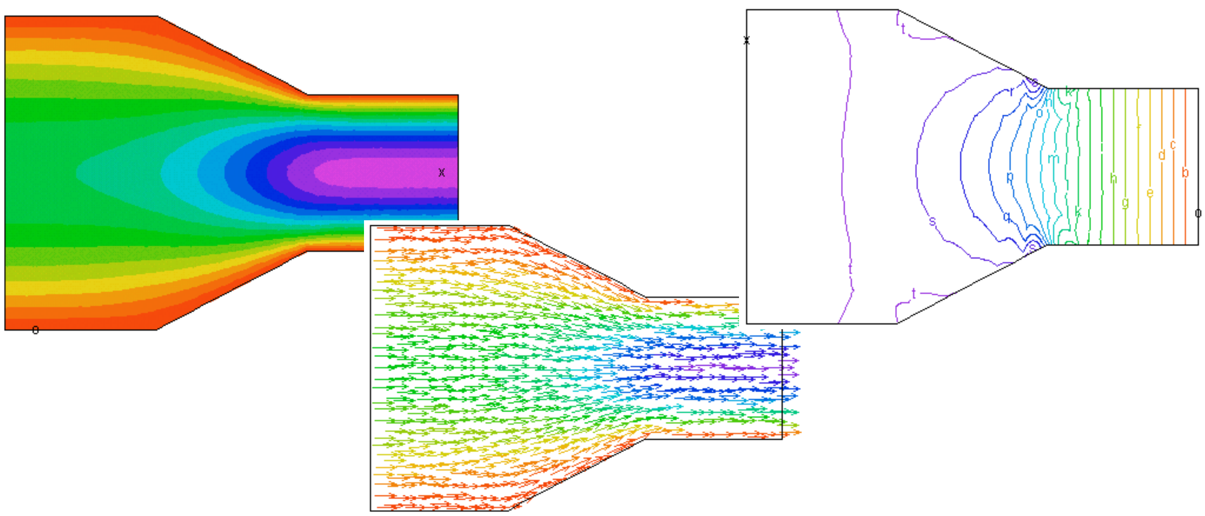
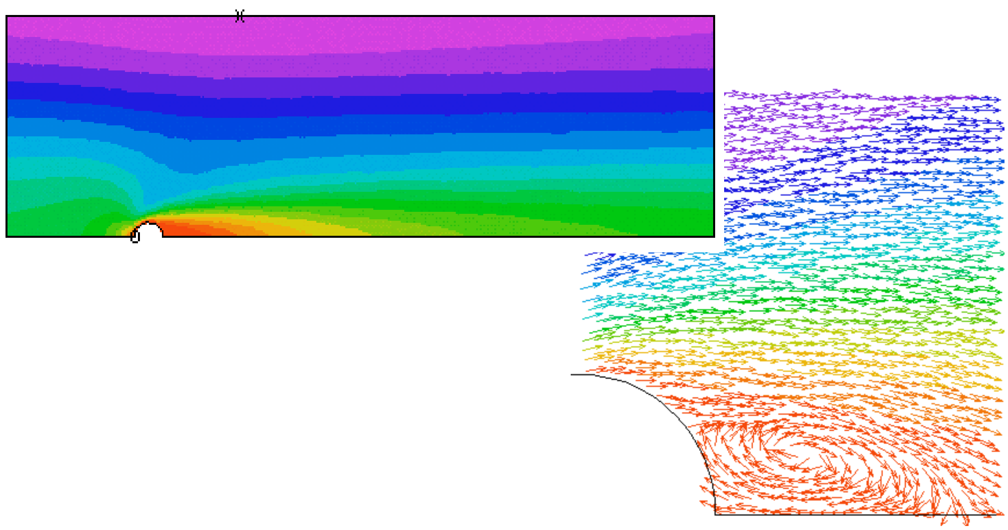
Fluid Dynamics

by Finite Element Analysis

Irrotational and Viscous Flow in 2D and 3D

Using FlexPDE Version **5**





GB Publishing

Gunnar Backstrom

How to Start Reading

Double-click on the PDF icon to start the Adobe® Acrobat® Reader.
View the list of contents by pushing the F5 keyboard button.
Toggle by F5 to view the text only.

Preface

This book is a sequel to the e-book *Deformation and Vibration by Finite Element Analysis*. The present volume hence starts with Chapter 18. Using the same software (FlexPDE version 5) it expands the applications to irrotational and viscous flow of incompressible fluids.

The preceding part started with an introductory chapter on graphical facilities, which may be studied without applying boundary conditions and without solving any PDE. There seems to be no reason to repeat this material here, and hence it is omitted.

As before, there is no *index* since the *Acrobat* program lets you search for words and even word combinations. After selecting *Edit*, *Find* (or pushing the keys *Ctrl+f*) if suffices to enter the item of interest. The table of contents is also available and may be brought up to the left of the text by clicking on *Bookmarks* (or by pushing *F5*). A simple click on a subtitle opens that section immediately.

Since this is the last of the four *Fields* volumes, I should again like to thank my late friend Dr. Russell Ross, University of East Anglia, for reading and commenting the work. The admirable programmer behind FlexPDE, Mr. Bob Nelson, kindly continued to support this final round of applications.

Gunnar Backstrom

The finite-element software package used for this book (FlexPDE®) is marketed by

PDE Solutions Inc
PO Box 4217, Antioch, CA 94531-4217, USA
Phone: +1 925 776 2407 Fax: +1 925 776 2406
Email: sales@pdesolutions.com
http://www.pdesolutions.com

Contents

18 Irrotational Flow of Liquids in (x,y)

This is the second volume on mechanical fields, and the introductory chapters on graphics, Laplace, and Poisson equations will not be repeated here. Instead, we occasionally refer to the preceding book for elementary details.

Since the density of a liquid normally changes little within the range of pressures occurring in practical applications, we assume the density to be constant. In this chapter we also make more daring assumptions, i.e. that the liquid slips freely over solid surfaces and that viscous forces are vanishingly small compared to inertial forces. These assumptions are known to be useful, however, in many situations.

The conservation of mass may be expressed as[8p52]

$$\nabla \cdot (\rho_0 \mathbf{v}) = -\frac{\partial \rho_0}{\partial t}$$

where ρ_0 is the mass density and $\mathbf{v}$ the velocity vector. Assuming constant density, this leads us to the conservation of volume

$$\nabla \cdot \mathbf{v} = 0$$

or in explicit form

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = 0 \qquad \bullet$$

This PDE is not of 2nd order, which is a prerequisite for solving it by FlexPDE. Fortunately, we may arrange this by expressing the velocity components as derivatives of a common function ϕ. Hence, let us choose the definitions

$$v_x = \frac{\partial \phi}{\partial x}, \quad v_y = \frac{\partial \phi}{\partial y}$$

In this manner we arrive at

$$\frac{\partial^2\phi}{\partial^2 x}+\frac{\partial^2\phi}{\partial^2 y}=0 \qquad \bullet$$

which is the well known Laplace equation (see Chapter 5 in *Deformation and Vibration*).

So far, we have only used the principle of conservation of mass, but it is important to note that any solution to the above PDE will also be irrotational ($\nabla\times\mathbf{v}=0$), because

$$(\nabla\times\mathbf{v})_z=\frac{\partial v_y}{\partial x}-\frac{\partial v_x}{\partial y}=\frac{\partial^2\phi}{\partial x\partial y}-\frac{\partial^2\phi}{\partial y\partial x}=0$$

Energy conservation next leads us to the Bernoulli equation of motion, which states[8p116]

$$\frac{1}{2}\rho_0 v^2+p+\rho_0 g\,y=\text{constant} \qquad \bullet$$

where $v=\sqrt{v_x^2+v_y^2}$ is the magnitude of the velocity (speed), p the pressure, and g the acceleration due to gravity (assuming the y-axis to be vertical).

Flow through a Constricted Channel

Our first application of the above equations will be to the flow through a horizontal channel, limited by plane surfaces perpendicular to our domain.

The following descriptor defines the problem and introduces the PDE for the velocity potential ϕ. After solving for phi we simply differentiate to obtain the components of velocity. Having obtained these components we then form the magnitude of the velocity (v). Assuming horizontal flow, where the gravity terms cancel, the Bernoulli equation gives us

$$\frac{1}{2}\rho_0 v^2+p=\frac{1}{2}\rho_0 v_0^2+p_0$$

and we finally obtain the expression for the pressure p included in the *definitions* segment.

In order to make optimum use of the adaptive gridding provided by the program, we specify the modest initial ngrid=1. The *Student Version* of FlexPDE is sufficient for solving this problem.

In the *boundaries* segment we specify the input velocity vx0 by a natural statement (Chapter 5 in the preceding volume). For the output end we just impose a constant value for the potential phi. Its absolute value is of course arbitrary since only the derivatives will be used, but by specifying a constant value over this boundary we also stipulate that vy=dy(phi) is to vanish, i.e. we force the liquid to exit in the *x* direction.

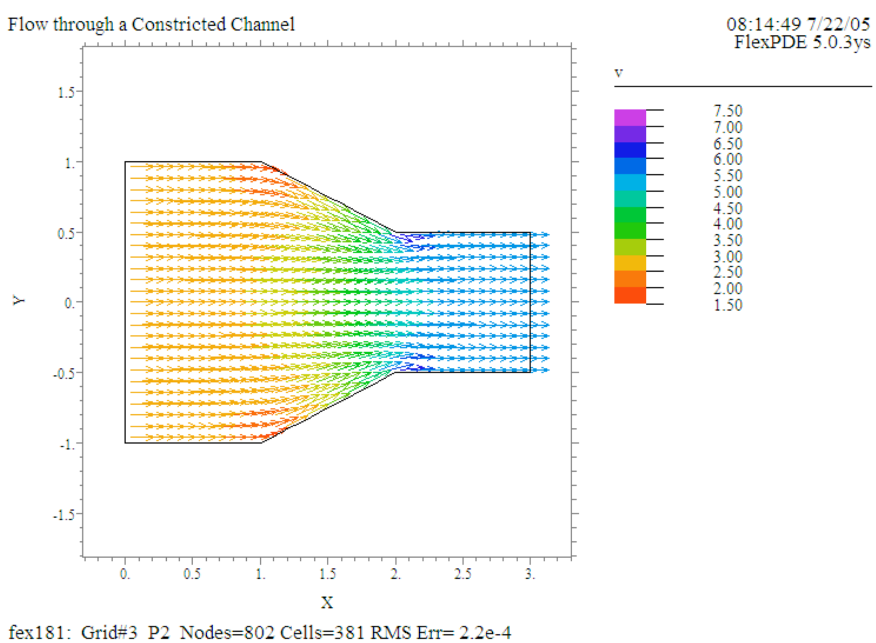
If you are not already familiar with FlexPDE graphics, you should refer to the introductory chapters in *Fields of Physics* or *Deformation and Vibration*. Also note the *Help* facility included in the program.

```
TITLE    'Flow through a Constricted Channel'             { fex181.pde }
SELECT    errlim=1e-5    ngrid=1    spectral_colors      { Rainbow }
                                                          { Student Version }
VARIABLES    phi                                          { Velocity potential }
DEFINITIONS                                               { SI units }
  Lx=1    Ly=1
  coef=0.5                                                { Constriction coefficient }
  vx0=3.0                                                 { Velocity at input end }
  p0=1e5                                                  { Atmospheric pressure }
  dens=1e3                                                { Mass density }
  vx=dx(phi)    vy=dy(phi)                                { Velocity components }
  v=vector( vx,vy)    vm=sqrt( vx^2+ vy^2)                { Speed }
  p=p0+ 1/2*dens*(vx0^2-vm^2)                             { Pressure }
  div_v=dx( vx)+ dy( vy)                                  { Divergence, or div( v) }
  curl_z=dx( vy)- dy( vx)                                 { Vorticity, or curl( v) }
EQUATIONS
  dxx( phi)+ dyy( phi)=0                                  { Or div( grad( v)) }
BOUNDARIES
region 'domain' start 'outer' (0,Ly)
  natural( phi)=-vx0  line to (0,-Ly)                     { In }
  natural( phi)=0 line to (Lx,-Ly)  to (2*Lx,-Ly*coef)  to (3*Lx,-Ly*coef)
  value( phi)=0 line  to (3*Lx,Ly*coef)                   { Out }
  natural( phi)=0 line to (2*Lx,Ly*coef) to (Lx,Ly) to close
PLOTS
  contour( phi)    vector( v) norm    contour( vm) painted
  contour( p) painted    contour( p) zoom(1.5*Lx,0,  Lx,Ly)
  surface( p) zoom(1.5*Lx,0,  Lx,Ly)
  elevation( vm) on 'outer'                               { Verify boundary conditions }
```

```
  contour( div_v)     contour( curl_z)
END
```

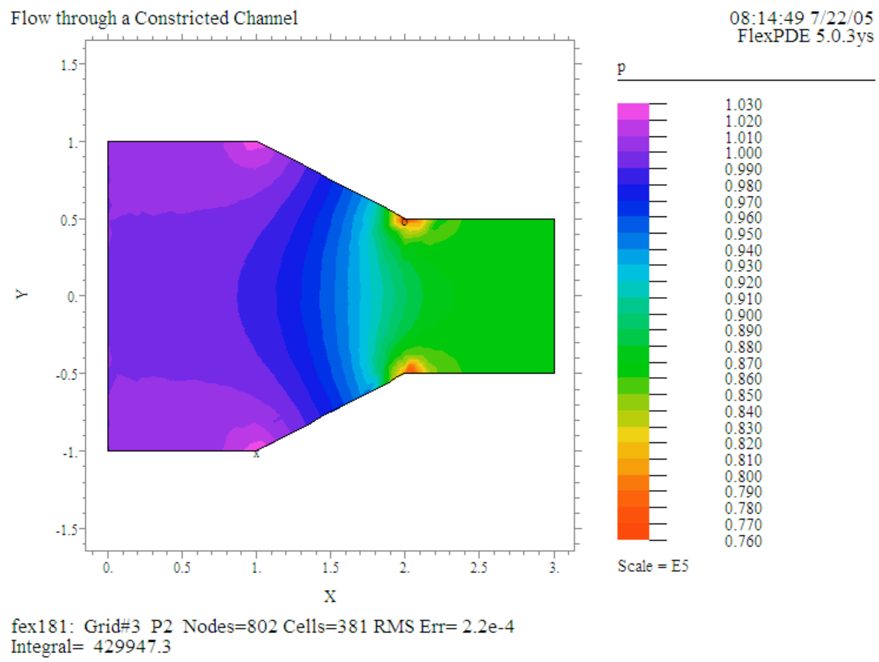
The modifier norm here follows the vector plot command. This means that the arrows will be normalized to a standard length, but the color code indicates the magnitude, i.e. the speed. This plot shows that the speed is constant across the ends and that there is an increase from input to output. The elevation plot on the boundary shows this fact more clearly.

The vector plot below thus represents the velocity field. We notice that the streamlines are parallel to the boundaries, where they come close, but the speed does not vanish there. We also notice that the speed distribution at the exit appears to be roughly twice that at the entrance.

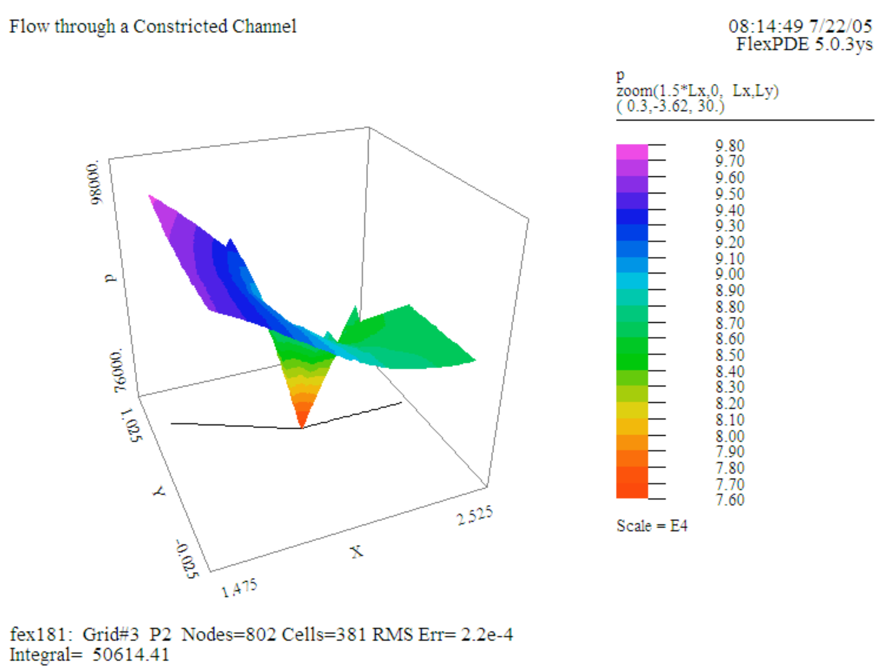


We also see from the plot of vm (not shown here) that the speed increases by a factor of about two from input to output, i.e. in inverse proportion to the channel width. This is of course in accord with the conservation of mass and volume, a principle that was incorporated into the PDE by the vanishing divergence. Notice, however, that we did not explicitly introduce this constancy in the boundary conditions at the ends, although we could well have done so.

The following is a painted contour plot of the pressure. It shows that the pressure variation is small close to the input and output ends.



Other facts to notice are that the speed vm (not shown) takes a minimum at the corner where the width of the channel starts to decrease and a maximum where it becomes constant again.



From the above surface plot of p it is evident that the pressure has a minimum at the corner where the speed peaks. The color code indicates that this minimum is about 70% of the value at the input end, as gathered from the un-zoomed plot. From the latter plot we also deduce that the pressure decreases from input to output, but not in proportion to the width.

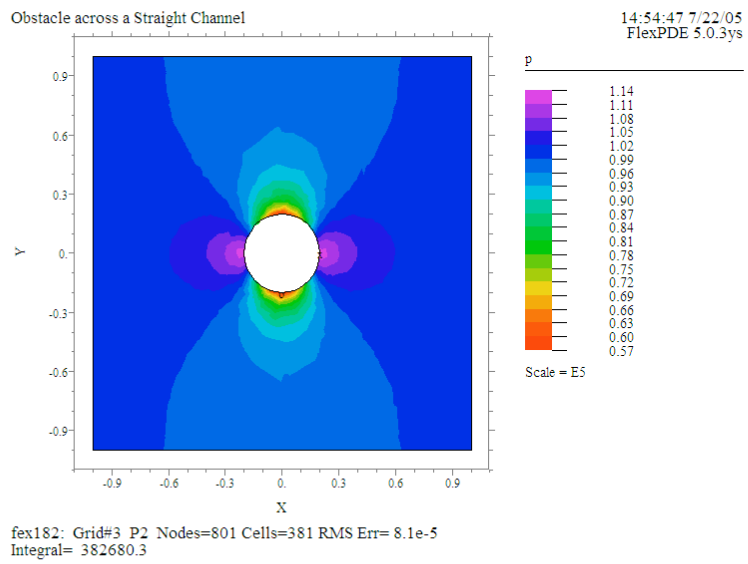
The last two plots demonstrate that the divergence as well as the curl of the velocity field vanishes.

Cylindrical Obstacle across a Straight Channel

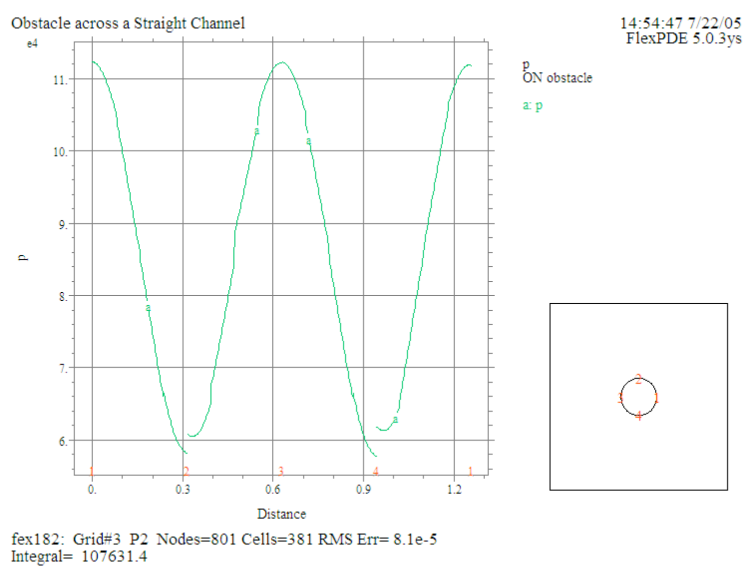
We shall next consider flow around an obstacle, and in particular the forces exerted on it by the stream. In the descriptor, which is based on *fex181*, we introduce a bar of circular cross-section across the channel.

```
TITLE    'Obstacle across a Straight Channel'              { fex182.pde }
SELECT    errlim=1e-4    ngrid=1    spectral_colors
VARIABLES    phi                    { Velocity potential }
DEFINITIONS
  Lx=1.0    Ly=1.0    a=0.2
  vx0=5.0                           { x-component of velocity at left end }
  p0=1e5                            { Atmospheric pressure at left end }
  dens=1e3                          { Mass density }
  vx=dx( phi)    vy=dy( phi)        { Velocity components }
  v=vector( vx,vy)    vm=magnitude( v)
  p=p0+ 0.5*dens*( vx0^2- vm^2)  { Pressure }
EQUATIONS
  div( grad( phi))=0
BOUNDARIES
region 'domain'
  start 'outer' (-Lx,Ly) point value( phi)= 0
  natural( phi)=-vx0  line to (-Lx,-Ly)                    { In }
  natural( phi)=0 line to (Lx,-Ly)                         { Wall }
  natural( phi)=vx0 line  to (Lx,Ly)                       { Out }
  natural( phi)=0 line to close                            { Wall }
  start 'obstacle' (a,0)                                   { Cut-out }
  natural( phi)=0  arc( center=0,0) angle=360  close
PLOTS
  contour( vm) painted    vector( v) norm
  vector( v) norm zoom(-3*a/2,-a/2,  2*a,2*a)
  contour( p) painted    elevation( p) on 'obstacle'
END
```

Since the liquid is assumed non-viscous, drag forces could only be caused by the pressure distribution. The plot below suggests that the pressure is symmetric with respect to both axes. From this symmetry we would expect the upstream and downstream forces to be equal and oppositely directed.



The elevation plot below presents the pressure variation on the surface of the obstacle. The left-right symmetry, as well as the up-down symmetry, is clearly evident from this figure.



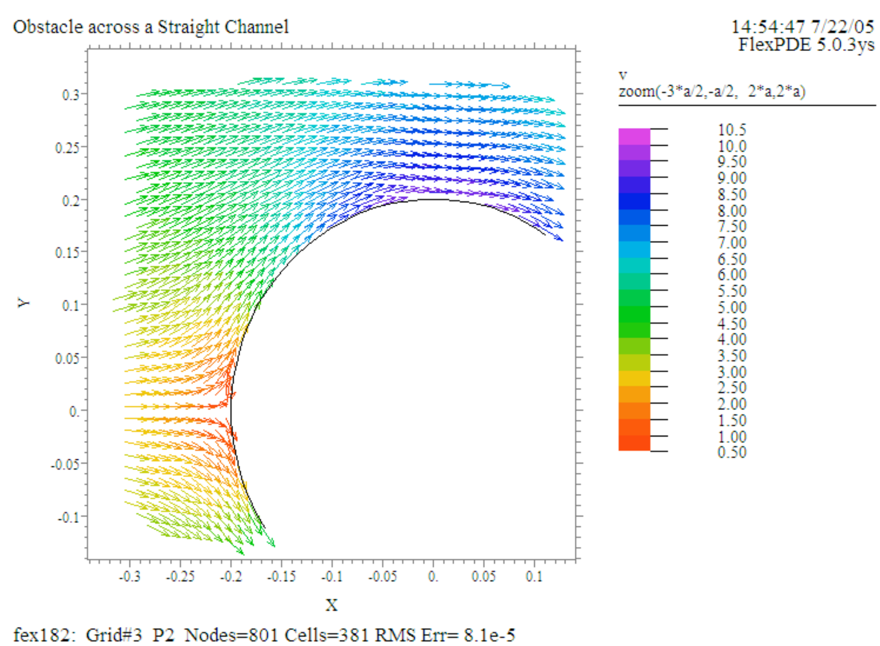
To the left and also to the right of the obstacle (points 1 and 3), we find pressure values larger than the ambient value (1e5). We could also calculate this maximum value directly from the Bernoulli equation (p.227●2)

$$\frac{1}{2}\rho v^2 + p = \frac{1}{2}\rho v_0^2 + p_0$$

for a point of flow stagnation $(v = 0)$, obtaining the result p=1.125e5.

The pressure on the sides parallel to the mainstream (points 2 and 4) is much lower than the ambient value. This pressure reduction is a well-known consequence of the Bernoulli equation.

The left-right symmetry of the pressure plot indicates the absence of a force dragging the object along the stream. This may be surprising at first. As is apparent from the following vector plot, however, the incoming flow deviates to become parallel to the front face, but the acceleration required is equal and opposite to that required for making the stream parallel again on the opposite side.



The plot also illustrates the phenomenon of stagnation. The color serves to indicate the magnitude. Here, the speed vanishes at $y = 0$ on a line perpendicular to the figure.

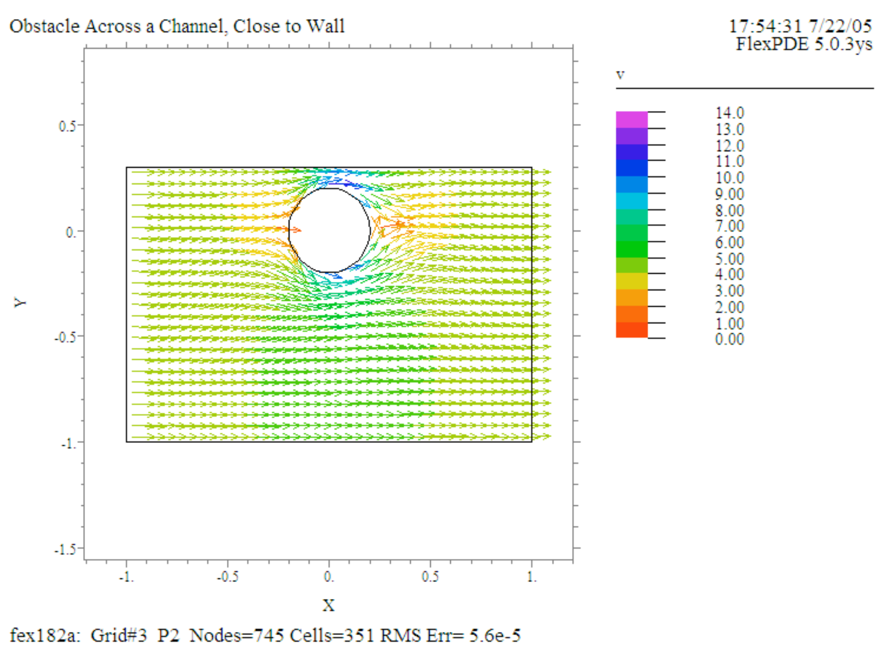
It is clear from the above (symmetric) plots of p that the pressure forces on the liquid sum to the value zero. Since the liquid slips on the boundaries, the total force vanishes, and hence the force on the obstacle.

Even if there is no resultant force on the cylinder, we do find excess pressure on the left and right sides and a deficit at the bottom and top sides. Hence, if the obstacle were elastic it would deform.

Obstacle Close to a Wall

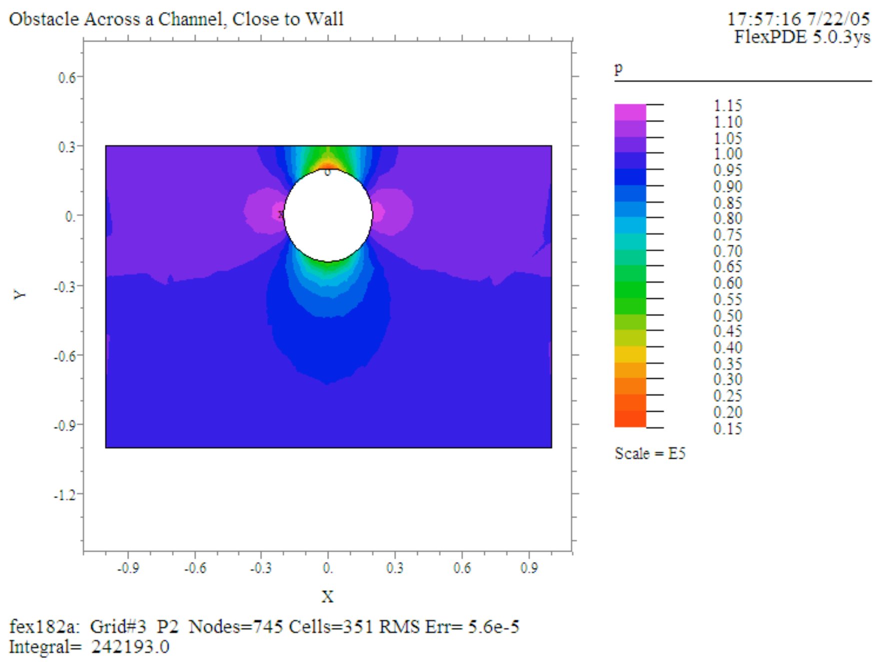
It is easy to modify *fex182* to make the upper boundary line come closer to the obstacle. The changes are evident from the following lines.

```
TITLE     'Obstacle Across a Channel, Close to Wall'        { fex182a.pde }
...
region 'domain'
   start 'outer'(-Lx,0.3*Ly) point value( phi)=0
   natural( phi)= -vx0      line to (-Lx,-Ly)
   natural( phi)=0      line to (Lx,-Ly)
   natural( phi)=vx0      line  to (Lx,0.3*Ly)
   natural( phi)=0 line to close                    { Keep 'obstacle' below }
...
   elevation( p) from (-Lx,-Ly) to (Lx,-Ly)
   elevation( p) from (-Lx,0.3*Ly) to (Lx,0.3*Ly)
END
```



The above vector plot shows the flow pattern in this case.

As is evident from the following plot, p is still left-right symmetric, but the pressure is now lower on the top than at the bottom of the cylinder. From this it is clear that an upward force acts on the obstacle.



Evidently, the pressure pushes the obstacle toward the nearby wall. This effect is vaguely analogous to the suction felt when you stand close to a passing train.

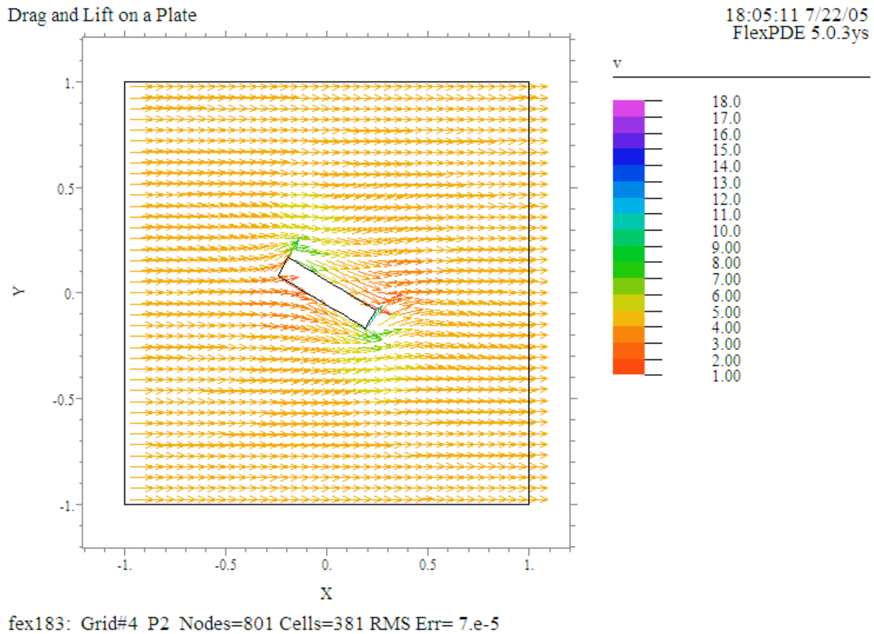
The elevation plots present the pressure on the top and bottom sides of the domain, and from these we may read off the integrals, which are equal to the forces on the liquid, caused by the cylinder. The conclusion is that the force on the latter is 195527-188798=6729.

Drag and Lift on an Inclined Plate

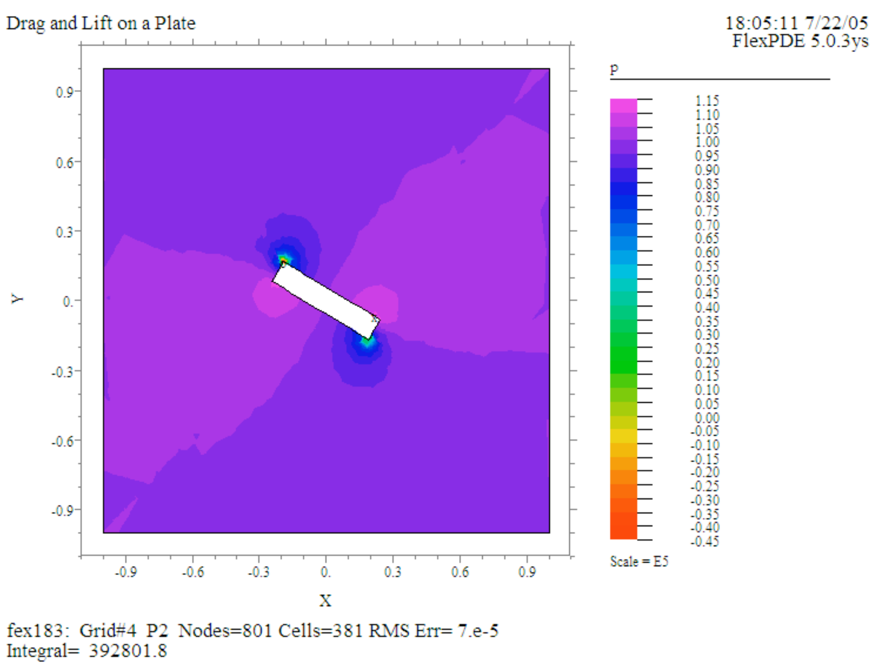
Let us now turn to a situation where we all know from experience that a lifting force may occur, both in air and in water. The geometry should be clear from the figure below. As before, we have a stream of liquid from left to right with constant velocity at the vertical boundaries. In the following descriptor the obstacle is a rectangular plate at an angle of attack (alpha) with respect to the main stream. The general expressions for the corner coordinates of the plate permit us to change the angle of attack at will.

```
TITLE     'Drag and Lift on a Plate'                              { fex183.pde }
SELECT     errlim=1e-4     ngrid=1     spectral_colors
VARIABLES     phi                        { Velocity potential }
DEFINITIONS
   Lx=1.0     Ly=1.0     a=0.5*Ly     d=0.2* a
   vx0=5.0                               { x-component of velocity at left end }
   alpha=30* pi/180                      { Angle of attack, radians }
   si=sin( alpha)     co=cos( alpha)
   x1=-d/2*si- a/2*co     y1=-d/2*co+ a/2*si       { Corner coordinates }
   x2=d/2*si- a/2*co     y2=d/2*co+ a/2*si
   x3=-x1     y3=-y1     x4=-x2     y4=-y2
   p0=1e5                                { Atmospheric pressure at left end }
   dens=1e3                              { Mass density }
   vx= dx(phi)     vy= dy(phi)           { Velocity components }
   v=vector( vx, vy)     vm=magnitude( v)
   p=p0+ 0.5*dens*(vx0^2-vm^2)    { Pressure }
   brute_force=p0* 2*y2
EQUATIONS
   dxx( phi)+ dyy( phi)=0
BOUNDARIES
region 'domain'
   start 'outer' (-Lx,Ly) point value( phi)=0
   natural( phi)=-vx0     line to (-Lx,-Ly)
   natural( phi)=0     line to (Lx,-Ly)
   natural( phi)=vx0     line  to (Lx,Ly)
   natural( phi)=0     line to close
   start 'obstacle' (x1,y1)                                   { Cut-out }
   natural( phi)=0 line to (x2,y2) to (x3,y3) to (x4,y4) to close
PLOTS
   contour( vm) painted     vector( v) norm     contour( p) painted
   elevation( p) from (-Lx,-Ly) to (-Lx,Ly) report(brute_force)       { Left }
   elevation( p) from (Lx,-Ly) to (Lx,Ly)                             { Right }
   elevation( p) from (-Lx,-Ly) to (Lx,-Ly)                       { Bottom }
   elevation( p) from (-Lx,Ly) to (Lx,Ly)                         { Top }
END
```

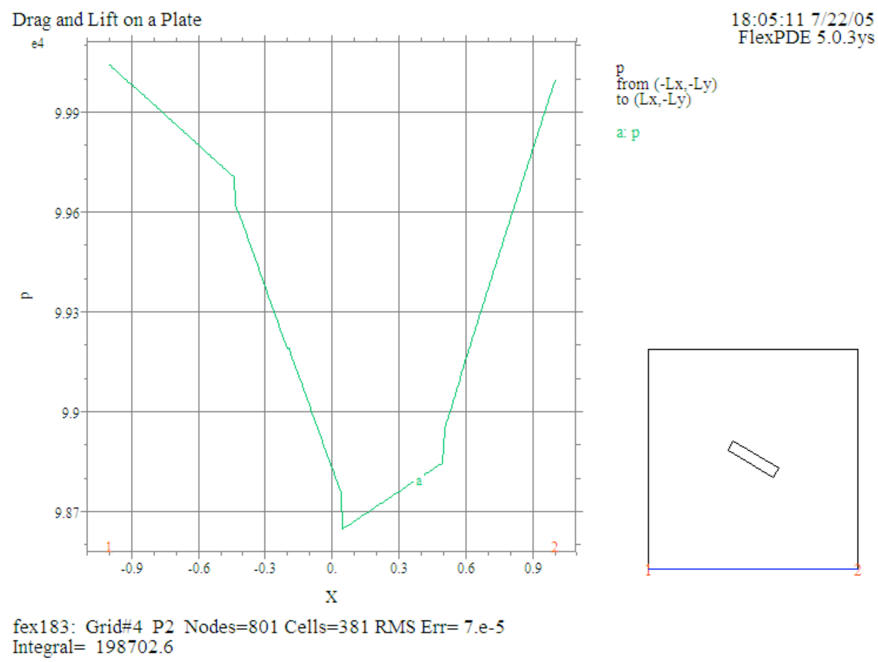
The following vector plot illustrates the geometry and also shows that there is a strong variation of speed at the corners.



Although the pressure distribution below seems to be symmetric with respect to the coordinate axes, this is less obvious than in the preceding example. It is clear, on the other hand, that the stream exerts a torque on the plate.



The elevation plots of p give us quantitative information about the pressure and the forces acting on the *liquid*. The following figure shows the pressure variation along the lower boundary line.



Taking the difference of the force integrals we find the value –7.7 for the *x*-component and –0.1 for the *y*-component. We compare these forces to an estimate (brute_force) of that acting on the surface facing the stream, i.e. p0*2*y2. We find that the force components are smaller than the reference value by a factor of at least 4000.

Hence, in the case of the sloping obstacle there seems to be no net force, neither drag nor lift. In fact, analytic theory shows that this is a general property of potential flow. This result is of course contrary to common experience, and the paradox stems from our unrealistic assumptions about the velocity at the solid interfaces. Molecules move randomly and cannot slide without friction along a boundary surface, but collide against it, thereby losing the velocity component along the surface. The boundary condition at a solid obstacle must obviously be zero tangential velocity, but this cannot be obtained with curl-free flow, as we shall see in later chapters.

In the next chapter we shall discover that the present kind of potential flow is not the most general class of irrotational motion.

Exercises

❑ Change the boundary condition at the *output* end of the constricted channel (*fex181*), such that you specify the appropriate horizontal velocity. Compare the results to those of the original example. Then change the output speed in the boundary condition by 10 % and observe the consequences.

❑ Change *fex181* so that the horizontal input velocity will vary across the channel according to the function $u_{x0} = 3[1-(y/L_y)^2]$, still keeping ϕ equal to zero at the output end.

❑ Use an input speed of 7.0 m/s in *fex181* and notice the minimum value of pressure resulting from the solution. Suggest a physical interpretation of the astonishing outcome.

❑ Change the angle of attack and the thickness of the inclined plate (*fex183*) according to your own taste.

❑ Expand *fex181* to fit the simplest model of a symmetrical Venturi tube[8p120].

19 Circulation around an Obstacle

In the preceding chapter on potential flow we obtained velocity fields with vanishing *curl*, known as irrotational. We shall now find that whenever there is an obstacle in the stream, alternative irrotational solutions exist. By adding such a solution to that of potential flow we obtain a more general kind of motion.

Let us start from the expression for the *curl* component relevant to motion in the (x, y) plane.

$$(\nabla \times \mathbf{v})_z = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} = \omega \qquad \bullet$$

The quantity ω is usually called *vorticity*. In irrotational flow, the vorticity has to vanish everywhere *in the liquid*, but the velocity field *inside* the obstacle does not have to obey this condition. Of course, nothing will be moving in the solid obstacle, but the solution may *formally* extend into this region.

The problem at hand is to solve the above PDE, which is of first order only. We thus proceed as on p.226 to transform it into a standard 2nd order PDE involving a new potential function, ψ. Since this type of flow as well involves a potential, we could call it *circulating* potential flow. With the definitions

$$v_x = \frac{\partial \psi}{\partial y}\ , \qquad v_y = -\frac{\partial \psi}{\partial x} \qquad \bullet$$

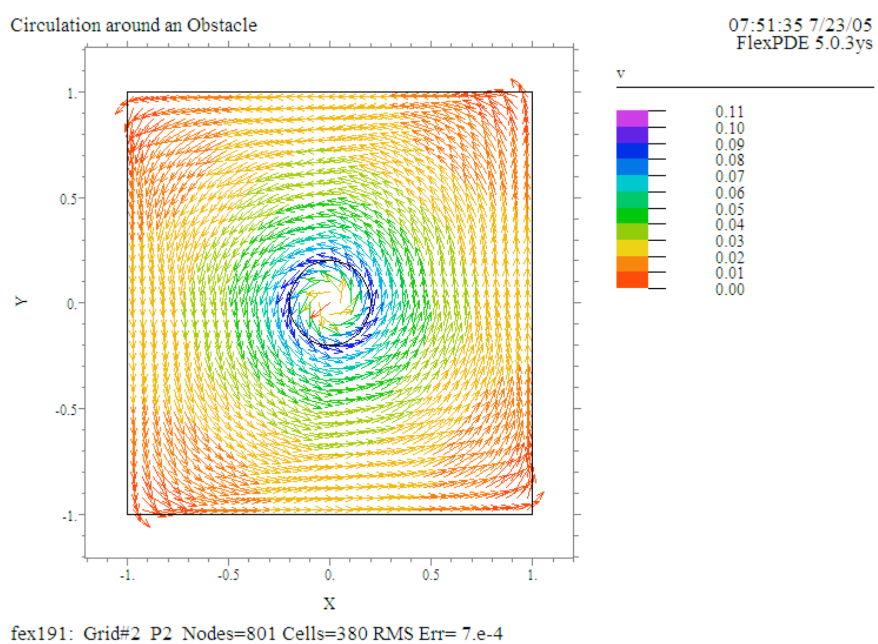
the above PDE takes the form of a Poisson equation, viz.

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \omega \equiv \nabla^2 \psi + \omega = 0 \qquad \bullet$$

where ω must be zero in the liquid while it may take different values in the region inside the obstacle. The descriptor below implements this idea in the simplest way, using the Student Version of FlexPDE.

```
TITLE     'Circulation around an Obstacle'                          { fex191.pde }
SELECT     errlim=1e-4     spectral_colors       { Student Version }
VARIABLES     psi                                { Circulation potential }
DEFINITIONS                                      { SI units }
  Lx=1.0     Ly=1.0     a=0.2
  omega                                          { Source of curl, vorticity }
  vx=dy( psi)     vy=-dx( psi)                   { Velocity components }
  v=vector( vx, vy)     vm=magnitude( v)
EQUATIONS
  div( grad( psi))+ omega=0
BOUNDARIES
region 'domain'     omega=0
  start 'outer' (-Lx,Ly)   value( psi)=0      { Vanishing normal velocity }
  line to (-Lx,-Ly) to (Lx,-Ly) to (Lx,Ly) to close
region 'obstacle'   omega=1.0   start 'circle' (0,-a)
  natural( psi)=0                             { Vanishing normal velocity }
  arc( center=0,0) angle=360
PLOTS
  contour( psi)     contour( vm) painted     vector( v) norm
  contour( div( v)) on 'domain'     contour( curl( v))
  elevation( normal( v), tangential( v)) on 'outer'
  elevation( normal( v), tangential( v)) on 'circle'
END
```

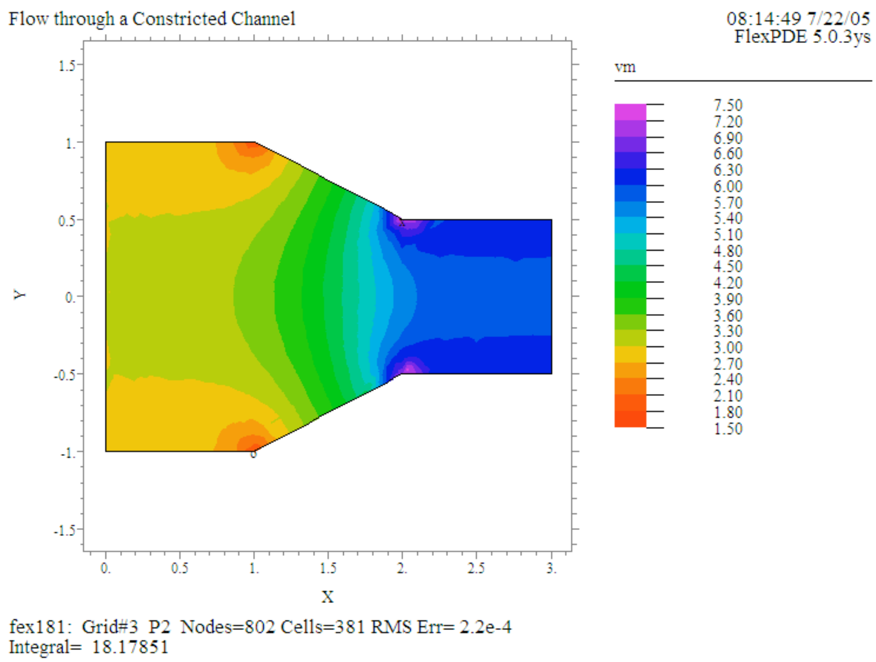
The vector plot below demonstrates (by color) that the flow is fastest close to the obstacle.



According to the above plot, the velocity on the outer boundary is parallel to the border. The first elevation plot, comparing the normal and tangential components of v confirms this fact.

The above plot also demonstrates that the velocity follows the borderline of the *obstacle*, which is essential. The last elevation plot of normal(v) shows this even more clearly.

The following contour plot of the speed vm indicates the maximum and minimum points clearly.



In the *definitions* segment we declared the vorticity omega to be a variable, but we waited until *boundaries* to assign values to it. The plot of div(v) shows the result to be zero in the liquid, and the next plot suggests that curl(v) also vanishes, as required.

The solution inside the obstacle is of course purely fictitious and is only used to introduce circulation in the liquid by means of the PDE. In the solid, curl(v) should be equal to $\omega = 1.0$ according to our definition. The plot suggests that curl(v) is about unity in that region, but the sparse node points give us only a very rough confirmation.

Circulation Integral

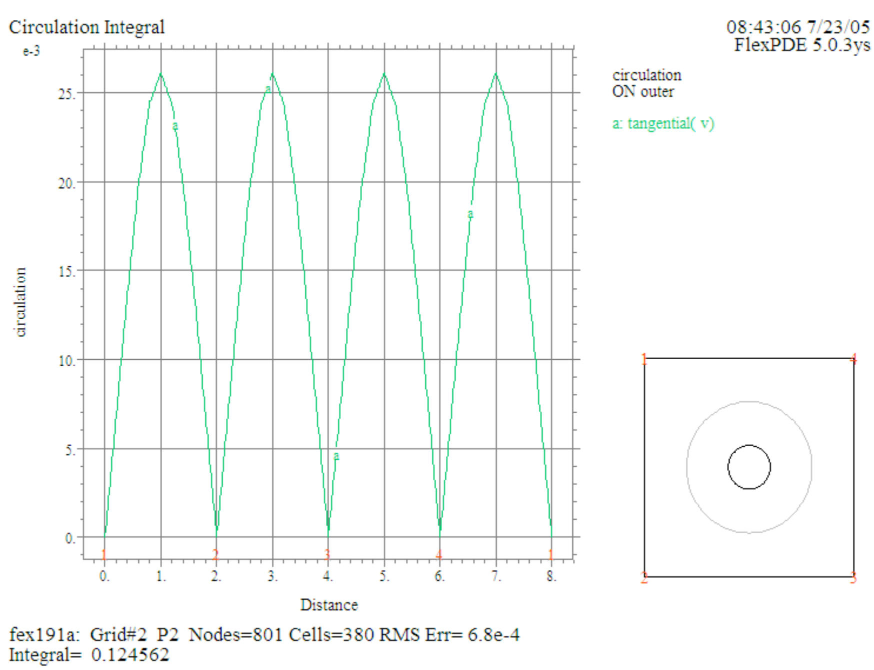
We shall now explore the circulation of the vector field quantitatively by line integrals along closed curves. The formal definition of *circulation* is

$$\Gamma = \oint \mathbf{v} \cdot d\,\mathbf{l} = \oint v_t\, dl \qquad \bullet$$

Since an elevation plot may present the tangential velocity v_t using the length as the independent variable, the integral value cited at the bottom of the plot is in fact equal to Γ. The following modifications to *fex191* are needed to calculate a few line integrals of this kind.

```
TITLE    'Circulation Integral '                          { fex191a.pde }
...
feature
  start 'circle3' (0,-3*a) arc( center=0,0) angle=360
PLOTS
  elevation( tangential( v)) on 'circle' as 'circulation'
  elevation( tangential( v)) on 'circle3' as 'circulation'
  elevation( tangential( v)) on 'outer' as 'circulation'
END
```

Under *boundaries* we have added a new closed curve with a radius three times that of the obstacle. The command feature lets us add lines inside the domain in the way we create regions.



We now calculate the circulation over three different curves, all enclosing the region where ω is non-zero. The above plot shows the tangential velocity-component on the square boundary. We find the other two integral values to be about the same.

Here, we may compare the circulation to an analytic expression by means of the Stokes integral theorem[1p364]

$$\oint_C \mathbf{v} \cdot d\mathbf{l} = \iint_A (\nabla \times \mathbf{v})_z \, dx \, dy = \iint_A \omega \, dx \, dy \qquad \bullet$$

where the first integral refers to a closed curve *C*, and the second and third ones to a region of area *A* enclosed by it. Since we have $\omega = 1.0$ inside the cross-section of the obstacle, the last integral evaluates to $\omega \pi a^2 = 0.12566$, in fair agreement with the line integrals.

Combined Velocity Fields

In order to obtain a more general solution for irrotational liquid flow we add a circulating field to the potential field from *fex182*. A convenient way of adding these fields is to calculate both by the same descriptor. We are perfectly free to solve for ϕ and ψ simultaneously, but the solution domains must be identical. Unfortunately, the potential field had a void for the obstacle, while the domain for the circulating velocity field was defined over a the entire square without an excluded region.

In order to solve for ϕ over the full domain, we may use a PDE that is slightly different from p.226●1, i.e.

$$\frac{\partial(c v_x)}{\partial x} + \frac{\partial(c v_y)}{\partial y} \equiv \nabla \cdot (c\,\mathbf{v}) = \nabla \cdot (c \nabla \phi) = 0 \qquad \bullet$$

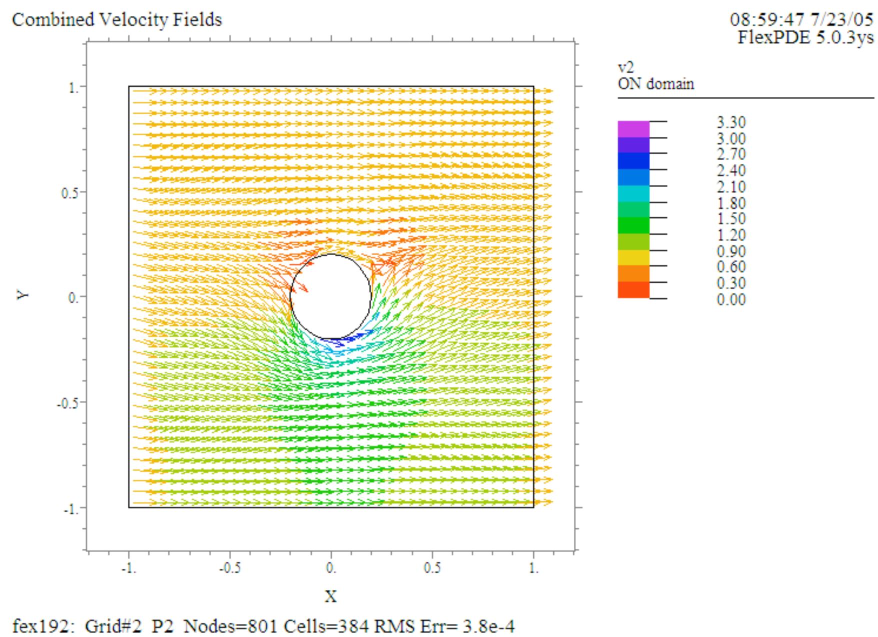
The idea is to define the constant c to be unity in the liquid and to take a suitably small value c_o in the region of the obstacle. The FEA program arranges to make the normal component $(c\,\mathbf{v})_n$ continuous across the interface from obstacle to liquid. This means that the relation $1 \cdot v_n = c_o v_{on}$ will make v_n on the liquid side much smaller than $|\mathbf{v}|$ in the region of the obstacle, which in turn is of the order of the input speed. In other words, the velocity will be closely tangential on the outside of the obstacle, as we assumed in the preceding chapter.

The following descriptor, combining the non-circulating and circulating fields, is based on *fex191*, some features from *fex191a* being added. The sum (v2) of the two velocities involves the coefficient c2. We use the latter to specify the amount of circulation.

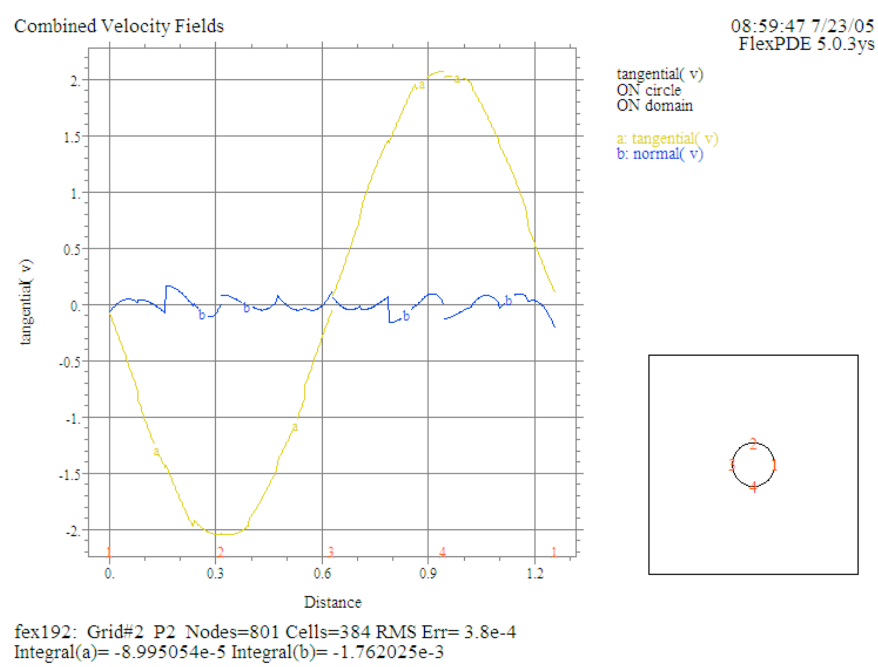
```
TITLE     'Combined Velocity Fields'                              { fex192.pde }
SELECT     errlim=1e-4     ngrid=1     spectral_colors
VARIABLES     phi     psi
DEFINITIONS
  Lx=1     Ly=1.0     a=0.2     vx0=1.0
  dens=1e3     p0=1e5          { Atmospheric pressure }
  omega     c                  { Angular velocity, parameter c for PDE }
  vx=dx( phi)     vy=dy( phi)                  { Velocity v from potential phi }
  v=vector( vx,vy)     vm=magnitude( v)
  vcx=dy( psi)     vcy=-dx( psi)       { Circulating velocity vc from psi }
  vc=vector( vcx, vcy)
  c2=10     v2x=vx+ c2*vcx     v2y=vy+ c2*vcy
  v2=vector( v2x, v2y)     v2m=magnitude( v2)
  p=p0+ 0.5*dens*( vx0^2- v2m^2)                 { Pressure }
  unit_x=vector( 1,0)     unit_y=vector( 0,1)
  force_x=-p*normal( unit_x)     force_y=-p*normal( unit_y)
EQUATIONS                              { Tagged with the dominant variable }
  phi:         div( c*grad( phi))=0                      { Potential flow }
  psi:         div( grad( psi))+ omega=0             { Circulating flow }
BOUNDARIES
region 'domain'   omega=0   c=1
  start 'outer'(-Lx,Ly)   natural( phi)=-c*vx0   value( psi)=0        { In }
  line to (-Lx,-Ly)   natural( phi)=0   line to (Lx,-Ly)
  natural( phi)=c*vx0                                             { Out }
  line  to (Lx,Ly)  natural( phi)=0  line to close
region 'obstacle'   omega=1   c=1e-10   start 'circle' (a,0)
  natural( phi)=0   natural( psi)=0   arc( center=0,0) angle=360
PLOTS
  vector( v) norm on 'domain'     vector( vc) norm on 'domain'
  vector( v2) norm on 'domain'
  contour( p) painted on 'domain'
  elevation( tangential( v), normal( v)) on 'circle' on 'domain'
  elevation( p) on 'circle' on 'domain'
  elevation( force_x, force_y) on 'circle'
  elevation( dens*vx0*tangential( v2)) on 'circle' on 'domain'
```

```
  contour( curl( v2))     contour( div( v2))
END
```

The following figure is a vector plot of the combined velocity v2. It shows that the speed is now higher below the obstacle, as we might expect.



