Building a System to Perform Fluid Dynamics Simulations and Experiments

OU Supercomputing Symposium
Oct. 6, 2010

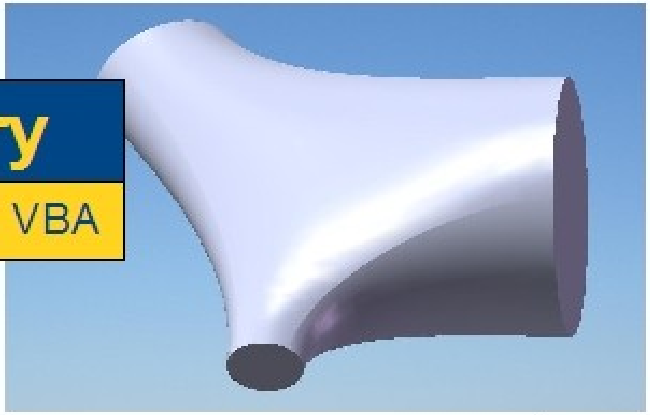
Willy Duffle, Chris Kiser, Grant Armstrong, Andrew Baker, Andrew Henderson, Alex Macharia, Lamar Williams, and Evan Lemley

Department of Engineering and Physics
University of Central Oklahoma

Dimitrios Papavassiliou, Henry Neeman, Ed O'Rear
University of Oklahoma

Generate Junction Geometry

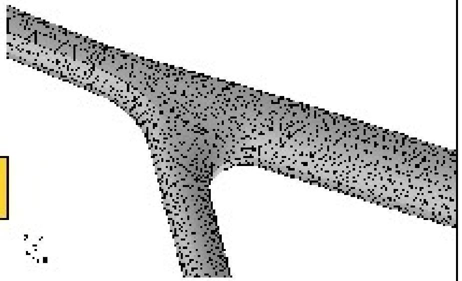
Junction code (C++) → SolidWorks VBA



Simulations

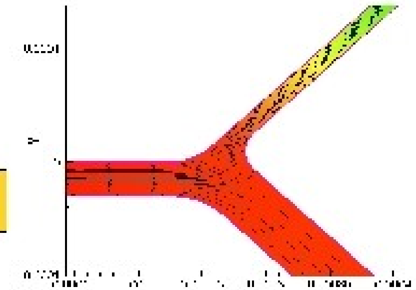
Generate Mesh

Junction Code → GAMBIT



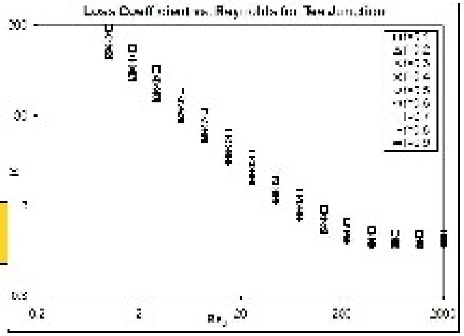
Simulate Flow

Junction Code → FLUENT



Analyze Results

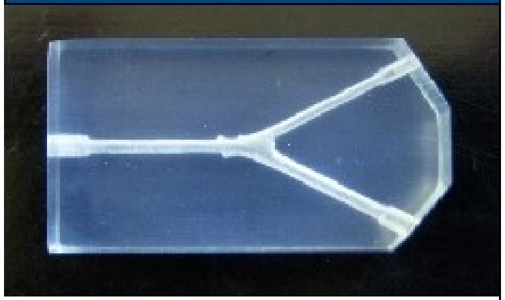
Junction Code → tsv

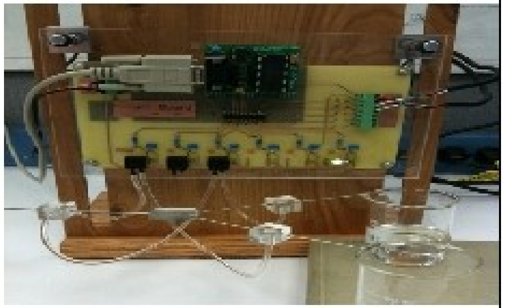


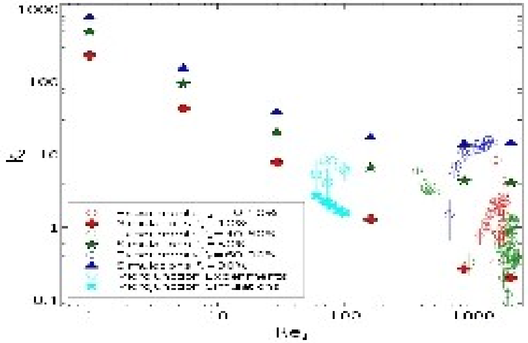
Experiments

Milliscale

Microscale







Overview

Background

Simulations

- Flow Through Porous Media (FTPM)
- Flow in Junctions/Microjunctions
- Microelbows
- Entrance Length in Microtubes
- Reynolds Number Dependence tees and wyes

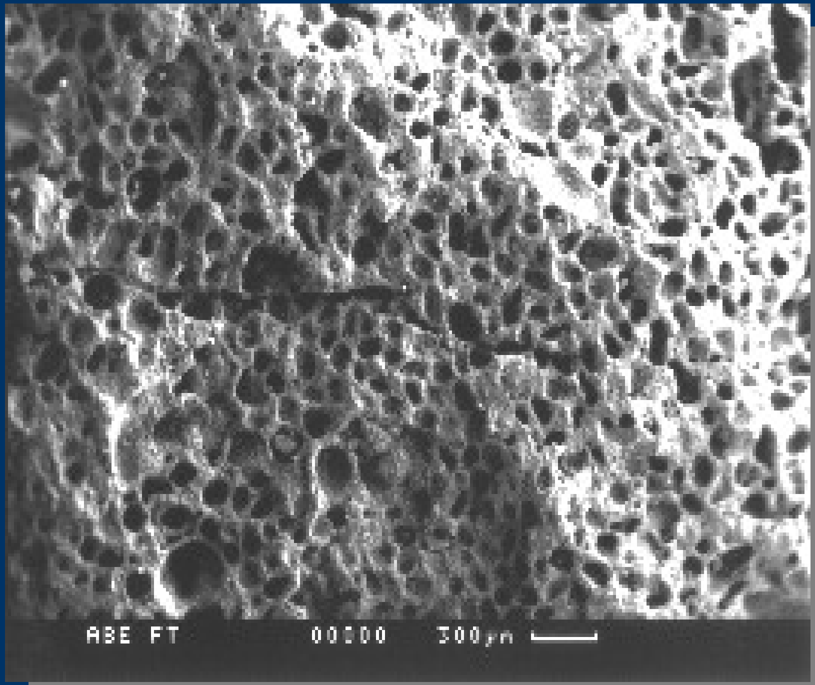
Experimental

- Millijunctions & MicroJunctions

Current and Future Directions

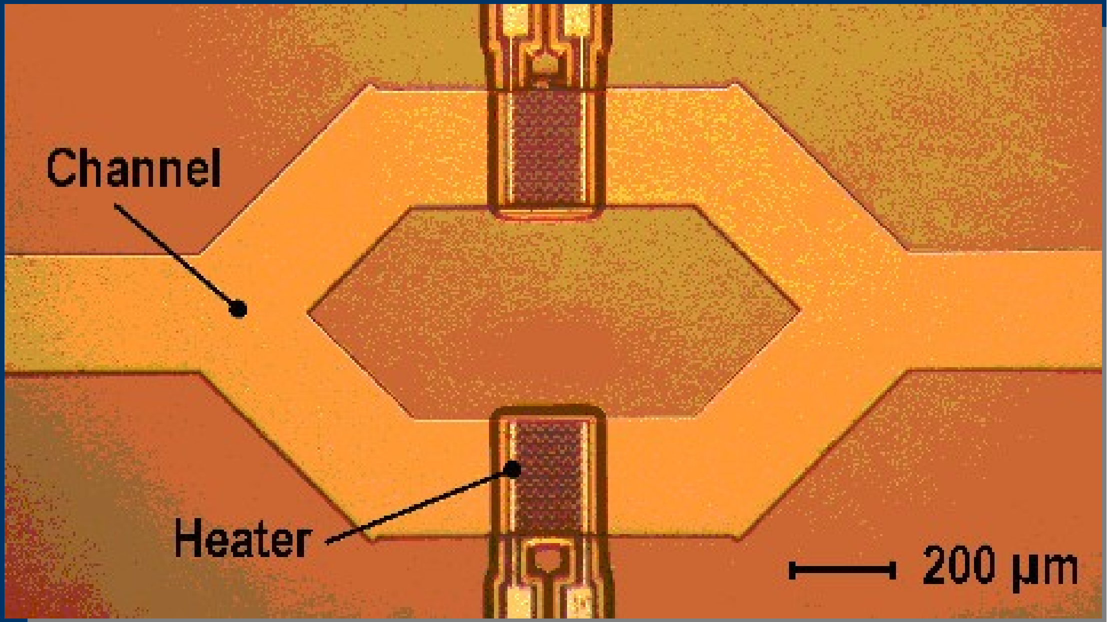
- Bio-Scaffolds
- Renal Artery Aneurysm
- Microtubes

Porous Media



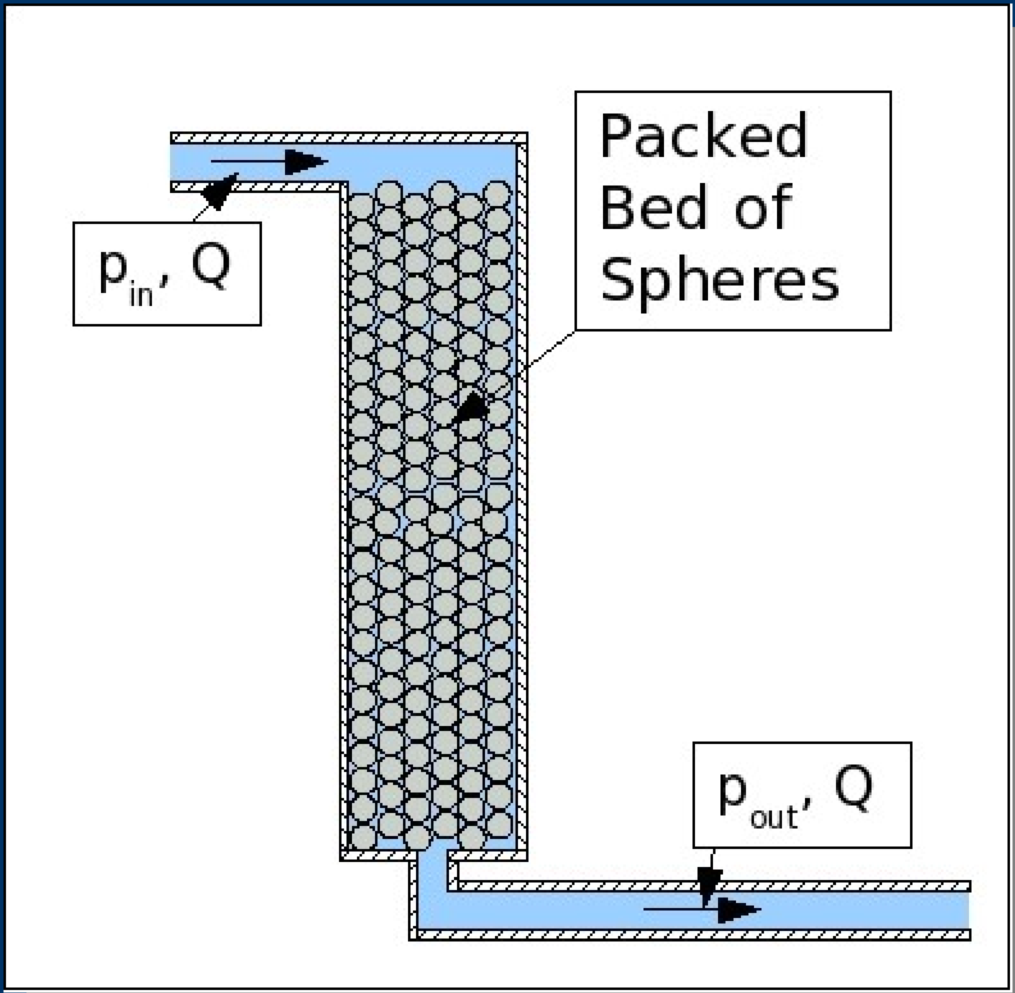
Highly porous magnesian limestone.
(www.dawntnicholson.org.uk)

Microfluidic Devices



Microfluidic Valve Structure.
(http://www.cchem.berkeley.edu/sjmgrp/people/boris/boris.htm)

Artificial Porous Media



Packed Beds, Gas and Liquid Filters

Sphere sizes μm to cm

Hold-up for chemical reaction, thermal processing, or filtering

Basics of Porous Media

Low Speed Flow – Darcy's Law

$$\frac{dp}{dx} = \frac{\mu}{\kappa} u$$

$p = \text{pressure}$ $\quad x = \text{position}$
$\mu = \text{viscosity}$ $\quad \kappa = \text{permeability}$
$u = \text{filtration velocity}$

High Speed Flow – Forchheimer's Law

$$\frac{dp}{dx} = \frac{\mu}{\kappa} u + \rho \beta u^2$$

$\rho = \text{density}$
$\beta = \text{Forccheimer's Coefficient}$

Packed Beds – Ergun's Equation (empirical)

$$\frac{-\Delta p}{\Delta L} = \frac{150\,\mu}{d_p^{\,2}} \frac{(1-\phi)^2}{\phi^3} u + \frac{1.75\,\rho(1-\phi)}{d_p \phi^3} u^2$$

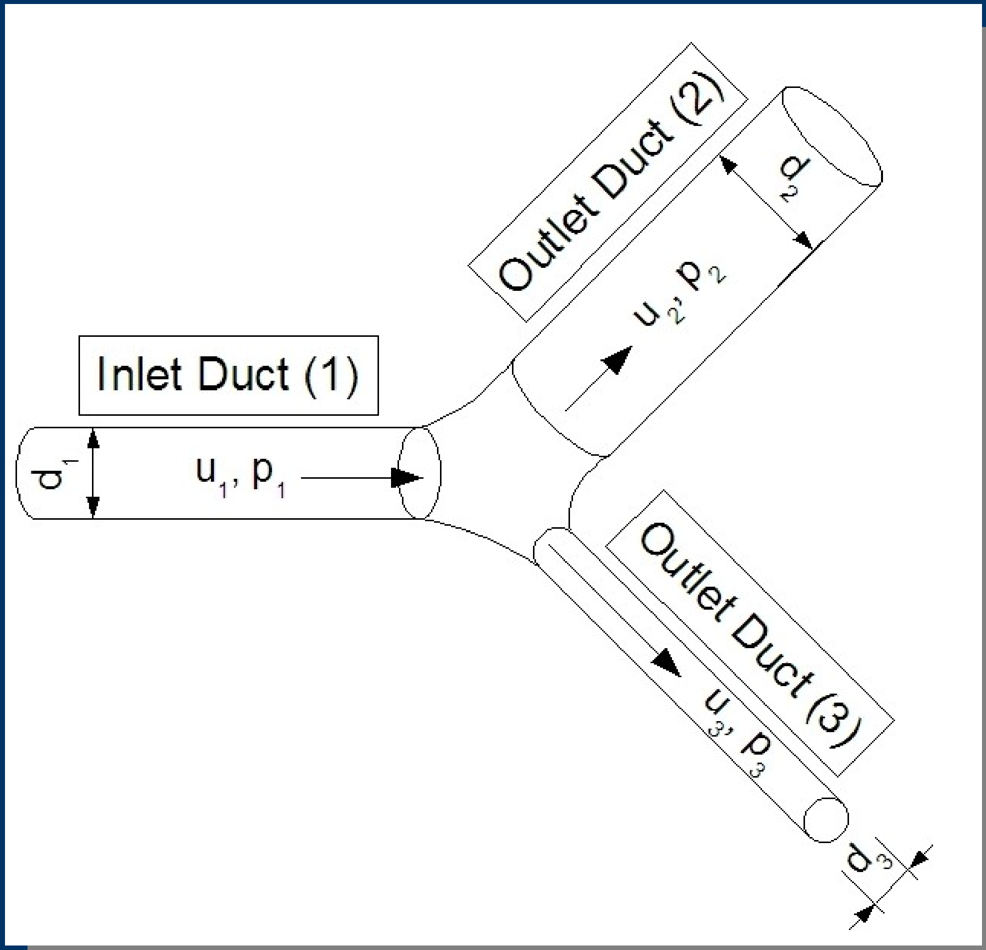
$\phi = \text{porosity}$
$d_p = \text{mean sphere diameter}$

Flow Through Porous Media

Collaborative Effort with Dimitrios Papavassiliou and Henry Neeman from OU (began Fall 2004)
Simulation of Flow of Fluids through Porous Media

Code FTPM – Flow Through Porous Media.
Solves for velocity and pressure at pore junctions in a randomly generated pore network.

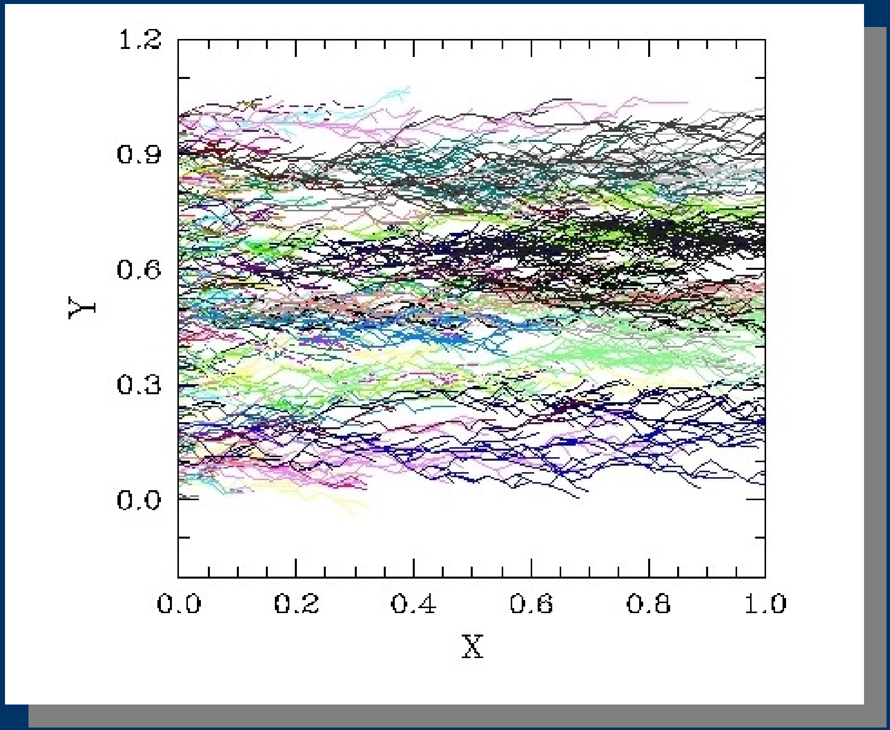
Flow Network Analysis



Design and Analysis of networks depends on knowledge of flow and energy losses in arbitrary branches.

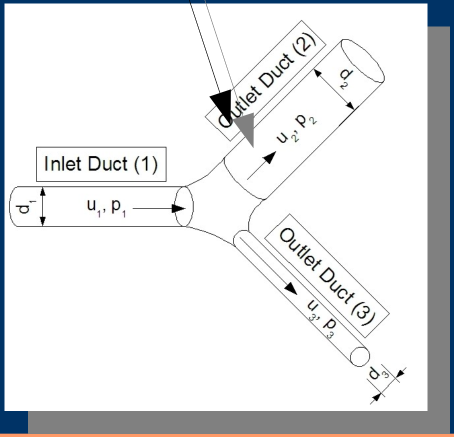
No systematic studies to generalize these bifurcations

Porous Network Simulator (Collaboration with Univ. of Oklahoma)

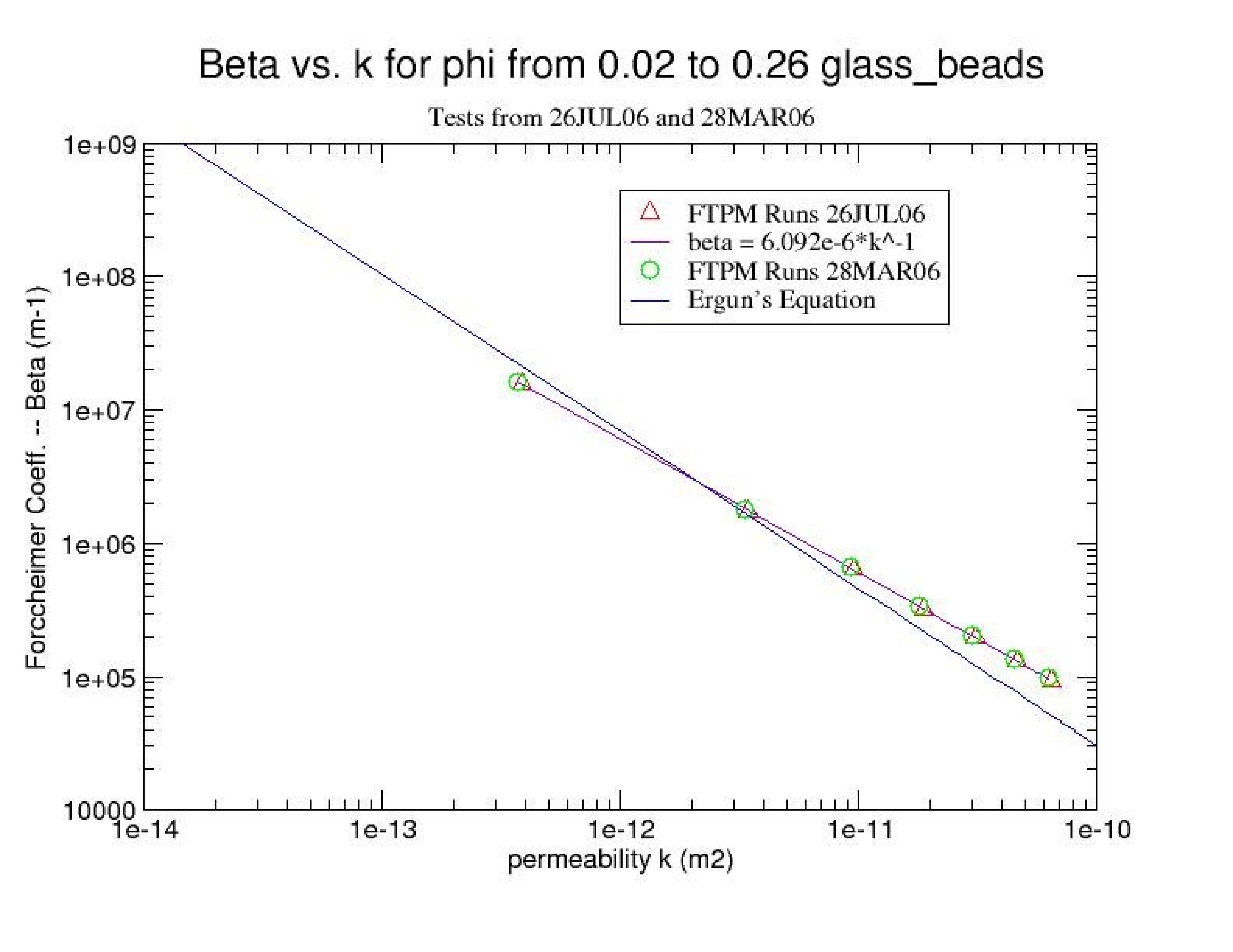


Projection on the xy plane of a 3D network that has 200 entry points at x=0, porosity equal to 10% and a range of ±60˚ relative to the x axis and ±30˚ relative to the y axis.

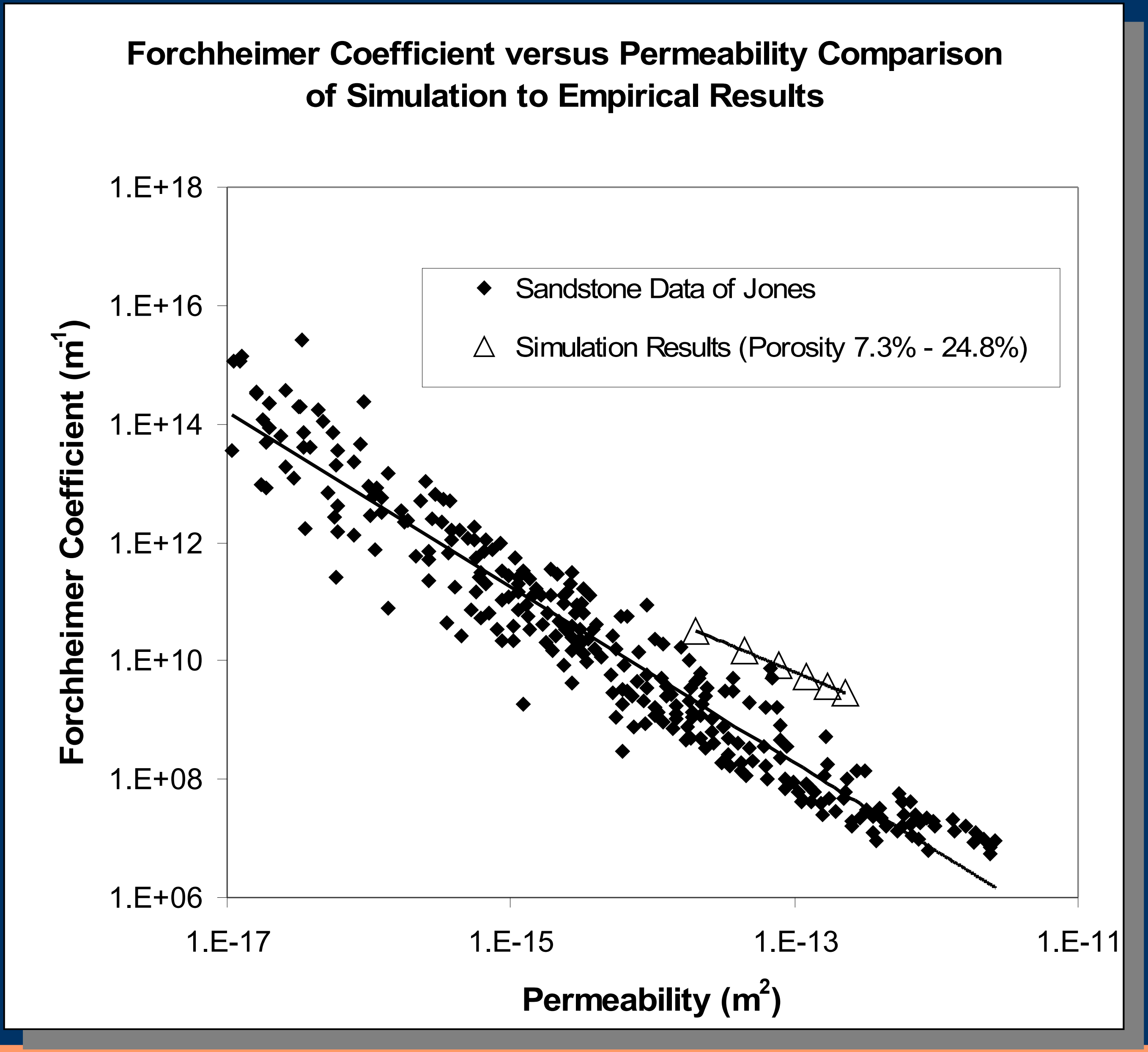
3D Monte Carlo networks from normal, beta, or empirical distribution (pore size pdf)
Coordination Number (1, 2, 3)
number of pores entering and leaving a junction
$\theta \pm 90^{\circ}$



FTPM Results



FTPM Results



Literature

T's and Y's – limited configurations and most are for turbulent flow

Basset, M.D., Winterbone, D.E., and Pearson, R.J., 2001, “Calculation of Steady Flow Pressure Loss Coefficients for Pipe Junctions,” Proc. Instn. Mech. Engrs., Part C, Journal of Mechanical Engineering Science, **215** (8), pp. 861-881.

W.H. Hager, 1984, “An Approximate Treatment of Flow in Branches and Bends,” Proc. Instn. Mech. Engrs., Part C, Journal of Mechanical Engineering Science, **198**(4) pp. 63-9.

Blaisdell, F.W., and Manson, P.W., 1967, “Energy loss at pipe junctions,” J. Irrig. and Drainage Div., ASCE, **93**(IR3), pp. 59-78.

