figure 4 we change the viscosity on the left to $50\frac{kg}{ms}$ and show the solution at $t = .0025$. Once again the spatial profile of the tangential velocity is linear, however there is a kink in the linear function at the interface due to the jump in viscosity. We used the "no-slip" boundary condition.

In figure 5 we change the viscosity on the left to zero, which in turn implies the use of a "slip" boundary condition. At $t = .001s$, the viscous fluid on the right is seeking a flat profile (uniform flow) as shown in figure 5. Note that the discontinuous tangential velocity is indicative of a shear wave.

6.2 Example 2

In this example we use $\phi = -y+2x-.5$ which defines the $\phi = 0$ interface by the straight line $y = 2x - .5$. This interface is rotated 22.5 degrees clockwise from the interface in the last example. In addition, we change the initial velcoity field to $u = 300\sin(\frac{\pi\phi}{.9d00})\sin(67.5^o)\frac{m}{s}$ and $v = 300\sin(\frac{\pi\phi}{.9d00})\cos(67.5^o)\frac{m}{s}$ as shown in figure 6. We choose a line perpendicular to the interface and plot a cross-section of the tangential velocities in 7. We treat the boundary of the problem by linear extrapolation in the direction defined by the line $y = 2x$. This leaves four boundary values in the upper left hand corner and the lower right hand corner undefined, and we define them by constant extrapolation from the interior corner points, i.e. points (2,19) and (19,2) respectively.

As in the last example, we show the solution at $t = .001s$ in figure 8 for the case where both fluids have a viscosity of $100\frac{kg}{ms}$. In figure 9, we show the solution at $t = .0025$ where the fluid on the left had its viscosity changed to $50\frac{kg}{ms}$. In figure 10, we show the solution at $t = .001s$ where the fluid on the left has zero viscosity. Once again, the first two cases had a "no-slip" boundary condition while the last case has a "slip" boundary condition. We get the expected results in each case.

6.3 Example 3

In this example we return to the interface defined by $\phi = x - .5$ in Example 1. All initial data is the same as in Example 1, except that we define a uniform normal velocity of $u = 50\frac{m}{s}$ so that the interface will move to the right, crossing grid nodes. The initial velcoity field and a cross-section of the tangential velocity are shown in figure 11 and 12 respectively.

We show the solution at $t = .00175s$ for the case where both fluids have viscosities of $100\frac{kg}{ms}$ in figure 13. $x = .6m$. We change the viscosity of the fluid on the left to $50\frac{kg}{ms}$ and show the results at $t = .0025$ in figure 14. We change the viscosity of the fluid on the left to zero and show the results at $t = .00175s$ in figure 15. The first two cases had a "no-slip" boundary condition while the last case has a "slip" boundary condition. In the first two cases, the interface has crossed two grid nodes and is located near $x = .6$, while in the last case the interface has only crossed one grid node.

7 Acknowledgements

Research supported in part by ONR N00014-97-1-0027 and ONR N00014-97-1-0968.

References

[1] Aslam,T.D., *A Level Set Algorithm for Tracking Discontinuities in Hyperbolic Conservation Laws II: Systems of Equations*, (in preparation).

[2] Cocchi, J.-P., Saurel, S., *A Riemann Problem Based Method for the Resolution of Compressible Multimaterial Flows*, Journal of Computational Physics, vol. 137, (1997) pp. 265-298.

[3] Davis, S., *An interface tracking method for hyperbolic systems of conservation laws*, Applied Numerical Mathematics, 10 (1992) 447-472.

[4] Fedkiw, R., Aslam, T., Merriman, B., and Osher, S., *A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method)*, J. Computational Physics, vol. 152, n. 2, 457-492 (1999).

[5] Fedkiw, R., Aslam, T., and Xu, S., *The Ghost Fluid Method for deflagration and detonation discontinuities*, J. Computational Physics 154, n. 2, 393-427 (1999).

[6] Fedkiw, R., Marquina, A., and Merriman, B., *An Isobaric Fix for the Overheating Problem in Multimaterial Compressible Flows*, J. Computational Physics 148, 545-578 (1999).

[7] Fedkiw, R., Merriman, B., and Osher, S., *Efficient characteristic projection in upwind difference schemes for hyperbolic systems (The Complementary Projection Method)*, J. Computational Physics, vol. 141, 22-36 (1998).

[8] Fedkiw, R., Merriman, B., and Osher, S., *High accuracy numerical methods for thermally perfect gas flows with chemistry*, J. Computational Physics 132, 175-190 (1997).

[9] Fedkiw, R., Merriman, B., and Osher, S., *Simplified Discretization of Systems of Hyperbolic Conservation Laws Containing Advection Equations*, J. Computational Physics, 157, 302-326 (2000).

[10] Fedkiw, R., Merriman, B., and Osher, S., *Numerical methods for a one-dimensional interface separating compressible and incompressible flows*, Barriers and Challenges in Computational Fluid Dynamics, pp 155-194, edited by V, Venkatakrishnan, M. Salas, and S. Chakravarthy, Kluwer Academic Publishers (Norwell,MA), 1998.

[11] Karni, S., *Multicomponent Flow Calculations by a Consistent Primitive Algorithm*, Journal of Computational Physics, v. 112, 31-43 (1994).

[12] Liu, X-D., and S. Osher, *Convex ENO High Order Schemes Without Field-by-Field Decomposition or Staggered Grids*, J. Comput Phys, v142, pp 304-330, (1998).

[13] Mulder, W., Osher, S., and Sethian, J.A., *Computing Interface Motion in Compressible Gas Dynamics*, J. Comput. Phys., v. 100, 209-228 (1992).

[14] Osher, S. and Sethian, J.A., *Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*, Journal of Comput. Phys., vol. 79, n. 1, pp. 12-49, (1988).

[15] Shu, C.W. and Osher, S., *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes II (two)*, Journal of Computational Physics; Volume 83, (1989), pp 32-78.

[16] Sussman, M., Smereka, P. and Osher, S., *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys., v. 114, (1994), pp. 146-154.