The plot below tests to what extent the normal velocity vanishes on the circle.



The above plot shows that v_n is in fact smaller than v_t but fluctuates noticeably around zero. This is the best that can be done, however, with this limited number of nodes. Using the Professional Version with a smaller value of errlim we obtain less scatter and no visible net variation.

In order to calculate the force acting on the obstacle, we integrate $-p\cos(n,x)$ over the circle to obtain the *x*-component of the force, and so on. In practice, we construct a unit vector field unit_x, which combines with normal to give us the direction cosine.

In this example, the pertinent velocity field exists in the *liquid* region, which we have to keep in mind when plotting and calculating line integrals. Under *boundaries* we first define a total domain and then reserve a circular region for the obstacle. As a consequence of this definition, 'domain' becomes equivalent to the remainder, i.e. the liquid region.

For the line integrals, FlexPDE permits us to specify both the *curve* for integration ('circle') and the *region* where the data are to be fetched. We specify this by the modifier on 'circle' on 'domain'.

The elevation plot of the local forces shows that the *integral* of force_x is now small compared to force_y, which takes a negative value (-1302). The force is thus perpendicular to the main stream and directed downwards, as is also evident from the contour plot of the pressure.

Kutta and Joukovski[8p156] used a complex formalism to derive an expression for the force on a cylindrical object of general shape. The result for the lift force is

$$F_y = -\rho v_{x0}\Gamma \qquad \bullet$$

Judging from the last elevation plot, which yields the circulation (Γ), our integrated value agrees reasonably well with the analytic result for the negative lift force.

We have seen that the circulating mode of motion may produce a force on the obstacle, transverse to the input velocity vx0. This is similar to the Magnus effect[8p159], which is easily observed in a tennis court. There is no drag force, however, on a cylinder in a non-viscous liquid.

Finally, the contour plots of div(v2) and curl(v2) confirm that the combined velocity field conserves mass and is *irrotational*.

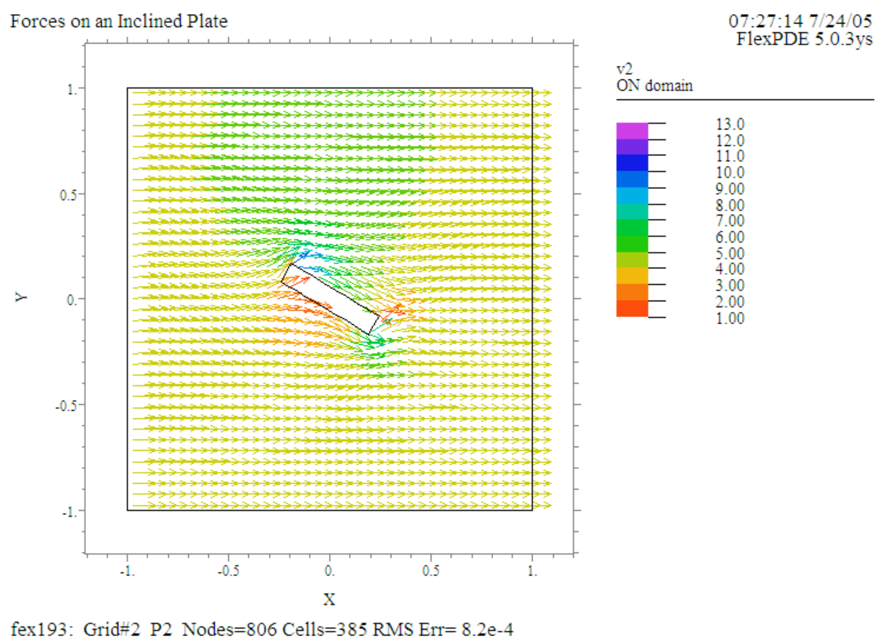
Forces on an Inclined Plate

Let us now apply the above PDEs to *fex183* in the preceding chapter, exploiting suitable fractions of *fex192*. Here, we exploit the feature that a boundary condition need not be repeated if unchanged.

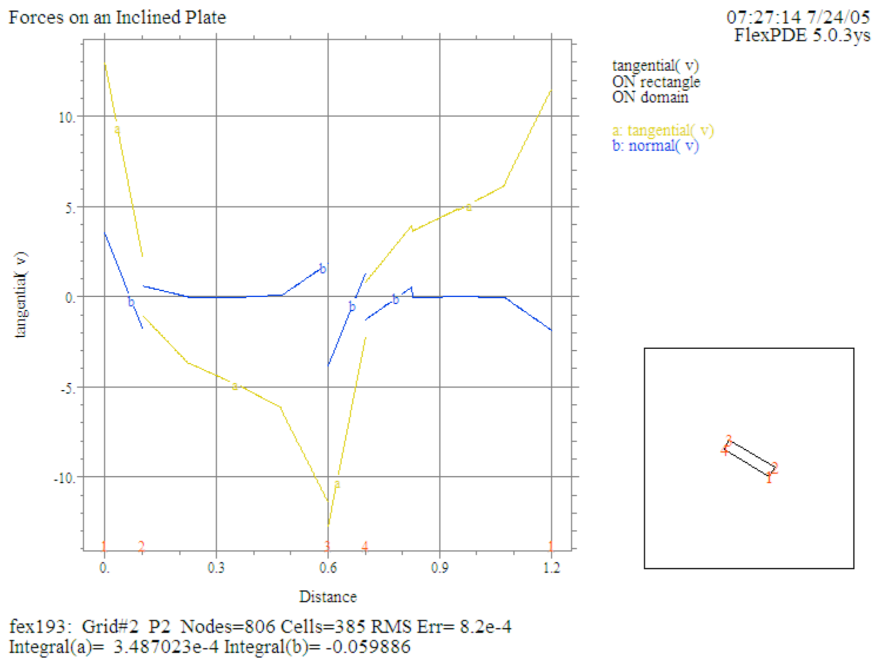
```
TITLE     'Forces on an Inclined Plate'                          { fex193.pde }
SELECT    errlim=1e-4    ngrid=1    spectral_colors
VARIABLES    phi    psi
DEFINITIONS
  Lx=1.0    Ly=1.0    a=0.5*Ly    d=0.2*a
{ Geometric parameters for inclined plate }
    alpha=30* pi/180                      { Angle of attack, radians }
    si=sin( alpha)    co=cos( alpha)
    x1=-d/2*si- a/2*co    y1=-d/2*co+ a/2*si          { Corners }
    x2=d/2*si- a/2*co    y2=d/2*co+ a/2*si
    x3=-x1    y3=-y1    x4=-x2    y4=-y2
  dens=1e3    p0=1e5               { Atmospheric pressure at left end }
  vx0=5    vx=dx(phi)    vy=dy(phi)      { Velocity from potential phi }
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector( 1,0)    unit_y=vector( 0,1)
  omega    c            { Angular velocity and parameter for PDE }
  vcx=dy( psi)    vcy=-dx( psi)      { Circulating field from psi }
  vc=vector( vcx, vcy)    vcm=magnitude( vc)
{ Combining velocities v and vc to obtain v2 }
  c2=-30    v2x=vx+ c2*vcx    v2y=vy+ c2*vcy
  v2=vector( v2x, v2y)    v2m=magnitude( v2)
  p2=p0+ 0.5*dens*( vx0^2- v2m^2)                  { Pressure }
  force_x=-p2*normal( unit_x)    force_y=-p2*normal( unit_y)
EQUATIONS                                          { Tagged }
  phi:       div( c*grad( phi))=0                  { Potential flow }
  psi:       div( grad( psi))+ omega=0             { Circulating flow }
BOUNDARIES
region 'domain'   omega=0   c=1
  start 'outer' (-Lx,Ly)
  natural( phi)=-c*vx0   value( psi)=0   line to (-Lx,-Ly)          { In }
  natural( phi)=0    line to (Lx,-Ly)
  natural( phi)=c*vx0    line  to (Lx,Ly)                          { Out }
```

```
  natural( phi)=0     line to close
region 'obstacle'   omega=1   c=1e-10
  start 'rectangle' (x4,y4)   natural( phi)=0   natural( psi)=0
  line to (x3,y3) to (x2,y2) to (x1,y1) to close
PLOTS
  vector( v) norm on 'domain'     vector( vc) norm on 'domain'
  vector( v2) norm on 'domain'     contour( p2) painted on 'domain'
  elevation( tangential( v), normal( v)) on 'rectangle' on 'domain'
  elevation( force_x, force_y) on 'rectangle' on 'domain'
  elevation( dens*vx0* tangential( v2)) on 'rectangle' on 'domain'
END
```

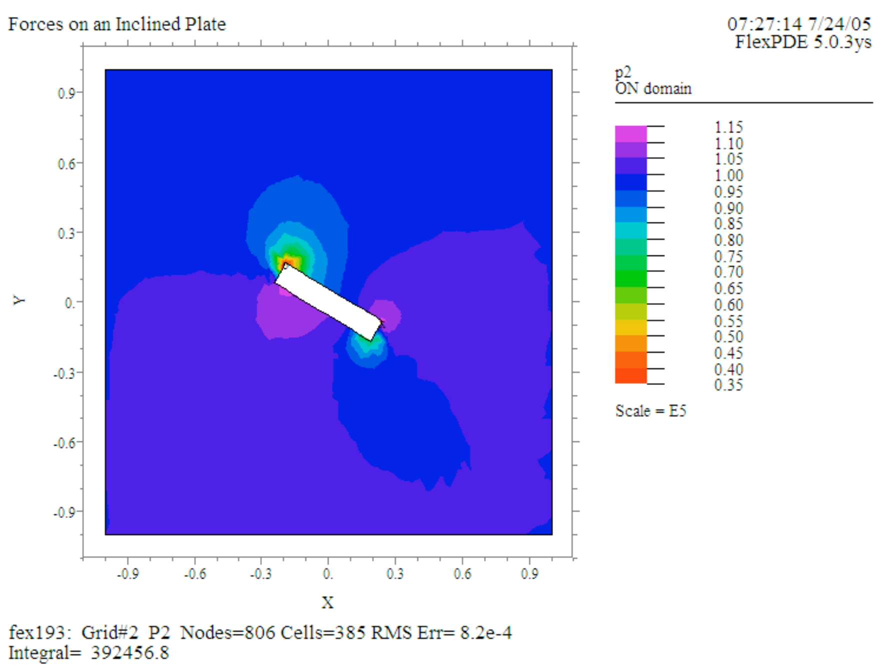
The following vector plot indicates that the liquid flows along the sides of the plate as required. Since we now have chosen a negative value of c2, the combined velocity is higher on the top face of the plate, which we expect to result in a lift force.



The following plot of v_t and v_n illustrates that the normal component is relatively small on the long sides, but that the few node points on the short sides do not yield the ideal zero, but positive and negative slopes.



The plot of the pressure below makes it clear that there is an upward force on the left part of the plate and a downward force on the right part, which should produce a torque.



Integrating by means of the second elevation plot we find that the net vertical force is in fact positive. Its magnitude (6584) is in rough agreement with the Kutta-Joukovski value (7373), given by the integral on the last plot. There is also a right-directed drag force in the direction of flow.

In summary of this chapter, we note that an added circulating field reproduces to some extent the lift force found by experience. The required amount of circulation (Γ) is not directly given by the Kutta-Joukovski theory, however, which means that the coefficient c2 has to be determined by trial and error to provide smooth flow-off.

Most importantly, combining potential flow and circulating flow does *not* yield zero speed on the surface of the obstacle. This means that the detailed velocity field is *unphysical*, even if it predicts reasonable forces.

Exercises

❑ Investigate if *fex191* may be modified to accommodate an obstacle of square cross-section.
❑ Explore how the lift and drag forces obtained by *fex193* vary with the angle of attack. What happens at negative alpha?
❑ Adapt *fex193* to treat flow around an obstacle of square cross-section.

20 Viscous Flow in Channels

In this chapter we shall deal with realistic situations in (x, y), where a liquid locally is at rest with respect to the solid objects in contact with it. Under such conditions curl($\mathbf{v}$) will in general be non-zero.

Classical mechanics applied to a liquid yields the Navier-Stokes equation[8p59]. That equation expresses Newton's law of motion

$$\rho_0 \frac{d\mathbf{v}}{dt} = \mathbf{f}_{tot}$$

for the total force $\mathbf{f}_{tot}$ on a fluid element that is carried along with the stream. (That kind of derivative is also commonly denoted $\mathrm{D}\mathbf{v}/\mathrm{D}t$.) Here, ρ_0 is the *constant* mass density of the fluid. Since the velocity in a chosen volume element is a function of (t, x, y), we may write

$$\frac{d\mathbf{v}}{dt} = \frac{\partial \mathbf{v}}{\partial t} + \frac{\partial \mathbf{v}}{\partial x}\frac{dx}{dt} + \frac{\partial \mathbf{v}}{\partial y}\frac{dy}{dt} = \frac{\partial \mathbf{v}}{\partial t} + v_x \frac{\partial \mathbf{v}}{\partial x} + v_y \frac{\partial \mathbf{v}}{\partial y} = \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v}$$

With this expression for the derivative, Newton's law takes the form

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} + \rho_0 (\mathbf{v} \cdot \nabla)\mathbf{v} - \mathbf{F} + \nabla p - \eta \nabla^2 \mathbf{v} = 0 \qquad \bullet$$

where $\mathbf{F}$ is an *external* force (e.g. gravity), $-\nabla p$ the force due to pressure, and $\eta \nabla^2 \mathbf{v}$ the one proportional to viscosity[8pp57,69]. This vector PDE is known as the *Navier-Stokes* (N-S) equation.

The second term, $\rho_0 (\mathbf{v} \cdot \nabla)\mathbf{v}$, has the dimension of force but it is really part of the time derivative and hence called an *inertial* force. This term is obviously second-order in $\mathbf{v}$.

The last term corresponds to the *viscous* force on the volume element. Normally, ∇^2 operates on a scalar and $\nabla^2 \mathbf{v}$ should be taken as shorthand for the vector $(\mathbf{i}\nabla^2 v_x + \mathbf{j}\nabla^2 v_y)$.

The simplest case of flow occurs at such small speeds that the non-linear inertial force become negligible compared to viscous force, and in the present chapter we shall consider liquid motion under such conditions. The ratio of inertial-to-viscous forces is usually expressed in the form of the dimensionless *Reynolds number*, defined by

$$\mathrm{Re} = \frac{\rho_0 v_0 L_0}{\eta} \qquad \bullet$$

where v_0 is a typical speed and L_0 a typical size of the solution domain. This number gives us an order-of-magnitude indication of the sort of flow we are dealing with. At sufficiently small values of Re, the inertial term is negligible compared to the viscous force and the problem can be treated as linear in the dependent variables. The PDEs then yield solutions corresponding to laminar flow.

Above the first critical value ($\mathrm{Re} = 1$) the solutions may remain laminar, even if the PDEs are non-linear. Above a much higher value (Re=100 or much more depending of the details of the problem) the solution becomes turbulent and time-dependent (permanently unstable).

In Cartesian coordinates, the component Navier-Stokes equations may thus be written (for the *x*- and *y*-directions respectively)

$$\rho_0 \begin{Bmatrix} \dfrac{\partial v_x}{\partial t} \\ \dfrac{\partial v_y}{\partial t} \end{Bmatrix} + \rho_0 (\mathbf{v} \cdot \nabla) \mathbf{v} - \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} + \begin{Bmatrix} \dfrac{\partial p}{\partial x} \\ \dfrac{\partial p}{\partial y} \end{Bmatrix} - \eta \begin{Bmatrix} \dfrac{\partial^2 v_x}{\partial x^2} + \dfrac{\partial^2 v_x}{\partial y^2} \\ \dfrac{\partial^2 v_y}{\partial x^2} + \dfrac{\partial^2 v_y}{\partial y^2} \end{Bmatrix} = 0 \qquad \bullet$$

Here, we have kept the second term unexpanded, since it may be disregarded until a later chapter.

So far, we have only two equations for the three dependent variables v_x, v_y, and *p*. Conservation of mass at constant density gives us a third equation[8p52], i.e.

$$\nabla \cdot (\rho_0 \mathbf{v}) = \rho_0 \nabla \cdot \mathbf{v} = \rho_0 \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \right) = 0 \qquad \bullet$$

but unfortunately this is a PDE of first order only, which FlexPDE would not accept.

Using $\nabla \cdot \mathbf{v} = 0$ together with the equation of motion we may, however, generate a relation containing second-order derivatives in p. Applying the divergence operator to the N-S equation we obtain[11]

$$\rho_0 \frac{\partial}{\partial t} \nabla \cdot \mathbf{v} + \rho_0 \nabla \cdot [(\mathbf{v} \cdot \nabla)\mathbf{v}] - \nabla \cdot \mathbf{F} + \nabla^2 p - \eta \nabla \cdot (\nabla^2 \mathbf{v}) = 0$$

where the first term vanishes because of mass conservation. Furthermore, we may eliminate the last term using the identities

$$\eta \nabla \cdot (\nabla^2 \mathbf{v}) = \eta \nabla^2 (\nabla \cdot \mathbf{v}) = \eta \nabla^2 (0) = 0$$

The remainder of the modified N-S equation is

$$\nabla^2 p + \rho_0 \nabla \cdot [(\mathbf{v} \cdot \nabla)\mathbf{v}] - \nabla \cdot \mathbf{F} = 0 \qquad \bullet$$

If the volume force **F** is constant in space the last term will vanish.

Expressed in Cartesian coordinates, this PDE takes the form

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \rho_0 \nabla \cdot [(\mathbf{v} \cdot \nabla)\mathbf{v}] - \nabla \cdot \mathbf{F} = 0$$

Even in this equation we leave the term containing ρ_0 unexpanded, since it will not be used in the present chapter.

We now have a total of three PDEs for calculating v_x, v_y and p. Although we derived the equation for p using mass conservation, it would be wrong to assume that any solution to these three PDEs would necessarily satisfy $\nabla \cdot \mathbf{v}=0$. In fact, one may show[11] that this is true only in special cases. We shall see that the first two examples in this chapter are sufficiently simple for the divergence to vanish automatically.

It could never be wrong, however, to add $\nabla \cdot \mathbf{v}$, multiplied by a factor, to the equation for p, since the divergence should vanish in the final stage of the solution process. Hence we settle for the following form

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \rho_0 \nabla \cdot [(\mathbf{v} \cdot \nabla)\mathbf{v}] - \nabla \cdot \mathbf{F} - f_\nabla \nabla \cdot \mathbf{v} = 0 \qquad \bullet$$

where we may choose the factor f_∇ freely according to the problem at hand, to ensure vanishing divergence. Trial runs lead us to employ a negative factor. We may always verify by means of plots that the divergence vanishes for a given solution.

The factor f_∇ may not be taken as a fixed number, however, since it has a physical dimension, in fact the same as η / L_0^2. Hence, we should write

$$f_\nabla = C \frac{\eta}{L_0^2} \qquad \bullet$$

where the parameter L_0 is a typical size of the domain. The number C is to be chosen empirically, large enough to ensure vanishing $\nabla \cdot \mathbf{v}$, but not so large that it impairs convergence in FlexPDE calculations or requires unreasonably long runtimes.

Although the divergence term was introduced on intuitive grounds and proves itself in practical use, we may understand approximately how it works. In the derivation of p.252●1 we used the term $\mathbf{f} = -\nabla p$ for the force generated by pressure. The Gauss theorem[6p43] now yields

$$\iiint \nabla^2 p \, dV = \iiint \nabla \cdot \nabla p \, dV = -\iiint \nabla \cdot \mathbf{f} \, dV = -\oiint f_n \, ds$$

Let us now consider a small region around a point of interest. By subtracting a certain amount from the $\nabla^2 p$ term in p254●2 we effectively create an outward force on the boundary of that region, which transports fluid away from the point considered. This nudges the calculations toward vanishing divergence.

Boundary Conditions

Now that we have a PDE for pressure, we must find out what boundary conditions to use with it. This is easy enough where the pressure takes known values, but what about boundaries that just limit the fluid flow?

The alternative to *value* is a *natural* statement. In the latter case we need an expression for $\partial p / \partial n \equiv \mathbf{n} \cdot \nabla p$, where $\mathbf{n}$ is the outward normal ($|\mathbf{n}| = 1$) at the boundary of the domain. The N-S equation (p.252) provides the answer rather directly[11]:

$$\nabla p = \mathbf{F} + \eta \nabla^2 \mathbf{v} - \rho_0 \frac{\partial \mathbf{v}}{\partial t} - \rho_0 (\mathbf{v} \cdot \nabla)\, \mathbf{v}$$

If the pressure is not known on a boundary segment, we may thus use the following general expression for the *natural* boundary condition

$$\partial p / \partial n \; = \mathbf{n} \cdot \nabla p = \mathbf{n} \cdot \mathbf{F} + \eta \mathbf{n} \cdot \nabla^2 \mathbf{v} - \rho_0 \mathbf{n} \cdot \frac{\partial \mathbf{v}}{\partial t} - \rho_0 \mathbf{n} \cdot \left[(\mathbf{v} \cdot \nabla)\, \mathbf{v} \right] =$$
$$n_x F_x + n_y F_y + \eta \left[n_x \nabla^2 v_x + n_y \nabla^2 v_y \right] - \rho_0 \mathbf{n} \cdot \frac{\partial \mathbf{v}}{\partial t} - \rho_0 \mathbf{n} \cdot \left[(\mathbf{v} \cdot \nabla)\, \mathbf{v} \right] \qquad \bullet$$

where $\rho_0 \, \partial v_n / \partial t$ will vanish in the steady state, and we defer the expansion of the last term until it is required later.

Steady Flow at Small Speeds (Re<<1)

In this chapter and the next one we shall only be concerned with steady flow, which means that we omit the time derivative. We also assume Re to be small enough to permit us to neglect the PDE term proportional to the density. The three PDEs then take the simpler form

$$\begin{Bmatrix} \dfrac{\partial p}{\partial x} \\ \dfrac{\partial p}{\partial y} \end{Bmatrix} - \eta \begin{Bmatrix} \dfrac{\partial^2 v_x}{\partial x^2} + \dfrac{\partial^2 v_x}{\partial y^2} \\ \dfrac{\partial^2 v_y}{\partial x^2} + \dfrac{\partial^2 v_y}{\partial y^2} \end{Bmatrix} = 0 \qquad \bullet$$

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} - C \frac{\eta}{L_0^2} \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \right) = 0 \qquad \bullet$$

We shall soon see that in the most elementary examples, involving parallel flow, we may even neglect the last (divergence) term.

For small Re, the *natural* boundary condition for pressure simplifies into

$$\partial p / \partial n \; = n_x F_x + n_y F_y + \eta \left(n_x \nabla^2 v_x + n_y \nabla^2 v_y \right) \qquad \bullet$$

Flow Due to a Moving Wall at Re<<1

We shall now consider the motion of a liquid confined between two parallel walls. One wall is kept stationary and the other one moves with speed v_{x0}, at constant spacing between the walls. In order to obtain a small Reynolds number with the usual domain size and reasonable velocity, we have chosen a hypothetical liquid of very high viscosity.

In the two preceding chapters we imposed the *ambient pressure* p0, because there was a risk of large negative pressures at corners, leading to voids. In the N-S PDE, only derivatives of p occur, and hence we may ignore p0 in the solution process. We may always add the ambient pressure later to the solution for p to ensure that the total pressure remains positive.

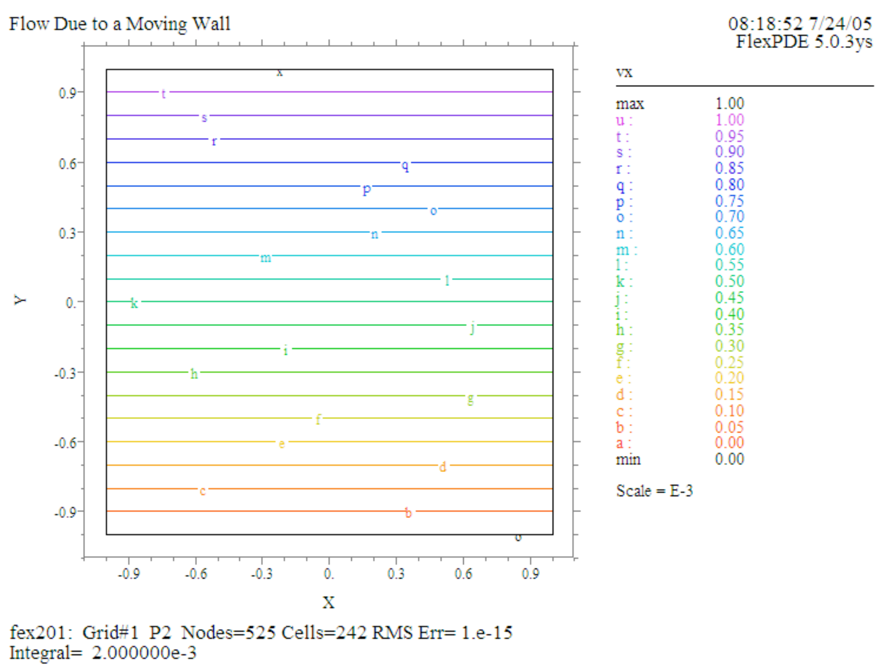
Under *boundaries* we specify the velocity components on the solid surfaces. We assume that the moving wall, rather than a pressure difference, drives the motion and hence the pressure is taken to be zero on both of the vertical sides. In the above expression for $\partial p/\partial n$ we have $n_y = 1$ on the upper horizontal side, since the outward normal to the boundary points in the direction of positive y. On the lower boundary we must enter $n_y = -1$.

```
TITLE   'Flow Due to a Moving Wall'                              { fex201.pde }
SELECT    errlim=1e-5    spectral_colors                    { Student Version }
VARIABLES    vx    vy    p
DEFINITIONS
  Lx=1.0    Ly=1.0    vx0=1e-3    visc=1e4              { Viscosity }
  dens=1e3    Re=dens*vx0*2*Ly/visc                        { Reynolds number }
  v=vector( vx, vy)    vm=magnitude( v)                   { Speed }
EQUATIONS    { Tagged by dominant variable }    { For vanishing Re }
  vx:        dx( p)- visc*div( grad( vx))=0
  vy:        dy( p)- visc*div( grad( vy))=0
  p:         div( grad( p))=0         { Divergence term neglected }
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0   value( vy)=0   value(p)=0   line to (-Lx,-Ly)    { Left }
  value( vx)=0   value( vy)=0   natural(p)=-visc*div( grad( vy))
  line to (Lx,-Ly)   natural( vx)=0   value( vy)=0   value(p)=0      { Right }
  line to (Lx,Ly)   value( vx)=vx0   value( vy)=0                   { Upper }
```

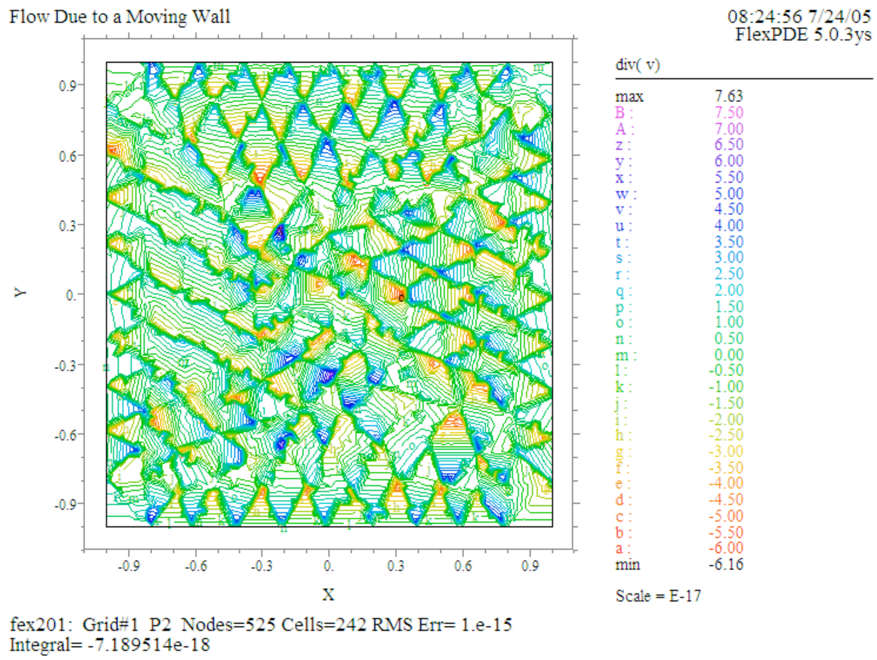
```
  natural(p)=visc*div( grad( vy))  line to close
PLOTS
  elevation( vx, vy) on 'outer' report( Re)
  contour( vx)     contour( vy)     contour( p)
  vector( v) norm
  contour( div( v))     contour( curl( v)) painted
END
```

The elevation plot taken on the outer boundary is useful for checking that the velocity boundary conditions have been fulfilled. The liquid evidently does not slip over the solid boundaries.

The distribution of vx shown by the plot below is extremely simple. The velocity vector turns out to be highly parallel to the *x*-axis (laminar flow), which is confirmed by the plot of vy.



The plot below suggests that $\nabla \cdot \mathbf{v}$ vanish everywhere. Applying the integral definition of the divergence (pp.21,23) to a small box parallel to the axes, we find that only the sides perpendicular to the *x*-axis contribute, and with opposite signs. The plot of div(v) does not yield exactly zero, but the contours are irregular and the values are small compared to an estimate of the maximum possible derivative, $\delta v_x / \delta y \cong 10^{-3}$.



Using the line integral definition of *curl* (p.21), we first notice that its value must be the same everywhere. The local *curl* must thus equal the average value we obtain from a line integral along the boundary, which amounts to $(0 - v_{x0} 2L_x)/(2L_x \cdot 2L_y) = -v_{x0}/2 = -5e-4$. This result is borne out by the plot of curl(v).

Pressure-Driven Flow through a Channel

As a second elementary example we study steady flow between two parallel walls, driven by a prescribed pressure difference δp. Since the main velocity component will not be known beforehand, we calculate the Reynolds number using globalmax, which yields the largest value over the solution domain.

We shall use *fex201* as a template for the following descriptor. The natural boundary conditions are equally simple in this case, since only n_y is non-zero on the solid boundaries. On the left and right boundaries we specify natural(vx)=0, which means $\partial v_x / \partial x = 0$ on the end faces.

This problem has a simple analytic solution[8p8], i.e.

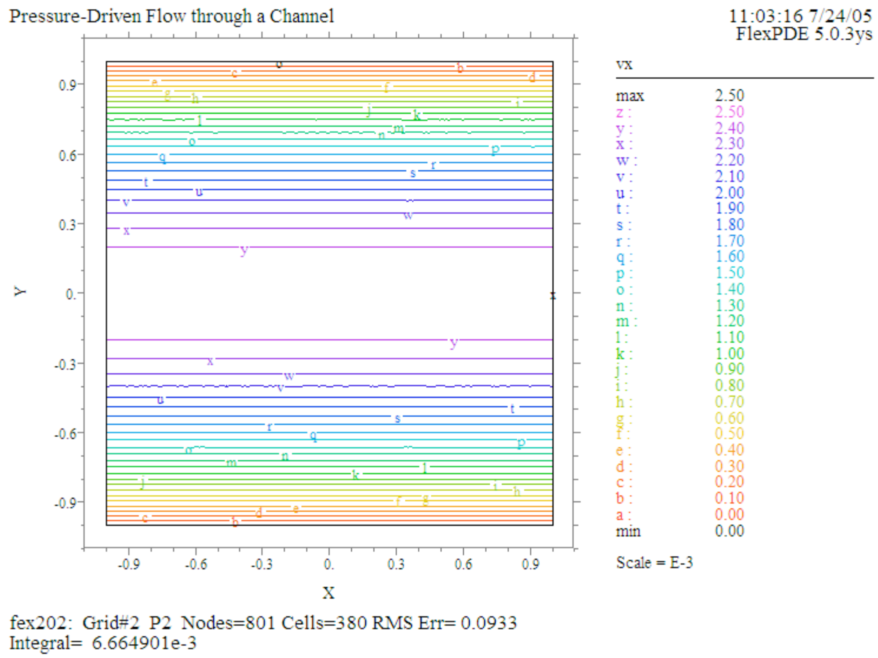
$$v_x = \frac{\delta p / \ell}{2\eta}\left(w^2 - y^2\right), \quad v_y = 0 \qquad \bullet$$

where δp is the pressure difference between the ends, ℓ the length of the channel, and $2w$ its width. Since v_x is independent of x, the pressure gradient in that direction must be constant for symmetry reasons. The pressure $p(x)$ must thus be a linear function. This set of functions may easily be shown to satisfy the PDEs and the boundary conditions. We enter the expression for the horizontal velocity under the notation vx_ex. The modifications to *fex201* are as follows.

```
TITLE    'Pressure-Driven Flow through a Channel'         { fex202.pde }
…
  Lx=1.0    Ly=1.0    visc=1e4    delp=100          { Driving pressure }
  vx_ex=delp/(2*Lx)/(2*visc)*(Ly^2- y^2)            { Exact solution }
  dens=1e3    Re=dens*globalmax( vx)*2*Ly/visc
…
region 'domain'
  start 'outer' (-Lx,Ly)
  natural( vx)=0   value( vy)=0    value(p)=delp                   { In }
  line to (-Lx,-Ly)  value( vx)=0   value( vy)=0
  natural(p)=-visc*div( grad( vy))
  line to (Lx,-Ly) natural( vx)=0   value( vy)=0   value(p)=0      { Out }
  line to (Lx,Ly)   value( vx)=0   value( vy)=0
  natural(p)=visc*div(grad( vy))
  line to close
…
  contour( vx- vx_ex) report( globalmax( vx))
END
```

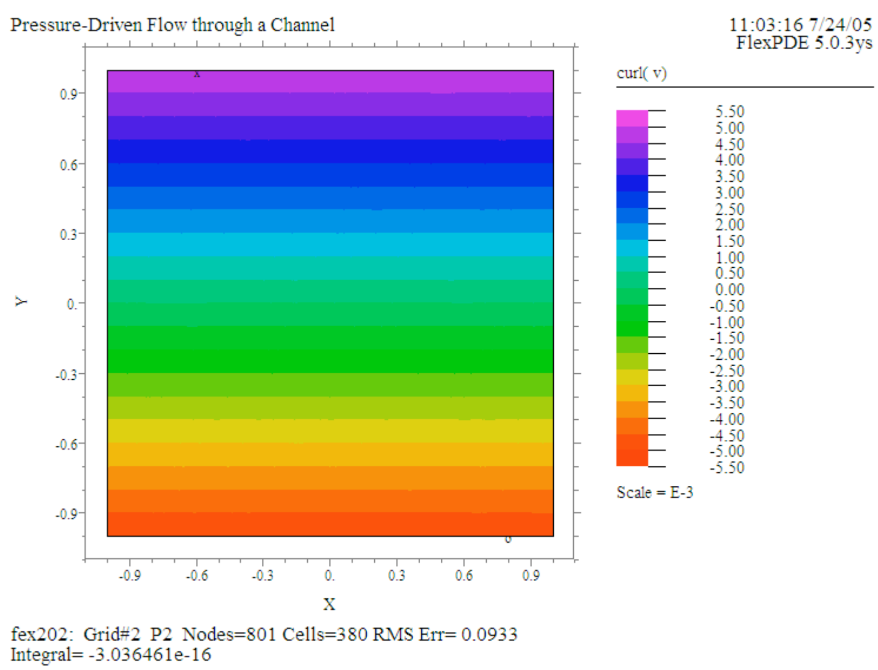
The plot below shows the solution for the horizontal component of velocity, vx. The value is zero at the horizontal boundaries and takes a maximum at mid-distance.

Comparing the contour plots of vx and vy we find that the velocity is accurately horizontal everywhere. This is another example of laminar flow, and the simplicity of the motion makes it obvious that div(v) must be zero.



The plot of the speed error (not shown here) indicates that vx is true to about one part in 10^{12}.

The plot below illustrates that curl(v) is non-zero everywhere, except in the symmetry plane, where this function changes sign.



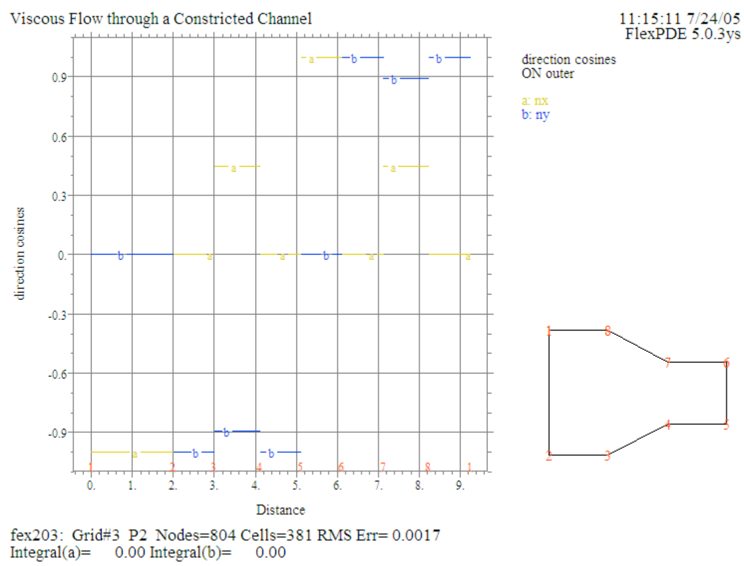
We may also calculate the vorticity from the analytic solution as $-\partial v_x / \partial y$. This is another example of innocent-looking, laminar flow that proves to be rotational.

Viscous Flow through a Constricted Channel

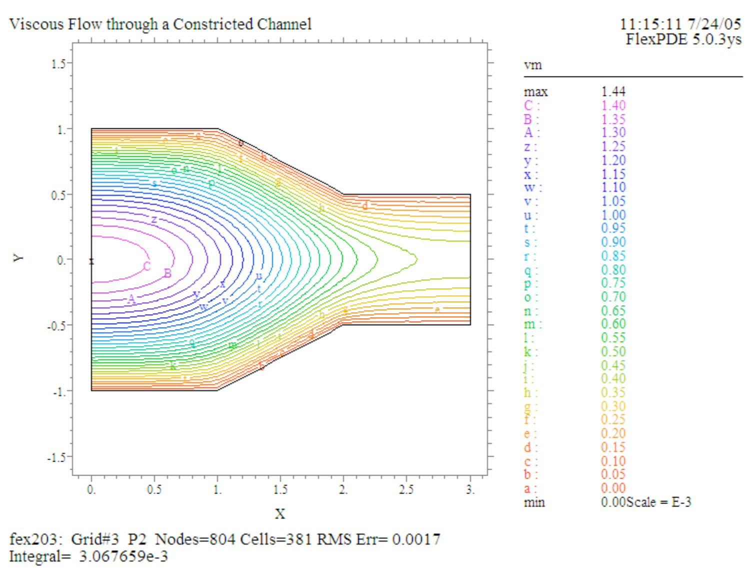
The following is a modification of *fex181*, which should make it valid for viscous flow. Here, we have used the *unit vector field* which is expedient for expressing the direction cosines (n_x, n_y) occurring in the natural boundary conditions for p. On the input and output faces we have specified $\partial v_x/\partial x = 0$, assuming that there is negligible change in v_x close to the ends.

```
TITLE     'Viscous Flow through a Constricted Channel'      { fex203.pde }
SELECT    errlim=1e-4    ngrid=1    spectral_colors
VARIABLES    vx    vy    p
DEFINITIONS
  Lx=1.0    Ly=1.0    coef=0.5    visc=1e4
  delp=100                                           { Driving pressure }
  dens=1e3    Re=dens*globalmax( vx)*2*Ly/visc
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector(1,0)    unit_y=vector(0,1)        { Unit vector fields }
  nx=normal( unit_x)    ny=normal( unit_y)      {Direction cosines }
{ Natural boundary condition for p: }
  natp=visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
EQUATIONS
  vx:         dx( p)- visc*div( grad( vx))=0
  vy:         dy( p)- visc*div( grad( vy))=0
  p:          div( grad( p))=0
BOUNDARIES
region 'domain' start 'outer' (0,Ly)
  natural( vx)=0   value( vy)=0   value( p)=delp               { In }
  line to (0,-Ly)  value( vx)=0   value( vy)=0   natural( p)=natp
  line to (Lx,-Ly) to (2*Lx,-Ly*coef) to (3*Lx,-Ly*coef)
  natural( vx)=0   value( vy)=0   value( p)=0                  { Out }
  line  to (3*Lx,Ly*coef)  value( vx)=0   value( vy)=0   natural( p)=natp
  line to (2*Lx,Ly*coef) to (Lx,Ly) to close
PLOTS
  elevation( nx, ny) on 'outer' as 'direction cosines'
  contour( vx) report(Re)    contour( vm)
  vector( v) norm    contour( p)
  contour( div( v)) painted    contour( curl( v)) painted
  elevation( vx) from (0.5*Lx,-Ly) to (0.5*Lx,Ly)
  elevation( vx) from (2.5*Lx,-Ly*coef) to (2.5*Lx,Ly*coef)
END
```

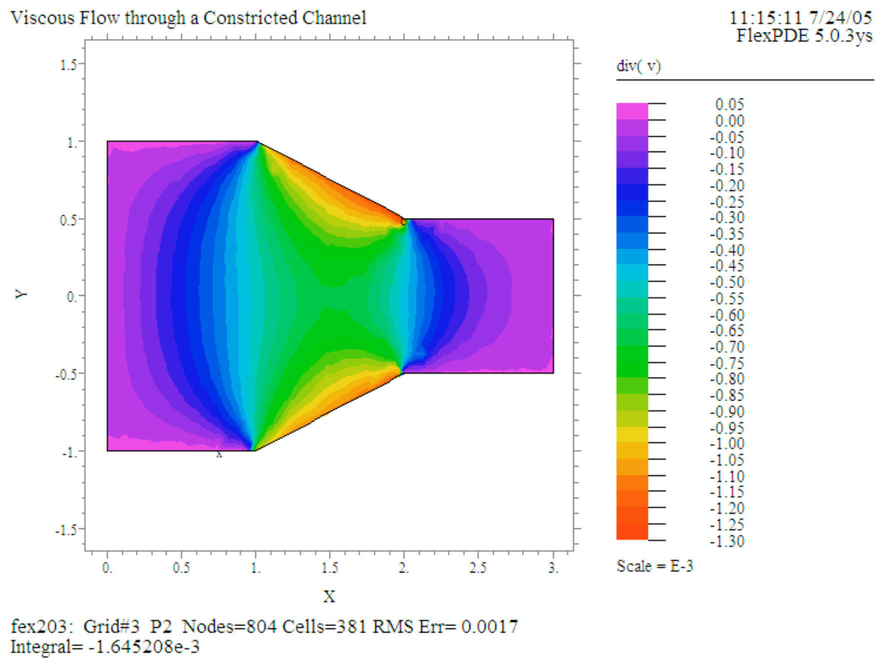
The following plot of nx and ny shows how the direction cosines change as we go along the contour.



The solution converges rapidly, and the plot of vm (below) demonstrates that the speed indeed vanishes on the walls. It is surprising to find, however, that the output speed is smaller than the input value.



The next plot shows that the divergence definitely is non-zero. We also obtain a similar indication from the two elevation plots. The integral value reported on the bottom line is obviously equal to the volume of liquid transported through the cross-section (per second and meter of depth in *z*). The two integral values show that the fluxes through the cross-sections are different. In short, the solution does not conserve mass and is definitely *wrong*!

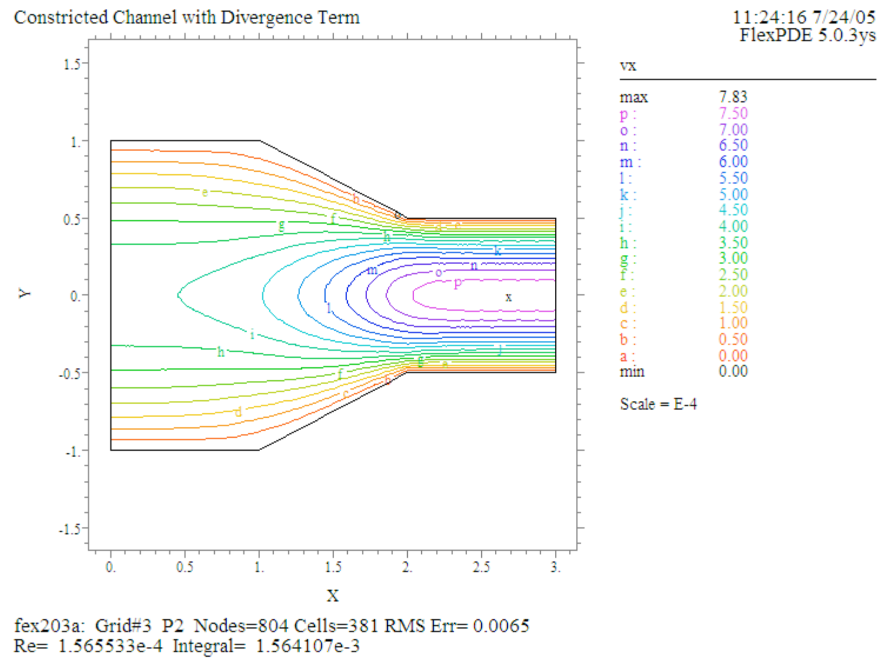


The cause of this discrepancy is that we have not yet used the extra term in the 3rd PDE that was designed to suppress div(v).

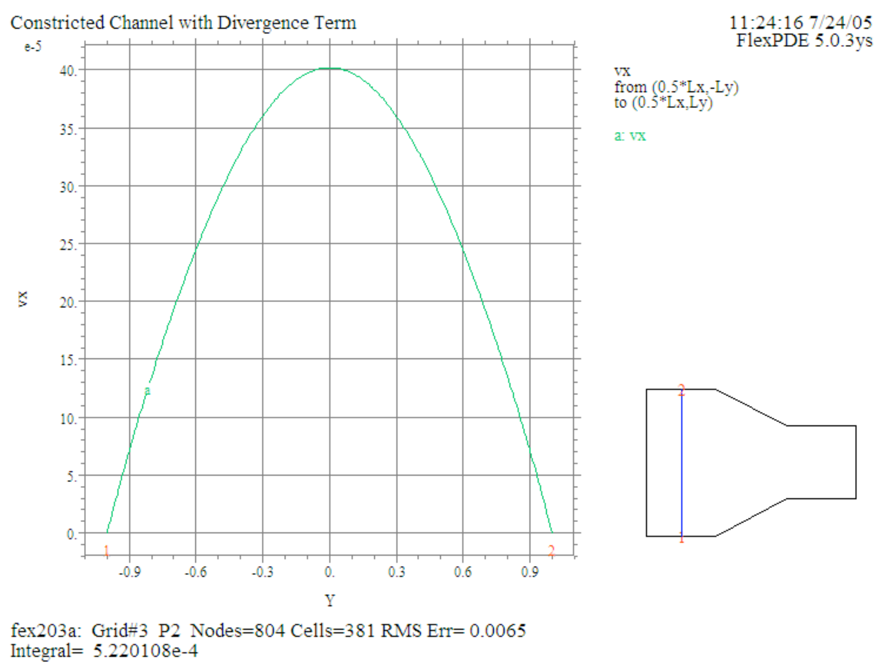
Acting on the warning received, we now introduce the term containing div(v) in the last PDE of *fex203*. The numerical factor 1e4 has been found suitable by trial and error.

```
TITLE    'Constricted Channel with Divergence Term'     { fex203a.pde }
...
EQUATIONS
  vx:        dx( p)- visc*div( grad( vx))=0
  vy:        dy( p)- visc*div( grad( vy))=0
  p:         div( grad( p))- 1e4*visc/Ly^2*div(v)=0
...
  contour( div( v))
  elevation( natp) on 'outer'
END
```

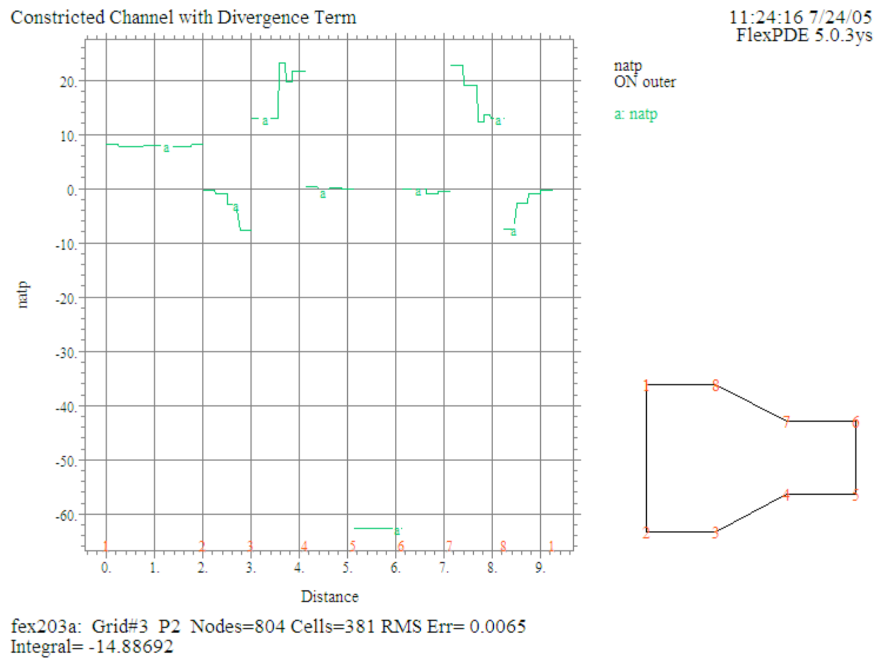
From the following plot of vx we notice that the maximum speed at the exit now is about twice that at the entrance. The plot of div(v) now exhibits the irregular contours that characterize a vanishing function.



The elevation plots across the channel both exhibit parabolic velocity profiles (below). They also demonstrate the conservation of mass and volume, because we find the two flux integral values to agree within 0.02%.



The last plot presents the variation of natp over the entire boundary. This expression contains two second-order derivatives and will hence appear as a staircase function.

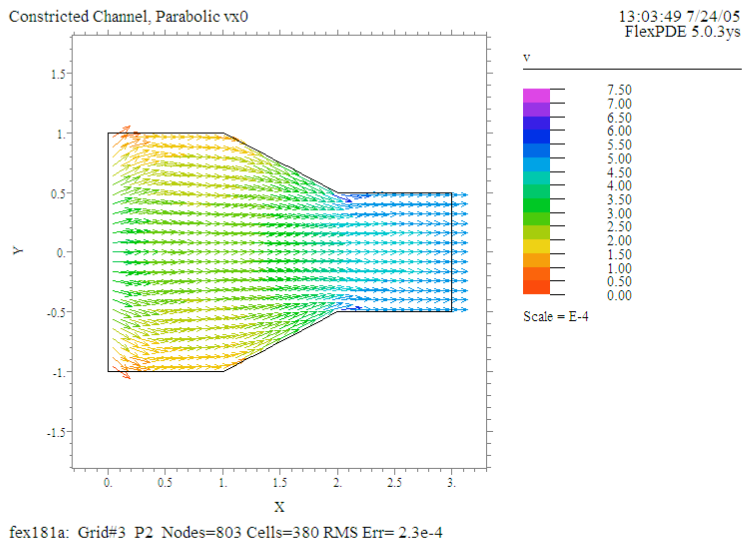


Comparison with Irrotational Flow

It might be interesting to compare viscous flow through a constricted channel with that pertaining to a velocity potential ϕ (p.226). In order to bring the boundary conditions into closer agreement we change the speed distribution at the input, such as to produce a parabolic velocity profile. The definition of vx0 in *fex181* needs to be modified, and we should adapt the plots to the new situation.

```
TITLE   'Constricted Channel, Parabolic vx0'                    { fex181a.pde }
...
  vx0=4.0e-4*(Ly^2- y^2)/Ly^2                          { Velocity at input end }
...
PLOTS
  elevation( vx0) from (0,-Ly) to (0,Ly)
  elevation( vx) from (Lx/2,-Ly) to (Lx/2,Ly)
  elevation( vx) from (3*Lx,-Ly) to (3*Lx,Ly)
  vector( v) norm
  contour( vx) painted     contour( vm) painted
  contour( div(v))     contour( curl(v))
END
```

The elevation plots illustrate that the initially parabolic distribution changes and finally becomes nearly flat at the end, in contrast to what we observed in *fex203a*. The following vector plot confirms this.



This example shows that the behavior of viscous flow is dramatically different from that of potential flow. It is possible, however, to consider viscous flow through a channel as potential flow in a region sufficiently far from the walls. The region close to the wall, where the vorticity is large (the boundary layer), may be treated separately.

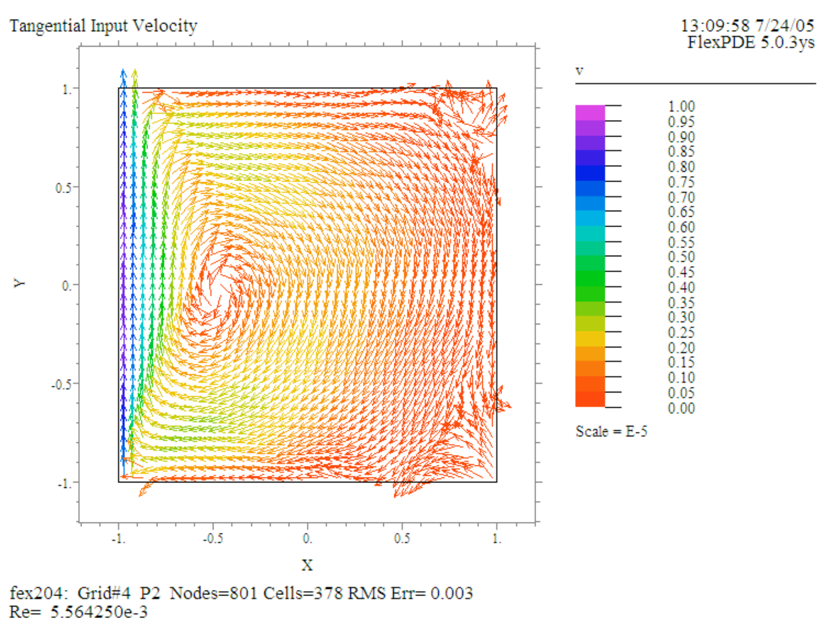
Tangential Input Velocity

We now return to an example involving a rectangular domain. Here, we specify a constant vertical velocity v_{y0} at the left face, while keeping the other three sides closed by fixed walls. In practice, we could impose this lateral velocity by an endless tape, driven over rollers at constant velocity past the left face. The template for this example is *fex203a*.

```
TITLE   'Tangential Input Velocity'                    { fex204.pde }
SELECT     errlim=1e-4     ngrid=1     spectral_colors
VARIABLES     vx     vy     p
DEFINITIONS
  Lx=1.0     Ly=1.0     visc=1.0
```

```
  vy0=1e-5                                                    { Input velocity }
  dens=1e3    Re=dens*globalmax( vx)*2*Ly/visc
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector(1,0)    unit_y=vector(0,1)      { Unit vector fields }
  nx=normal( unit_x)    ny=normal( unit_y)      { Direction cosines }
  natp=visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
EQUATIONS
  vx:         dx( p)- visc*div( grad( vx))=0
  vy:         dy( p)- visc*div( grad( vy))=0
  p:          div( grad( p))- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES
region 'domain' start 'outer' (-Lx,-Ly)
  value( vx)=0   value( vy)=0   natural(p)=natp
  line to (Lx,-Ly) to (Lx,Ly) to (-Lx,Ly)
  value( vx)= 0   value( vy)=vy0   line to close
PLOTS
  vector( v) norm report(Re)    contour( vm)
  contour( p)    contour( div( v))    contour( curl( v)) painted
END
```

We again exploit a convenient feature of FlexPDE that makes any boundary condition valid for the following segments, until modified. For instance, natp need not be repeated for each of the sides.



The above vector plot displays a kind of circulation, centered on a point not far from the left face. This is not circulation in the sense of the preceding chapter, however, because another plot shows that curl(v) is definitely non-zero. In fact, the vorticity appears to take opposite signs in different regions.

The contour plot of div(v) yields the irregular contours of value zero that we usually associate with a vanishing function.

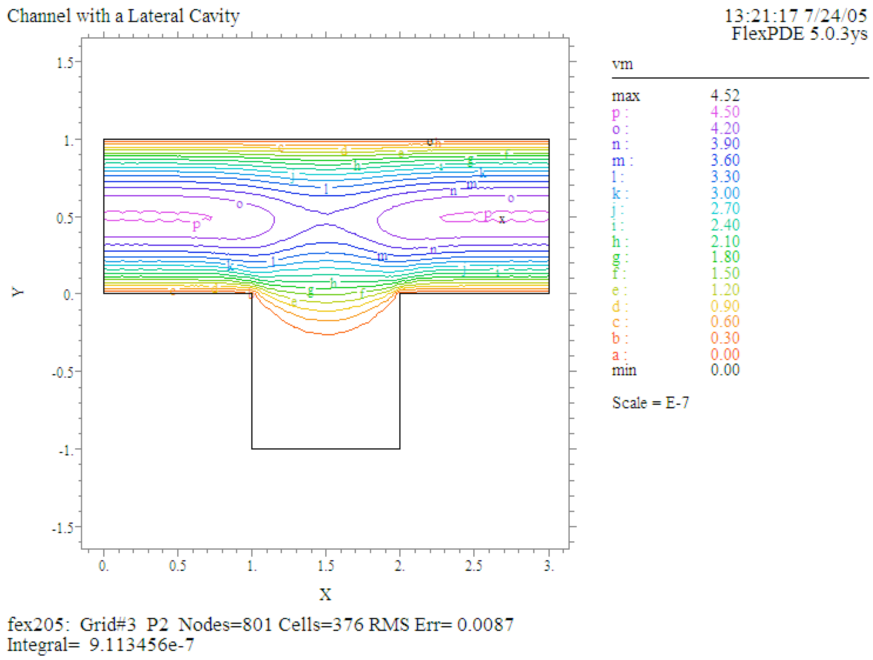
Channel with a Lateral Cavity

In *fex202*, the channel walls assured parallel flow. Let us now study the case of a channel provided with a lateral cavity as shown in the next figure. We keep a few lines from the descriptor *fex204* and modify the others as follows.