Schohl, G.A., 2003, “Modeling of Tees and Manifolds in Networks,” *Proceedings of the 4th ASME/JSME Joint Fluids Engineering Conference*, **2**, Part D, pp. 2779-2786.

Bassett, M.D., Pearson, R.J., and Winterbone, D.E., 1998, “Estimation of Steady Flow Loss Cofficients for Pulse Converter Junctions in Exhaust Manifolds,” *IMechE Sixth International Conference on Turbocharging and Air Management Systems*, IMechE HQ, London, UK, **C554/002**, pp.209-218.

Ruus, E., 1970, “Head Losses in Wyes and Manifolds,” J. Hyd. Div., ASCE, **96**(HY3), 593-608.

Laminar loss coefficients and elbows, reductions, contractions – much *larger loss coefficients than turbulent case – strong dependence on Reynold's number*.

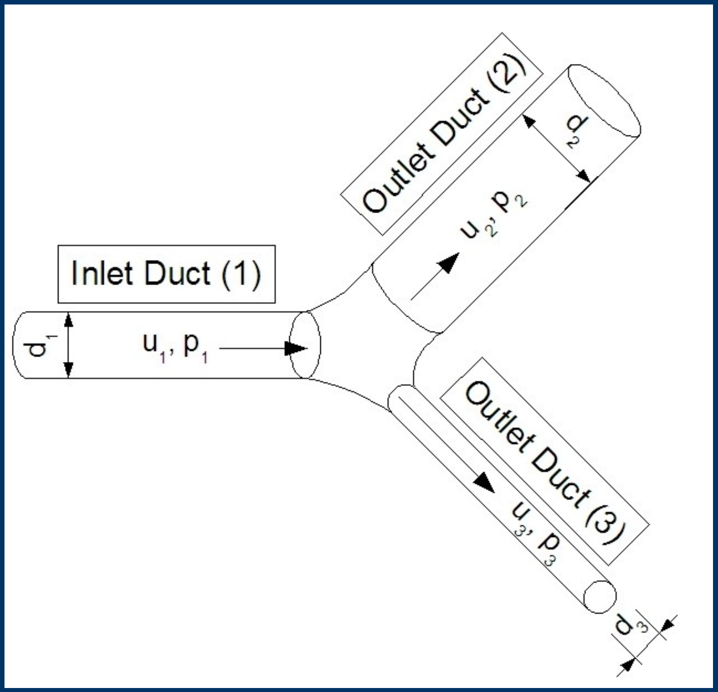
Edwards, M.F., Jadallah, M.S.M., and Smith, R., 1985, “Head Losses in Pipe Fittings at Low Reynolds Numbers,” Chem. Engr. Res. Des., **63**(1), pp. 43-50.

Importance of roughness at microscale

Problem Description

Stagnation Loss Coefficient

$$K_2 = \frac{\left[\left(\frac{p_1}{\rho} + \frac{u_1^{\,2}}{2}\right) - \left(\frac{p_2}{\rho} + \frac{u_2^{\,2}}{2}\right)\right]}{\frac{u_1^{\,2}}{2}}$$

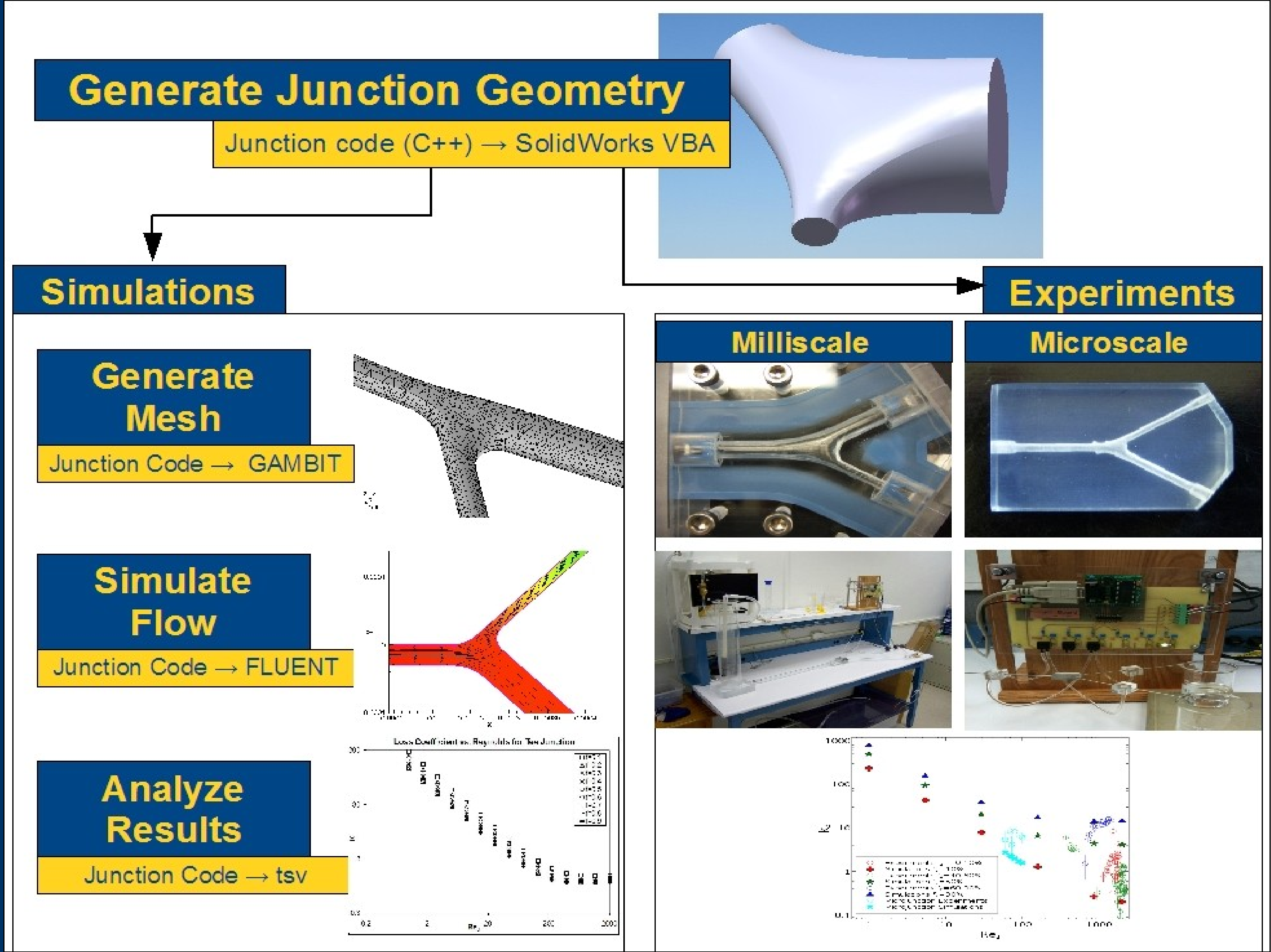


Parameters:

d_2/d_1, d_3/d_1

θ_2 and θ_3

f_2 (this sets f_3) – (why? other literature and possibility of simulations where this is unknown initially)


Generate Junction Geometry
Junction code (C++) → SolidWorks VBA
Simulations
Experiments
Generate Mesh
Junction Code → GAMBIT
Simulate Flow
Junction Code → FLUENT
Analyze Results
Junction Code → tsv
Milliscale
Microscale

Automation of Geometry Generation and CFD Runs

Custom Code was written to:

(a) create GAMBIT journal files that instantiate the desired geometry *based on existing 2D geometries.*

(c) create a script that loads journal files into GAMBIT and meshes

(d) create all necessary preprocessing files for FLUENT.

(e) create post-processing files for FLUENT results and to tabulate results for a complete set of runs

Solution Methodology
2D Geometry Generalization

$L_1 = 5\ d_{max}$

L_2 and $L_3 = 10d_{max}$

If $d_2 > d_{avg}$,

then $r_2 = 3d_2$;

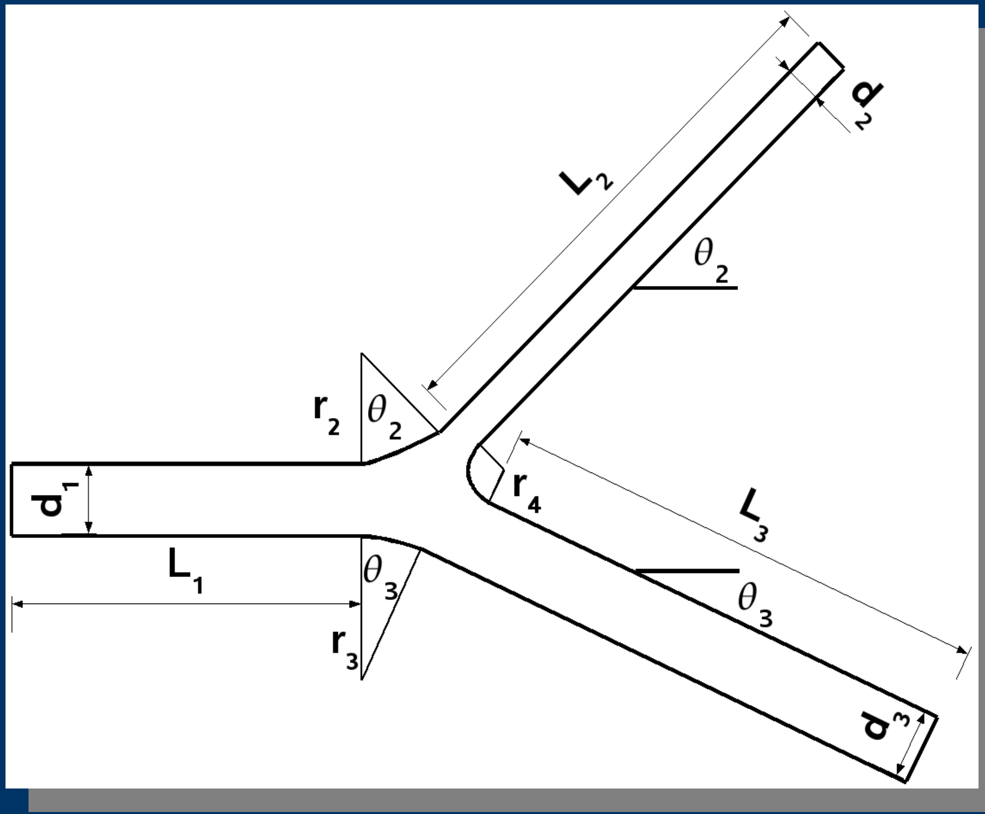
else $r_2 = 2d_2$

If $d_3 > d_{avg}$,

then $r_3 = 3d_3$;

else $r_3 = 2d_3$

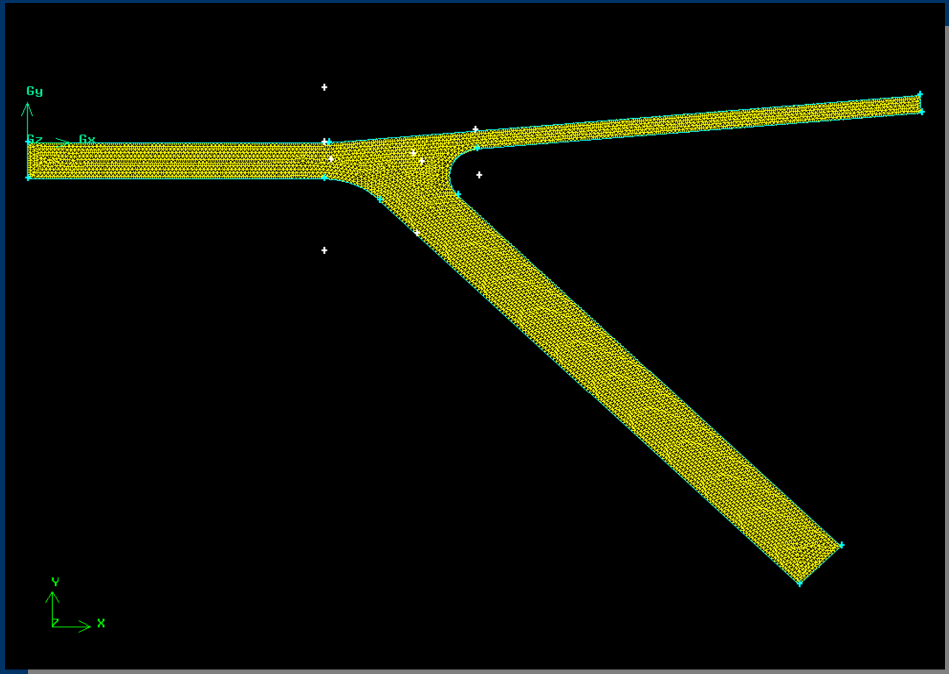
$r_4 = d_{max}/2$



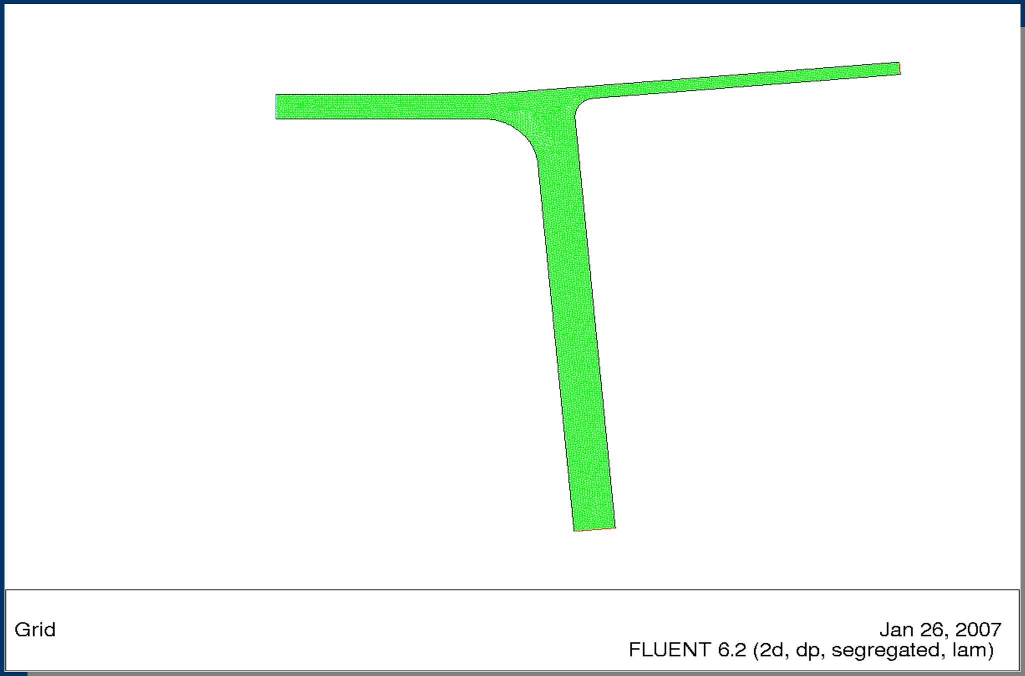
Generalized Geometry

larger and smaller outlet ducts ; 2Dim. - 3Dim are underway

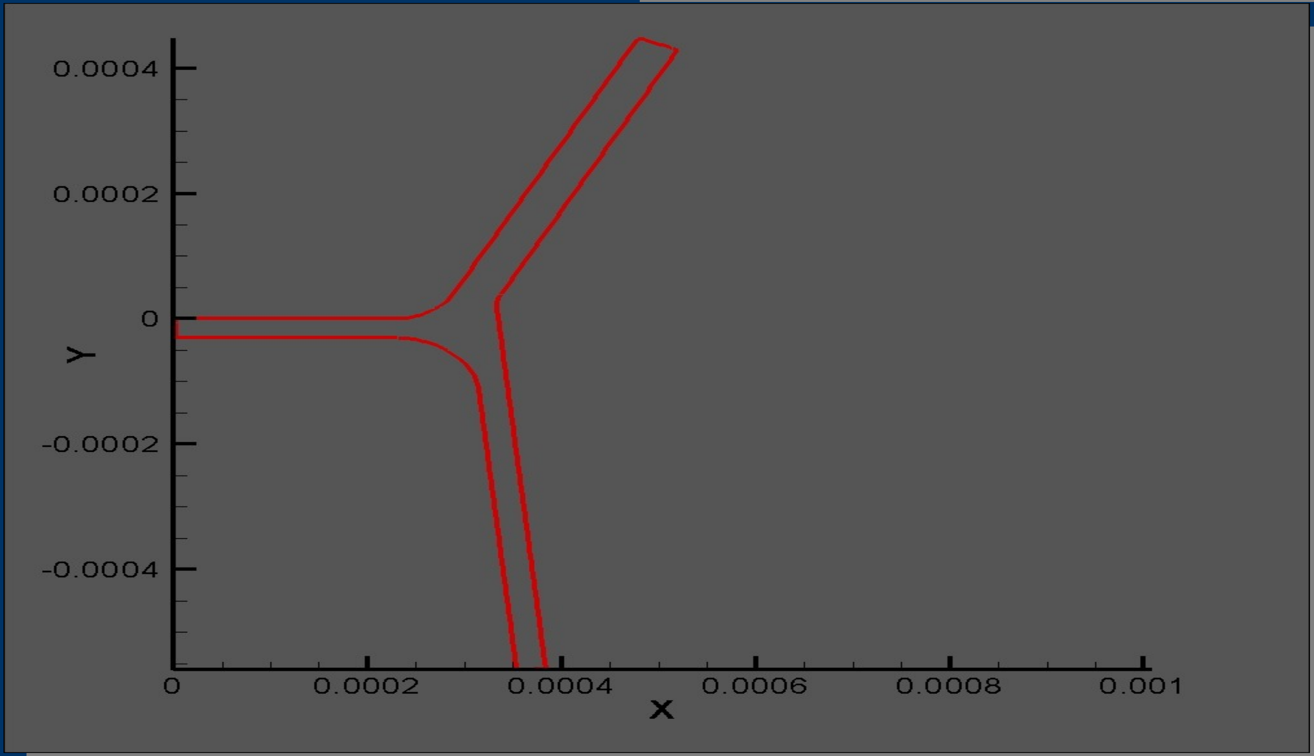
avoid sharp edges ; 5 – 90 degrees for angles



Gy
Gz
Gx
Y
Z
X

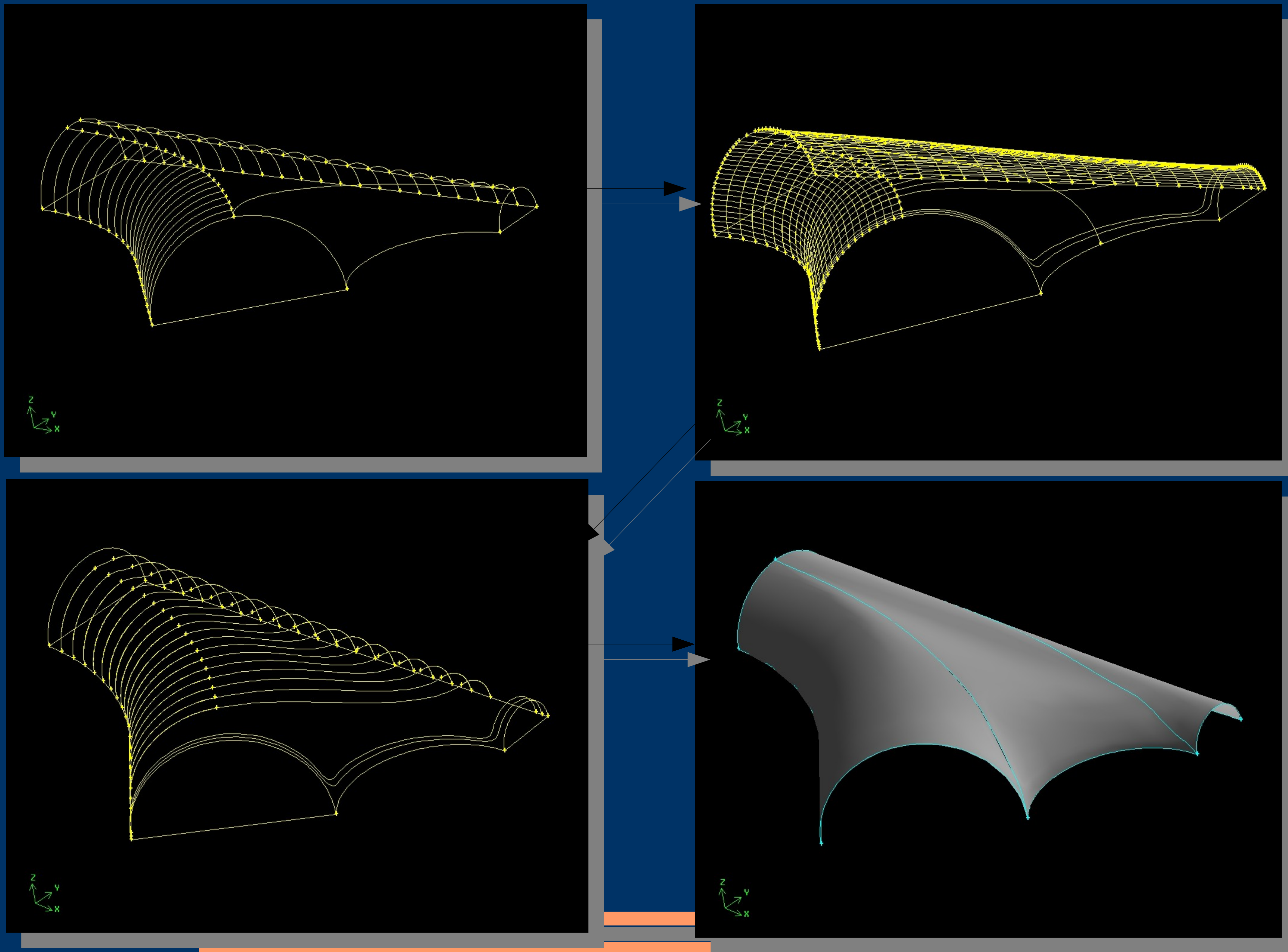


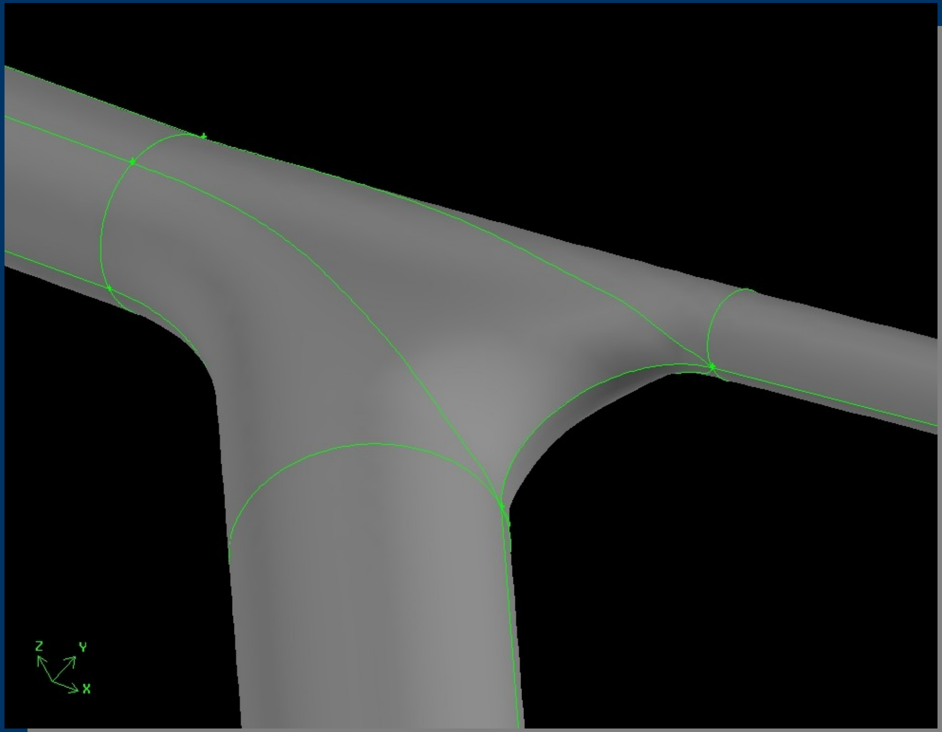
Grid
Jan 26, 2007
FLUENT 6.2 (2d, dp, segregated, lam)

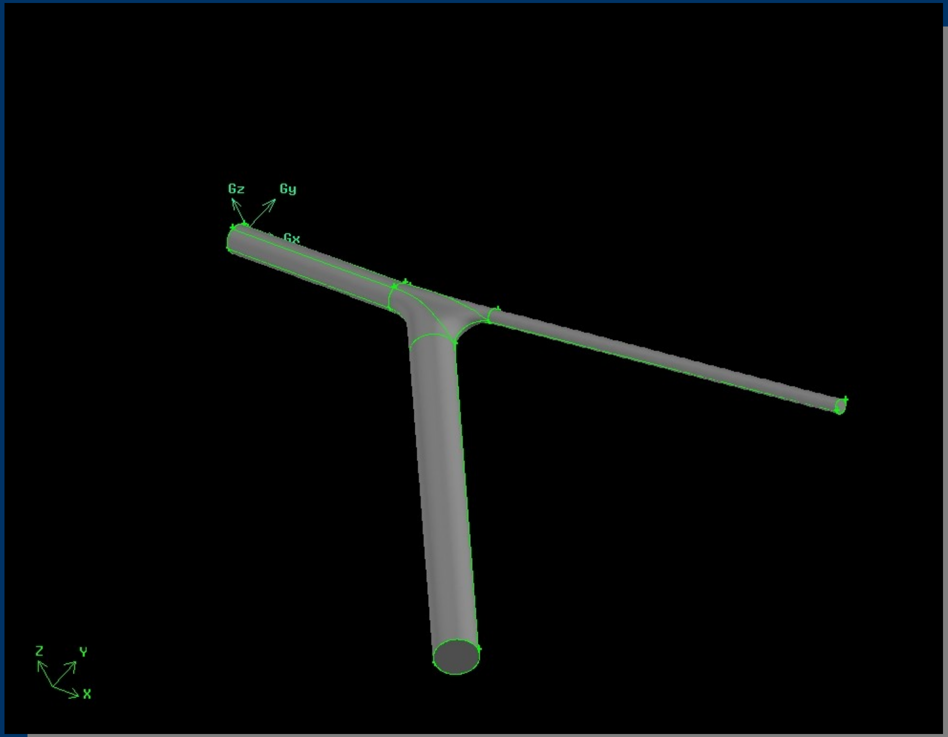


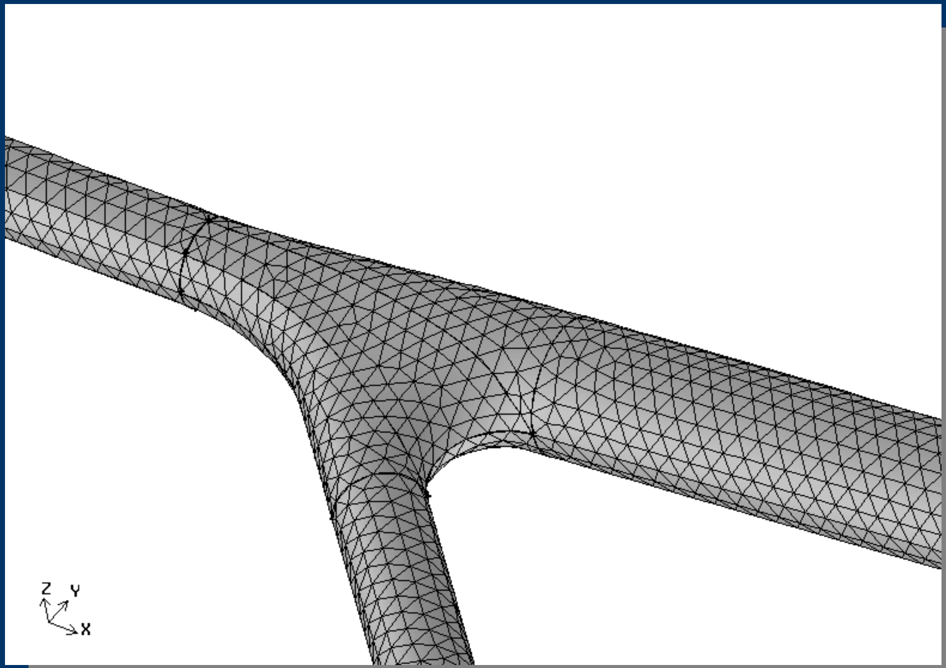
0.0004
0.0002
0
-0.0002
-0.0004
Y
0
0.0002
0.0004
0.0006
0.0008
0.001
X