[17] Wardlaw, A., "Underwater Explosion Test Cases", IHTR 2069, 1998

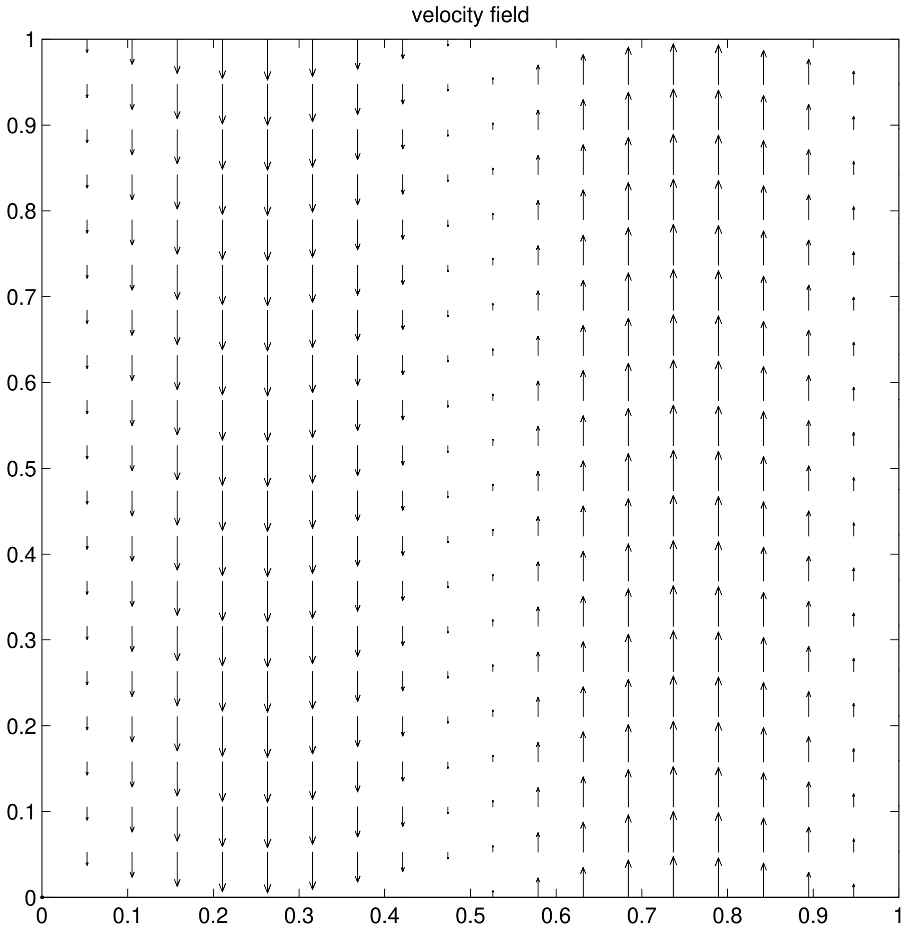

Figure 1: Initial velocity field

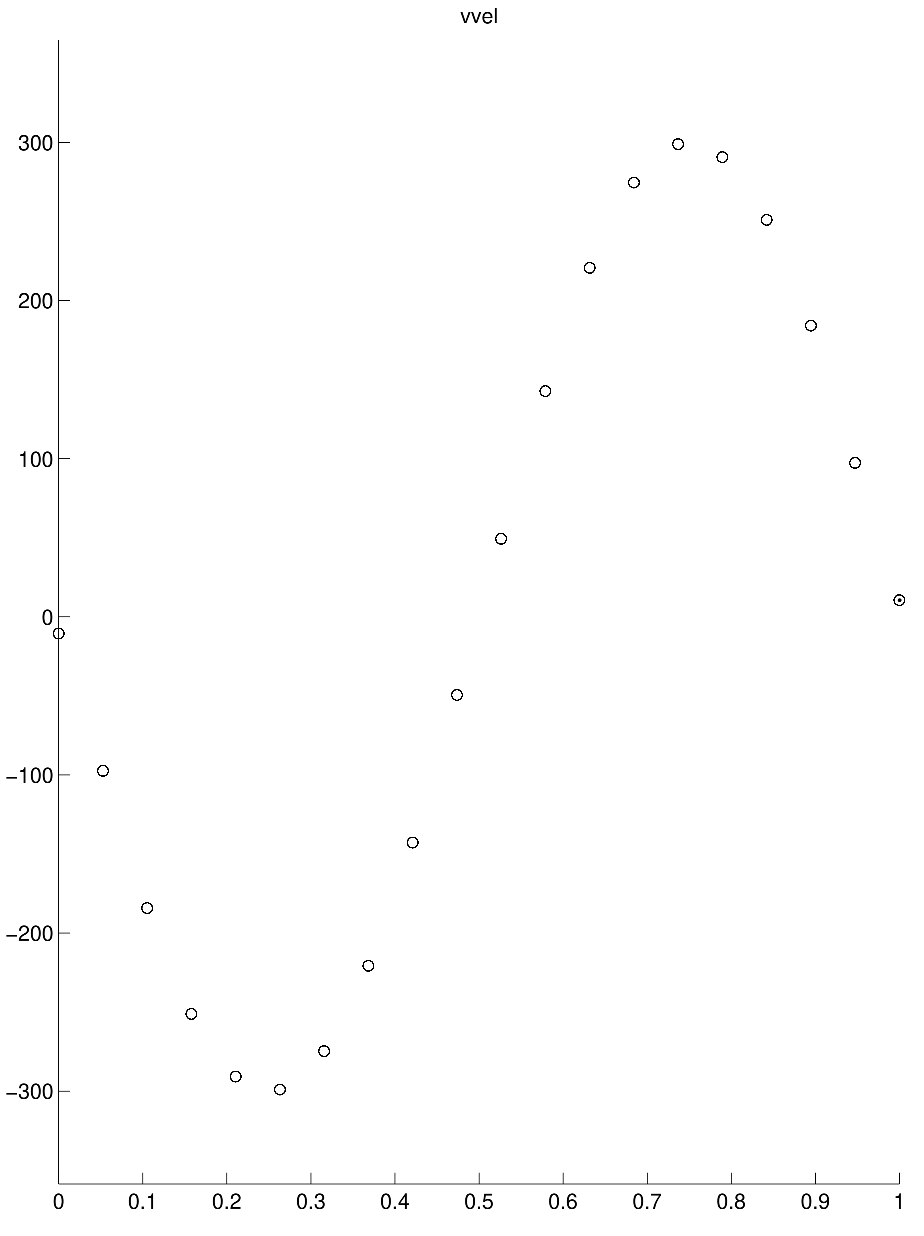

Figure 2: Initial tangetial velocity profile