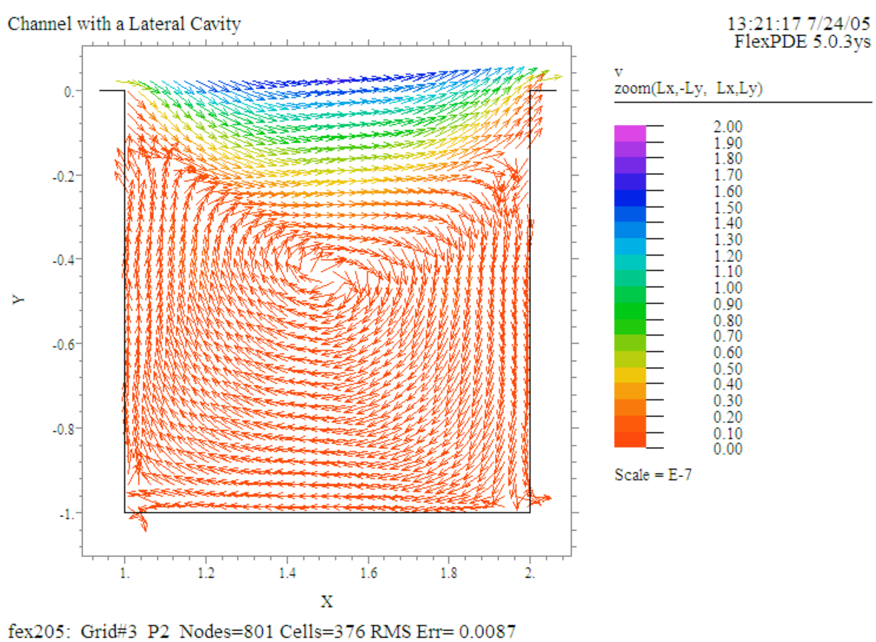
Under *boundaries*, five line segments will have equal boundary conditions, and hence we may simplify by specifying those only once.

```
TITLE    'Channel with a Lateral Cavity'                    { fex205.pde }
…
DEFINITIONS
  Lx=1.0    Ly=1.0    visc=0.1
  delp=1e-6                                           { Replaces vy0 }
...
region 'domain'  start 'outer' (0,Ly)
  natural(vx)=0   value( vy)=0   value(p)=delp         { In }
  line to (0,0)   value(vx)=0   value(vy)=0   natural(p)=natp
  line to  (Lx,0) to (Lx,-Ly) to (2*Lx,-Ly) to (2*Lx,0) to (3*Lx,0)
  natural(vx)= 0    value(p)=0    line to (3*Lx,Ly)      { Out }
  value(vx)= 0   natural(p)=natp  line to close
PLOTS
  vector( v) norm report( Re)     contour( vm)
  vector( v) norm  zoom(Lx,-Ly,  Lx,Ly)     contour( p)
  contour( div( v))     contour( curl(v)) painted
  elevation( vx) from (0.5*Lx,0) to (0.5*Lx,Ly)
  elevation( vx) from (1.5*Lx,-Ly) to (1.5*Lx,Ly)
  elevation( vx) from (2.5*Lx,0) to (2.5*Lx,Ly)
END
```

The plot of vm below shows that the flow is mainly confined to the through part of the channel, the velocity being much lower in the adjacent cavity. Clearly, this plot is symmetric with respect to the plane $x = 1.5$.



Both vector plots demonstrate that circulation occurs in the cavity, and the *curl* is again non-zero. The plot below clearly shows the center of circulation.



The plot of div(v) demonstrates that the solution is compatible with mass and volume conservation. In particular, the three elevation plots across the channel quantitatively confirm that no mass is lost along the stream.

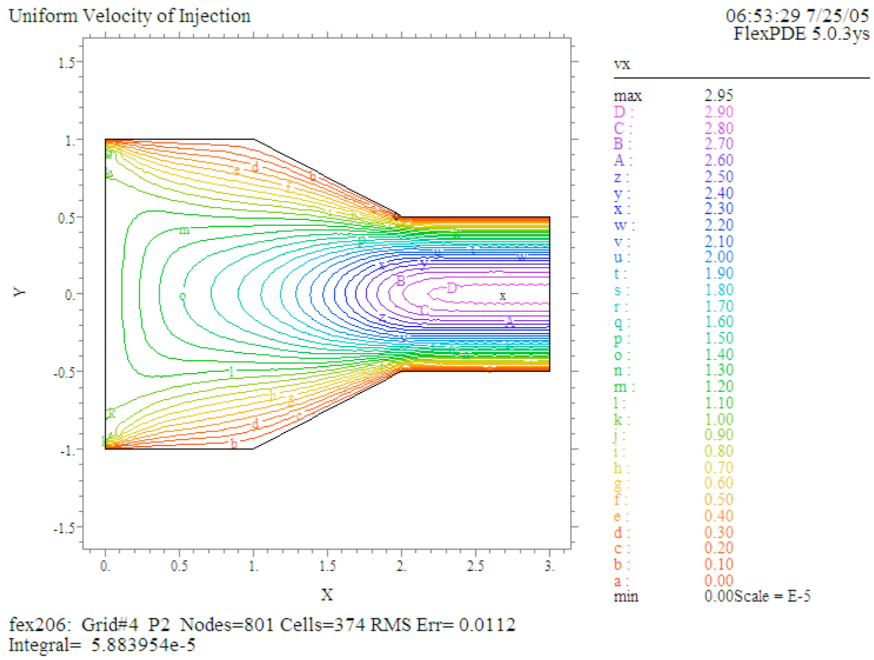
Uniform Velocity of Injection

So far we have injected fluid into a channel at uniform *pressure*. An alternative would be to impose uniform input *velocity*, resulting in a non-uniform pressure distribution over the input area. We shall now solve this problem for the constricted channel (*fex203*).

The boundary conditions for pressure are by derivatives (natural), except at the exit where we specify the value $p = 0$. To obtain the total pressure we just add the ambient value. We now modify *fex204* to obtain the following file.

```
TITLE   'Uniform Velocity of Injection'                          { fex206.pde }
SELECT     errlim=1e-3    ngrid=1    spectral_colors
VARIABLES     vx     vy     p  { Pressure minus ambient }
DEFINITIONS
  Lx=1.0     Ly=Lx     coef=0.5     visc=1.0
  vx0=1e-5                                      { Input velocity }
  dens=1e3     Re=dens*vx0*2*Ly/visc
  v=vector( vx, vy)     vm=magnitude( v)
  unit_x=vector(1,0)     unit_y=vector(0,1)       { Unit vector fields }
  nx=normal( unit_x)     ny=normal( unit_y)      {Direction cosines }
  natp=visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
EQUATIONS
  vx:          dx( p)- visc*div( grad( vx))=0
  vy:          dy( p)- visc*div( grad( vy))=0
  p:           div( grad( p))- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES
region 'domain'  start 'outer' (0,Ly)
  value( vx)=vx0   natural( vy)=0  natural( p)=natp          { In }
  line to (0,-Ly)  value( vx)=0   value( vy)=0   natural( p)=natp
  line to (Lx,-Ly) to (2*Lx,-Ly*coef) to (3*Lx,-Ly*coef)        { Wall }
  natural( vx)=0   natural( vy)=0   value( p)=0               { Out }
  line  to (3*Lx,Ly*coef)  value( vx)=0   value( vy)=0   natural( p)=natp
  line to (2*Lx,Ly*coef) to (Lx,Ly) to close                   { Wall }
PLOTS
  elevation( vx) from (0,-Ly) to (0,Ly)
  elevation( vx, 0.1*dy( vx)) from (3*Lx,-Ly*coef) to (3*Lx,Ly*coef)
  elevation( p) on 'outer'     vector( v) norm report(Re)
  contour( vx)     contour( vy)     contour( vm)
  contour( p)     contour( div( v))     contour( curl( v)) painted
END
```

The contour plot of vx below illustrates the change of the initially uniform velocity component.



The elevation plots of vx across the ends show in more detail how the initially uniform profile modifies into a parabolic one, as is clearly confirmed by the derivative. Obviously, the velocity component vx is not strictly uniform over the input, but that is caused by the discontinuity at the walls. The integrals confirm that flux is conserved.

The elevation plot of p on the boundary demonstrates that the pressure varies considerably over the input area, being highest near the walls. The output pressure, however, seems to be uniform as required.

Dynamic Similarity

We have already used the Reynolds number $\mathrm{Re} = \rho_0 v_0 L_0 / \eta$ to assess whether a given flow problem may be treated in terms of a linear PDE. The factors involved, such the typical speed v_0, are of course arbitrary to some extent. Hence, we can only expect Re to be an order-of-magnitude indicator in this connection.

Another application of Re is to exploit knowledge gained from calculation or experiment to predict the flow in an enlarged or reduced geometry – still at the same value of Re. Here, the prediction is *accurate*, as we shall see.

To reveal the similarity between situations characterized by a given value of Re, we start from the N-S vector equation. (The additional equation (p.254●1) only arranges to make $\nabla \cdot \mathbf{v} = 0$ and need not concern us here.) We thus consider the PDE

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} + \rho_0 (\mathbf{v} \cdot \nabla) \mathbf{v} - \mathbf{F} + \nabla p - \eta \nabla^2 \mathbf{v} = 0$$

The key to the prediction is a transformation of the variables into *non-dimensional* form. The new (primed) variables may be expressed as follows.

$t' = t\, v_0 / L$ (time) $\quad$ $\mathbf{r}' = \mathbf{r} / L$ (position) $\quad$ $\mathbf{v}' = \mathbf{v} / v_0$ (velocity)

$p' = p / (\rho_0 v_0^2)$ (pressure) $\quad$ $\mathbf{F}' = \mathbf{F} / (\rho_0 v_0^2 / L)$ (volume force)

The new variables t', $\mathbf{r}'$, and $\mathbf{v}'$ are obviously non-dimensional, but the reference length L and speed v_0 must be *identically defined* in the problems to be compared. For instance, we might choose the maximum value of the variable, or the value for a point at the middle of the stream. The actual *values*, however, would be different.

All five terms in the above N-S equation have the same dimension. Comparing the second and fourth terms we see that (dimensionally)

$$\rho_0 \frac{{v_0}^2}{L} \Leftrightarrow \frac{p}{L}$$

and hence that $\rho_0 v_0^2$ must have the same dimension as p. Of course, you can also see this by expanding the dimensional expressions.

A similar comparison of the third and fourth terms leads us to an expression for the non-dimensional variable $\mathbf{F}'$.

Applying the above transformations we obtain the non-dimensional form for the N-S equation.

$$\rho_0 \frac{v_0}{L/v_0} \frac{\partial \mathbf{v}'}{\partial t'} + \rho_0 \frac{v_0^2}{L} (\mathbf{v}' \cdot \nabla) \mathbf{v}' - \frac{\rho_0 v_0^2}{L} \mathbf{F}' + \frac{\rho_0 v_0^2}{L} \nabla p' - \eta \frac{v_0}{L^2} \nabla^2 \mathbf{v}' = 0$$

Multiplying through by $L/(\rho_0 v_0^2)$ gives us the simpler PDE

$$\frac{\partial \mathbf{v}'}{\partial t'} + (\mathbf{v}'\cdot\nabla)\mathbf{v}' - \mathbf{F}' + \nabla p' - \frac{1}{\mathrm{Re}}\nabla^2 \mathbf{v}' = 0 \qquad \bullet$$

From this it is clear that the solution in terms of primed variables only depends on the value of Re, if the boundary conditions are the same. Knowing the solution to one such problem we can thus generate solutions to an infinite number of problems having the same value of Re.

Let us now explore whether two problems with similar boundary conditions and proportional geometric dimensions have the same primed solutions. We first modify *fex206* to display the primed variables, using vx0 and Lx as reference values. We need not transform x and y, since the geometrical factors will be reduced automatically on plotting. In anticipation we also calculate the mean value of $|v'|$ (vpm) to facilitate comparison.

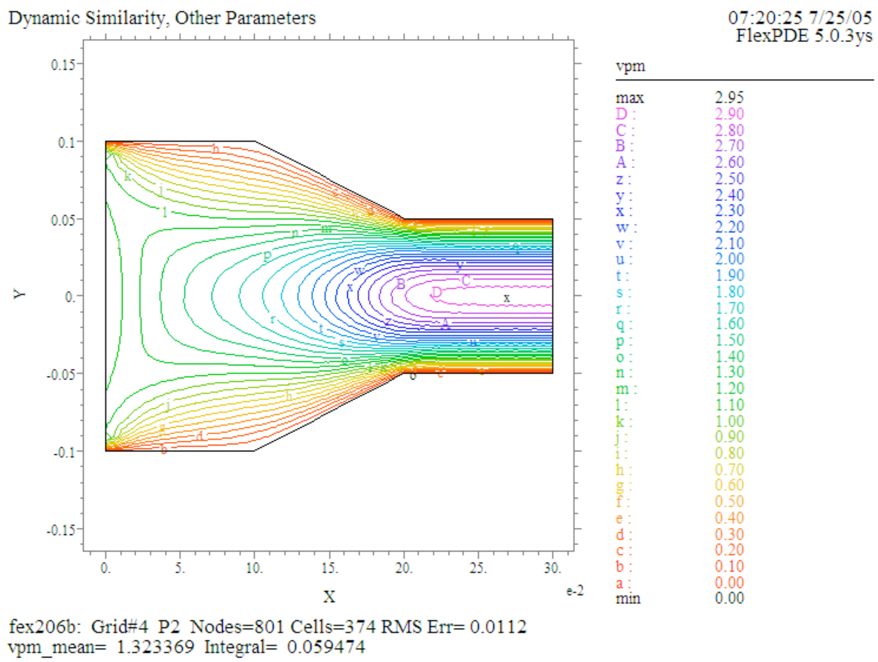
```
TITLE   'Dynamic Similarity'                                { fex206a.pde }
…                                                     { Primed variables: }
  vxp=vx/vx0     vyp=vy/vx0     vp=vector( vxp, vyp)
  vpm=magnitude( vp)     pp=p/(dens*vx0^2)
  area=area_integral(1)     vpm_mean=area_integral( vpm)/area
EQUATIONS
…
PLOTS
  vector( vp) norm report(Re)     contour( vpm) report(vpm_mean)
  contour( pp)     contour( abs( pp)/area)
END
```

We are now ready to compare to a problem with other parameters. Since Re depends on four quantities, we must change at least two of them to produce the same value. In the following descriptor, based on *fex206a*, we modify three factors.

```
TITLE   'Dynamic Similarity, Other Parameters'              { fex206b.pde }
…
  Lx=0.1     Ly=Lx     coef=0.5     visc=0.01
  vx0=1e-6
…
```

If we make both scripts show the plots, enlarging the second plot of each, we may compare corresponding figures quickly by clicking on the tabs at the top. We can then proceed similarly with the other plots.

We find that the following plot reports the same mean value for the magnitude vpm to five digits. It is possible to transform back to unprimed variables in order to obtain results in the usual form.



From the third plot it appears that pp varies over the same range in the two descriptors. In order to make a more accurate comparison, we could integrate the results. Considering that the geometrical sizes are different (by a factor of 100), we should also divide by the area to obtain mean values. These also turn out to be nearly equal.

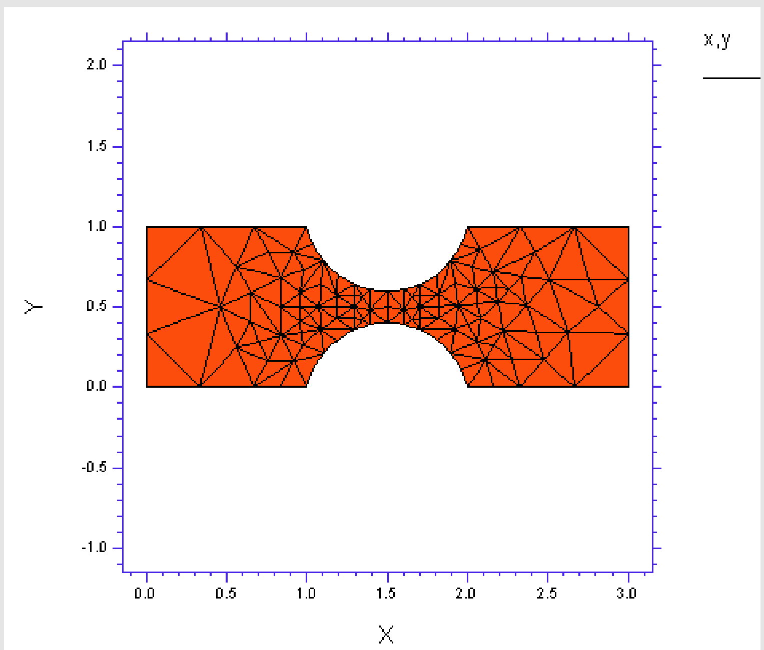
We finally note that the dimensional expression for the ratio of the inertial-to-viscous terms is

$$\left|\frac{\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v}}{\eta\nabla^2\mathbf{v}}\right| \cong \frac{\rho_0 v_0^2/L_0}{\eta v_0/L_0^2} = \mathrm{Re} \qquad \bullet$$

which means that Re is a rough measure of the importance of the non-linear term in the PDE.

Exercises

❑ Show analytically that the function vx_ex (p.260) satisfies the PDEs and the boundary conditions.
❑ Verify the numeric calculation of curl(v) in *fex202* using the function vx_ex.
❑ Modify *fex203a* such as to produce a sudden constriction at $x = L_x$.
❑ Modify *fex203a* to produce a sudden widening of the channel at $x = L_x$. Use Ly=0.4 and coef=2.0.
❑ Explore the results of *fex206* using coef=1.0 and coef=2.0. Repeat the solution for Lx=2.0.
❑ Create circular constrictions on the channel in *fex205* as indicated by the figure below. Let the minimum channel width be 0.2.



21 Viscous Flow past an Obstacle

We shall now study slow viscous flow in a channel containing an obstacle. The practical difference with respect to the preceding chapter is that we shall have to exclude a region corresponding to the obstacle and specify boundary conditions on its surface.

Viscous Flow past a Circular Cylinder

Here, we revisit an example from the chapter on irrotational flow (*fex182*, p.231). We need to add the PDEs for viscous flow and the pertinent boundary conditions, using the convenient formulation for a general orientation (natp) from *fex203*.

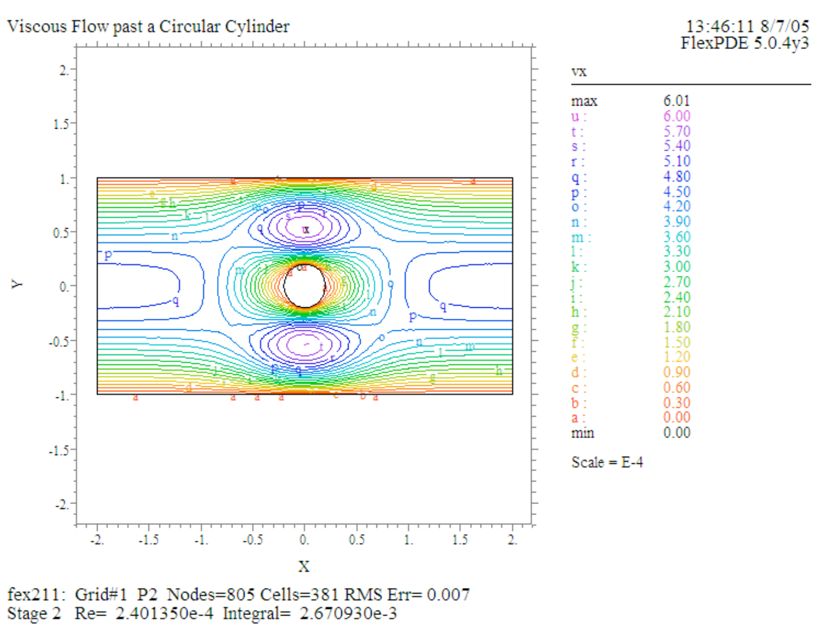
Empirically it has been found that natp=0 often is a sufficiently good approximation to the full expression, at least for Re<<1. In the next example we test this simplification by successive runs, using the stages device. The program sets the parameter stage to be 1 in the first run and 2 in a second run, where we use the full natp.

```
TITLE     'Viscous Flow past a Circular Cylinder'              { fex211.pde }
SELECT    errlim=1e-3    ngrid=1    spectral_colors    stages=2
VARIABLES    vx    vy    p
DEFINITIONS
  Lx=2.0    Ly=1.0    a=0.2    visc=1e4
  delp=100                                            { Driving pressure }
  dens=1e3    Re=dens*globalmax( vx)*2*Lx/visc
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector(1,0)    unit_y=vector(0,1)      { Unit vector fields }
  nx=normal( unit_x)    ny=normal( unit_y)      { Direction cosines }
  natp=
     if stage=2 then visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]  else 0
EQUATIONS
  vx:        dx( p)- visc*div( grad( vx))=0
  vy:        dy( p)- visc*div( grad( vy))=0
  p:         div( grad( p))- 1e4*visc/Ly^2*div( v)=0
```

```
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0   natural( vy)=0    value(p)=delp                    { In }
  line to (-Lx,-Ly)  value( vx)=0   value( vy)=0   natural(p)=natp { Wall}
  line to (Lx,-Ly)  natural( vx)=0   natural( vy)=0   value(p)=0     { Out }
  line to (Lx,Ly)   value( vx)=0   value( vy)=0   natural(p)=natp    { Wall }
  line to close
  start 'outline' (a,0)                                       { Exclude cylinder }
  value( vx)=0  value( vy)=0   natural(p)=natp
  arc( center=0,0) angle=360 close
PLOTS
  contour( vx) report( Re)     contour( vy)
  contour( vm) painted     contour( p)
  vector( v) norm     vector(v) norm  zoom(-2*a,-2*a, 4*a,4*a)
  contour( div( v))     contour( curl( v)) painted
  elevation( vx, vy) from (-Lx,0) to (Lx,0)
END
```

The program runs the script in two stages. By clicking on *File,View* we may easily compare the results with natp=0 to those exploiting the full expression for natp. We only need to select the two plots and then switch from one to the other by means of *Ctrl-Shift n* (for *n*ext) and *Ctrl-Shift b* (for *b*ack).

The following plot for stage=2 illustrates that the speed vanishes on the solid surfaces and that the maximum speed occurs approximately midway between the cylinder and the wall.

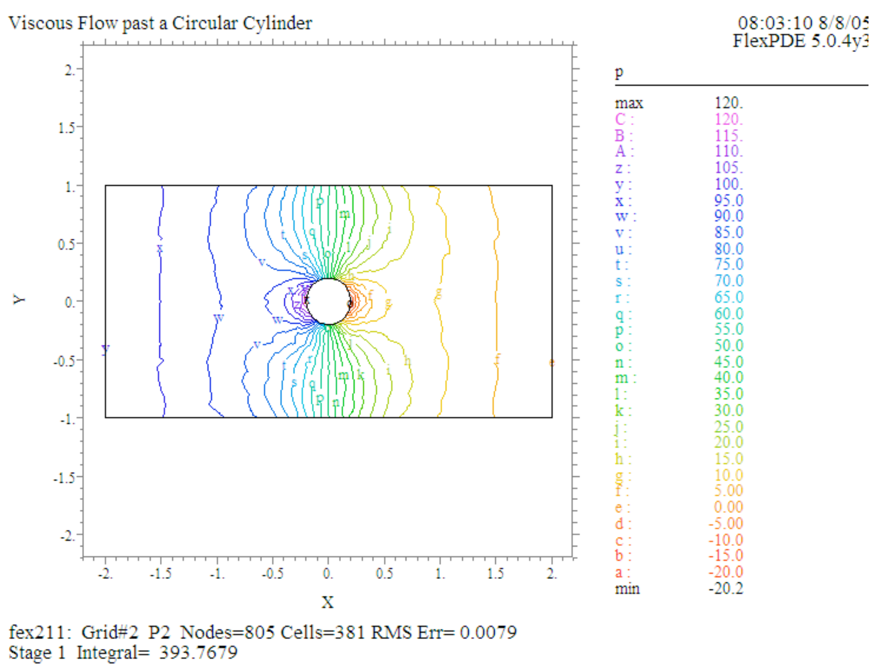


Switching between stages 1 and 2 we find no visible change of the contours of vx, and the integral values differ only in the fourth decimal. The variation is about as small for vm and p. For the rest of the examples in this chapter we shall thus replace natp by 0 on the basis of experience.

The information contained here and in the vector plots indicates that the flow is symmetric with respect to $y = 0$. Thus, there is no circulation of the liquid around the obstacle.

In addition, the above plot suggests that the speed is symmetric with respect to $x = 0$, as also appears from the two vector plots of **v**. The final elevation plot illustrates this symmetry in more detail.

The next figure illustrates the pressure field. We notice that there are high values on the front side of the obstacle and negative values on the rear side. The effect of this is to create a pressure force on the obstacle, in addition to viscous drag.



To the above pressure values we may add the ambient pressure (1e5), which makes the total pressure positive everywhere.

Viscous Force on a Solid Surface

In order to gain deeper insight, we shall consider the forces on the walls and on a solid cylinder in the channel. In an earlier chapter (p.245) we only calculated the force due to pressure, but we must now include the effects of viscosity. For a solid surface perpendicular to the y-axis, the definition of viscosity directly gives us the viscous force per unit area[8p4], i.e.

$$f_x = \eta \frac{\partial v_x}{\partial y}$$

For a solid surface of arbitrary orientation we may write the tangential force per unit area as

$$f_t = \eta \frac{\partial v_t}{\partial n}$$

With $v_t = \mathbf{v} \cdot \mathbf{t} = v_x t_x + v_y t_y$, where $\mathbf{t}$ is the tangential unit vector, we obtain the general expression

$$f_t = \eta \frac{\partial v_t}{\partial n} = \eta \left(\frac{\partial v_x}{\partial n} t_x + \frac{\partial v_y}{\partial n} t_y \right)$$

or after expanding the derivatives

$$f_t = \eta \left\{ \left(\frac{\partial v_x}{\partial x} \frac{\partial x}{\partial n} + \frac{\partial v_x}{\partial y} \frac{\partial y}{\partial n} \right) t_x + \left(\frac{\partial v_y}{\partial x} \frac{\partial x}{\partial n} + \frac{\partial v_y}{\partial y} \frac{\partial y}{\partial n} \right) t_y \right\}$$

$$f_t = \eta \left\{ \left(\frac{\partial v_x}{\partial x} n_x + \frac{\partial v_x}{\partial y} n_y \right) t_x + \left(\frac{\partial v_y}{\partial x} n_x + \frac{\partial v_y}{\partial y} n_y \right) t_y \right\} \qquad \bullet$$

where we have used the components of the normal unit vector $\mathbf{n}$. For the Cartesian components of this force per unit area we obtain

$$f_x = f_t \, t_x, \quad f_y = f_t \, t_y \qquad \bullet$$

After having developed the expressions required, we now return to the example of the circular cylinder to explore the forces caused by the flow.

We are now in a position to calculate the forces occurring in *fex211*. On the parallel walls, the drag forces will only be of viscous nature, while the obstacle is also exposed to unbalanced pressure. The above expressions for force per unit area contain direction cosines, and as we have already seen FlexPDE provides simple expressions for these, referred to a boundary curve.

Instead of reading off integrals on several different elevation plots, we prefer to use line_integral under *definitions*. The values obtained can then be reported as a summary.

Evidently, there are rather many expressions related to velocities and forces, and we will find it expedient to store them in an include file named *visc_xy*.

```
{ Include file related to viscous flow in (x,y) }                 { visc_xy.pde }
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector(1,0)    unit_y=vector(0,1)      { Unit vector fields }
  nx=normal( unit_x)    ny=normal( unit_y)      { Direction cosines }
  tx=tangential( unit_x)    ty=tangential( unit_y)
  natp=visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
  force_vt=-visc*[ (dx(vx)*nx+ dy(vx)*ny)*tx+
    (dx(vy)*nx+ dy(vy)*ny)*ty]                  { Viscous force }
  force_vx=force_vt*tx
  force_vy=force_vt*ty
  force_px=p*nx                                 { Pressure force }
  force_py=p*ny
  force_x=force_vx+ force_px
  force_y=force_vy+ force_py
```

This file must be saved in the same folder as the other descriptors.

The force components on the obstacle are caused both by viscosity and by pressure. The normal and tangential vectors also have signs, and we may verify the signs assumed by FlexPDE by plotting (nx,ny,tx,ty). Alternatively, we may check the signs by plotting each force component.

Let us now introduce the above commands into *fex211* and calculate the various force components. Here, we employ the standard name 'outline' for the contour of the obstacle.

```
TITLE     'Flow past a Circular Cylinder, Forces'            { fex211a.pde }
SELECT     errlim=1e-3     ngrid=1     spectral_colors
VARIABLES     vx      vy       p
DEFINITIONS
  Lx=2.0     Ly=1.0     a=0.2     visc=1e4
  delp=100                                           { Driving pressure }
  dens=1e3     Re=dens*globalmax( vx)*2*Lx/visc
#include 'visc_xy.pde'
  F_wall_x=line_integral( force_vx,'outer')     { Force on walls }
  F_vx=line_integral( force_vx,'outline')       { Viscous force }
  F_px=line_integral( force_px,'outline')       { Pressure force }
  F_x=line_integral( force_x,'outline')         { Sum of x-forces }
  F_vy=line_integral( force_vy,'outline')       { Viscous force }
  F_py=line_integral( force_py,'outline')       { Pressure force }
  F_y=line_integral( force_y,'outline')         { Sum of y-forces }
EQUATIONS
  vx:         dx( p)- visc*div( grad( vx))=0
  vy:         dy( p)- visc*div( grad( vy))=0
  p:          div( grad( p))- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0   natural( vy)=0    value(p)=delp                  { In }
  line to (-Lx,-Ly)   value( vx)=0   value( vy)=0   natural(p)=0    { natp=0 }
  line to (Lx,-Ly)  natural( vx)=0   natural( vy)=0   value(p)=0    { Out }
  line to (Lx,Ly)   value( vx)=0   value( vy)=0   natural(p)=0      { natp=0 }
  line to close
  start 'outline' (a,0)                         { Exclude }
  value( vx)=0  value( vy)=0   natural(p)=0                         { natp=0 }
  arc( center=0,0) angle=360 close
PLOTS
  contour( vx) report( Re)     elevation( force_vx) on 'outline'
summary
  report(F_wall_x)
  report(   F_vx) report(   F_px) report(F_x)
  report(   F_vy) report(   F_py) report(F_y)
END
```

The last lines assemble the integral values under the common title summary. FlexPDE lists all results in one column as shown below.

```
Flow past a Circular Cylinder, Forces                    08:07:22 8/9/05
                                                         FlexPDE 5.0.4y3

SUMMARY

F_wall_x= 119.1550
  F_vx= 34.49756
  F_px= 44.55249
F_x= 79.05005
  F_vy= 0.080583
  F_py= 0.287250
F_y= 0.367833
```

We first notice that the drag force F_px on the cylinder due to pressure is of the same order of magnitude as the viscous force. Evidently, the total vertical force F_y is smaller than the drag force by a factor of about 500, consistent with vanishing lift force due to symmetry.

Force Equilibrium

It is also interesting to compare the forces acting on the volume of the *liquid*. In addition to those listed in the above table we have the forces due to the pressure at the left and right ends, the sum being delp*2*Ly=200. Although the liquid accelerates locally as it flows through various regions, the mass does not accelerate as a whole. In other words, we expect the forces acting on the liquid to balance.

From the above table we gather that the drag force on the walls is 119.2 and that on the cylinder 78.8. The forces acting on the liquid are the *negative* of these values, or in total –198.0. Thus the forces on the liquid volume balance to within 1%. Using the Professional Version we may reduce this error to a very small value, at the expense of a longer run time.

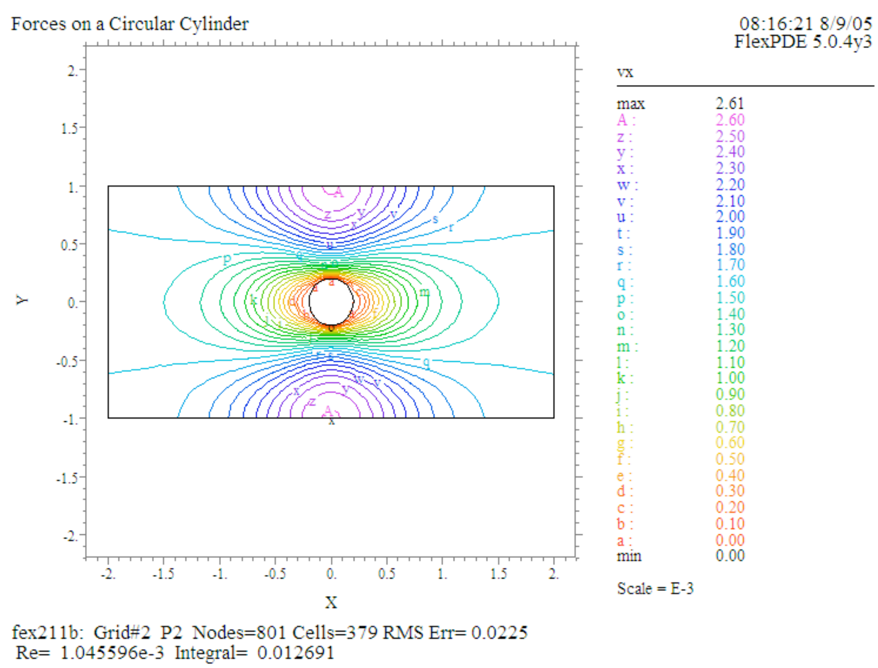
Let us now calculate the force on a solid object by a simpler method, using the following modification of *fex211a*.

```
TITLE    'Forces on a Circular Cylinder'        { fex211b.pde }
…
region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0   natural( vy)=0   value(p)=delp                  { In }
  line to (-Lx,-Ly)  natural( vx)=0   value( vy)=0   natural(p)=0
  line to (Lx,-Ly)  natural( vx)=0   natural( vy)=0   value(p)=0     { Out }
```

```
line to (Lx,Ly) natural( vx)=0   value( vy)=0   natural(p)=0
line to close
…
```

The difference is that we now specify essentially *slip* (natural) boundary conditions on the walls. This increases the average speed and reduces the viscous force on the wall to negligible proportions. We thus expect the drag force on the object to balance the pressure force on the liquid domain.

The plot of vx below shows that the speed takes its maximum near the wall. This increase is caused by the constriction of the flow due to the presence of the obstacle.



The drag force reported in the table below is F_x=194.2, while we obtain delp*2*Ly=200 for the same force from the applied pressure, the difference being about 3%. The viscous force on the wall (F_wall_x) is evidently negligible. Using the Professional Version we again obtain much better agreement.

```
Forces on a Circular Cylinder                         08:16:21 8/9/05
                                                      FlexPDE 5.0.4y3
SUMMARY

F_wall_x= -0.106168
  F_vx= 91.82783
  F_px= 102.3899
F_x= 194.2177
  F_vy= -0.010765
  F_py= 0.220998
F_y= 0.210233
```

Viscous Dissipation

Motion in a viscous medium involves internal friction that will generate heat. From an expression for the rate of change of kinetic energy and the N-S equation one obtains[8p193,9p153] for the dissipated power per unit volume

$$P_d = \eta \sum_{ik} \frac{\partial v_i}{\partial k}\left(\frac{\partial v_i}{\partial k} + \frac{\partial v_k}{\partial i}\right)$$

where *i* and *k* are understood to run through the symbols *x* and *y*. In explicit form this sum expands into

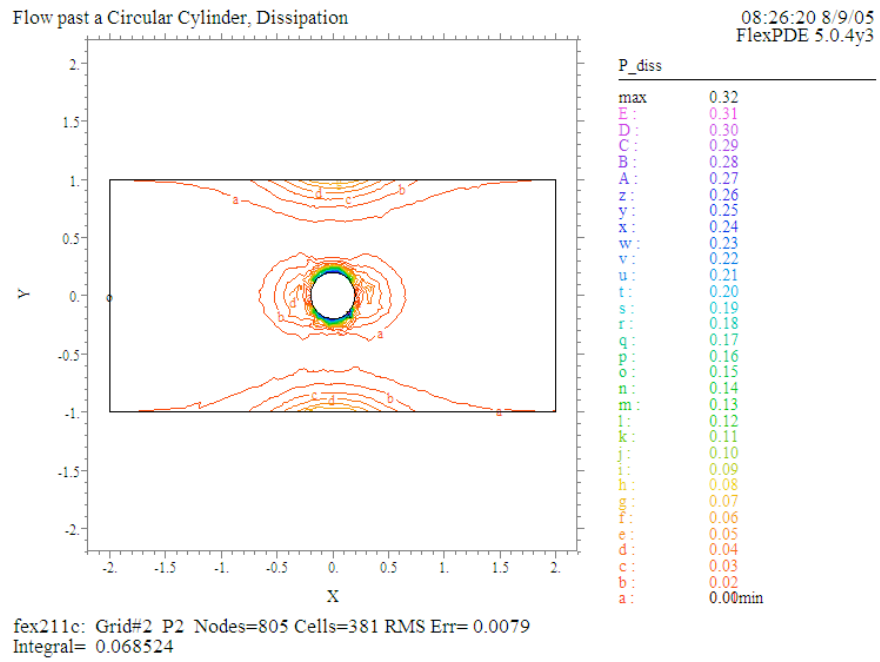
$$P_d = \eta \left\{ 2\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y}\right)^2 + \left(\frac{\partial v_y}{\partial x}\right)^2 + 2\frac{\partial v_x}{\partial y}\frac{\partial v_y}{\partial x} + 2\left(\frac{\partial v_y}{\partial y}\right)^2 \right\}$$

$$P_d = \eta \left\{ 2\left(\frac{\partial v_x}{\partial x}\right)^2 + \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)^2 + 2\left(\frac{\partial v_y}{\partial y}\right)^2 \right\} \quad \bullet$$

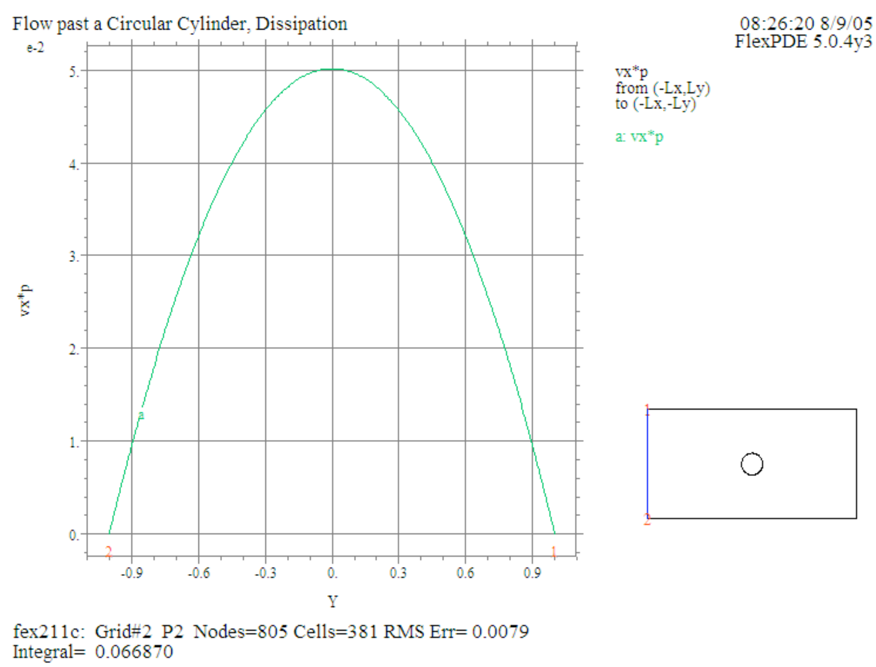
The following descriptor, which is a modified *fex211a*, explores the energy balance by comparing the dissipated power with the rate of work done at the entrance.

```
TITLE     'Flow past a Circular Cylinder, Dissipation'        { fex211c.pde }
…
  P_diss=visc*[2*dx(vx)^2+ (dy(vx)+ dx(vy))^2+ 2*dy(vy)^2]
EQUATIONS
…
PLOTS
  contour( vm) painted     contour( P_diss)
  elevation( vx*p) from (-Lx,Ly) to (-Lx,-Ly)
END
```

The following plot shows that the dissipated power is largest on the solid surfaces and close to the speed maximum, which could be expected.



The elevation plot below permits us to compare this dissipated power (0.0685 per unit length in z) with the expended work (0.0669) on driving the liquid through the channel. The integral values for the rates of dissipated energy and work evidently agree rather well.



Drag and Lift on an Inclined Plate

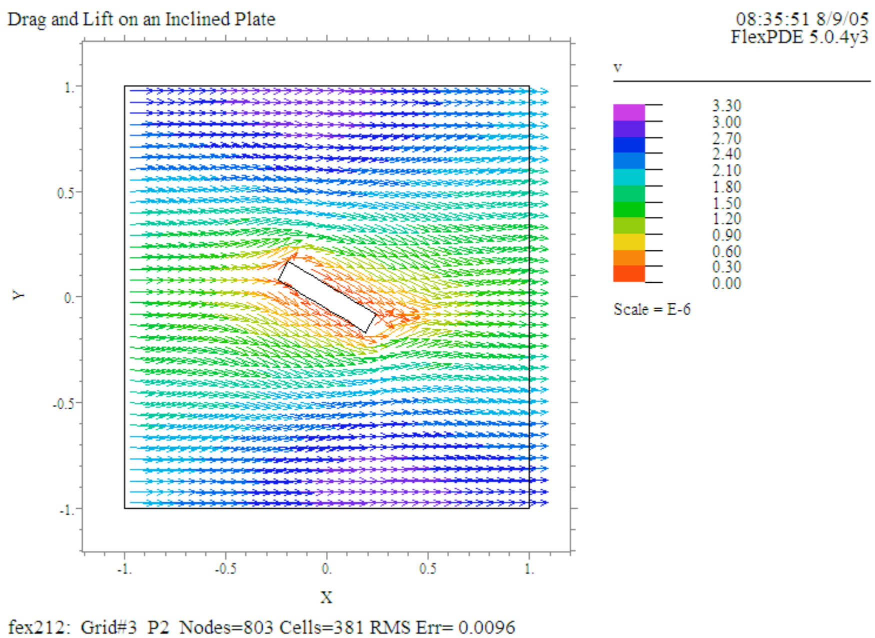
Having developed formalisms for forces we can now extend the analysis to an inclined plate, combining features of *fex193* and *fex211b*. The elevation plot on p.249 suggests that the expressions for the force components, involving derivatives of vx and vy, would be difficult to integrate because of the sharp corners. Hence, we prefer the boundary force formalism, based on the integrated force on the liquid volume.

```
TITLE    'Drag and Lift on an Inclined Plate'                    { fex212.pde }
SELECT    errlim=1e-3    ngrid=1    spectral_colors
VARIABLES    vx     vy     p
DEFINITIONS
  Lx=1.0    Ly=1.0    a=0.5    d=0.1    visc=1.0
{ Geometric parameters for inclined plate }
     alpha=30* pi/180                         { Angle of attack, radians }
     si=sin( alpha)    co=cos( alpha)
     x1=-d/2*si- a/2*co    y1=-d/2*co+ a/2*si
     x2=d/2*si- a/2*co    y2=d/2*co+ a/2*si
     x3=-x1    y3=-y1    x4=-x2    y4=-y2
  delp=1e-5                                            { Driving pressure }
  dens=1e3    Re=dens*globalmax( vx)*2*Lx/visc
#include 'visc_xy.pde'
EQUATIONS
  vx:        dx( p)- visc*div( grad( vx))=0
  vy:        dy( p)- visc*div( grad( vy))=0
  p:         div( grad( p))- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0   natural( vy)=0   value(p)=delp                { In }
  line to (-Lx,-Ly)   natural( vx)=0   value( vy)=0   natural(p)=0
  line to (Lx,-Ly)  natural( vx)=0   natural( vy)=0   value(p)=0     { Out }
  line to (Lx,Ly)   natural( vx)=0   value( vy)=0   natural(p)=0
  line to close
  start 'outline' (x1,y1)                              { Exclude }
  value( vx)=0  value( vy)=0   natural(p)=0
  line to (x2,y2) to (x3,y3) to (x4,y4) to close
PLOTS
  contour( vx) report( Re)    contour( vm) painted    vector( v) norm
  contour( p) painted
  elevation( visc*dy( vx)) from (-Lx,-Ly) to (Lx,-Ly)    { Fx on "wall" }
```

```
  elevation( p) from (-Lx,-Ly) to (Lx,-Ly)        { Force_y on liquid }
  elevation( p) from (-Lx,Ly) to (Lx,Ly)
  elevation( -p*normal( unit_y)) on 'outer'       { Total Fy on plate }
END
```

Here, we let the liquid slip over the upper and lower boundaries in order to reduce the viscous drag on the walls compared to that acting on the obstacle.

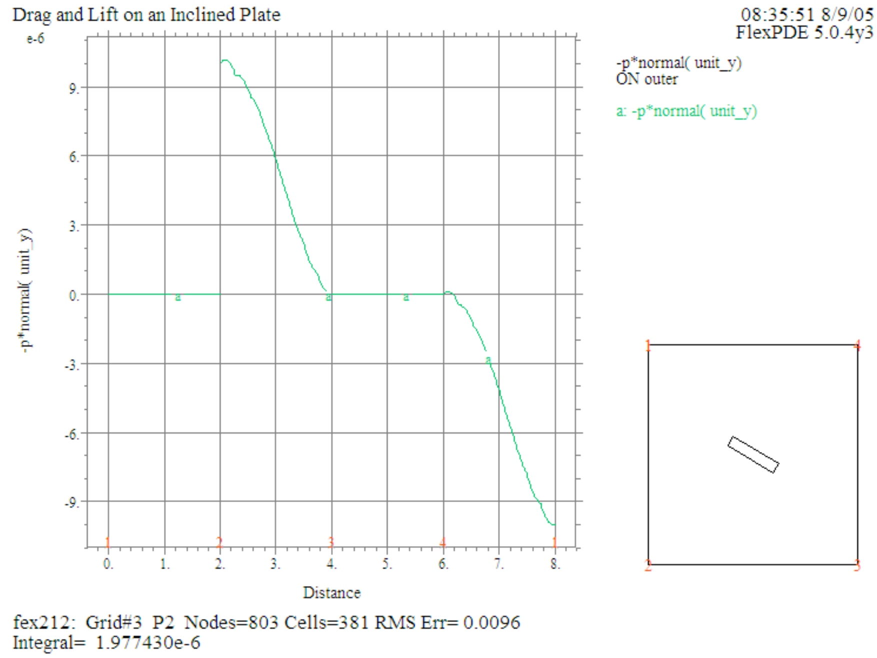
The figure below shows the velocity distribution. The vector field may look somewhat like that on p.249, but notice that the speed now vanishes on the surface of the plate.



From the first elevation plot we find that the viscous force on the lower boundary is about –1.9e-8. The force on this boundary is thus negligible compared with the pressure force on the entrance (2.0e-5).

The second and third elevation plots, with their integrals, yield the *y*-component of the force. The pressure is higher on the lower wall, and the result is that the *liquid* is subjected to a lift force of about 1.98e-6, which becomes transmitted to the *plate*.

The following plot on the outer boundary *combines* the two preceding ones on the walls. The factor normal(unit_y) eliminates the contributions from the ends and provides signs for the major forces.



We have thus found that the combined force from both walls is small compared to the force (2.0*delp=2e-5) applied by the driving pressure. The lift force is only about 10% of the drag force on the plate, which demonstrates that an airplane does not fly well at small speed in a highly viscous medium.

Exercises

❑ From *fex211a* it might appear that the force on the wall is equal to that on the obstacle. Change the radius to a=0.3 to decide whether this is true.

❑ Change *fex211a* to calculate the drag force on a bar of square cross-section, the side length being equal to the previous diameter.

❑ Calculate the viscous dissipation for a square obstacle across the channel. Compare to the work supplied at the ends of the channel.

❑ Repeat the preceding exercise for a bar rotated by 45^0 around its axis.

❑ Investigate how the drag and lift forces vary with the angle of attack (alpha) in *fex212*. Try 0, 20 and 40 degrees.

22 Irrotational Flow in (ρ,z) Space

In earlier chapters we solved flow problems in (x, y) space. The present software also permits us to treat certain three-dimensional problems as two-dimensional, specifically in cases where the geometry, the forces, and the boundary conditions are axially symmetric. To achieve this, we need to convert the descriptors to *cylindrical coordinates*, i.e. (ρ, φ, z).

For irrotational flow without circulation (p.227) the PDE required is

$$\nabla^2\phi \equiv \mathrm{div}(\mathrm{grad}\phi) = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0$$

The expression for the gradient is similar in cylindrical coordinates, but the divergence[6p82] takes a different form in (ρ, z), viz.

$$\nabla \cdot \mathbf{v} = \frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho v_\rho\right) + \frac{\partial v_z}{\partial z} \qquad \bullet$$

Using the pertinent definitions of the velocity components

$$v_\rho = \frac{\partial\phi}{\partial\rho}, \qquad v_z = \frac{\partial\phi}{\partial z} \qquad \bullet$$

the above PDE now becomes

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\phi}{\partial\rho}\right) + \frac{\partial^2\phi}{\partial z^2} = 0 \qquad \bullet$$

This equation may be solved as easily as the one in (x, y), after applying the boundary conditions.

Constricted Tube

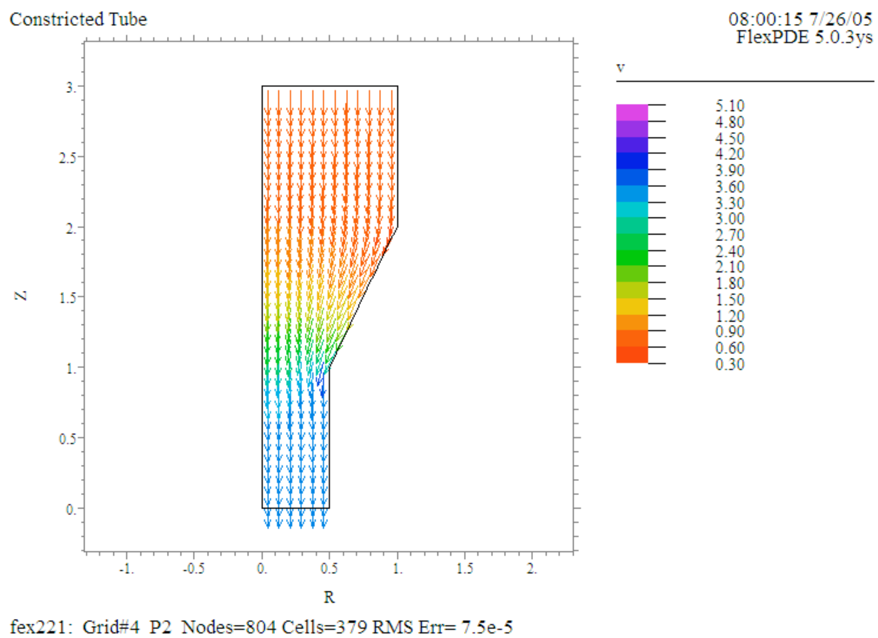
We shall first apply this equation to a problem somewhat similar to *fex181*, which involved a constricted channel. The descriptor below defines a tube of circular cross-section with a reduced radius in the lower part. We shall neglect the effect of gravity.

In the descriptor below we declare cylindrical coordinates, now called (r,z), in a special segment. The command ycylinder means that we choose the previous *y*-axis to be the axis of symmetry.

```
TITLE    'Constricted Tube'                                      { fex221.pde }
SELECT    errlim=1e-5    ngrid=1    spectral_colors
COORDINATES    ycylinder('r','z')                        { Student Version }
VARIABLES    phi
DEFINITIONS
  r0=0.5    r1=1.0    L=1.0
  vz1=1.0    p1=1e5    dens=1e3
  vr=dr( phi)    vz=dz( phi)
  v=vector( vr, vz)    vm=magnitude( v)
  p=p1+ 0.5*dens*(vz1^2-vm^2)
  div_v=1/r*dr( r*vr)+dz( vz)    curl_v=dz( vr)-dr( vz)
  K1l=line_integral( 2*pi*r*(-vz)*0.5*dens*vm^2, 'upper'){ Kinetic E. }
  K1=surf_integral( -vz*0.5*dens*vm^2, 'upper')
  K0=surf_integral(-vz*0.5*dens*vm^2, 'lower')
  W1=surf_integral(-vz*p, 'upper')                          { Work }
  W0=surf_integral(-vz*p, 'lower')
EQUATIONS
  (1/r)*dr( r*dr(phi))+ dzz( phi)=0               { Gravity neglected }
BOUNDARIES
region 'domain'   start 'outer' (r1,3*L)
  natural( phi)=-vz1    line to (0,3*L)             { In }
  natural( phi)=0    line to (0,0)
  value( phi)=0    line to (r0,0)                   { Out }
  natural( phi)=0    line to (r0,L) to (r1,2*L) to close
feature    start 'upper' (r1,3*L) line to (0,3*L)  { Lines for integration }
  start 'lower' (0,0) line to (r0,0)
PLOTS
  contour( vm) painted    contour( p) painted
  contour( curl_v)    contour( div_v)    vector( v) norm
  elevation( p) from (0,3*L) to  (0,0)    elevation( vz) on 'outer'
summary
  report(K1l) report(K1) report(K0)
```

```
  report(W1) report(W0)
  report(W1- W0) report(K0- K1)
END
```

At the top we inject liquid downward at uniform speed. From the plot below we see that the flow causes increased speed and consequently a pressure drop as the stream narrows toward the lower end. To imagine a 3D picture one has to rotate this figure around the vertical axis of symmetry.



The pertinent *curl* component is $(\nabla \times \mathbf{v})_\varphi$, which is perpendicular to the (ρ, z) plane, pointing into the screen. We may obtain that from the determinant expression[6p83]

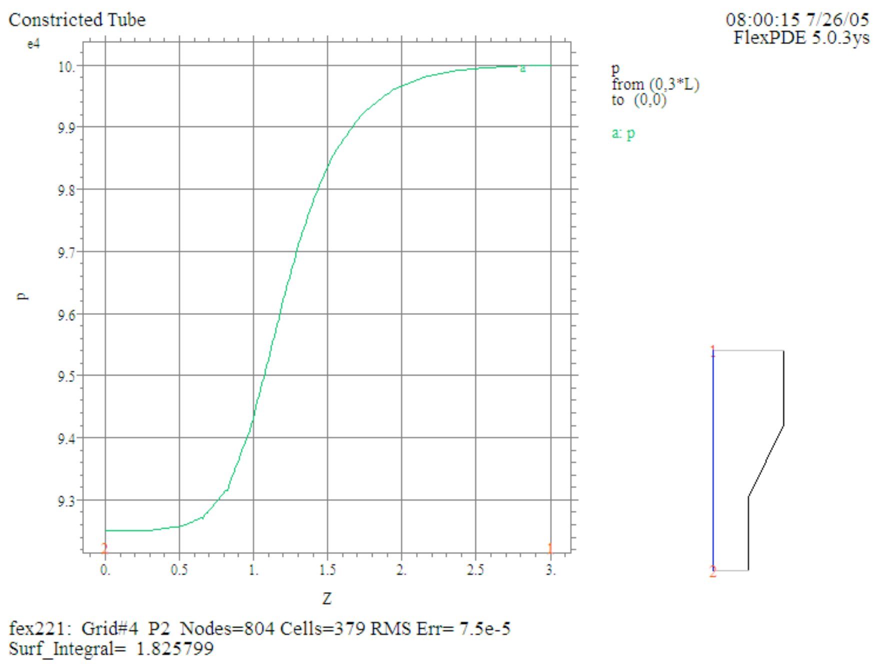
$$\nabla \times \mathbf{v} = \frac{1}{\rho} \begin{vmatrix} \mathbf{e}_\rho & \rho \mathbf{e}_\varphi & \mathbf{e}_z \\ \dfrac{\partial}{\partial \rho} & \dfrac{\partial}{\partial \varphi} & \dfrac{\partial}{\partial z} \\ v_\rho & \rho v_\varphi & v_z \end{vmatrix} \qquad \bullet$$

as $(\nabla \times \mathbf{v})_\varphi = \dfrac{\partial v_\rho}{\partial z} - \dfrac{\partial v_z}{\partial \rho}$.

The contour plot of curl_v evidently yields exactly zero, which shows that the flow is irrotational, as we could also have inferred from p.290●2. The plot of div_v confirms that the flow also is divergence-free.

The energy balance in this example is worth studying. The liquid injected at the top eventually exits at the bottom at four times higher speed. This means that the kinetic energy of a horizontal layer of liquid increases considerably in going from entrance to exit. The forces acting on the liquid at the entrance and exit surfaces supply this increased kinetic energy. The entrance pressure acts on a larger area and the pressure is also smaller at the exit, which means that the work done is positive.

The following figure illustrates that the pressure decreases toward the exit.



