

Visual Python (vPython.org) is a library for developing 3D graphics and simulation programs. To introduce you to its features, I will show you how to create a simulation of the Poiseuille Flow Model for a fluid flowing through a cylindrical pipe. The Poiseuille Flow Model is summarized by the following differential equation with an initial-condition

$$\frac{1}{r} d \left( r \frac{dv}{dr} \right) = -P, \quad v(a) = 0,$$

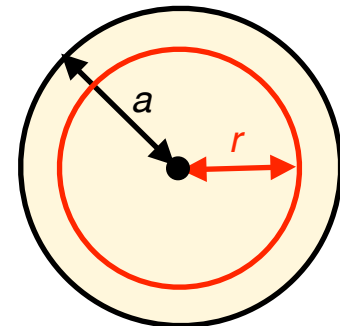
where

$a$  is the radius of the cylindrical pipe (a constant),

$r$  is the radial distance from the center of the pipe,  $0 \leq r \leq a$ ,

$v(r)$  is the velocity of the fluid at  $r$ ,

$P$  is the pressure (a constant) that causes the fluid to flow through the pipe.



Pipe Cross Section

Those of you that have studied differential equations should be able to show that the solution of this initial-value problem is

$$v(r) = \frac{P(a^2 - r^2)}{4}, \quad 0 \leq r \leq a.$$

Thus, the velocity decreases from  $\frac{Pa^2}{4}$  at the center of the pipe to zero at the pipe's boundary. Moreover, the velocity is independent of time!

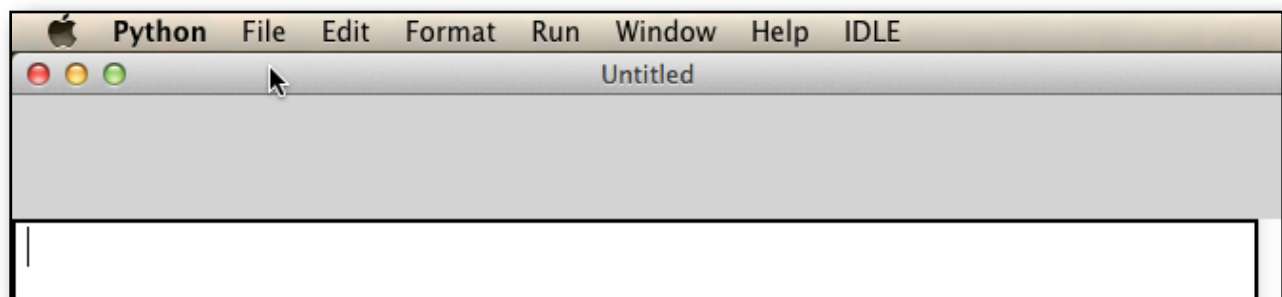
To start Visual Python and enter the Python code for this simulation do the following:



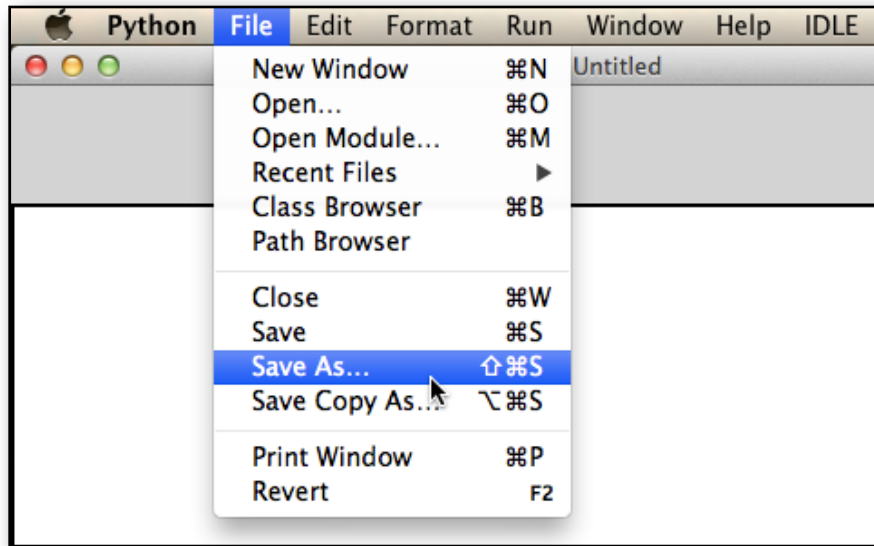
By selecting **Applications** from the **Go Menu** or other means, navigate to the **Applications** folder and *double-click* the Visual Python's VIDLE-Py2.7 app icon shown below.