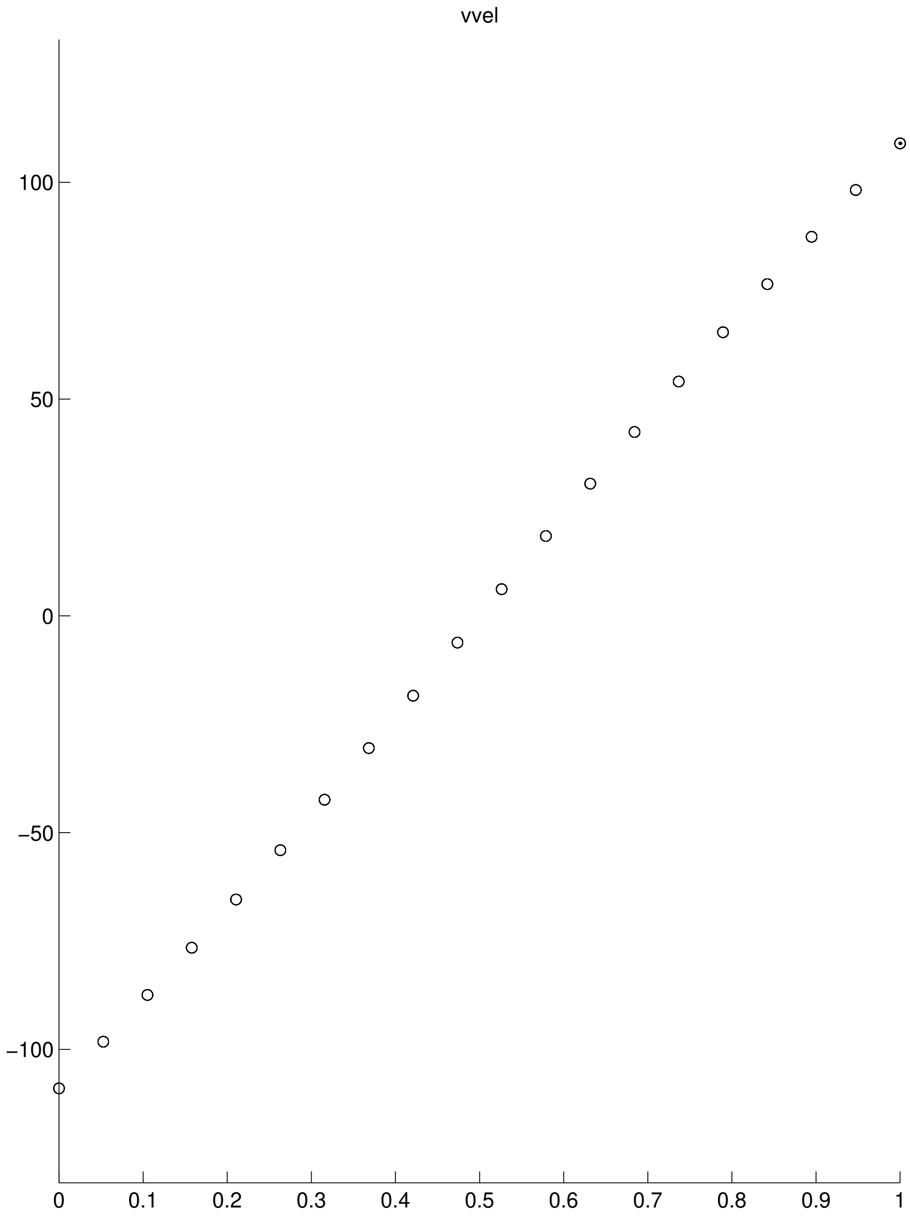

Figure 3: $\mu_{left} = \mu_{right} = 100$, "no-slip"

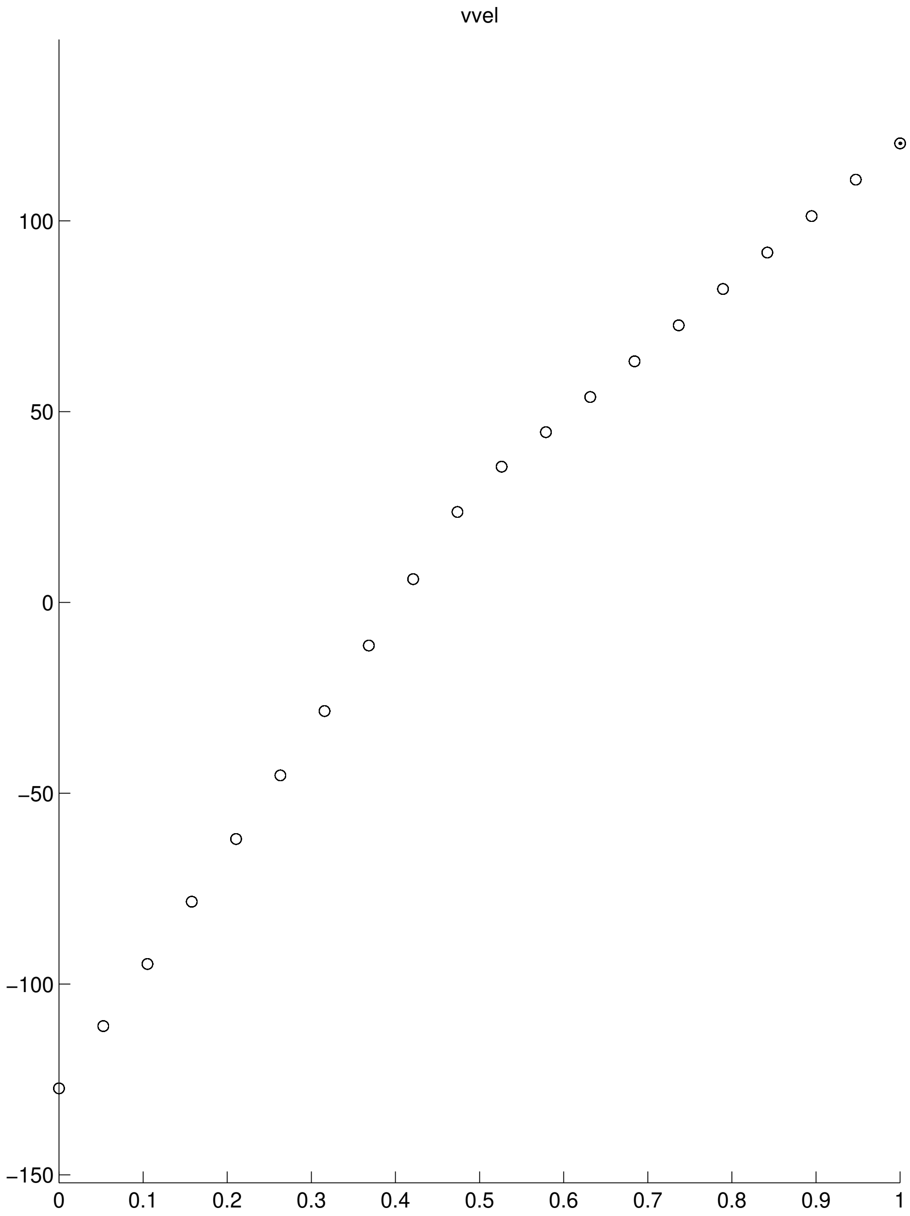

Figure 4: $\mu_{left} = 50$, $\mu_{right} = 100$, “no-slip”