Let us compare the kinetic energy of the mass injected during a small time interval δt to that ejected during the same interval. The vertical displacement of the liquid during this time is $-v_z\delta t$. In the *definitions* segment we prepare for the calculation of the kinetic energies *per unit time* (K0, etc.), i.e. we have divided by δt.

Under feature we define the two radial lines that represent the circular ends of the liquid volume. Since we have declared cylindrical coordinates, FlexPDE assumes that there is axial symmetry. An automatic integral for an elevation plot along a radial line would thus refer to the corresponding circular area.

The FlexPDE command line_integral explicitly integrates along a curve, given by name. We may also use surf_integral to indicate that the integration is to be taken over a surface.

Mathematically, we wish to integrate a function $f(\rho)$ over a circular cross-section, i.e.

$$\iint f(\rho)\,\rho\,d\varphi\,d\rho = \int_0^R \rho\,f(\rho)\,d\rho \int_0^{2\pi} d\varphi = 2\pi \int_0^R \rho\,f(\rho)\,d\rho$$

Thus we may perform this integration in two ways, either explicitly by line_integral according to the above expression, or implicitly by means of a surface integral (surf_integral). For the kinetic energy we use both ways for comparison. The table below shows the results.

```
Constricted Tube                                    08:00:15 7/26/05
                                                     FlexPDE 5.0.3ys

 SUMMARY

 K1l=  1569.260
 K1=  1569.260
 K0=  25131.08
 W1=  314051.9
 W0=  290592.0
 W1- W0=  23459.84
 K0- K1=  23561.82
```

The first two results are identical, as expected. The last two lines demonstrate that the work agrees with the change in kinetic energy within about 0.5%.

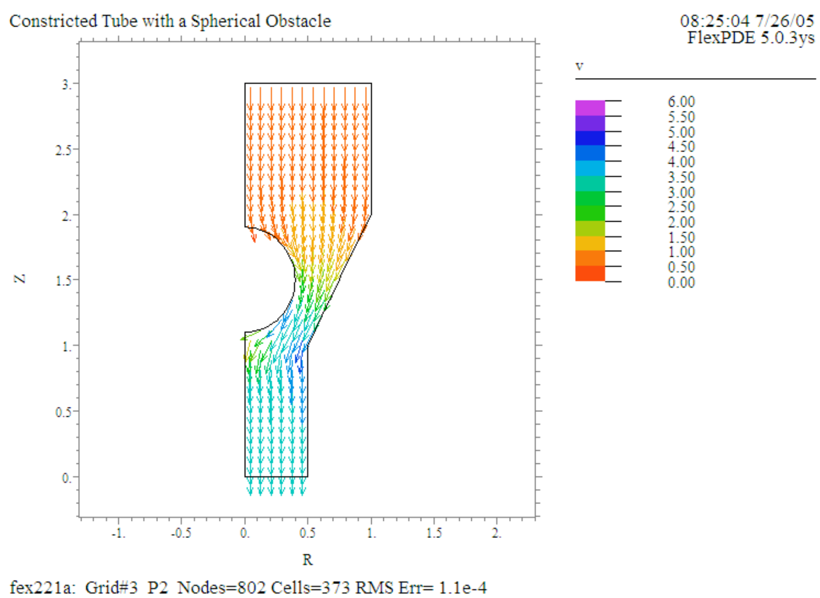
Constricted Tube with a Spherical Obstacle

We next proceed to a variation of the above example, where a ball on the symmetry axis partly blocks the flow. The changes with respect to the *fex221* descriptor mainly concern the *domain* section. In this projection, we define the ball by indenting the domain by a half-circle on the axis of symmetry.

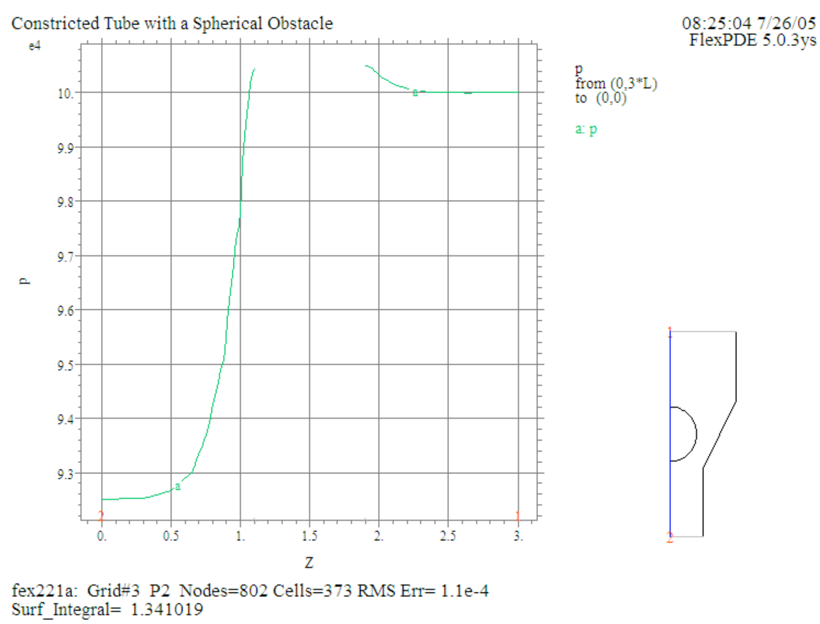
```
TITLE     'Constricted Tube with a Spherical Obstacle'     { fex221a.pde }
…
DEFINITIONS
  r00=0.4     r0=0.5     r1=1.0     L=1.0
…
region 'domain'
  start 'outer' (r1,3*L)   natural( phi)= -vz1   line to (0,3*L)          { In }
  natural( phi)=0   line to (0,1.5*L+r00)   natural( phi)=0
```

```
arc( center=0,1.5*L) angle=-180  natural( phi)=0   line to (0,0)
value( phi)=0   line to (r0,0)                                   { Out }
natural( phi)=0   line to (r0,L) to (r1,2*L) to close
…
```

The following vector plot illustrates the geometry of the tube and the spherical obstacle.



The elevation plot below exhibits a gap over the region of the ball, where no data are available. On the top and bottom sides of the obstacle we notice the effect of *stagnation*, leading to excess pressure.



It may be surprising to discover that the gain in kinetic energy per unit time from input to output is closely the same as without the obstacle.

Exercises

- ❑ Modify *fex221* by introducing a sudden constriction at half-height.
- ❑ Expand *fex221* to apply to a symmetrical Venturi tube.
- ❑ Calculate the total kinetic energy of the liquid in *fex221*.
- ❑ Solve for the velocity field around a short, solid cylinder in a cylindrical tube.

23 Viscous Flow in (ρ,z) Space

There are many axially symmetric problems that FlexPDE can solve in cylindrical coordinates. We just need to transform the PDE and the boundary conditions accordingly.

We start with the N-S equation (p.252)

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} + \rho_0 (\mathbf{v} \cdot \nabla)\mathbf{v} - \mathbf{F} + \nabla p - \eta \nabla^2 \mathbf{v} = 0 \qquad \bullet$$

where we use ρ_0 to denote the density, in view of the possible confusion with the radial coordinate ρ. In the case of steady flow, the first term vanishes and we are left with

$$\rho_0 (\mathbf{v} \cdot \nabla)\mathbf{v} - \mathbf{F} + \nabla p - \eta \nabla^2 \mathbf{v} = 0 \qquad \bullet$$

The first term we leave unexpanded until it is needed in a later chapter. The last term looks simpler but is awkward to transform, because the *formal definition* of $\nabla^2 \mathbf{v}$ really is[6p36]

$$\nabla^2 \mathbf{v} = \nabla(\nabla \cdot \mathbf{v}) - \nabla \times (\nabla \times \mathbf{v})$$

which happens to take a very simple form in (x, y) space (p.253). The result of the expansion is[8p60]

$$\nabla^2 \mathbf{v} = \begin{Bmatrix} \dfrac{1}{\rho}\dfrac{\partial}{\partial \rho}\left(\rho \dfrac{\partial v_\rho}{\partial \rho}\right) - \dfrac{v_\rho}{\rho^2} + \dfrac{\partial^2 v_\rho}{\partial z^2} \\ \dfrac{1}{\rho}\dfrac{\partial}{\partial \rho}\left(\rho \dfrac{\partial v_z}{\partial \rho}\right) + \dfrac{\partial^2 v_z}{\partial z^2} \end{Bmatrix}$$

where we have combined the two first derivatives into one term, knowing that FlexPDE prefers this form.

Collecting the above terms we obtain the component PDEs in the form

$$\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v}-\begin{Bmatrix}F_\rho\\F_z\end{Bmatrix}+\begin{Bmatrix}\dfrac{\partial p}{\partial\rho}\\\dfrac{\partial p}{\partial z}\end{Bmatrix}-\eta\begin{Bmatrix}\dfrac{1}{\rho}\dfrac{\partial}{\partial\rho}\left(\rho\dfrac{\partial v_\rho}{\partial\rho}\right)-\dfrac{v_\rho}{\rho^2}+\dfrac{\partial^2 v_\rho}{\partial z^2}\\\dfrac{1}{\rho}\dfrac{\partial}{\partial\rho}\left(\rho\dfrac{\partial v_z}{\partial\rho}\right)+\dfrac{\partial^2 v_z}{\partial z^2}\end{Bmatrix}=0 \qquad \bullet$$

The *third* PDE, including the $\nabla\cdot\mathbf{v}$ term (p.254●2), is

$$\nabla^2 p+\rho_0\nabla\cdot\left[(\mathbf{v}\cdot\nabla)\mathbf{v}\right]-\nabla\cdot\mathbf{F}-C\frac{\eta}{L_0^2}\nabla\cdot\mathbf{v}=0 \qquad \bullet$$

For any vector **V** we have, in (ρ,z) space[6p82],

$$\nabla\cdot\mathbf{V}=\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho V_\rho\right)+\frac{\partial V_z}{\partial z}$$

Using this relation with $\mathbf{V}=\nabla p$ to expand $\nabla^2 p$ we obtain

$$\nabla^2 p=\nabla\cdot\nabla p=\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial p}{\partial\rho}\right)+\frac{\partial^2 p}{\partial z^2}$$

and with the above expression for the divergence we finally obtain

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial p}{\partial\rho}\right)+\frac{\partial^2 p}{\partial z^2}+\rho_0\nabla\cdot\left[(\mathbf{v}\cdot\nabla)\mathbf{v}\right]-\nabla\cdot\mathbf{F}-C\frac{\eta}{L_0^2}\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho v_\rho\right)+\frac{\partial v_z}{\partial z}\right)=0 \qquad \bullet$$

This is the 3rd PDE we need for the descriptor.

Boundary Conditions

The *value* boundary conditions in cylindrical coordinates are similar to those used before, but it remains to adapt the natural boundary condition for pressure to the case of axial symmetry. This is almost immediate, since we already have an expression for $\nabla^2\mathbf{v}$. From the form of the N-S equation in (ρ,z), we immediately find an expression for the natural boundary condition.

$$\frac{\partial p}{\partial n} \equiv \mathbf{n}\cdot\nabla p = \mathbf{n}\cdot\mathbf{F} + \eta\,\mathbf{n}\cdot\nabla^2\mathbf{v} - \rho_0\,\mathbf{n}\cdot\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right] =$$

$$n_\rho F_\rho + n_z F_z + \eta\left[n_\rho\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial v_\rho}{\partial\rho}\right) - \frac{v_\rho}{\rho^2} + \frac{\partial^2 v_\rho}{\partial z^2}\right) + n_z\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial v_z}{\partial\rho}\right) + \frac{\partial^2 v_z}{\partial z^2}\right)\right] - \rho_0\,\mathbf{n}\cdot\left[(\mathbf{v}\cdot\nabla)\,\mathbf{v}\right] \quad \bullet$$

Steady Flow at Small Speeds

In this chapter we shall be concerned with small Reynolds numbers ($\mathrm{Re} << 1$). This means that we neglect the terms proportional to the density ρ_0, which leaves us with the three PDEs

$$\begin{Bmatrix} \dfrac{\partial p}{\partial\rho} \\ \dfrac{\partial p}{\partial z} \end{Bmatrix} - \begin{Bmatrix} F_\rho \\ F_z \end{Bmatrix} - \eta\begin{Bmatrix} \dfrac{1}{\rho}\dfrac{\partial}{\partial\rho}\left(\rho\dfrac{\partial v_\rho}{\partial\rho}\right) - \dfrac{v_\rho}{\rho^2} + \dfrac{\partial^2 v_\rho}{\partial z^2} \\ \dfrac{1}{\rho}\dfrac{\partial}{\partial\rho}\left(\rho\dfrac{\partial v_z}{\partial\rho}\right) + \dfrac{\partial^2 v_z}{\partial z^2} \end{Bmatrix} = 0 \quad \bullet$$

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial p}{\partial\rho}\right) + \frac{\partial^2 p}{\partial z^2} - \nabla\cdot\mathbf{F} - C\frac{\eta}{L_0^2}\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho v_\rho\right) + \frac{\partial v_z}{\partial z}\right) = 0 \quad \bullet$$

The natural boundary condition for p also simplifies if we neglect the term in ρ_0, i.e.

$$\partial p/\partial n = n_\rho F_\rho + n_z F_z + \eta\left[n_\rho\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial v_\rho}{\partial\rho}\right) - \frac{v_\rho}{\rho^2} + \frac{\partial^2 v_\rho}{\partial z^2}\right) + n_z\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial v_z}{\partial\rho}\right) + \frac{\partial^2 v_z}{\partial z^2}\right)\right] \quad \bullet$$

Tube with Uniform Driving Pressure

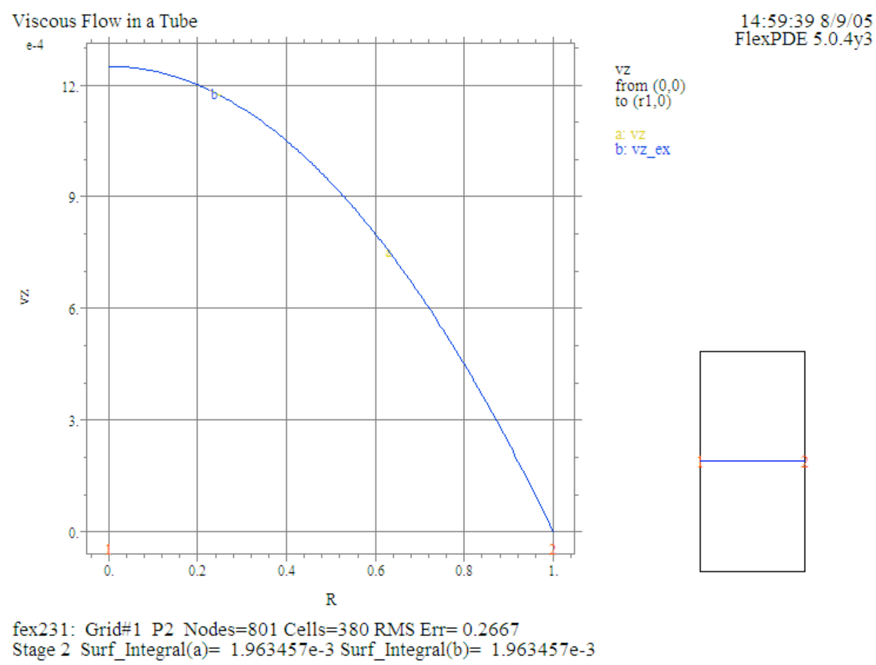
In accord with our usual policy we first apply the equations to the simplest possible case, that of laminar flow through a tube at Re<<1. The expression vz_ex is the exact, analytical solution[8p12] that we shall use for testing the numerical accuracy.

As in Chapter 21 (pp.277ff) we switch between natp=0 and the full expression for a comparison of results.

```
TITLE     'Viscous Flow in a Tube'                              { fex231.pde }
SELECT    errlim=1e-3    ngrid=1    spectral_colors    stages=2
COORDINATES    ycylinder('r','z')                          { Student Version }
VARIABLES    vr    vz    p
DEFINITIONS
  L=1.0    r1=1.0    delp=100    visc=1e4    dens=1e3
  v=vector( vr, vz)       vm=magnitude( v)
  Re=dens*globalmax(vm)*r1/visc
  div_v=1/r*dr(r*vr)+ dz(vz)    curl_phi=dz(vr)-dr(vz)
  vz_ex=delp/(2*L* 4*visc)*(r1^2-r^2)                       { Exact solution }
  natp=   if stage=2 then visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]  else 0
  force_v=-visc*dr(vz)    force_p=delp*pi*r1^2
EQUATIONS
  vr:          dr(p)- visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]=0
  vz:          dz(p)- visc*[ 1/r*dr(r*dr(vz))+ dzz(vz)]=0  { No gravity }
  p:           1/r*dr( r*dr(p))+ dzz(p)- 1e4*visc/L^2*div_v=0
BOUNDARIES
region 'domain' start 'outer' (0,-L)
  value(vr)=0   natural(vz)=0   value(p)=delp   line to (r1,-L)      { In }
  value(vr)=0   value(vz)=0  natural(p)=natp  line to (r1,L)         { Wall }
  value(vr)=0   natural(vz)=0   value(p)=0   line to (0,L)           { Out }
  value(vr)=0   natural(vz)=0   natural(p)=0   line to close         { Axis }
PLOTS
  contour( vz) report(Re)    contour( vr)    contour( p) painted
  elevation(vz, vz_ex) from (0,0) to (r1,0)
  elevation(vz-vz_ex) from (0,0) to (r1,0) report(globalmax( vm))
  vector( v) norm
  elevation( vz) from (0,-L) to (r1,-L)                           { Flux… }
  elevation( vz) from (0,0) to (r1,0)
  elevation( vz) from (0,L) to (r1,L)
  elevation( force_v) from (r1,-L) to (r1,L) report(force_p)
  contour( div_v)   contour( curl_phi) painted
END
```

On switching between the corresponding plots from stage=1 and stage=2 we find no difference of importance. Hence, we shall adhere to natp=0 for the remainder of this chapter, where Re<<1.

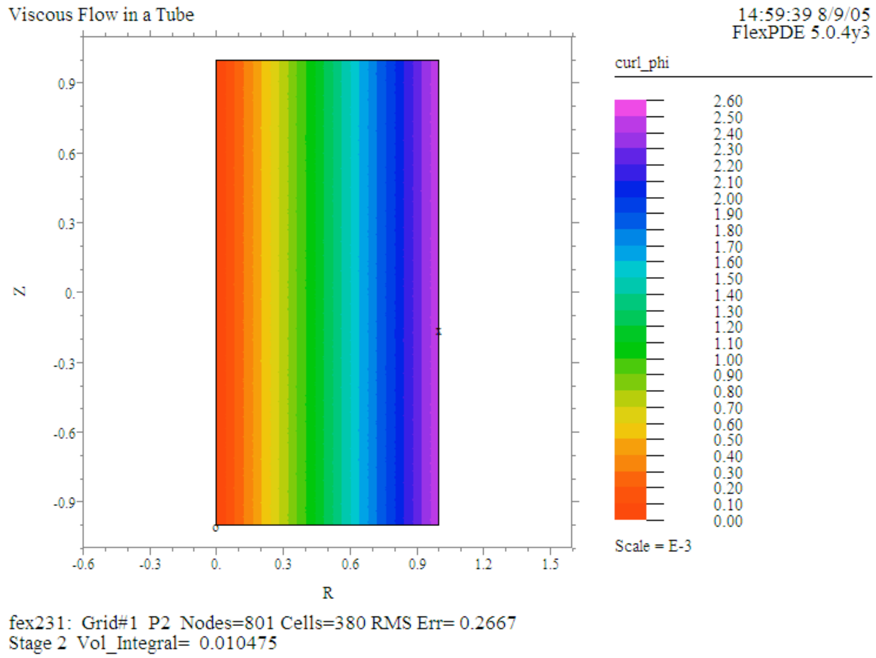
The plot below displays the parabolic velocity profile we have already seen in the case of a 2D channel, only in this case we are looking at a cross-section of axially symmetric flow. The curve for the exact solution over-writes that of vz, and the integral values indicate to what extent the functions are equal.



The elevation plot of the difference of the two solutions for vz shows that the FEA solution is good to within 1 part in 10^{14}.

The only vertical force acting on the liquid is viscous, and the force per unit area (force_v) takes a particularly simple form in this case. The last elevation plot automatically integrates that (constant) quantity over the surface of the tube, taking the factor 2*pi*r into account. The forces generated by the driving pressure (force_p) and by the viscous drag have accurately equal magnitudes, as shown by the reported value under the plot (not shown here).

The figure below shows that the vorticity, $\text{curl}_{\varphi}(\mathbf{v})$, is non-zero over the entire domain, except on the axis of symmetry. Evidently, laminar flow need not be irrotational.



Tube with Uniform Input Velocity

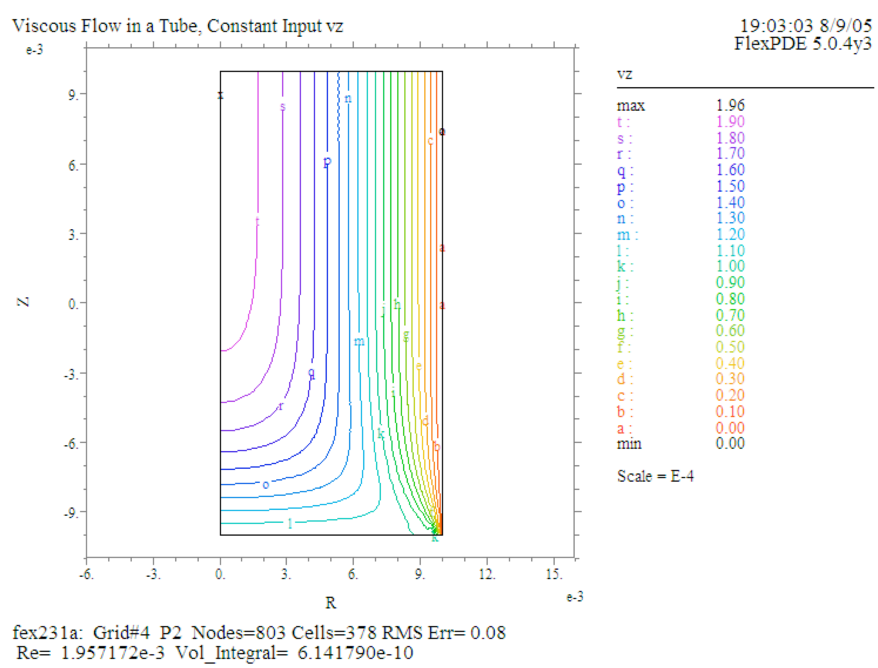
The preceding example was simple in the sense that it only involved parallel flow. We shall now engage more terms in the equations by imposing uniform vz0 at the input. Only a few modifications of *fex231* are necessary.

If we introduce uniform vz over the input, there will be a discontinuity at r=r1. The results show that FlexPDE is able to handle this problem. Based on experience, we simplify by putting natp=0 in this example and the following ones in this chapter.

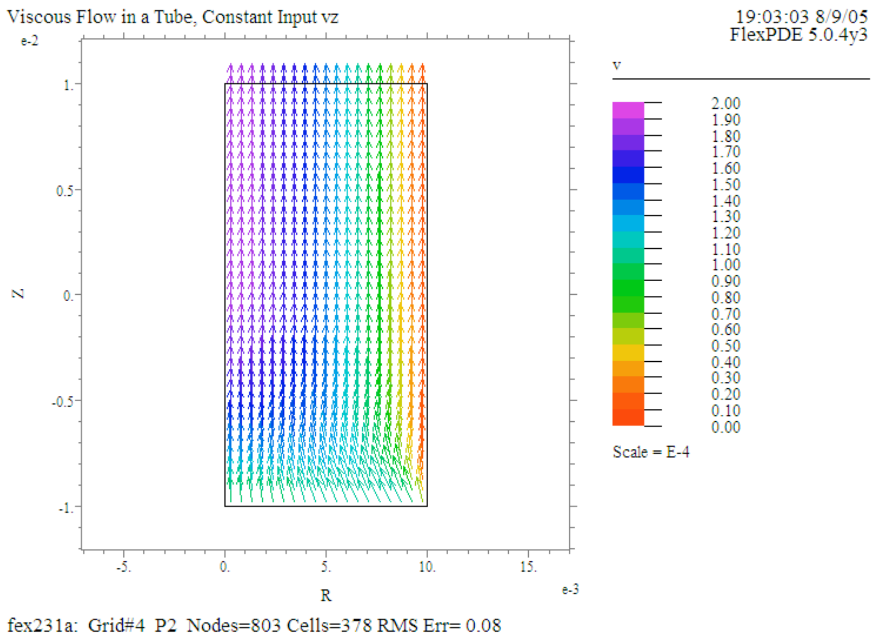
```
TITLE     'Viscous Flow in a Tube, Constant Input vz'      { fex231a.pde }
SELECT     errlim=1e-4    ngrid=2     spectral_colors
COORDINATES     ycylinder('r','z')
VARIABLES     vr     vz     p
DEFINITIONS
  L=1.0e-2     r1=1.0e-2
  vz0=1e-4     visc=1.0     dens=1e3
  v=vector( vr, vz)        vm=magnitude( v)
  Re=dens* globalmax(vm)*r1/ visc
  div_v=1/r*dr(r*vr)+ dz(vz)     curl_phi=dz(vr)- dr(vz)
  unit_r=vector(1,0)     unit_z=vector(0,1)
  nr=normal( unit_r)     nz=normal( unit_z)
```

```
  natp=0                                              { Simplification }
EQUATIONS
  vr:       dr(p)- visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]=0
  vz:       dz(p)- visc*[ 1/r*dr(r*dr(vz))+ dzz(vz)]=0
  p:        1/r*dr( r*dr(p))+ dzz(p)- 1e4*visc/L^2* div_v=0
BOUNDARIES
region 'domain' start 'outer' (0,-L)
  natural(vr)=0  value(vz)=vz0  natural(p)=natp line to (r1,-L)   { In }
  value(vr)=0  value(vz)=0  natural(p)=natp  line to (r1,L)       { Wall }
  value(vr)=0  natural(vz)=0  value(p)=0  line to (0,L)           { Out }
  value(vr)=0  natural(vz)=0  natural(p)=0  line to close         { Axis }
MONITORS
  contour( vz)   elevation( vz) from (0,-L) to (r1,-L)
PLOTS
  contour( vz) report( Re)   contour( vr)   contour( p) painted
  vector( v) norm
  contour( div_v)  contour( curl_phi) painted
  elevation( vz) from (0,-L) to (r1,-L)                    { Fluxes }
  elevation( vz) from (0,0) to (r1,0)
  elevation( vz, -5e-3*dr( vz)) from (0,L) to (r1,L)
END
```

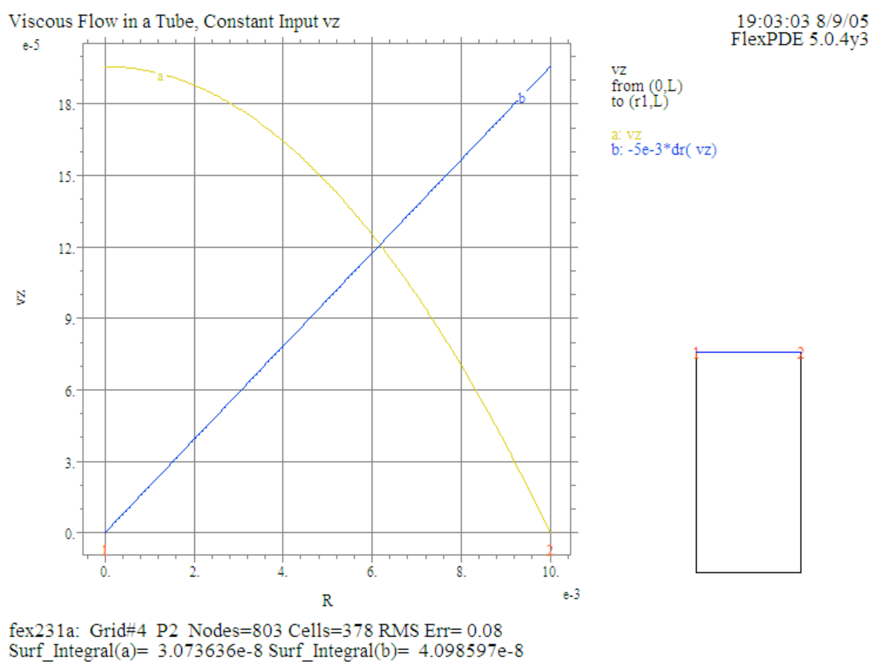
The first plot (below) shows how vz, which is constant at the entrance, gradually changes to a distribution that looks parabolic.



The following vector plot illustrates how the velocity finally takes a direction parallel to the z-axis. The plot of p shows that the pressure becomes uniform across the end.



The three elevation plots demonstrate the constancy of the flux along the tube. The last plot (below) indicates that the derivative of vz is linear, i.e. that the velocity profile has becomes parabolic.



Viscous Flow by Gravity through a Funnel

Earlier, we completely ignored the influence of the gravitational field on the flow, tacitly assuming the process to take place in a region of zero gravity. The reason for this choice was that it is more illuminating to consider one driving force at a time. We shall now study a case of liquid flow driven only by gravity.

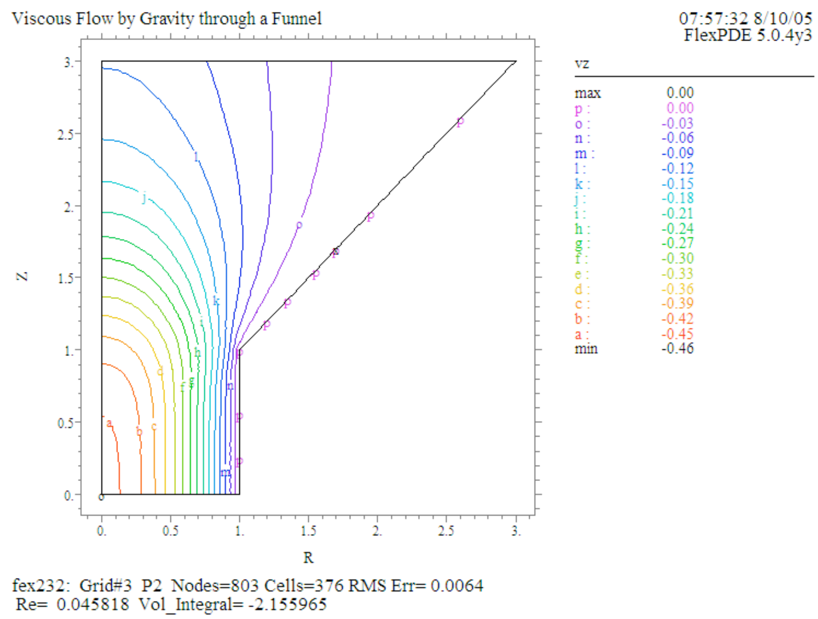
On p.297 we included a volume force **F** that could be used to take gravitation into account. Let us set the *z*-axis to be vertical, so that this force may be written as $F_z = -\rho_0 g$, the last factor being the gravitational acceleration. In the following descriptor, this new term occurs in the 2nd equation.

For the natural boundary conditions (natp) at the wall we apply the expression from p.299●1, where F_z now is non-zero.

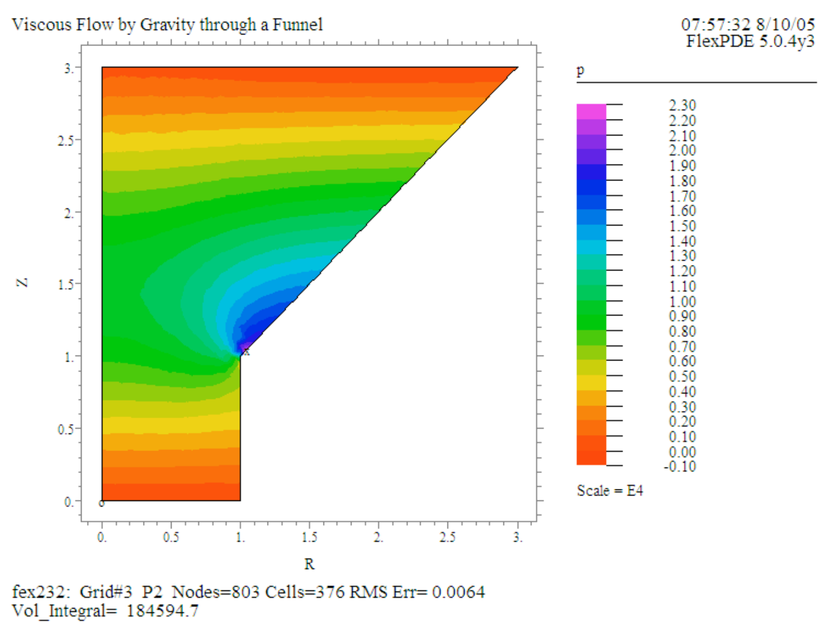
```
TITLE   'Viscous Flow by Gravity through a Funnel'          { fex232.pde }
SELECT     errlim=1e-3     ngrid=5     spectral_colors
COORDINATES     ycylinder('r','z')
VARIABLES     vr     vz     p
DEFINITIONS
  L=1.0     r1=1.0     g=9.81     dens=1e3
  visc=1e4     Fz=-dens*g
  v=vector( vr, vz)     vm=magnitude( v)
  Re=dens*globalmax(vm)*r1/ visc
  div_v=1/r*dr(r*vr)+ dz(vz)     curl_phi= dz(vr)-dr(vz)
  unit_r=vector(1,0)     unit_z=vector(0,1)
  nr=normal( unit_r)     nz=normal( unit_z)
  natp=nz*Fz                                              { Simplified }
EQUATIONS
  vr:          dr( p)- visc*[ 1/r*dr( r*dr(vr))- vr/r^2+ dzz( vr)]=0
  vz:          dz( p)- Fz- visc*[ 1/r*dr(r*dr(vz))+ dzz( vz)]=0   { Gravity }
  p:           1/r*dr( r*dr(p))+ dzz( p)- 1e4*visc/L^2*div_v=0
BOUNDARIES
region 'domain' start 'outer' (0,0)
  natural(vr)=0   natural(vz)=0   value(p)=0   line to (r1,0)          { Out }
  value(vr)=0   value(vz)=0   natural(p)=natp   line to (r1,L) to (3*r1,3*L)
  natural(vr)=0   natural(vz)=0   value(p)=0   line to (0,3*L)         { In }
  value(vr)=0   natural(vz)=0   natural(p)=0   line to close
PLOTS
  contour( vz) report( Re)     contour( vr)     contour( p) painted
  vector( v) norm     contour( curl_phi) painted
```

```
  elevation( vz) from (0,3*L) to (3*r1,3*L)
  elevation( vz, -0.5*dr(vz)) from (0,0) to (r1,0)        { Scale factor –0.5 }
END
```

The following contour plot of vz suggests that the velocity profile approaches the shape of a parabola near the exit.



The next plot (p) is evidently completely different from what we would expect from the flow produced by a pressure difference. The latter now vanishes.



The first elevation plot across the stream shows that the vertical velocity vz is far from parabolic, except near the axis. The elevation plot of vz over the exit does look parabolic, which we confirm by a curve of the radial derivative in the same figure. We multiply by the factor –0.5 in order to bring the derivative into the same plot frame.

Forces on the Funnel

We shall now carry the exploration of the preceding problem one step further by calculating the forces on the funnel and by comparing that to the driving force, which is the weight of the liquid within the domain.

The expression for the force per unit area on a solid (p.280) may be transformed directly to cylindrical coordinates as follows.

$$f_t = \eta\left\{\left(\frac{\partial v_\rho}{\partial \rho} n_\rho + \frac{\partial v_\rho}{\partial z} n_z\right) t_\rho + \left(\frac{\partial v_z}{\partial \rho} n_\rho + \frac{\partial v_z}{\partial z} n_z\right) t_z\right\} \qquad \bullet$$

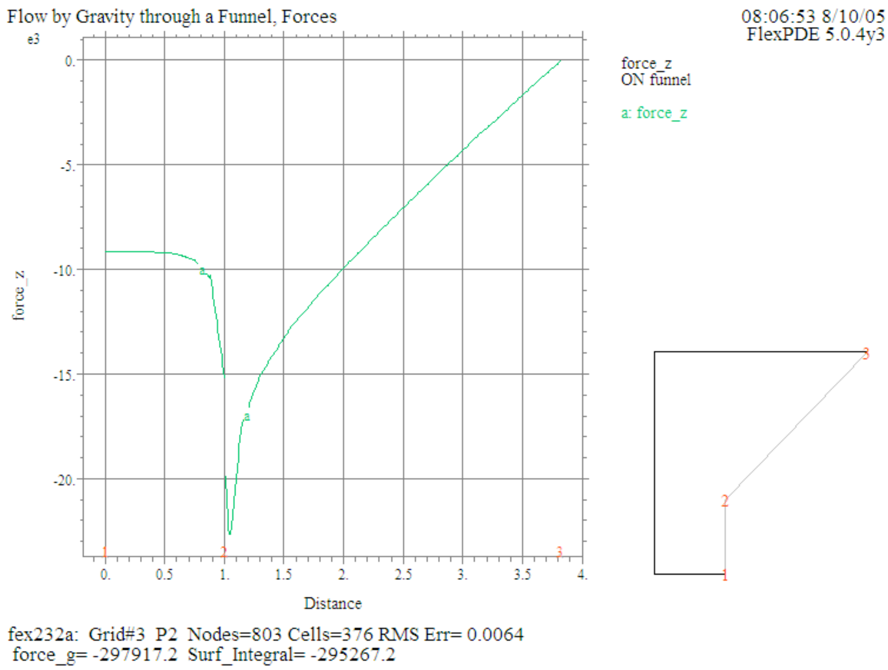
where **t** is the tangential unit vector. For the corresponding components we finally obtain

$$f_\rho = f_t\, t_\rho, \qquad f_z = f_t\, t_z \qquad \bullet$$

The following list shows how to modify *fex232*.

```
TITLE   'Flow by Gravity through a Funnel, Forces'        { fex232a.pde }
…
  tr=tangential( unit_r)     tz=tangential( unit_z)
  force_vt=-visc*[( dr( vr)*nr +dz( vr)*nz)*tr
     +( dr( vz)*nr+ dz( vz)*nz)*tz]
  force_vz=force_vt*tz                { Viscous force }
  force_pz=p*nz     force_z=force_vz+ force_pz
  force_g=vol_integral( -dens*g)
EQUATIONS
…
feature     start 'funnel' (r1,0) line to (r1,L) to (3*r1,3*L)
PLOTS
  elevation( force_z) on 'funnel'  report( force_g)
END
```

We find the drag force by integrating over the feature named 'funnel', indicated in the plot below. The total force of gravitation on the liquid we calculate by integrating ρg over the volume. The program automatically includes the volume element factor. According to the values on the bottom line of the plot the total forces agree within better than 1%.



Dissipation in the Funnel

As we have already discussed, viscous flow leads to the production of heat in the liquid. For use with cylindrical coordinates, we obtain the expression for the power of dissipation per unit volume (p.285) simply replacing x by ρ, and y by z.

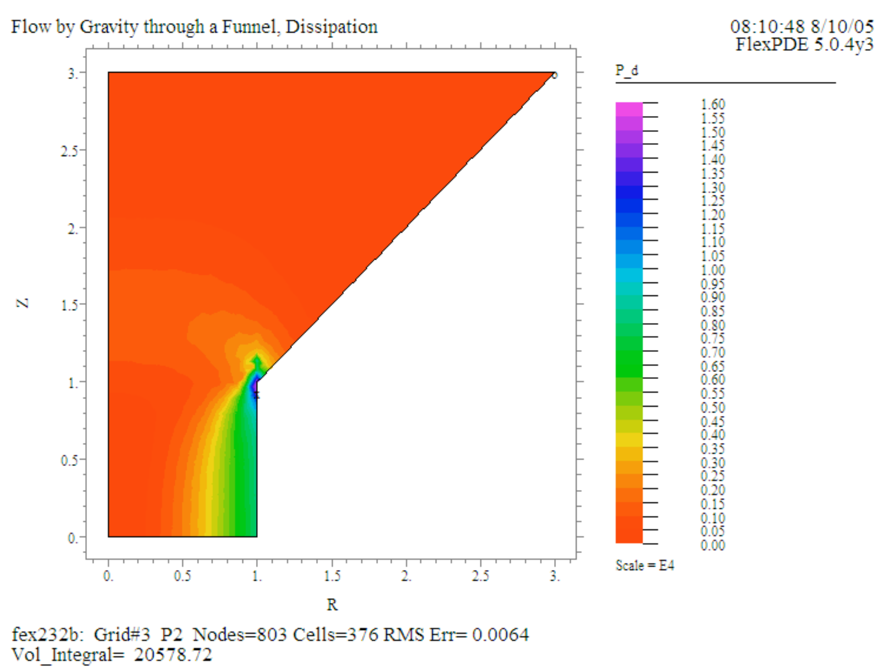
$$P_d = \eta \left\{ 2\left(\frac{\partial v_\rho}{\partial \rho}\right)^2 + \left(\frac{\partial v_\rho}{\partial z} + \frac{\partial v_z}{\partial \rho}\right)^2 + 2\left(\frac{\partial v_z}{\partial z}\right)^2 \right\} \qquad \bullet$$

This expression, integrated over the volume, yields the power dissipated.

The decrease in potential energy (per unit time), as the liquid flows downward in the gravitational field, must balance the increase in kinetic energy (p.293) plus dissipation. Under *definitions* we include expressions for these *power* terms. The following are the changes required with respect to *fex232*.

```
TITLE   'Flow by Gravity through a Funnel, Dissipation'    { fex232b.pde }
…
  K1=surf_integral( -vz*0.5*dens*vm^2, 'upper')
  K0=surf_integral( -vz*0.5*dens*vm^2, 'lower')
  P_d=visc*[ 2*dr(vr)^2+ (dz(vr)+dr(vz))^2+ 2*dz(vz)^2]
  P_diss=vol_integral( P_d)                       { Dissipation power }
  P_grav=vol_integral( -vz*dens*g)                { Gravitational power }
EQUATIONS
…
feature
  start 'upper' (0,3*L) line to (3*r1,3*L)        { Lines for integration… }
  start 'lower' (0,0) line to (r1,0)
PLOTS
  contour(P_d) painted
  summary
  report( K1) report( K0) report( P_diss)
  report( P_grav) report( K0-K1+P_diss)
END
```

The plot below shows that the dissipation occurs mainly close to the surface of the funnel, and there is a sharp maximum where the cone meets the tube.



The following summary lists th integral values.

```
Flow by Gravity through a Funnel, Dissipation                08:10:48 8/10/05
                                                              FlexPDE 5.0.4y3

SUMMARY

K1= 1.871311
K0= 37.76110
P_diss= 20437.36
P_grav= 21178.68
K0-K1+P_diss= 20473.24
```

The kinetic power terms, K1 and K0, are small in comparison with the others. The last two values, which should balance, evidently do so within about 3%.

Viscous Flow past a Sphere

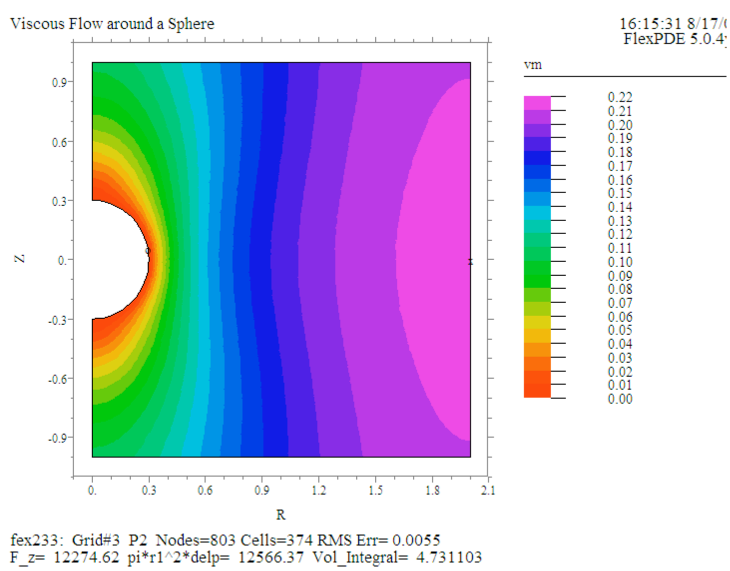
We shall now study viscous flow around a spherical obstacle in a tube with slip boundary conditions on the wall. Specifically, we shall calculate the drag force on the sphere for later comparison to the result of an analytic solution. The force terms on p.280 contain only first-order derivatives, and we may simply replace the coordinates (x, y) with (ρ, z) in the expression for f_t. The signs of the components also have to be watched.

The following descriptor is similar to *fex232a* in several respects, the essential differences being in the boundaries segment. We remove the projection of the sphere from the domain by an indentation.

```
TITLE     'Viscous Flow around a Sphere'                    { fex233.pde }
SELECT     spectral_colors
COORDINATES     ycylinder('r','z')
VARIABLES     vr     vz     p
DEFINITIONS
  L=1.0     r1=2.0     r0=0.3
  delp=1e3     visc=1e4     dens=1e3
  v=vector( vr, vz)     vm=magnitude( v)
  Re=dens*globalmax(vm)*r1/ visc
  div_v=1/r*dr(r*vr)+ dz(vz)     curl_phi=dz(vr)- dr(vz)
  natp=0                                                  { Simplified }
  unit_r=vector(1,0)     unit_z=vector(0,1)
  nr=normal( unit_r)     nz=normal( unit_z)
  tr=tangential( unit_r)     tz=tangential( unit_z)
  force_vt=-visc*[ ( dr( vr)*nr +dz( vr)*nz)*tr
```

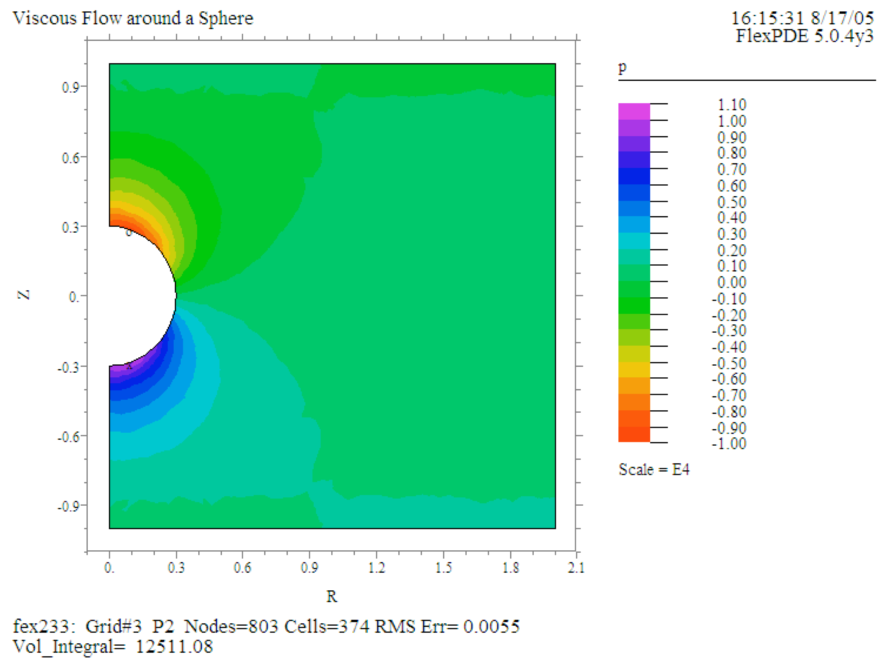
```
    +( dr( vz)*nr+ dz( vz)*nz)*tz]
  force_vz=force_vt*tz                    { Viscous force }
  force_pz=p*nz      force_z=force_vz+ force_pz
  F_z=surf_integral( force_z, 'sphere')                 { Total force }
EQUATIONS
  vr:          dr(p)- visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]=0
  vz:          dz(p)- visc*[ 1/r*dr(r*dr(vz))+ dzz(vz)]=0  { No gravitation }
  p:           1/r*dr( r*dr(p))+ dzz(p)- 1e4*visc/L^2*div_v=0
BOUNDARIES
region 'domain'   start 'outer' (0,-L)
  natural(vr)=0   natural(vz)=0   value(p)=delp   line to (r1,-L)      { In }
  value(vr)=0   natural(vz)=0   natural(p)=natp   line to (r1,L)       { Wall }
  natural(vr)=0   natural(vz)=0   value(p)=0   line to (0,L)           { Out }
  value(vr)=0   natural(vz)=0   natural(p)=0   line to (0,r0)          { Axis }
  value(vr)=0   value(vz)=0   natural(p)=natp                          { Ball }
    arc(center=0,0) angle=-180
  value(vr)=0   natural(vz)=0   natural(p)=0   line to close           { Axis }
feature     start 'sphere' (0,r0)  arc( center=0,0) angle=-180
MONITORS     contour( vm) painted report(F_z) report(pi*r1^2*delp)
PLOTS
  contour( vz) report(Re)
  contour( vm) painted report(F_z) report(pi*r1^2*delp)
  contour( p) painted     vector( v) norm
  contour( div_v)     contour( curl_phi) painted
END
```

The plot of vm below confirms that the speed vanishes on the surface of the sphere.



We have included a comparison with the force due to the driving pressure, which evidently is 2.4% higher than the value obtained by integration.

The following contour plot of p shows that most of the pressure variation occurs close to the sphere. This variation is an order of magnitude larger than the driving pressure applied between the ends.



Comparison with an Analytic Solution

There is a classical analytic solution by Stokes to the problem of a spherical obstacle in a stream of viscous liquid for Re<<1. The boundary conditions are different, however, in that the liquid is unbounded in space. We shall now consider that situation, in order to prepare for a detailed comparison with the FEA results.

The analytic solution[8p109] due to Stokes is available in *spherical* coordinates (R,θ) as follows.

$$v_{sR} = v_{z0}\cos(\theta)\left(1-\frac{3r_0}{2R}+\frac{r_0^3}{2R^3}\right) = v_{z0}\,\frac{z}{R}\left(1-\frac{3r_0}{2R}+\frac{r_0^3}{2R^3}\right)$$

$$v_{s\theta} = v_{z0}\sin(\theta)\left(-1+\frac{3r_0}{4R}+\frac{r_0^3}{4R^3}\right) = v_{z0}\frac{\rho}{R}\left(-1+\frac{3r_0}{4R}+\frac{r_0^3}{4R^3}\right)$$

$$p = p_0 - \eta\frac{3v_{z0}r_0}{2R^2}\cos(\theta) = p_0 - \eta\frac{3v_{z0}r_0}{2R^2}\frac{z}{R}$$

where θ is the angle to the symmetry axis (z) and r_0 the radius of the sphere. In the second member of each expression we have rewritten the trigonometric functions in terms of *cylindrical* coordinates (ρ, z).

Transforming the velocity components as well into cylindrical coordinates we have

$$v_\rho = v_{sR}\sin\theta + v_{s\theta}\cos\theta = v_{sR}\frac{\rho}{R} + v_{s\theta}\frac{z}{R}$$ ●

$$v_z = v_{sR}\cos\theta - v_{s\theta}\sin\theta = v_{sR}\frac{z}{R} - v_{s\theta}\frac{\rho}{R}$$ ●

and for the pressure (p_0 being the ambient pressure)

$$p = p_0 - \eta\frac{3v_{z0}r_0}{2R^2}\frac{z}{R}$$ ●

The *drag force* on the sphere is given by

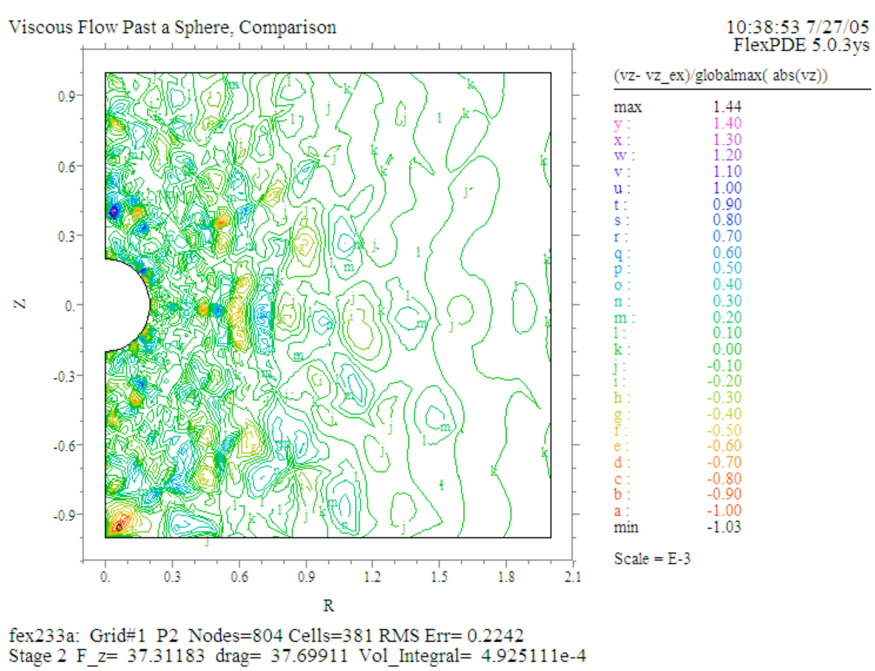
$$D = 6\pi\eta r_0 v_{z0}$$ ●

In order to compare the FEA results to this analytic solution we must adapt the boundary conditions. The exact solution assumes that the space for the liquid is unbounded, both radially and axially, and that the axial velocity at *infinite* distance is v_{z0}. In the preceding example, however, the liquid was conducted through a tube, the ball being on its axis.

For the comparison, let us introduce boundary conditions that are identical for the two solutions. To achieve this, we use the exact solution as the *value* boundary condition in the FEA descriptor. With this strategy, the complication of infinite space does not arise. We need to modify *fex233* as follows.

```
TITLE     'Viscous Flow Past a Sphere, Comparison'       { fex233a.pde }
…
  vz0=1e-3         drag=6*pi*visc*r0*vz0                     { Due to Stokes }
  rad=sqrt( r^2+ z^2)                                        { rad=R }
  vsr=vz0*z/rad*( 1- 3*r0/2/rad+ r0^3/2/rad^3)
  vst=vz0*r/rad*( -1+ 3*r0/4/rad+ r0^3/4/rad^3)
  vr_ex=vsr*r/rad+ vst*z/rad     vz_ex=vsr*z/rad- vst*r/rad
  p_ex=-visc*3/2*vz0*r0/rad^2*z/rad
EQUATIONS
…
region 'domain'   start 'outer' (0,-L)
  value(vr)=vr_ex   value(vz)=vz_ex   value(p)=p_ex
  line to (r1,-L) to (r1,L) to (0,L) to (0,r0)
  arc(center=0,0) angle=-180 line to close
feature
  start 'sphere' (0,-r0) arc( center=0,0) angle=180
PLOTS
  contour( vz) report( Re)
  contour( (vr- vr_ex)/globalmax( abs(vr)))
  contour( (vz- vz_ex)/globalmax( abs(vz))) report(F_z) report(drag)
  contour( (p- p_ex)/globalmax( abs(p)))
END
```

In this descriptor we plot the *relative* deviation from the Stokes solutions, using the globalmax command.



The above plot shows that the maximum relative error in the case of vz is about 0.1%, but the largest values occur in a small region of the domain. The agreement is roughly 1% for vr and p.

The drag force obtained by integration (F_z) agrees within about 1% with that given by Stokes. The Professional Version will yield much better agreement.

Exercises

❑ Show analytically that the solution vz_ex in *fex231* (with vr=0 and a linear function for pressure) satisfies the PDEs and the boundary conditions.

❑ Run *fex231* again with the parameters L=r1=1e-3, delp=1.0, and visc=1.0.

❑ Modify *fex232a* to study the flow through a straight tube under the influence of gravity only. Restore the plots from *fex232*.

❑ Change *fex233* and *fex233a* to zero velocity conditions at the tube wall.

24 Seeping through Porous Materials

The resultant flow through a porous solid, such as sand or soil, may be modeled as distributed leakage through narrow, meandering channels. In most practical cases, the thickness of these channels would be small enough to ensure Re<<1, even for liquids of modest viscosity, such as water. This mode of flow is known as *seeping* or *percolation*.

Percolation in (*x,y*) Space

As we have seen in the beginning of the preceding chapter, the average speed in a tube is proportional to the pressure difference. If we generalize to flow through a porous solid, we could write[9p223]

$$\mathbf{v} = -\frac{k}{\eta}\nabla p \qquad \bullet$$

where k is the permeability to flow, which we assume to be constant in space.

For this velocity field, we obtain

$$\nabla \times \mathbf{v} = -\frac{k}{\eta}\nabla \times \nabla p = 0$$

which vanishes since the mixed derivatives in this expression cancel. Seeping flow through a porous material under these conditions is thus irrotational. If we assume that there is no source term, the divergence $(\nabla \cdot \mathbf{v})$ must also vanish.

This means that we can use the auxiliary function ϕ to express the velocity components, as we did on p.226.

$$v_x = \frac{\partial\phi}{\partial x}, \quad v_y = \frac{\partial\phi}{\partial y} \qquad \bullet$$

$$\frac{\partial^2\phi}{\partial^2 x}+\frac{\partial^2\phi}{\partial^2 y}=0 \qquad \bullet$$

In a practical calculation, we would want to specify boundary conditions in terms of pressure. To obtain a second PDE involving p we may take the divergence of p.316●1, assuming constant η/k.

$$\nabla^2 p = -\frac{\eta}{k}\nabla\cdot\mathbf{v}=0 \qquad \bullet$$

We shall now apply these equations to the seeping of water through a block of concrete. The descriptor below refers to a cross-section of the block, which extends far in the directions of $\pm z$. The two faces on the top and to the right are water-tight. The left face is also watertight except over the middle third, while the bottom is open.

