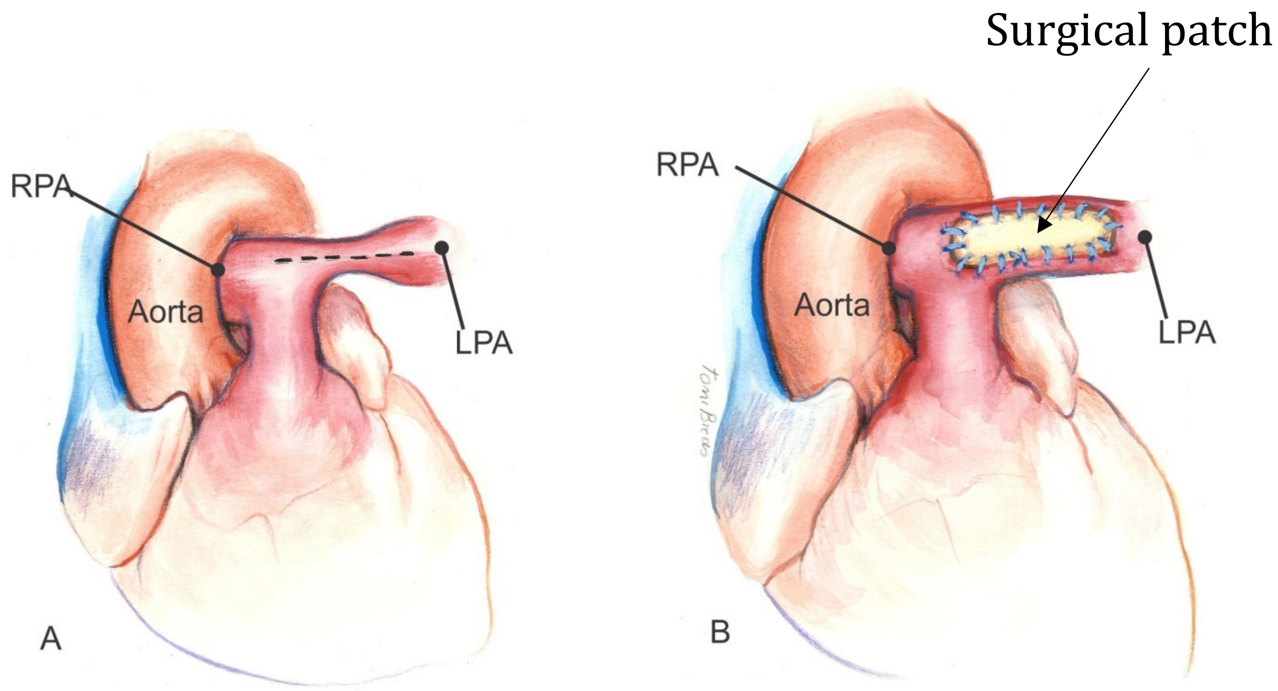
Probability-Based Surgical Guidance Tool for Balloon Angioplasty/Stenting

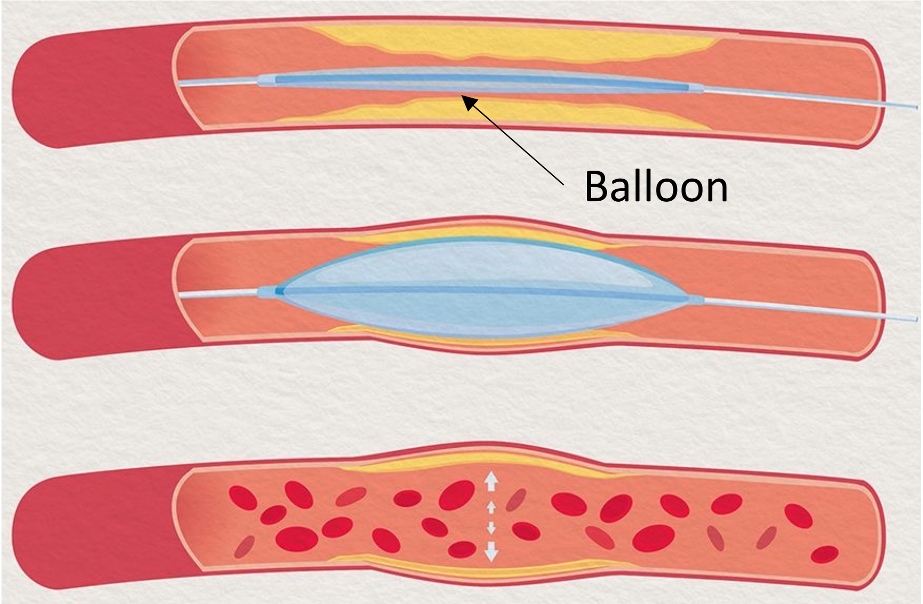
John Lee
In collaboration with: Jason Szafron, Karthik Menon, Daniele Schiavazzi

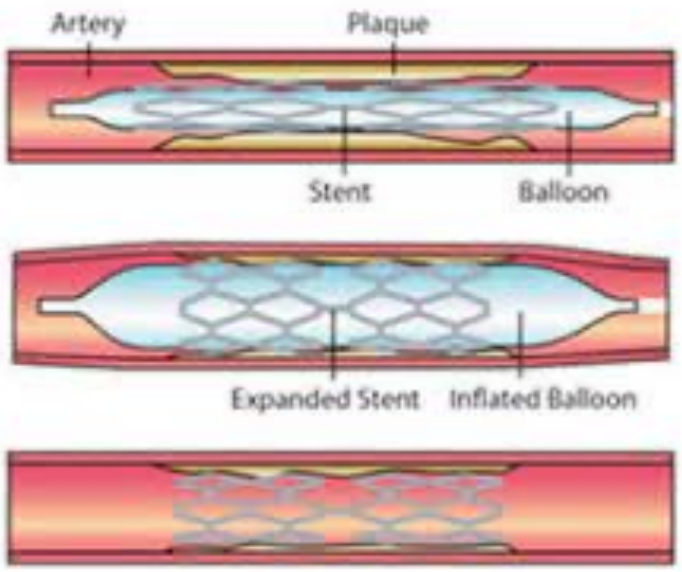
8/4/2022

Background



Optimal treatment: Surgical reconstruction

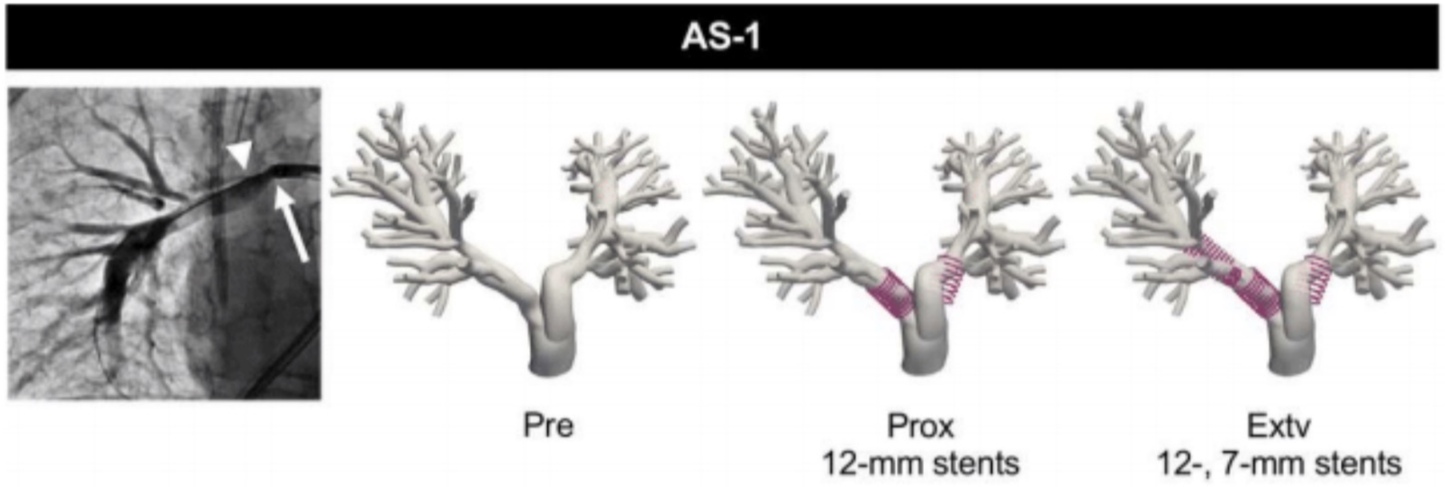




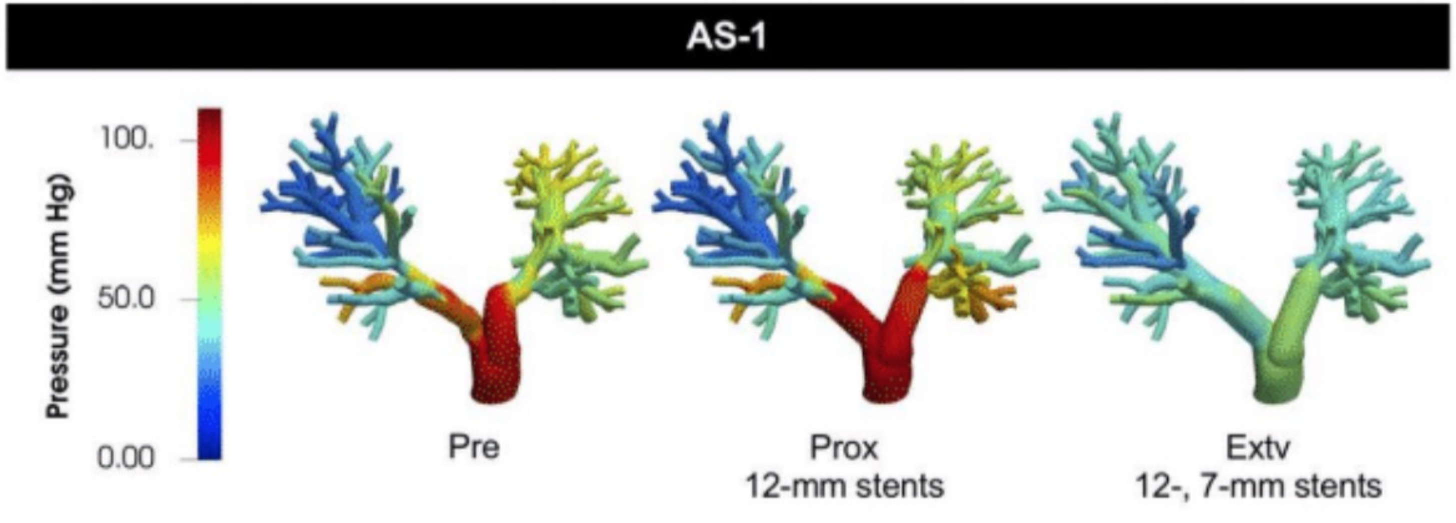
Transcatheter treatment: Balloon Angioplasty Process w/ and w/o Stent