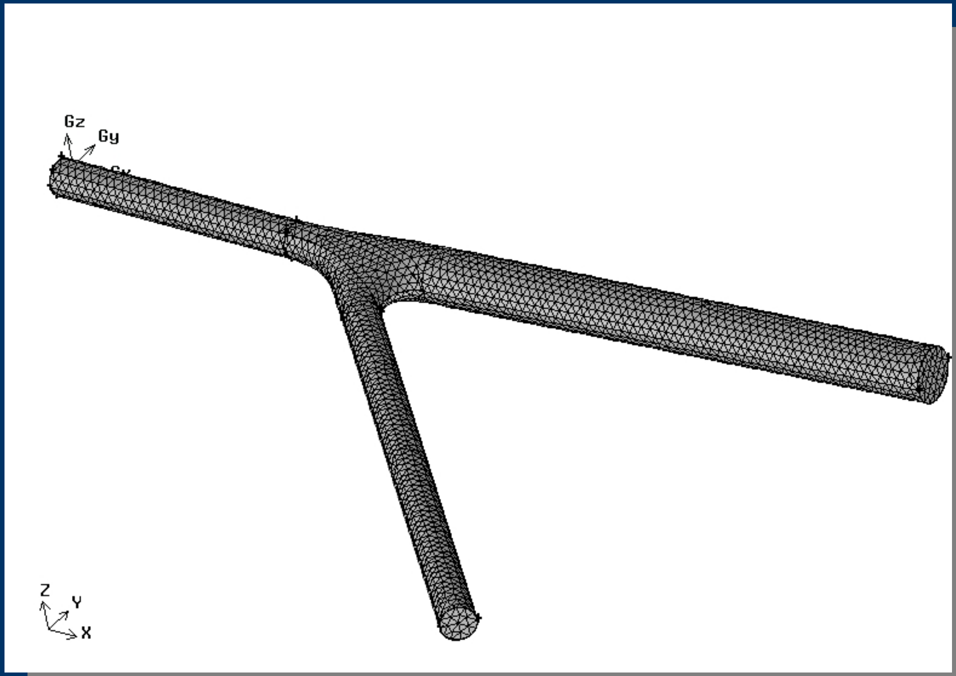
3D Geometry – Mark I




Z
Y
X


Gz
Gy
Gx
Z
Y
X

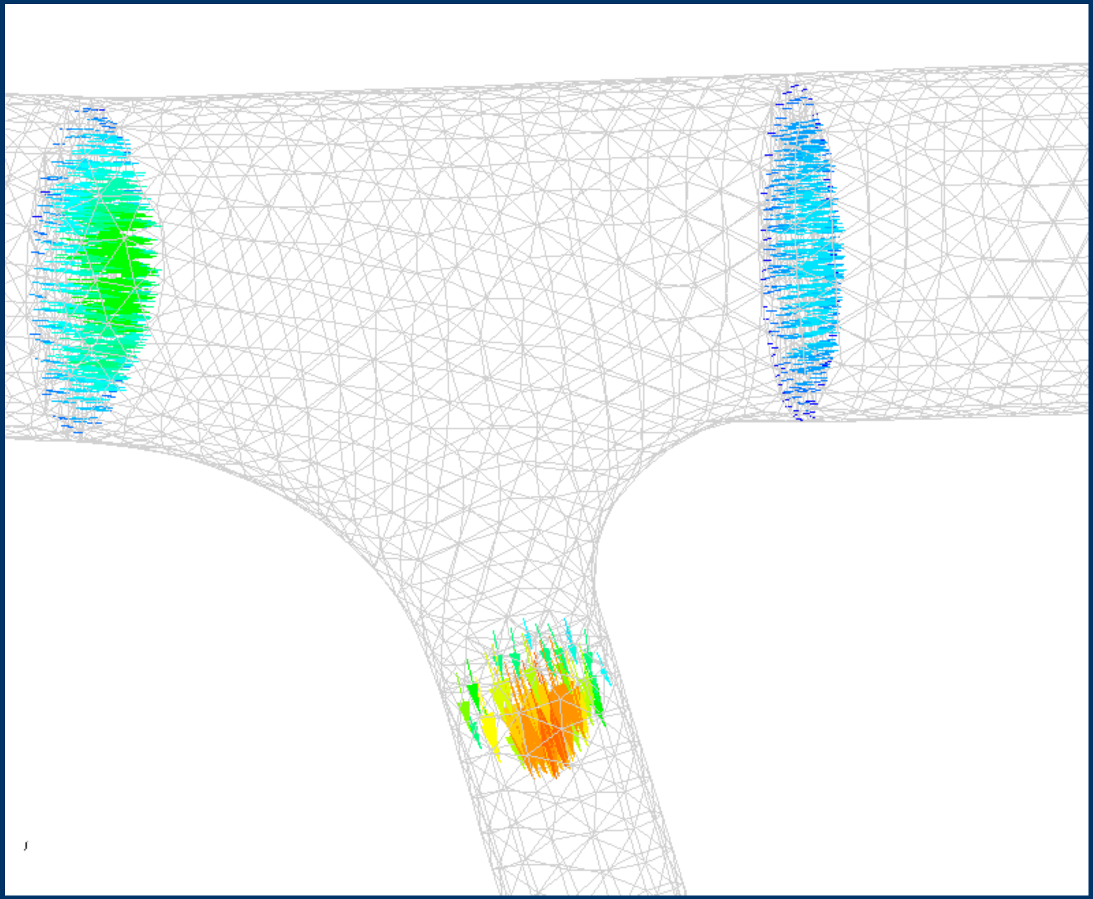

Z
Y
X


Gz
Gy
Z
Y
X

Simulation Parameters

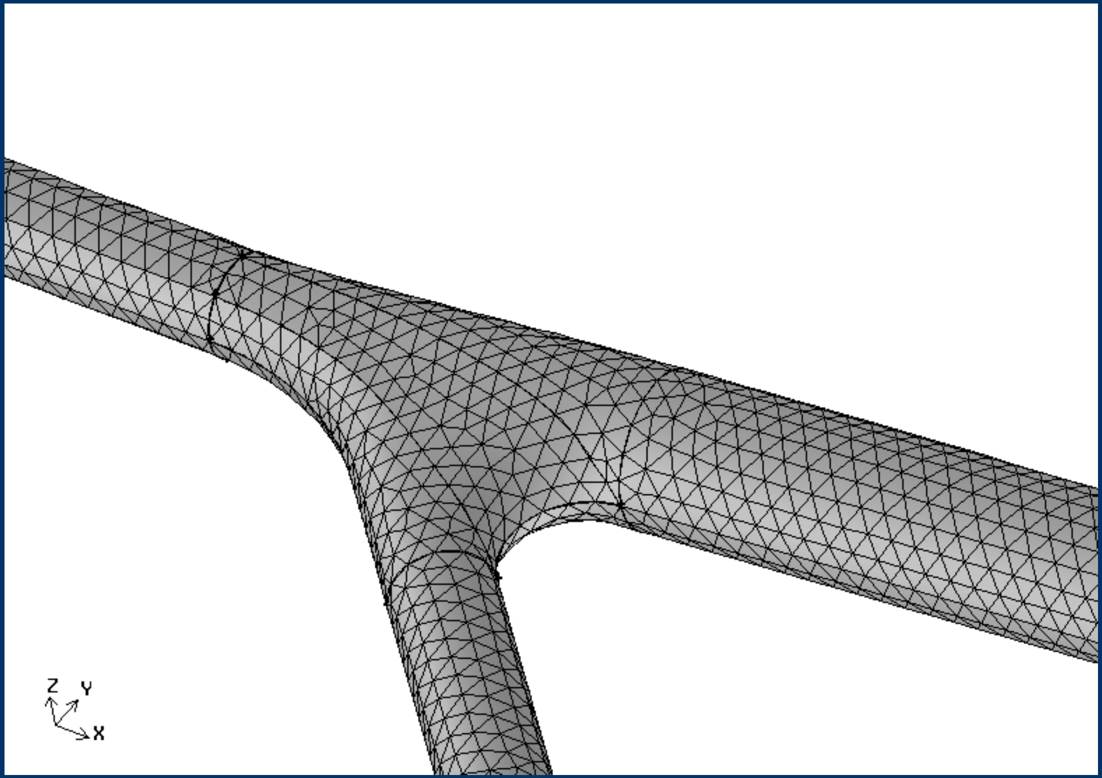
- Re_D ***(Reynolds Number = proportional to speed)*** was maintained at constant value at the inlet duct
- d_1 was 30 microns. The fluid was chosen to be liquid water at 20˚C. The inlet flow velocity, u_{1}, was set to 0.5 m/s giving a Reynolds number of 15
- $Le_D = 0.06Re_D\, D$ --- gives 0.9 D for $Re_D = 15$
- FLUENT output files contain surface averaged static pressure and magnitude of flow velocity at duct cross-sections at the duct inlets and exits.

Duct Inlet/Outlets



Duct inlet and exit sections considered to be where geometry of duct is the same as the downstream portion for outlets and upstream portion for inlet.

Mesh Automation



- Mesh was set to 1/4 of smallest duct
- Tetrahedral Mesh
- Large number of tests to assess ability to generalize the mesh (1/4 factor determined in this manner)
- Some testing to verify no change in results with change in mesh size.
- Inlet was specified as velocity inlet
- Outlets were outflow boundaries – allowed specification of flow fraction

Numerical Methods

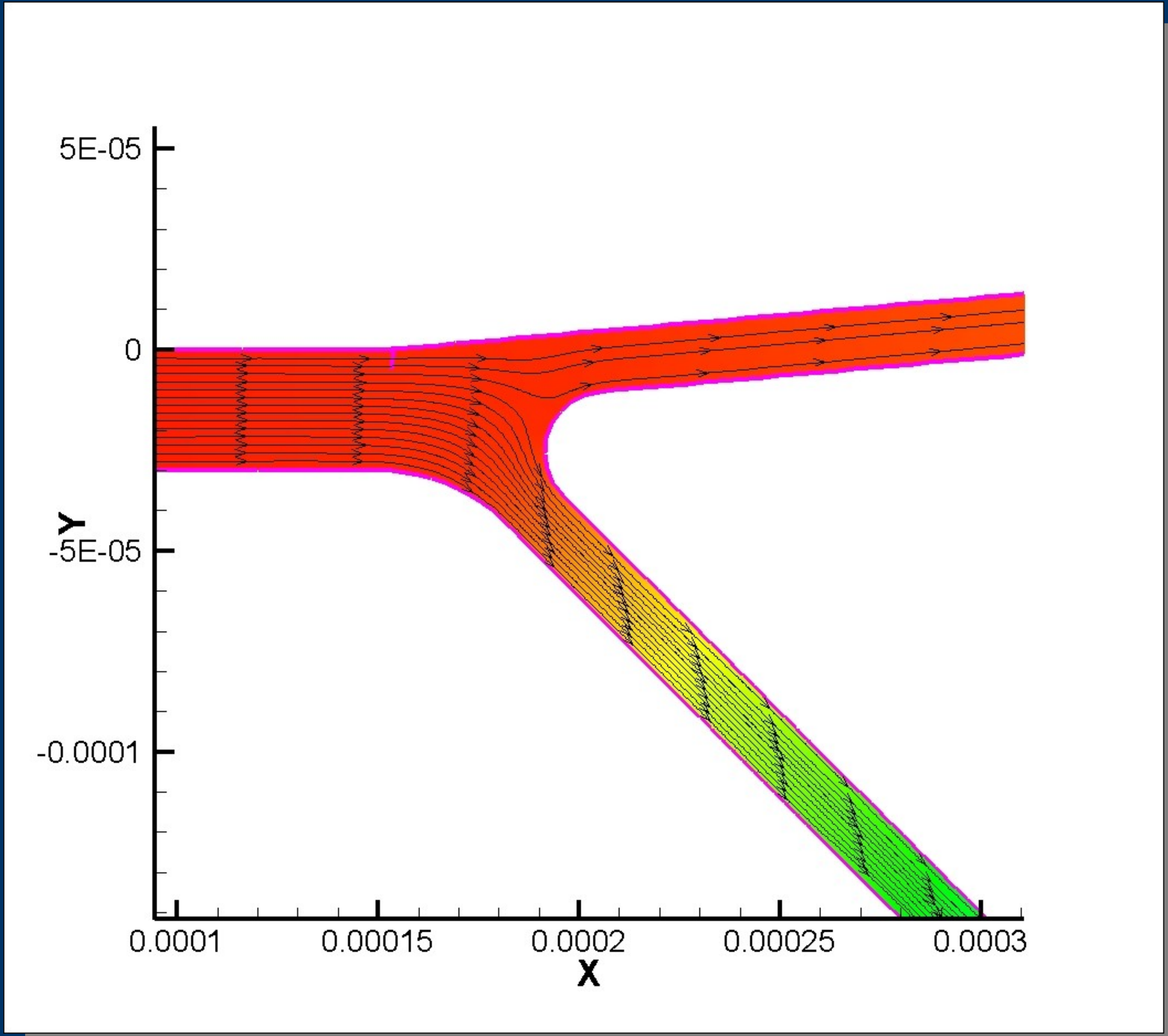
Finite Volume solution of integral Navier Stokes
Steady-State 3D
Implicit
SIMPLE for pressure velocity coupling
1st order upwind scheme of momentum discretization
Max number of iterations
Convergence criterion = 0.1%

Parameter Values

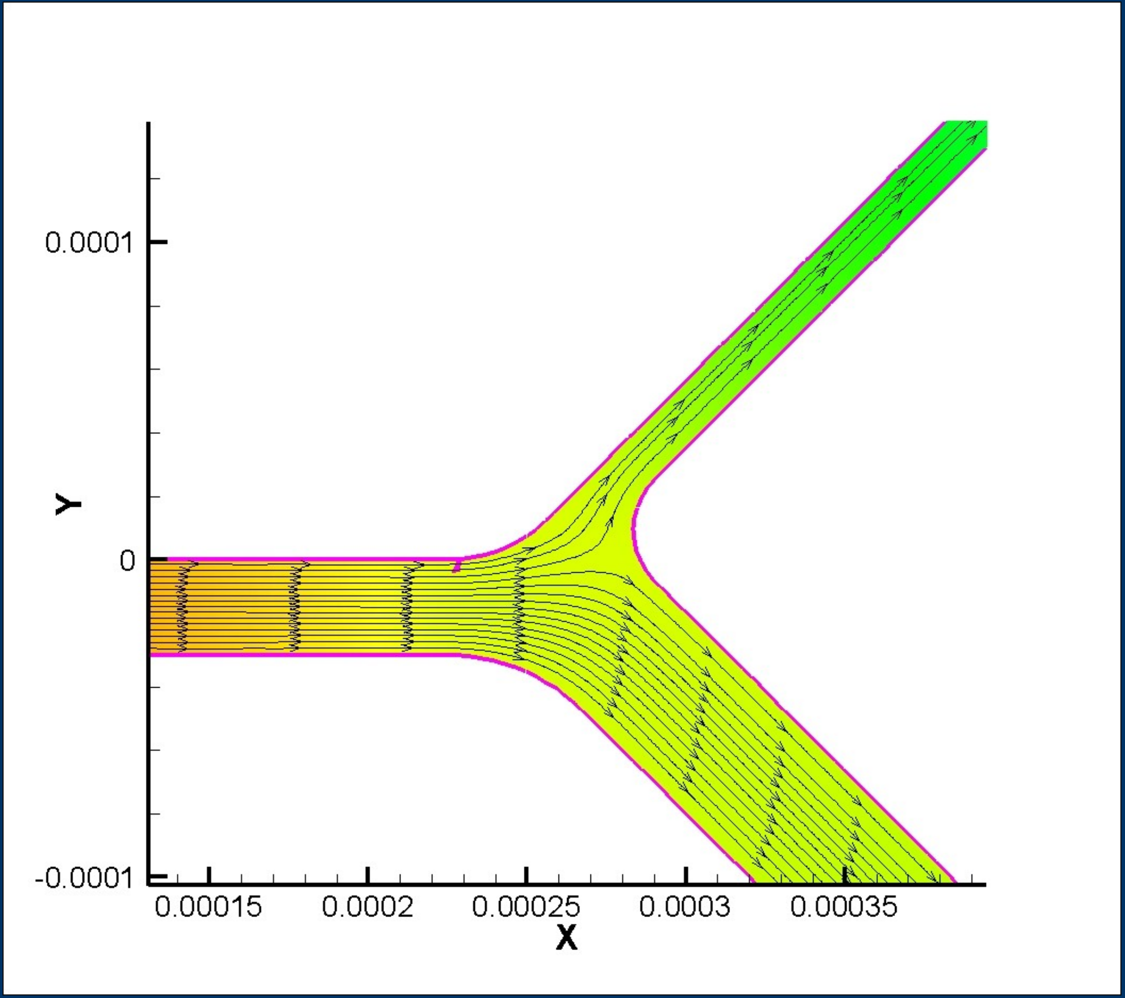
d_2/d_1 and d_3/d_1 = 0.5, 1.0, 1.5
f_2 = 0.1, 0.3, 0.5, 0.7, 0.9
θ_2 and θ_3 = 5°, 25°, 45°, 65°, 85°
600 runs attempted – 475 completed (*geometry issues on remainder*)
Suite of C++ procedures to create geometries, input files, read and collate results
Create GAMBIT script to create geometries
Create input files for GAMBIT and FLUENT
Read results files for static pressures and velocities averaged over surfaces in and out of junctions.

Fluent Result

f2 = 0.1, d2/d1 = 0.5, θ2 = 5˚, d3/d1 = 0.5, and θ3 = 45˚

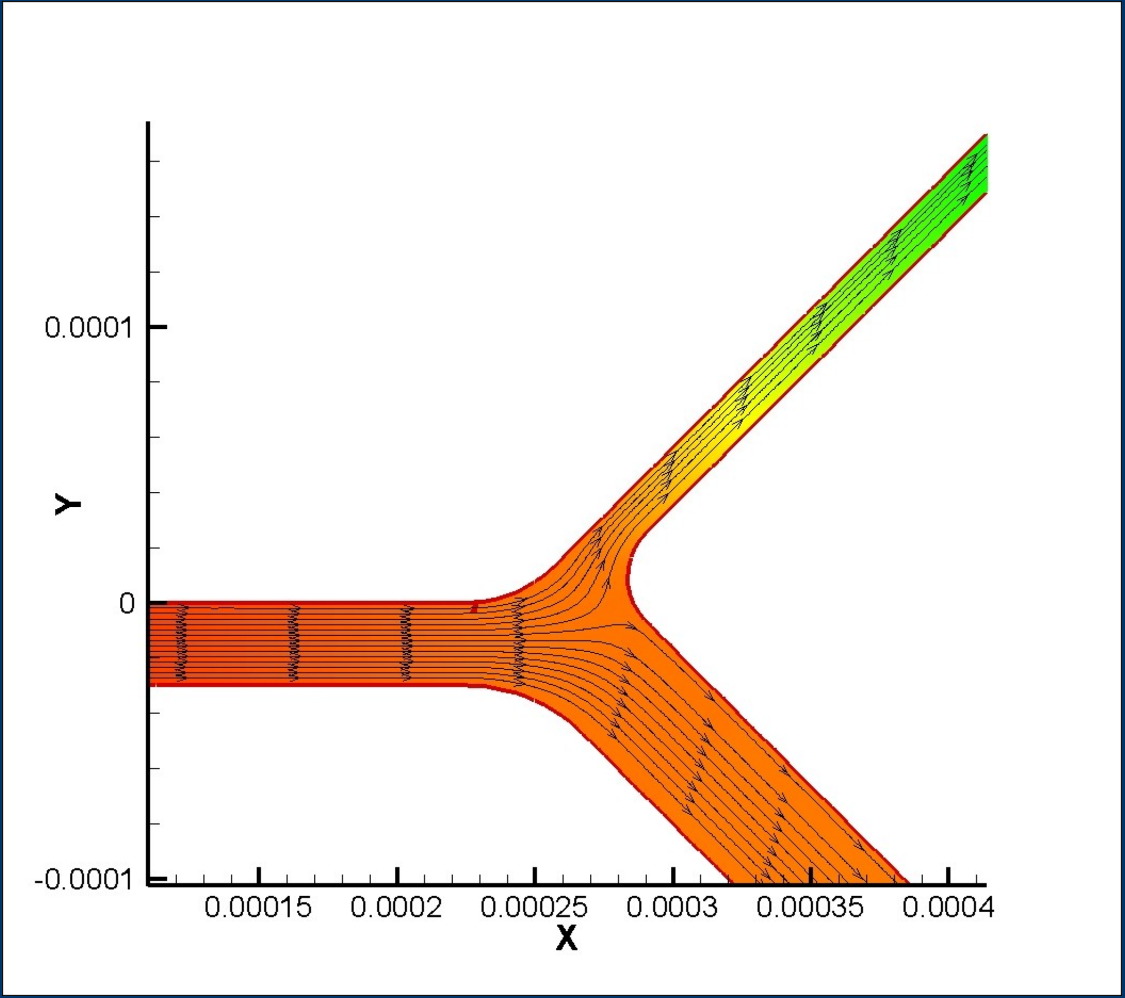


f2 = 0.1, θ2=45°, θ3=45°, d2/d1=0.5, d3/d1=1.5.



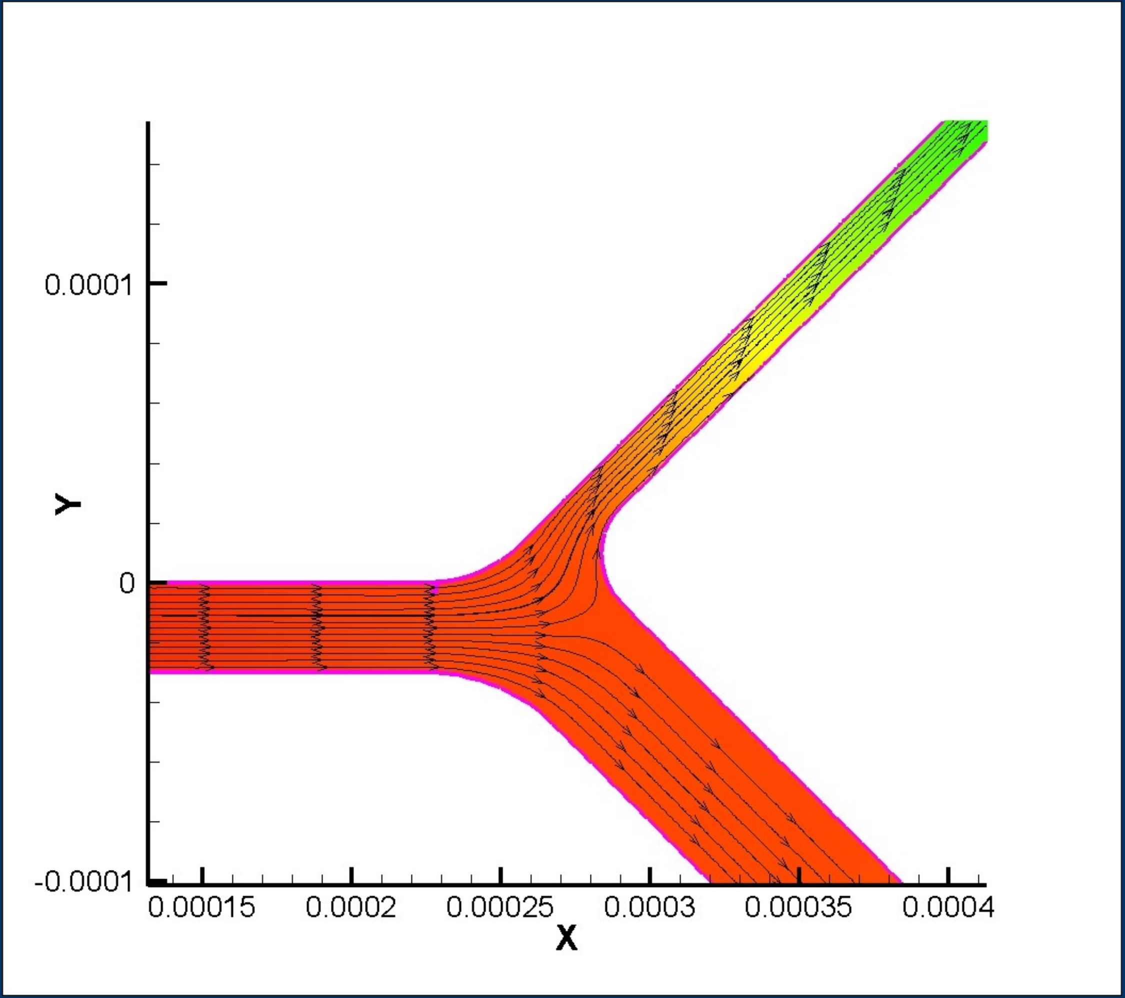
$K_2 = 5.47$

f2 = 0.3, θ2=65°, θ3=45°, d2/d1=0.5, d3/d1=1.5.



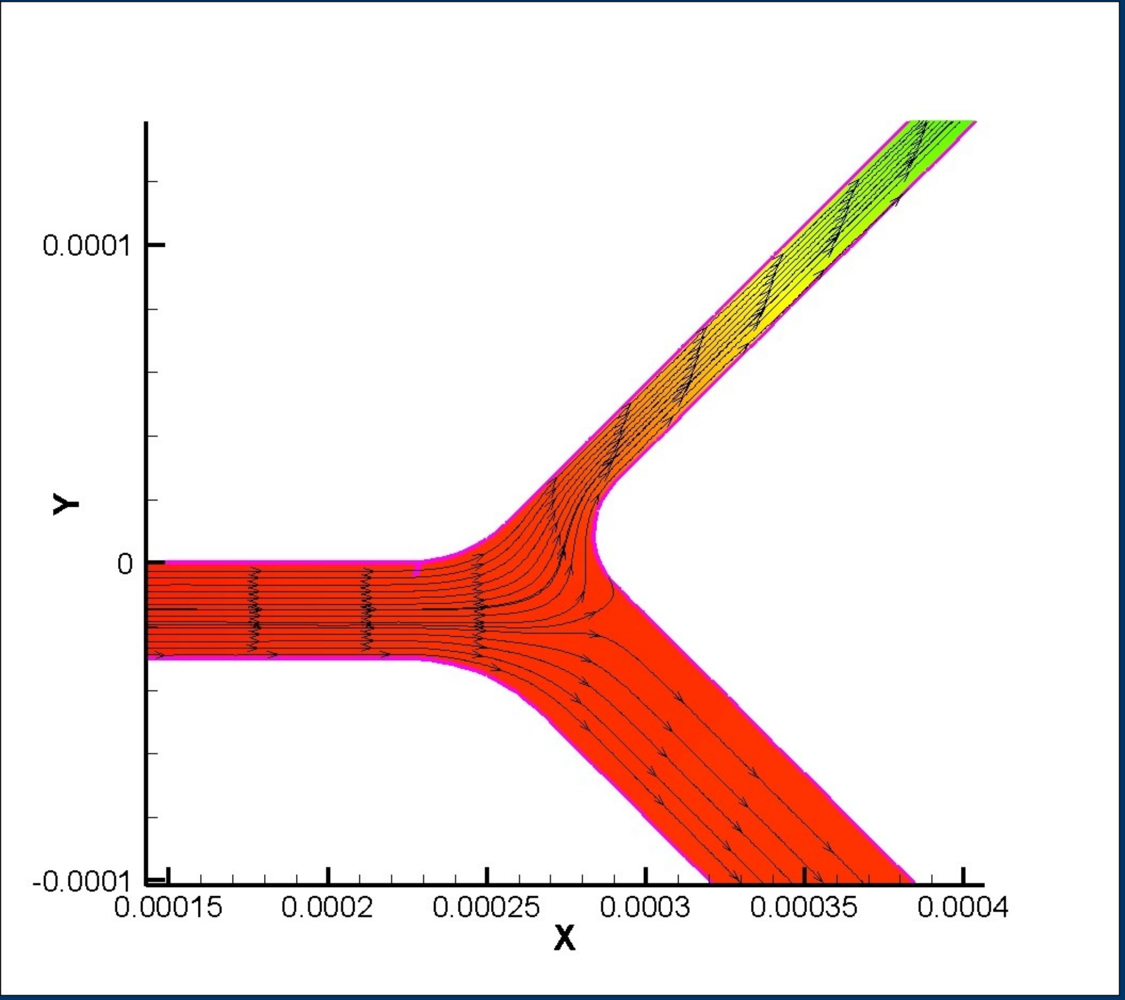
$K_2 = 11.6$

f2 = 0.5, θ2=65°, θ3=45°, d2/d1=0.5, d3/d1=1.5.