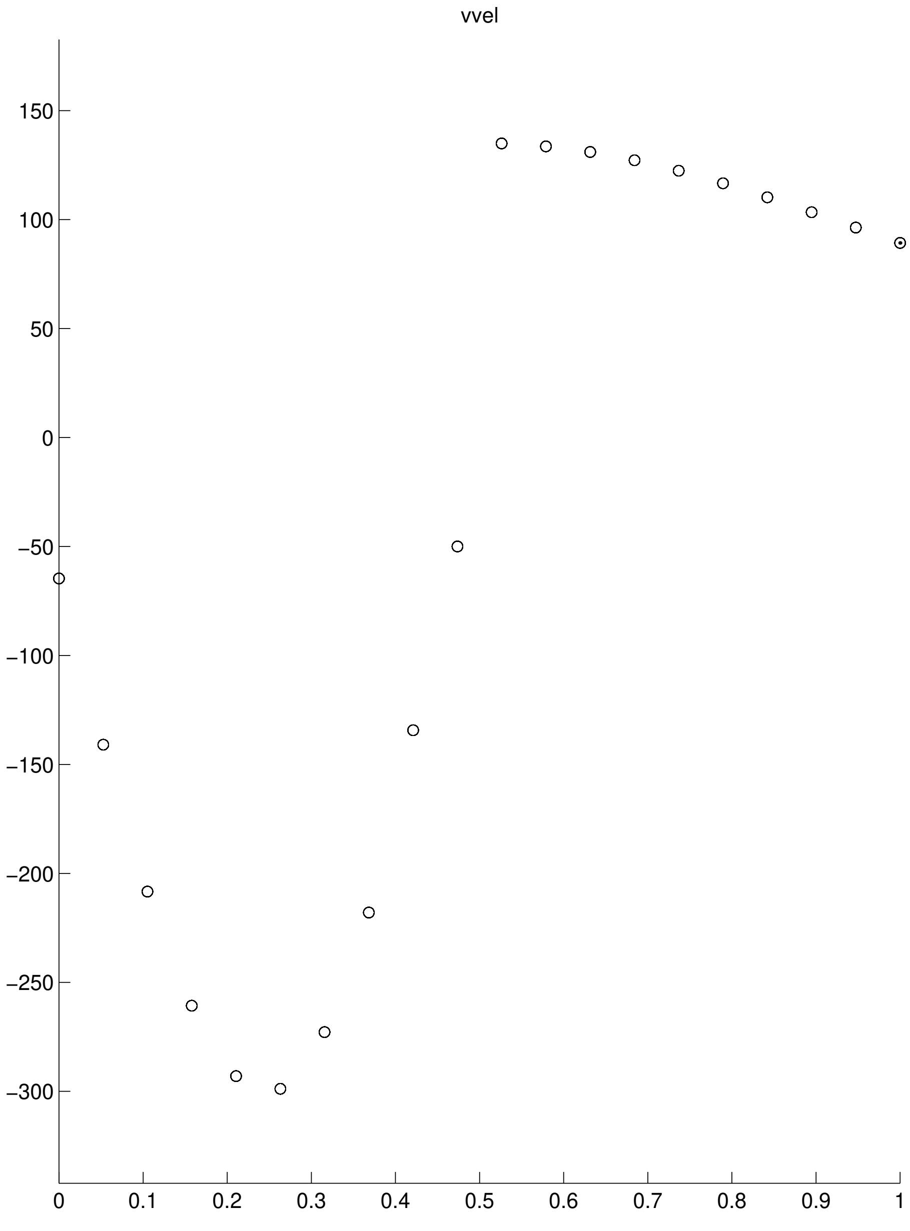

Figure 5: $\mu_{left} = 0$, $\mu_{right} = 100$, "slip"