Previous Work

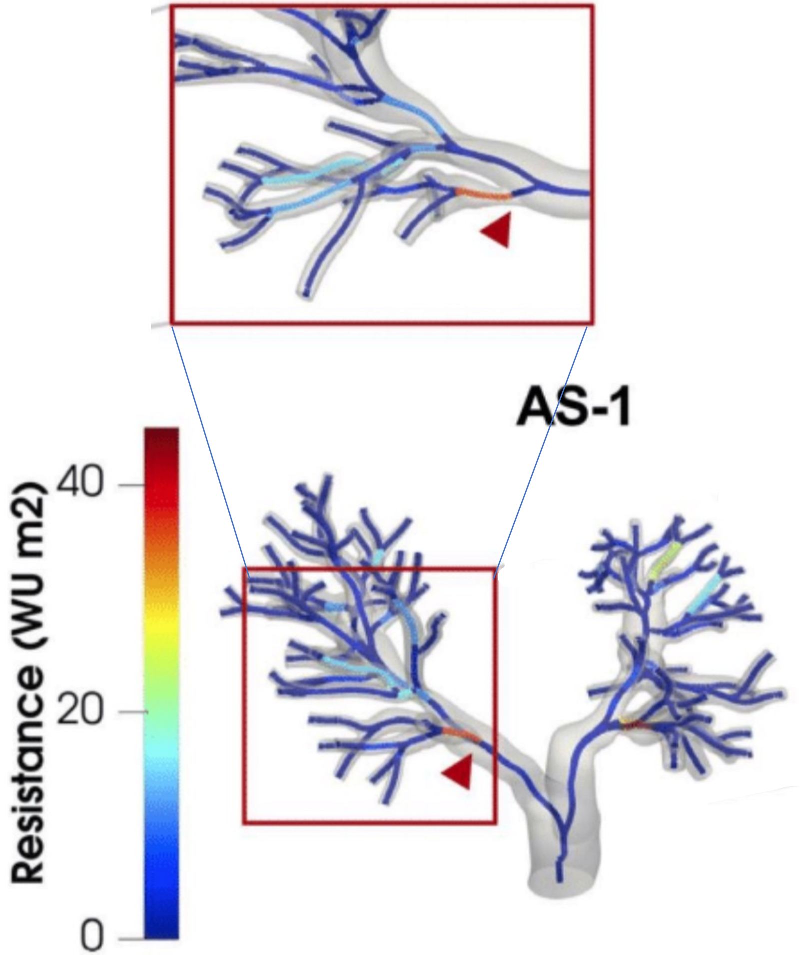
Lan et al., AHA Journal, 2022



Stenting Locations



3D CFD simulations at peak systole



Resistance Map

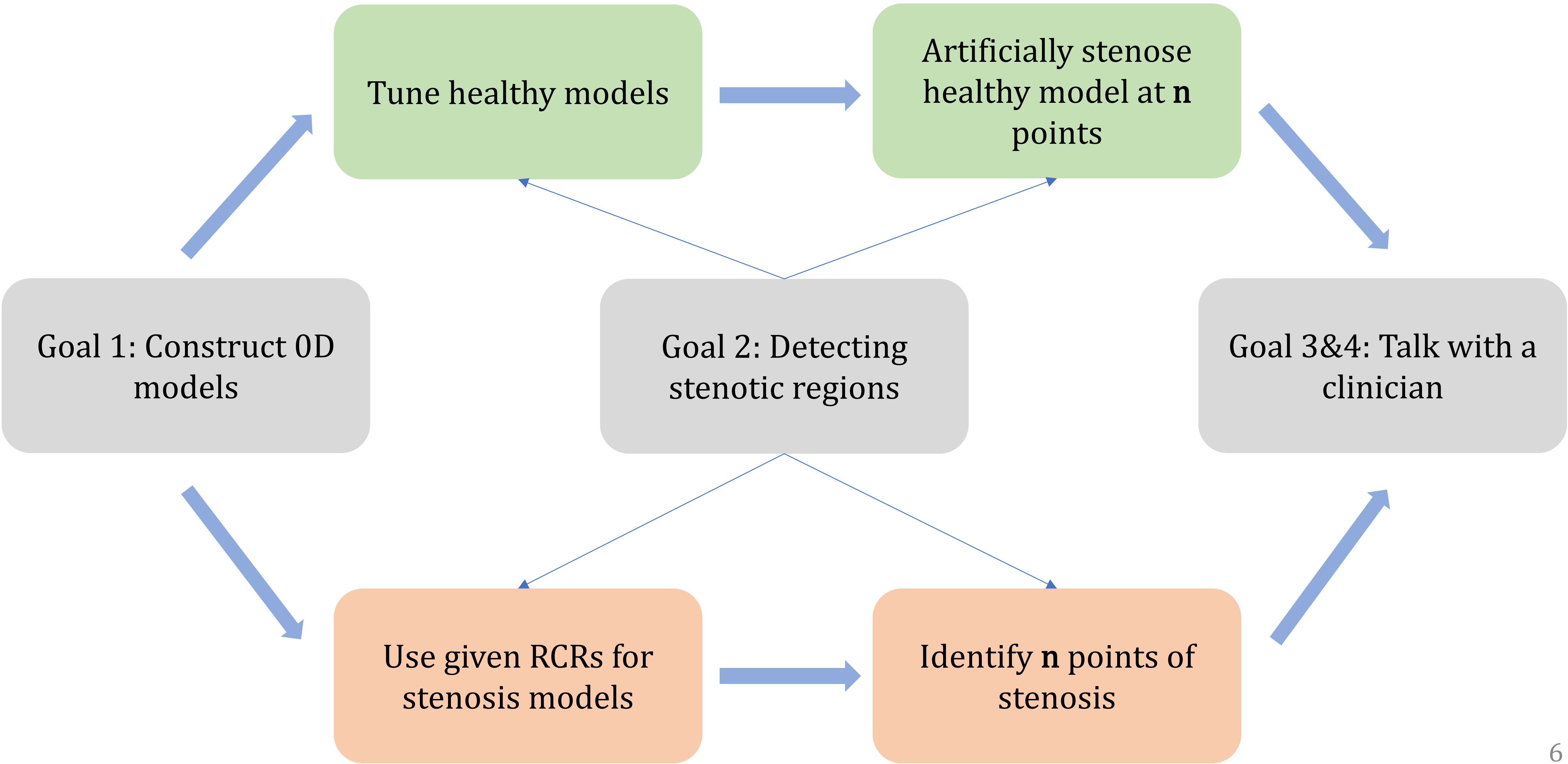
Motivation

- Extend previous studies to include a probabilistic representation of a successful stenosis repair
- Reduce computational expense of 3D simulations by utilizing 0D ROM models, enabling real-time virtual treatment
- Implement a first prototype for a probability-based virtual treatment guidance tool.
- Highlight the importance of minor pressure losses at bifurcations
- Improve the accuracy of 0D formulations to model pulmonary anatomies with multiple generations

Project Goals/Steps

1. Given a 3D pulmonary model, construct an accurate lumped parameter (0D) low-fidelity representation
2. Identify **n** stenosis locations on the 0D model for repair.
3. Define a probabilistic model for repair success.
4. Identify locations to measure pressures and flow to characterize a successful outcome
5. Train a feed-forward neural network using 0D simulations to capture the changes in pressure/flow corresponding to various degrees of successful repair
6. Compute marginal/conditional probabilities to answer relevant clinical questions

Overview

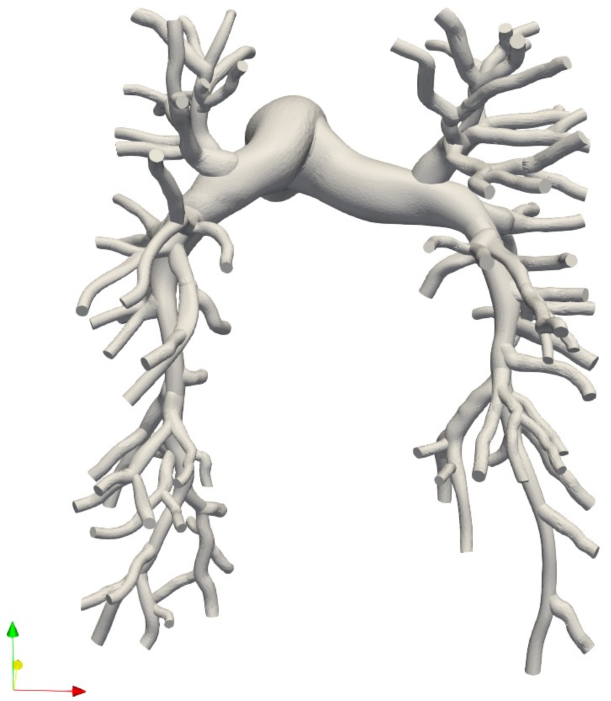

Tune healthy models
Artificially stenose healthy model at **n** points
Goal 1: Construct 0D models
Goal 2: Detecting stenotic regions
Goal 3&4: Talk with a clinician
Use given RCRs for stenosis models
Identify **n** points of stenosis