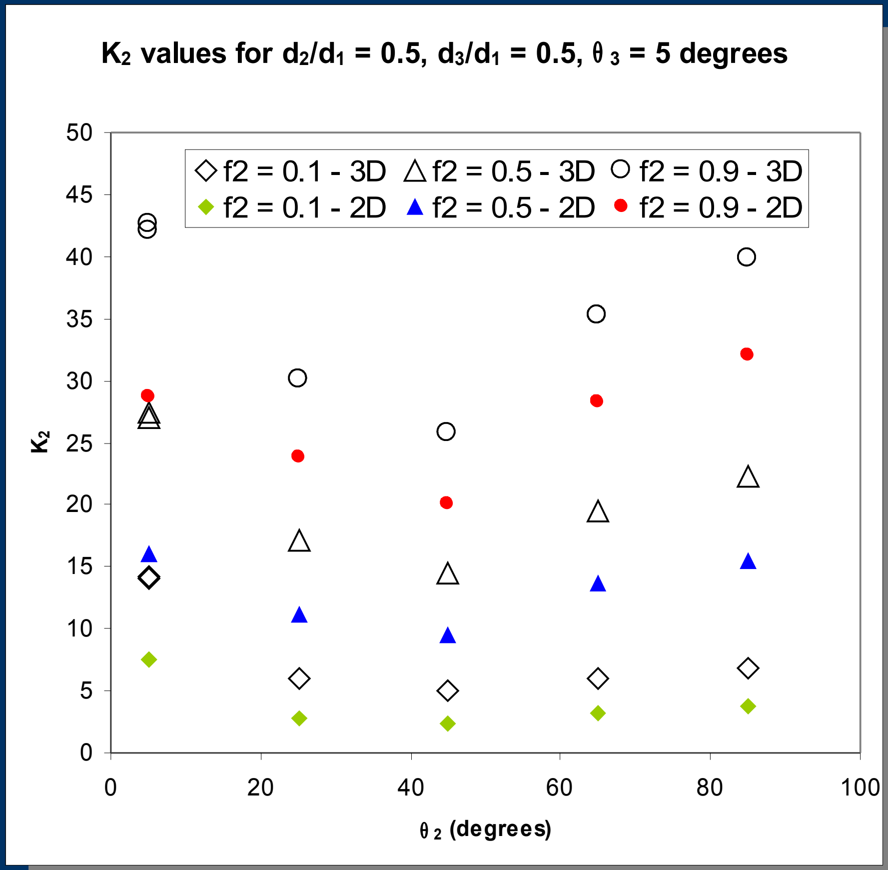
$K_2 = 18.4$

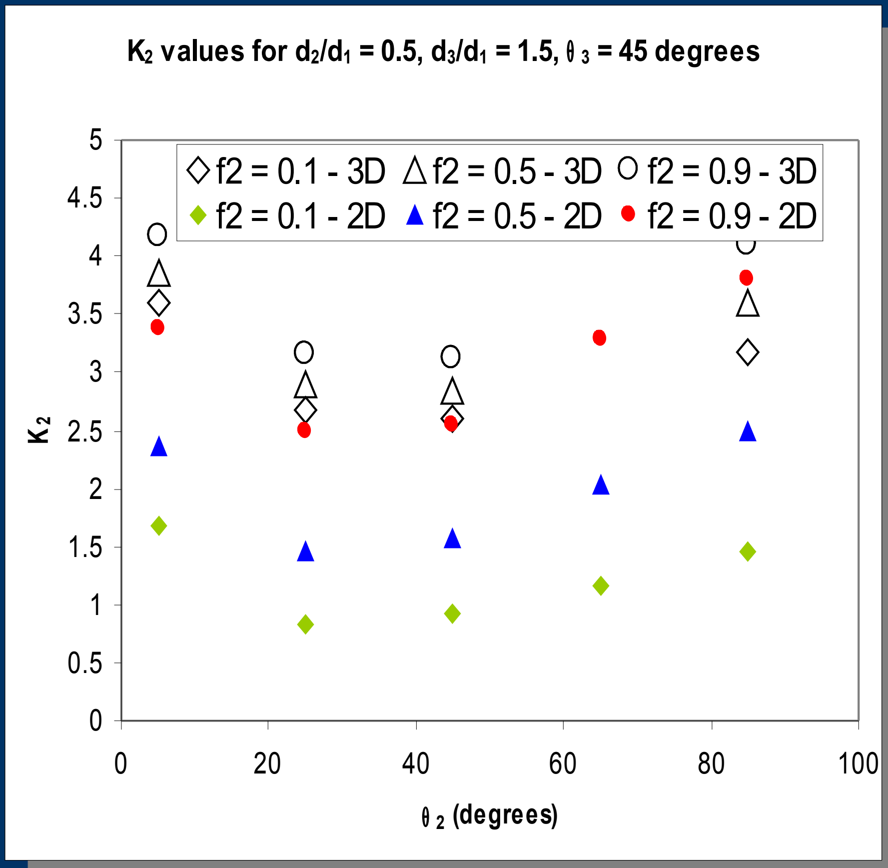
f2 = 0.7, θ2=65°, θ3=45°, d2/d1=0.5, d3/d1=1.5.



$K_2 = 25.7$

2D vs. 3D Differences





$$\left[\frac{k_2^{\,3D}}{k_2^{\,2D}}\right]_{average} = 1.55 \quad (std.\ \ dev.) = 0.45$$

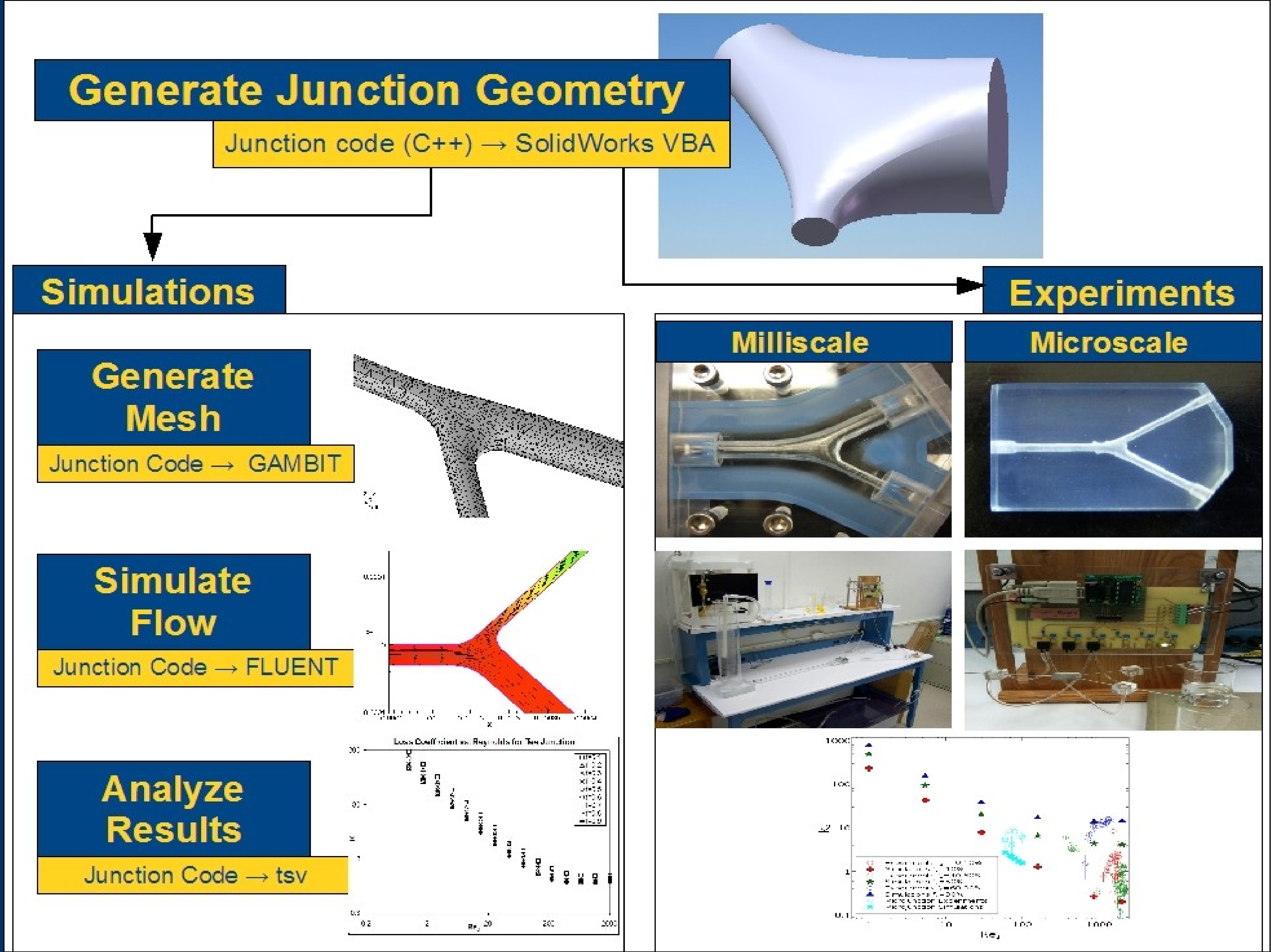
$$k_2 = \frac{\Delta p/\rho}{u_1^{\,2}/2} - \frac{u_2^{\,2}}{u_1^{\,2}} + 1$$

pressure effects

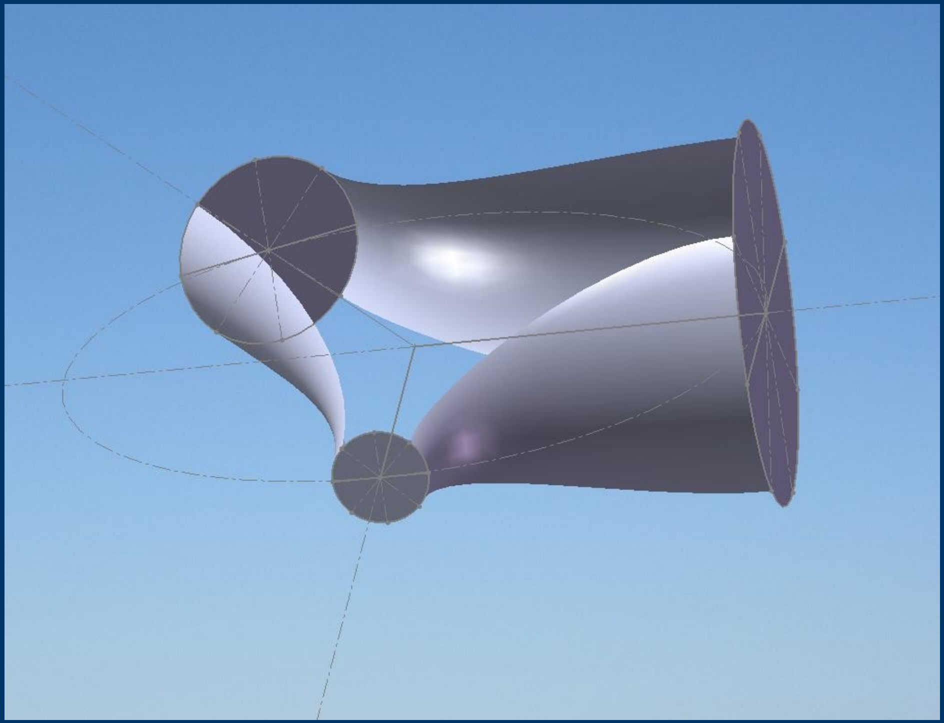
pressure effects

kinetic energy effects

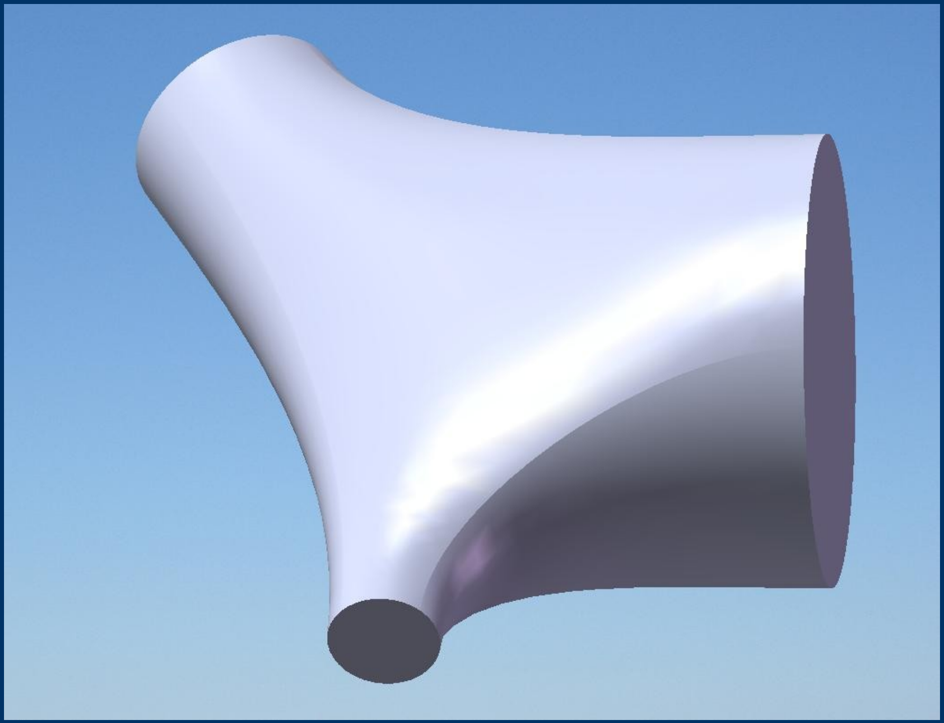
kinetic energy effects


Generate Junction Geometry
Junction code (C++) → SolidWorks VBA
Simulations
Generate Mesh
Junction Code → GAMBIT
Simulate Flow
Junction Code → FLUENT
Analyze Results
Junction Code → tsv
Experiments
Milliscale
Microscale

3D Mark II Junction



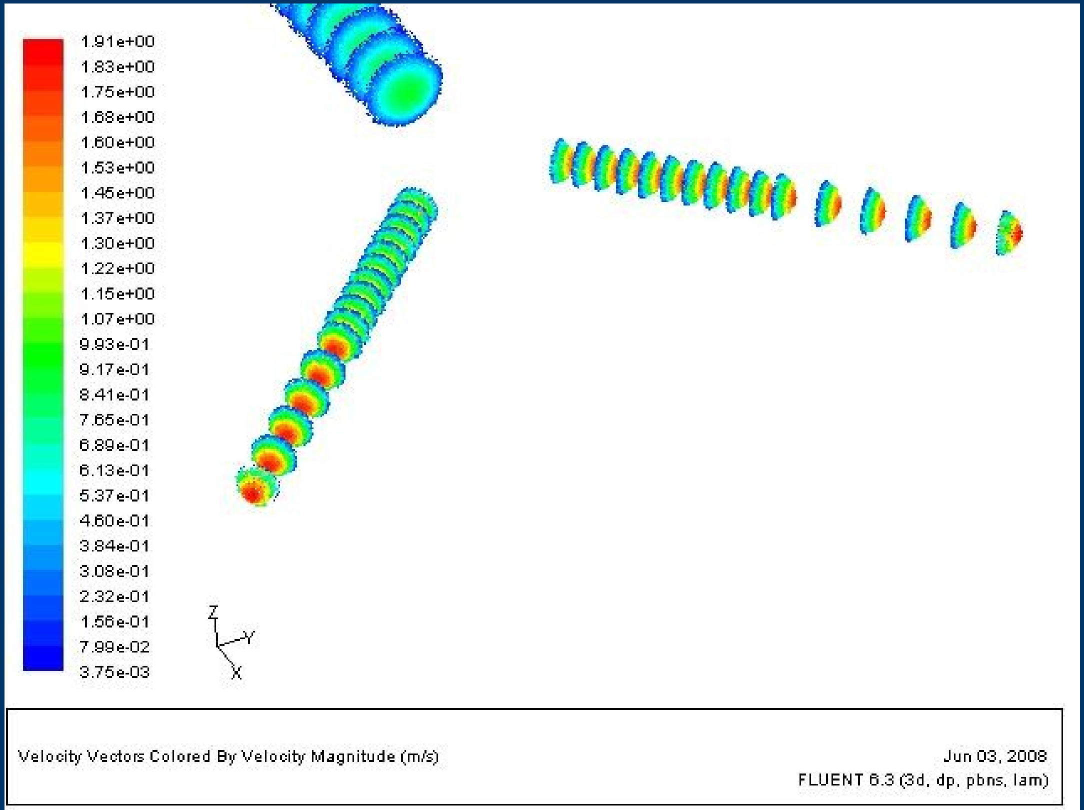
Junction Without Surface Fills



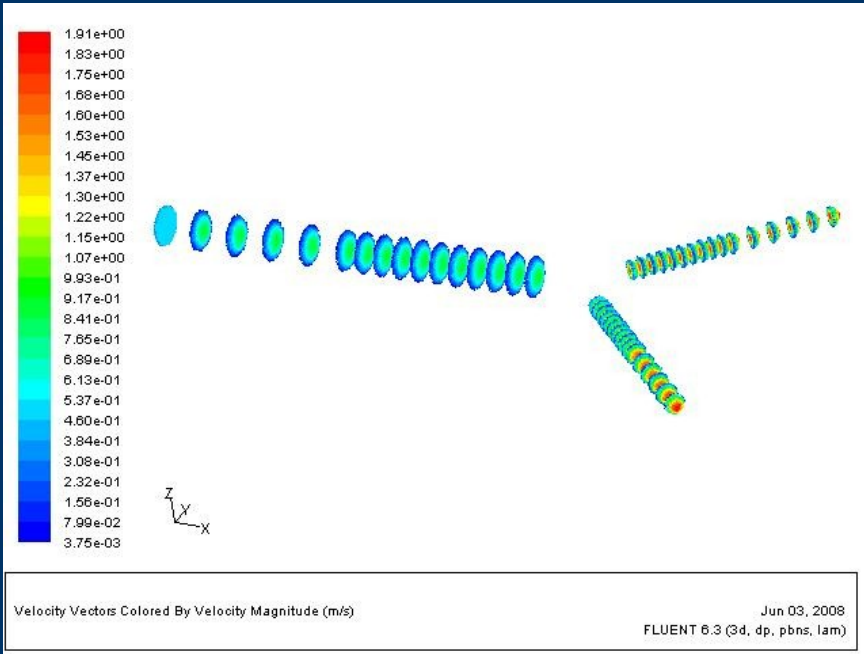
Junction With Surface Fills

Initial Simulations – 3D Mark II

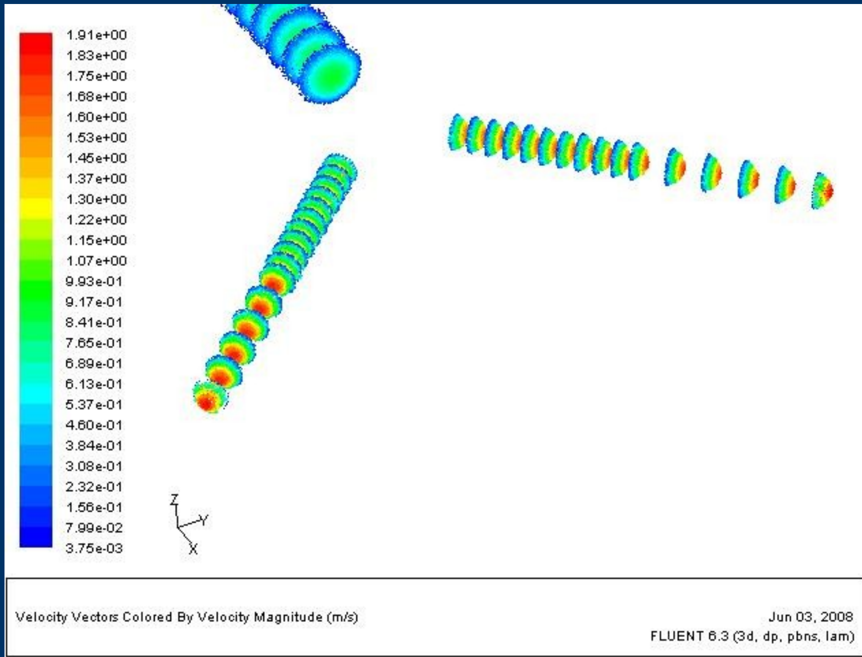
- Small batch
- Comparable results to original 3D runs



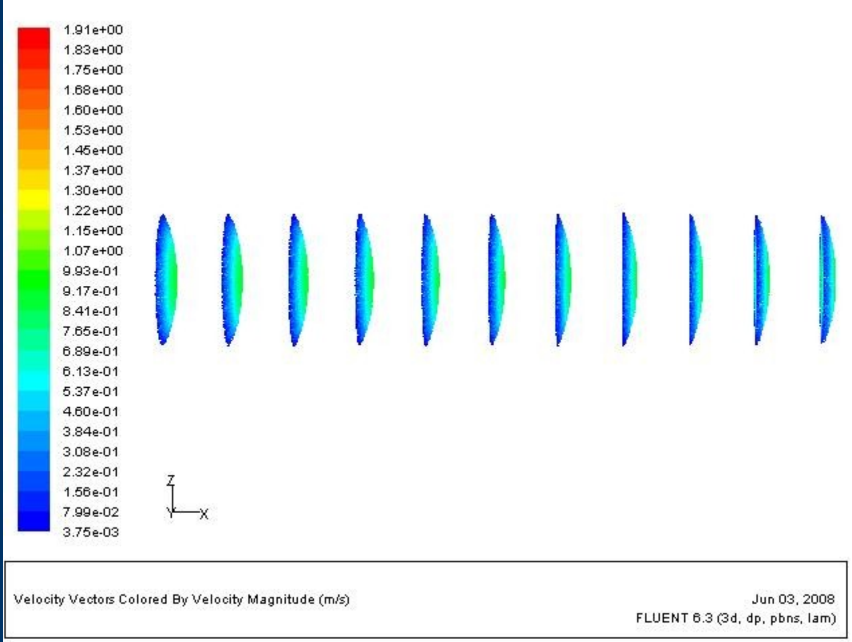
Reynolds number of 14 Inlet diameter of 30microns



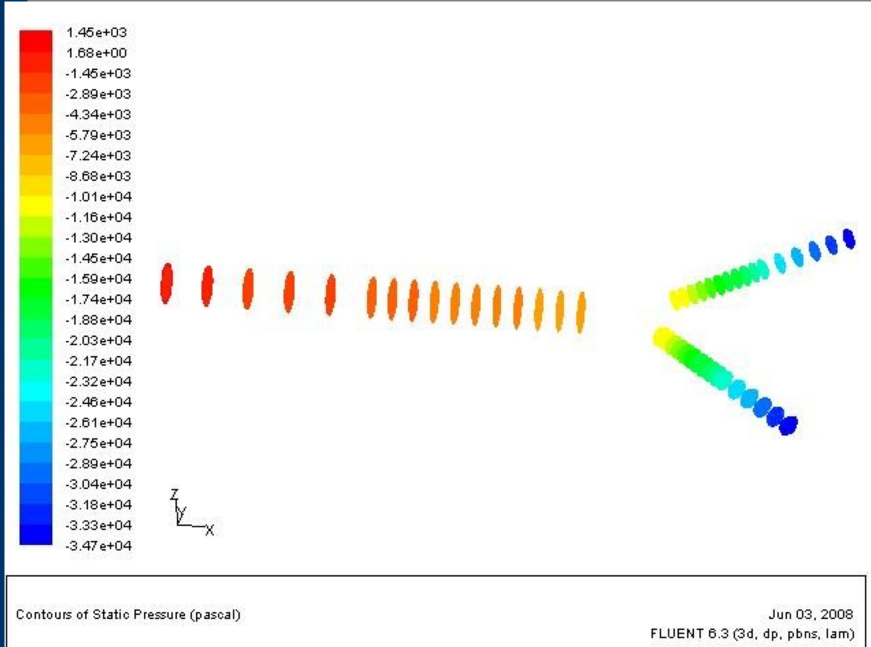
Reynolds number of 14 Inlet diameter of 30microns



Reynolds number of 14 Inlet diameter of 30microns



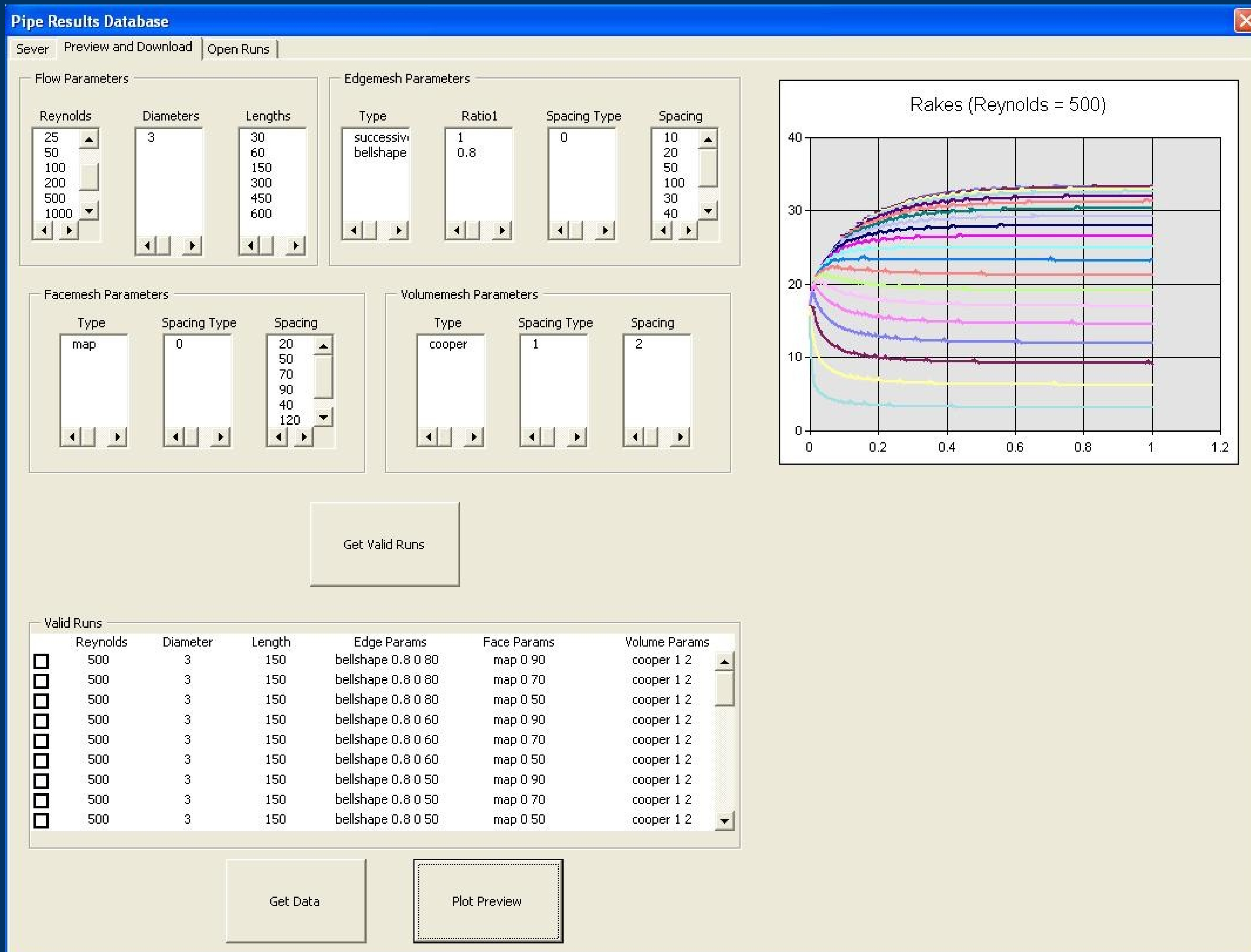
Reynolds number of 14 Inlet diameter of 30microns