We furthermore let the left face be exposed to water from a big reservoir, so that the pressure over the permeable part of the face is constant at delp+p0, thus higher than the ambient pressure p0 at the bottom.

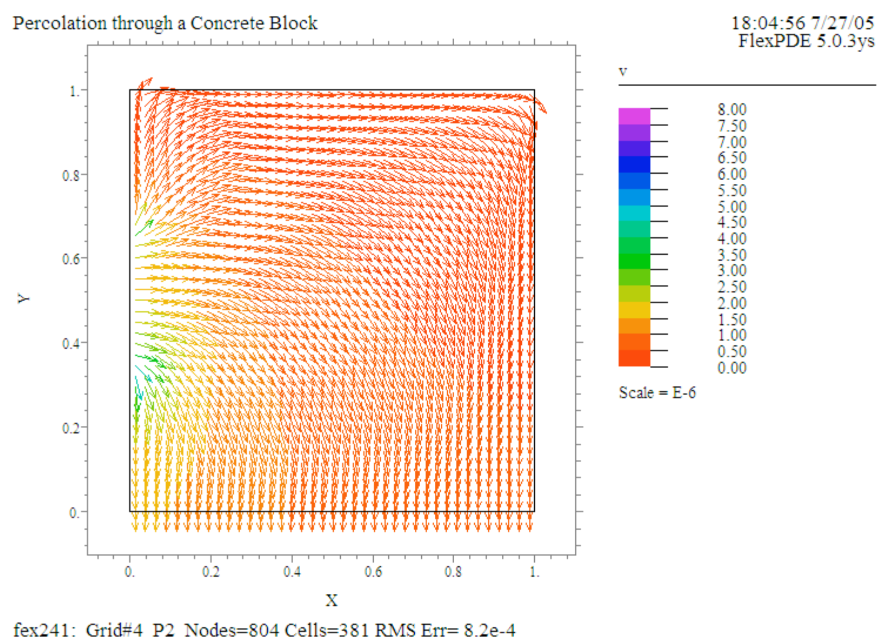
On the right, impermeable boundary we specify natural(phi)=0, which ensures $v_x = 0$ according to p.316●2. Similar considerations apply to the other walls. Over the seeping window, we use p.316●1 to specify a non-zero value for v_x, remembering that the normal is opposite to the direction of the x-axis.

As regards the pressure, we can only assume that it varies little close to the impermeable walls, hence that natural(p)=0.

```
TITLE   'Percolation through a Concrete Block'             { fex241.pde }
SELECT     errlim=1e-4     ngrid=1     spectral_colors
VARIABLES     phi     p                                   { Student Version }
DEFINITIONS
  L=1.0     visc=1e-3     k=1e-12
  p0=1e5     delp=1e3
  vx=dx( phi)     vy=dy( phi)     v=vector( vx, vy)     vm=magnitude( v)
EQUATIONS
  phi:        div( grad( phi))=0
  p:          div( grad( p))=0
BOUNDARIES
region 'domain' start 'outer' (0,0)
  value( phi)=0   value( p)=p0  line to (L,0)              { Bottom }
  natural( phi)=0   natural( p)=0   line to (L,L) to (0,L) to (0,2/3*L)
```

```
  natural( phi)=k/visc*dx( p)   value( p)=delp+p0     { Seeping window }
  line to (0,L/3)
  natural( phi)=0   natural(p)=0 line to close
PLOTS
  contour( phi)     contour( p) painted     contour( vm) painted
  vector( v) norm     contour(div( v))     contour( curl( v))
  elevation( p) on 'outer'
END
```

The figure below shows the velocity field. Clearly, the boundary conditions are satisfied.



Although this plot looks plausible, we did not include the force of gravitation, which is of appreciable magnitude in this problem. In fact, there is no obvious way of including any volume force in these equations.

Percolation in (x,y) by Navier-Stokes PDE

The simplest model of percolation does not take gravity into account. We may include this volume force by using the Navier-Stokes equation (p.252) in a novel manner. For small values of Re, the N-S equation takes the form (p.256●2)

$$-\begin{Bmatrix} F_x \\ F_y \end{Bmatrix} + \begin{Bmatrix} \dfrac{\partial p}{\partial x} \\ \dfrac{\partial p}{\partial y} \end{Bmatrix} - \eta \begin{Bmatrix} \dfrac{\partial^2 v_x}{\partial x^2} + \dfrac{\partial^2 v_x}{\partial y^2} \\ \dfrac{\partial^2 v_y}{\partial x^2} + \dfrac{\partial^2 v_y}{\partial y^2} \end{Bmatrix} = 0$$

In other applications in this chapter, the wall or the obstacle provided friction, which retarded the flow. In the case of a porous substance, the friction is present all over the volume, and the additional effect of a wall is of minor importance. Thus we apply *slip* boundary conditions on the walls.

The viscosity η occurs only in the 3rd term, which we may now replace by the percolation force. From p.316●1 we obtain $\nabla p = -(\eta/k)\mathbf{v}$ as the expression for the viscous volume force. With the gravity force included, the Navier-Stokes PDE for percolation (small Re) thus simplifies into

$$\begin{Bmatrix} \dfrac{\partial p}{\partial x} \\ \dfrac{\partial p}{\partial y} \end{Bmatrix} - \begin{Bmatrix} 0 \\ F_{gy} \end{Bmatrix} + \frac{\eta}{k} \begin{Bmatrix} v_x \\ v_y \end{Bmatrix} = 0$$ ●

The 3rd PDE (p.254●2) becomes

$$\nabla^2 p - \nabla \cdot \mathbf{F} - f_\nabla \nabla \cdot \mathbf{v} = 0$$ ●

The expression for natp (p.256●1) using the new PDE is

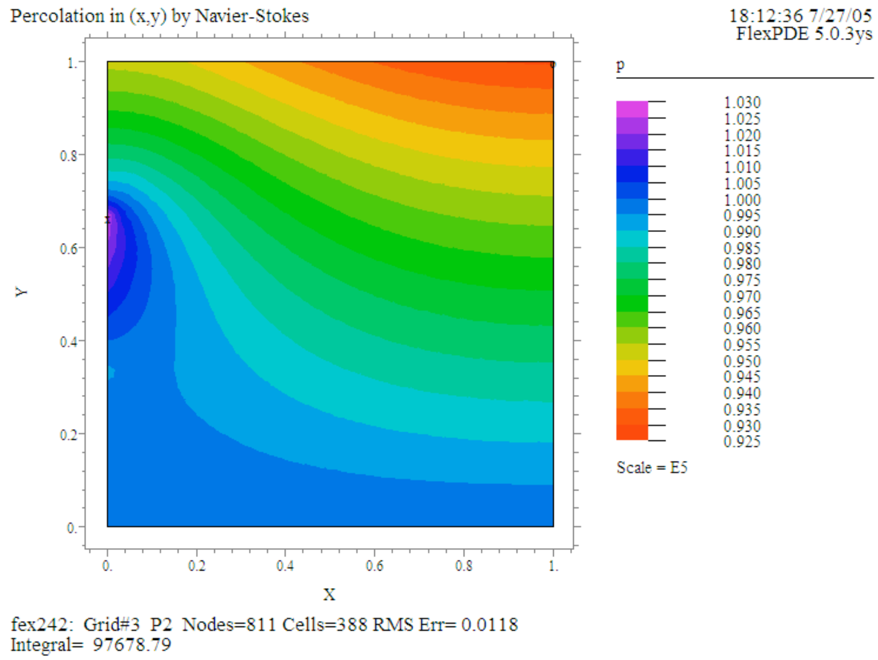
$$\partial p/\partial n = n_x F_x + n_y F_y - (\eta/k)\left(n_x v_x + n_y v_y\right)$$ ●

where we have replaced $\nabla^2 v_x$ by $(1/k)v_x$, and so on.

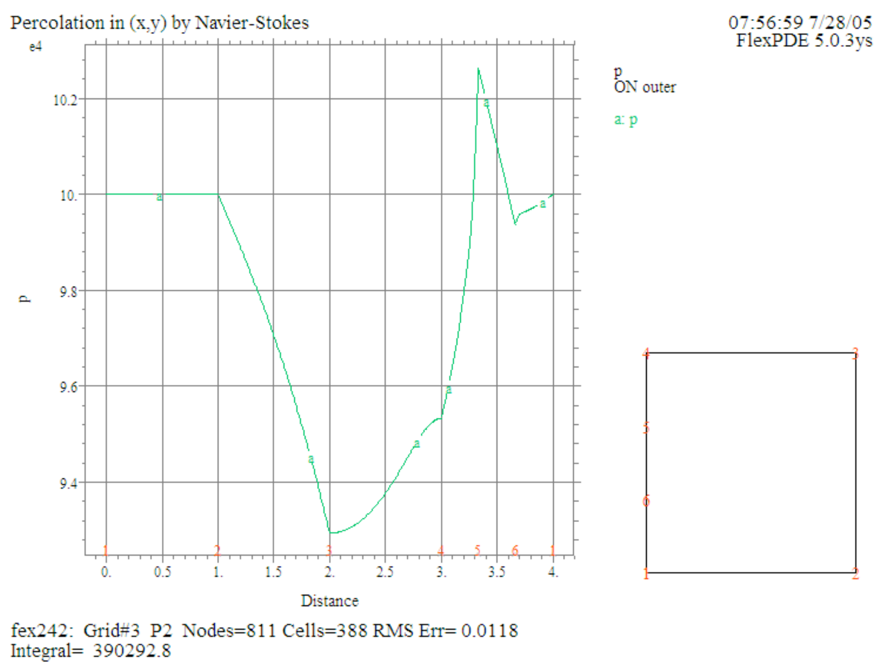
The descriptor based on these new principles is not much more complicated than before, as is evident from the following. We define the force per unit volume, Fgy, and use it in three lines. In the last of these, we arrange for the average pressure over the seeping window to be the same as in the preceding descriptor. Of course, gravitation makes it vary with *y*. Over the same window, we put vx equal to an expression derived in the same way as the natural condition used before. At the walls we apply natural boundary conditions by natp.

```
TITLE     'Percolation in (x,y) by Navier-Stokes'                 { fex242.pde }
SELECT     errlim=1e-4     ngrid=1     spectral_colors
VARIABLES     vx     vy     p
DEFINITIONS
  L=1.0     visc=1e-3     k=1e-12     dens=1e3
  p0=1e5     delp=1e3     Fgy=-dens*9.81               { Gravity }
  v=vector( vx, vy)     vm=magnitude(v)
  unit_x=vector(1,0)     unit_y=vector(0,1)
  nx=normal( unit_x)     ny=normal( unit_y)
  natp=nx*0+ ny*Fgy- visc/k*( nx*vx+ ny*vy)
EQUATIONS
  vx:          dx( p)+ visc/k*vx=0
  vy:          dy( p)- Fgy+ visc/k*vy=0
  p:           div( grad( p))- 1e4*visc/L^2*div( v)=0
BOUNDARIES
region 'domain' start 'outer' (0,0)
  natural( vx)=0   natural( vy)=0    value( p)=p0  line to (L,0)  { Bottom }
  value( vx)=0   natural( vy)=0   natural( p)=natp   line to (L,L)    { Slip: }
  natural( vx)=0   value( vy)=0   natural( p)=natp  line to (0,L)
  value( vx)=0   natural( vy)=0   natural( p)=natp   line to (0,2/3*L)
  value( vx)=-k/visc*dx( p)   value( vy)=0
    value( p)=p0+delp+(L/2-y)*Fgy   line to (0,L/3)  { Seeping window }
  value( vx)=0   natural( vy)=0 natural(p)=natp line to close
PLOTS
  contour( vx)     contour( vy)
  vector( v) norm      contour( vm) painted
  contour( p) painted     elevation( p) on 'outer'
  contour( div( v))     contour( curl( v)) painted
  elevation( vx, vy) on 'outer'                          { Verification of BCs }
END
```

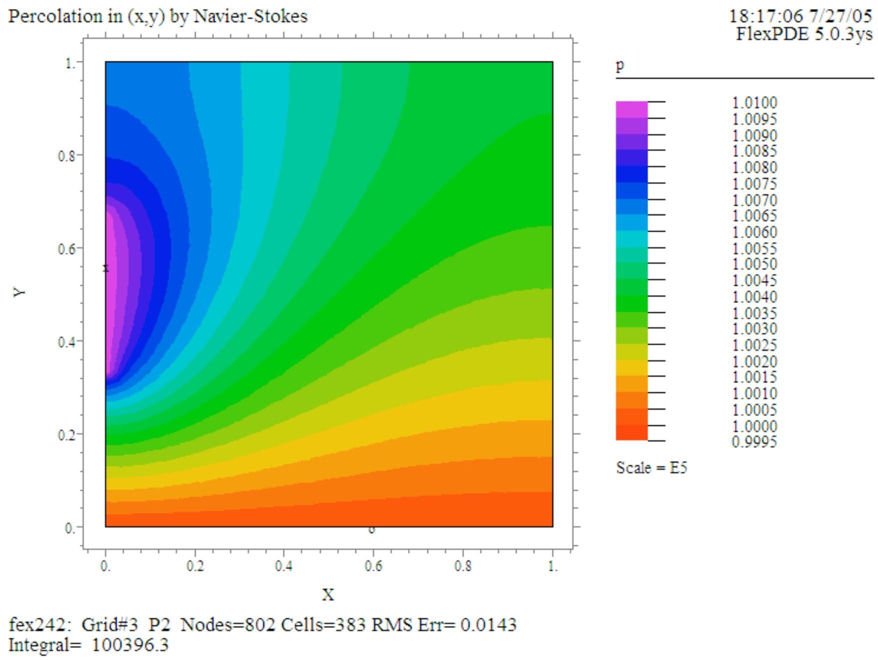
The vector plot of the velocity (not shown here) is rather similar to what we have just seen. The following contour plot of the pressure, however, exhibits some new features. The maximum value occurs at the upper edge of the seeping window, and we find the lowest value in the upper-right corner. The range of variation of *p* is nearly ten times as large as before.



The elevation plot of p below shows the difference even more clearly. Here, the input pressure appears as a ramp between the points 5 and 6. The steep decrease between points 2 and 3 is an evidence for the gravity term.



Let us now make a direct *comparison* between the above descriptors by defining Fgy to be *zero* in *fex242*, giving the new file the name *fex242a*. The contour plot of p becomes as follows.



We find that the above plot of p is almost identical to what we saw when running *fex241*. The same is true for the corresponding elevation plots. This is remarkable in view of the difference in the PDEs as well as in the boundary conditions.

Percolation in (ρ,z) Space

The relations analogous to those on p.316 for cylindrical coordinates are

$$v_\rho = \frac{\partial \phi}{\partial \rho}\,, \qquad v_z = \frac{\partial \phi}{\partial z}, \qquad \text{or } \mathbf{v} = \nabla \phi \qquad \bullet$$

The assumption of vanishing divergence leads to

$$\nabla \cdot \mathbf{v} = \frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho v_\rho\right) + \frac{\partial v_z}{\partial z} = \frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho \frac{\partial \phi}{\partial \rho}\right) + \frac{\partial^2 \phi}{\partial z^2} = 0$$

which is the familiar Laplace equation in cylindrical coordinates (p.290●3).

In (ρ, z) the pressure equation thus becomes

$$\frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho\frac{\partial p}{\partial \rho}\right)+\frac{\partial^2 p}{\partial z^2}=0 \qquad \bullet$$

We shall now apply these equations to an arrangement consisting of a vertical tube, pushed into a pot containing concrete, with more concrete being loaded inside the tube. The tube is then topped off with water to provide driving pressure. The figure below illustrates the axially symmetric geometry of the porous solid.

We formally assume that the liquid seeping through to the exit is pumped back to the central tube, in order to keep levels unchanged. In this case there is no water source inside the material.

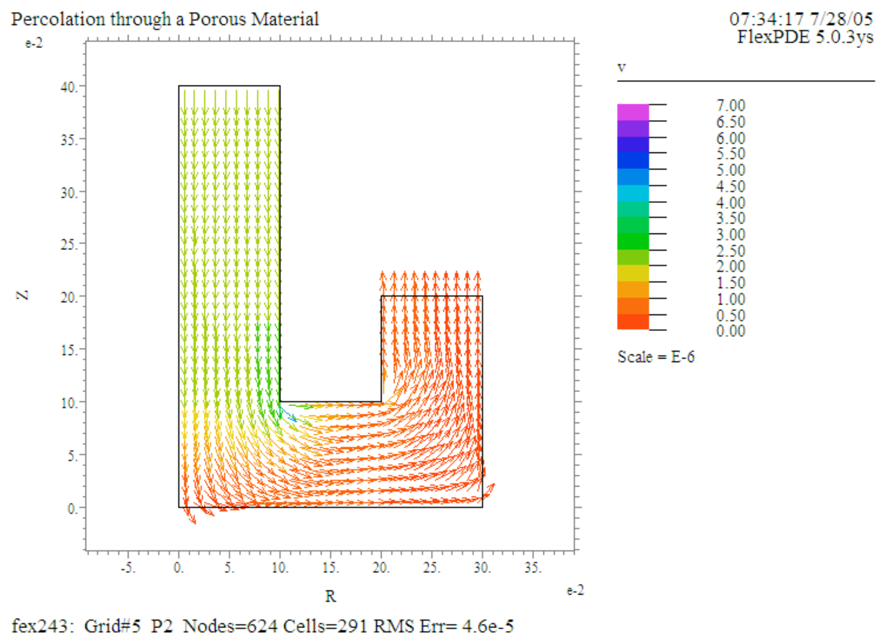
```
TITLE   'Percolation through a Porous Material'                { fex243.pde }
SELECT     errlim=1e-4     ngrid=1     spectral_colors
COORDINATES     ycylinder('r','z')
VARIABLES     phi     p
DEFINITIONS
  L=0.1     r1=0.1     r2=0.2     r3=0.3
  p0=1e5     delp=1e3     visc=1e-3     k=1e-12
  vr=dr(phi)     vz=dz(phi)
  v=vector( vr, vz)     vm=magnitude(v)
  div_v=1/r*dr(r*vr)+ dz(vz)     curl_phi=dz(vr)-dr(vz)
EQUATIONS
  phi:          1/r*dr( r*dr(phi))+ dzz(phi)=0
  p:            1/r*dr( r*dr(p))+ dzz(p)=0
BOUNDARIES
region 'domain' start 'outer' (r1,4*L)
  natural(phi)=-k/visc*dz(p)   value(p)=p0+delp   line to (0,4*L)  { In }
  natural(phi)=0   natural(p)=0   line to (0,0) to (r3,0)  to (r3,2*L)
  value(phi)=0   value(p)=p0   line to (r2,2*L)           { vr=0 }        { Out }
  natural(phi)=0   natural(p)=0   line to (r2,L) to (r1,L) to close
PLOTS
  contour( phi)     vector( v) norm     contour( vm) painted
  contour( p)     contour( div_v)     contour( curl_phi)
END
```

The boundary conditions are nearly obvious. The first one stems from two equations (pp.316, 322), which in (ρ, z) combine to yield

$$\frac{\partial \phi}{\partial z}=v_z=-\frac{k}{\eta}\frac{\partial p}{\partial z}$$

As regards the output side we should note that both the potential phi and the pressure only occur in derivatives and may be assigned an arbitrary value (zero).

The flow is driven by a slight overpressure on the central part, and the figure below shows how the speed decreases as the water approaches the free surface to the right. This is entirely a geometric effect, due to the increasing annular area at larger radius.



We notice from the plots that phi and p are proportional. The equations on p.316 and p.322 in fact give us

$$\nabla\phi + \frac{k}{\eta}\nabla p = \nabla\left(\phi + \frac{k}{\eta}p\right) = 0$$

which means that the expression in parentheses must be a constant, which could be defined to be zero.

Percolation in (ρ,z) by Navier-Stokes

We may also take gravity into account as we just did in (x, y). By analogy with p.319●1, the N-S equation takes the form (p.298●1)

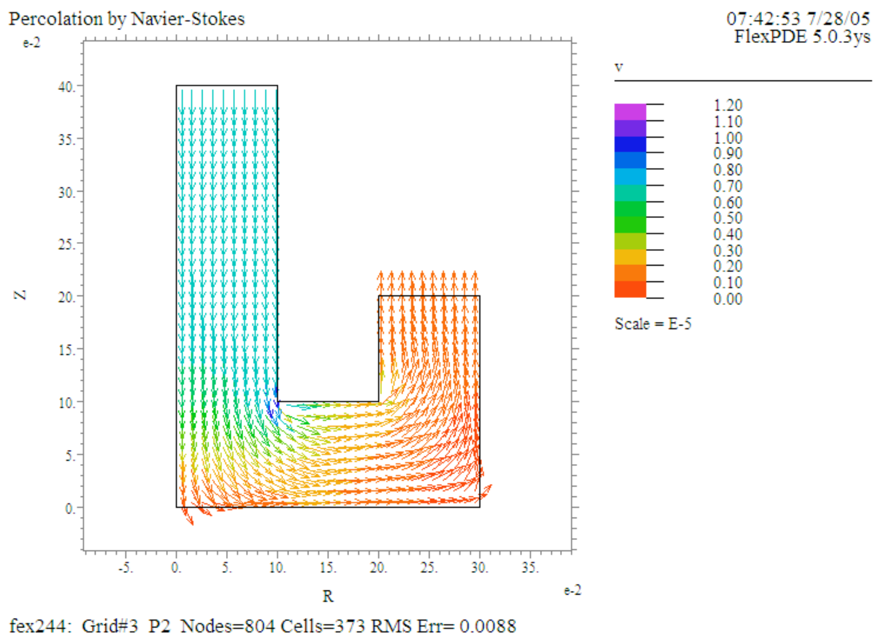
$$\begin{Bmatrix} \dfrac{\partial p}{\partial \rho} \\ \dfrac{\partial p}{\partial z} \end{Bmatrix} - \begin{Bmatrix} 0 \\ F_{gz} \end{Bmatrix} + \frac{\eta}{k} \begin{Bmatrix} v_\rho \\ v_z \end{Bmatrix} = 0 \qquad \bullet$$

The third PDE for pressure remains as in *fex241*. The expression for natp (p.319●3) must be revised, however, and by direct analogy we obtain

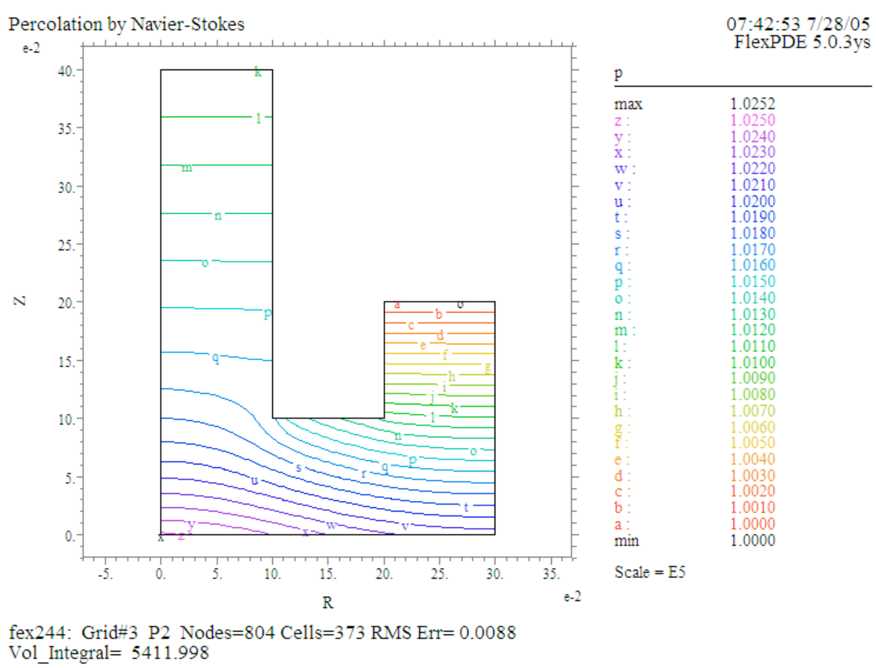
$$\partial p / \partial n = n_\rho F_\rho + n_z F_z - (\eta / k)(n_\rho v_\rho + n_z v_z) \qquad \bullet$$

```
TITLE     'Percolation by Navier-Stokes'                        { fex244.pde }
SELECT     errlim=1e-4     ngrid=1     spectral_colors
COORDINATES     ycylinder('r','z')
VARIABLES     vr     vz     p
DEFINITIONS
  L=0.1     r1=0.1     r2=0.2     r3=0.3     dens=1e3
  p0=1e5     delp=1e3     visc=1e-3     k=1e-12     Fgz=-dens*9.81
  v=vector( vr, vz)     vm=magnitude(v)
  div_v=1/r*dr(r*vr)+ dz(vz)     curl_phi=dz(vr)-dr(vz)
  unit_r=vector(1,0)     unit_z=vector(0,1)
  nr=normal( unit_r)     nz=normal( unit_z)
  natp=nz*Fgz- visc/k*( nr*vr+ nz*vz)
EQUATIONS
  vr:          dr(p)+ visc/k*vr=0
  vz:          dz(p)- Fgz+ visc/k*vz=0
  p:           1/r*dr( r*dr(p))+ dzz(p)- 1e4*visc/L^2*div_v=0
BOUNDARIES
region 'domain'  start 'outer' (r1,4*L)
  value(vr)=0   natural(vz)=0   value(p)=p0+delp   line to (0,4*L)   { In }
  value(vr)=0   natural(vz)=0   natural(p)=0   line to (0,0)    { Symmetry }
  natural(vr)=0   value(vz)=0   natural(p)=natp   line to (r3,0)
  value(vr)=0   natural(vz)=0   natural(p)=natp   line to (r3,2*L)
  value(vr)=0   natural(vz)=0   value(p)=p0   line to (r2,2*L)          { Out }
  value(vr)=0   natural(vz)=0   natural(p)=natp   line to (r2,L)
  natural(vr)=0   value(vz)=0   natural(p)=natp   line to (r1,L)
  value(vr)=0   natural(vz)=0   natural(p)=natp   line to close
PLOTS
  vector( v) norm     contour( vm) painted
  contour( p)     contour( div_v)     contour( curl_phi)
  elevation( vr, vz) on 'outer'
END
```

As seen from the following vector plot, the streamlines are much like those we just obtained by the simpler approach. The maximum speed is higher, however, no doubt due to gravity acting on the liquid in the central tube, above half-height.



The plot of p below is indeed different from before. The pressure maximum is now at the bottom of the pot.



We may again compare the two formalisms by putting Fgz=0 in the *definitions* section. The contour plots of pressure then become very similar. The vector plots of the velocity are also much the same as before. A sensitive test is to compare the plots of vm, where we note that the volume integral now is 4.789e-8 against 4.820e-8 in *fex243*. The agreement is remarkable, considering that the PDEs and the boundary conditions are different.

Exercises

❑ Modify *fex242* for a concrete block open for seeping over one-third of the top face, where the pressure is 1e3 over the ambient value. The rest of the top and the sides are impermeable while the bottom face is open. Repeat the calculation without gravity.

❑ Modify *fex244* to study the seeping flow through a porous material in a straight, vertical tube.

❑ Adapt *fex244* again to study seeping flow in a cylinder of concrete with the height equal to the diameter, measuring 1.0. Let a circular area, 0.5 in diameter at the bottom, be open to an overpressure of 1e4 while the rest of the bottom is impermeable. The cylindrical and top surfaces are assumed to be open to the liquid at ambient pressure. Compare to the velocity field in a gravity-free environment.

25 Viscous Flow at Re>>1 in (*x*,*y*)

This book mostly concerns *steady* flow, where the velocity field does not change with time. This of course requires that the boundary conditions be independent of time. In this type of flow, fluid particles seem to follow sheets as they travel through space and hence the flow is often referred to as *laminar*. At large values of Re, however, a transition to a turbulent state is known to occur, which is time-dependent and random. When we find that FEA procedures for steady flow do not converge we assume that time-dependent analysis would be required and that it would exhibit turbulent flow.

We have seen that the Navier-Stokes equation for steady flow ($\partial \mathbf{v}/\partial t = 0$) may be written (p.253●2)

$$\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v} - \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} + \begin{Bmatrix} \dfrac{\partial p}{\partial x} \\ \dfrac{\partial p}{\partial y} \end{Bmatrix} - \eta \begin{Bmatrix} \nabla^2 v_x \\ \nabla^2 v_y \end{Bmatrix} = 0$$

Since we can no longer neglect inertial terms proportional to ρ_0, we must transform the first term explicitly into derivatives as follows.

$$\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v} = \rho_0\left(v_x\frac{\partial}{\partial x} + v_y\frac{\partial}{\partial y}\right)\begin{Bmatrix} v_x \\ v_y \end{Bmatrix} = \rho_0\begin{Bmatrix} v_x\dfrac{\partial v_x}{\partial x} + v_y\dfrac{\partial v_x}{\partial y} \\ v_x\dfrac{\partial v_y}{\partial x} + v_y\dfrac{\partial v_y}{\partial y} \end{Bmatrix}$$

With this expression for the first term, the N-S vector equation reads

$$\rho_0\begin{Bmatrix} v_x\dfrac{\partial v_x}{\partial x} + v_y\dfrac{\partial v_x}{\partial y} \\ v_x\dfrac{\partial v_y}{\partial x} + v_y\dfrac{\partial v_y}{\partial y} \end{Bmatrix} - \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} + \begin{Bmatrix} \dfrac{\partial p}{\partial x} \\ \dfrac{\partial p}{\partial y} \end{Bmatrix} - \eta\begin{Bmatrix} \nabla^2 v_x \\ \nabla^2 v_y \end{Bmatrix} = 0 \qquad \bullet$$

We also need to include the second term in the *pressure* equation (p.254●2) as follows.

$$\nabla^2 p + \rho_0 \nabla \cdot \left[(\mathbf{v} \cdot \nabla) \mathbf{v} \right] - \nabla \cdot \mathbf{F} - C \frac{\eta}{L_0^2} \nabla \cdot \mathbf{v} = 0$$

We shall find that $C = 10^4$ is a suitable numeric value. Having already expanded the expression within square brackets, we easily recast the second term to obtain derivatives.

$$\nabla^2 p + \rho_0 \frac{\partial}{\partial x}\left(v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} \right) + \rho_0 \frac{\partial}{\partial y}\left(v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} \right) - \qquad \bullet$$

$$\nabla \cdot \mathbf{F} - C \frac{\eta}{L_0^2} \nabla \cdot \mathbf{v} = 0$$

The above expressions are *non-linear* in the dependent variables. For instance, the last equation involves v_x multiplied by its derivative. Analytic solutions are usually not available in such cases, which means that numerical calculation is the standard solution procedure.

We have thus obtained the three PDEs required, and it only remains to specify the natural *pressure boundary condition* in its complete form (p.256●1). The expression in the last term we have in fact already dealt with.

$$\partial p / \partial n = \mathbf{n} \cdot \mathbf{F} + \eta \mathbf{n} \cdot \nabla^2 \mathbf{v} - \rho_0 \mathbf{n} \cdot \left[(\mathbf{v} \cdot \nabla) \mathbf{v} \right] =$$

$$n_x F_x + n_y F_y + \eta \left[n_x \nabla^2 v_x + n_y \nabla^2 v_y \right] - \qquad \bullet$$

$$\rho_0 \left[n_x \left(v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} \right) + n_y \left(v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} \right) \right]$$

We notice that the expressions within parentheses on the last line occur several times in the PDEs, which we may utilize to simplify the descriptor.

Viscous Flow in a Channel

In this first example, we shall study the simple case of flow between two parallel walls. We utilize the stages feature, which permits us to chain several solutions with successive values of the input velocity. The first value of the input velocity vx0 will be 1e-6, and so on.

We extend the channel far toward the exit, so that the flow becomes reasonably parallel there.

By using the functions vxdvx, etc., which occur repeatedly, we make the expressions for natp and the PDEs somewhat shorter.

In order to shorten the run as far as possible we limit the number of nodes at 400. In addition, we tentatively put natp=0 and dens_term=0 in the 3rd PDE. The full expressions are second-order and are expected to grow strongly with Re. Hence, we verify the solution in the last stage by applying the full expressions.

In this and following examples we let FlexPDE decide about the error limit and the initial gridding.

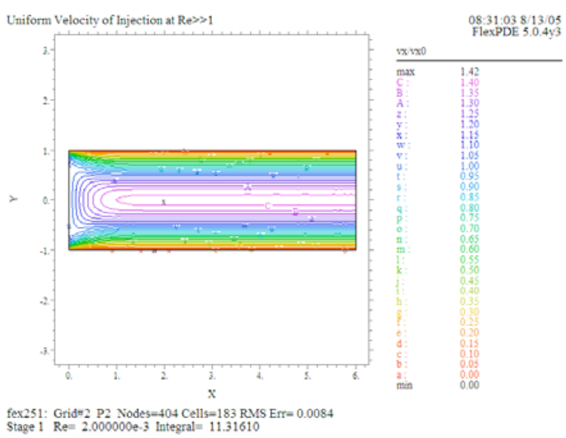
From this chapter onwards, the run times will generally be longer than before, and it might be wise to run the files over coffee breaks.

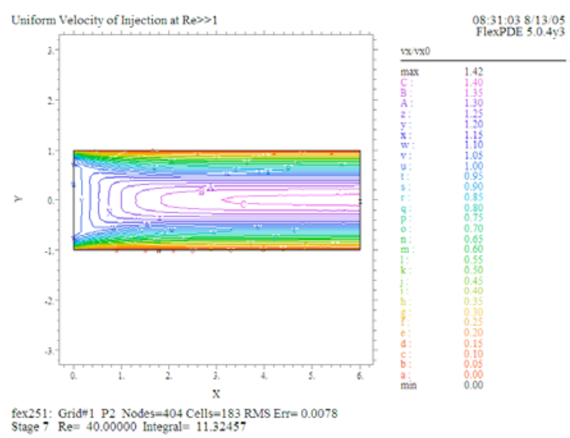
```
TITLE   'Uniform Velocity of Injection at Re>>1'              { fex251.pde }
SELECT    stages=7    nodelimit=400
   spectral_colors                                          { Student Edition }
VARIABLES     vx     vy     p                 { Pressure minus ambient }
DEFINITIONS
  Lx=6     Ly=1.0      visc=1.0        { Input velocities: }
  vx0=staged( 1e-6, 1e-4, 1e-3, 3e-3, 1e-2, 2e-2, 2e-2)
  dens=1e3     Re=dens*vx0*2*Ly/visc
  v=vector( vx, vy)     vm=magnitude( v)
  unit_x=vector(1,0)     unit_y=vector(0,1)
  nx=normal( unit_x)     ny=normal( unit_y)
  vxdvx=vx*dx( vx)+ vy*dy( vx)     vxdvy=vx*dx( vy)+ vy*dy( vy)
  natp=  if stage=7 then
            visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
           -dens*[ nx*vxdvx+  ny*vxdvy] else 0
  dens_term=  if stage=7 then dens*( dx( vxdvx)+ dy( vxdvy)) else 0
EQUATIONS
  vx:         dens*vxdvx+ dx( p)- visc*div( grad( vx))=0
  vy:         dens*vxdvy+ dy( p)- visc*div( grad( vy))=0
  p:          div( grad( p))+ dens_term- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES
region 'domain'  start 'outer' (0,Ly)
  value( vx)=vx0   natural( vy)=0  natural( p)=natp                { In }
  line to (0,-Ly)    value( vx)=0   value( vy)=0   natural( p)=natp   { Wall }
  line to (Lx,-Ly)  natural( vx)=0   value( vy)=0   value( p)=0       { Out }
```

```
  line  to (Lx,Ly)  value( vx)=0   value( vy)=0   natural( p)=natp   { Wall }
  line to close
PLOTS
  contour( vx/vx0) report( Re)
  vector( v) norm report(Re)     contour( vm) painted
  contour( p) painted     contour( div( v))     contour( curl( v)) painted
  elevation( vx) from (0,-Ly) to (0,Ly)
  elevation( vx) from (Lx/2,-Ly) to (Lx/2,Ly)
  elevation( vx) from (Lx,-Ly) to (Lx,Ly)
END
```

For comparison at successive speeds we plot vx/vx0, which should be the same in the regime of small Re. Any change of the initial pattern must be due to the non-linear term.

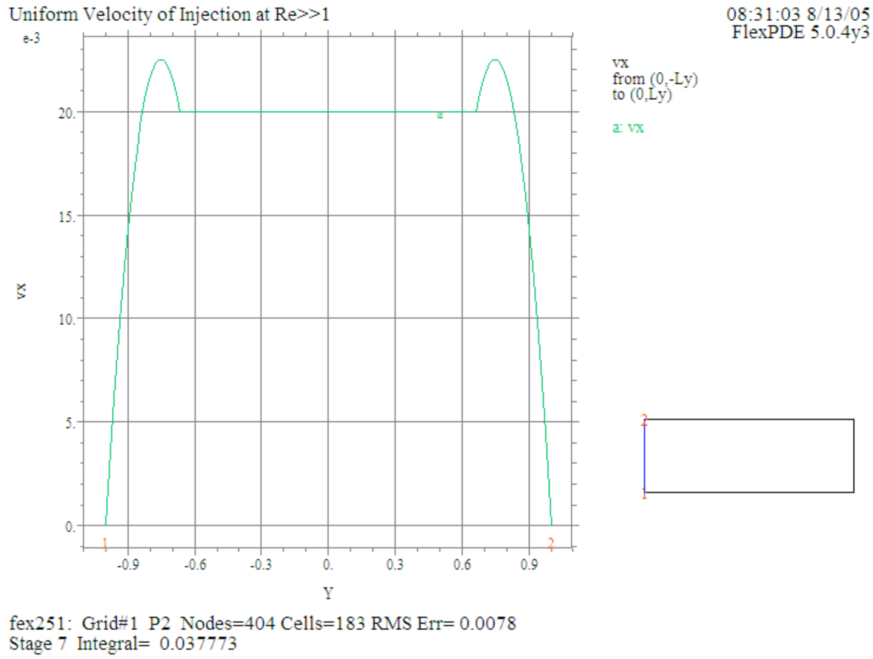
The following contour plots show the velocity ratio vx/vx0 for the first and the last stages, the latter corresponding to $\mathrm{Re} = 40$. Using *File,View* to display all the plots we notice that the pattern around the symmetry plane changes after the first stages.





The following elevation plot shows the variation of vx across the entrance. The numerical integral is slightly smaller than the expected flux vx0*2*Ly, because of the vanishing velocity at the walls.

The corresponding profiles of vx at the middle and at the end approach a parabolic shape, with closely the same flux. After specifying a small errlim, we shall find even better results with the Professional Version, at the expense of longer run times.



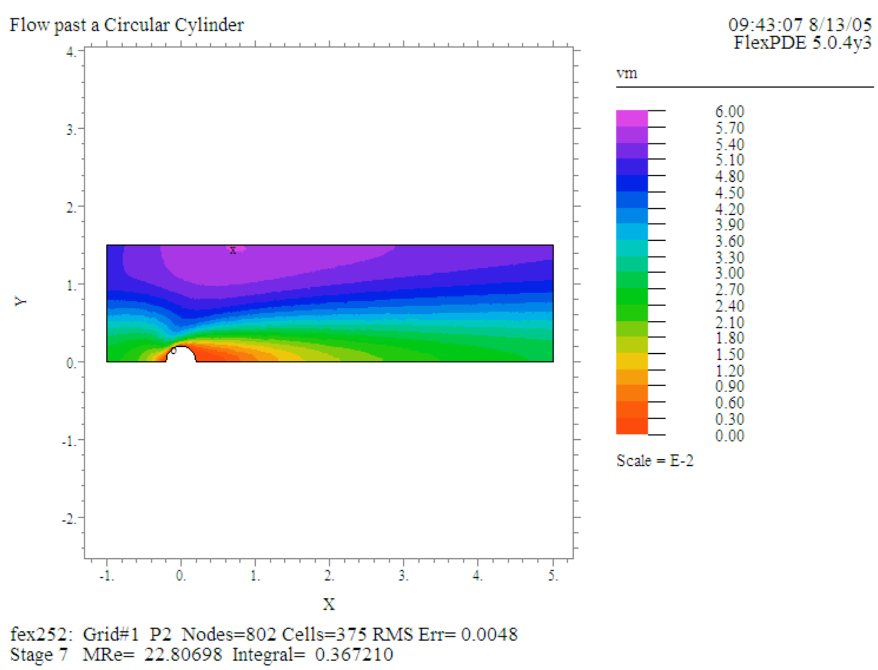
Viscous Flow past a Circular Cylinder

We shall next revisit the example on p.277, proceeding to larger values of Re. Here, we let the liquid *slip* on the wall, thereby reducing the drag force on the wall to a very small value. Also, we solve over only one-half of the real domain, using appropriate boundary conditions on the symmetry plane.

In view of the fact that the definition of Re is rather arbitrary in this case, we use a modified reference value, MRe, which relates to the size of the obstacle.

```
TITLE     'Flow past a Circular Cylinder'                        { fex252.pde }
SELECT    stages=7    spectral_colors
VARIABLES     vx     vy     p
DEFINITIONS
  Lx=1.0     Ly=1.5     r0=0.2     visc=1.0
  delp=staged( 1e-6, 1e-3, 1e-2, 3e-2, 0.1, 0.2, 0.2)   { Pressures}
  dens=1e3    MRe=dens*globalmax( vx)*2*r0/visc    { Modified Re }
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector(1,0)     unit_y=vector(0,1)
  nx=normal( unit_x)     ny=normal( unit_y)
  vxdvx=vx*dx(  vx)+ vy*dy( vx)     vxdvy=vx*dx( vy)+ vy*dy( vy)
  natp=    if stage=7 then visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
```

```
    -dens*[ nx*vxdvx+  ny*vxdvy]   else 0
  dens_term=    if stage=7 then dens*( dx( vxdvx)+ dy( vxdvy))  else 0
EQUATIONS
  vx:          dens*vxdvx+ dx( p)- visc*div( grad( vx))=0
  vy:          dens*vxdvy+ dy( p)- visc*div( grad( vy))=0
  p:           div( grad( p))+ dens_term- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0  natural( vy)=0   value( p)=delp               { In }
  line to (-Lx,0)   natural( vx)=0   value( vy)=0   natural(p)=0   { Symm. }
  line to (-r0,0)   value(vx)=0   value(vy)=0   natural(p)=natp
    arc( center=0,0) angle=-180 to (r0,0)                        { Cylinder }
  natural( vx)=0   value( vy)=0   natural(p)=0                   { Symmetry }
  line to (5*Lx,0)   natural( vx)=0   natural( vy)=0   value(p)=0     { Out }
  line to (5*Lx,Ly)   natural( vx)=0   value( vy)=0   natural(p)=natp
  line to finish                                                 { Wall }
PLOTS
  contour( vm) painted report( MRe)    contour( vx/delp) report( MRe)
  vector( v) norm     vector( v) norm zoom(0,0, 3*r0,3*r0) report( MRe)
  contour( p) painted report( MRe) report( delp*2*Ly/MRe)
  elevation( vx) from (-Lx,0) to (-Lx, Ly)
  elevation( vx) from (5*Lx,0) to (5*Lx, Ly)
  contour( abs( dens*vxdvx/visc/div( grad(vx)))/(5*Lx*Ly) /MRe)
END
```

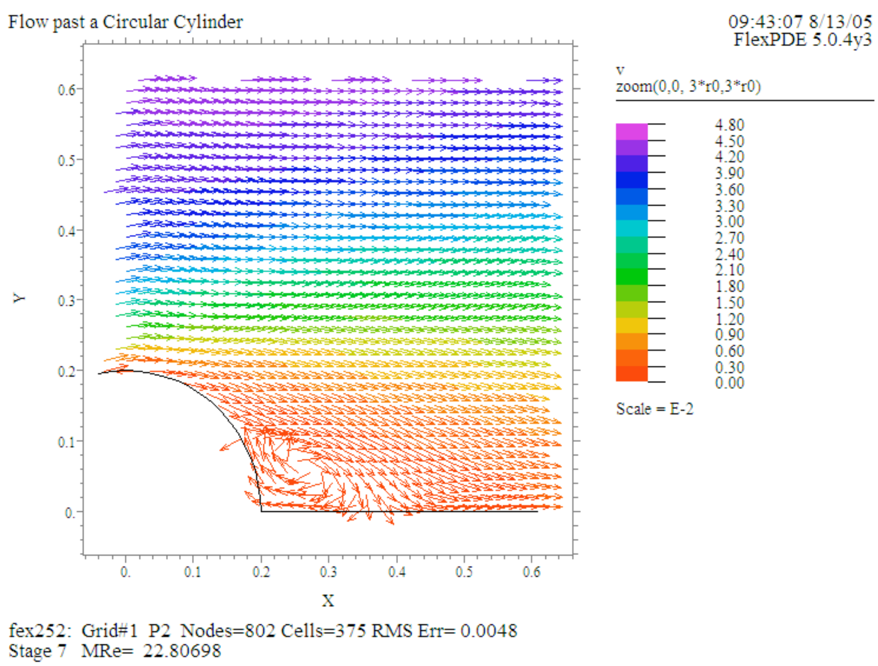


The above plot shows the speed vm at MRe≅23. The left-right symmetry we found at small MRe (p.278) is evidently broken. There is now an extended region in the wake of the cylindrical obstacle where the speed is very small.

At small MRe, the N-S equation is linear, and the velocity components are hence proportional to the pressure gradient. If we use *File,View* to compare the present contour plots of vx/delp, we find that the flow pattern starts to change significantly above MRe≅1.

On the pressure plot we also report the ratio of the drag force to the value of MRe. For small velocities we expect this to be constant, but here we find that it rises noticeably after the first stage. This trend is similar to that reported experimentally for a spherical object[8p111].

The next figure zooms on a region to the right of the obstacle, where we find evidence for slowly circulating flow. This circulation is entirely absent in the result for small MRe.



The last plot shows the absolute value of the ratio of inertial-to-viscous terms, divided by the domain area and by MRe. In view of the equation on p.275 we expect this average ratio to be about equal to unity. The plotted data are very scanty, but in essence the integrals bear out this relation.

Viscous Boundary Layer

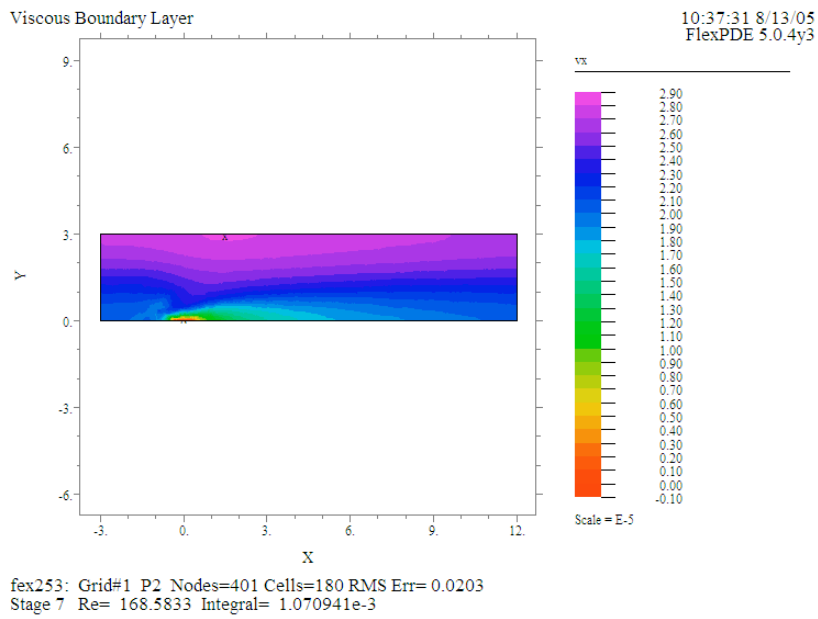
It is well known that the Bernoulli equation (pp.226ff) describes the flow rather well at large Re, even if it assumes that the liquid slips freely over solid surfaces. This fact led to the idea that the liquid is locked to the solid only over a thin layer, i.e. that the tangential speed increases rapidly from zero to a large value, approximately corresponding to the speed associated with the slip condition.

Let us explore whether we can find evidence for such a boundary layer phenomenon using the N-S equation. The file below is based on *fex251*, and we have introduced slip ($\partial v_x / \partial y = 0$) on the boundaries, except for a length of $2a$ on the lower wall.

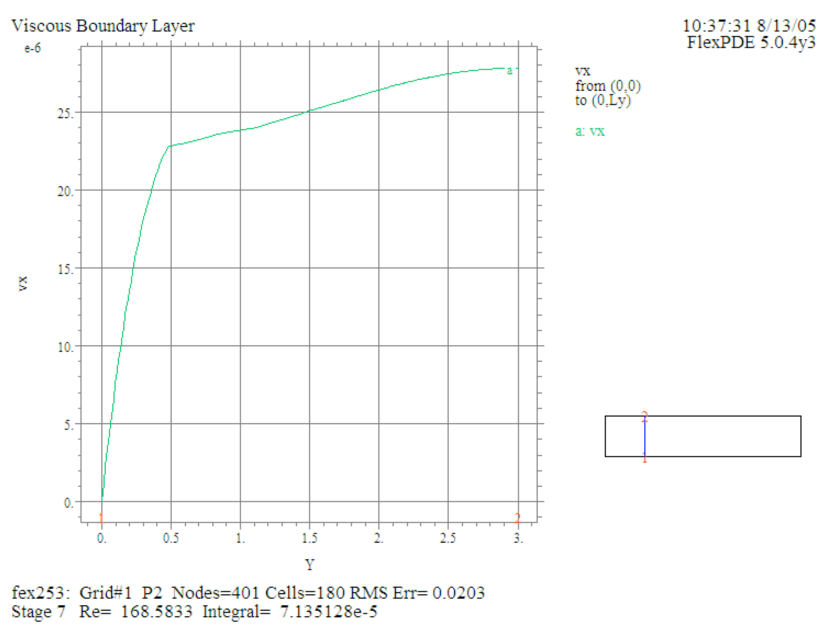
```
TITLE     'Viscous Boundary Layer'                                    { fex253.pde }
SELECT     stages=7     spectral_colors     nodelimit=400
VARIABLES     vx     vy     p
DEFINITIONS
  Lx=3.0     Ly=3.0     a=0.3     visc=1e-3
  delp=staged( 1e-11, 1e-10, 1e-9, 3e-9, 1e-8, 3e-8, 3e-8)
  dens=1e3     Re=dens*globalmax( vx)*2*Ly/visc
  v=vector( vx, vy)     vm=magnitude( v)
  unit_x=vector(1,0)     unit_y=vector(0,1)
  nx=normal( unit_x)     ny=normal( unit_y)
  vxdvx=vx*dx(  vx)+ vy*dy( vx)     vxdvy=vx*dx( vy)+ vy*dy( vy)
  natp=    if stage=7 then visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
    -dens*[ nx*vxdvx+  ny*vxdvy]   else 0
  dens_term=    if stage=7 then dens*( dx( vxdvx)+ dy( vxdvy))  else 0
EQUATIONS
  vx:          dens*vxdvx+ dx( p)- visc*div( grad( vx))=0
  vy:          dens*vxdvy+ dy( p)- visc*div( grad( vy))=0
  p:           div( grad( p))+ dens_term- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0   value( vy)=0   value(p)=delp                  { In }
  line to (-Lx,0)   natural( vx)=0   value( vy)=0   natural(p)=natp   { Slip }
  line to (-a,0)
  value(vx)=0   value(vy)=0   natural(p)=natp   line to (a,0)  { No slip }
  natural( vx)=0   value( vy)=0   natural(p)=natp   line to (4*Lx,0)
  natural( vx)=0   value( vy)=0   value(p)=0                       { Out }
  line to (4*Lx,Ly)   natural( vx)=0   value( vy)=0   natural(p)=natp
  line to close
```

```
PLOTS
  contour( vx) painted report( Re)     contour( p) painted
  vector( v) norm     contour( div( v))     contour( curl( v)) painted
  elevation( vx) from (0,0) to (0,Ly) report( Re)
END
```

The following plot shows that the flow pattern at Re≅170 spreads out in the direction of motion and that the speed rises steeply close to the sticky part of the wall.



The next plot gives us a more detailed view of the way vx varies across the boundary layer.



It is possible to make a rough estimate of the thickness δ of the boundary layer from a simplified N-S equation[8p101], known as the Euler equation. The result is

$$\delta = L_y \sqrt{\frac{1}{\mathrm{Re}}}$$

For the last stage of the calculations this gives us a thickness of 0.23, which is about what we can read from the above plot.

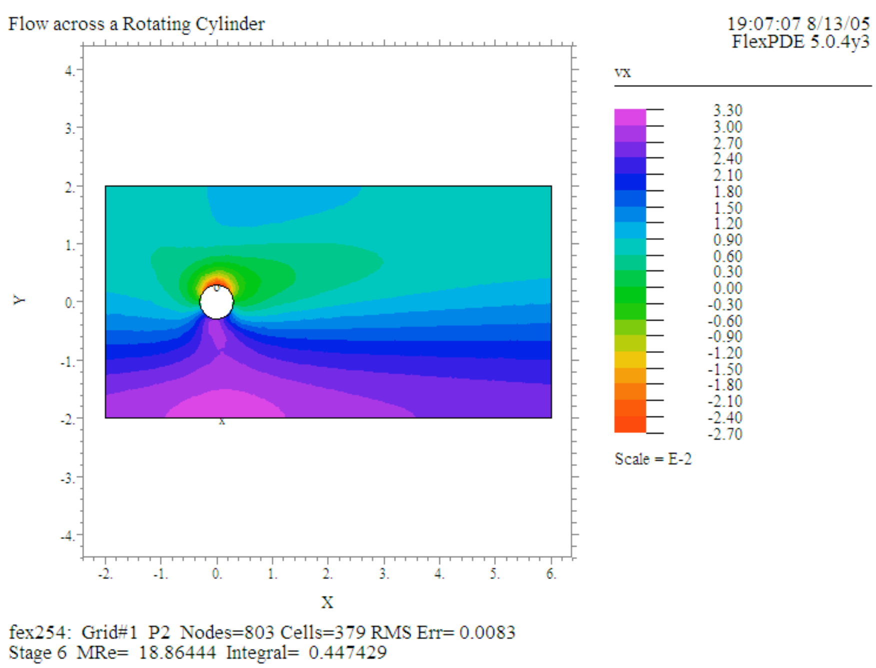
Viscous Flow past a Rotating Cylinder

Next we shall study the flow past a rotating cylinder, using slip conditions on the outer walls. Starting from *fex251*, we modify and add lines to obtain the descriptor below. As we increase the driving pressure and hence the speed of flow, we must also increase the speed of rotation omega in order to obtain a sequence of roughly similar velocity fields close to the cylinder.

```
TITLE     'Flow across a Rotating Cylinder'                    { fex254.pde }
SELECT     stages=6     spectral_colors
VARIABLES     vx     vy     p
DEFINITIONS
  Lx=2.0     Ly=2.0     r0=0.3     visc=1.0
  delp=staged( 1e-5, 1e-3, 3e-3, 0.01, 0.03, 0.03)     omega=3*delp
  dens=1e3     MRe=dens*globalmax( vx)*2*r0/visc
  v=vector( vx, vy)     vm=magnitude( v)
  unit_x=vector(1,0)     unit_y=vector(0,1)
  nx=normal( unit_x)     ny=normal( unit_y)
  vxdvx=vx*dx(  vx)+ vy*dy( vx)     vxdvy=vx*dx( vy)+ vy*dy( vy)
  natp=    if stage=6 then visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
    -dens*[ nx*vxdvx+  ny*vxdvy]   else 0
  dens_term=    if stage=6 then dens*( dx( vxdvx)+ dy( vxdvy))  else 0
  int_circ=line_integral( tangential( v),'circle')          { Circulation }
  fx=delp*2*Ly                                              { Force on liquid }
EQUATIONS
  vx:          dens*vxdvx+ dx( p)- visc*div( grad( vx))=0
  vy:          dens*vxdvy+ dy( p)- visc*div( grad( vy))=0
  p:           div( grad( p))+ dens_term- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
```

```
  natural( vx)=0   natural( vy)=0   value( p)=delp                        { In }
  line to (-Lx,-Ly)   natural( vx)=0   value( vy)=0   natural( p)=natp { Slip }
  line to (3*Lx,-Ly)   natural( vx)=0   natural( vy)=0   value( p)=0   { Out }
  line to (3*Lx,Ly) natural( vx)=0   value( vy)=0   natural( p)=natp  { Slip }
  line to close
start 'outline' (r0,0)                                   { Exclude cylinder }
  value( vx)=-omega*y   value( vy)=omega*x   natural(p)=natp
  arc( center=0,0) angle=360 close
feature
  start 'circle' (3*r0,0) arc( center=0,0) angle=360
PLOTS
  contour( vx) painted report(MRe)
  contour( vm) painted     vector( v) norm report(int_circ) report( fx)
  vector( v) norm zoom(-2*r0,-2*r0, 4*r0,4*r0)
  contour( p) painted     elevation( vx/2/Ly) from (3*Lx,-Ly) to (3*Lx,Ly)
  elevation( -p*normal( unit_y)) on 'outer' report(delp*2*Ly)       { p.288 }
END
```

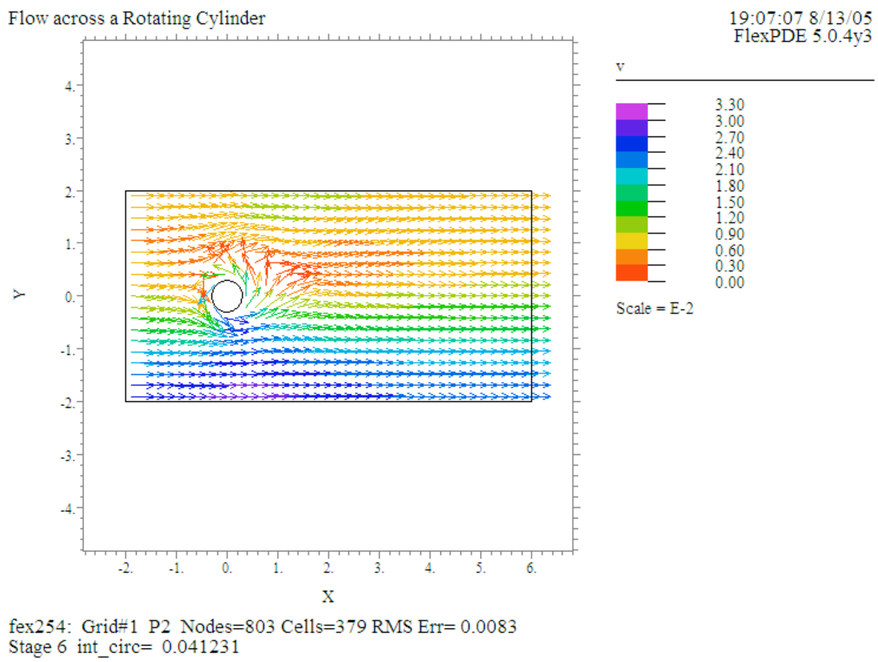
The following contour plot shows the final distribution of the speed vm, corresponding to MRe=19.