Goal 1

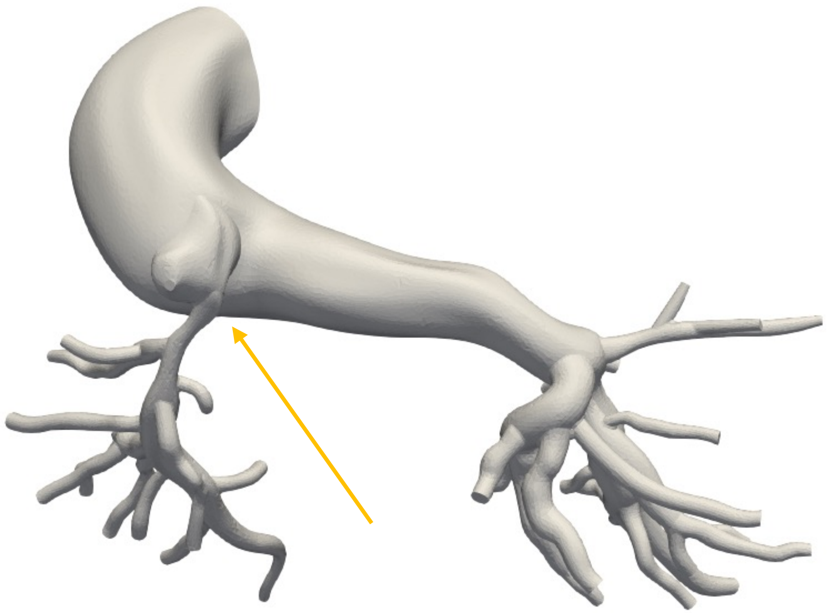
Construct ROM from 3D model

Model Gathering

0080_0001 (Healthy)



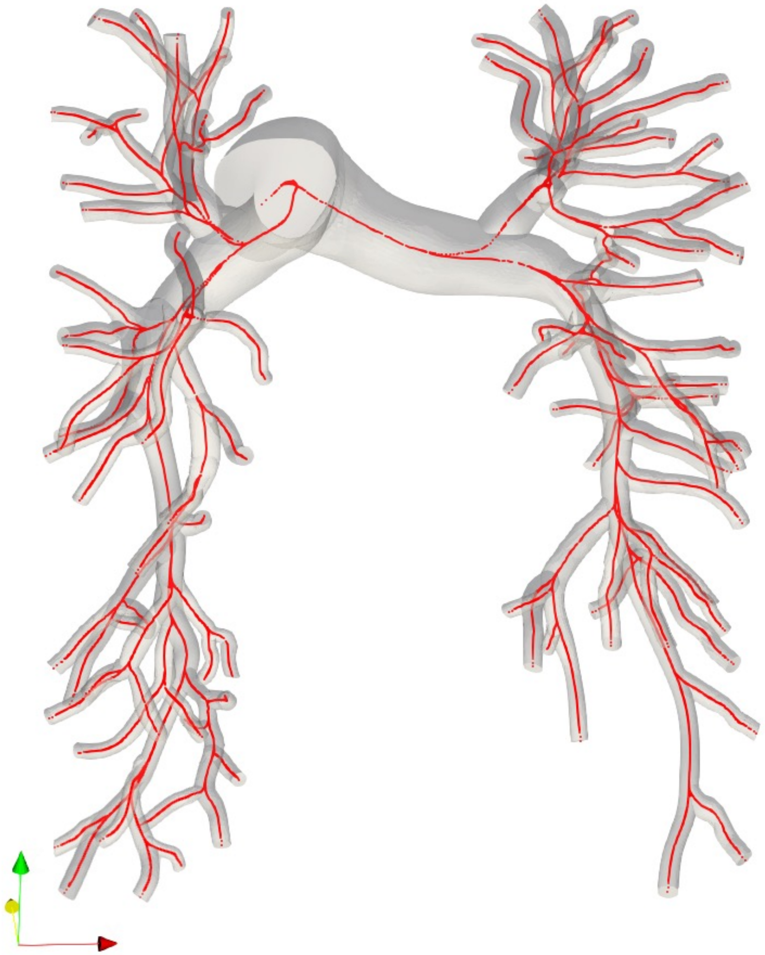
0118_1000 (Alagille)



- Gather both healthy and stenosis
  - 9 Healthy (VMR)
  - 2 Alagille (VMR)
  - 3 A&W (Lan et al.)
- Working models
  - 3 Healthy
  - 2 Stenosis
- Problems with VMR models
  - Not finely meshed enough to generate centerlines
  - Insufficiently blended (unable to load faces for blending)

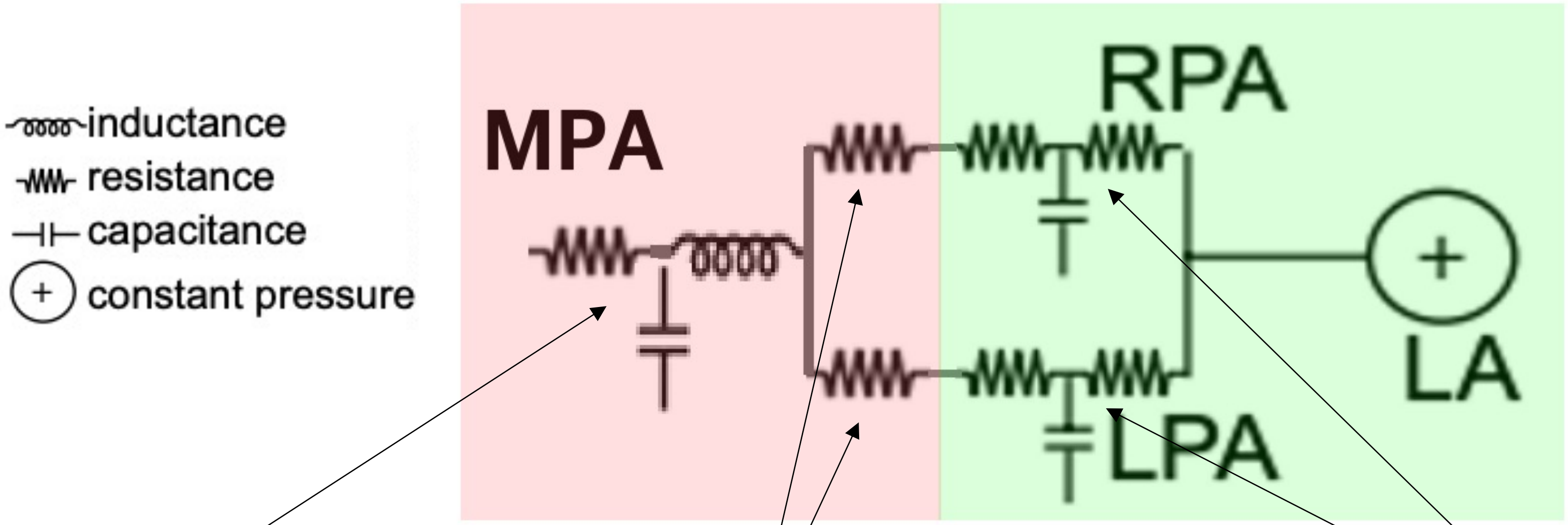
Centerline/0D Model Generation

- Generate Centerlines using VMTK
- Automatically generate 0D model using Simvascular 0D pipeline
  - Linear wall
  - $EH/r = 1.26 * 10^6$ dyne/cm^2

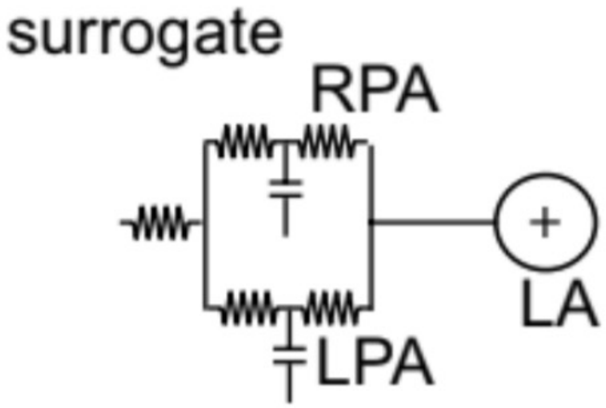


Healthy Models BC tuning

Yang et. al. 2019 Surrogate model (Modified)



Original



Solver-determined RCL values.