Reynolds number of 14 Inlet diameter of 30microns

Pipe Mesh Testing

Database Interface



Pipe Mesh Testing

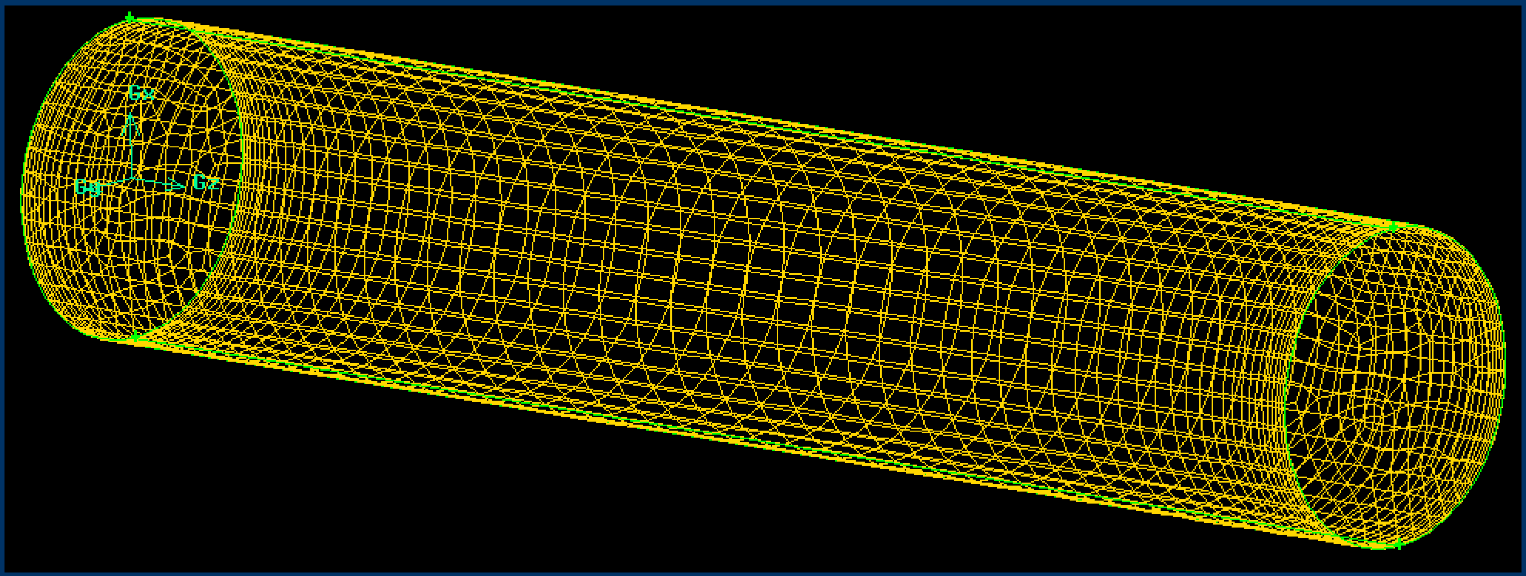
Flow-aligned hex core

Varied axial and radial spacing

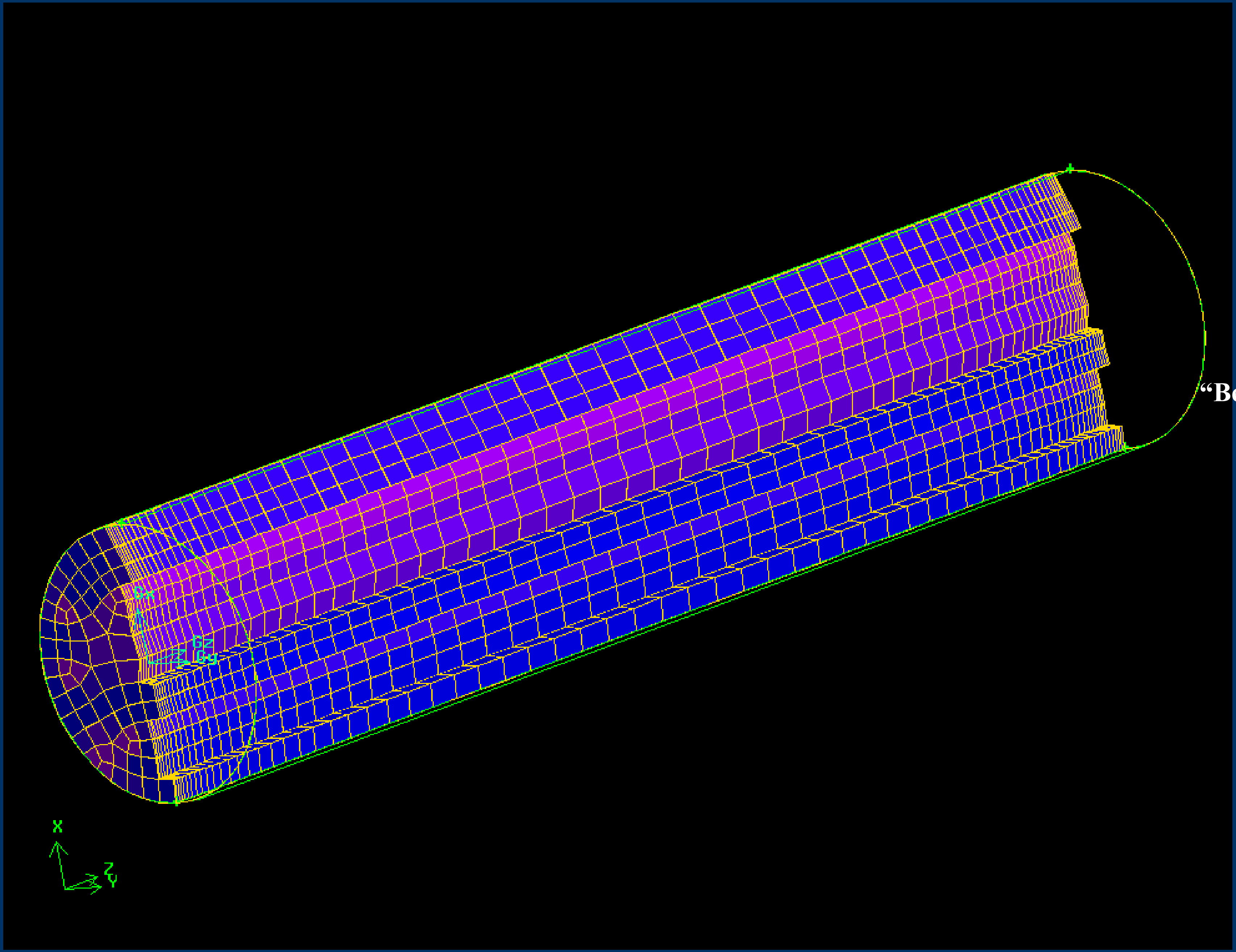
MySQL results database

Automated

Results database easy to setup for junction runs

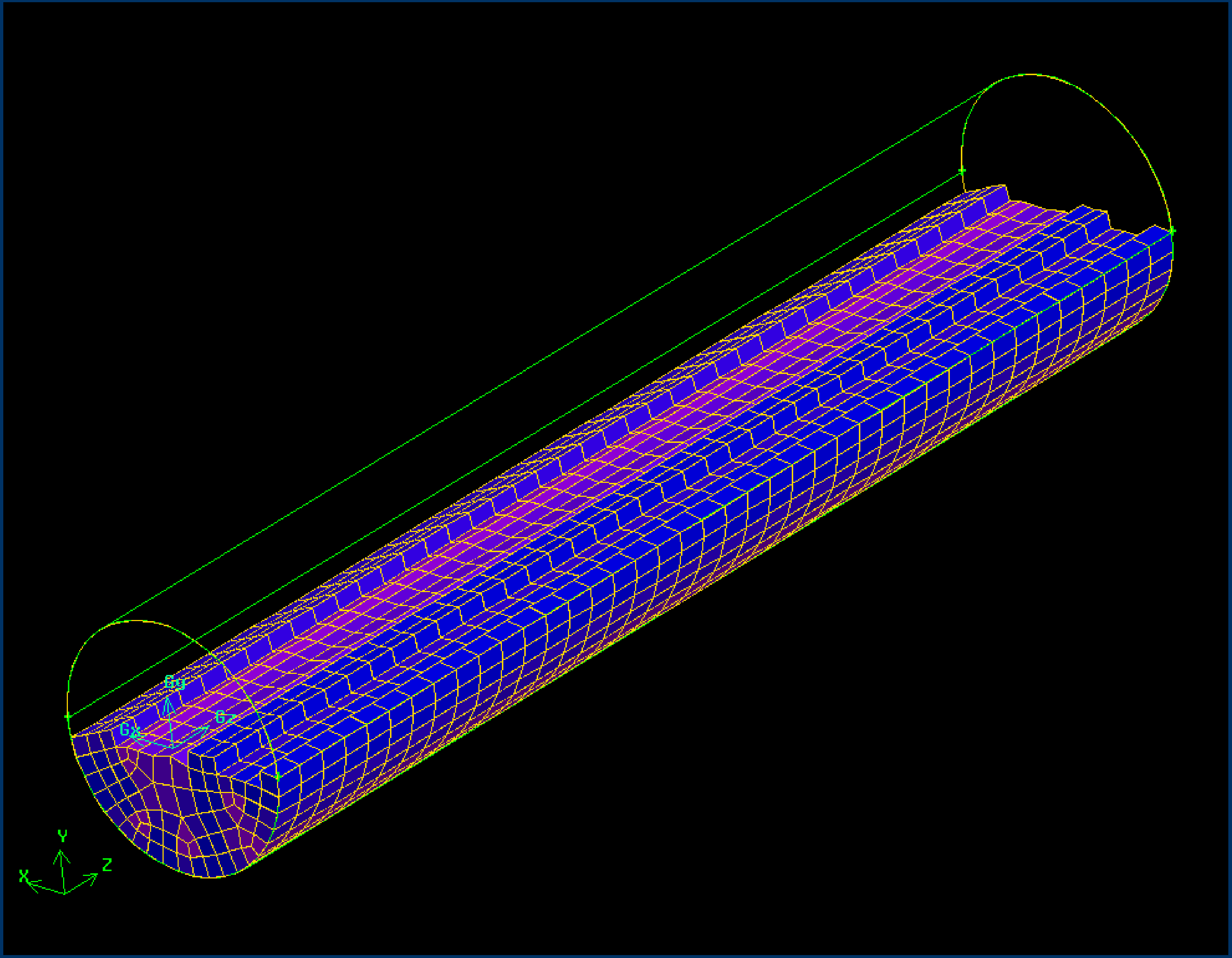


Pipe Mesh Testing



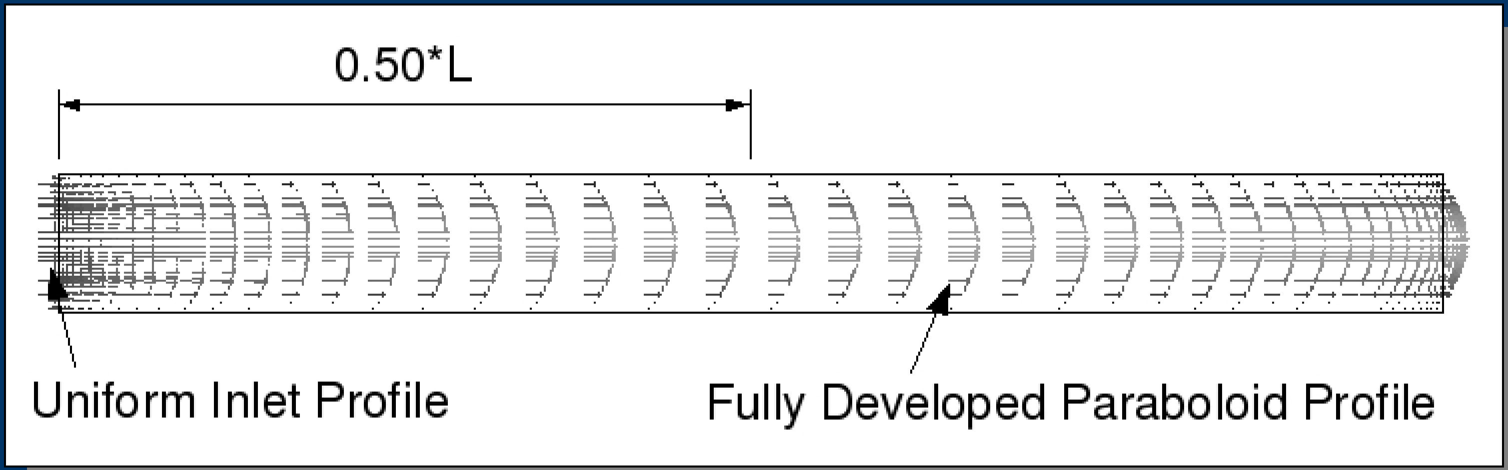
"Bell Shaped" Axial Spaci

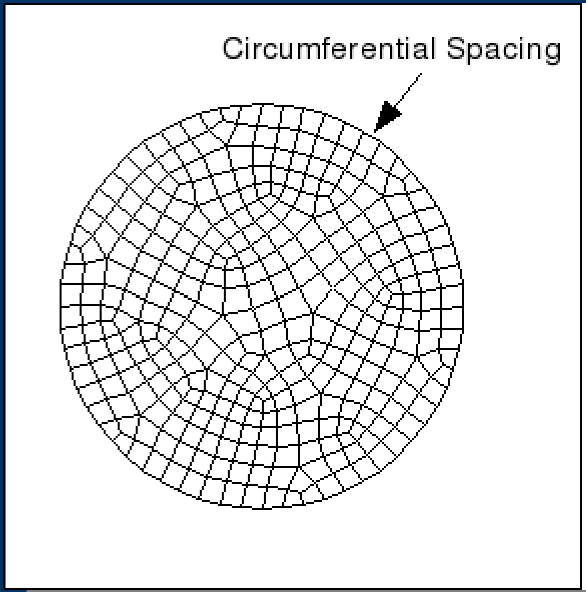
Pipe Mesh Testing

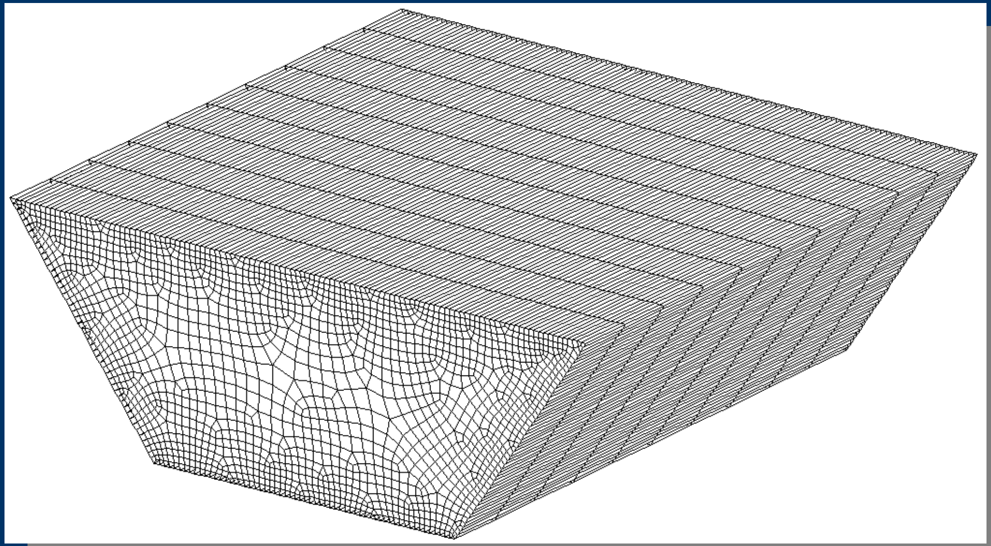


Uniform Axial Spacing

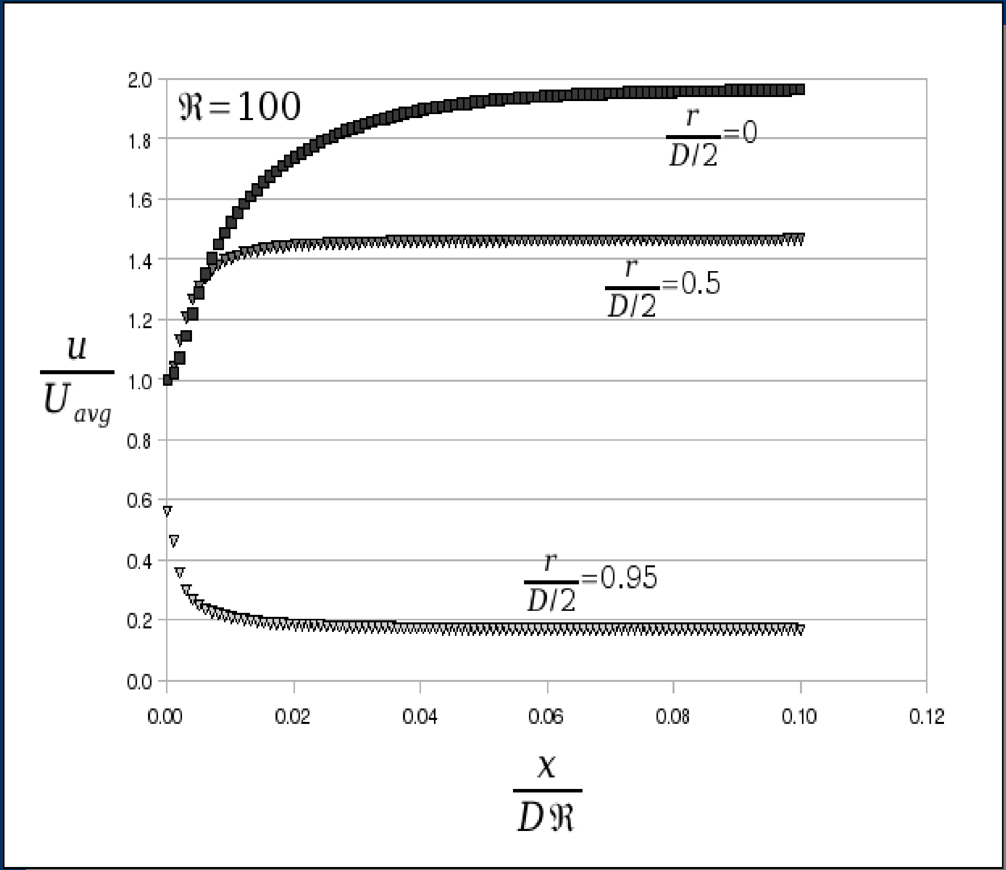
Entrance Length in Microtubes

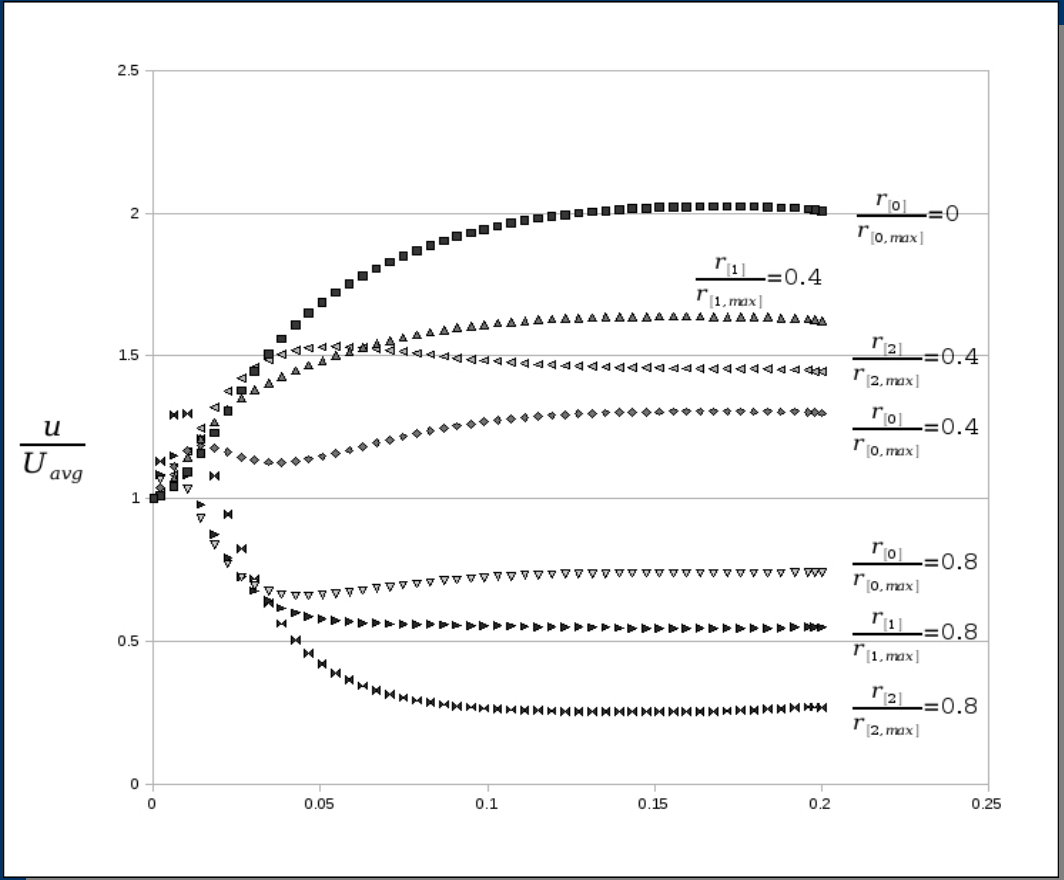


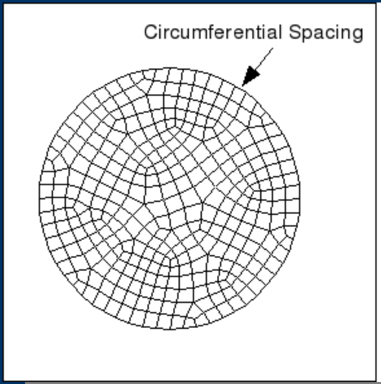


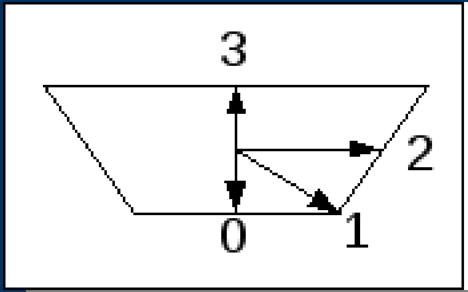


Entrance Length Results

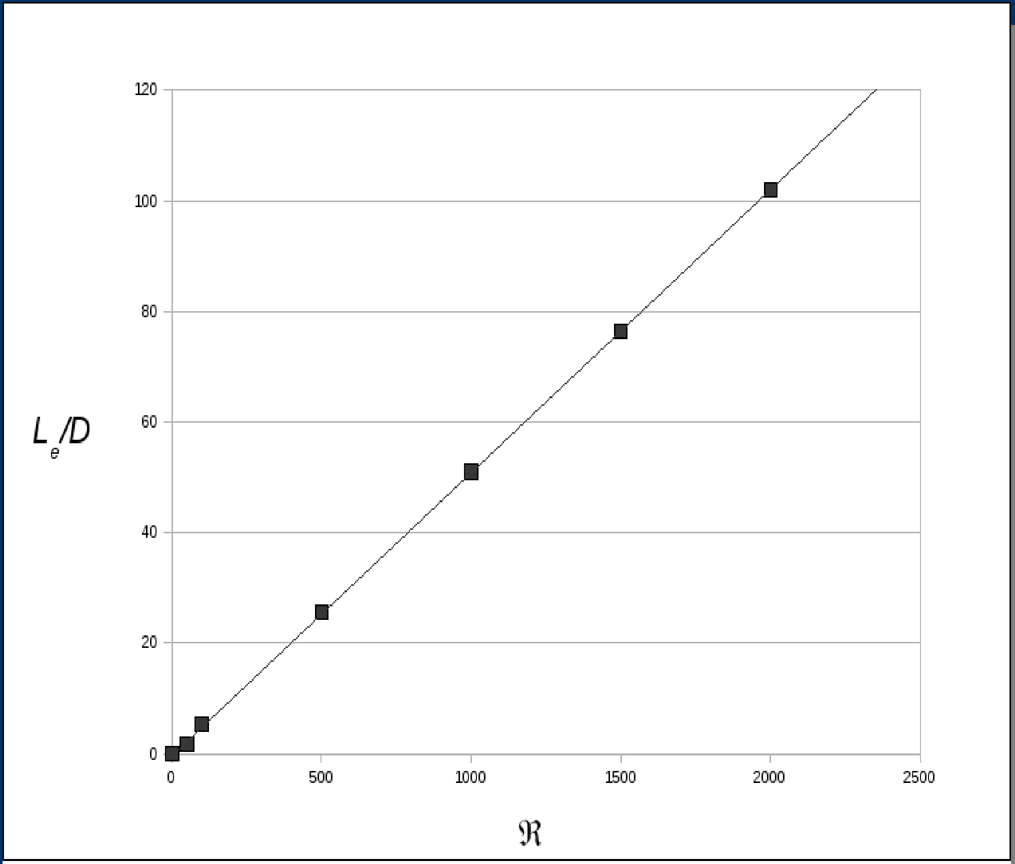


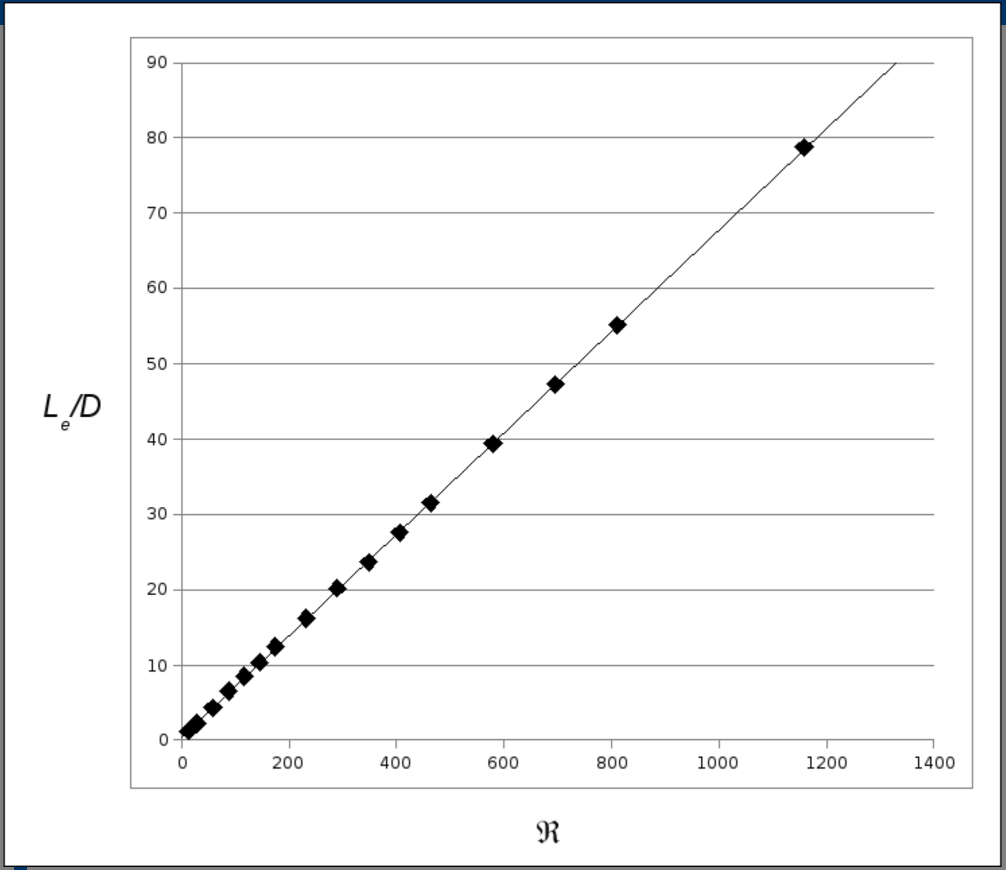




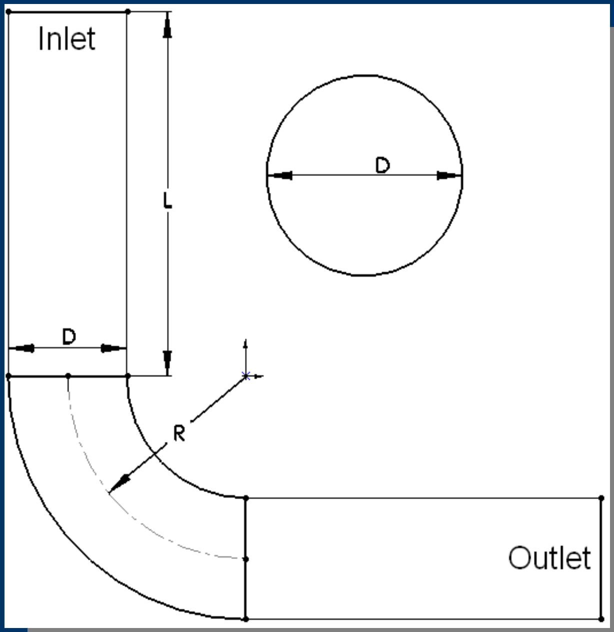


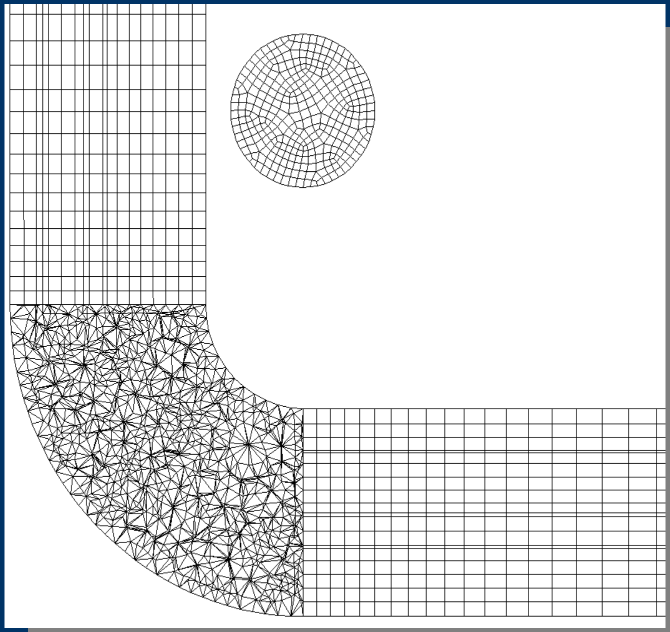
Entrance Length Results

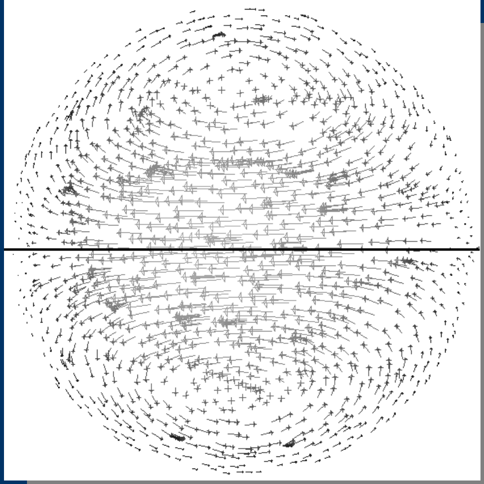


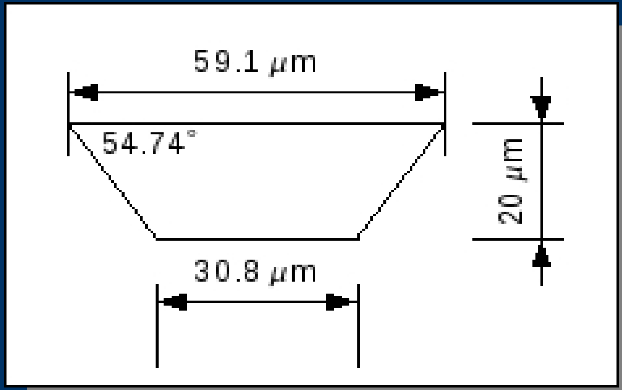


Loss Coefficients in Microelbows





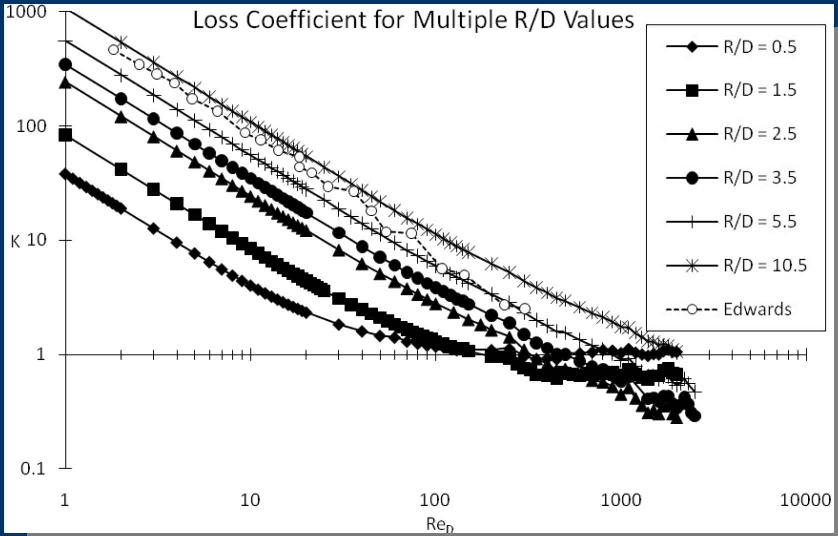


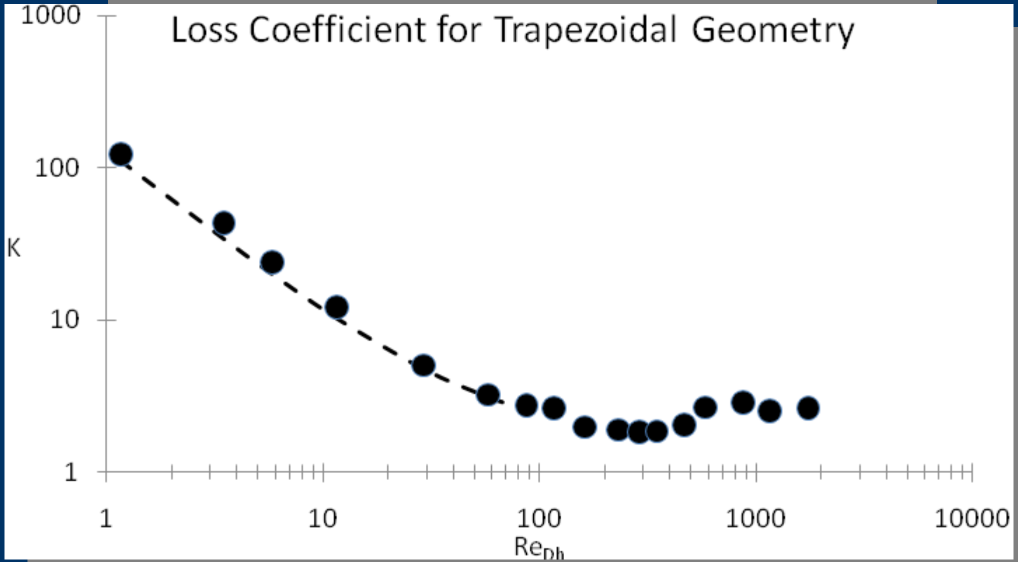