In the above plot for the highest pressure, we notice that the region of small speed, which was symmetrical for small MRe, now has shifted downstream.

The vector plot below reports a positive value for the circulation (p.243) on the *circle* enclosing the obstacle. This value is evidently smaller than what we obtain by integrating over the cylinder itself ($\omega r_0 \cdot 2\pi r_0 = 0.051$).

A trivial consequence of the circulation (Γ) is increased speed in the lower part of the plot and a corresponding decrease above the cylinder. This is similar to what we obtained by imposing a circulating field on a solution for a scalar potential (p.245).



Since the liquid slips over the walls, the horizontal force f_x on the cylinder (per unit length) must be closely equal to the driving force on the liquid at the left end. For the final stage we thus have $f_x = 2 * Ly * \mathrm{delp} = 0.12$.

We obtain the y-component of the force on the liquid from the combined elevation plot on 'outer'. The pressure integral for the highest value of MRe yields -0.27. This negative force on the liquid implies a downward force on the cylinder.

Estimating the force on the cylinder by means of the Kutta-Joukovski formula we find

$$f_y = -\rho v_{x0} \Gamma \cong -10^3 \cdot 0.014 \cdot 0.041 = -0.57,$$

taking the average value of v_{x0} from the elevation plot. The result is much larger than the pressure integral for f_y, not surprising in view of the different conditions assumed in the K-J theory.

Inspecting the results for all stages we discover that the vertical force is much smaller than the drag force for $\mathrm{MRe} \ll 1$ and increases to about 2.3 times the drag at the highest value.

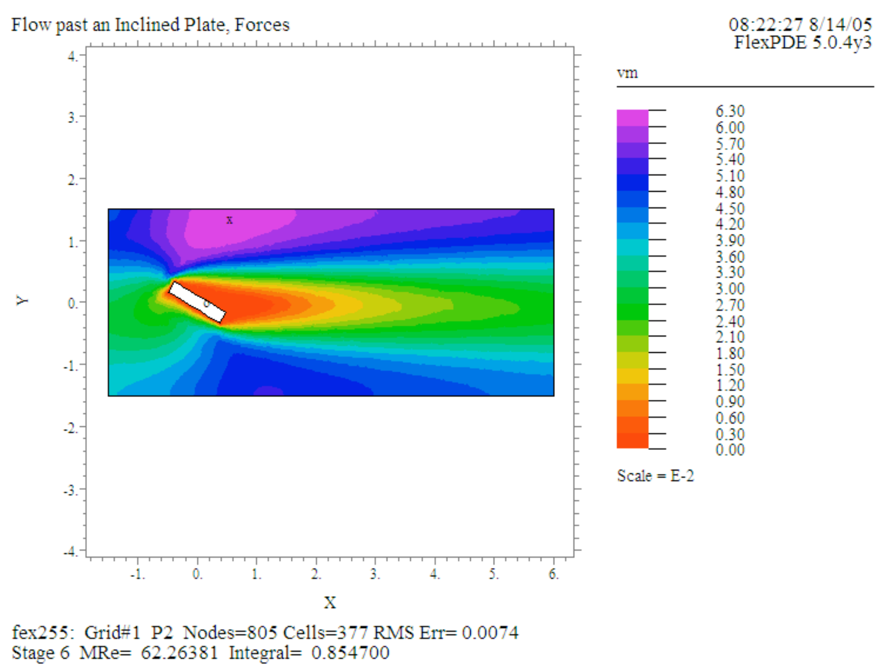
Viscous Flow past an Inclined Plate

Using the same PDEs as in recent examples, we may now revisit that of an inclined plate in an initially parallel stream. We need to modify *fex254* as follows, using parts of *fex212* (p.287).

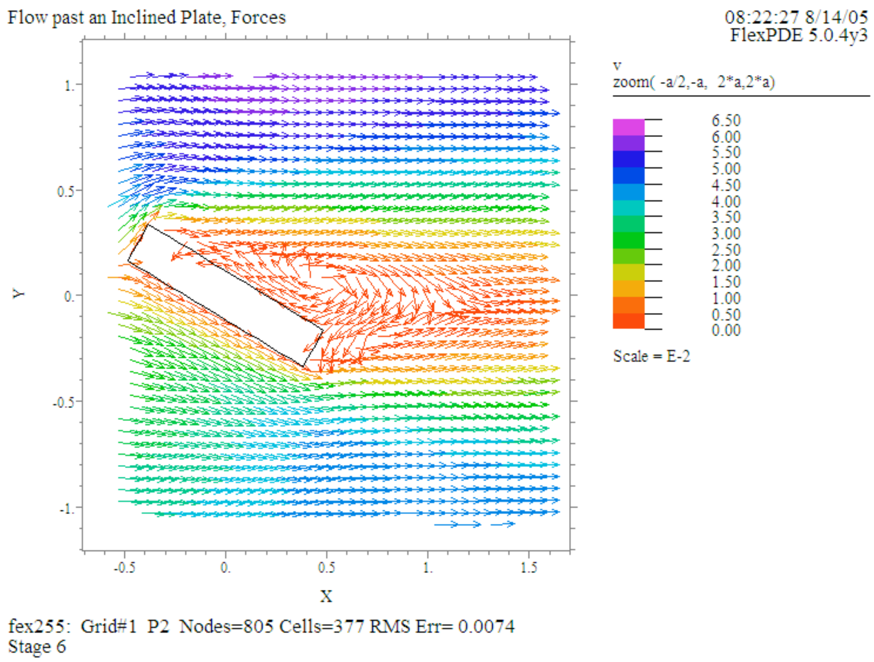
```
TITLE     'Flow past an Inclined Plate, Forces'                { fex255.pde }
SELECT    stages=6    spectral_colors
VARIABLES    vx    vy    p
DEFINITIONS
  Lx=1.5    Ly=1.5    a=1.0    d=0.2
  alpha=30*pi/180                          { Angle of attack, radians }
  si=sin( alpha)    co=cos( alpha)
  x1=-d/2*si- a/2*co    y1=-d/2*co+ a/2*si
  x2=d/2*si- a/2*co    y2=d/2*co+ a/2*si
  x3=-x1    y3=-y1    x4=-x2    y4=-y2
  visc=1.0    delp=staged(1e-5, 0.03, 0.1, 0.2, 0.4, 0.4)
  dens=1e3    MRe=dens*globalmax( vx)*a/visc
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector(1,0)    unit_y=vector(0,1)
  nx=normal( unit_x)    ny=normal( unit_y)
  vxdvx=vx*dx(  vx)+ vy*dy( vx)    vxdvy=vx*dx( vy)+ vy*dy( vy)
  natp=    if stage=6 then visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
    -dens*[ nx*vxdvx+  ny*vxdvy]   else 0
  dens_term=    if stage=6 then dens*( dx( vxdvx)+ dy( vxdvy))  else 0
EQUATIONS
  vx:        dens*vxdvx+ dx( p)- visc*div( grad( vx))=0
  vy:        dens*vxdvy+ dy( p)- visc*div( grad( vy))=0
  p:         div( grad( p))+ dens_term- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES    region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0   natural( vy)=0    value(p)=delp                    { In }
  line to (-Lx,-Ly)   natural( vx)=0   value( vy)=0   natural(p)=natp { Slip }
  line to (4*Lx,-Ly)  natural( vx)=0   natural( vy)=0   value(p)=0     { Out }
  line to (4*Lx,Ly)   natural( vx)=0   value( vy)=0   natural(p)=natp { Slip }
  line to close
  start 'outline' (x1,y1)                                  { Exclude plate }
  value( vx)=0  value( vy)=0   natural(p)=natp
```

```
  line to (x2,y2) to (x3,y3) to (x4,y4) to close
PLOTS
  contour( vx/delp) report(MRe)     contour( vm) painted report(MRe)
  vector( v) norm zoom( -a/2,-a,  2*a,2*a)
  contour( p) painted report(delp*2*Ly)                        { Force_x }
  elevation( -p*normal( unit_y)) on 'outer' report(delp*2*Ly)        { p.288 }
END
```

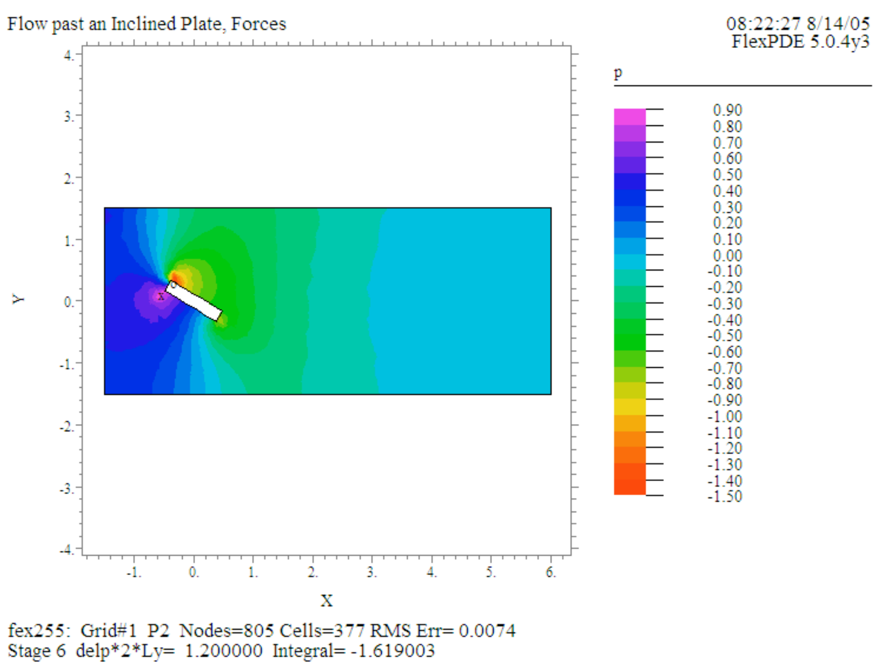
There are several points to note in the results of this run. The smallest driving pressure yields MRe<<1, and the corresponding velocity plots are essentially left-right symmetric. As delp increases, the velocity contours extend to the right and a region of small velocities appears in the wake. The plot below shows this phenomenon for the final stage at MRe≅62.



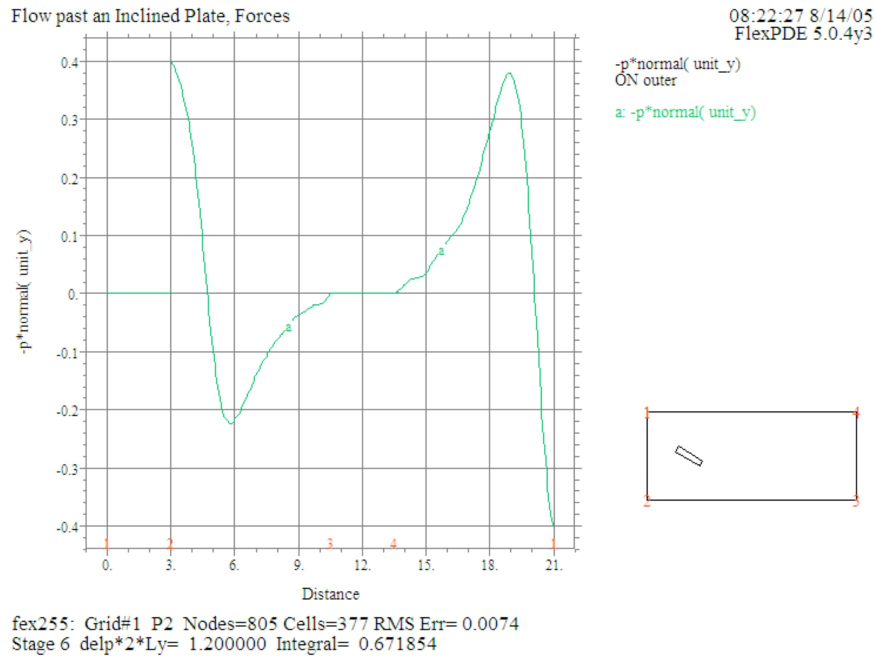
In the vector plot below we notice more details concerning the region of small speed just to the right of the plate. The flow lines indicate full circulation in the wake.



The plot below demonstrates that the pressure at large MRe becomes higher under the airfoil, which partly explains the lift force. The plot also reports the force on the liquid domain, which should be equal to the *drag* force on the slab.



From the combined integral in the elevation plot (below) we obtain the lift force. For MRe≅62 it becomes 0.67, or only 56% of the drag.



The viscosity of air is five orders of magnitude smaller than in the above example, the ensuing speeds being correspondingly higher, and we may thus expect lift to dominate in the aerodynamic case.

Viscous Flow past an Airfoil

Before leaving the subject of lift on an obstacle we shall study a more realistic case, viz. that of an airfoil. This example is mainly a reminder that FlexPDE allows you to trace rather complicated shapes.

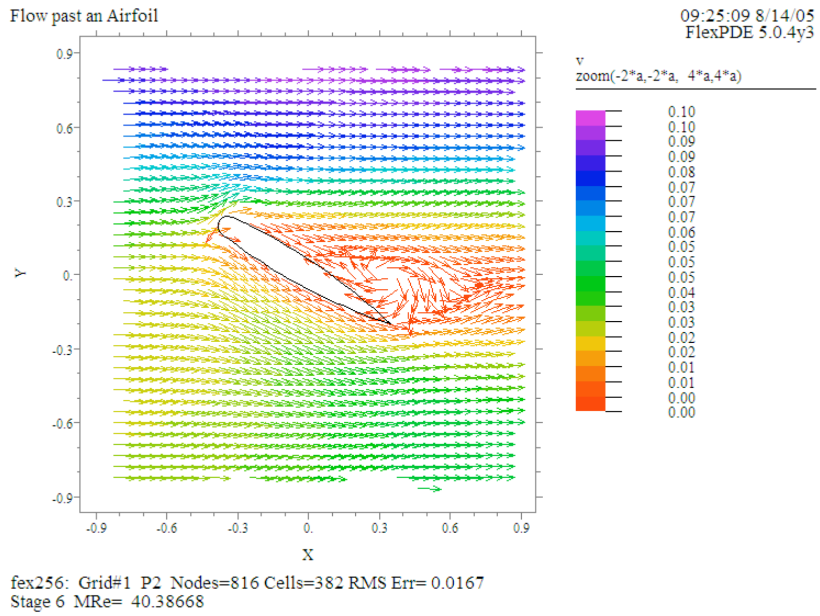
Under *definitions* we define the geometrical parameters of the airfoil, assumed cylindrical. Three arcs are sufficient for creating a symmetric shape. The *radius of curvature* is positive when the center is to the left of the curve, with respect to the direction in which it is traced.

```
TITLE     'Flow past an Airfoil'                              { fex256.pde }
SELECT     stages=6     spectral_colors
VARIABLES     vx     vy     p
DEFINITIONS
  Lx=1.0     Ly=1.0     a=0.4     d=0.1*a
{ Geometric parameters for inclined airfoil: }
  alpha=30*pi/180                         { Angle of attack, radians }
  si=sin( alpha)     co=cos( alpha)
  x1=-a*co- d*si     y1=a*si- d*co        { New definitions }
```

```
  x2=a*co    y2=-a*si
  x3=-a*co+ d*si    y3=a*si+ d*co
  visc=1.0    delp=staged(1e-4, 0.03, 0.1, 0.2, 0.4, 0.4)
  dens=1e3    MRe=dens*globalmax( vx)*a/visc
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector(1,0)    unit_y=vector(0,1)
  nx=normal( unit_x)    ny=normal( unit_y)
  vxdvx=vx*dx(  vx)+ vy*dy( vx)    vxdvy=vx*dx( vy)+ vy*dy( vy)
  natp=   if stage=6 then visc*[ nx*div( grad( vx))+ ny*div( grad( vy))]
    -dens*[ nx*vxdvx+  ny*vxdvy]   else 0
  dens_term=   if stage=6 then dens*( dx( vxdvx)+ dy( vxdvy))  else 0
EQUATIONS
  vx:         dens*vxdvx+ dx( p)- visc*div( grad( vx))=0
  vy:         dens*vxdvy+ dy( p)- visc*div( grad( vy))=0
  p:          div( grad( p))+ dens_term- 1e4*visc/Ly^2*div( v)=0
BOUNDARIES    region 'domain'  start 'outer' (-Lx,Ly)
  natural( vx)=0   natural( vy)=0    value(p)=delp                    { In }
  line to (-Lx,-Ly)   natural( vx)=0   value( vy)=0   natural(p)=natp { Slip }
  line to (4*Lx,-Ly)  natural( vx)=0   natural( vy)=0   value(p)=0     { Out }
  line to (4*Lx,Ly)   natural( vx)=0   value( vy)=0   natural(p)=natp { Slip }
  line to close
  start 'airfoil' (x1,y1)                                         { Exclude }
  value( vx)=0   value( vy)=0   natural(p)=natp
  arc( radius=6*a) to (x2,y2)
  arc( radius=6*a) to (x3,y3) arc( radius=1.006*d)   to close
PLOTS
  elevation( nx) on 'airfoil'                              { Direction cosine }
  contour( vx/delp) report(MRe)    contour( vm) painted
  vector( v) norm  zoom(-2*a,-2*a,  4*a,4*a) report(MRe)
  elevation( -p*normal( unit_y)) on 'outer' report(delp*2*Ly)        { p.288 }
END
```

The first plot is just a test of the shape of the airfoil. It shows the variation of the direction cosine along the border, and the resulting curve should be continuous on the front surface. The present choice of 1.006*d as the smaller radius of curvature turns out to yield compatible directions at point 1.

The following figure is a zoomed vector plot of the velocity. The flow lines show that there is circulation in the wake, but not as pronounced as in the case of the slab.



From the integral value on the last elevation plot we gather that the lift now exceeds the drag for the largest MRe.

Exercises

❑ Introduce the pressure difference delp between the ends of the channel in *fex251* instead of the input velocity vx0. Use a suitable number of stages and values up to delp=0.5.

❑ Using *fex251* as a template, put vx equal to the analytic expression for vx_ex in *fex202*. Furthermore, put vy=0 and p=-delp/3*x+ 2*delp/3. Make contour plots of the left members of the PDEs to show that this solution remains valid even at very large Re.

❑ Reduce the width of the channel in *fex251* to one-half over the first half of its length, using the same input velocities as before.

❑ Exploit the inherent symmetry of *fex203a* to halve the solution domain. Write a staged descriptor that extends calculations Re>>1.

❑ Deform the obstacle in *fex252* into a cylinder of ellipsoidal cross-section. Let the diameter in the direction of the *x*-axis be 2*a.

❑ Explore the effects of changing the angle of attack in *fex255* to zero, then to 60^0.

❑ Modify *fex256* to compare with the case where the airfoil is turned through 180 degrees.

26 Viscous Flow at Re>>1 in (ρ,z)

Let us now consider a few examples of steady, axially symmetric flow at high speed. The PDEs for this case were developed in a previous chapter (p.297). We found that the Navier-Stokes equation could be written

$$\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v}-\begin{Bmatrix}F_\rho\\F_z\end{Bmatrix}+\begin{Bmatrix}\dfrac{\partial p}{\partial\rho}\\\dfrac{\partial p}{\partial z}\end{Bmatrix}-\eta\begin{Bmatrix}\dfrac{1}{\rho}\dfrac{\partial}{\partial\rho}\left(\rho\dfrac{\partial v_\rho}{\partial\rho}\right)-\dfrac{v_\rho}{\rho^2}+\dfrac{\partial^2 v_\rho}{\partial z^2}\\\dfrac{1}{\rho}\dfrac{\partial}{\partial\rho}\left(\rho\dfrac{\partial v_z}{\partial\rho}\right)+\dfrac{\partial^2 v_z}{\partial z^2}\end{Bmatrix}=0$$

The first term of this equation may now be expanded as follows.

$$\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v}=\rho_0\left(v_\rho\frac{\partial}{\partial\rho}+v_z\frac{\partial}{\partial z}\right)\begin{Bmatrix}v_\rho\\v_z\end{Bmatrix}=\rho_0\begin{Bmatrix}v_\rho\dfrac{\partial v_\rho}{\partial\rho}+v_z\dfrac{\partial v_\rho}{\partial z}\\v_\rho\dfrac{\partial v_z}{\partial\rho}+v_z\dfrac{\partial v_z}{\partial z}\end{Bmatrix}\qquad\bullet$$

which yields the first two Navier-Stokes PDEs in their final form

$$\rho_0\begin{Bmatrix}v_\rho\dfrac{\partial v_\rho}{\partial\rho}+v_z\dfrac{\partial v_\rho}{\partial z}\\v_\rho\dfrac{\partial v_z}{\partial\rho}+v_z\dfrac{\partial v_z}{\partial z}\end{Bmatrix}-\begin{Bmatrix}F_\rho\\F_z\end{Bmatrix}+\begin{Bmatrix}\dfrac{\partial p}{\partial\rho}\\\dfrac{\partial p}{\partial z}\end{Bmatrix}-$$

$$\eta\begin{Bmatrix}\dfrac{1}{\rho}\dfrac{\partial}{\partial\rho}\left(\rho\dfrac{\partial v_\rho}{\partial\rho}\right)-\dfrac{v_\rho}{\rho^2}+\dfrac{\partial^2 v_\rho}{\partial z^2}\\\dfrac{1}{\rho}\dfrac{\partial}{\partial\rho}\left(\rho\dfrac{\partial v_z}{\partial\rho}\right)+\dfrac{\partial^2 v_z}{\partial z^2}\end{Bmatrix}=0\qquad\bullet$$

For the third equation we had

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial p}{\partial\rho}\right)+\frac{\partial^2 p}{\partial z^2}+\rho_0\nabla\cdot\left[(\mathbf{v}\cdot\nabla)\mathbf{v}\right]-\nabla\cdot\mathbf{F}-$$

$$C\frac{\eta}{L_0^2}\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho v_\rho\right)+\frac{\partial v_z}{\partial z}\right)=0$$

and it only remains to expand the term containing ρ_0. We already have an expression for $\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v}$ above, and it suffices to take the divergence according to the definition for cylindrical coordinates (p.290●1).

$$\rho_0\nabla\cdot\left[(\mathbf{v}\cdot\nabla)\mathbf{v}\right]=\rho_0\nabla\cdot\begin{Bmatrix} v_\rho\dfrac{\partial v_\rho}{\partial\rho}+v_z\dfrac{\partial v_\rho}{\partial z} \\ v_\rho\dfrac{\partial v_z}{\partial\rho}+v_z\dfrac{\partial v_z}{\partial z}\end{Bmatrix}=$$

$$\rho_0\left[\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\left(v_\rho\frac{\partial v_\rho}{\partial\rho}+v_z\frac{\partial v_\rho}{\partial z}\right)\right)+\frac{\partial}{\partial z}\left(v_\rho\frac{\partial v_z}{\partial\rho}+v_z\frac{\partial v_z}{\partial z}\right)\right]$$

Hence, the third PDE may be written

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial p}{\partial\rho}\right)+\frac{\partial^2 p}{\partial z^2}+$$

$$\rho_0\left[\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\left(v_\rho\frac{\partial v_\rho}{\partial\rho}+v_z\frac{\partial v_\rho}{\partial z}\right)\right)+\frac{\partial}{\partial z}\left(v_\rho\frac{\partial v_z}{\partial\rho}+v_z\frac{\partial v_z}{\partial z}\right)\right]- \quad \bullet$$

$$\nabla\cdot\mathbf{F}-C\frac{\eta}{L_0^2}\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho v_\rho\right)+\frac{\partial v_z}{\partial z}\right)=0$$

We also need the complete expression for the natural boundary condition for the pressure. From p.256 we recall the formula

$$\partial p/\partial n=\mathbf{n}\cdot\mathbf{F}+\eta\mathbf{n}\cdot\nabla^2\mathbf{v}-\rho_0\,\mathbf{n}\cdot\left[(\mathbf{v}\cdot\nabla)\mathbf{v}\right]$$

(omitting the term with the time derivative). We have just dealt with the expression within square brackets, so the result is almost immediate.

$$\partial p/\partial n \; = n_\rho F_\rho + n_z F_z + \eta\, n_\rho \left(\frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho \frac{\partial v_\rho}{\partial \rho} \right) - \frac{v_\rho}{\rho^2} + \frac{\partial^2 v_\rho}{\partial z^2} \right) +$$

$$\eta\, n_z \left(\frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho \frac{\partial v_z}{\partial \rho} \right) + \frac{\partial^2 v_z}{\partial z^2} \right) - \qquad \bullet$$

$$\rho_0 \left[n_\rho \left(v_\rho \frac{\partial v_\rho}{\partial \rho} + v_z \frac{\partial v_\rho}{\partial z} \right) + n_z \left(v_\rho \frac{\partial v_z}{\partial \rho} + v_z \frac{\partial v_z}{\partial z} \right) \right]$$

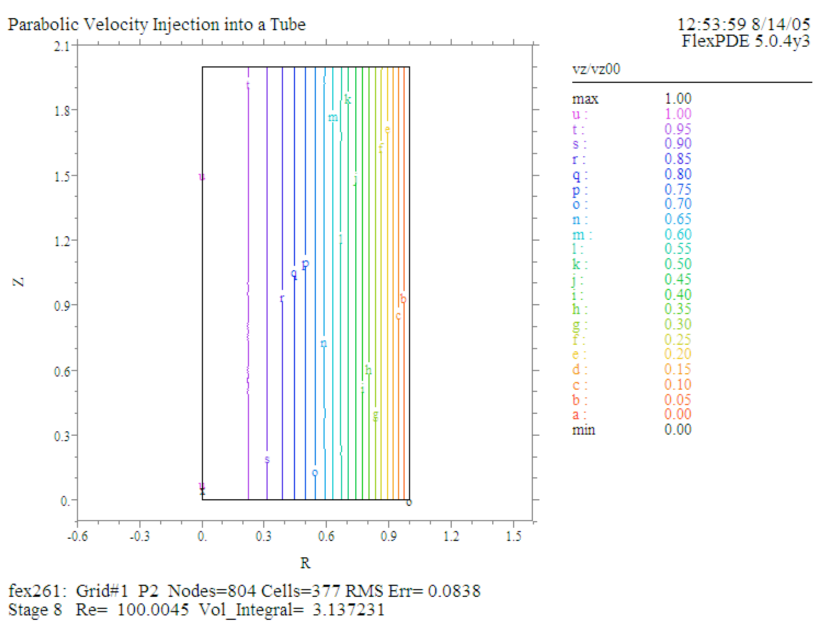
Parabolic Velocity Injection into a Tube

As a first application we change *fex251* from flow in a channel to flow in a tube. Using *fex231* as a template, we now include the terms proportional to ρ_0 for high Re as follows. In order to avoid a discontinuity of vx at the input we introduce parabolic input velocity. For shorter run times we use nodelimit. In the last stage we verify the agreement with the full formalism.

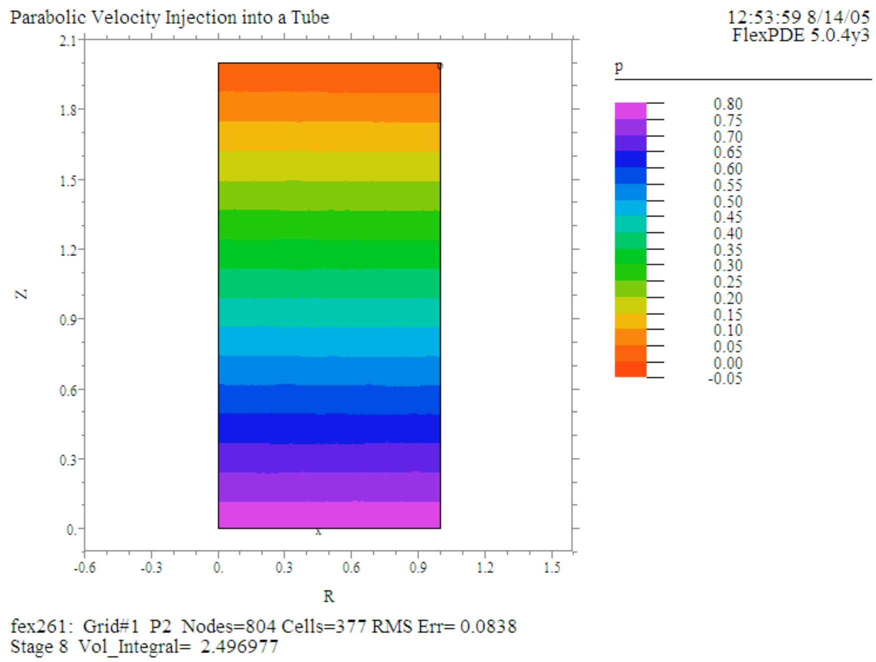
```
TITLE     'Parabolic Velocity Injection into a Tube'            { fex261.pde }
SELECT     stages=8     spectral_colors
COORDINATES     ycylinder('r','z')                         { Student Version }
VARIABLES     vr(1e-3)     vz(1e-3)     p(1e-3)     { Threshold }
DEFINITIONS
  L=2.0     r1=1.0     visc=1.0     dens=1e3
  vz00=staged(1e-6, 1e-3, 3e-3, 0.01, 0.03, 0.06, 0.1, 0.1)
  vz0=vz00*(1-(r/r1)^2)                         { Parabolic input velocity }
  v=vector( vr, vz)     vm=magnitude( v)
  Re=dens*globalmax( vm)*r1/visc
  div_v=1/r*dr(r*vr)+ dz(vz)     curl_phi=dz(vr)-dr(vz)
  unit_r=vector(1,0)     unit_z=vector(0,1)
  nr=normal( unit_r)     nz=normal( unit_z)
  vrdvr=vr*dr(vr)+ vz*dz(vr)     vrdvz=vr*dr(vz)+ vz*dz(vz)
  natp=   if stage=8 then  visc*nr*[1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]
   +visc*nz*[1/r*dr(r*dr(vz))+ dzz(vz)]- dens*[nr*vrdvr+  nz*vrdvz]   else 0
  dens_term=  if stage=8 then  dens*[1/r*dr(r*vrdvr)+dz( vrdvz)]   else 0
EQUATIONS
  vr:          dens*vrdvr+ dr(p)- visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]=0
  vz:          dens*vrdvz+ dz(p)- visc*[ 1/r*dr(r*dr(vz))+ dzz(vz)]=0
```

```
  p:          1/r*dr( r*dr(p))+ dzz(p)+ dens_term- 1e4*visc/L^2*div_v=0
BOUNDARIES
  region 'domain'  start  'outer' (0,0)
  natural(vr)=0   value(vz)=vz0   natural(p)=natp   line to (r1,0)  { In }
  value(vr)=0   value(vz)=0   natural(p)=natp   line to (r1,L)        { Wall }
  natural(vr)=0   natural(vz)=0   value(p)=0   line to (0,L)          { Out }
  value(vr)=0   natural(vz)=0   natural(p)=0   line to close          { Axis }
PLOTS
  contour( vz/vz00) report( Re)     contour( vr)
  contour( p) painted     vector( v) norm
  elevation( vz) from (0,0) to (r1,0) report(Re)          { Flux }
  elevation( vz) from (0,L/2) to (r1,L/2) report( Re)     { Flux }
  elevation( vz) from (0,0) to (r1,0) report(Re)          { Flux }
  elevation( p) from (0,0) to (r1,0) report(Re)           { Force_z }
  elevation( visc*dr(vz)) from (r1,0) to (r1,L)           { Viscous force }
END
```

Inspecting the plots of vz/vz00 for increasing Re by means of *File,View* we find that there is virtually no change over the range of Re from 1e-5 to 100. The results for the tube look similar to those for the channel (p.260). The vector plots indicate parallel flow, the distribution of vz over the cross-section remains parabolic, and the flux is closely constant along the tube.



The next plot of the pressure reveals that this mode of flow corresponds to linear variation of p along the tube, and the same remains true for all values of Re.



The last two elevation plots demonstrate the balance of driving force and viscous force.

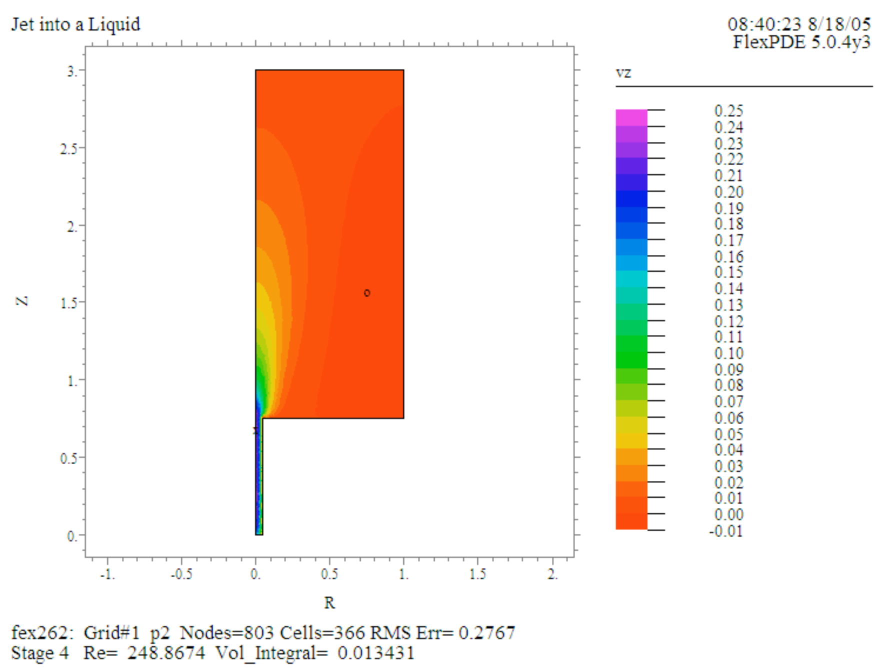
Jet into a Liquid

The next example illustrates the behavior of a thin pencil of liquid as it enters a tube containing liquid of the same kind. Much of the *fex261* descriptor remains valid, and the changes required should be clear from the list below. This script requires the Professional Version, although the nodelimit is only 800.

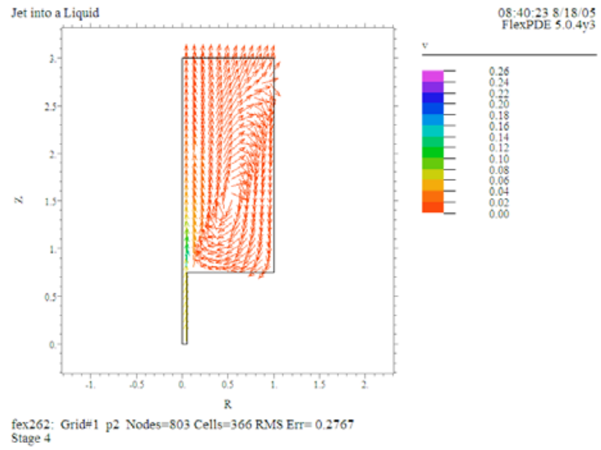
```
TITLE     'Jet into a Liquid'                              { fex262.pde }
SELECT    ngrid=20    stages=4    spectral_colors
  nodelimit=800                              { Professional Version }
COORDINATES    ycylinder('r','z')
VARIABLES    vr    vz    p
DEFINITIONS
  L=3.0    r1=1.0    r0=r1/20    visc=1.0    dens=1e3
  delp=staged( 1e-3, 100, 200, 300)                  { Replaces vz0 }
  v=vector( vr, vz)    vm=magnitude( v)
  Re=dens*globalmax( vm)*r1/visc
  div_v=1/r*dr(r*vr)+ dz(vz)    curl_phi=dz(vr)-dr(vz)
  unit_r=vector(1,0)    unit_z=vector(0,1)
  nr=normal( unit_r)    nz=normal( unit_z)
```

```
  vrdvr=vr*dr(vr)+ vz*dz(vr)     vrdvz=vr*dr(vz)+ vz*dz(vz)
  natp= visc*nr*[1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]
     +visc*nz*[1/r*dr(r*dr(vz))+ dzz(vz)]- dens*[nr*vrdvr+  nz*vrdvz]
EQUATIONS
  vr:  dens*vrdvr+ dr(p)- visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]=0
  vz:  dens*vrdvz+ dz(p)- visc*[ 1/r*dr(r*dr(vz))+ dzz(vz)]=0
  p:   1/r*dr( r*dr(p))+ dzz(p)+ dens*[1/r*dr(r*vrdvr)+dz( vrdvz)]
       -1e4*visc/L^2*div_v=0
BOUNDARIES
region 'domain'  start 'outer' (0,0)
  value(vr)=0   natural(vz)=0   value(p)=delp   line to (r0,0)        { In }
  value(vr)=0   value(vz)=0   natural(p)=natp
  line to (r0,L/4) to (r1,L/4) to (r1,L)
  natural(vr)=0   natural(vz)=0   value(p)=0   line to (0,L)          { Out }
  value(vr)=0   natural(vz)=0   natural(p)=0   line to finish
PLOTS
  contour( vz) painted report( Re)    contour( vr)    contour( p) painted
  vector( v) norm     contour( div_v)    contour( curl_phi) painted
END
```

The character of the flow changes dramatically after stage 1. In the last stage, at $\mathrm{Re} \cong 250$, the stream forms a long brush in the wider cylinder (below).



The corresponding vector plot shows that the axially symmetric circulation involves almost all of the volume in the wider part of the tube.



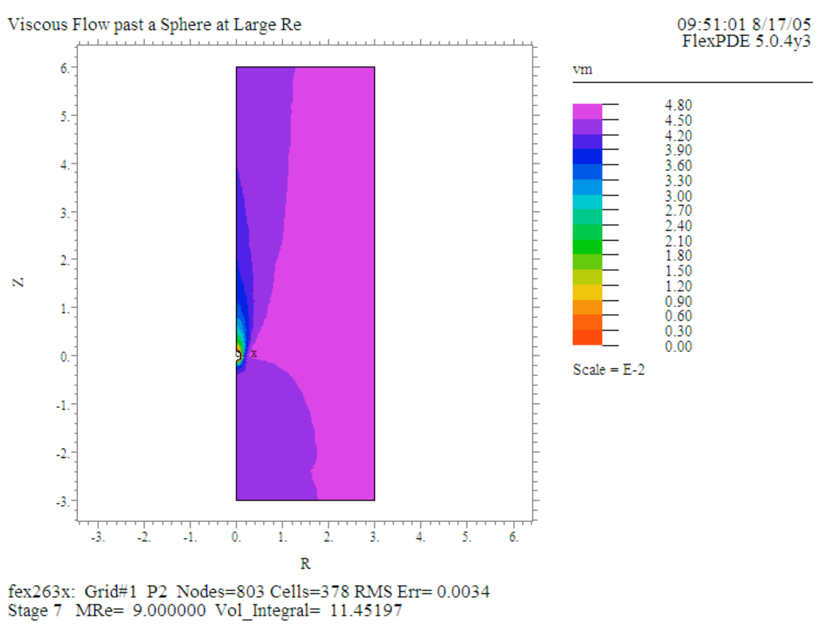
Viscous Flow past a Sphere

We shall now revisit the problem of a ball exposed to parallel flow (*fex233*). Again we may reuse some of the code in *fex261* as follows.

Under *definitions*, we define a modified Reynolds number MRe, based on the diameter of the ball, rather than the diameter of the tube. This is convenient for comparison with experimental data. On the wall of the tube we apply *slip* conditions.

```
TITLE     'Viscous Flow past a Sphere at Large Re'          { fex263.pde }
SELECT     stages=7     spectral_colors
COORDINATES     ycylinder('r','z')
VARIABLES     vr     vz     p
DEFINITIONS
  L=3.0     r1=3.0     r0= 0.1
  visc=1.0     dens=1e3
  vz0=staged( 1e-5,1e-3, 0.01, 0.02, 0.03, 0.04, 0.045)   { Input values }
  MRe=dens*vz0*2*r0/ visc          { Modified Re }
  v=vector( vr, vz)     vm=magnitude( v)
  div_v=1/r*dr(r*vr)+ dz(vz)     curl_phi=dz(vr)-dr(vz)
  unit_r=vector(1,0)     unit_z=vector(0,1)
  nr=normal( unit_r)     nz=normal( unit_z)
  vrdvr=vr*dr(vr)+ vz*dz(vr)     vrdvz=vr*dr(vz)+ vz*dz(vz)
  natp= visc*nr*[1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]
```

```
    +visc*nz*[1/r*dr(r*dr(vz))+ dzz(vz)]- dens*[nr*vrdvr+  nz*vrdvz]
  drag_S=6*pi*visc*r0*vz0            { After Stokes for small MRe }
EQUATIONS
  vr:        dens*vrdvr+ dr(p)- visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]=0
  vz:        dens*vrdvz+ dz(p)- visc*[ 1/r*dr(r*dr(vz))+ dzz(vz)]=0
  p:         1/r*dr( r*dr(p))+ dzz(p)+ dens*[1/r*dr(r*vrdvr)+dz( vrdvz)]
              -1e4*visc/L^2*div_v=0
BOUNDARIES
region 'domain' start(0,-L)
  natural(vr)=0   value(vz)=vz0   natural(p)=natp   line to (r1,-L)  { In }
  value(vr)=0   natural(vz)=0   natural(p)=natp   line to (r1,2*L)  { Wall }
  natural(vr)=0   natural(vz)=0   value(p)=0   line to (0,2*L)       { Out }
  value(vr)=0   natural(vz)=0   natural(p)=0   line to (0,r0)        { Axis }
  value(vr)=0   value(vz)=0   natural(p)=natp
    arc( center=0,0) angle=-180
  value(vr)=0   natural(vz)=0   natural(p)=0   line to close
PLOTS
  contour( vz/vz0)    contour( vm) painted report( MRe)
  contour( p) painted     vector( v) norm
  contour( div_v)    contour( curl_phi) painted
  elevation( p/drag_S) from (0,-L) to (r1,-L) report(MRe)   { Force_z }
END
```

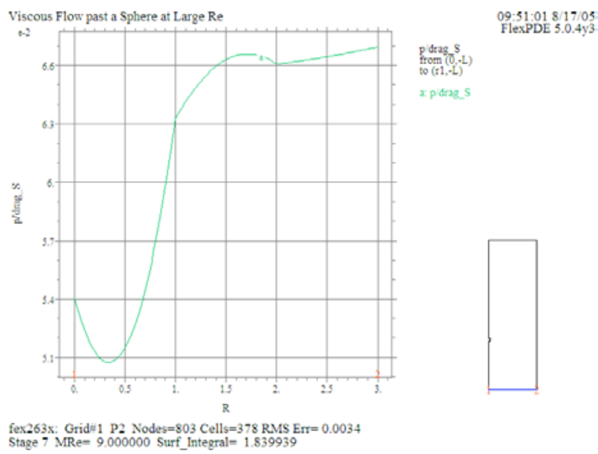


The above contour plot of vm shows the speed distribution in the last stage of the calculation. The low-speed region extends far toward the exit.

The ratio of the force obtained by FEA to that due to Stokes is particularly interesting, because experimental data for this ratio are available as a function of MRe. We obtain the force on the sphere from the final elevation plot of pressure. There we plot p/drag_S, which yields *force/drag_Stokes* after integration.

The first stage, for MRe<<1, reports a drag force that is about 1% larger than what we obtain from the Stokes formula. The latter is based on a parallel velocity vz0 at infinite distance, however, and the small deviation is probably caused by the limited size of our domain.

At higher speeds, this ratio increases to reach 1.84 at MRe=9 (figure below), which means that the FlexPDE results are in reasonable agreement with experimental data[8p111].



Exercises

❑ Reverse the direction of flow in *fex262*.

❑ Using *fex232* as a template, superimpose a pressure difference on gravity so as to produce upward flow. First study the low-pressure range where the velocity goes to zero, then use a pressure high enough to correspond to Re=100.

❑ Modify *fex263* to incorporate a cylindrical obstacle with its length equal to its diameter.

❑ Modify *fex263* by replacing the sphere with a cone of height equal to the diameter. Try both orientations.

27 Transient Viscous Flow at Re<<1

In previous chapters we have been concerned with steady motion of liquids. We shall now study a few cases of time-dependent flow in the regime of small Reynolds number ($\mathrm{Re} << 1$).

Let us start with the time-dependent form of the Navier-Stokes equation (p.252●1), which was based on Newton's law of motion.

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} + \left\{ \rho_0 (\mathbf{v} \cdot \nabla) \mathbf{v} - \mathbf{F} + \nabla p - \eta \nabla^2 \mathbf{v} \right\} = 0$$

Here, we have separated the mass-acceleration term from the force terms by brackets. Since $\mathrm{Re} << 1$ we neglect the non-linear part, $\rho_0 (\mathbf{v} \cdot \nabla) \mathbf{v}$, with respect to the other terms in the force bracket. Hence, in (x, y) space we are left with

$$\rho_0 \begin{Bmatrix} \dfrac{\partial v_x}{\partial t} \\ \dfrac{\partial v_y}{\partial t} \end{Bmatrix} - \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} + \begin{Bmatrix} \dfrac{\partial p}{\partial x} \\ \dfrac{\partial p}{\partial y} \end{Bmatrix} - \eta \begin{Bmatrix} \dfrac{\partial^2 v_x}{\partial x^2} + \dfrac{\partial^2 v_x}{\partial y^2} \\ \dfrac{\partial^2 v_y}{\partial x^2} + \dfrac{\partial^2 v_y}{\partial y^2} \end{Bmatrix} = 0 \qquad \bullet$$

The third PDE for pressure (p.254●2) similarly becomes

$$\nabla^2 p - \nabla \cdot \mathbf{F} - C \frac{\eta}{L_0^2} \nabla \cdot \mathbf{v} = 0 \qquad \bullet$$

Analogously, the natural boundary condition for pressure (p.256●1) may be written (for Re<<1)

$$\partial p / \partial n = n_x F_x + n_y F_y + \eta \left(n_x \nabla^2 v_x + n_y \nabla^2 v_y \right) - \rho_0 \, \partial v_n / \partial t$$

The last term, $-\rho_0 \, \partial v_n / \partial t$, may be omitted on a *fixed* boundary.

Transient Flow due to a Moving Wall in (x,y)

Our first example is related to *fex201* (p.257). The liquid is constrained by two walls, one stationary and one moving to the right at constant speed, starting at time $t = 0$. Before that instant, the entire volume is at rest.

Time-dependent problems require three new descriptor features, highlighted in the following descriptor. For the error estimate it is necessary to provide some coarse indication about the *range* of the dependent variables (see *Help*, *Threshold*).

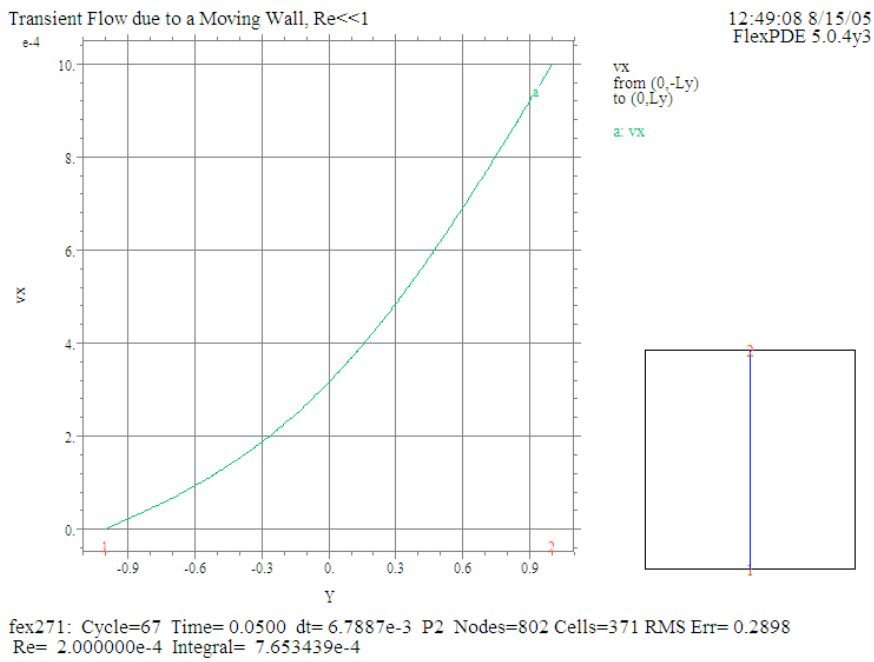
In the segment *initial values* we need to specify values for the dependent variables at $t = 0$. We also need to declare the problem to be time-dependent, and this is expressed by the time command. The statement under time specifies that the calculations are to start at time zero and to be continued up to a maximum value of 5e-2.

Finally, the *plot* segment includes a line beginning by for t=. It obviously lists the times at which we want plots. Since the last plot time should be the same as the last calculation time, we may use endtime to avoid entering a different value by oversight.

```
TITLE   'Transient Flow due to a Moving Wall, Re<<1'        { fex271.pde }
SELECT     spectral_colors                                { Student Version }
VARIABLES     vx( threshold=1e-5)    vy(1e-5)    p(1e-5)
DEFINITIONS
  Lx=1.0     Ly=1.0     vx0=1e-3     visc=1e4
  dens=1e3     Re=dens*vx0*2*Lx/visc          { Reynolds number }
  v=vector( vx, vy)     vm=magnitude( v)
INITIAL VALUES
  vx=0     vy=0     p=0                                   { For t<=0 }
EQUATIONS                                                 { For Re<<1 }
  vx:          dens*dt(vx)+ dx( p)- visc*div( grad( vx))=0
  vy:          dens*dt(vy)+ dy( p)- visc*div( grad( vy))=0
  p:           div( grad( p))- 1e4*visc/Ly^2*div(v)=0
BOUNDARIES                   { Normal velocity vn=0 on boundary }
region 'domain' start 'outer' (-Lx,Ly)
  natural( vx)=0   value( vy)=0   value(p)=0   line to (-Lx,-Ly)
  value( vx)=0   value( vy)=0   natural(p)=-visc*div( grad( vy))    { Wall }
  line to (Lx,-Ly)   natural( vx)=0   value( vy)=0   value(p)=0
  line to (Lx,Ly)   value( vx)=vx0   value( vy)=0                   { Wall }
  natural(p)=visc*div( grad( vy))   line to close
TIME
```

```
  from 0 to 5e-2
PLOTS
  for t=1e-4, 1e-3, 2e-3, 5e-3, 1e-2, 2e-2, endtime
  elevation( vx) from (0,-Ly) to (0,Ly) report( Re)
  contour( vx) painted     vector( v) norm   contour( p) painted
END
```

The plot below displays the variation of the horizontal velocity at t=0.05, recorded along a central line.



There is an analytic solution[9p191] in terms of Fourier series that could be used for comparison. At large times we would expect to recover the linear result from *fex201*.

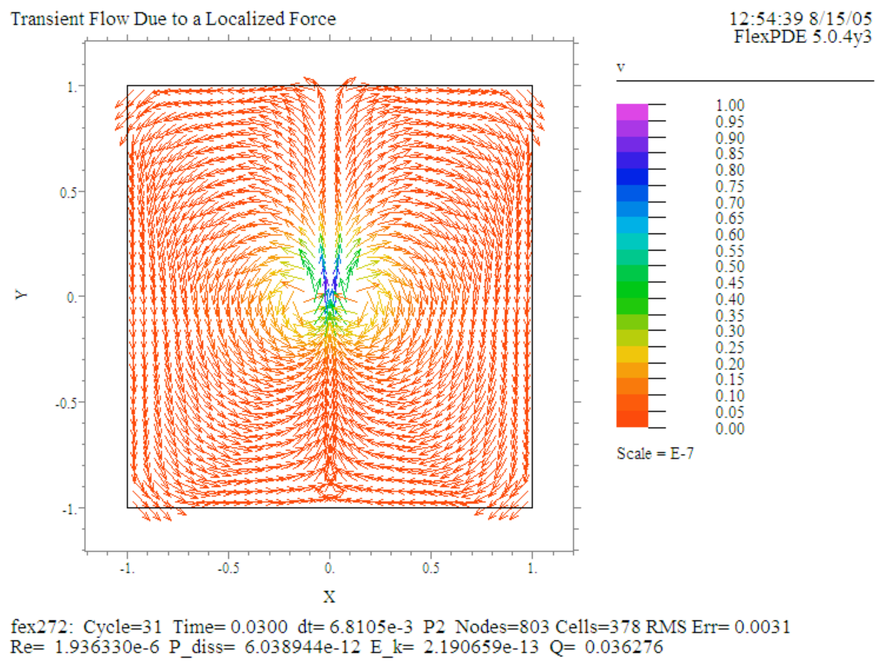
Transient Flow Due to a Localized Force

Let us next consider the flow caused by a vertical volume force Fy acting from $t = 0$ at the center of the domain. We may imagine this force to be generated by a laser beam heating the liquid over a thin cylinder along the *z*-axis, providing buoyancy. We disregard, however, the buoyancy force on the cooler liquid being transported away from the center.

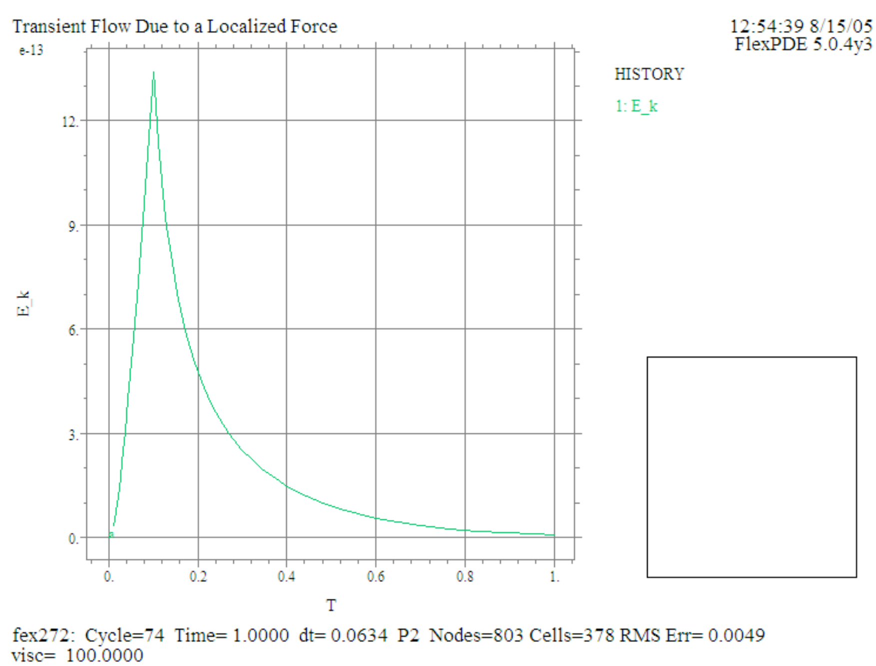
In order to reduce the run time we distribute the force in a Gaussian way, rather than specify a constant value inside a circular cylinder, which would create a space discontinuity. We stop the heating after part of the run by means of the discontinuous function ustep.

```
TITLE     'Transient Flow Due to a Localized Force'          { fex272.pde }
SELECT     spectral_colors
VARIABLES     vx(1e-9)     vy(1e-9)     p(1e-5)          { Thresholds }
DEFINITIONS
  Lx=1.0     Ly=1.0     visc=100     dens=1e3
  rad=sqrt(x^2+y^2)                                      { Radius }
  Fy=1e-2*exp(-rad^2/0.1^2)*ustep( 0.1- t)     { Force }
  v=vector( vx, vy)     vm=magnitude( v)
  unit_x=vector(1,0)     unit_y=vector(0,1)       { Unit vector fields }
  nx=normal( unit_x)     ny=normal( unit_y)      { Direction cosines }
  natp=ny*Fy                                             { Simplified }
  Re=dens*globalmax(vm)*2*Lx/visc               { Reynolds number }
  P_diss=                                                { Power dissipation }
    vol_integral( visc*[ 2*dx(vx)^2+ (dy(vx)+ dx(vy))^2+ 2*dy(vy)^2] )
  E_k=vol_integral( 1/2*dens*vm^2)               { Kinetic energy }
  Q=E_k/P_diss
INITIAL VALUES
  vx=0     vy=0     p=0
EQUATIONS
  vx:          dens*dt( vx)+ dx( p)- visc*div( grad( vx))=0
  vy:          dens*dt( vy)- Fy+ dy( p)- visc*div( grad( vy))=0
  p:           div( grad( p))- dy(Fy)- 1e4*visc/Ly^2*div(v)=0
BOUNDARIES
region 'domain' start 'outer' (-Lx,-Ly)  point value(p)=0
  value( vx)=0   value( vy)=0   natural(p)=natp
  line to (Lx,-Ly)  to (Lx,Ly) to (-Lx,Ly) to close
TIME
  from 0 to 1.0
PLOTS
  for t=1e-3, 0.01, 0.03, 0.1, 0.2, 0.3, 0.6, endtime
  contour( Fy) painted     contour( p) painted
  vector( v) norm report(Re) report(P_diss) report(E_k) report(Q)
  history( Fy)     history( E_k) report(visc)
END
```

The following vector plot shows the velocity at an intermediate time, before the force has been interrupted. The maximum speed is just above the center and there are regions of circulation to the left and right of the volume being heated. While the heating is on, the velocity component vy is mostly positive and increasing in the central region.