RPA/LPA resistances computed using resistances in series and parallel

Tunable RCR boundary conditions for LPA and RPA, with constant capillary wedge pressure of 7 mmHg

Optimization Procedure

Procedure

1. Receive a segmented 0D model w/ dummy RCRs
2. Compute RPA/LPA R values only using Poiseuille resistance values
3. Optimize until goal is reached
4. Split RCRs, run model once
5. Recompute RPA/LPA R values, including stenosis coefficients using approximate flow from run
6. Optimize until goal is reached

Goal

- Ideal: ObjFunc $< .01$ (1% error)
- Early stop: Patience (5), Patience Tolerance (1e-6)

Split RCRS (Yang et. al. 2019)

$$R_{p_i} = \frac{A}{A_i} R_p \quad C_i = \frac{A_i}{A} C, \quad R_{d_i} = \frac{A}{A_i} R_d,$$

Objective Function

Error Terms

- Average Pulmonary Arterial Pressure (in mmHg)
  - $PiecewiseLoss(\ 12, 16, \overline{PAP})$
- Max (systolic) Pulmonary Arterial Pressure (in mmHg)
  - $PiecewiseLoss(18, 25, \max PAP)$
- Flow Split
  - $SquaredErrorLoss(0.55 * \overline{Q_{inflow}}, \overline{Q_{RPA}})$
- MSE of flow over time
  - Cubic spline of $Q_{RPA} \rightarrow \widehat{Q_{RPA}}$
  - For n timesteps in Q_{inflow},
  - $.01 * \frac{1}{n}\sum_{t=0}^{n} SquaredErrorLoss(.55 * {Q_{inflow}}_t, \widehat{Q_{RPA_t}})$

Formulas

$*PiecewiseLoss(lower, upper, sim)$

If $sim > upper$: $\left(\frac{sim - upper}{upper}\right)^2$

If $sim < lower$: $\left(\frac{sim - lower}{lower}\right)^2$

Else: 0

$*SquaredErrorLoss(target, sim)$

$$\left(\frac{(sim - target)}{target}\right)^2$$

Stenosis Model BC

- Only use models where an rcrt.dat file is provided for 3D simulations
- Must be changed from 3D ordering to 0D mapping organization

Create mapping using .svpre file

Takes in 3D RCR

```
3367.7036414
4.474925571e-06
24892.9062086
0.0 10664.0
1.0 10664.0
```

```
$mesh_and_adjncy_vtu mesh-complete/mesh-complete.mesh.vtu
set_surface_id_vtp mesh-complete/mesh-complete.exterior.vtp 1
set_surface_id_vtp mesh-complete/mesh-surfaces/inflow.vtp 2
set_surface_id_vtp mesh-complete/mesh-surfaces/RPA_001.vtp 3
set_surface_id_vtp mesh-complete/mesh-surfaces/RPA_002.vtp 4
set_surface_id_vtp mesh-complete/mesh-surfaces/RPA_003.vtp 5
set_surface_id_vtp mesh-complete/mesh-surfaces/RPA_004.vtp 6
```

```
Number of RCR Surfaces: 34
List of RCR Surfaces: 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36
RCR Values From File: True
```

Retrieve order using solver.inp file

Maps to 0D

```
RPA_001
3367.7036414
4.474925571e-06
24892.9062086
0.0 10664.0
1.0 10664.0
```

Goal 2

Identify Stenotic Regions

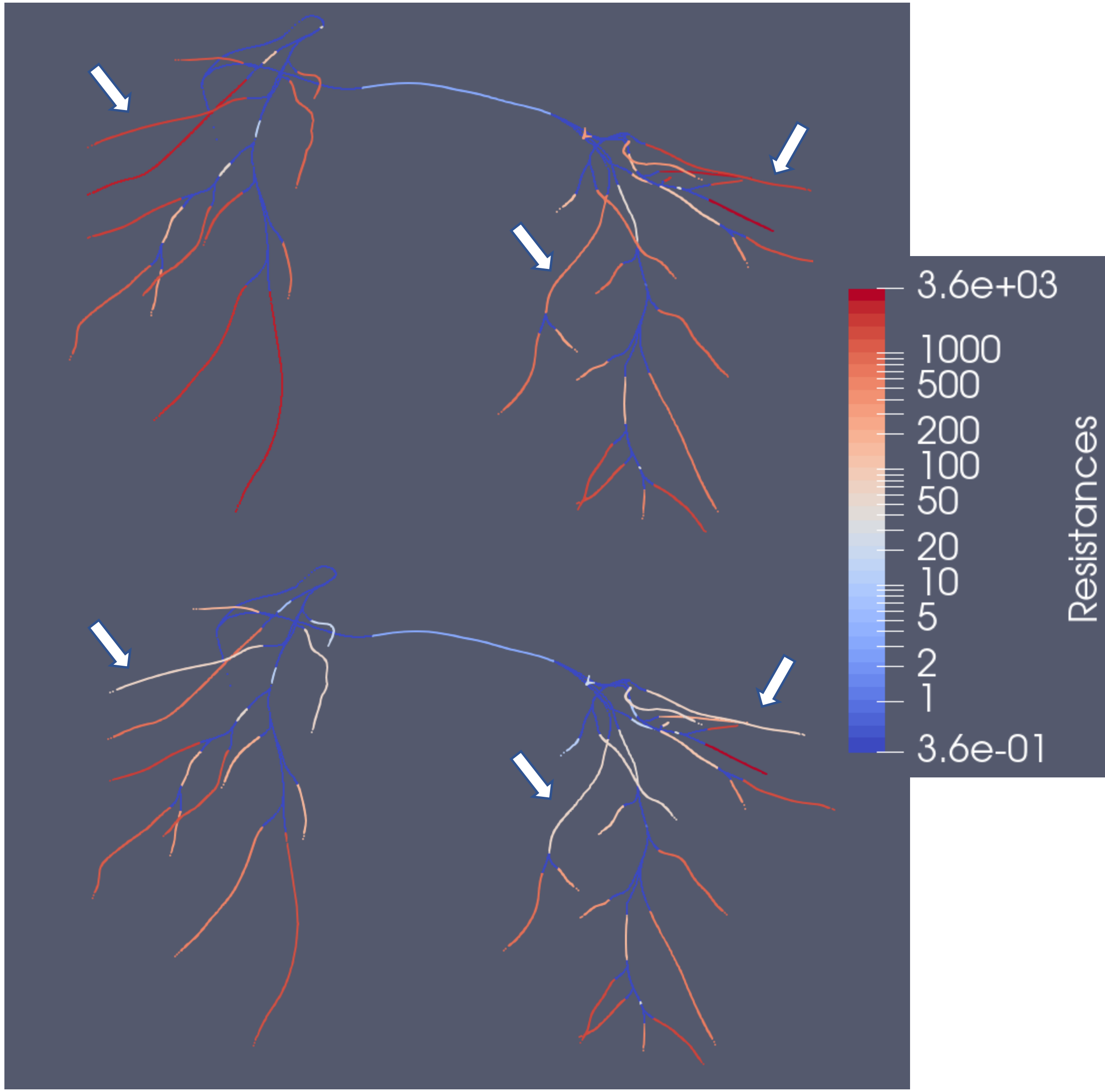
Identifying Stenosis

$$D = 0.001 + a * order * e^{(b * order)} * age^{c}$$

- Formula to compute theoretical diameters for a healthy ”control” vessel
    - a = 1.203e-4, b = 0.3927, c = 0.2381
    - Dong et. al. 2020
- Construct a generational branch tree w/ gen0 = MPA, gen1 = LPA & RPA …
- For each branch,
    - Compute control resistance from diameter formula above
    - If (true resistance > **r_threshold(4)** * control resistance), consider it a stenosis point.
    - If not, iterate through individual vessel segments in the branch and repeat comparison