$$K = \frac{\Delta p}{\frac{1}{2}\rho V^2}$$

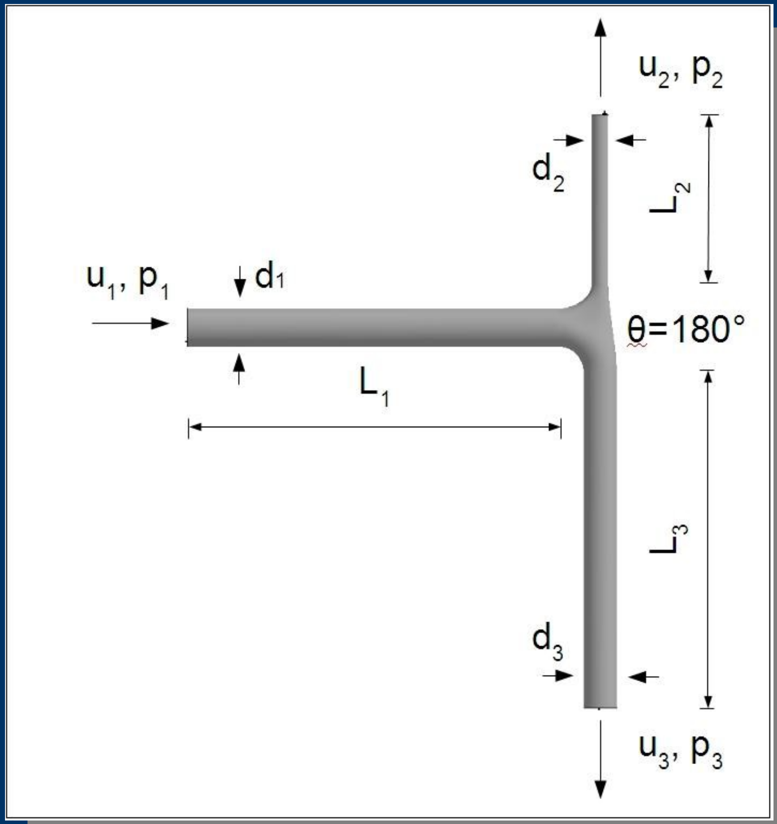
Microelbow Loss Coefficients

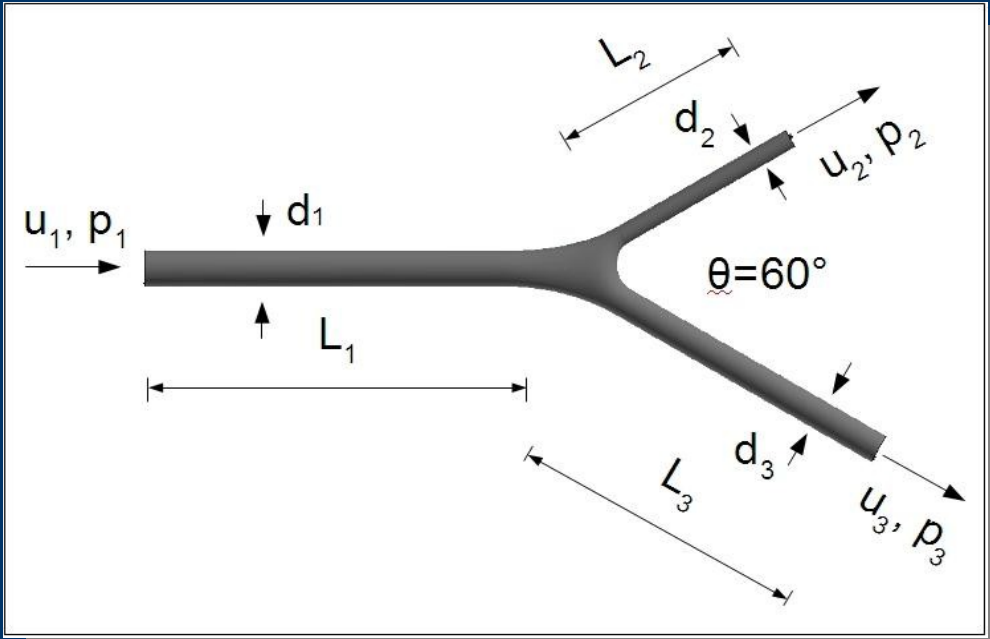
$$K = \frac{\Delta p}{\frac{1}{2}\rho V^2}$$



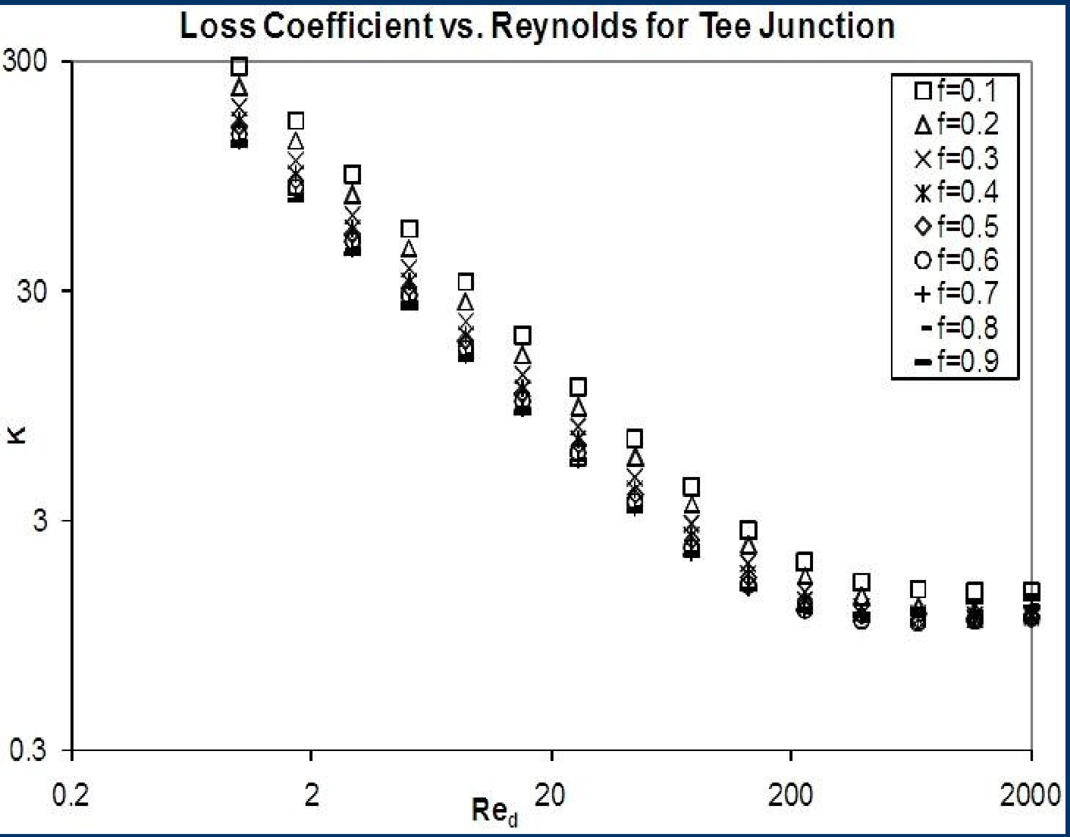


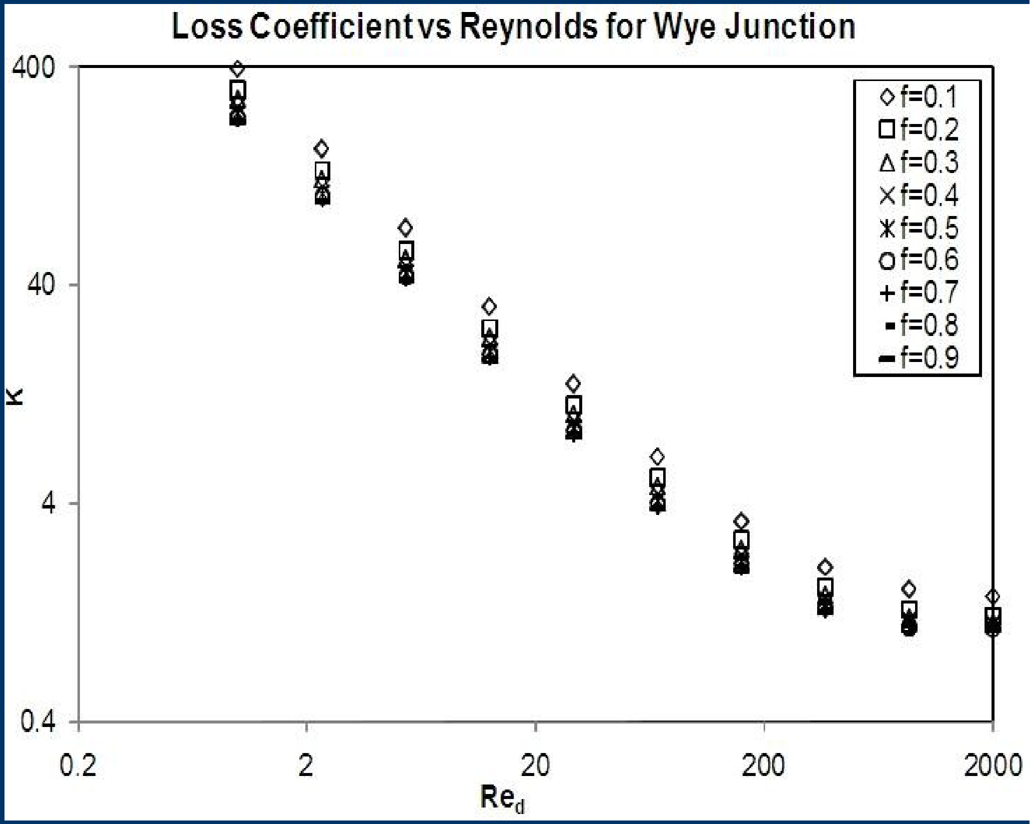
Tee and Wye Reynolds Number Dependence





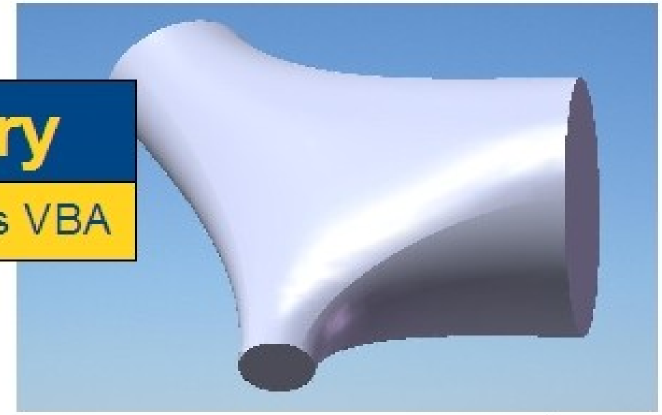
Loss Coefficient Versus Re = proportional to speed





Generate Junction Geometry

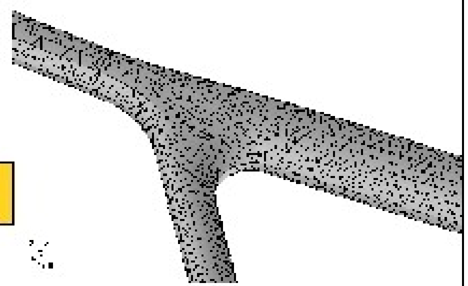
Junction code (C++) → SolidWorks VBA



Simulations

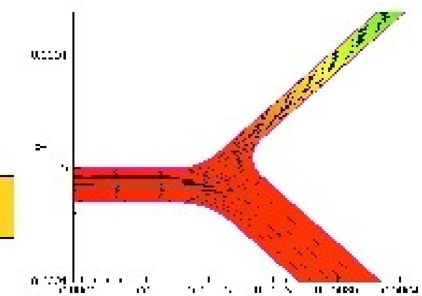
Generate Mesh

Junction Code → GAMBIT



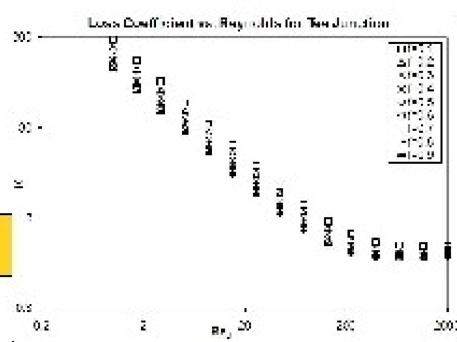
Simulate Flow

Junction Code → FLUENT



Analyze Results

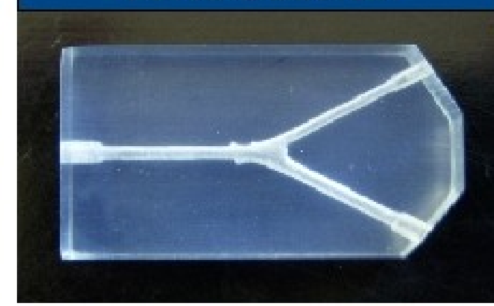
Junction Code → tsv

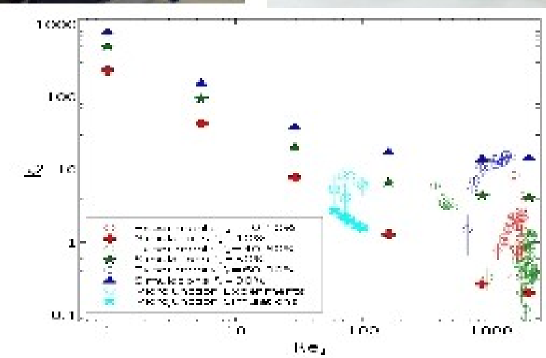


Experiments

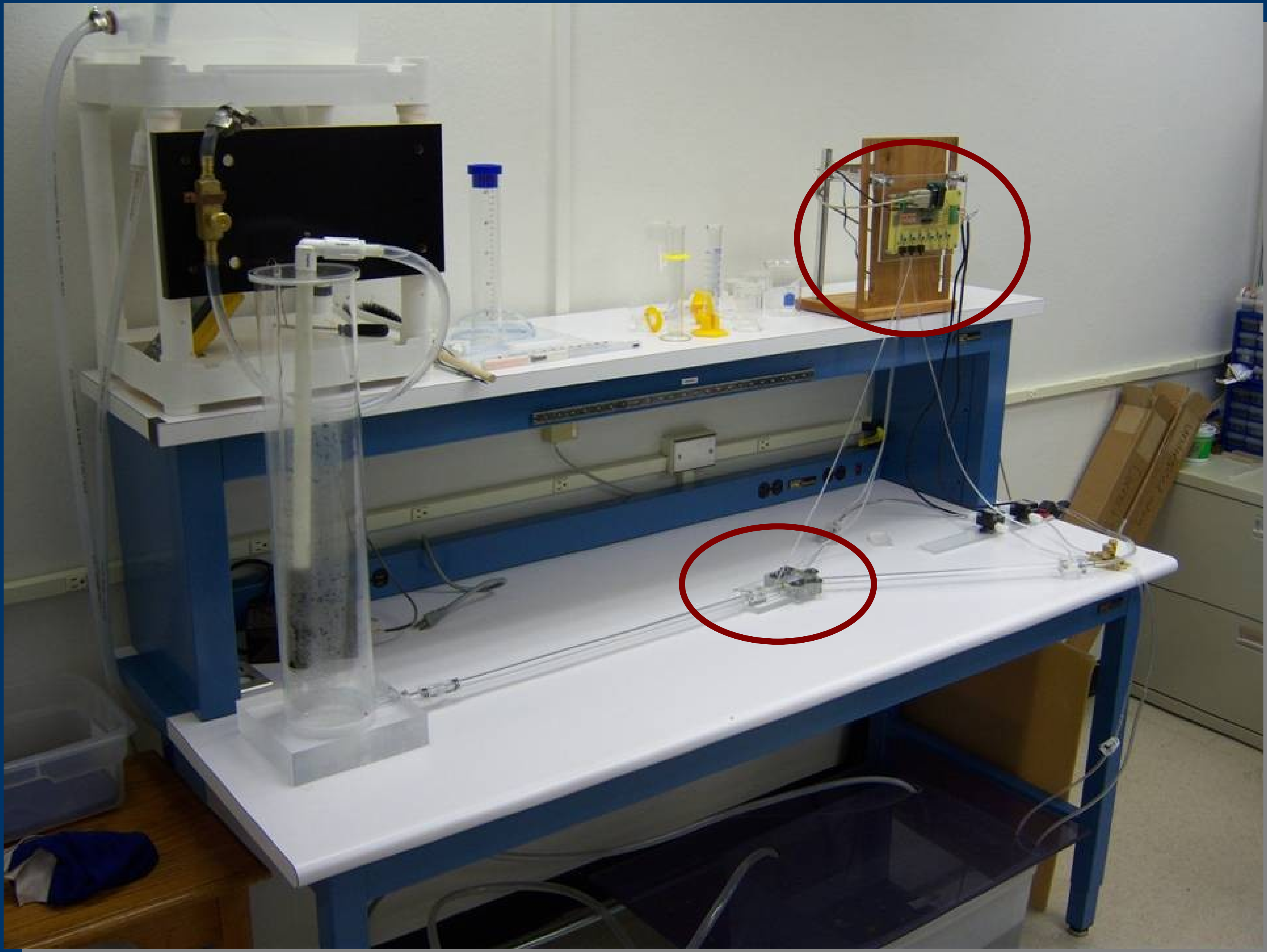
Milliscale

Microscale

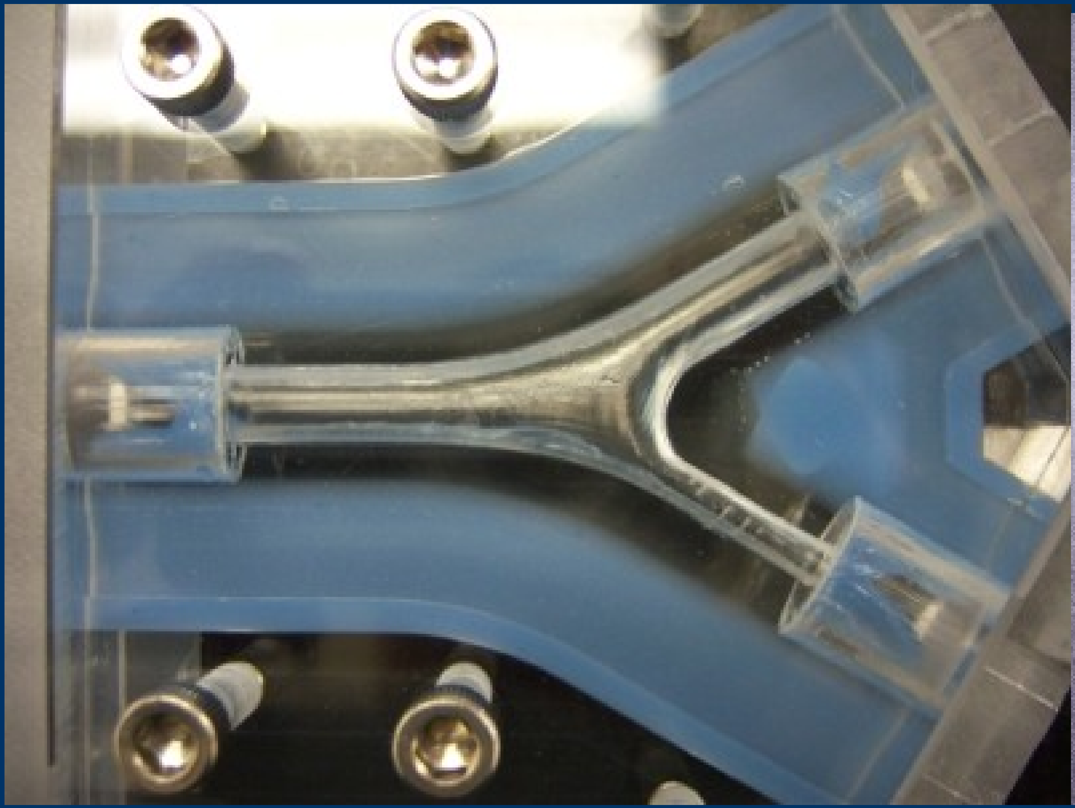


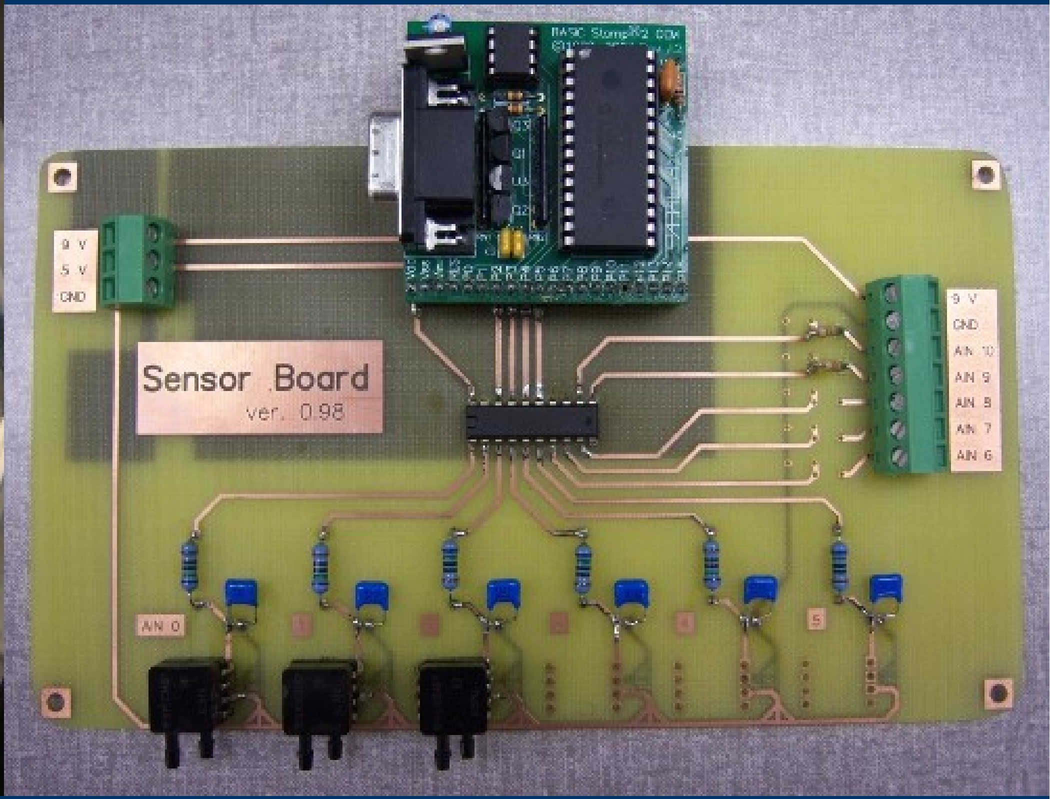


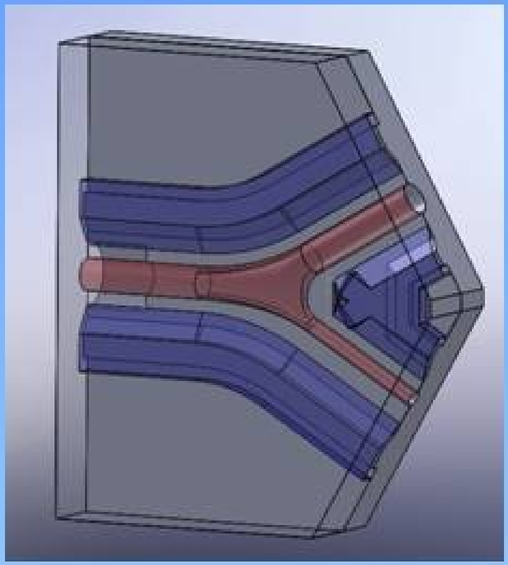
Millijunction Experiments



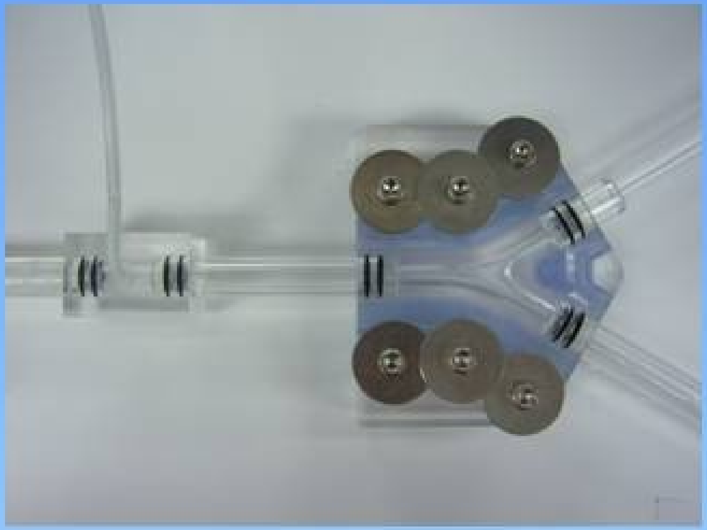
Junction and Measurements







Here is the Solidworks model as well as the physical realization of our efforts to create a useable bifurcation.