In the *definitions* segment we prepared to calculate the total kinetic energy (per unit depth of the domain), and also the total power of viscous dissipation (P_diss, p.285). We have combined these to form a ratio Q, analogous to the quality of a resonant cavity.

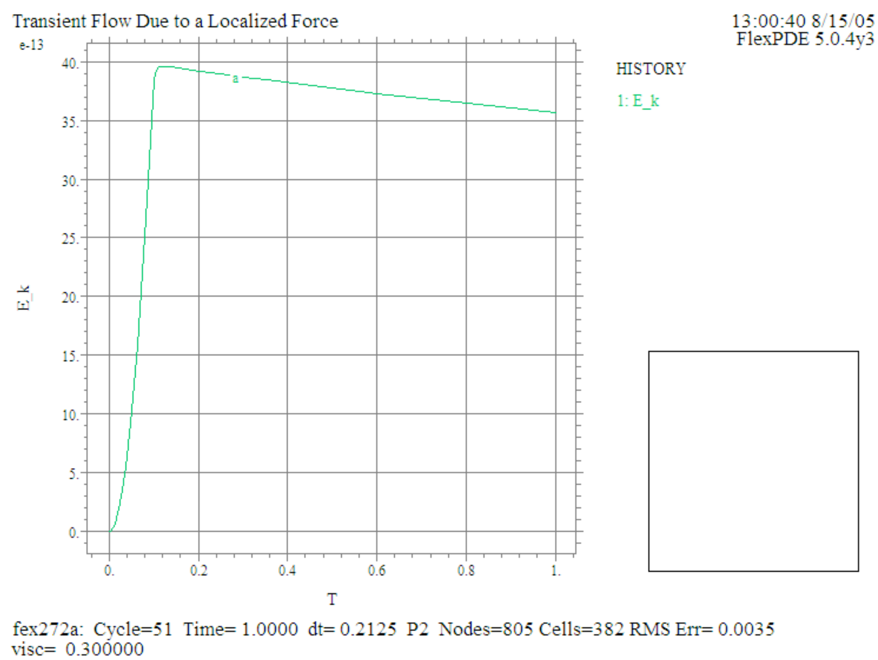


The above figure is a history plot displaying the variation of the kinetic energy versus time. Evidently, the accumulated energy increases steadily while the force is on, but after the latter is reduced to zero it gradually decreases. This obviously means that the residual motion of the fluid for $t > 0.1$ dissipates the kinetic energy stored.

It is instructive to extend the above problem to a liquid of much smaller viscosity, but still in the regime of Re<<1. In the next descriptor, only the viscosity value is different.

```
TITLE    'Transient Flow Due to a Localized Force'        { fex272a.pde }
…
  Lx=1.0    Ly=1.0    visc=0.3    dens=1e3
…
```

The new plot of E_k is shown below.



While the Q-value in the preceding case was much smaller than unity, it is now about 8.8. The above curve demonstrates the striking fact that the major part of the kinetic energy remains at the end of the run (1.0 s).

A common case of transient flow is *natural convection*, where a non-uniform temperature induces flow by buoyancy. In order to treat that kind of problem, we need to include the temperature as a dependent variable. As a first step in this development, we study a combination of heat transport by forced convection (liquid motion) and by conduction.

We are already somewhat familiar with convection from the preceding example. Heat conduction was briefly treated on p.120 in *Deformation and Vibration*. The fundamental PDE of heat conduction, in the case of a stationary medium[5p10], is

$$\nabla \cdot (-\lambda \nabla T) - h + \rho_0 c_p \frac{\partial T}{\partial t} = 0 \qquad \bullet$$

where λ is the thermal conductivity, T the temperature, h the heating power per unit volume, ρ_0 the mass density, and c_p the specific heat capacity.

This is valid for a *stationary* volume element. In a *moving* medium the time derivative must allow for the volume element traveling along the stream, just as in the analysis of viscous motion (p.252). If T is a function of (t, x, y), we obtain

$$\frac{DT}{Dt} = \frac{\partial T}{\partial t} + \frac{\partial T}{\partial x}\frac{dx}{dt} + \frac{\partial T}{\partial y}\frac{dy}{dt} = \frac{\partial T}{\partial t} + \frac{\partial T}{\partial x} v_x + \frac{\partial T}{\partial y} v_y = \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T$$

Substituting the expression for this new derivative we obtain

$$\nabla \cdot (-\lambda \nabla T) - h + \rho_0 c_p \frac{\partial T}{\partial t} + \rho_0 c_p \mathbf{v} \cdot \nabla T = 0 \qquad \bullet$$

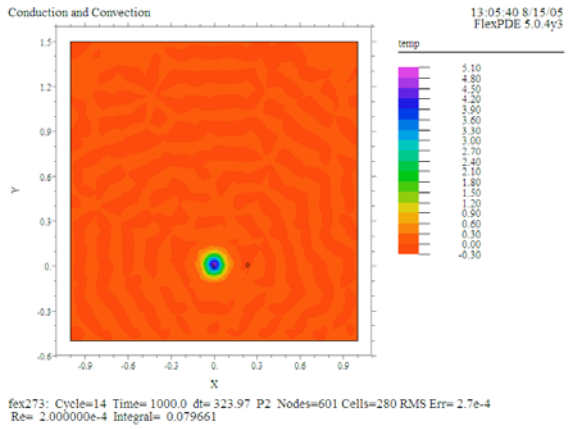
or for constant conductivity λ

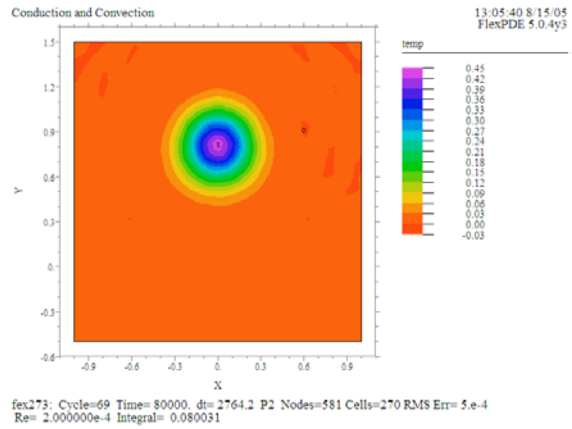
$$\frac{\lambda}{\rho_0 c_p} \nabla^2 T + \frac{h}{\rho_0 c_p} - \frac{\partial T}{\partial t} - v_x \frac{\partial T}{\partial x} - v_y \frac{\partial T}{\partial y} = 0 \qquad \bullet$$

Let us demonstrate simultaneous transport by the following example. We define a temperature distribution by an expression under *initial values*. Furthermore, we impose vertical liquid flow at the boundaries by suitable conditions. Then we expect the hot region to spread by conduction while it travels vertically (natural convection).

```
TITLE     'Conduction and Convection'                         { fex273.pde }
SELECT     spectral_colors
VARIABLES     vx(1e-5)     vy(1e-5)     p(1e-5)          { Thresholds }
  temp(1e-3)       { Temperature excess }
DEFINITIONS
  Lx=1.0     Ly=1.5     visc=1e2     dens=1e3
  cond=0.5     rcp=3e6     vy0=1e-5     rad=sqrt(x^2+y^2)
  v=vector( vx, vy)     vm=magnitude( v)
  unit_x=vector(1,0)     unit_y=vector(0,1)       { Unit vector fields }
  nx=normal( unit_x)     ny=normal( unit_y)       { Direction cosines }
  natp=0                                          { Simplified }
  Re=dens*globalmax(vm)*2*Lx/visc                 { Reynolds number }
INITIAL VALUES
  vx=0     vy=vy0     p=0     temp=10*exp(-rad^2/0.05^2)
EQUATIONS
  vx:          dens*dt( vx)+ dx( p)- visc*div( grad( vx))=0
  vy:          dens*dt( vy)+ dy( p)- visc*div( grad( vy))=0
  p:           div( grad( p))- 1e4*visc/Lx^2*div(v)=0
  temp:        (cond/rcp)*div( grad( temp))- dt( temp)
                - vx*dx( temp)- vy*dy( temp)=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,-Ly/3)
  point value(p)=0   point value( temp)=0
  value( vx)= 0   value( vy)=vy0   natural(p)=natp   natural( temp)=0
  line to (Lx,-Ly/3) to (Lx,Ly) to (-Lx,Ly) to close
TIME
  from 0 to 8e4
PLOTS
  for t=1e3, 3e3, 2e4, 4e4, 6e4, endtime
  contour( temp) painted report( Re)     vector(v) norm report( Re)
END
```

The plots below show the temperature distribution corresponding to the smallest and largest value of time. The hot region evidently expands while it rises.





Natural Convection in (x,y)

Local heating of a liquid causes a buoyancy force, which induces flow. This motion in turn transports heat, in addition to the well-known thermal conduction. The next example illustrates this effect. Here, the liquid is initially at rest, both inside the volume and on the boundary. The liquid is heated from below by a metal foil, which maintains a local temperature distribution while transmitting the ambient pressure.

We must now introduce the buoyancy force into the 2nd PDE, containing vy. Although heating will influence the density ρ_0 of the liquid, we assume this change to be small enough to be neglected in the Navier-Stokes equation. We take it into account, however, in the form of a vertical force

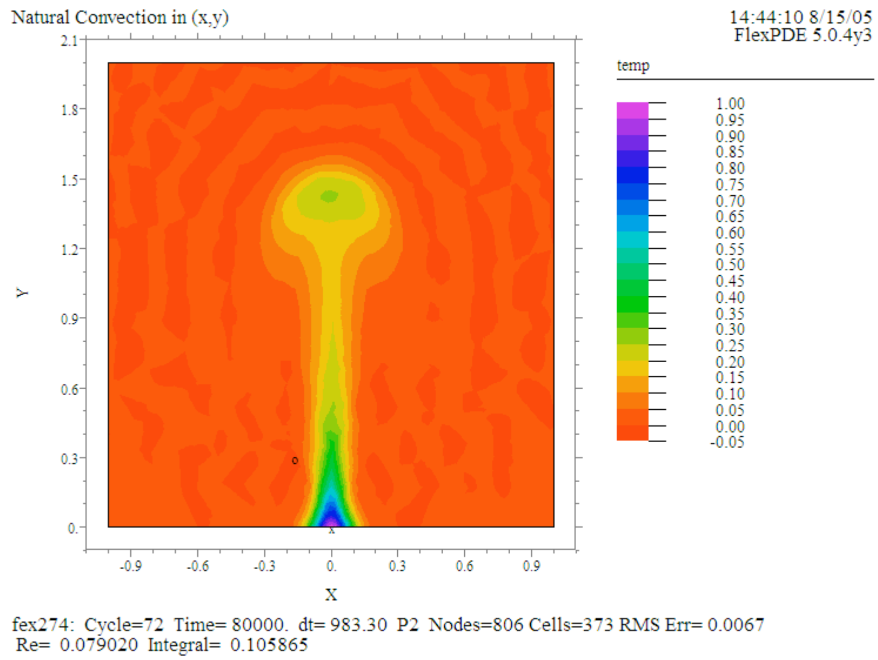
$$F_y = g\alpha\,\delta T \qquad \bullet$$

where g is the acceleration of gravity, α the volume thermal expansivity, and δT the temperature excess.

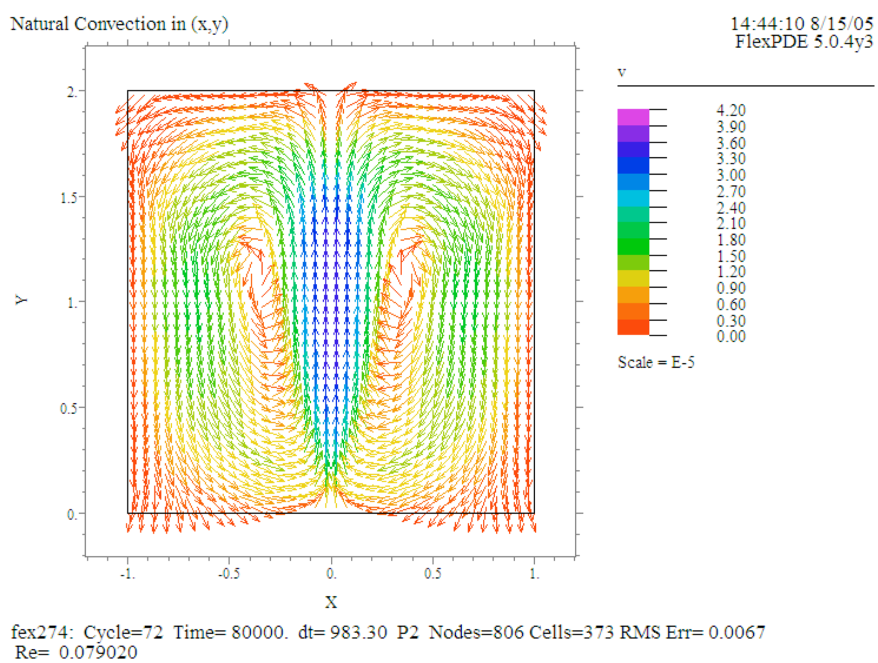
In order to shorten the calculation time we introduce an additional approximation. From the preceding examples we have seen that the propagation of fluid velocity is orders-of-magnitude faster than the conduction of heat. Thus we neglect the time derivative terms in the Navier-Stokes PDEs but retain that in the 4th PDE. We shall later verify, in an exercise, that this yields practically identical results.

```
TITLE     'Natural Convection in (x,y)'                          { fex274.pde }
SELECT    spectral_colors
VARIABLES    vx(1e-5)    vy(1e-5)    p(1e-3)    temp(1e-3)
DEFINITIONS
  Lx=1.0    Ly=2.0    visc=1.0    dens=1e3
  cond=0.5    rcp=3e6    rad=sqrt(x^2+y^2)
  Fy=1e-2*temp                    { Volume force with g*alpha=1e-2 }
  v=vector( vx, vy)    vm=magnitude( v)
  unit_x=vector(1,0)    unit_y=vector(0,1)      { Unit vector fields }
  nx=normal( unit_x)    ny=normal( unit_y)      { Direction cosines }
  natp=ny*Fy                                    { Simplified }
  y_shift=area_integral( y*temp)/area_integral( temp)
  Re=dens*globalmax(vm)*2*Lx/visc               { Reynolds number }
INITIAL VALUES
  vx=0    vy=0    p=0    temp=0
EQUATIONS
  vx:       dx( p)- visc* div( grad( vx))=0
  vy:       dy( p)- Fy- visc*div( grad( vy))=0
  p:        div( grad( p))- dy(Fy)- 1e4*visc/Lx^2*div(v)=0
  temp:          (cond/rcp)*div( grad( temp))- dt( temp)- vx*dx( temp)-
vy*dy( temp)=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,0)
  value( vx)=0   value( vy)=0                          { On all boundaries }
    value( p)=0   value( temp)=exp(-(10*x/Lx)^2)   { Heating }
  line to (Lx,0)   natural( p)=natp   natural( temp)=0
  line to (Lx,Ly) to (-Lx,Ly) to close
TIME
  from 0 to 8e4
PLOTS
  for t=1e3, 2e4 by 1e4 to endtime              { At constant intervals }
  contour( temp) painted report( Re)    vector(v) norm report( Re)
HISTORIES
  history( y_shift)
END
```

The following is the contour plot of the temperature corresponding to the maximum time. Evidently, the hot region rises and expands into a shape reminiscent of the well known “mushroom cloud” caused by a nuclear explosion.



The following vector plot of the velocity suggests a mechanism that could create the "mushroom hat".



The history plot shows the mean position of the temperature distribution as a function of time.

Natural Convection in (ρ,z)

In order to extend natural convection to (ρ, z) we just expand p.297●1 to obtain

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} - \begin{Bmatrix} F_\rho \\ F_z \end{Bmatrix} + \begin{Bmatrix} \dfrac{\partial p}{\partial \rho} \\ \dfrac{\partial p}{\partial z} \end{Bmatrix} - \eta \begin{Bmatrix} \dfrac{1}{\rho}\dfrac{\partial}{\partial \rho}\left(\rho \dfrac{\partial v_\rho}{\partial \rho}\right) - \dfrac{v_\rho}{\rho^2} + \dfrac{\partial^2 v_\rho}{\partial z^2} \\ \dfrac{1}{\rho}\dfrac{\partial}{\partial \rho}\left(\rho \dfrac{\partial v_z}{\partial \rho}\right) + \dfrac{\partial^2 v_z}{\partial z^2} \end{Bmatrix} = 0 \qquad \bullet$$

For the 3rd PDE, we may adopt p.298●3 as it reads.

We quote the pressure natural boundary condition from p.256●1, using the definition of $\nabla^2 \mathbf{v}$ from the preceding equation. We also note that the time derivative $\partial v_n / \partial t$ vanishes on a fixed boundary.

The 4th PDE, however, requires some revision of the formalism for a stationary medium[5p10]. The equation for T in that case is

$$\frac{1}{\rho_0 c_p} \nabla \cdot (-\lambda \nabla T) - \frac{1}{\rho_0 c_p} h + \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = 0$$

We have already used the appropriate expression for the divergence operator ∇ (p.290●1). To transform the last term is easy. For constant λ the last PDE finally becomes

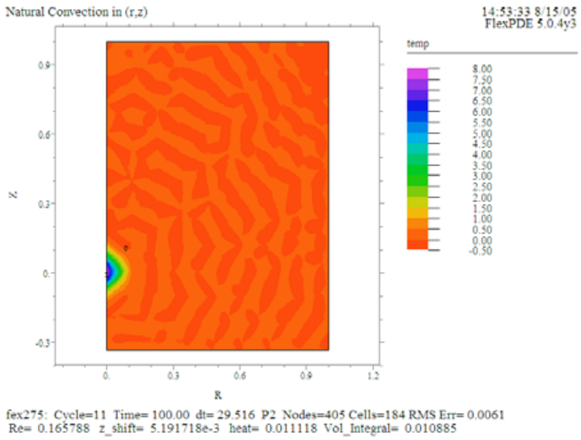
$$\frac{\lambda}{\rho_0 c_p} \left[\frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho \frac{\partial T}{\partial \rho}\right) + \frac{\partial}{\partial z}\left(\frac{\partial T}{\partial z}\right) \right] + \frac{h}{\rho_0 c_p} - \frac{\partial T}{\partial t} - v_\rho \frac{\partial T}{\partial \rho} - v_z \frac{\partial T}{\partial z} = 0 \qquad \bullet$$

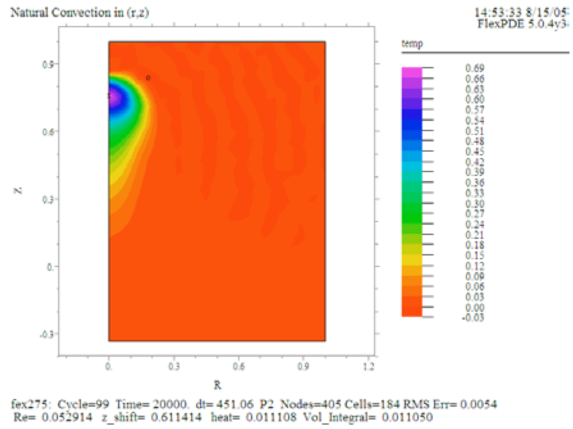
The following example is analogous to *fex274*, but the conditions are now axially symmetric.

```
TITLE     'Natural Convection in (r,z)'                          { fex275.pde }
SELECT     nodelimit=400     spectral_colors
COORDINATES     ycylinder('r','z')
VARIABLES     vr(1e-4)     vz(1e-4)     p(1e-4)     temp(1e-4)
DEFINITIONS
  Lr=1.0     Lz=1.0     visc=1.0     dens=1e3
  cond=0.5     rcp=3e6
```

```
  rad=sqrt(r^2+z^2)     v=vector( vr, vz)     vm=magnitude( v)
  Fz=1e-2*temp
  Re=dens*globalmax(vm)*2*Lr/visc               { Reynolds number }
  div_v=1/r*dr(r*vr)+ dz(vz)     curl_phi=dz(vr)-dr(vz)
  unit_r=vector(1,0)     unit_z=vector(0,1)
  nr=normal( unit_r)     nz=normal( unit_z)
  natp=nz*Fz                                    { Simplified }
  z_shift=vol_integral( z*temp)/vol_integral(temp)
  heat=vol_integral( temp)
INITIAL VALUES
   vr=0    vz=0    p=0    temp=10*exp(-rad^2/0.05^2)
EQUATIONS                                       { Simplified }
  vr:        dr(p)- visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]=0
  vz:        dz(p)- Fz- visc*[ 1/r*dr(r*dr(vz))+ dzz(vz)]=0
  p:         1/r*dr( r*dr(p))+ dzz(p)- dz(Fz)- 1e4*visc/Lr^2*div_v= 0
  temp:            (cond/rcp)* [1/r*dr( r*dr( temp))+ dz( dz(temp))]-
             dt( temp)-  vr*dr( temp)- vz*dz( temp)=0
BOUNDARIES
region 'domain' start 'outer' (0,-Lz/3)
  value( vr)=0   value( vz)=0   natural(p)=natp   natural( temp)=0  { Wall }
  line to (Lr,-Lz/3)   point value(p)=0   point value(temp)=0
  line to (Lr,Lz) to (0,Lz)
  value( vr)=0   natural( vz)=0   natural(p)=0   natural( temp)=0    { Axis }
  line to close
TIME
  from 0 to 2e4
PLOTS
  for t=1e2, 1e3, 2e3, 4e3, 6e3, 8e3, 1e4, endtime
  contour( temp) painted report( Re) report( z_shift) report( heat)
  history( z_shift)
END
```

The figures below show the first and last plots of the temperature. Here, an initially spherical distribution of hot liquid rises due to the density difference. The size of the heated volume also increases with time. We note that the full sequence of plots reports approximately constant values of the total amount of heat, as we should expect.





Exercises

❑ Modify *fex271* to study the flow after stopping the moving wall.

❑ In *fex274* we neglected the time derivatives of the velocity components. Modify this file to take these derivatives into account. Be warned that the run may take twice as long. Compare the resulting history curve with that obtained before, by superimposing the printed plots.

❑ Modify *fex274* by injecting the heat flux density 20*exp(-(10*x/Lx)^2)*ustep(1e4-t) from the bottom.

❑ After the model of *fex273*, modify *fex274* by introducing the initial temperature temp=2*exp(-rad^2/0.05^2) and the boundary condition value(temp)=0 on the bottom side.

❑ Change the initial value for the temperature in the preceding exercise to the anti-symmetric function
temp=20*x*exp(-rad^2/0.05^2)
Try to predict the results, and then execute the file.

❑ Using *fex274* as a model, modify *fex275* to study the corresponding natural convection process in (r,z), using the function value(temp)=2*exp(-(10*r/Lr)^2).
Compare and interpret the heat integrals.

28 Viscous Flow in Three Dimensions

In previous chapters, we have confined the flow analysis to problems in two dimensions, including some axially symmetric configurations. We shall now give examples of more general 3D calculations.

In the preceding volume on *Deformation and Vibration* we explored electric fields (p.141) by plotting analytic expressions corresponding to point charges, and there is no need to repeat those illustrations here. It would be wise to review this introduction to 3D, however, before reading further.

Extension of the Formalism to 3D

The Navier-Stokes equation takes the same general form in three dimensions, viz.

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} + \rho_0 (\mathbf{v} \cdot \nabla)\mathbf{v} - \mathbf{F} + \nabla p - \eta \nabla^2 \mathbf{v} = 0 \qquad \bullet$$

but we must be aware that **v** and **F** now have three components. The derived PDE for pressure also looks the same as before, i.e.

$$\nabla^2 p + \rho_0 \nabla \cdot \left[(\mathbf{v} \cdot \nabla)\mathbf{v} \right] - \nabla \cdot \mathbf{F} = 0$$

In order to reduce the divergence towards the ideal vanishing value we subtract the term $\nabla \cdot \mathbf{v}$, multiplied by a suitably chosen constant, on the left side. This leaves us with

$$\nabla^2 p + \rho_0 \nabla \cdot \left[(\mathbf{v} \cdot \nabla)\mathbf{v} \right] - \nabla \cdot \mathbf{F} - f_\nabla \nabla \cdot \mathbf{v} = 0 \qquad \bullet$$

where $f_\nabla = C\eta / L^2$, where C is a number, has the correct dimension.

The expansion of these equations for $\mathbf{v} = \{v_x, v_y, v_z\}$ is straightforward. The same is true of the natural boundary conditions, since we have

$$\partial p/\partial n = \mathbf{n}\cdot\nabla p = \mathbf{n}\cdot\mathbf{F} + \eta\,\mathbf{n}\cdot\nabla^2\mathbf{v} - \rho_0\,\mathbf{n}\cdot\frac{\partial \mathbf{v}}{\partial t} - \rho_0\,\mathbf{n}\cdot\left[(\mathbf{v}\cdot\nabla)\mathbf{v}\right] \qquad \bullet$$

We shall restrict the examples in this chapter to steady flow at small speeds, which means that the last two terms will vanish.

Flow through a Rectangular Duct

As an elementary problem, we shall study flow through a duct of rectangular cross-section. A pressure difference will drive the liquid through the tube in the direction of increasing z, from bottom to top.

The maximum number of nodes in the 2D Student Version is 800, but for 3D it has been increased to 1600.

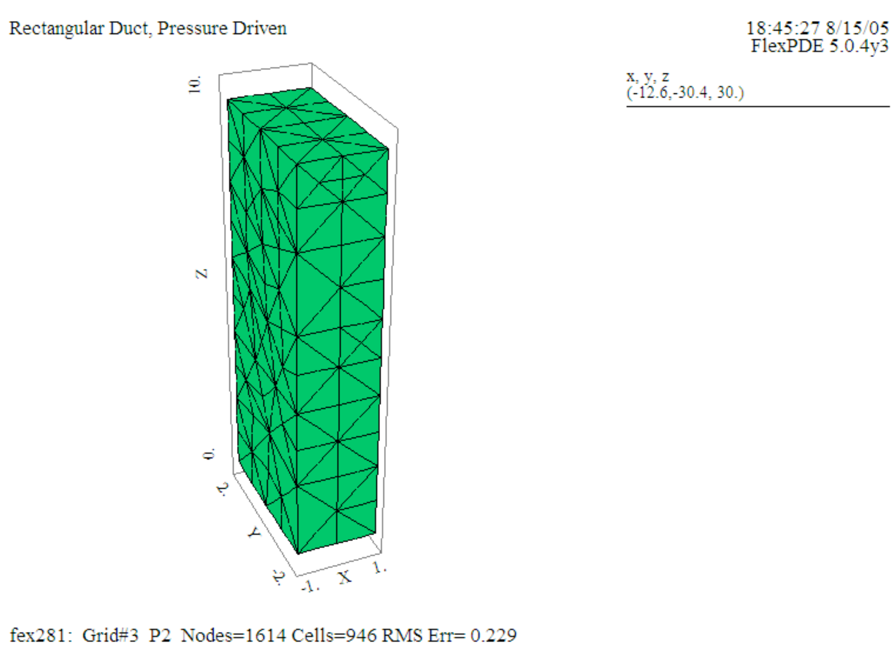
In the *boundaries* segment, under *surfaces*, we specify the pressure values delp and 0.

The section *region* defines a rectangle on the bottom plane. The duct is generated by extrusion of this curve into z space. Here, we also specify vanishing speed on the extruded walls.

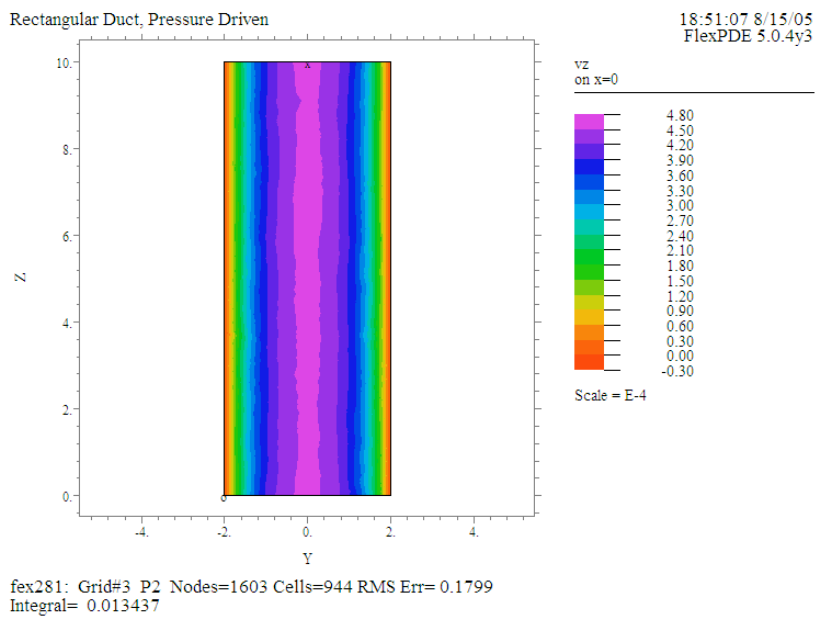
```
TITLE    'Rectangular Duct, Pressure Driven'                 { fex281.pde }
SELECT    errlim=1e-3    ngrid=4    spectral_colors
COORDINATES    cartesian3                              { Student Version }
VARIABLES    vx    vy    vz    p
DEFINITIONS
  Lx=1.0    Ly=2.0    Lz=10.0    visc=1e4    delp=1e2
  vz_a=delp/Lz/(2*visc)*(Lx^2- x^2)       { Analytic solution for channel }
  dens=1e3    Re=dens*globalmax( vz)*2*Ly/visc
  v=vector( vx, vy, vz)    vm=magnitude( v)
  unit_x=vector(1,0,0)    unit_y=vector(0,1,0)    unit_z=vector(0,0,1)
  nx=normal( unit_x)    ny=normal( unit_y)    nz=normal( unit_z)
{ Natural boundary condition for p }
  natp=0                                        { Simplified }
EQUATIONS                                       { For Re<<1 }
  vx:        dx( p)- visc*div( grad( vx))=0
  vy:        dy( p)- visc*div( grad( vy))=0
  vz:        dz( p)- visc*div( grad( vz))=0
  p:         div( grad( p))- 1e4*visc/Lx^2*div( v)=0
EXTRUSION                                       { Parallel surfaces }
  surface 'bottom' z=0
  layer 'liquid'                                { Layer containing liquid }
```

```
  surface 'top' z=Lz
BOUNDARIES
  surface 'bottom'  natural(vx)=0   natural(vy)=0   natural(vz)=0
    value(p)=delp
  surface 'top'   natural(vx)=0   natural(vy)=0   natural(vz)=0
  value(p)=0
region 'domain'                                   { Curve to be extruded }
  start 'outer' (-Lx,-Ly)  value( vx)=0   value( vy)=0   value( vz)=0
  natural( p)=natp   line to (Lx,-Ly)  to (Lx,Ly)  to (-Lx,Ly)  to close
MONITORS
  contour( vz) painted on x=0     contour( vz) painted on z=5.0
PLOTS
  grid( x, y, z)
  contour( p) painted on x=0 report(Re)
  contour( vz) painted on x=0   contour( vz) painted on y=0
  contour( vz) painted on z=0   contour( vz) painted on z=5.0
  contour( vz) painted on z=Lz
  elevation( p) from (0,0,0) to (0,0,Lz)
  elevation( vz, vz_a) from (0,0,0) to (0,0,Lz)
END
```

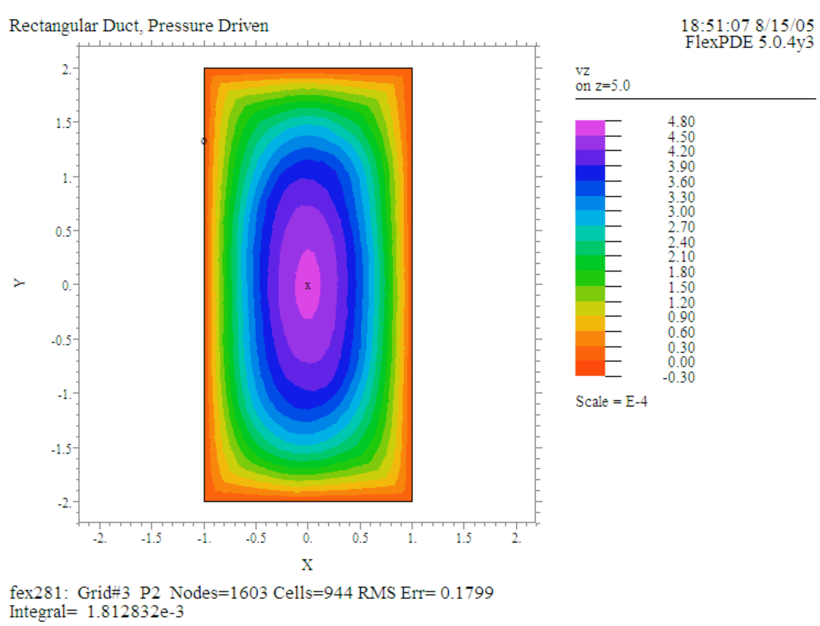
The following figure shows the geometry of the duct. The flow is upwards, in the direction of the z-axis. Of the two transverse sides, the one in the x direction is the smaller one.



The contour plot below shows the distribution of vz, which appears to be parabolic across the duct as in the 2D case (p.260). The variation along the stream is rather small and is probably due to random fluctuations in the numerical results.



The figure below illustrates that the longitudinal velocity component vz vanishes on the other walls, as required. The integral value gives us the flux through the cross-section. Comparing the three corresponding plots we find that the flux values are closely the same, which confirms that mass is conserved.

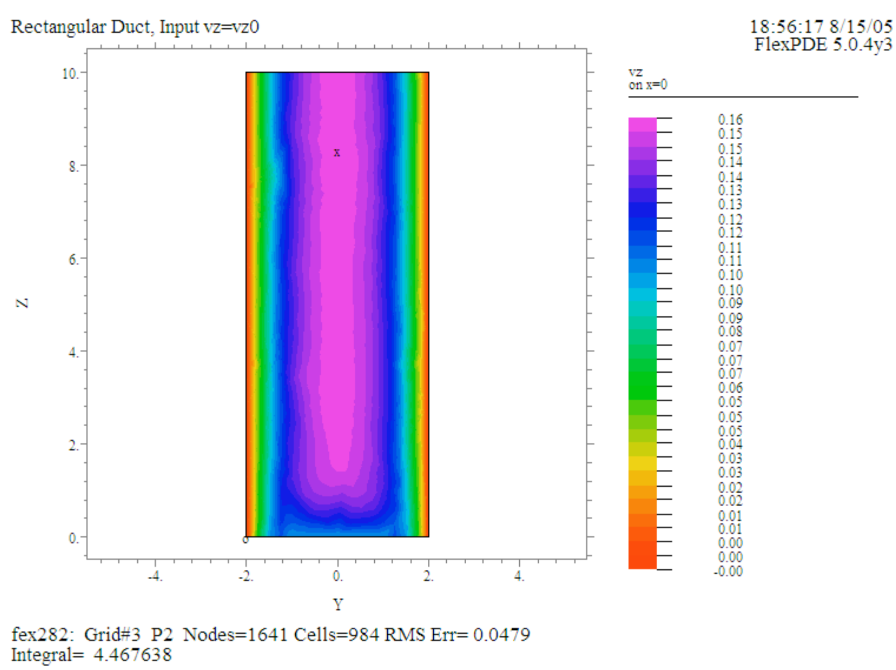


The last elevation plot shows that the actual velocity is about 10% lower than the analytical result for the channel (p.259). In the latter case, however, the larger side is infinite and hence the drag forces from the two distant walls are absent.

Let us next consider a modified example, where the input velocity has a constant value, vz0. The changes with respect to *fex281* are as follows.

```
TITLE 'Rectangular Duct, Input vz=vz0'                    { fex282.pde }
…
  Lx=1.0    Ly=2.0    Lz=10.0    visc=1e4    vz0=0.1
…                                           { No analytic estimate }
BOUNDARIES
  surface 'bottom'  natural(vx)=0   natural(vy)=0   value(vz)=vz0
    natural(p)=0
  surface 'top'   natural(vx)=0   natural(vy)=0   natural(vz)=0
  value(p)=0
…
  elevation( vz, vz0) from (0,0,0) to (0,0,Lz)
END
```

The plot of vz below is similar to that for a simple channel. The flux through successive cross-sections is still constant, although the first plot at z=0 looks different. This anomaly is caused by the discontinuity at the bottom, which FlexPDE cannot quite handle.



Flow through a Box with Two Orifices

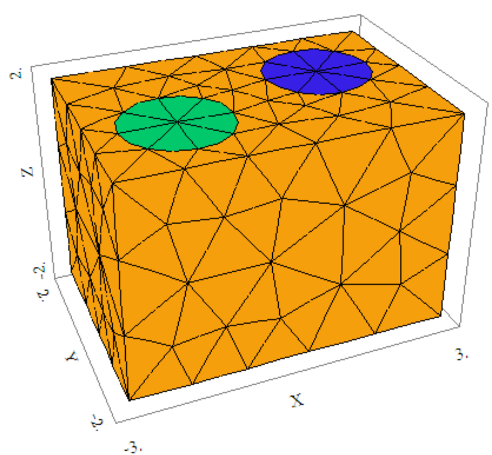
In the following problem we introduce three *regions*, one box and two circular cylinders. The latter, smaller objects are extruded from circles on the base plane. After this operation, the total volume becomes divided into three sub-regions, two cylinders and the remainder, which has a more complicated shape. The liquid properties are the same in all of them, but the cylinders trace out different parts of the bottom and top surfaces, which makes it possible to assign different pressures to the orifices.

We first assign global boundary conditions to the flat bottom and top surfaces. Under 'box' we then supply the corresponding conditions for the other sides of the envelope. Under the regions 'in' and 'out' we over-write conditions for the pressure on the orifices.

```
TITLE   'Flow through a Box with Two Orifices'                 { fex283.pde }
SELECT    errlim=1e-3    ngrid=4    spectral_colors
COORDINATES    cartesian3
VARIABLES    vx    vy    vz    p
DEFINITIONS
  Lx=3.0    Ly=2.0    Lz=2.0    r0=1.0    visc=1e4    delp=1e2
  dens=1e3    Re=dens*globalmax( vz)*2*Ly/visc
  v=vector( vx, vy, vz)    vm=magnitude( v)
  unit_x=vector(1,0,0)    unit_y=vector(0,1,0)    unit_z=vector(0,0,1)
  nx=normal( unit_x)    ny=normal( unit_y)    nz=normal( unit_z)
{ Natural boundary condition for p }
  natp=0                                          { Simplified }
EQUATIONS                                         { For Re<<1 }
  vx:       dx( p)- visc*div( grad( vx))=0
  vy:       dy( p)- visc*div( grad( vy))=0
  vz:       dz( p)- visc*div( grad( vz))=0
  p:        div( grad( p))- 1e4*visc/Lx^2*div( v)=0
EXTRUSION                                         { Parallel surfaces }
  surface 'bottom' z=-Lz
  layer 'liquid'                                  { Layer containing liquid }
  surface 'top' z=Lz
BOUNDARIES
  surface 'bottom'   value( vx)=0   value( vy)=0   value( vz)=0
    natural( p)=natp
  surface 'top'   value( vx)=0   value( vy)=0   value( vz)=0
    natural( p)=natp
```

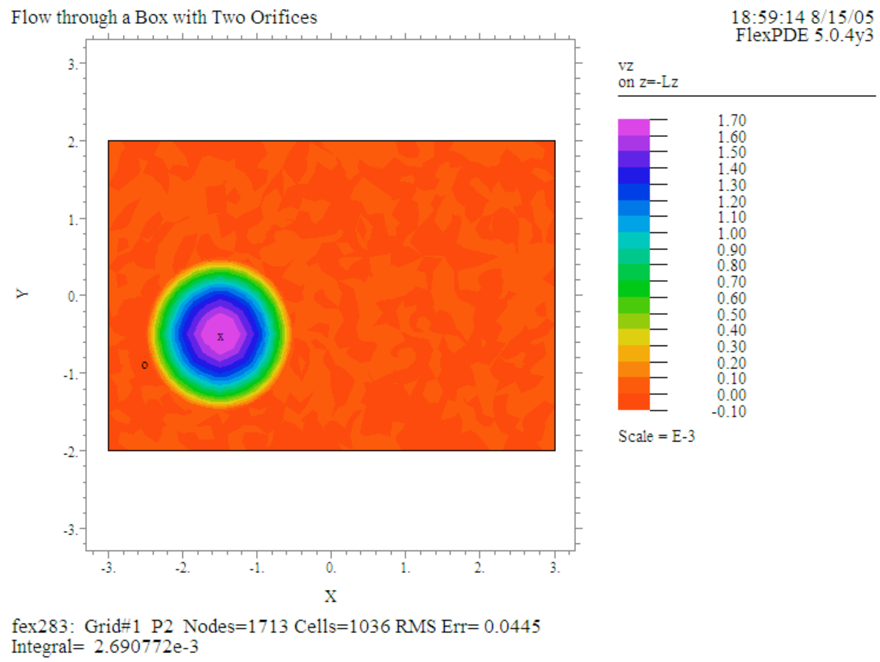
```
region 'box'                                          { Full solution domain }
  start 'outer' (-Lx,-Ly)
  value( vx)=0   value( vy)=0   value( vz)=0   natural( p)=natp
  line to (Lx,-Ly)  to (Lx,Ly)  to (-Lx,Ly)  to close
region 'in'
  surface 'bottom'   natural( vx)=0   natural( vy)=0   natural( vz)=0
    value( p)=delp
  start (-2.5,-0.5)  arc( center=-1.5,-0.5) angle=360 close
region 'out'
  surface 'top'   natural( vx)=0   natural( vy)=0   natural( vz)=0
    value( p)=0
  start (2.5,0.5)  arc( center=1.5,0.5) angle=360 close
MONITORS
  contour( p) painted on z=-Lz report(Re)
PLOTS
  contour( p) painted on z=-Lz report(Re)
  contour( vz) painted on z=-Lz     contour( vz) painted on z=Lz
  vector( v) norm  on y=0     vector( v) norm on y=2/3*x
  vector( v) norm on x=-2.0     vector( v) norm on x=2.0
END
```

The following figure shows the geometrical arrangement and the surface mesh.

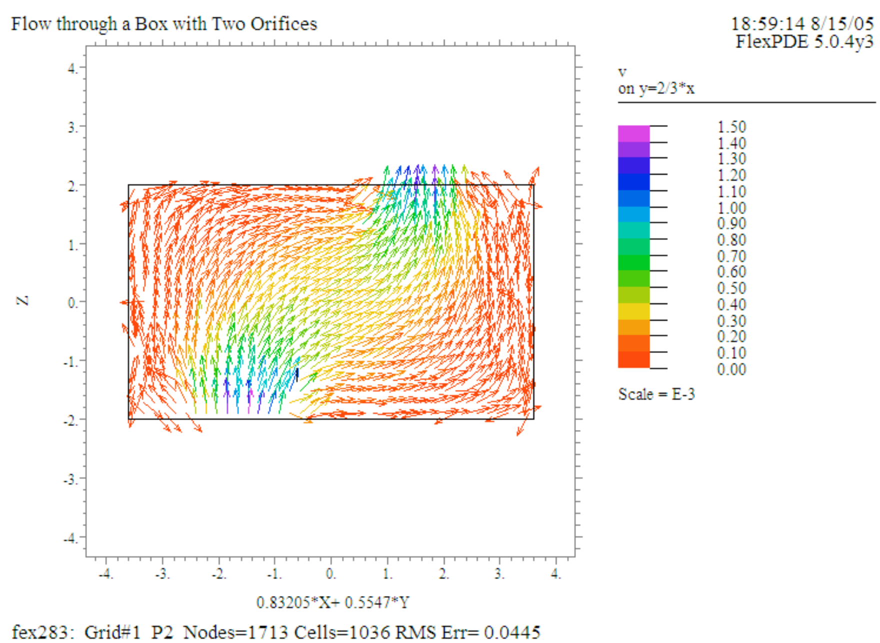


During the run, *monitors* show that the pressure is as intended over the left and right orifices. The monitor plots of vm on two perpendicular planes confirm that the magnitude of the velocity is zero on the boundaries.

The plot below shows the distribution of vz over the input orifice. The corresponding plot over the topside is similar.



The following vector plot illustrates the symmetry of this problem, further substantiated by the plot over the diagonal plane. There is no evidence for inertia, which is compatible with the very small value of Re.



Viscous Flow around a Cubical Obstacle

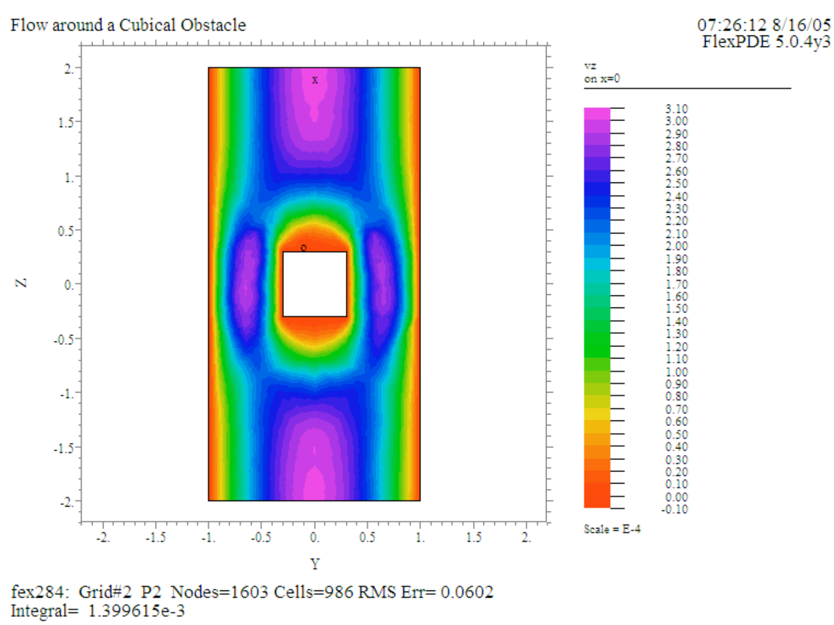
Here we shall consider viscous flow in a tube of circular cross-section containing a cube, partially blocking the stream. We use the tube radius r0 in Re and in the second term of the last PDE.

In this example, we have 3 liquid layers, cut by an extruded sub-region of square cross-section. This means that there are 6 distinct compartments, one of them constituting the cubical obstacle. We exclude the latter from the solution domain by the void declaration.

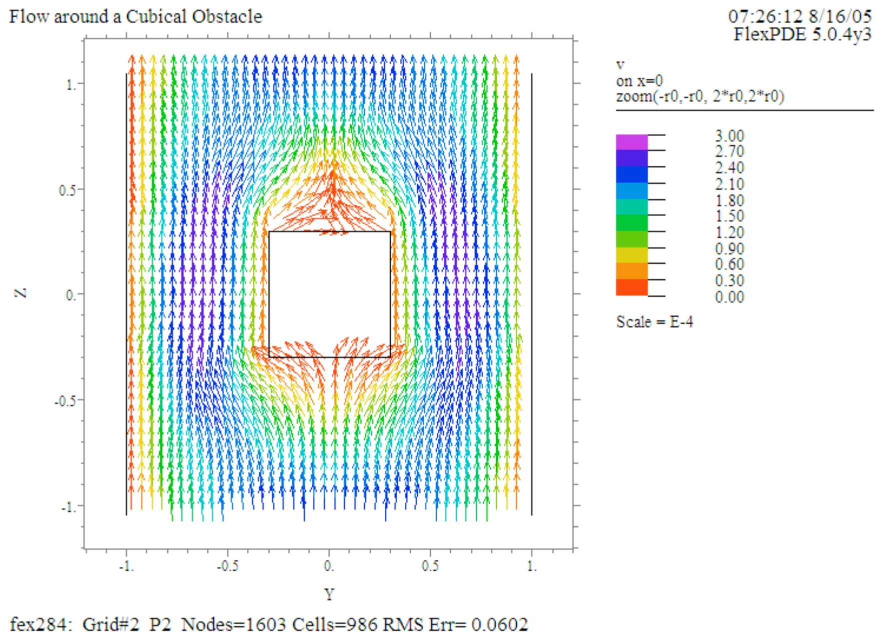
```
TITLE   'Flow around a Cubical Obstacle'                    { fex284.pde }
SELECT    errlim=1e-3    ngrid=4    spectral_colors
COORDINATES    cartesian3
VARIABLES    vx    vy    vz    p
DEFINITIONS
  r0=1.0    Lc=0.3    Lz=2.0    visc=1e4    delp=1e2
  dens=1e3    Re=dens*globalmax( vz)*2*r0/visc
  v=vector( vx, vy, vz)    vm=magnitude( v)
  unit_x=vector(1,0,0)    unit_y=vector(0,1,0)    unit_z=vector(0,0,1)
  nx=normal( unit_x)    ny=normal( unit_y)    nz=normal( unit_z)
  { Natural boundary condition for p }
  natp=0                                          { Simplified }
EQUATIONS                                         { For Re<<1 }
  vx:       dx( p)- visc*div( grad( vx))=0
  vy:       dy( p)- visc*div( grad( vy))=0
  vz:       dz( p)- visc*div( grad( vz))=0
  p:        div( grad( p))- 1e4*visc/r0^2*div( v)=0    { Tube radius r0 }
EXTRUSION                                         { Parallel surfaces }
  surface 'bottom' z=-Lz
  layer '1'                                       { Three liquid layers, 1-3 }
  surface 'low' z=-Lc
  layer '2'
  surface 'high' z=Lc
  layer '3'
  surface 'top' z=Lz
BOUNDARIES
  surface 'bottom'  natural(vx)=0   natural(vy)=0   natural(vz)=0
    value(p)=delp
  surface 'top'   natural(vx)=0   natural(vy)=0   natural(vz)=0
  value(p)=0
region 'domain'                         { Full solution domain }
  start (r0,0)   value( vx)=0   value( vy)=0   value( vz)=0
```

```
    natural( p)=natp
  arc( center=0,0) angle=360
region 'cube'
  surface 'low' value(vx)=0   value(vy)=0   value(vz)=0
     natural(p)=natp                    { Surfaces pertaining to cube }
  surface 'high'  value(vx)=0   value(vy)=0   value(vz)=0
     natural(p)=natp
  layer '2' void                        { Cube excluded from domain }
  start (-Lc,-Lc) layer '2'   value(vx)=0   value(vy)=0   value(vz)=0
  natural(p)=natp                       { These BCs limited to layer 2 }
  line to (Lc,-Lc) to (Lc,Lc) to (-Lc,Lc) to close
MONITORS
  contour( vm) painted on z=0     contour( vm) painted on x=0
PLOTS
  contour( p) painted on x=0 report(Re)
  contour( vz) painted on x=0     contour( vz) painted on y=0
  contour( vz) painted on z=-Lz     contour( vz) painted on z=0
  contour( vz) painted on z=Lz
  vector( v) norm on x=0 zoom(-r0,-r0, 2*r0,2*r0)
  elevation( p) from (0,0,-Lz) to (0,0,Lz)
  elevation( vz) from (0,0,-Lz) to (0,0,Lz)
END
```

In the contour plot of vz below we see the transverse and longitudinal surfaces limiting the cube. The velocity component vz evidently vanishes on the solid surfaces. The highest values occur around the axis and in an annular region around the cube.

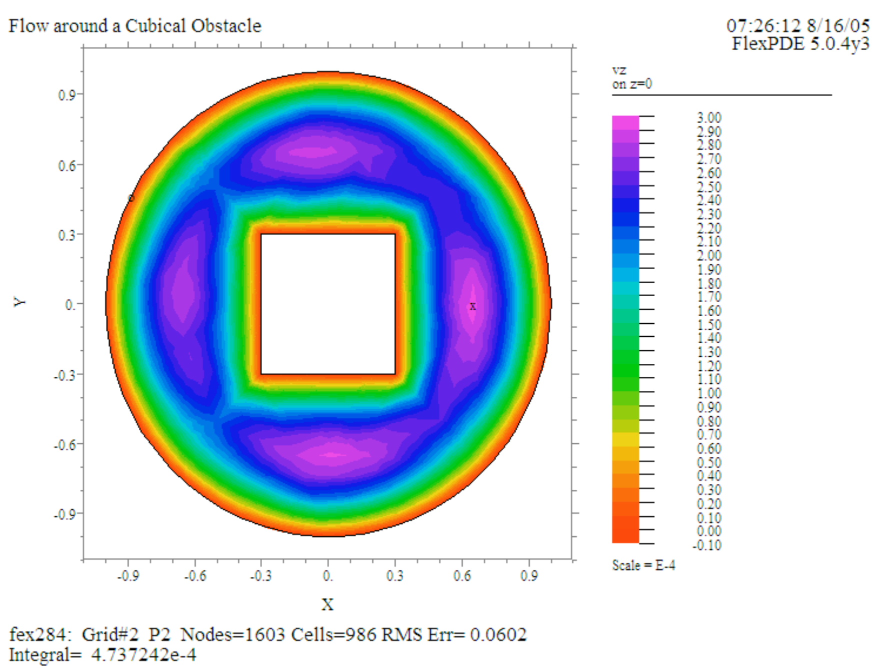


The next figure shows the direction of flow in the region close to the cube. It confirms that the speed vanishes on all the faces of the cube.



The three cross-sectional plots of vz show that the integrals have closely the same value, compatible with vanishing divergence of **v**.

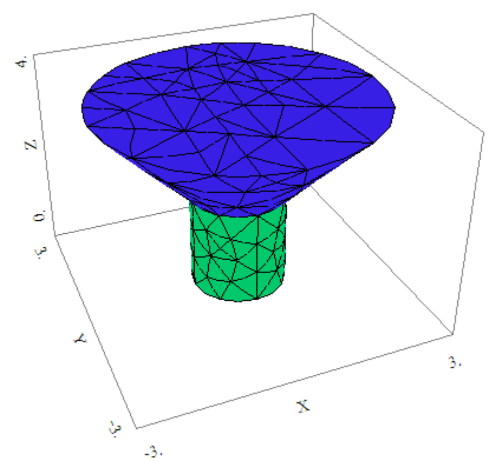
The plot below illustrates that the highest vz is in fact concentrated to four symmetric sites, approximately at mid-distance between the cube surfaces and the outer wall.



Viscous Flow by Gravity through a Funnel

In the next problem a liquid flows through a funnel under the influence of gravity and a downward driving pressure.

The extrusion scheme used in FlexPDE involves parallel extrusion, as from a tube of toothpaste. We obtain much more freedom if we define a volume *inside* the extruded volume and declare remaining parts as void. The final shape we want to achieve is illustrated in the figure below.



We create the region 'domain' by extruding a circle with radius r1, starting from the base plane, i.e. z=0. Thus, we initially obtain a volume limited by the planes '1' and '3' and a circular cylinder. It remains to *exclude* the parts outside the cone and the smaller cylinder below.

To generate the conical surface ('2'), we first define a set of points in (x, y) space by specifying that rad be larger than or equal to r0. Finally, we define this surface using zfun.

We thus have a cylindrical outer surface (r1) and two plane end surfaces. In addition there is a conical surface over part (L) of the length. Hence, there are two layers, one below the cone and one above. The inner cylinder (r0) we then create thus has two compartments of length L.

When the region 'domain' is first defined, it comprises the entire volume inside r1. When the region 'cylinder' is added, however, the 'domain' becomes redefined to mean everything outside the central cylinder of radius r0. If we thus specify the boundary conditions of surface '2' under 'domain', those boundary conditions become valid on the conical surface only.

In order to create empty space outside the funnel, we first declare the layer 'lower' (below the conical surface) to be void. The upper layer, on the other hand, is assigned the default viscosity value (visc).

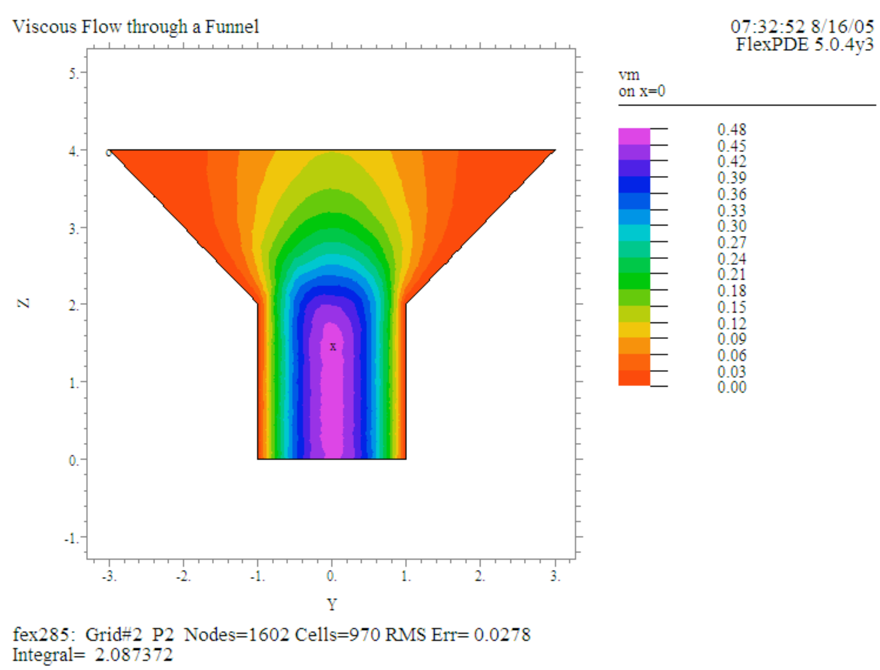
When we come to the cylinder of radius r0 the situation is simpler. Here, we assign the viscosity value visc to the lower layer. It is only in the layer 'lower', that the cylinder should have boundary conditions, which is the reason for specifying the name of the layer before imposing these conditions.