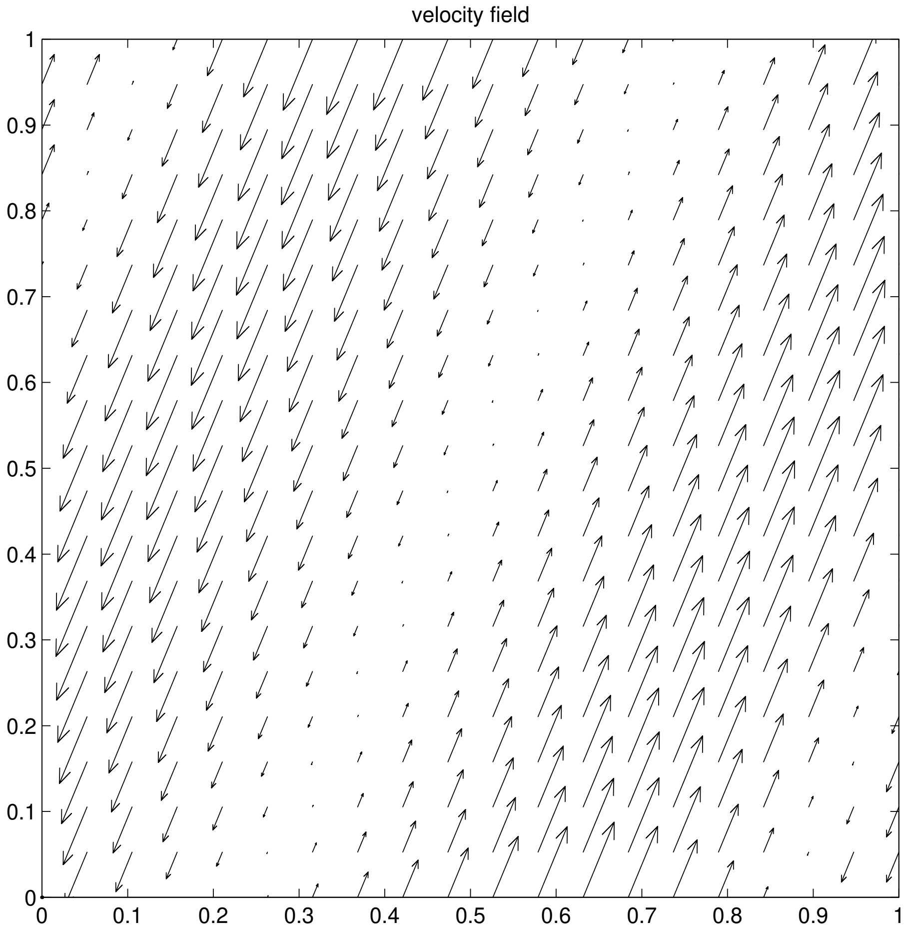

Figure 6: Initial velocity field

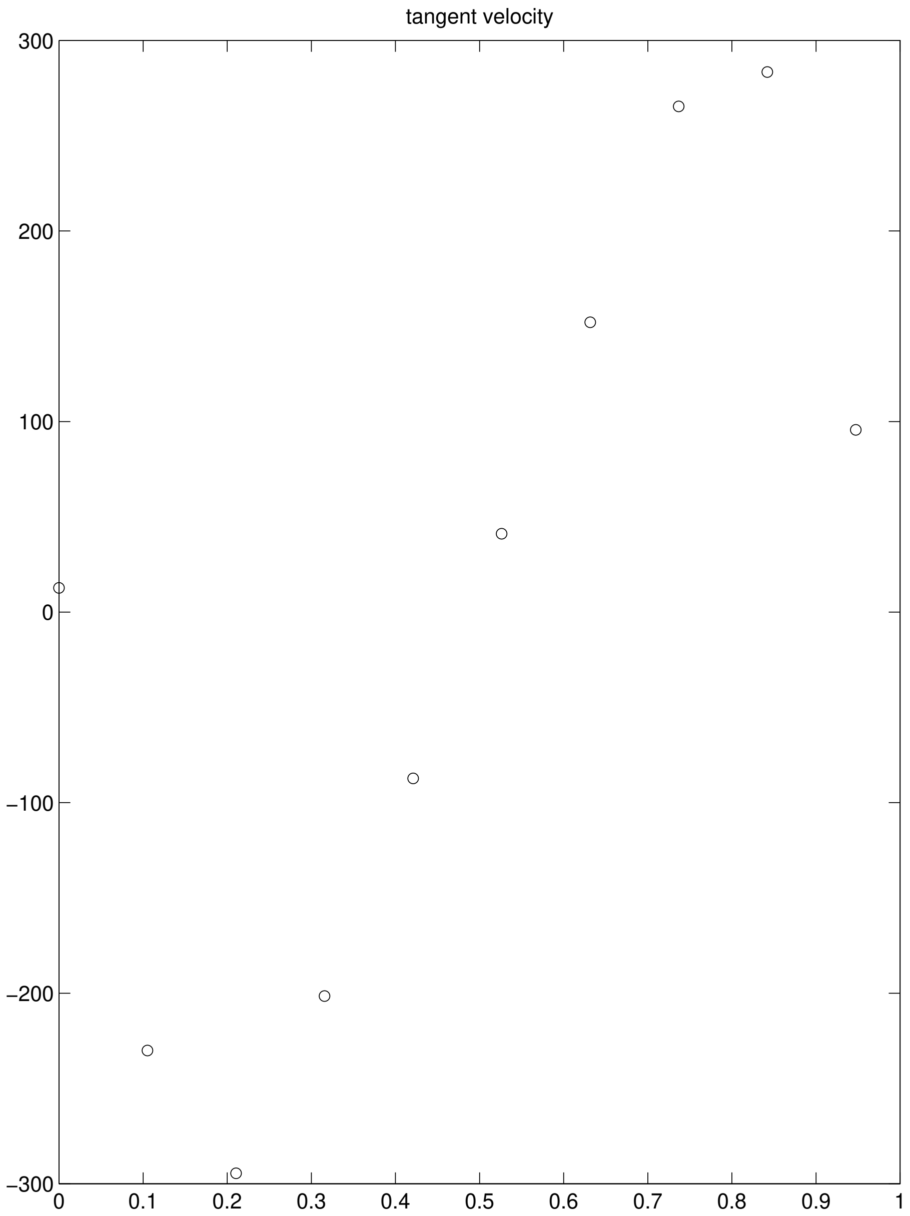

Figure 7: Initial tangetial velocity profile

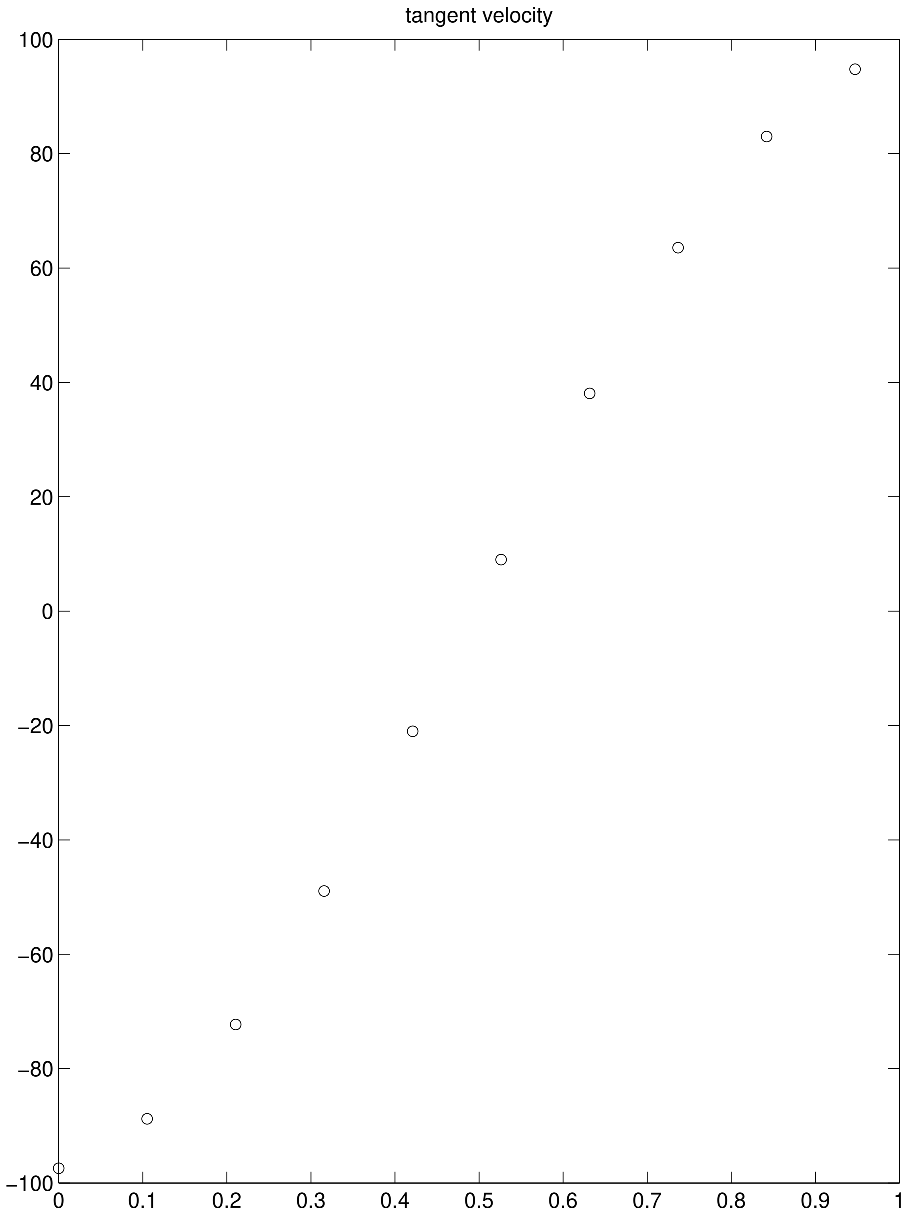

Figure 8: $\mu_{left} = \mu_{right} = 100$, “no-slip”

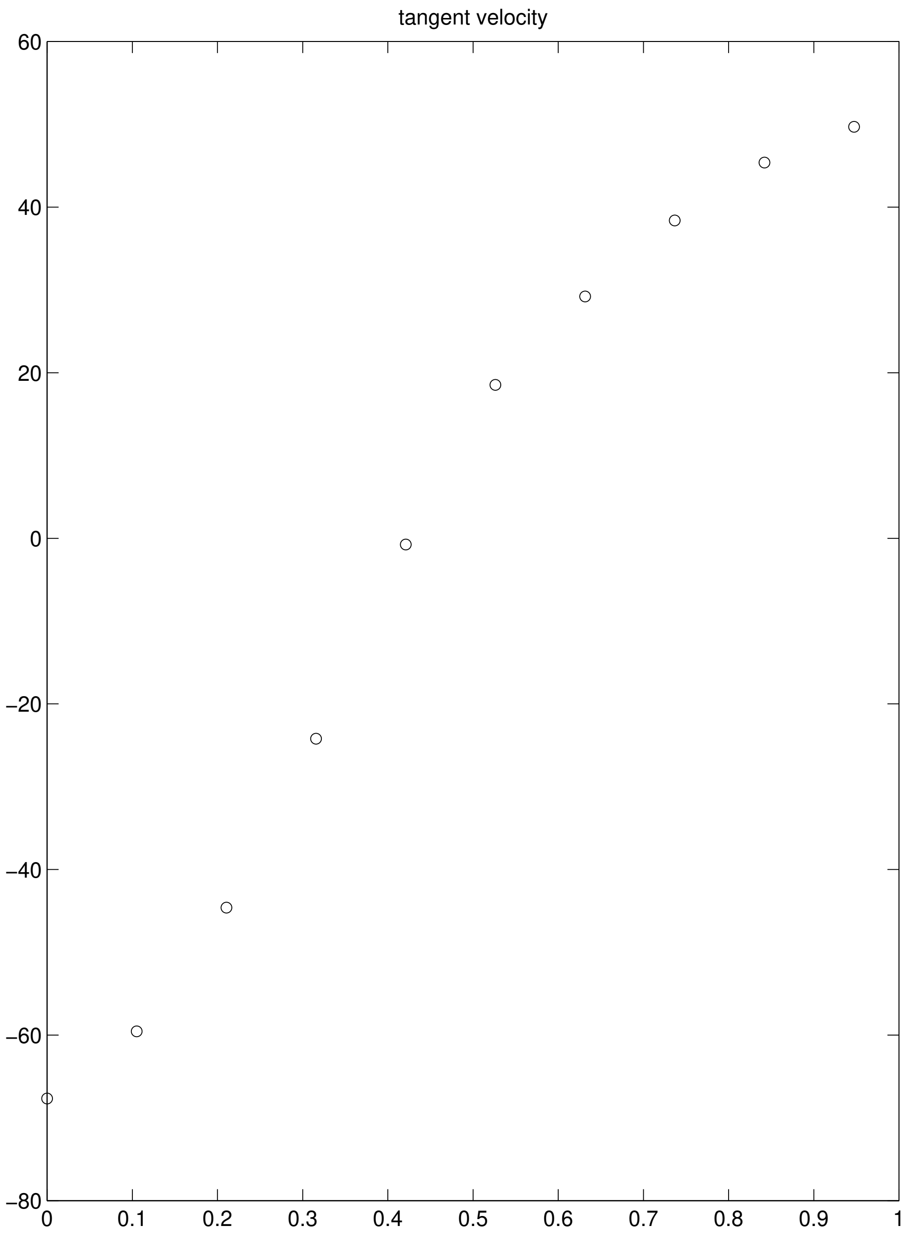

Figure 9: $\mu_{left} = 50$, $\mu_{right} = 100$, “no-slip”