$$\text{Occlusion} = 1 - \frac{1}{\sqrt{r_{threshold}}},$$

Pre-Treatment Resistance Map



Post-Treatment Resistance Map

Artificial Stenosis – Healthy Models

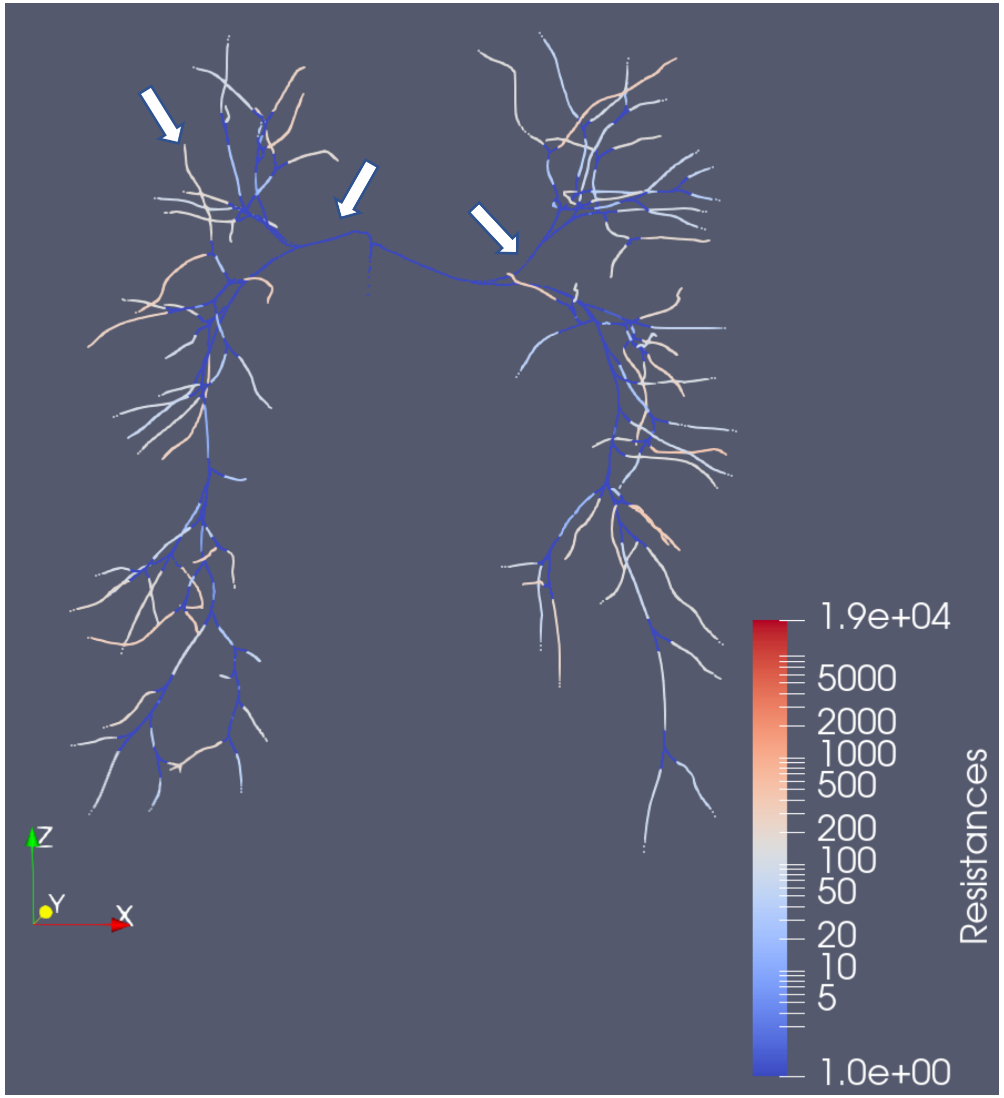

```json
{
    "generation_info": {
        "0": {
            "average_occlusion": 0.0,
            "std_occlusion": 0.0,
            "stenosis_length_ratio": 0.0
        },
        "1": {
            "average_occlusion": 0.9152886615713076,
            "std_occlusion": 0.0,
            "stenosis_length_ratio": 0.02423706432796031
        },
        "2": {
            "average_occlusion": 0.7533905605176203,
            "std_occlusion": 0.06028913436433614,
            "stenosis_length_ratio": 0.9268625986923515
        },
        "3": {
            "average_occlusion": 0.6406152649354111,
            "std_occlusion": 0.07655316024914335,
            "stenosis_length_ratio": 0.8047476356586705
        },
        "4": {
            "average_occlusion": 0.0,
            "std_occlusion": 0.0,
            "stenosis_length_ratio": 0.0
        },
        "5": {
            "average_occlusion": 0.0,
            "std_occlusion": 0.0,
            "stenosis_length_ratio": 0.0
        },
        "6": {
            "average_occlusion": 0.0,
            "std_occlusion": 0.0,
            "stenosis_length_ratio": 0.0
        }
    },
```


Config File for SU0238

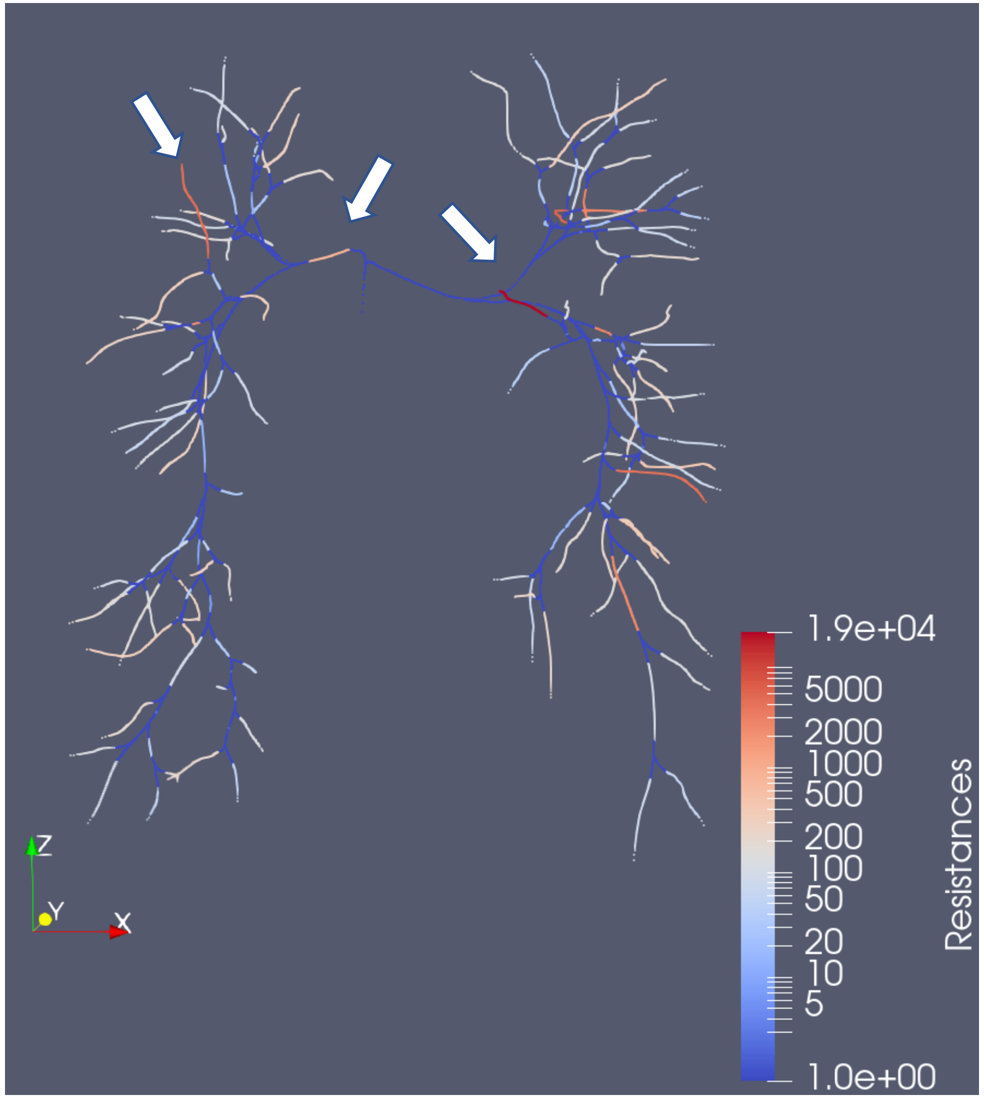
1. Identify stenosis points for all generations in an existing stenosis model (ex. SU0238)
2. Compute an average/std of occlusions for each generation
3. For each generation in the healthy base model, select vessels to stenose until the stenosis to generational length ratio matches the stenosis model
4. For each vessel, Occlusion (O) = $Normal\left(\mu = O_{gen_{avg}}, \sigma = O_{gen_{std}}\right)$ at all points
5. Make changes to R, C, L corresponding to occlusion changes in the branch

Artificial Stenosis Effects

Resistance Map: Before



Resistance Map: After



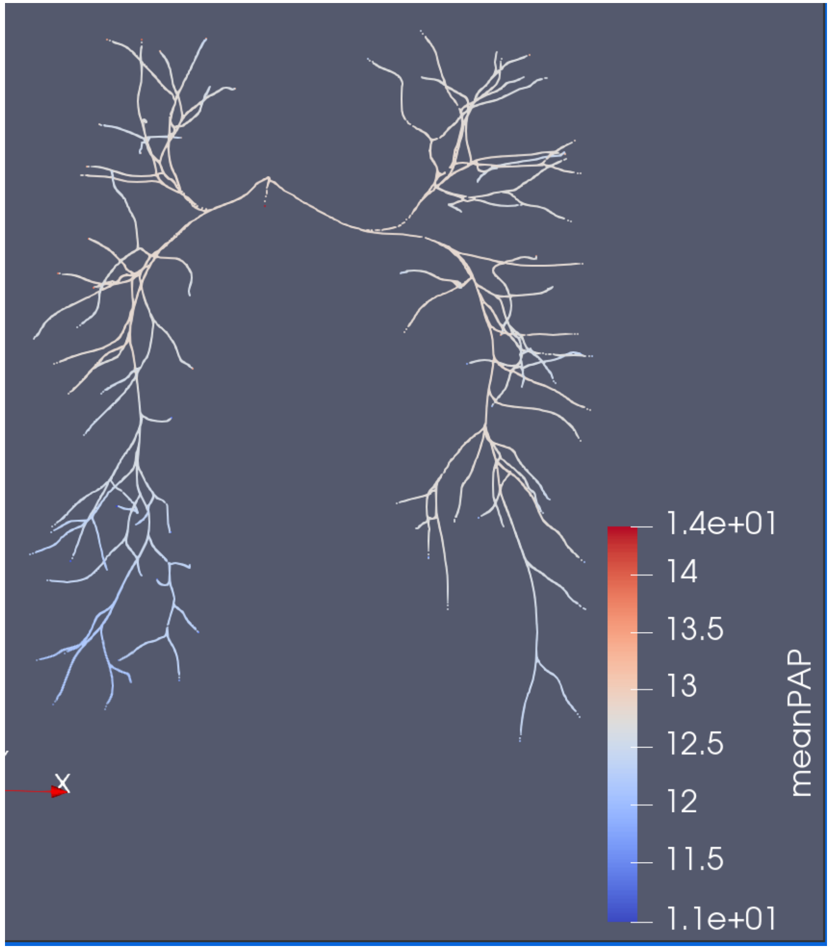
Code

- On Github at: https://github.com/JohnDLee/StenosisTool.git
- Let me know if you need more detail