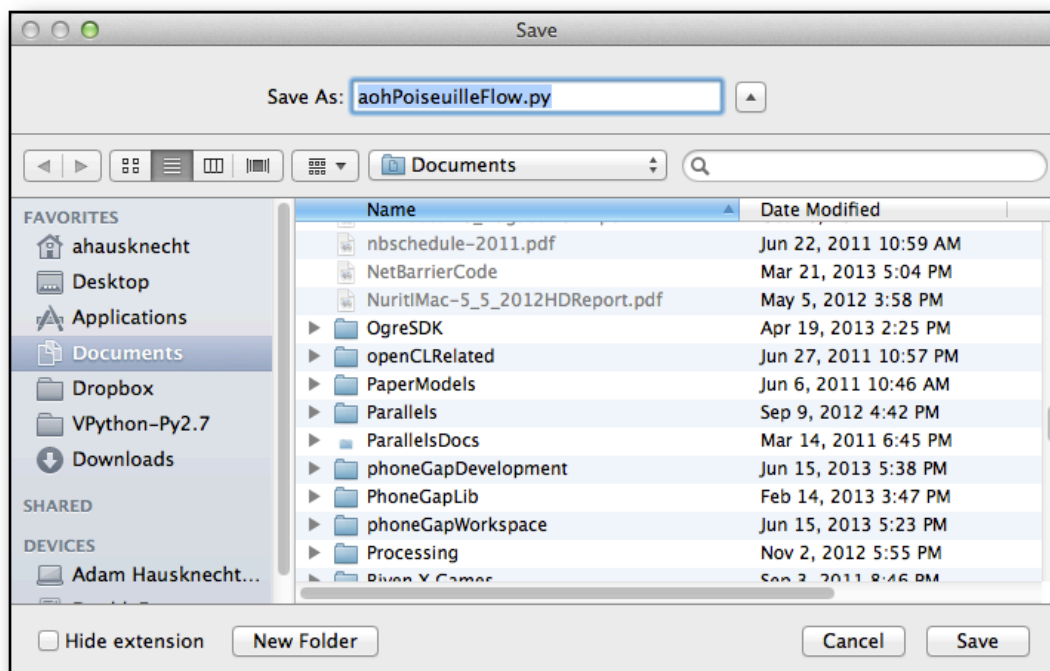
An **Untitled** VIDLE window will open where you can enter, save and execute Visual Python programs.



☞ Save the contents of the window to a file by, selecting **Save A...** from the **File Menu**.



☞ A **Save As** dialog will open.



☞ Use the **File** dialog to navigate to the **Documents** folder (directory) and save the VIDLE window with the name

**yourNamePoiseuilleFlow.py**

**NOTE:** The name of your file must end with the **.py** suffix!

☞ A **Save** dialog will close and the **VIDLE** program window's name will change to the file's name.



Enter the program below in the empty VIDLE program window.

```
from visual import * # Import the Visual Python module
# 1. Set up the scene for your simulation.
scene2 = display(title='Poiseuille Flow', x=0, y=0,
                 width=600, height=600, center=(15,0,0), background = color.white)
#
# 2. Create and initialize the model's variables.
a = 10; P = 0.1
#
# 3. Use a Visual Python's cylinder object to represent the pipe.
# Note that the cylinder's opacity is set to 0.4 so that you can
# see into the pipe's interior.
f = frame() # Create a common frame to rotate the entire simulation.
pipeLength = 30
pipe = cylinder(frame = f, pos=(0,0,0), axis=(pipeLength, 0, 0),
               radius = a, color = color.gray(0.5), opacity=0.4)
#
# 4. Use a Visual Python label object to label the simulation.
pLabel = label(pos = (15,12,0), text = 'V(r) = P/4(a^2 - r^2)',
               height = 24, color = color.red)
#
# 5. Use spheres to simulate the flow of the fluid through the pipe.
useZParticles = True # 5a. Enable particles in the z-direction.
particles = [] # 5b. Use a list to to store the particles.
# 5c. Use another list the store the particle's colors.
pColors = [color.red, color.green, color.blue,
           color.yellow, color.cyan, color.magenta]
# 5d. Use a for-loop to generate the particles in the vertical
# direction (= along the y-axis).
for r in arange(-9.5, 9.5, 0.5):
    # Use a small sphere to represent a particle of fluid.
    p = sphere(frame=f, pos=(0,r,0), radius=0.5, color=color.red)
    # Save the particle's starting position as a Visual Python vector
    # in the instance variable "startPos".
    p.startPos = vector(0, r, 0)
    # Compute and save the particle's velocity as a Visual Python
    # vector in the instance variable "velocity".
    V = P/4.0*(a**2 - r**2)
    p.velocity = vector(V, 0, 0)
    # Also, save the particle's travel 't' and color index.
    p.t = 0.0
    p.colorIndex = 0
    # Finally, add the particle to the list of particles.
    particles.append(p)
```

```

# 5e. Use a for-loop to generate the particles along the z-axis.
if useZParticles:
    for r in arange(-9.5, 9.5, 0.5):
        p = sphere(frame=f, pos=(0,0,r), radius=0.5, color=color.red)
        p.startPos = vector(0, 0, r)
        V = P/4.0*(a**2 - r**2)
        p.velocity = vector(V, 0, 0)
        p.t = 0.0
        p.colorIndex = 0
        particles.append(p)

# 6. Create and enter the animation-loop.
dt = 0.01 # The time increment
while True:
    # Repeat forever!
    rate(100) # Set the animation's "frame" rate.
    for p in particles: # for each particle ....
        p.t += dt # Increment the particle's travel time.
        if p.x < pipeLength: # p has not reached the end of the pipe.
            # Calculate a set the particle's position at time p.t .
            p.pos = p.startPos + p.velocity*p.t
        else:# p has reached the end of the pipe, so
            # restart it at the beginning of the pipe using
            # a different color!
            p.pos = p.startPos
            p.t = 0
            # Use "mod" to compute the next color index.
            p.colorIndex = (p.colorIndex + 1)%len(pColors)
            # Set the particle's new color.
            p.color = pColors[p.colorIndex]
    # Rotate the entire simulation.
    f.rotate(angle = dt/10, axis=(0, 1, 0), origin = (15, 0, 0))

```



Run your program by selecting **Run Module** from VIDLE's **Run** menu.

```

from visual import *
# 1. Set up the scene for your simulation
scene2 = display(title='Poiseuille Flow',
width=600, height=600, center=(15,0,0)
# 2. Create and initialize the model's va

```