Gravitation produces a volume force Fz, which enters in the third PDE. We add a driving pressure delp on the boundary planes.

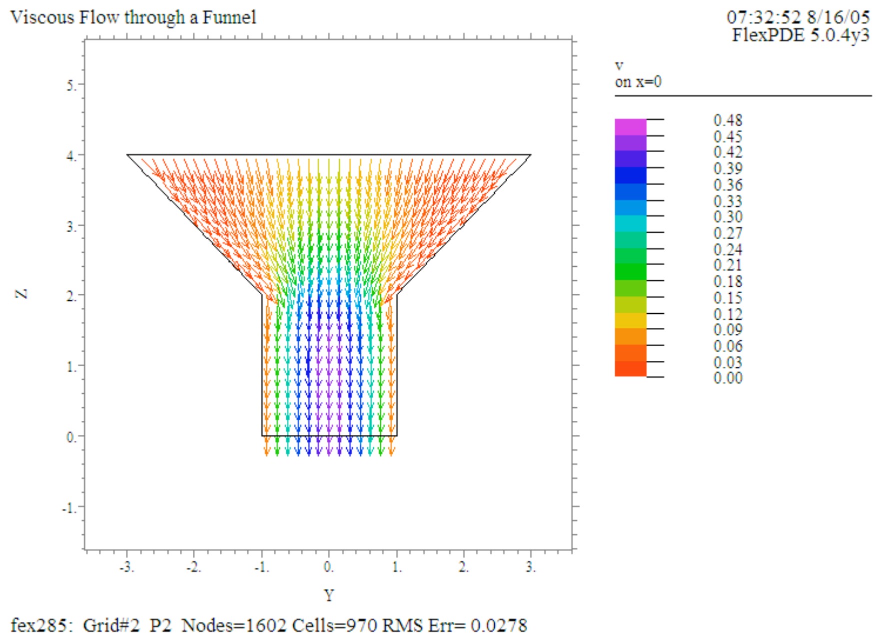
```
TITLE   'Viscous Flow through a Funnel'                         { fex285.pde }
SELECT     errlim=1e-3    ngrid=4     spectral_colors
COORDINATES     cartesian3
VARIABLES     vx     vy     vz     p
DEFINITIONS
  r0=1.0     r1=3.0     L=2.0     visc=1e4     delp=1e4
  dens=1e3     g=9.81     Fz=-dens*g           {Force due to gravity }
  Re=dens*globalmax( vz)*2*r0/visc
  v=vector( vx, vy, vz)     vm=magnitude( v)
  rad=max( r0, sqrt( x^2+y^2))
  zfun=L+L*(rad-r0)/(r1-r0)
  unit_x=vector(1,0,0)     unit_y=vector(0,1,0)     unit_z=vector(0,0,1)
  nx=normal( unit_x)     ny=normal( unit_y)     nz=normal( unit_z)
  natp=nz*Fz                                   { Simplified }
EQUATIONS                                      { For Re<<1 }
  vx:          dx( p)- visc*div( grad( vx))=0
  vy:          dy( p)- visc*div( grad( vy))=0
  vz:          dz( p)- Fz- visc*div( grad( vz))=0
  p:           div( grad( p))- 1e4*visc/r0^2*div( v)=0
EXTRUSION
  surface '1' z=0
  layer 'lower'
  surface '2' z=zfun
  layer 'upper'
  surface '3' z=2*L
BOUNDARIES
  surface '1'  natural(vx)=0   natural(vy)=0   natural(vz)=0   value( p)=0
  surface '3'  natural(vx)=0   natural(vy)=0  natural(vz)=0   value( p)=delp
region 'domain'                                { Outer region }
```

```
  layer 'lower'  void                              { Empty space }
  surface '2' value(vx)=0   value(vy)=0   value(vz)=0   natural(p)=natp
  layer 'upper'
  start 'outer' (r1,0)  arc( center=0,0) angle=360 close
limited region 'cylinder'                        { Limited to the lower layer }
  layer 'lower'                                  { Redefine void as visc }
  start (r0,0) layer 'lower'
  value(vx)=0   value(vy)=0   value(vz)=0   natural(p)=natp
  arc( center=0,0) angle=360 to close
MONITORS
  contour( vz) painted on x=0
PLOTS
  contour( vz) painted on x=0     contour( vm) painted on x=0
  vector( v) norm on x=0     contour( p) painted on x=0
  contour( vz) painted on z=0.01     contour( vz) painted on z=0.99*L
  contour( vz) painted on z=1.01*L     contour( vz) painted on z=1.95*L
END
```

The figure below demonstrates that the velocity in fact vanishes on the walls. The final contour plots verify that the flux is constant through various cross-sections.



The following vector plot of the velocity shows the expected flow pattern.

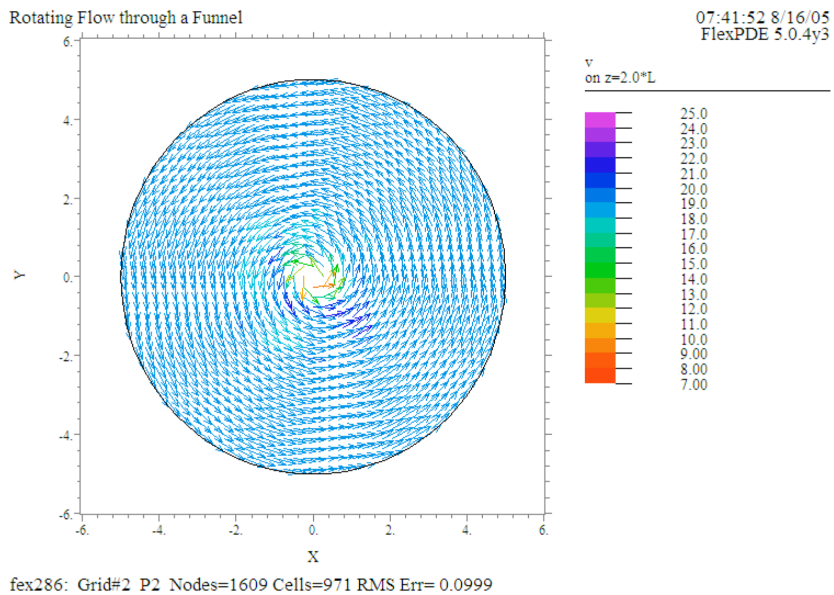


Rotating Flow through a Funnel

We now impose a rotating motion on the liquid at the top of the funnel. Most of the descriptor may be taken from the preceding example. For this example we require a larger number of nodes.

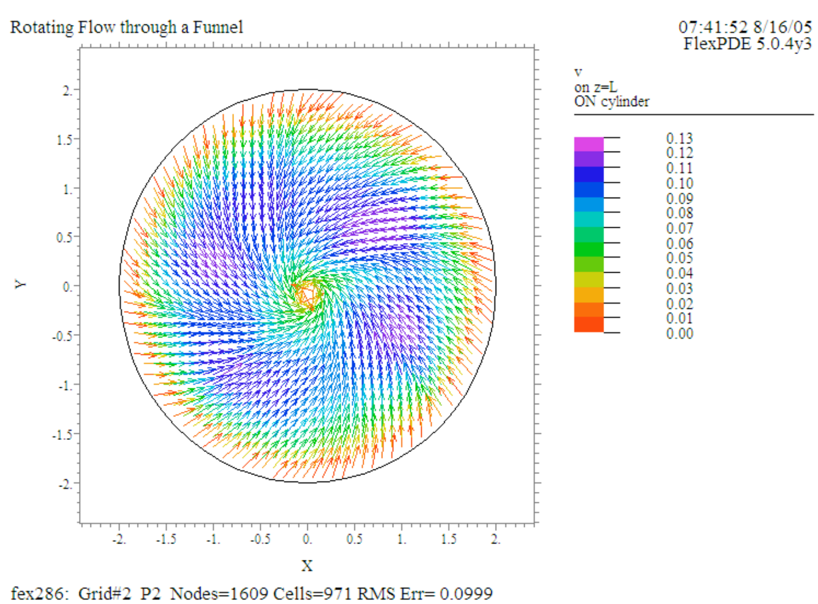
```
TITLE   'Rotating Flow through a Funnel'                          { fex286.pde }
SELECT     errlim=1e-3    ngrid=4     spectral_colors
…
  r0=2.0     r1=5.0     L=5.0     visc=1e4     delp=1e4     v0=20.0
…
BOUNDARIES
  surface '1'  natural(vx)=0   natural(vy)=0     natural(vz)=0
    value( p)=0
  surface '3'  value(vx)=-v0*y/sqrt(x^2+y^2)          { Rotation }
     value(vy)=v0*x/sqrt(x^2+y^2)  natural(vz)=0   value( p)=delp
region 'domain'
…
PLOTS
  contour( vz) painted on x=0     contour( vm) painted on x=0
  vector( v) norm on z=2.0*L
  vector( v) norm on z=1.5*L zoom(-4,-4, 8,8)
  vector( v) norm on z=L on 'cylinder'
END
```

At the top surface, the horizontal rotational speed is uniform at the impressed value (20.0), as confirmed by the following vector plot at z=2.0*L.



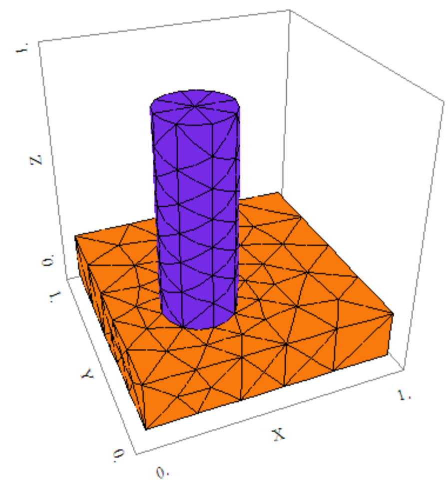
The next plot shows how the initial rotation is attenuated during the downward flow. The maximum vm in the plane z=1.5*L is now only 2.4, and the maximum occurs at an intermediate radius.

The figure below illustrates that the downward motion causes an inward spiral pattern. The velocity in the plane z=L is less than 1% of that impressed by rotation. The vertical velocity component dominates, as indicated by the contour plot.



Seeping through a Concrete Plate with a Pillar

The following example concerns percolation through a concrete plate, in contact with a pillar as shown in the first figure. The bottom is exposed to pressure and the top is open for outflow. The other boundaries are impermeable.



Here, we shall exploit the novel extension of the N-S equations to percolation that we already applied in 2D (p.320). The PDEs are trivial generalizations of those for 2D.

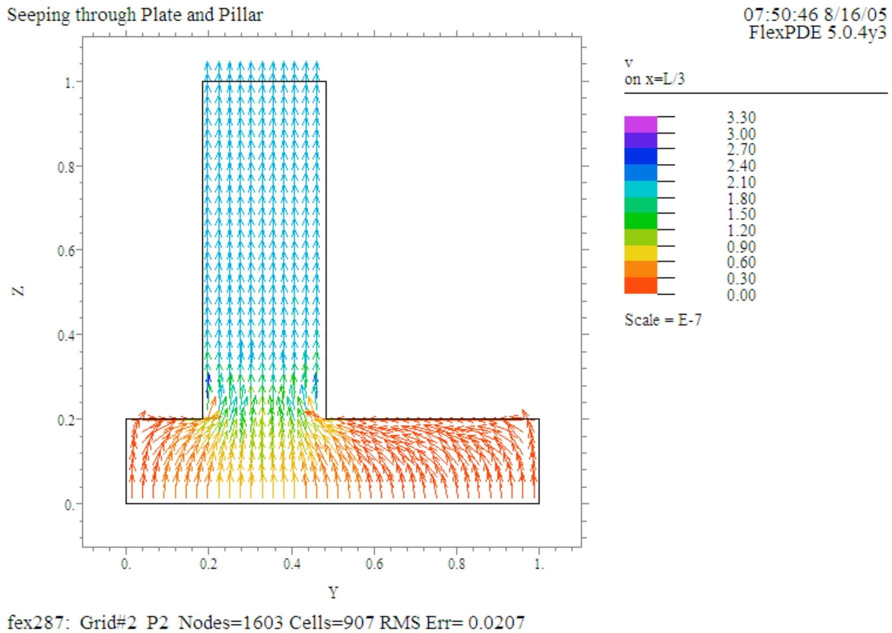
Under *extrusion* we first define the three parallel surfaces delimiting the components. Then we extrude a *cylinder* through a circle in the bottom plane. Region 'domain' thereby refers to the space outside the cylinder. Under 'domain', we thus declare the upper layer to be void, or empty space.

```
TITLE   'Seeping through Plate and Pillar'                    { fex287.pde }
SELECT     errlim=1e-3     ngrid=4     spectral_colors
COORDINATES     cartesian3                            { Student Version }
VARIABLES     vx     vy     vz     p
DEFINITIONS
  L=1.0     r0=0.15     z0=0.2
  visc=1e-3     k=1e-12     delp=1e4
  dens=1e3    Fgz=-dens*9.81
  v=vector( vx, vy, vz)     vm=magnitude( v)
  unit_x=vector(1,0,0)     unit_y=vector(0,1,0)     unit_z=vector(0,0,1)
  nx=normal( unit_x)     ny=normal( unit_y)     nz=normal( unit_z)
  natp=nz*Fgz- visc/k*( nx*vx+ ny*vy+ nz*vz)         { natural(p) }
EQUATIONS                                             { For Re<<1 }
```

```
  vx:        dx( p)+ visc/k*vx=0
  vy:        dy( p)+ visc/k*vy=0
  vz:        dz( p)- Fgz+ visc/k*vz=0
  p:         div( grad( p))- 1e4*visc/r0^2*div( v)=0
EXTRUSION                                   { Parallel surfaces }
  surface 'bottom' z=0
  layer 'plate'
  surface 'middle' z=z0                     { Interface }
  layer 'pillar'
  surface 'top' z=L
BOUNDARIES
  surface 'bottom'
  natural(vx)=0   natural(vy)=0   natural(vz)=0   value(p)=delp
  surface 'top'
  natural(vx)=0   natural(vy)=0   natural(vz)=0   value(p)=0
region 'domain'                             { Full solution domain }
  layer 'pillar'  void               { Exclude space outside pillar }
  surface 'middle'                          { Upper surface of plate }
  natural(vx)=0   natural(vy)=0   value(vz)=0   natural(p)=natp
  start 'outer' (0,0)     { Rectangle to be extruded, impermeable sides }
  value(vx)=0   value(vy)=0   natural(vz)=0   natural(p)=natp
  line to (L,0)  to (L,L) to (0,L)  to close
region 'cylinder'   start (L/3+r0,L/3)
  layer 'pillar'   value(vx)=0   value(vy)=0   natural(vz)=0   natural(p)=natp
  arc( center=L/3,L/3) angle=360 to close
MONITORS
  contour( vm) painted on x=L/3
PLOTS
  contour( vz) painted on z=z0/2     contour( vz) painted on z=L
  contour( vm) painted on x=L/3     vector( v) norm on x=L/3
  contour( p) painted on x=L/3
END
```

The plot below shows the velocity in a cross-section through the axis of the pillar.

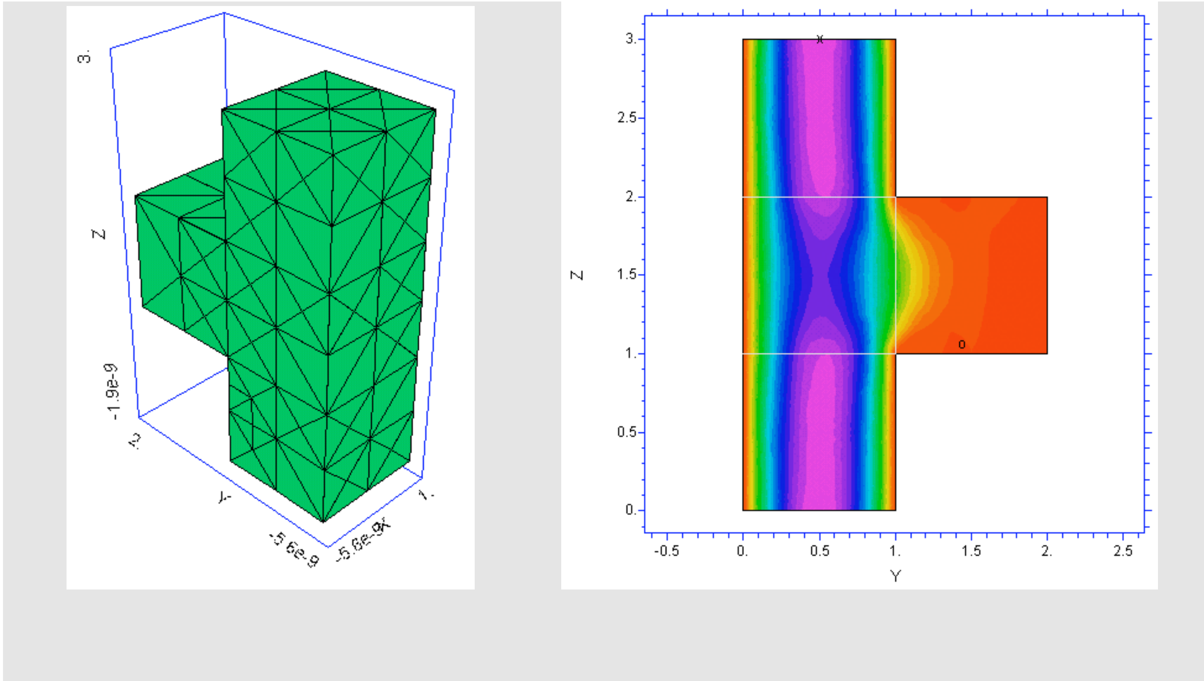
The speed fluctuations (vm) in the lower part of the pillar will gradually vanish as we use higher node numbers and the Professional Version.



Exercises

❑ Using *fex282* as a template, study the flow through a tube of radius r0=1.5 and uniform input velocity.
❑ In the preceding exercise, exploit the axial symmetry by solving over only one quarter of the tube.
❑ Try to introduce uniform input vz into *fex283*. Create a *feature*, concentric with the input orifice and with 20% smaller radius.
❑ Using *fex205* as a model, add two walls parallel to the (y,z) plane at a distance of 1.0 from each other (see figure below). The liquid hence enters from below through a square entrance, and the cavity is now cubic.

Hints: First create a rectangular box containing the entire structure and introduce the BCs appropriate to the vertical walls. Then introduce a *region* of square base for the cavity and declare the empty cubes on the left side as void. Specify wall BCs on the sides facing the void by *features* comprising single line segments in layers 1 and 3. Assign the remaining BCs on the cavity by *surface* statements in the appropriate region.



3.
Z
-1.9e-9
Y
-5.6e-9
-5.6e-9X
1.
3.
2.5
2.
1.5
1.
0.5
0.
-0.5
0.
0.5
1.
1.5
2.
2.5
Y
Z

29 Simplified PDEs for Viscous Flow

There is an alternative formulation of the PDEs for viscous flow, which does not require a supplementary equation for pressure[12]. The fundamental idea is to consider a liquid as being compressible, but in the limit of vanishing compressibility. In 2D, that concept leads to two equations only, which could be expected to shorten the solution time.

On reflection we realize that the flow itself may be thought of as *inducing* the pressure p. By imposing a velocity at the input boundary we create motion, and if the streaming liquid passes into a constriction, say, the ensuing force on it will produce local pressure. If the liquid is slightly compressible, this pressure should proportional to $-\nabla \cdot \mathbf{v}$. This pressure may hence be written

$$p = -c\nabla \cdot \mathbf{v} \qquad \bullet$$

where c is a constant to be found by trial and error. It should be large enough to make $\nabla \cdot \mathbf{v}$ nearly vanish. In practice we make it proportional to viscosity, in order to obtain the correct dimension. The value chosen for c will then be valid for any liquid.

The general N-S equation is (p.253)

$$\rho_0 \frac{\partial}{\partial t}\begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix} + \rho_0(\mathbf{v}\cdot\nabla)\mathbf{v} - \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} + \begin{Bmatrix} \partial p/\partial x \\ \partial p/\partial y \\ \partial p/\partial z \end{Bmatrix} - \eta\begin{Bmatrix} \nabla^2 v_x \\ \nabla^2 v_y \\ \nabla^2 v_z \end{Bmatrix} = 0 \qquad \bullet$$

where p now is to be copied from the above expression. We shall expand the second term when we need it.

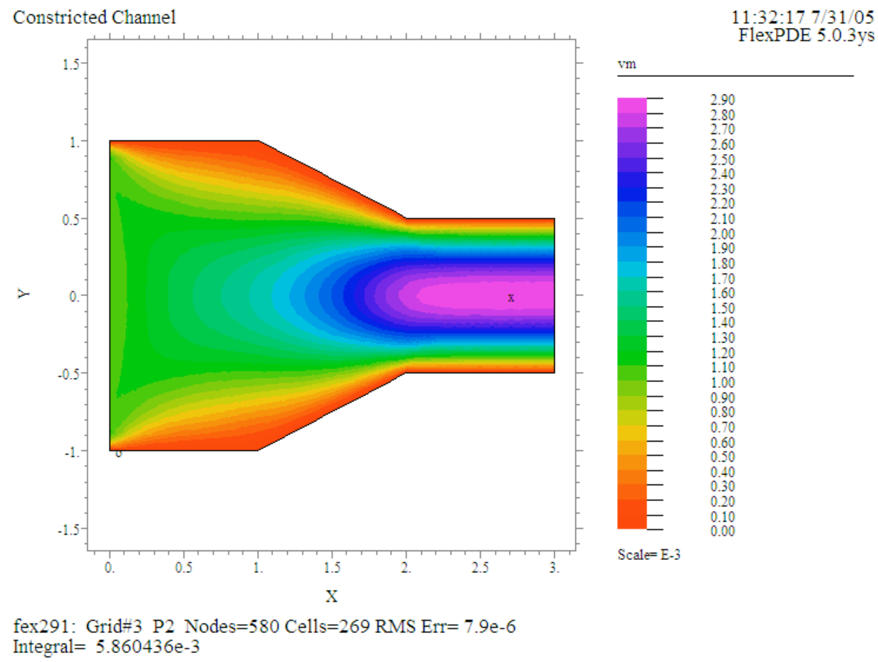
The first few examples concern steady flow at small Re, which means that the two first terms vanish.

Steady Flow in a Constricted Channel at Re<<1

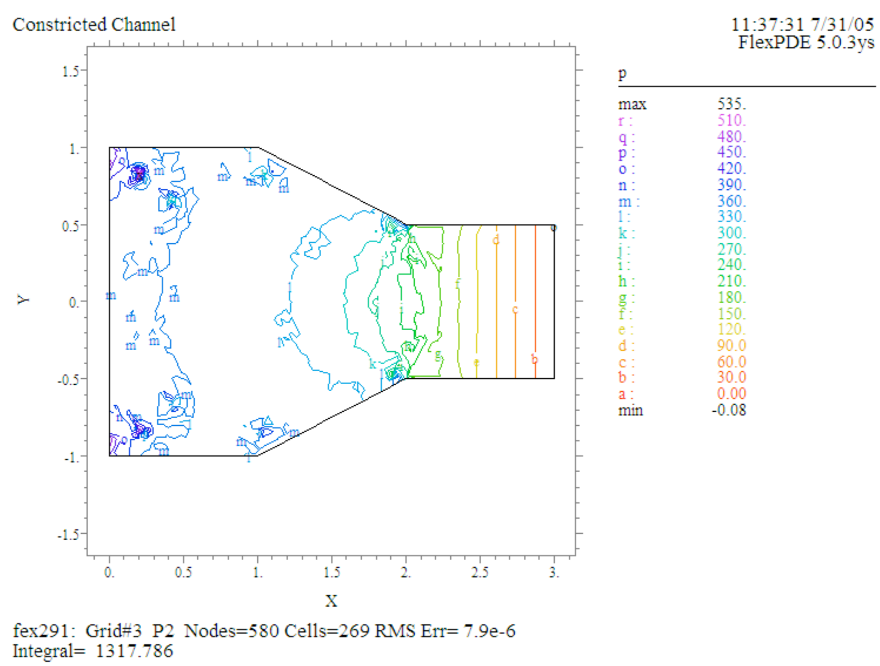
The following is a simplification of *fex203a*. On the input and output faces we specify $\partial v_y / \partial x = 0$, assuming negligible change in v_y close to the ends, and we proceed similarly for v_x at the exit.

```
TITLE   'Constricted Channel'                                   { fex291.pde }
SELECT     errlim=1e-5     ngrid=4     spectral_colors
VARIABLES     vx     vy                                   { Student Version }
DEFINITIONS
  L=1.0     coef=0.5     visc=1e4
  vx0=1e-3                                                  { Input velocity }
  dens=1e3     Re=dens*vx0*2*L/visc
  v=vector( vx, vy)     vm=magnitude( v)
  c=1e4*visc                           { Dimension of viscosity required }
  p=-c*div(v)                          { To be substituted into the PDEs }
EQUATIONS
  vx:          dx( p)- visc*div( grad( vx))=0                 { 2D and F=0 }
  vy:          dy( p)- visc*div( grad( vy))=0
BOUNDARIES
region 'domain'  start 'outer' (0,L)
  value( vx)=vx0   natural( vy)=0 line to (0,-L)              { In }
  value( vx)=0   value( vy)=0                                 { Wall }
  line to (L,-L) to (2*L,-L*coef) to (3*L,-L*coef)            { Wall }
  natural( vx)=0   natural( vy)=0                             { Out }
  line  to (3*L,L*coef)  value( vx)=0   value( vy)=0
  line to (2*L,L*coef) to (L,L) to close                      { Wall }
PLOTS
  vector( v) norm report(Re)     contour( vm) painted
  contour( div( v))     contour( curl( v)) painted
  contour( p)
  contour( p) painted  zoom( 1.5*L,-L, 2*L,2*L)
  elevation( vx) from (0,-L) to (0,L)
  elevation( vx) from (L,-L) to (L,L)
  elevation( vx) from (2*L,-L*coef) to (2*L,L*coef)
  elevation( vx) from (3*L,-L*coef) to (3*L,L*coef)
  elevation( dy( vx)) from (3*L,-L*coef) to (3*L,L*coef)
END
```

The run time is about the same as for *fex203a*, but the maximum error obtained is much smaller. The plot of vm (below) demonstrates that the speed indeed vanishes on the walls.



The plot of pressure, however, is very erratic in this case with high spots that only suggest the regular variation we found in *fex203a*.



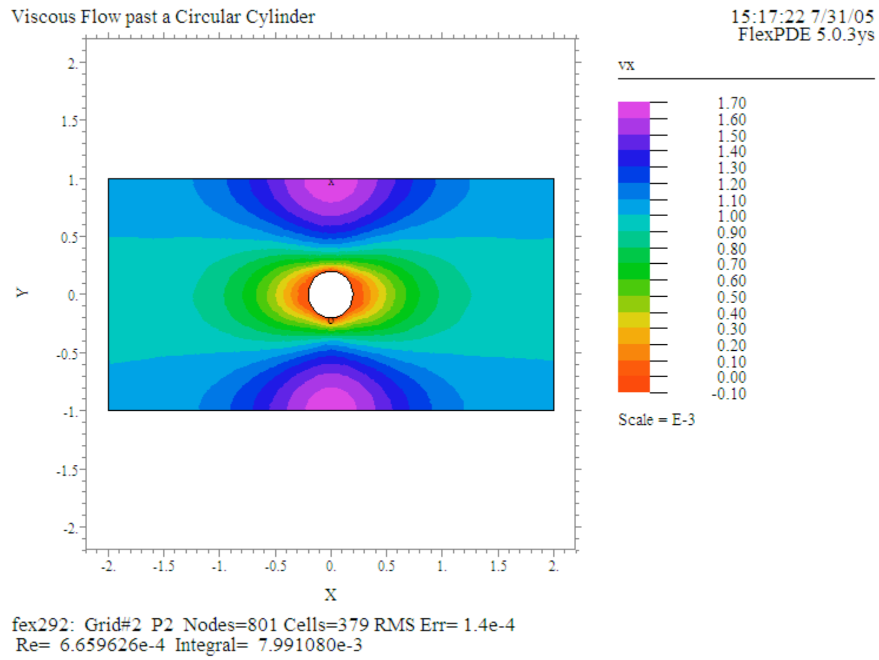
The elevation plots of vx across the channel show that the initially flat velocity distribution gradually changes to parabolic. The flux values reported at the bottom of each figure are closely the same, which indicates that mass and volume are conserved.

Flow past a Circular Cylinder at Re<<1

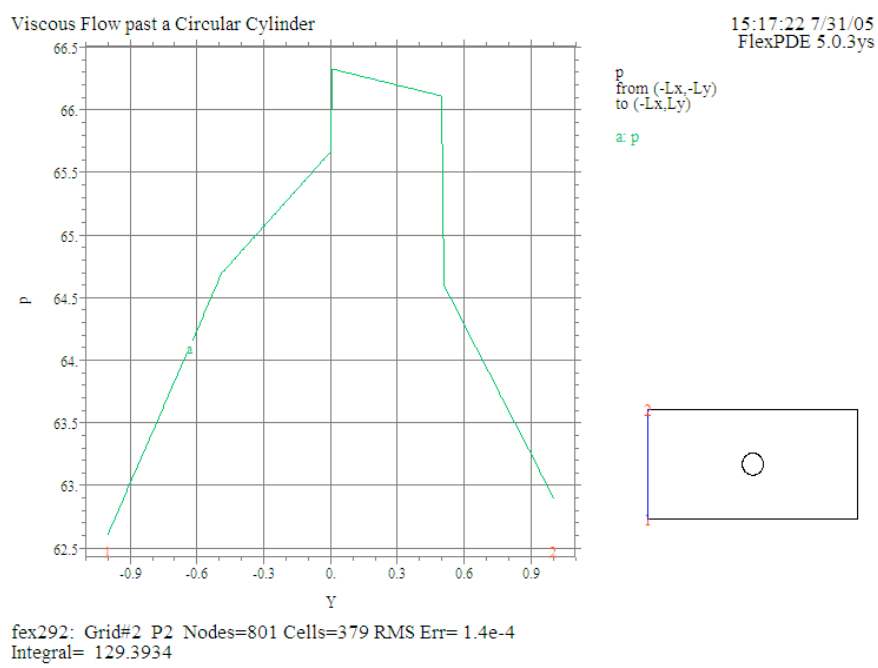
Here, we revisit the example *fex211b*, using *slip* conditions on the outer boundaries. We also integrate over the entrance to explore whether the pressure can be used to estimate the force on the obstacle.

```
TITLE    'Viscous Flow past a Circular Cylinder'           { fex292.pde }
SELECT    errlim=1e-5    ngrid=4    spectral_colors
VARIABLES    vx     vy
DEFINITIONS
  Lx=2.0    Ly=1.0    a=0.2    visc=1e4    vx0=1e-3
  dens=1e3    Re=dens*globalmax( vx)*2*Lx/visc
  v=vector( vx, vy)    vm=magnitude( v)
  c=1e4*visc    p=-c*div( v)
EQUATIONS                                                    { F=0 }
  vx:        dx( p)- visc*div( grad( vx))=0
  vy:        dy( p)- visc*div( grad( vy))=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
  value( vx)=vx0   natural( vy)=0                            { In }
  line to (-Lx,-Ly)  natural( vx)=0   value( vy)=0           { Wall}
  line to (Lx,-Ly)   natural( vx)=0   natural( vy)=0         { Out }
  line to (Lx,Ly)   natural( vx)=0   value( vy)=0   line to close   { Wall }
  start 'cylinder' (a,0)                               { Exclude }
  value( vx)=0  value( vy)=0   arc( center=0,0) angle=360 close
PLOTS
  contour( vx) painted report( Re)    contour( vy) painted
  elevation( vx) from (-Lx,-Ly) to (-Lx,Ly)
  elevation( vx) from (0,-Ly) to (0,Ly)
  elevation( vx) from (Lx,-Ly) to (Lx,Ly)    contour( vm) painted
  vector( v) norm    vector(v) norm  zoom(-2*a,-2*a, 4*a,4*a)
  contour( curl( v)) painted    contour( p) painted
  contour( p) painted zoom(-2*a,-2*a, 4*a,4*a)
  elevation( p) from (-Lx,-Ly) to (-Lx,Ly)        { ⇨ Force on liquid }
  elevation( p) from (Lx,-Ly) to (Lx,Ly)
END
```

The following plot shows that the uniform distribution of vx at the entrance again becomes essentially uniform near the exit. The elevation plots over the cross-sections indicate that mass is conserved.



We have already seen that the present formalism yields only scanty information about pressure. This impression is further illustrated by the zoomed contour plot of p. The pressure over the entrance also exhibits large scatter, as the plot below demonstrates.



The above plot suggests that it should be possible to trust the force value, obtained by integration over the entrance, to within a few percent.

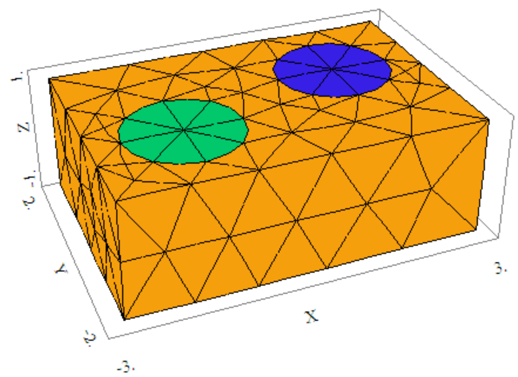
Flow through a Box with Two Orifices (3D)

Let us now use the simplified formalism with an example similar to *fex283*, extending the PDEs to three dimensions. Here, it is not possible to apply uniform pressure over the entrance as before. It is also difficult to obtain uniform vz over the orifice with a reasonable number of node points. What we can do, however, is to specify a paraboloidal distribution of vz over the entrance.

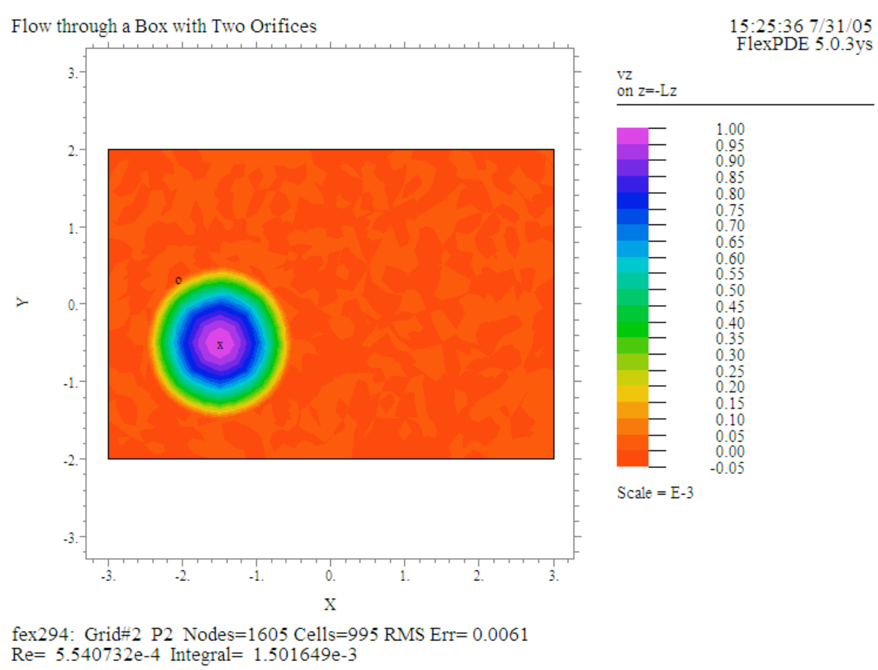
```
TITLE    'Flow through a Box with Two Orifices'                { fex293.pde }
SELECT     errlim=1e-4    ngrid=4     spectral_colors
COORDINATES     cartesian3
VARIABLES     vx     vy     vz
DEFINITIONS
  Lx=3.0     Ly=2     Lz=1.0     r0=1.0     visc=1e4     vz0=1e-3
  r_in=sqrt( (x+1.5)^2+ (y+0.5)^2)
  dens=1e3     Re=dens*globalmax( vz)*2*Ly/visc
  v=vector( vx, vy, vz)     vm=magnitude( v)
  c=1e4*visc     p=-c*div( v)
EQUATIONS
  vx:          dx( p)- visc*div( grad( vx))=0
  vy:          dy( p)- visc*div( grad( vy))=0
  vz:          dz( p)- visc*div( grad( vz))=0
EXTRUSION                                          { Parallel surfaces }
  surface 'bottom' z=-Lz
  layer 'liquid'                                   { Layer containing liquid }
  surface 'top' z=Lz
BOUNDARIES
  surface 'bottom'   value( vx)=0   value( vy)=0   value( vz)=0
  surface 'top'   value( vx)=0   value( vy)=0   value( vz)=0
region 'box'                                       { Full solution domain }
  start 'outer' (-Lx,-Ly)
  value( vx)=0   value( vy)=0   value( vz)=0
  line to (Lx,-Ly)  to (Lx,Ly)  to (-Lx,Ly)  to close
region 'in'
  surface 'bottom'   natural( vx)=0   natural( vy)=0
  value( vz)=vz0*(1- (r_in/r0)^2)                  { Paraboloid }
start (-2.5,-0.5)  arc( center=-1.5,-0.5) angle=360
region 'out'
  surface 'top'   natural( vx)=0   natural( vy)=0   natural( vz)=0
start (2.5,0.5)  arc( center=1.5,0.5) angle=360
MONITORS
```

```
  contour( vz) painted on z=-Lz     contour( vz) painted on z=Lz
  contour( vm) painted on z=0   contour( vm) painted on x=0
PLOTS
  contour( vz) painted on z=-Lz report(Re)
  contour( vz) painted on z=Lz    contour( vm) painted on z=0
  vector( v) norm  on y=0
  vector( v) norm on x=-2.0     vector( v) norm on x=2.0
  vector( v) norm on y=x/2                    { Through centers of orifices }
END
```

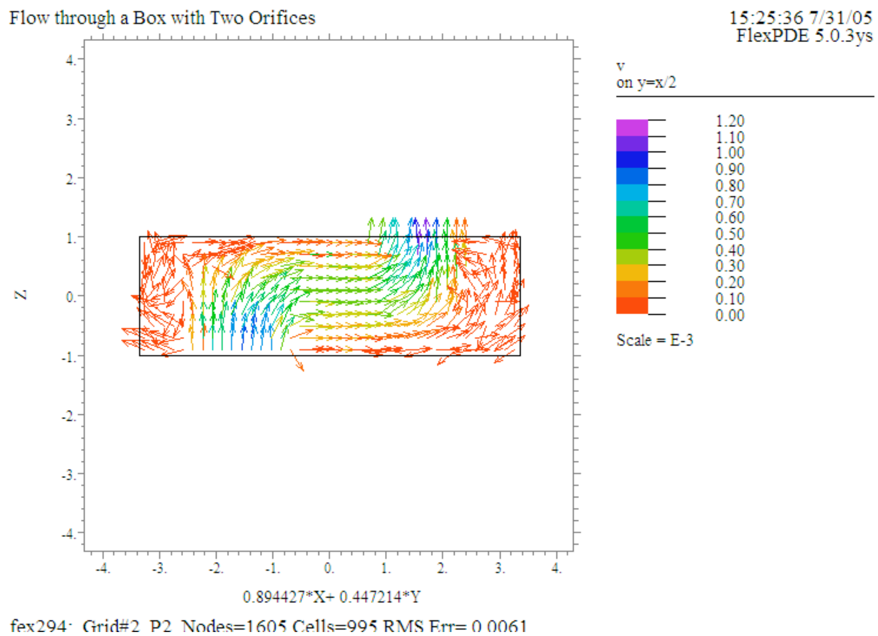
In the following miniature plot the regions are indicated by colors.



The next figure is a plot of vz over the entrance plane, the integral yielding the flux, to be compared to the value at the exit.



The next plot shows the flow in a plane going through the centers of both the entrance and the exit. The symmetry is evident.



Steady Viscous Flow at Re>>1

In order to proceed to higher speeds, we have to include the second term in the PDE (p.328). In 3D it becomes

$$\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v}=\rho_0\left(v_x\frac{\partial}{\partial x}+v_y\frac{\partial}{\partial y}+v_z\frac{\partial}{\partial z}\right)\begin{Bmatrix}v_x\\ v_y\\ v_z\end{Bmatrix}=$$

$$\rho_0\begin{Bmatrix}v_x\dfrac{\partial v_x}{\partial x}+v_y\dfrac{\partial v_x}{\partial y}+v_z\dfrac{\partial v_x}{\partial z}\\ v_x\dfrac{\partial v_y}{\partial x}+v_y\dfrac{\partial v_y}{\partial y}+v_z\dfrac{\partial v_y}{\partial z}\\ v_x\dfrac{\partial v_z}{\partial x}+v_y\dfrac{\partial v_z}{\partial y}+v_z\dfrac{\partial v_z}{\partial z}\end{Bmatrix}$$

Flow past a Circular Cylinder at Re>>1

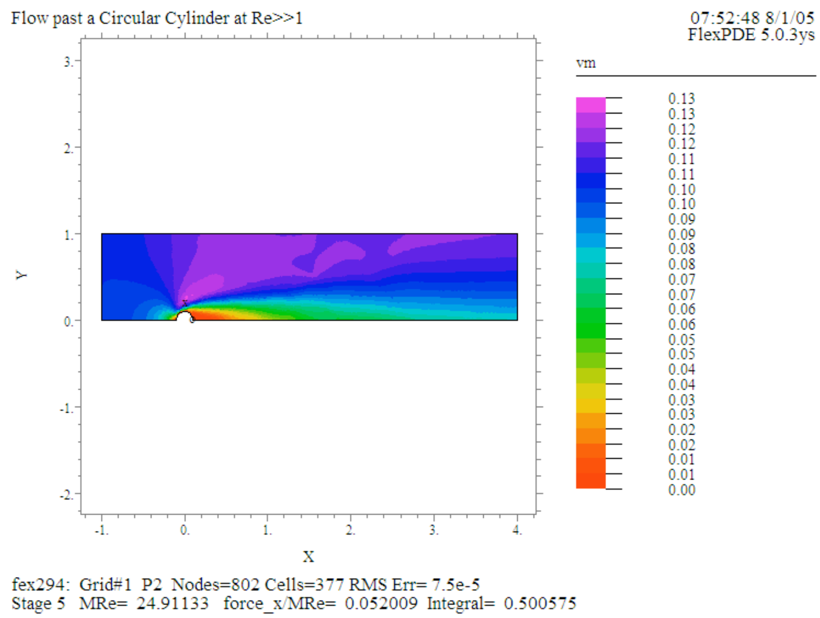
We shall now revisit the example *fex252*. As before, we let the liquid *slip* on the wall.

In view of the fact that the definition of Re is somewhat arbitrary in this case, we use a modified reference value, MRe, which relates to the size of the obstacle.

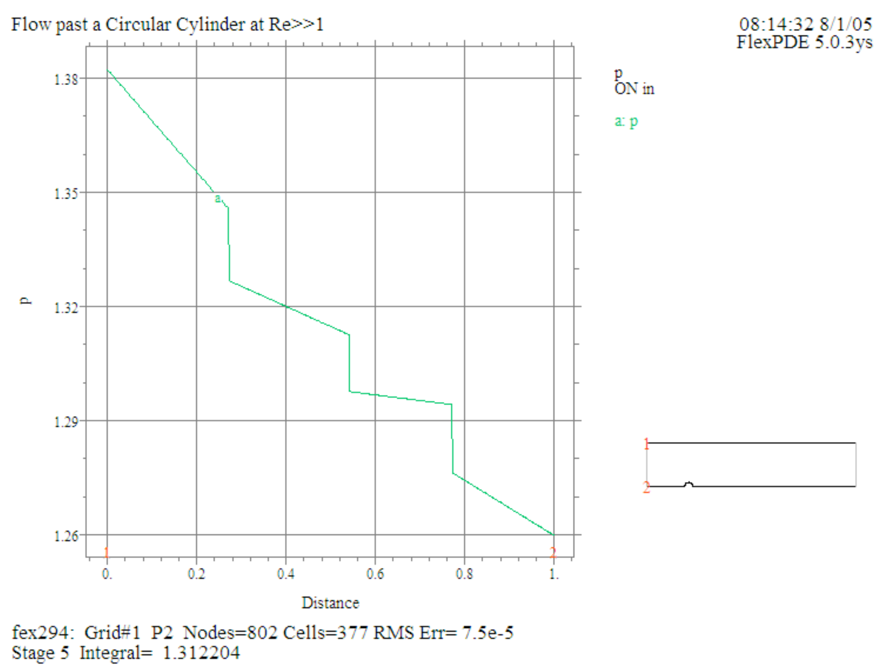
We attempt to calculate the driving pressure force_x by two line integrals over the ends.

```
TITLE    'Flow past a Circular Cylinder at Re>>1'            { fex294.pde }
SELECT    errlim=1e-5    ngrid=4    stages=5    spectral_colors
VARIABLES    vx    vy
DEFINITIONS
  Lx=1.0    Ly=1.0    r0=0.1    visc=1.0
  vx0=staged( 1e-5, 1e-4, 0.01, 0.03, 0.1)              { Input vx }
  dens=1e3    MRe=dens*globalmax( vx)*2*r0/visc    { Modified Re }
  v=vector( vx, vy)    vm=magnitude( v)
  vxdvx=vx*dx(vx)+ vy*dy(vx)    vxdvy=vx*dx(vy)+ vy*dy(vy)
  c=1e4*visc    p=-c*div( v)
  force_x=line_integral( p,'in')- line_integral( p,'out')
EQUATIONS
  vx:        dens*vxdvx+dx( p)- visc*div( grad( vx))=0
  vy:        dens*vxdvy+ dy( p)- visc*div( grad( vy))=0
BOUNDARIES
region 'domain'  start 'outer' (-Lx,Ly)
  value( vx)=vx0  natural( vy)=0                          { In }
  line to (-Lx,0)   natural( vx)=0   value( vy)=0         { Symmetry }
  line to (-r0,0)   value(vx)=0   value(vy)=0
    arc( center=0,0) angle=-180 to (r0,0)                 { Cylinder }
  natural( vx)=0   value( vy)=0                           { Symmetry }
  line to (4*Lx,0)   natural( vx)=0   natural( vy)=0      { Out }
  line to (4*Lx,Ly)   natural( vx)=0   value( vy)=0       { Slip on wall }
  line to close
feature
  start 'in' (-Lx,0) line to (-Lx,Ly)    start 'out' (4*Lx,0) line to (4*Lx,Ly)
PLOTS
  contour( vm) painted report( MRe) report( force_x/MRe)
  contour( vx/vx0) report( MRe)
  vector( v) norm    vector( v) norm zoom(0,0, 5*r0,5*r0) report( MRe)
  elevation( p) on 'in'    elevation( p) on 'out'    contour( p) painted
END
```

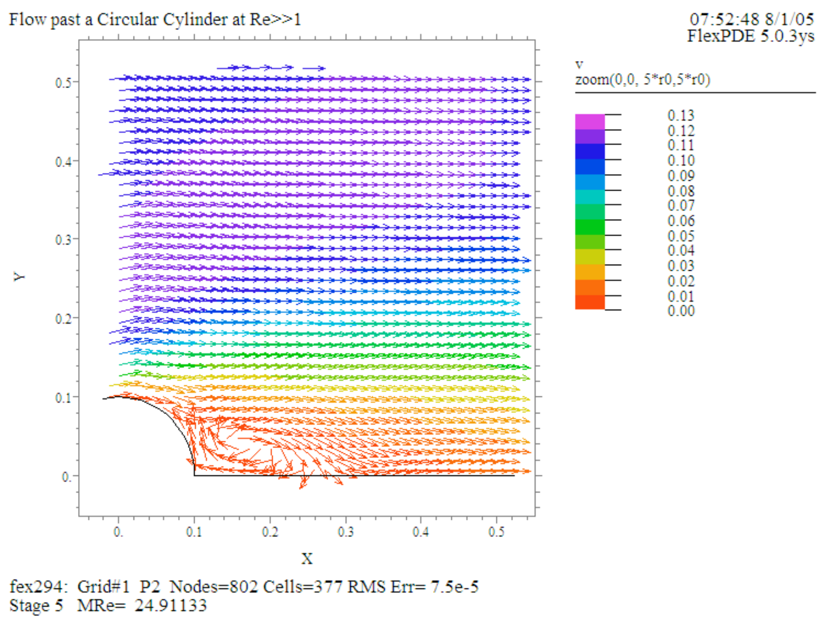
The following plot shows the speed distribution and reports the ratio of the force on the liquid (and the cylinder) to the value of MRe. As we have noted before, this ratio rises markedly above MRe>1.



The following plot of the pressure distribution over the entrance is rather ragged, but it should yield an integral value reliable within a few percent.



Using *File,View* we may inspect the flow pattern as the speed increases. At MRe≅25 we find the vector plot below, which is very similar to that obtained before by a different formalism (p.334).



In summary, this example in its two versions provides a convincing demonstration of the reliability of the formalisms used and of the FlexPDE software.

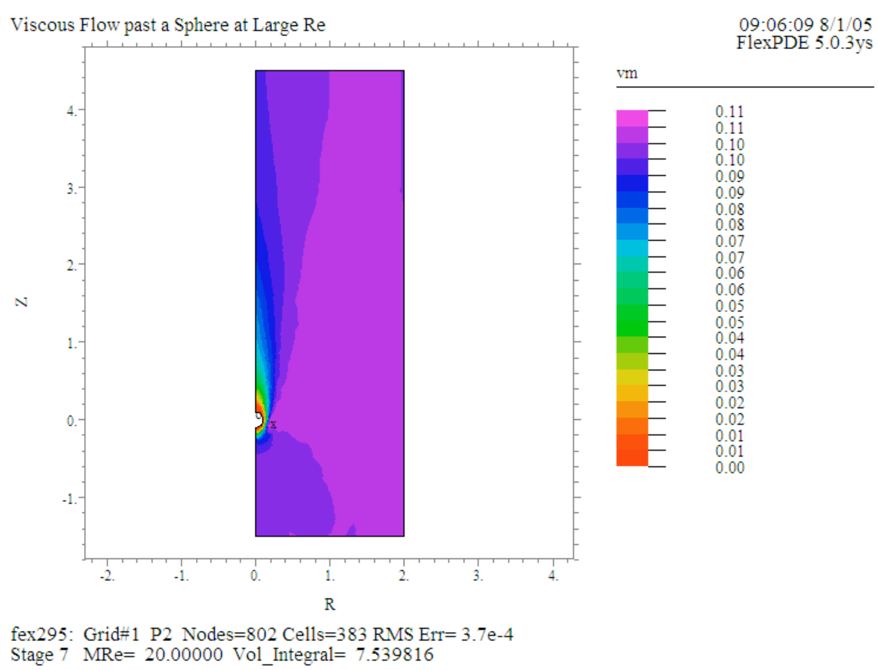
Flow in (ρ,z) past a Sphere at Re>>1

Let us now extend the simplified PDEs to a case of axial symmetry, repeating the calculations in *fex263*. Only a few modifications are required.

```
TITLE     'Viscous Flow past a Sphere at Large Re'          { fex295.pde }
SELECT    errlim=1e-5    ngrid=1    stages=7    spectral_colors
COORDINATES    ycylinder('r','z')
VARIABLES    vr    vz
DEFINITIONS
  L=1.5    r1=2.0    r0= 0.1
  visc=1.0    dens=1e3
  vz0=staged( 1e-6, 3e-3, 0.01, 0.02, 0.04, 0.07, 0.1)    { Input values }
  MRe=dens*vz0*2*r0/ visc                                 { Modified Re }
  v=vector( vr, vz)    vm=magnitude( v)
  vrdvr=vr*dr(vr)+ vz*dz(vr)    vrdvz=vr*dr(vz)+ vz*dz(vz)
```

```
  curl_phi=dz(vr)-dr(vz)
  drag_S=6*pi*visc*r0*vz0          { After Stokes for small MRe }
  c=1e4*visc    p=-c*div( v)
EQUATIONS
  vr:          dens*vrdvr+ dr(p)- visc*[ 1/r*dr(r*dr(vr))- vr/r^2+ dzz(vr)]=0
  vz:          dens*vrdvz+ dz(p)- visc*[ 1/r*dr(r*dr(vz))+ dzz(vz)]=0
BOUNDARIES
region 'domain' start(0,-L)
  natural(vr)=0   value(vz)=vz0   line to (r1,-L)        { In }
  value(vr)=0   natural(vz)=0   line to (r1,3*L)   { Wall }
  natural(vr)=0   natural(vz)=0   line to (0,3*L)        { Out }
  value(vr)=0   natural(vz)=0   line to (0,r0)              { Axis }
  value(vr)=0   value(vz)=0
    arc( center=0,0) angle=-180
  value(vr)=0   natural(vz)=0   line to close
PLOTS
  contour( vz)     contour( vm) painted report( MRe)
  contour( p) painted     vector( v) norm
  contour( div( v))     contour( curl_phi) painted
  elevation( p/drag_S) from (0,-L) to (r1,-L) report(MRe)    { Force_z }
END
```

The following plot shows the resulting speed distribution, which is again similar to what we found before, using three PDEs (p.353).



Because of the poor pressure data, the drag force differs considerably from the Stokes expression at the smallest values of MRe, the integrated ratio p/drag_S being as small at 0.13 in the first stage. Above stage 3, however, it is still possible to recognize the variation versus MRe that we recorded before.

Exercises

❑ Repeat *fex292* with zero-speed boundary conditions on the walls.

❑ Solve the problem in *fex204* using the simplified PDEs. Specify suitable vx0 and errlim. Try modifying the geometrical parameters and the viscosity.

❑ Solve the problem in *fex212* using the simplified PDEs. Replace *visc_xy* by its first line only.

❑ Solve the problem in *fex285* by the simplified PDEs.

References

As an example of the notation in this volume, the superscript^{3p55} means reference 3, page 55.

[1] Kreyszig, E. *Advanced Engineering Mathematics, 3rd ed.*, John Wiley and Sons, 1972.
[2] Crandall, S. E., Dahl, N. C. and Lardner, T. J. *An Introduction to the Mechanics of Solids, 2nd ed.*, McGraw-Hill, 1978.
[3] Timoshenko, S. P. and Goodier, J. N. *Theory of Elasticity, 3rd ed.*, McGraw-Hill, 1970 .
[4] Lipson, C. and Juvinall, R. *Handbook of Stress and Strength*, Macmillan, 1963.
[5] Carslaw, H. S. and Jaeger, J. C. *Conduction of Heat in Solids, 2nd edn*, Clarendon, 1959.
[6] Arfken, G. *Mathematical Methods for Physicists*, Academic Press, 1968
[7] Bickford, W. B. *A First Course in the Finite Element Method*, Richard D. Irwin, 1990.
[8] Tritton, D. J. *Physical Fluid Dynamics, 2nd ed.*, Oxford University Press, 1988
[9] Bachelor, G. K. *An Introduction to Fluid Dynamics*, Cambridge University Text, 1970
[10] Panton, R. L. *Incompressible Flow*, John Wiley, 1996.
[11] Gresho, P. M. and Sani, R. L. *International J. for Numerical Methods in Fluids*, Vol. 7, 1111-1145 (1987)
[12] Hughes, T., Liu, W., and Brooks, A. *J. Comput. Phys.* Vol. 30, 1979, pp.1-60.

Vocabulary of FlexPDE

The following table is a reminder of the syntax rules, given as descriptor fragments. Commands in color pertain to 3D. The numbers refer to pages in *Deformation and Vibration* and in this book, where the usage has been illustrated by examples. More details are available under *Help* while using the program.

		pages
SELECT		
spectral_colors		10, 228
errlim=1e-5 nodelimit=400		228, 330
ngrid=1 stages=2		228, 302
COORDINATES		
ycylinder('r','z')	{ Default: (x,y) }	291
cartesian3	{ x,y,z }	370
VARIABLES		
U		36
DEFINITIONS	{ SI units }	
v=vector(vx,vy) vm=magnitude(v)		18, 231
globalmax(vx)		260
natp= if stage=1 then 0 else …		303
#include 'visc_xy.pde'		282
unit_x=vector(1,0)		245
INITIAL VALUES		
vx=0		356
EQUATIONS		
div(grad(phi))=0		231
vx: dx(p)- visc*div(grad(vx))=0	{ Tagged }	257
CONSTRAINTS	{ Integral relations only }	

EXTRUSION { 3D only }
surface 'bottom' z=0 370
layer 'liquid' 370

BOUNDARIES { Drawn counterclock-wise }
region 'domain' start 'outer' (0,Ly) … to close 7, 228
start (r1,0) arc to (0,r1) to (-r1,0) to (0,-r1) close 18
start 'obstacle' (a,0) … arc(center=0,0) angle=360 231
value(phi)=0 natural(phi)=0 228
layer '2' void 378
limited region 'cylinder' 382
feature { Curve defined like domain, but without close } 243

TIME { For time-dependent problems }
from 0 to 5e-2 356

MONITORS { For debugging of scripts }
{ Same syntax as for PLOTS } 303

PLOTS
elevation(vm) on 'outer' elevation(p) from ... to … 228, 234
grid(x,y) vector(grad_f) as 'Gradient' surface(f) 10
contour(vm) painted report(brute_force) 228, 236
elevation(tangential(v)) on 'outer' 241
elevation(p) on 'circle' on 'domain' 245
vector(v) norm 228
contour(p) painted on 'domain' 245
contour(p) zoom(1.5*Lx,0, Lx,Ly) 228
grid(x,y,z)) 371
report(val(Ez,0,0.84,1)) 142
elevation(vz,vz_a) from (0,0,0) to (0,0,Lz) 371
contour(p) painted on x=0 report(Re) 371
vector(v) norm on y=2/3*x 375
history(E_k) report(visc) 358

END