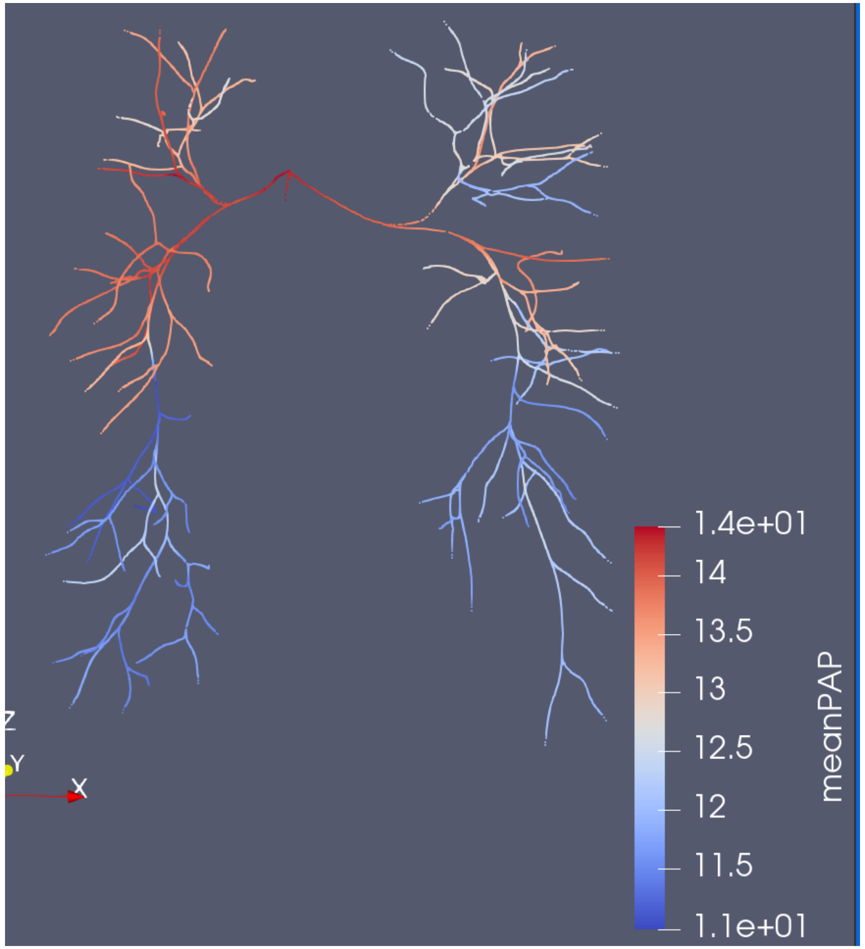
Junction Modeling

0D Junction Handling

0D

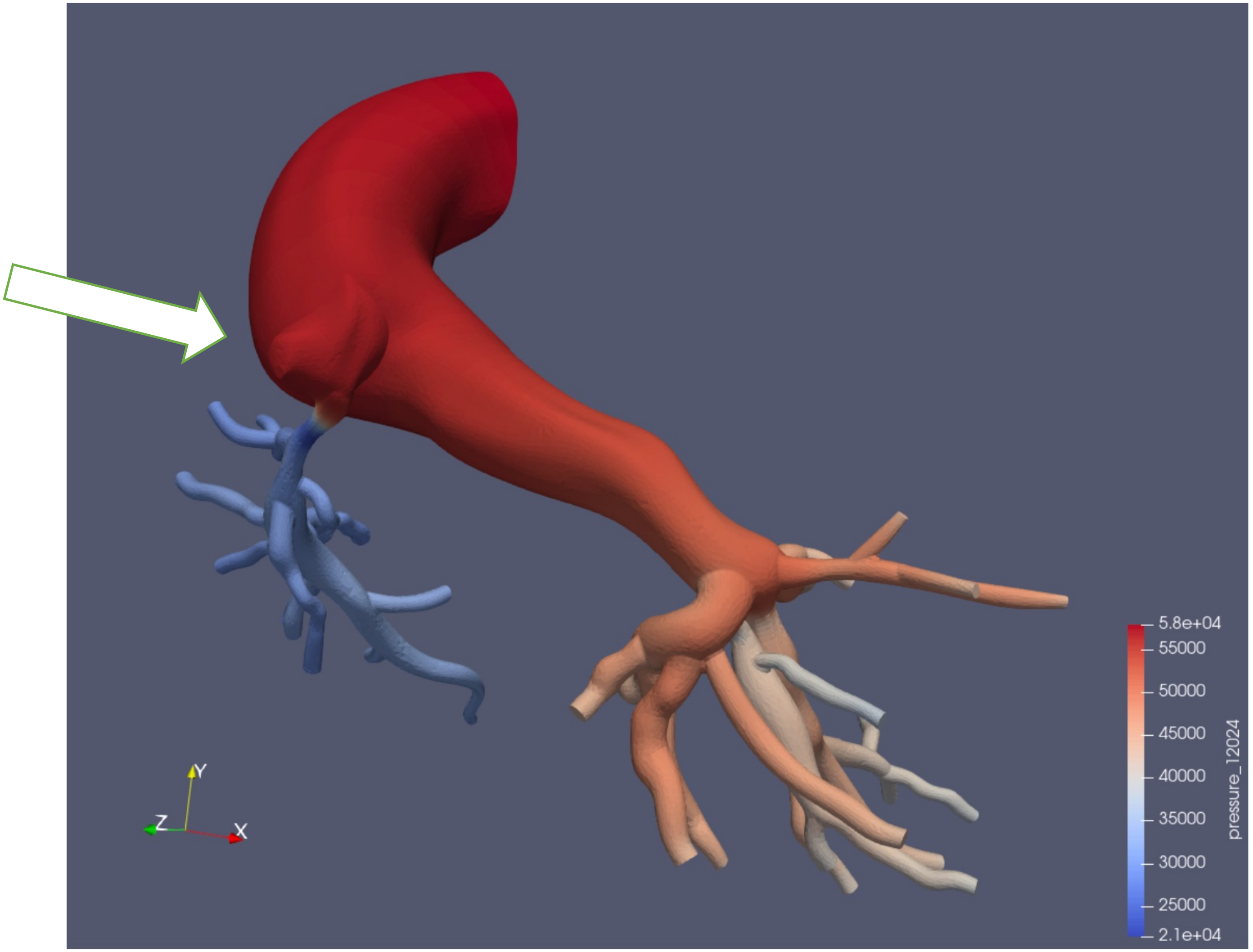


3D

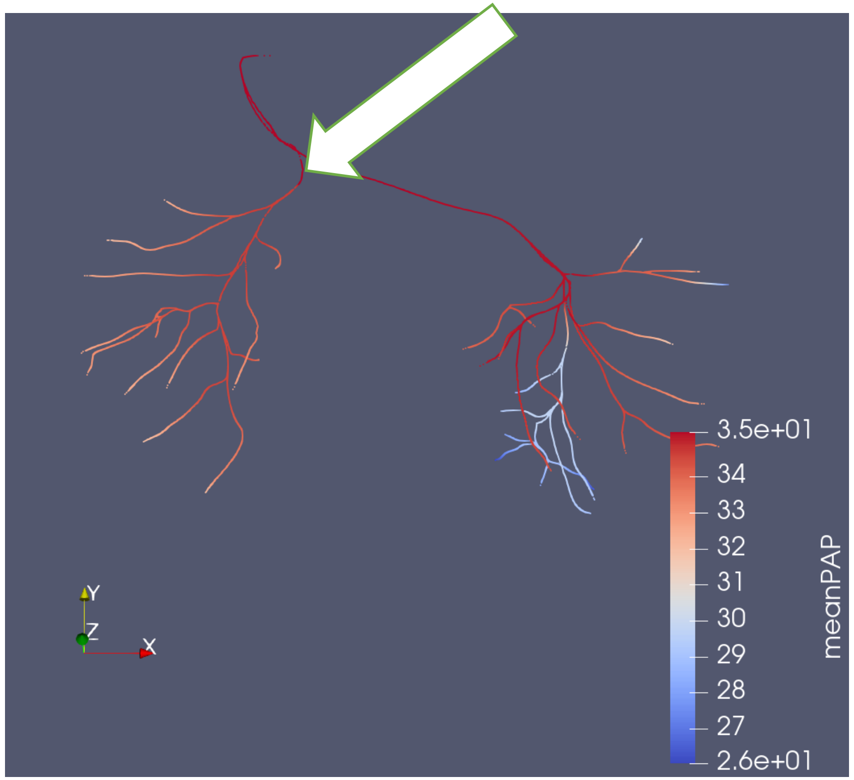


- 0D simulations fail to capture the pressure drop
- Previously noted that the significant number of junctions in pulmonary models induced a higher error in 0D simulations than other anatomies

Junction Stenosis Problem

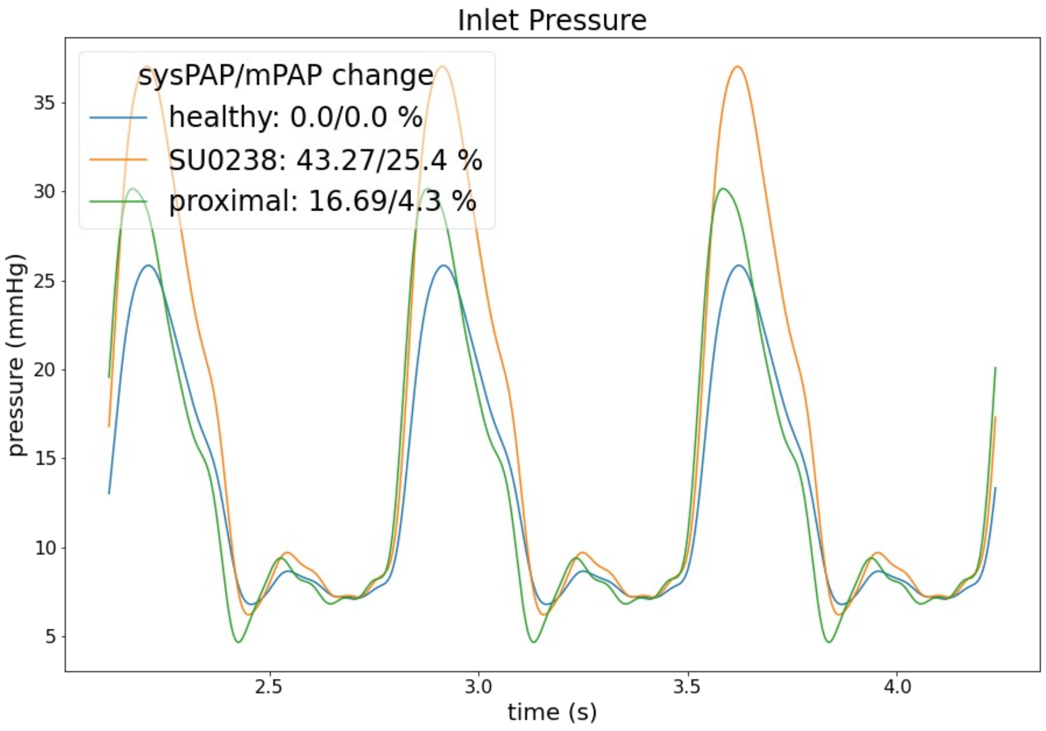


- 0D Junctions only conserve static pressure and flow
- Unable to capture any pressure drop occurring within junction