Upstream from the head tank and downstream from the junction are our new, highly sensitive needle valves used to adjust the head tank flow and the flow fraction through the bifurcation.

The flow straightener we designed has a replaceable core made of glass micropipettes that can be removed completely for unrestricted flow.

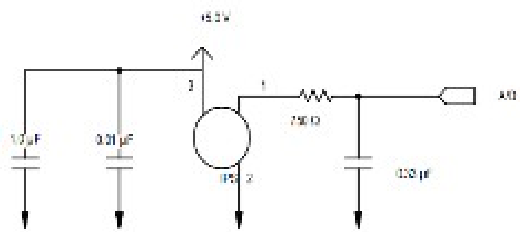
Collecting Data

Sensor

- Submersible
- Internal Instrumentation Op-Amp

Filter

- 750Hz Low Pass
- Reduces Noise from 10mV to 2mV



ADC

- 12 bit
- 11 Channel

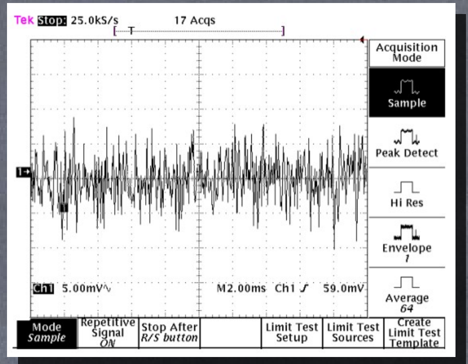
Basic Stamp

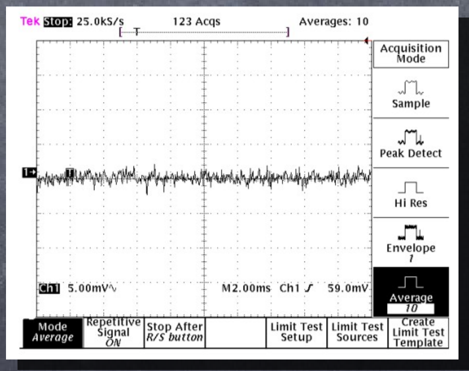
- Easy to program
- Affordable

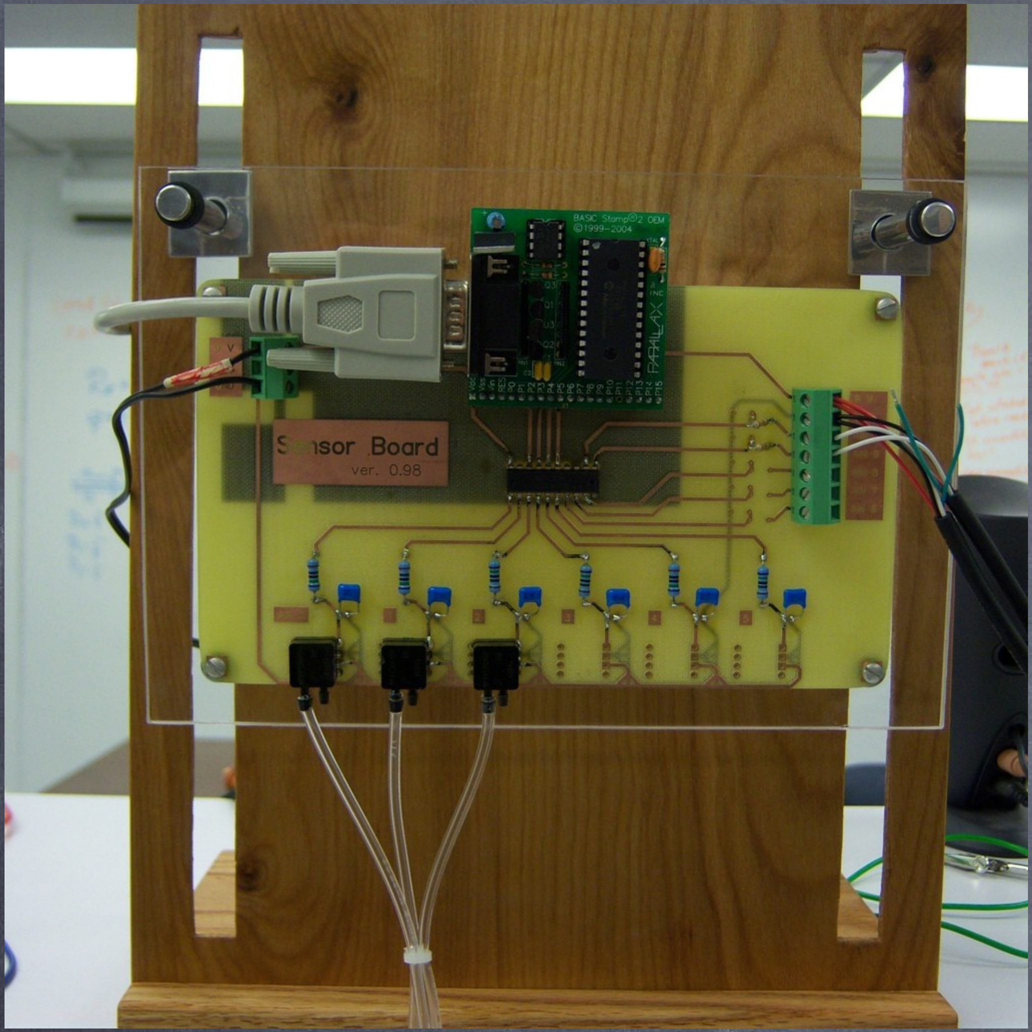
Computer

- Python
- Real Time Plotting with GNU Plot

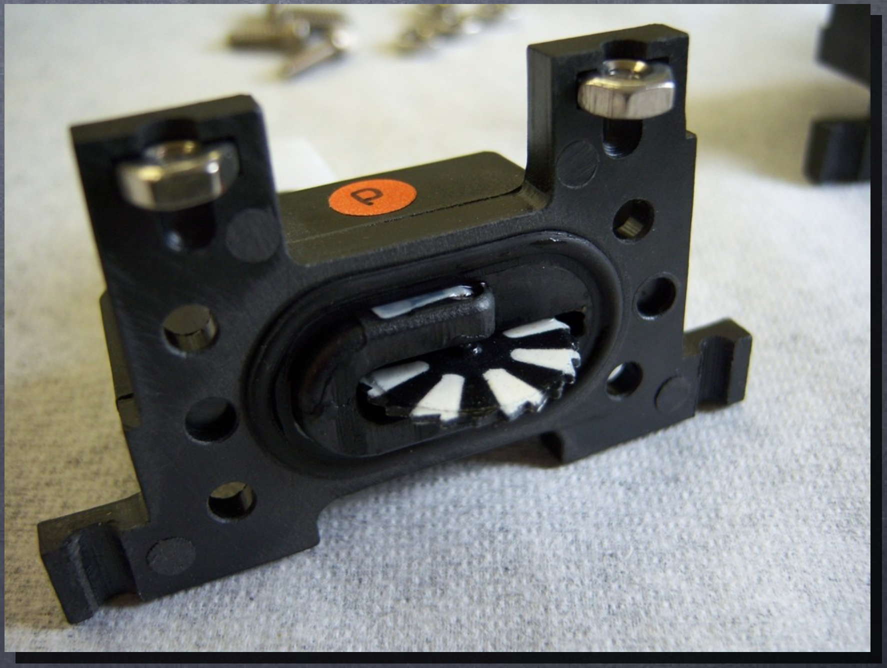
```python
if __name__ == '__main__':
    serial = connect_serial()
    filesave = csv.writer(open("data.csv", "wb"))
    while True:
        data = serial.readline().split()
        filesave.writerow(data)
        plot(data)
```

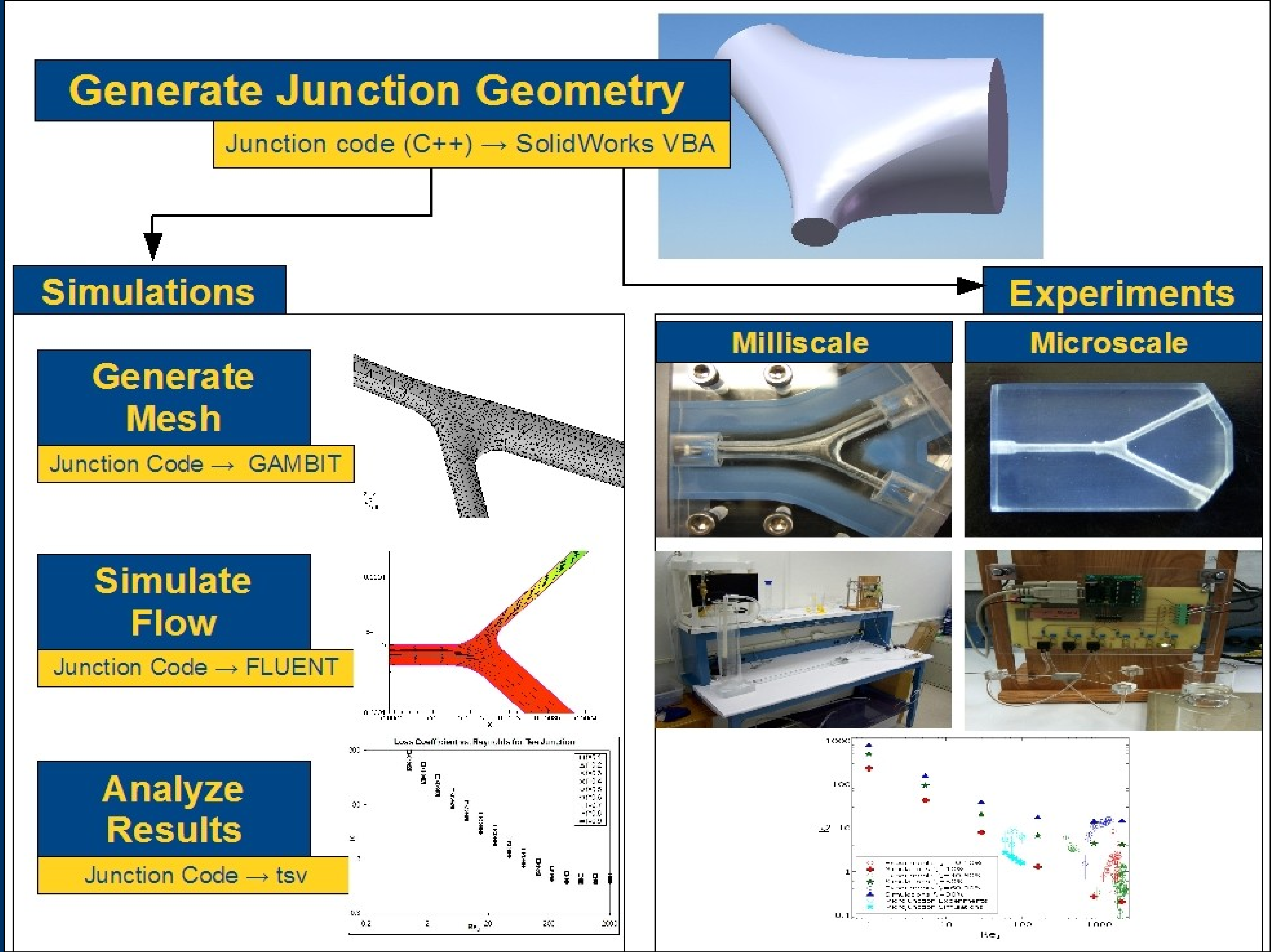






Basic Stamp
Passive Filter
Pressure Sensors
Flow Sensors
Ruby Interface



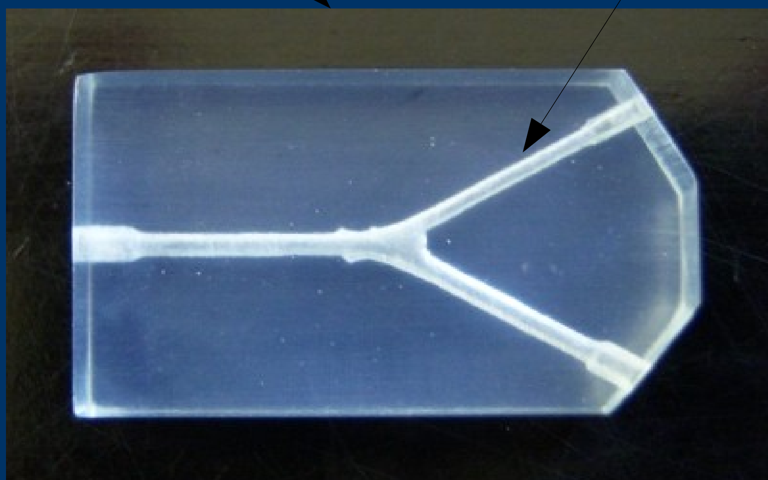


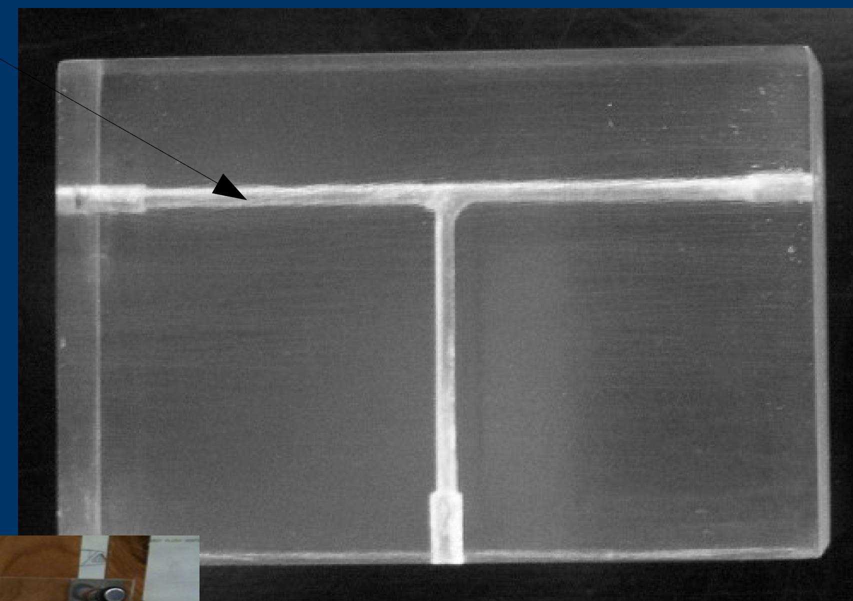
Generate Junction Geometry
Junction code (C++) → SolidWorks VBA
Simulations
Generate Mesh
Junction Code → GAMBIT
Simulate Flow
Junction Code → FLUENT
Analyze Results
Junction Code → tsv
Experiments
Milliscale
Microscale

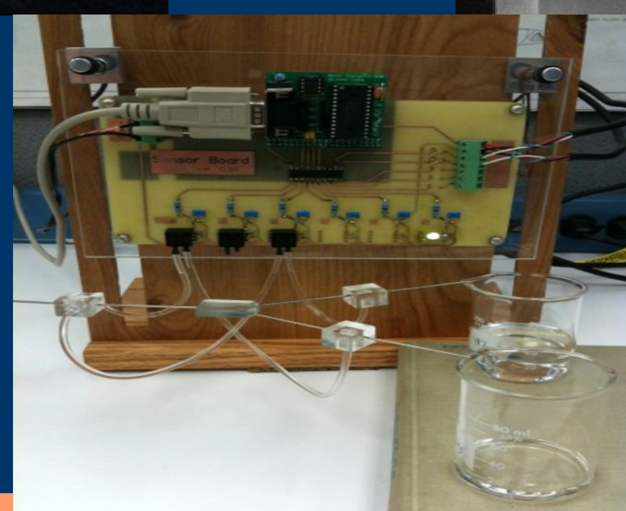
Microscale Junction Experiments

Stereo
Lithography
Technique

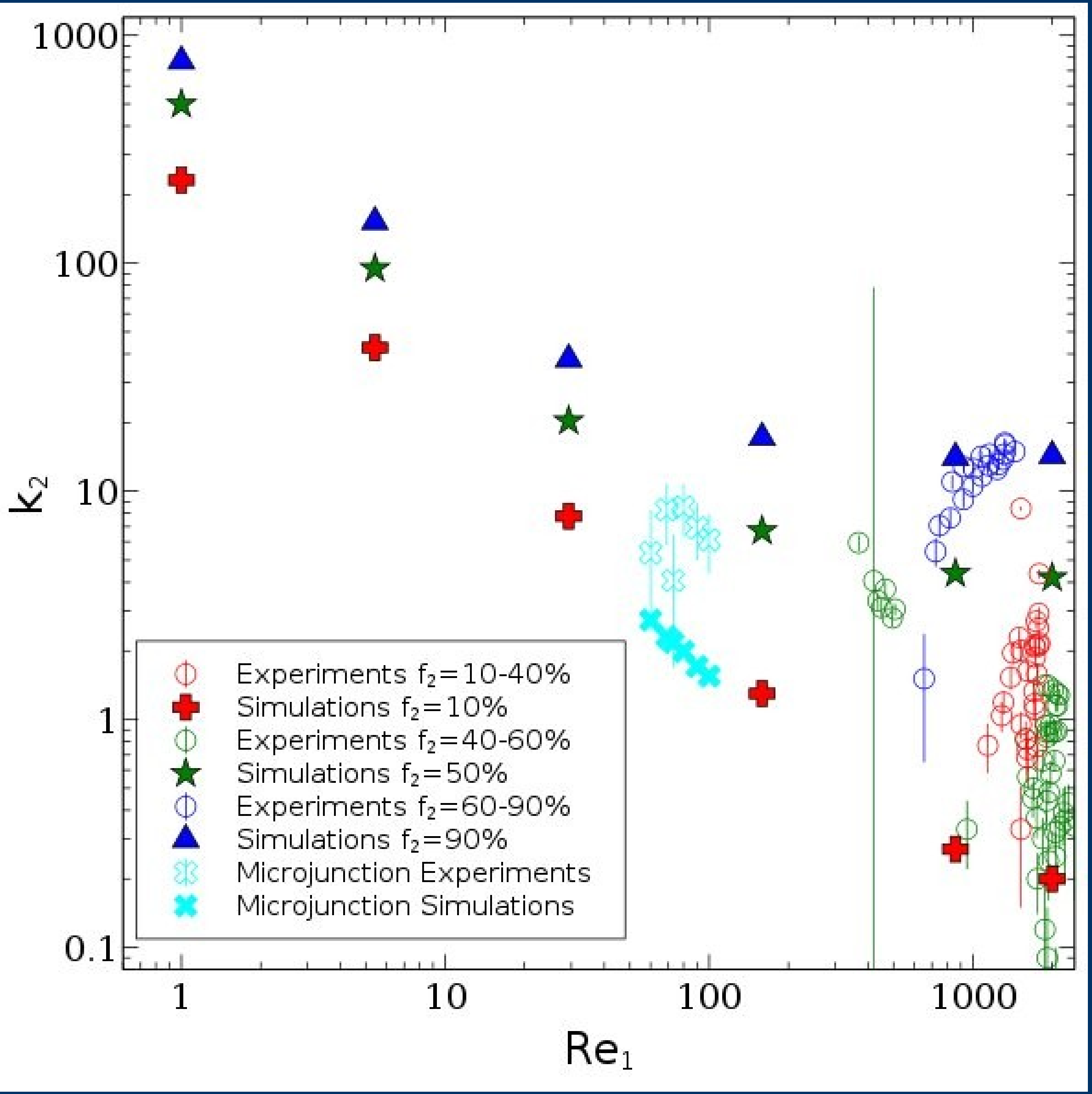
Diameter =
0.75 mm





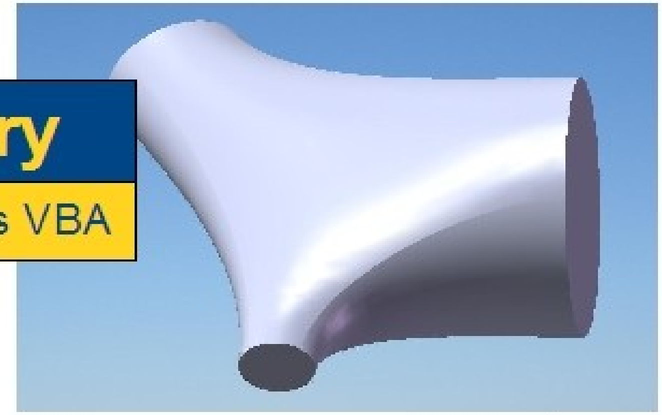


Results - Simulations/Milliscale/Microscale



Generate Junction Geometry

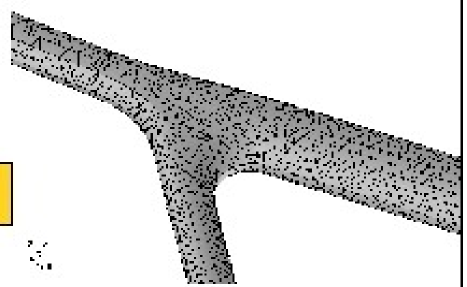
Junction code (C++) → SolidWorks VBA



Simulations

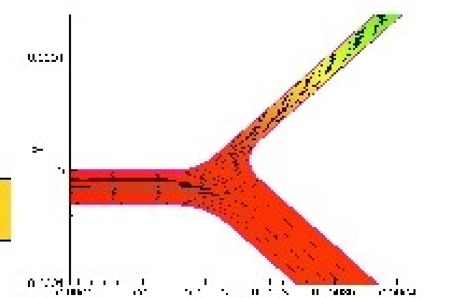
Generate Mesh

Junction Code → GAMBIT



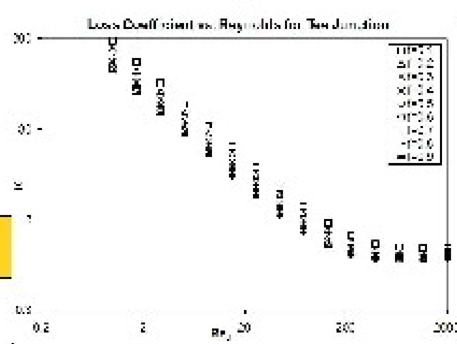
Simulate Flow

Junction Code → FLUENT



Analyze Results

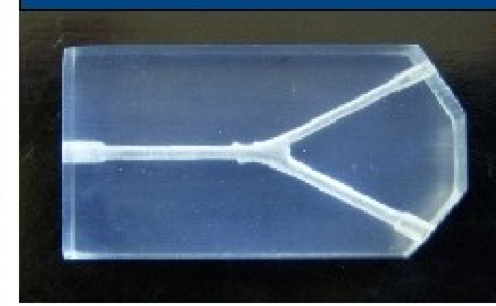
Junction Code → tsv

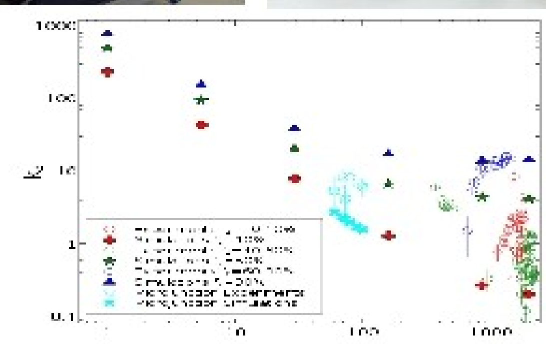


Experiments

Milliscale

Microscale





Using Network Simulations to Understand Non-Darcy Flow

Objectives

Develop an algorithm to create, mesh, and perform CFD on simplified models of real porous media networks.

Equal number of entry and exit pores (no splits)

90° elbows only

No overlap of pores within media

Compare results of CFD to FTPM and empirical data in literature.

Modify algorithm to allow for complex models of real porous media.