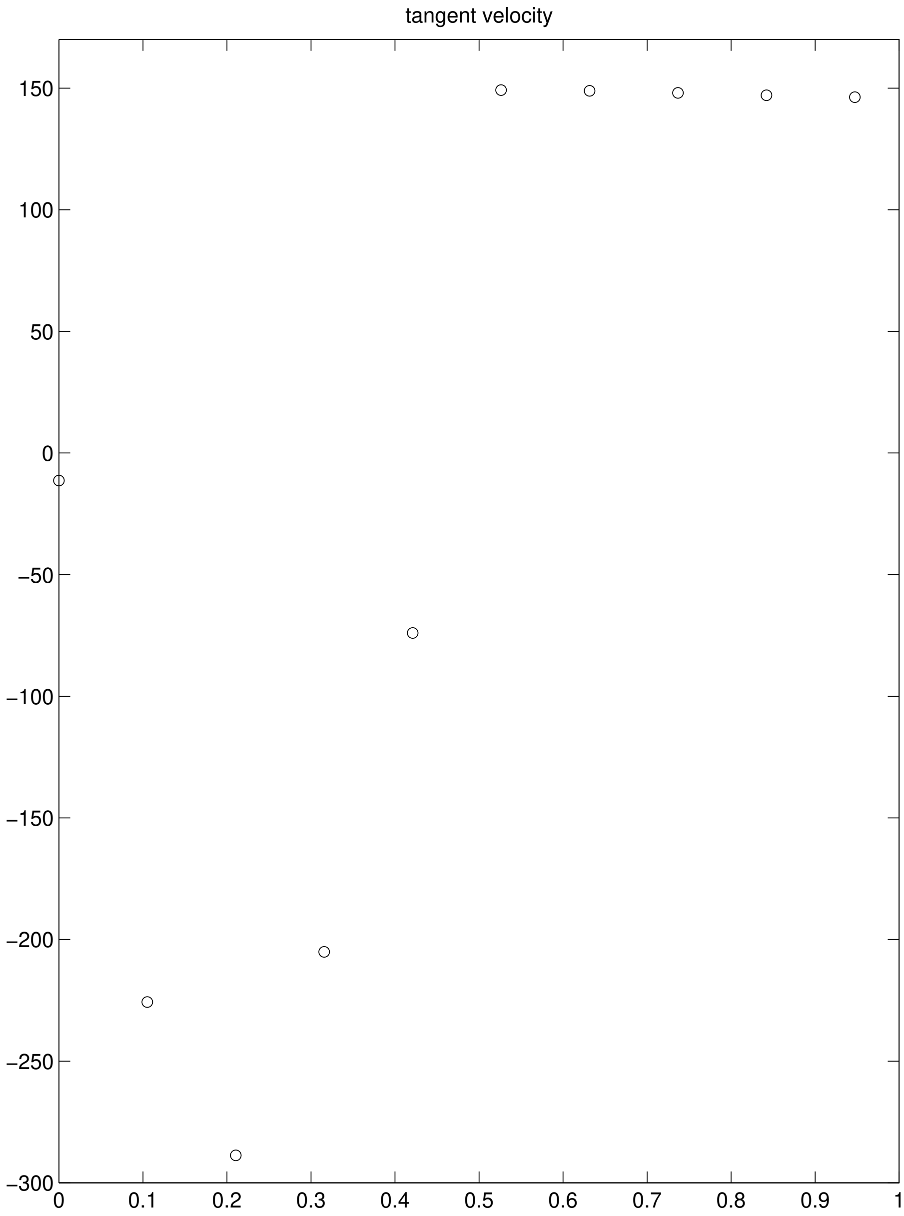

Figure 10: $\mu_{left} = 0$, $\mu_{right} = 100$, “slip”