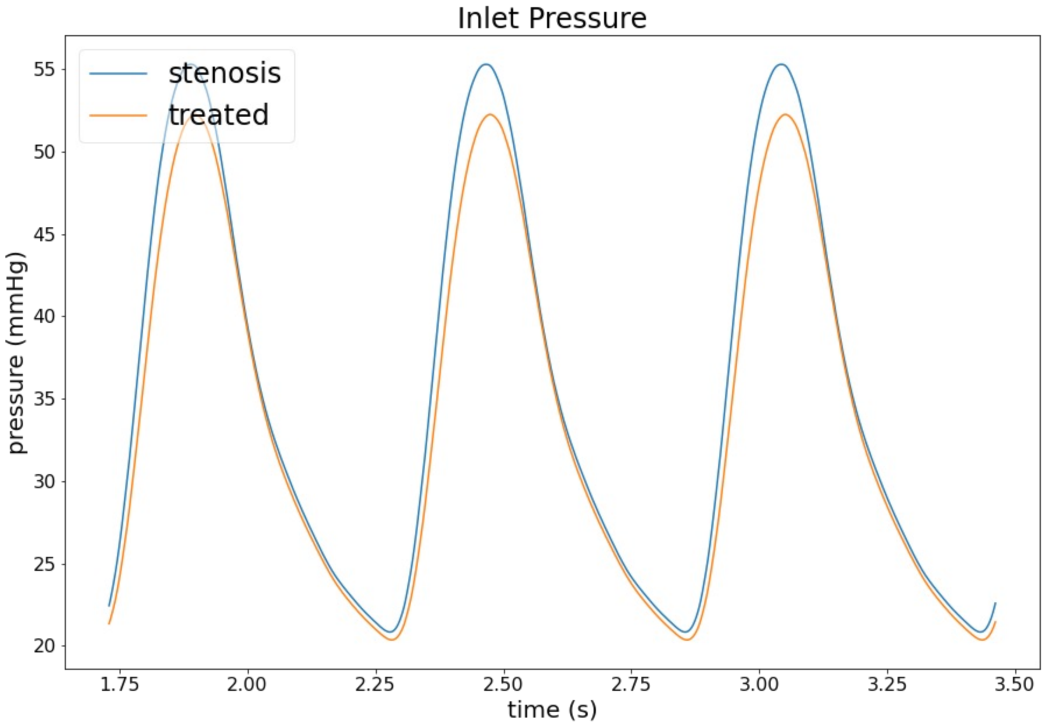
Unresponsive Treatment/Artificial Stenosis

Artificial Stenosis (0080_0001)



- Minimal effect of 75% occlusion (MPA/LPA/RPA) proximal stenosis
- Hardly effective even after stenosing 50% of all vessels by > 50% occlusion

Treatment (0118_1000)



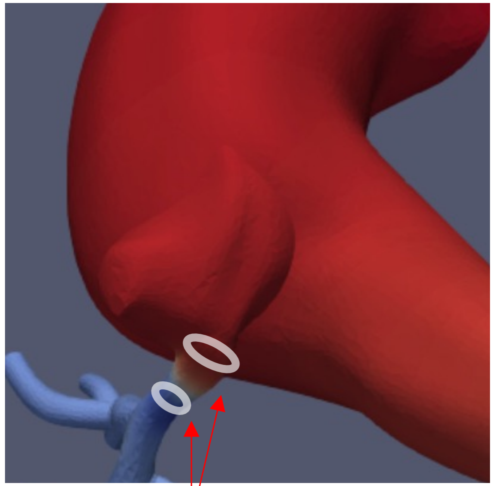
- Negligible change despite reducing resistances of nearly 50% vessels by more than 4x

$$Sten\_coeff = K_t \frac{\rho}{2S_0^2}\left(\frac{S_0}{S_s} - 1\right)^2$$

$K_t = 1.52,\ S_0 = A_i, S_s = A_o$

Potential Solutions

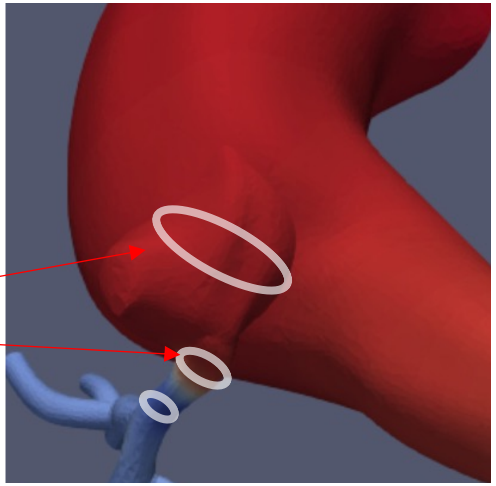
1. Junction Coefficients (Implemented)
   - Use vessel stenosis coefficient formula
   - Take inlet area of junction, and one outlet area, adding junction coeff to downstream vessel
2. More extensive/ML Junction solution
   - Currently Opened Issue on GitHub
3. 3D-optimization
   - Tune 0D RCL's to match pressure drop in a 3D model
   - Project-specific solution



← Original

Stenosis coefficient is small

Junction Coeff →

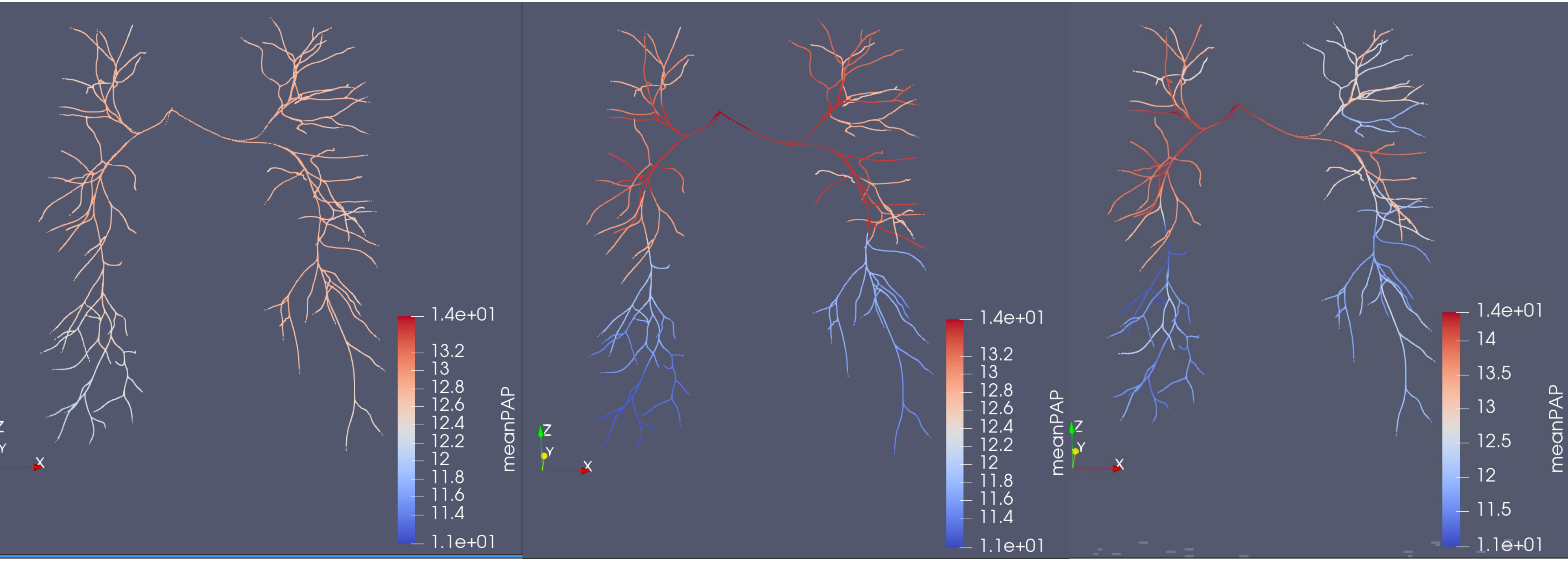


Additional Junction coefficient is much larger

Preliminary Results

0080_0001 (meanPAP)