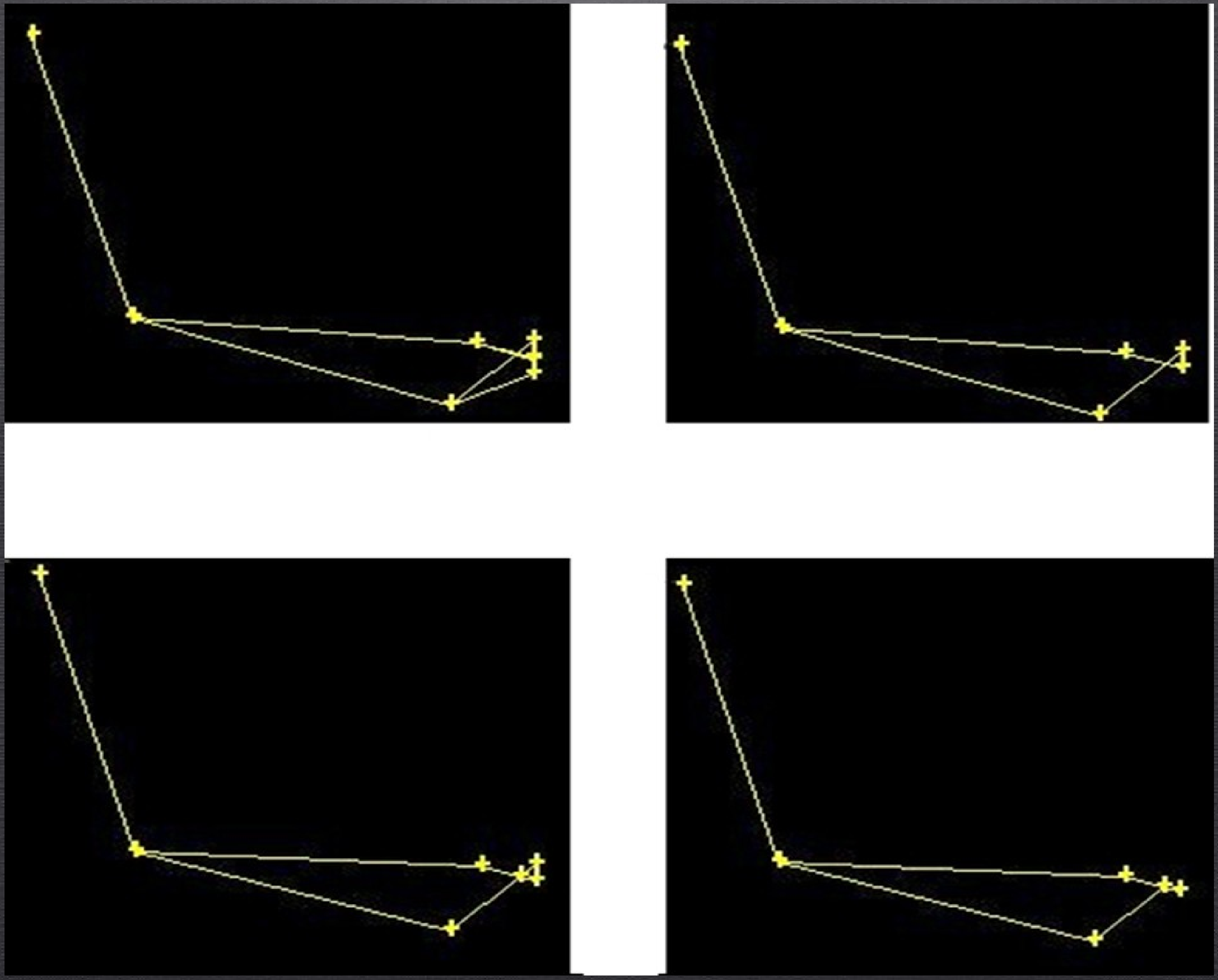
Splits with arbitrary angles

Overlaps

Using Network Simulations to Understand Non-Darcy Flow

Implementation of Objectives

Use custom codes to extract FTPM networks from code

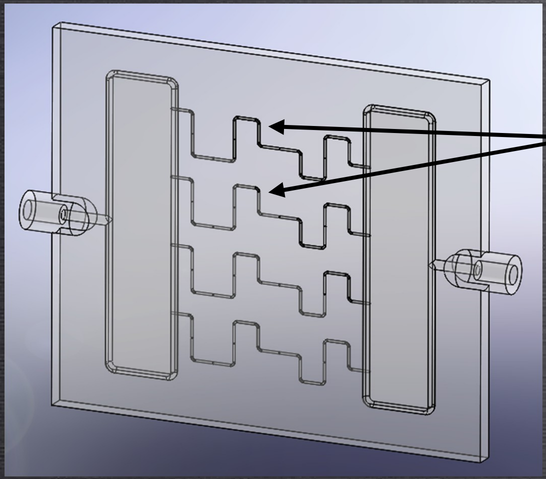


Using Network Simulations to Understand Non-Darcy Flow

Implementation of Objectives

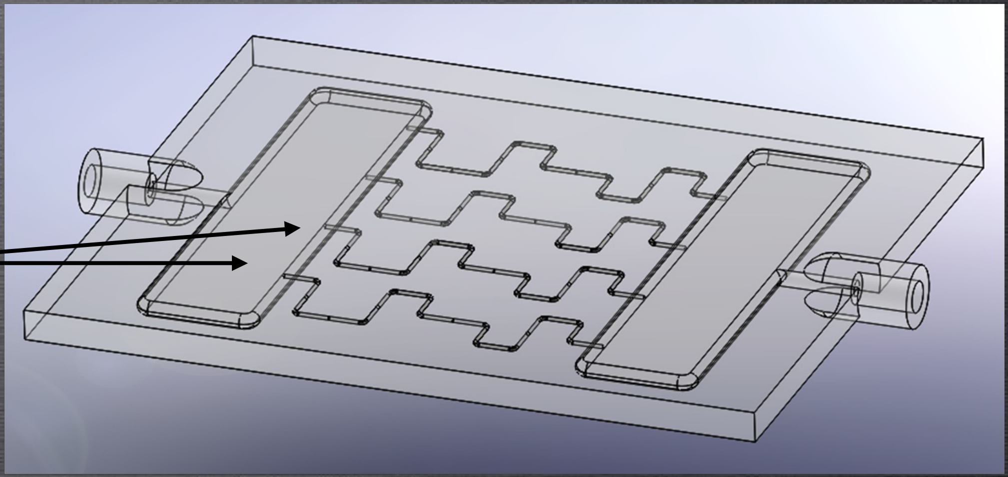
Use Solidworks to create network designs

Phase 1



90° elbows

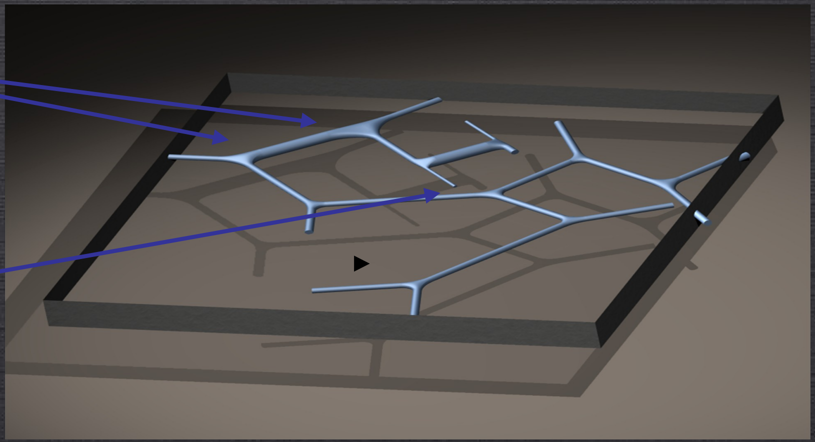
No overlapping



Phase 2

Arbitrary angles

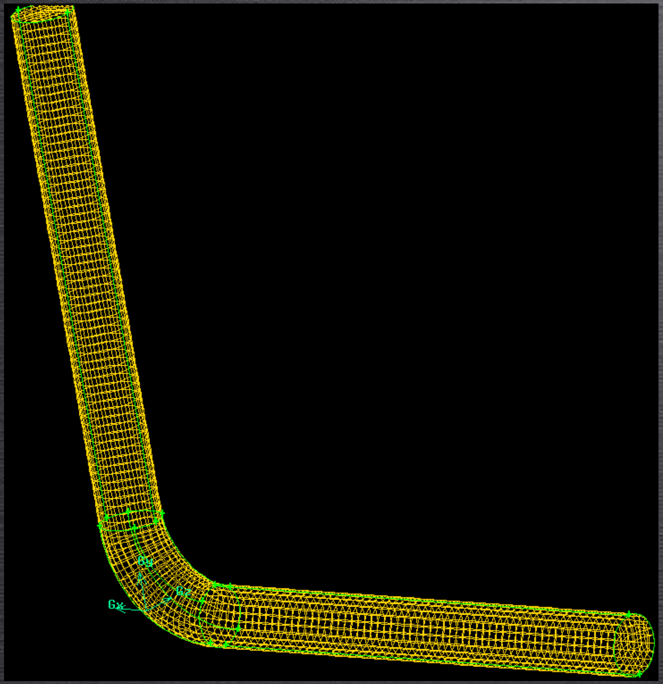
Possible overlaps



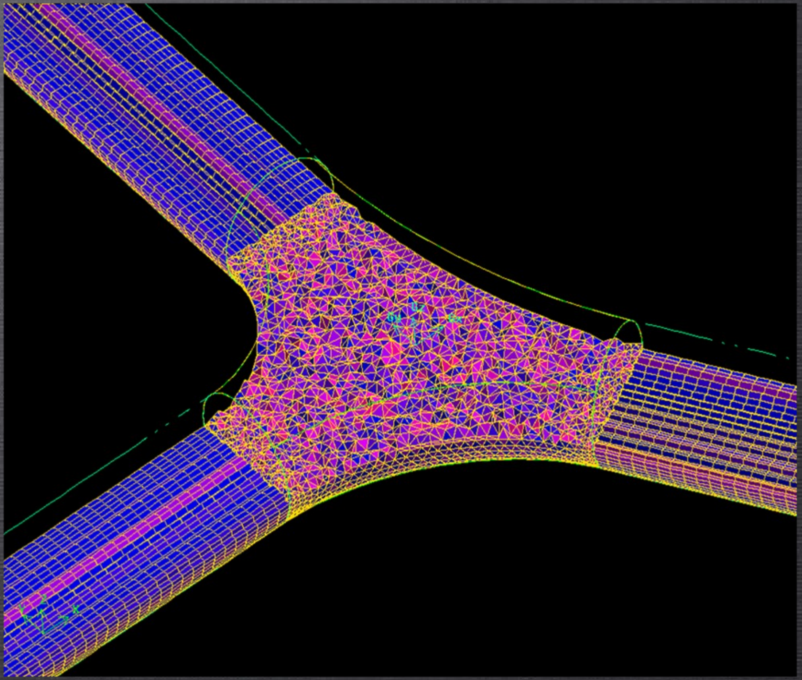
Using Network Simulations to Understand Non-Darcy Flow

Implementation of Objectives

Use Gambit to mesh networks



Close up of 90° elbow

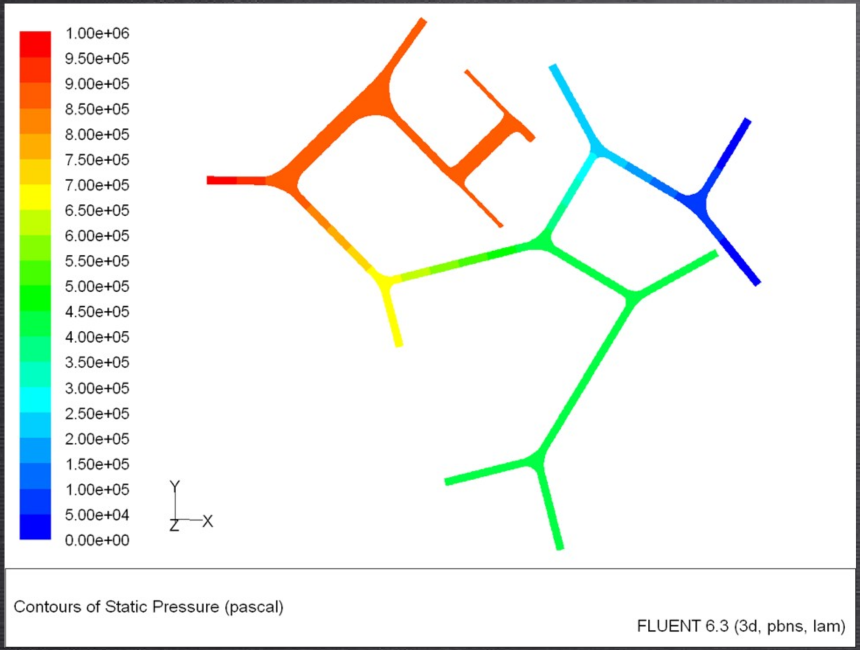


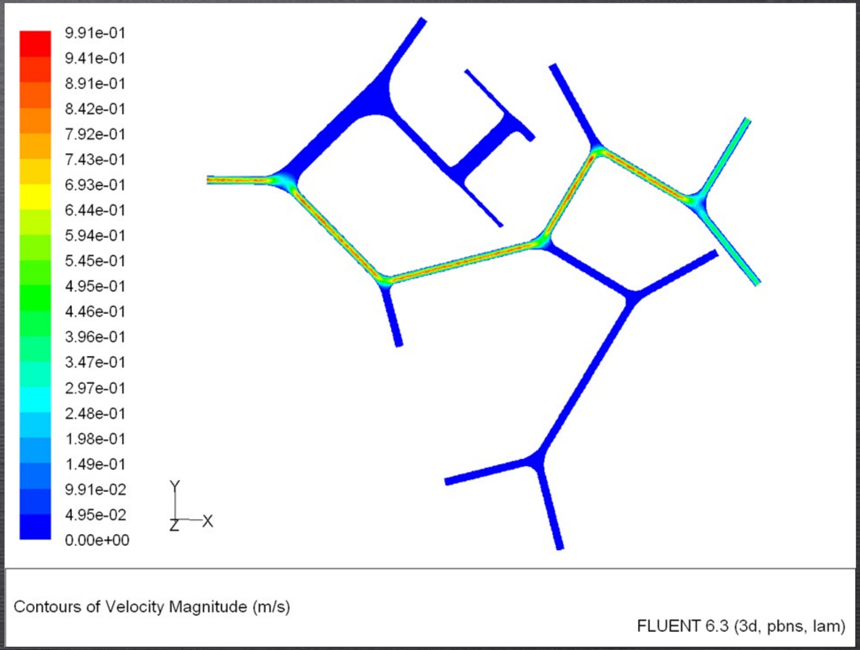
Close up of arbitrary junction

Using Network Simulations to Understand Non-Darcy Flow

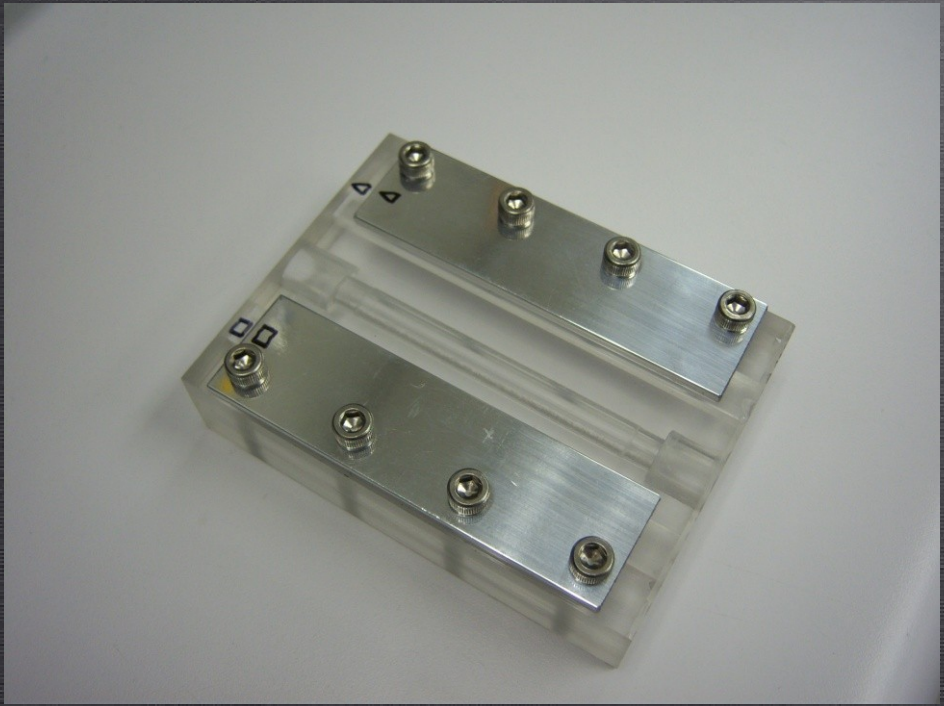
Implementation of Objectives

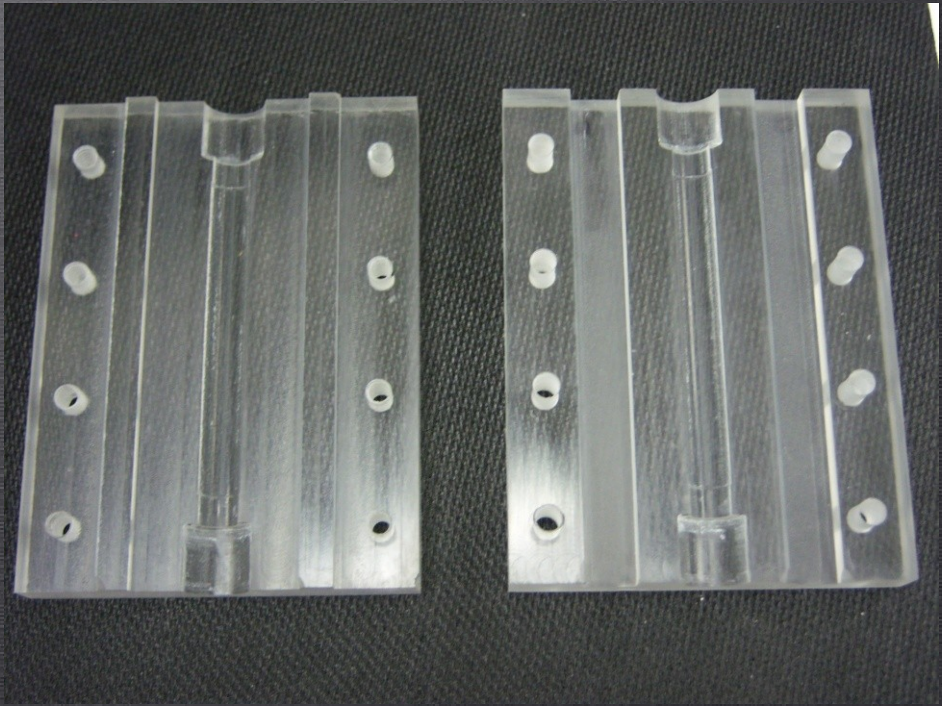
Use Fluent (CFD) to obtain κ and β from pressure and velocity data



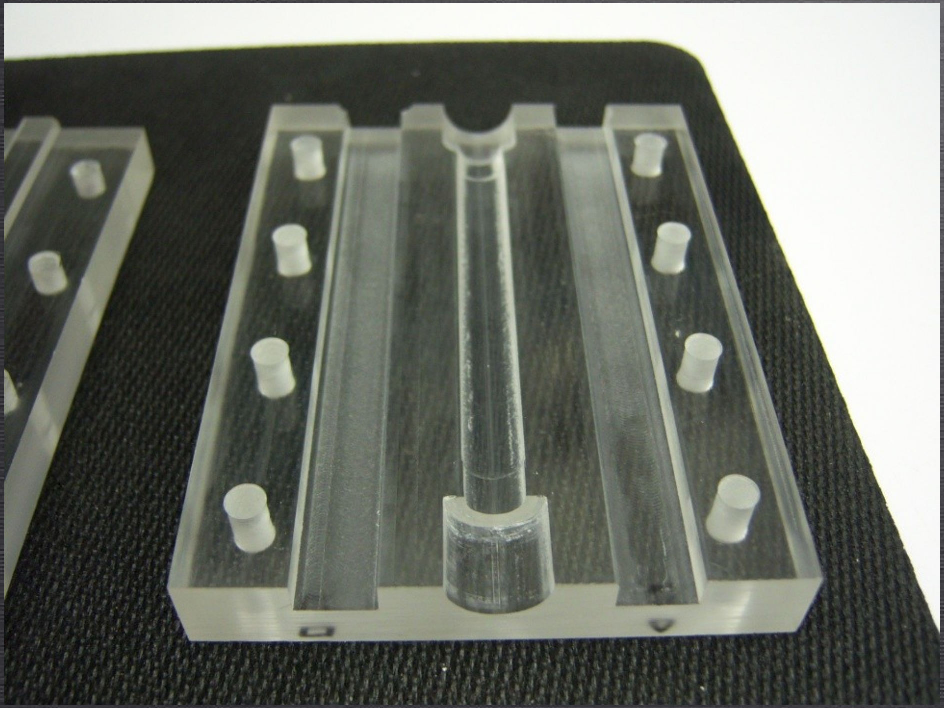


Bio-Scaffold Testing

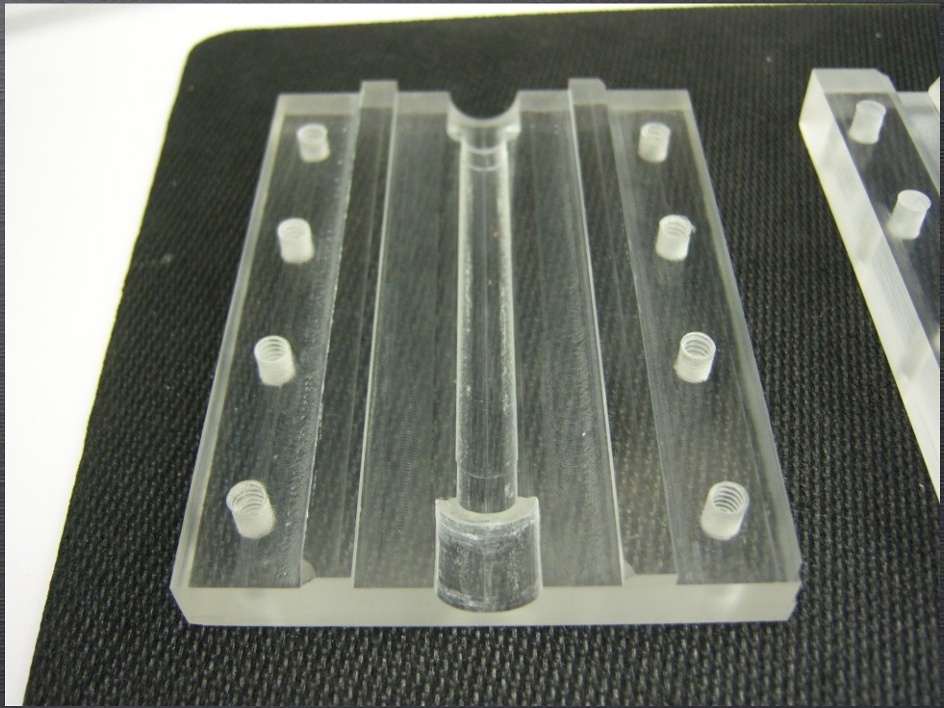




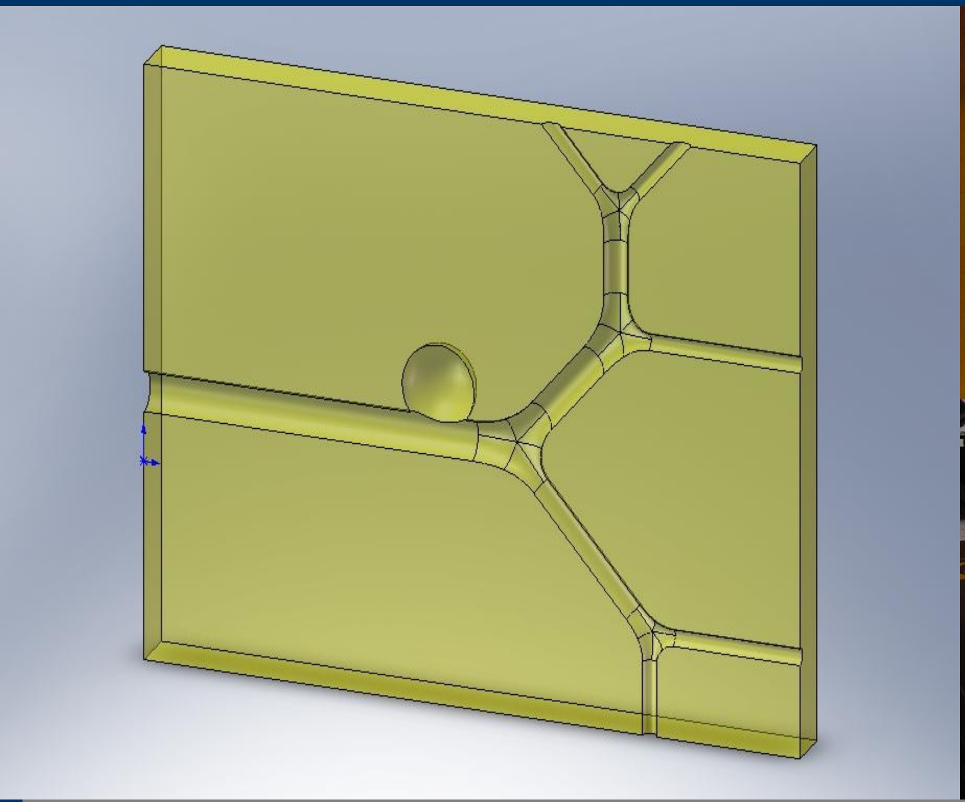
Top

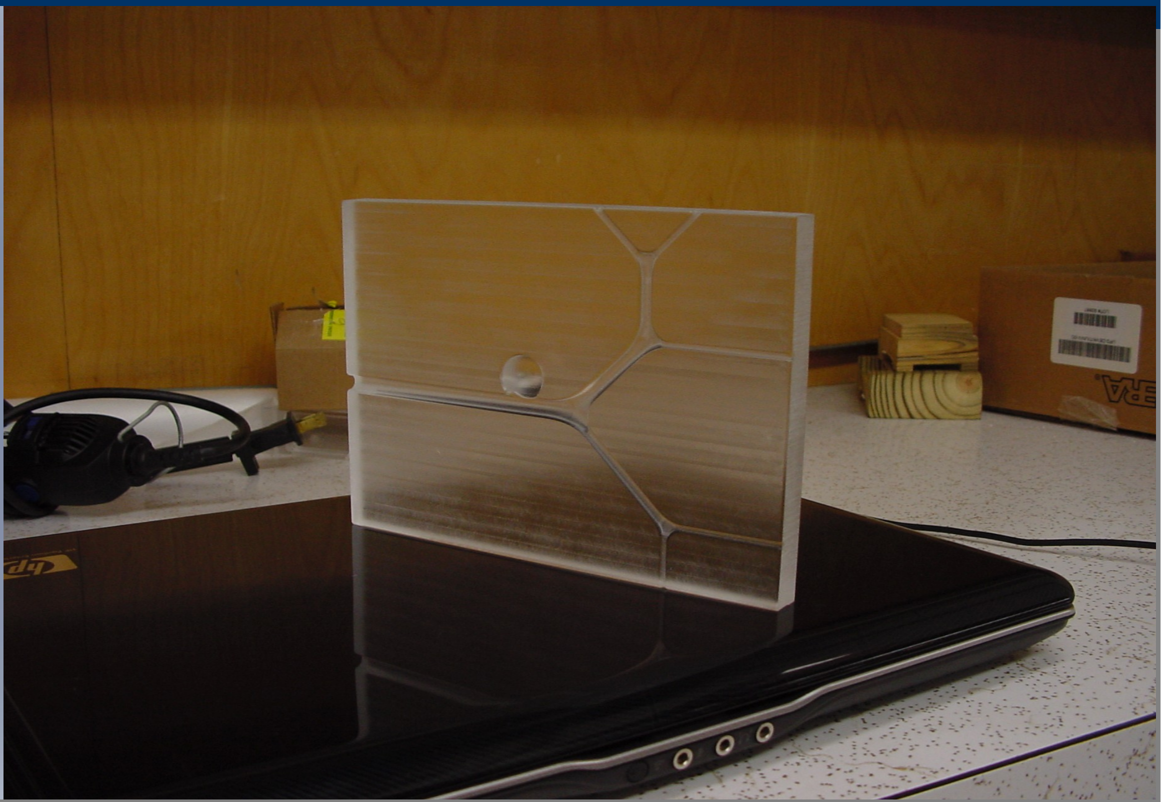


Bottom



Renal Artery Aneurysm Experiments





PDMS Microtubes

Acknowledgments

The Office of Research and Grants at the University of Central Oklahoma is acknowledged for support of this research.

The Donors of The Petroleum Research Fund, administered by the American Chemical Society, are also be acknowledged for support of this research through grant PRF# 47193-B9.

National Science Foundation EPSCoR Research Opportunity Award Program

Applications

Flow in microfluidic flow networks and flow in porous networks are of interest in many engineering applications.
Applications include porous media, micro-power generation, biomedical, computer chips, chemical separation processes, micro-valves, micro-pumps, and micro-flow sensors

Yanuka, M., Dullien, F.A.L., and D.E. Elrick, 1986, "Percolation Processes And Porous Media I. Geometrical And Topological Model Of Porous Media Using A Three-Dimensional Joint Pore Size Distribution," J.Colloid Interface Sci., **112**, pp. 24-41.

Lee, W.Y., Wong, M., and Zohar, Y., 2002, "Microchannels in Series Connected Via a Contraction/expansion Section", J. Fluid Mech., **459**, pp.187-206.

Judy, J., Maynes, D., and Webb, B.W., 2002, "Characterization of Frictional Pressure Drop for Liquid Flows Through Microchannels," Intl. J. Heat Mass Trans., **45**, pp.3477-3489.

Flow in these applications is usually laminar

Graveson, P., Branebjerg, J., and Jensen, O.S., 1993, "Microfluidics a Review," J. Micromech. Microeng., 3, pp.168-182.