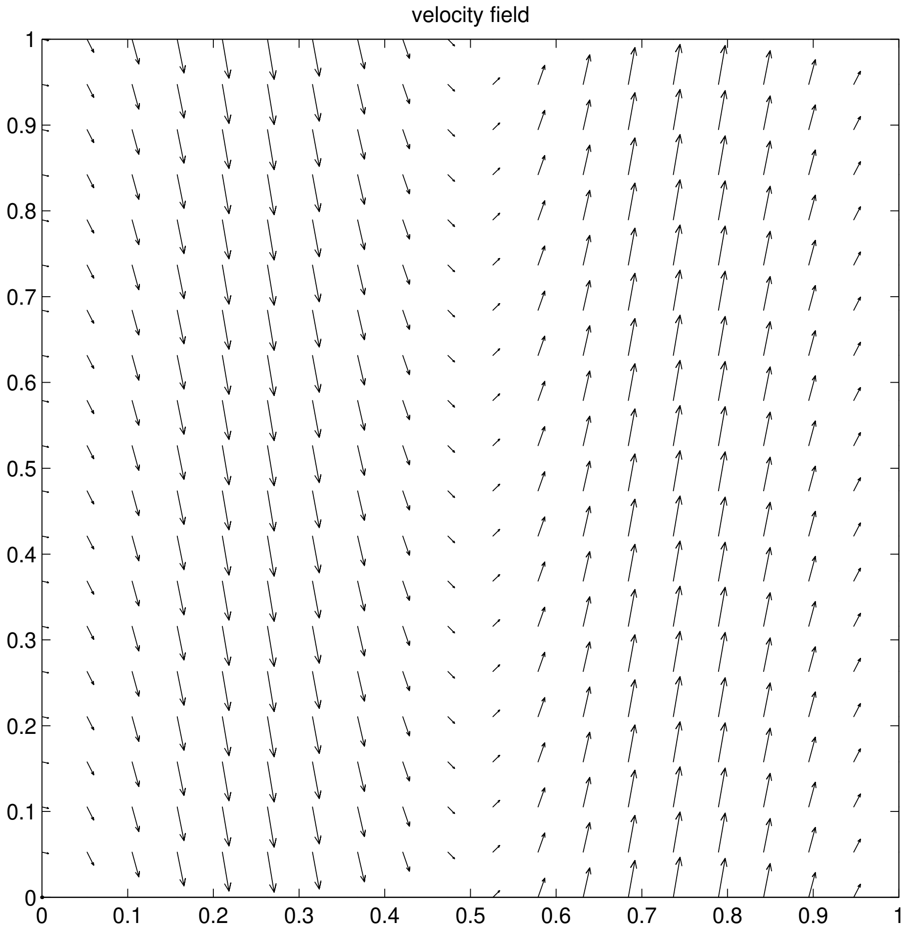

Figure 11: Initial velocity field

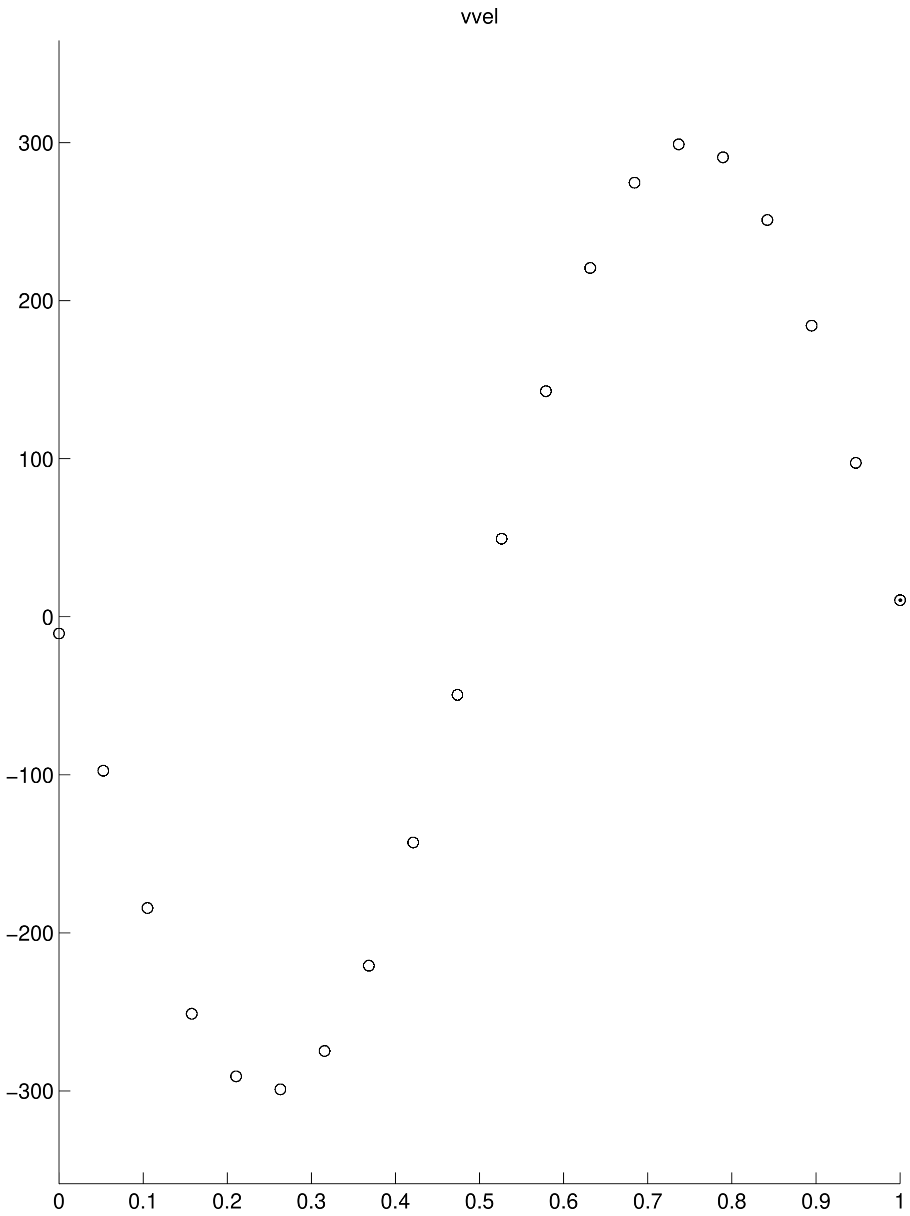

Figure 12: Initial tangetial velocity profile

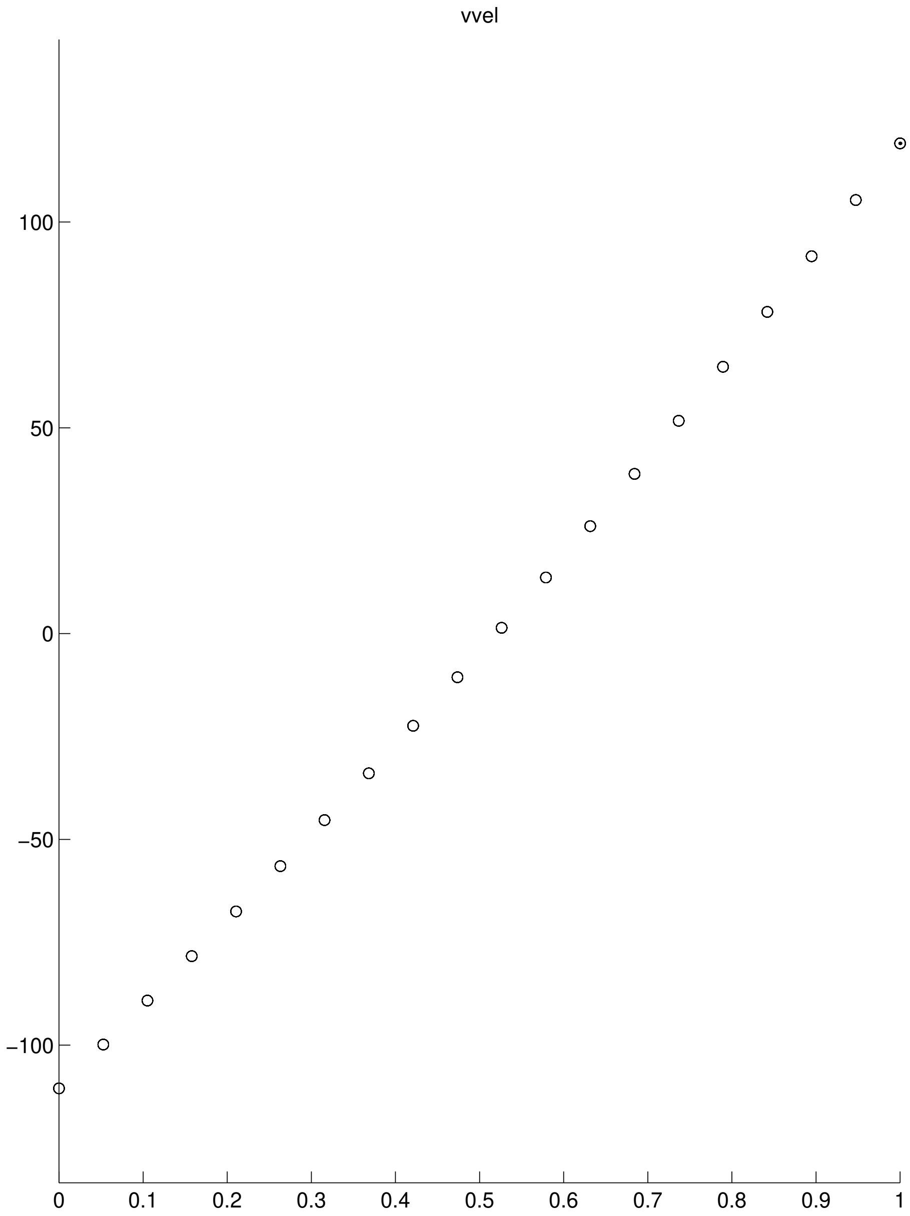

Figure 13: $\mu_{left} = \mu_{right} = 100$, "no-slip"

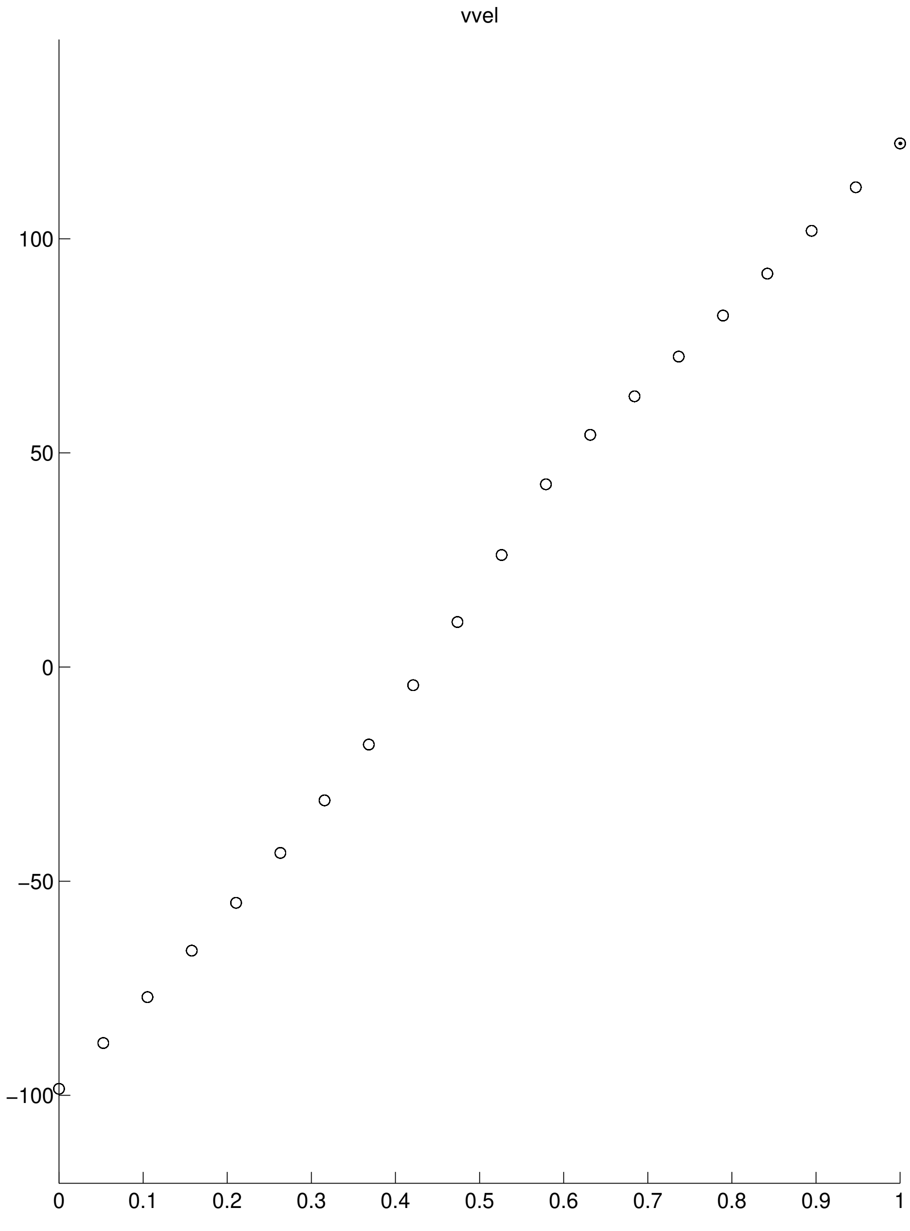

Figure 14: $\mu_{left} = 50$, $\mu_{right} = 100$, "no-slip"

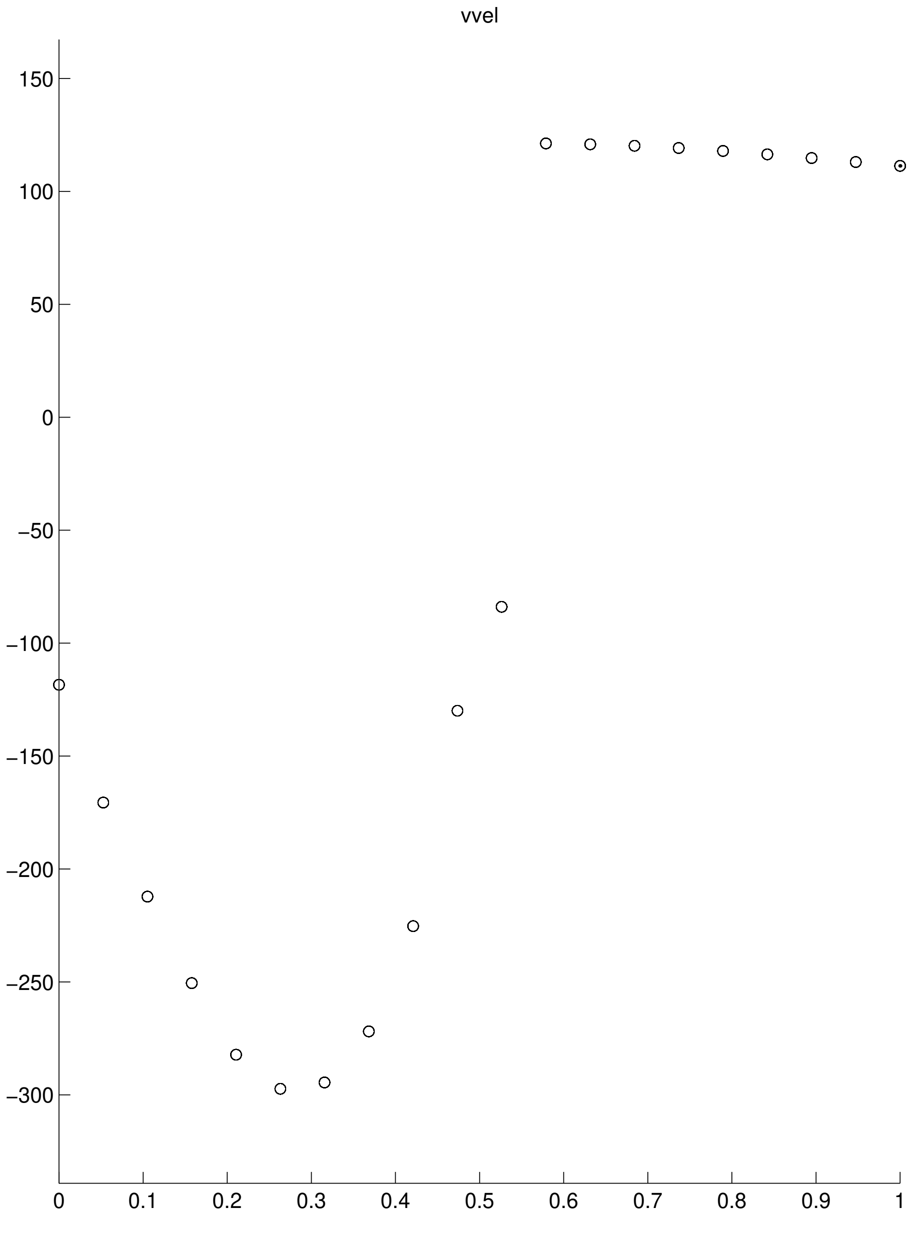

Figure 15: $\mu_{left} = 0$, $\mu_{right} = 100$, “slip”