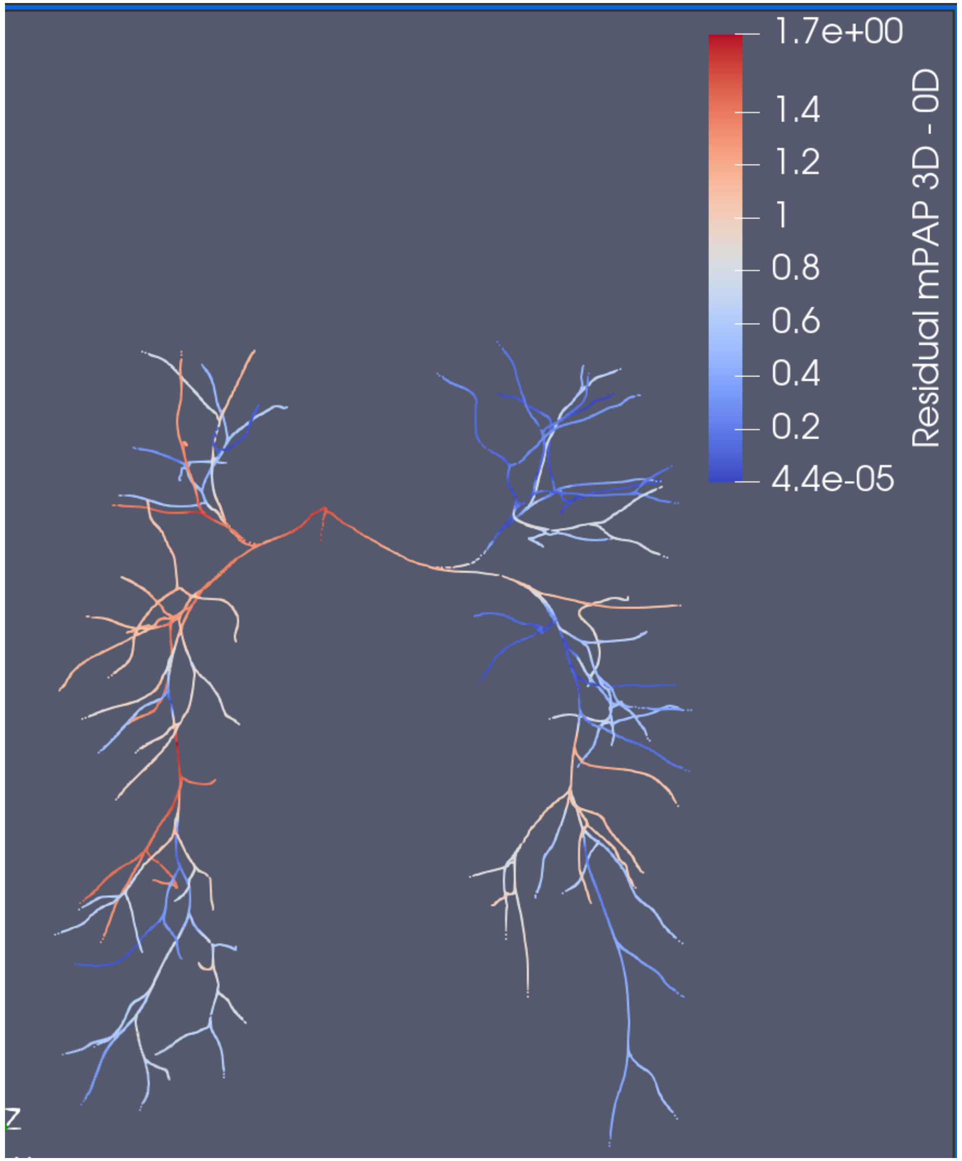
Original Solver | Junction Coefficient Solver | 3D



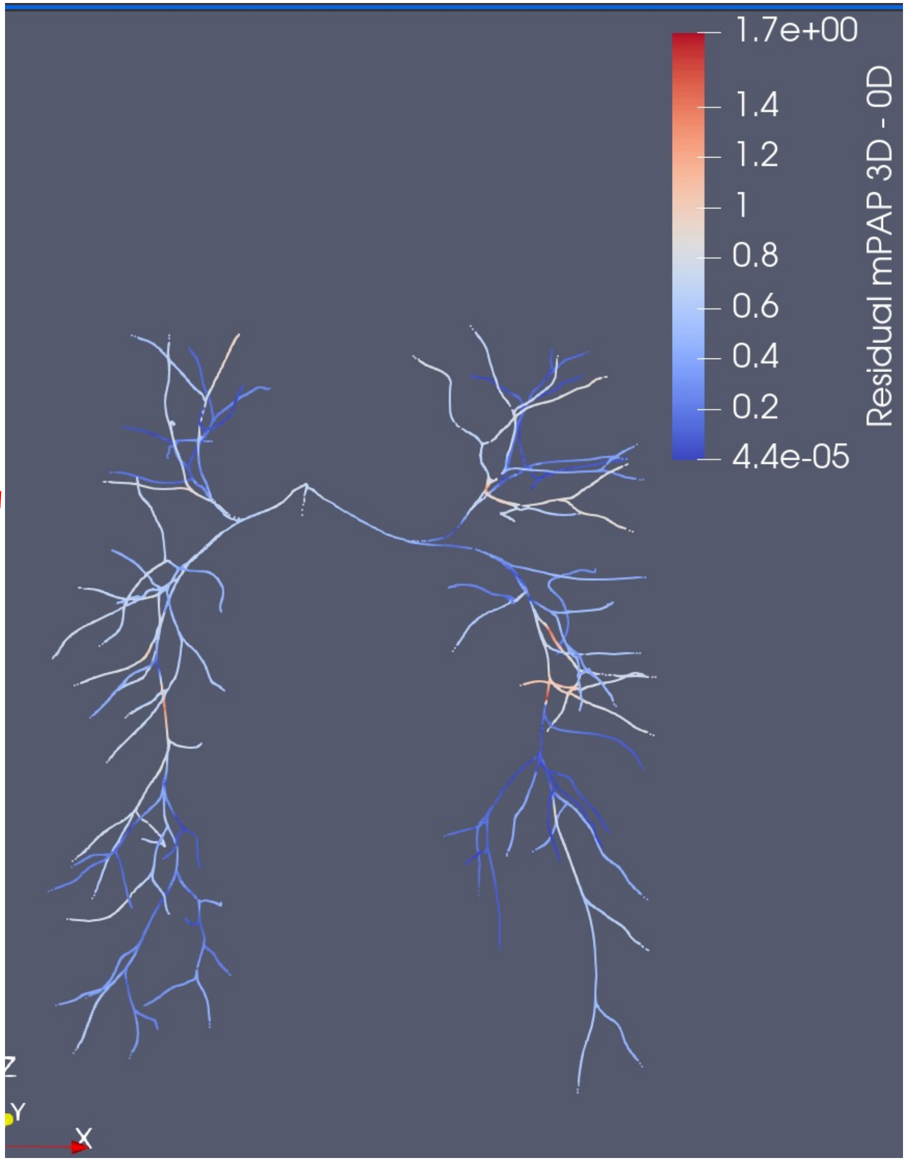
- Better Capture of the ~3 mmHg Pressure Drop

0080_0001 (Residual meanPAP)

Original Solver



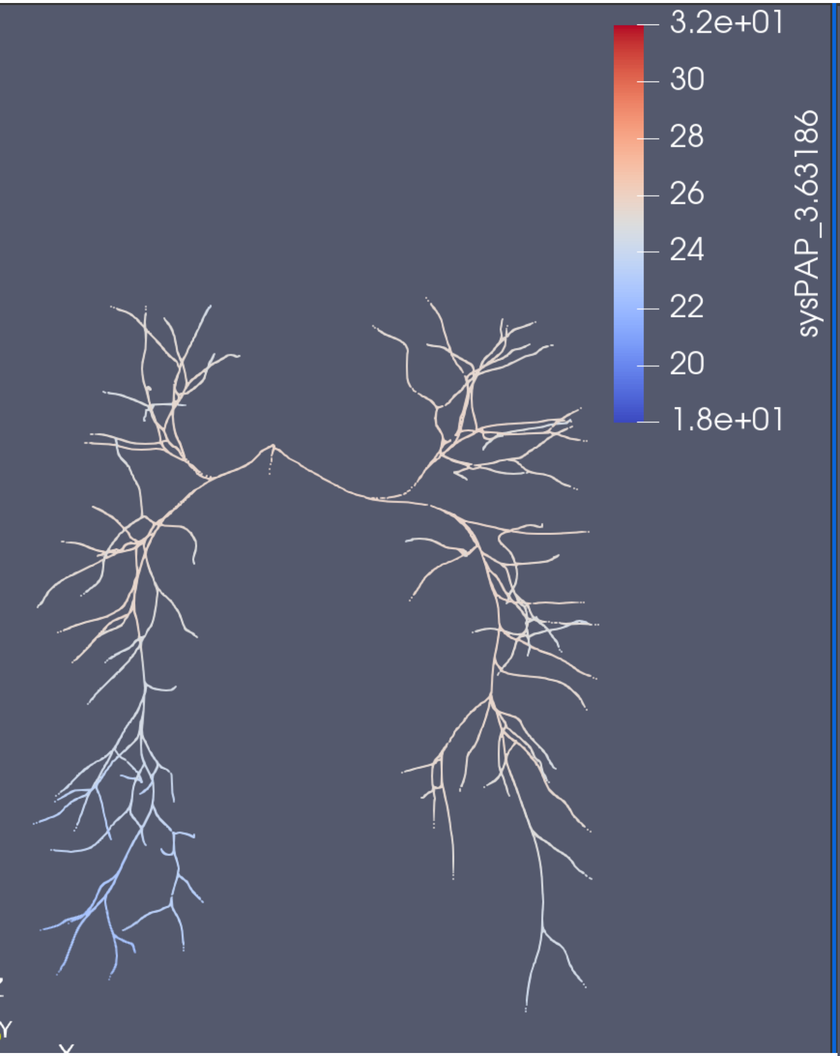
Junction Coefficient Solver



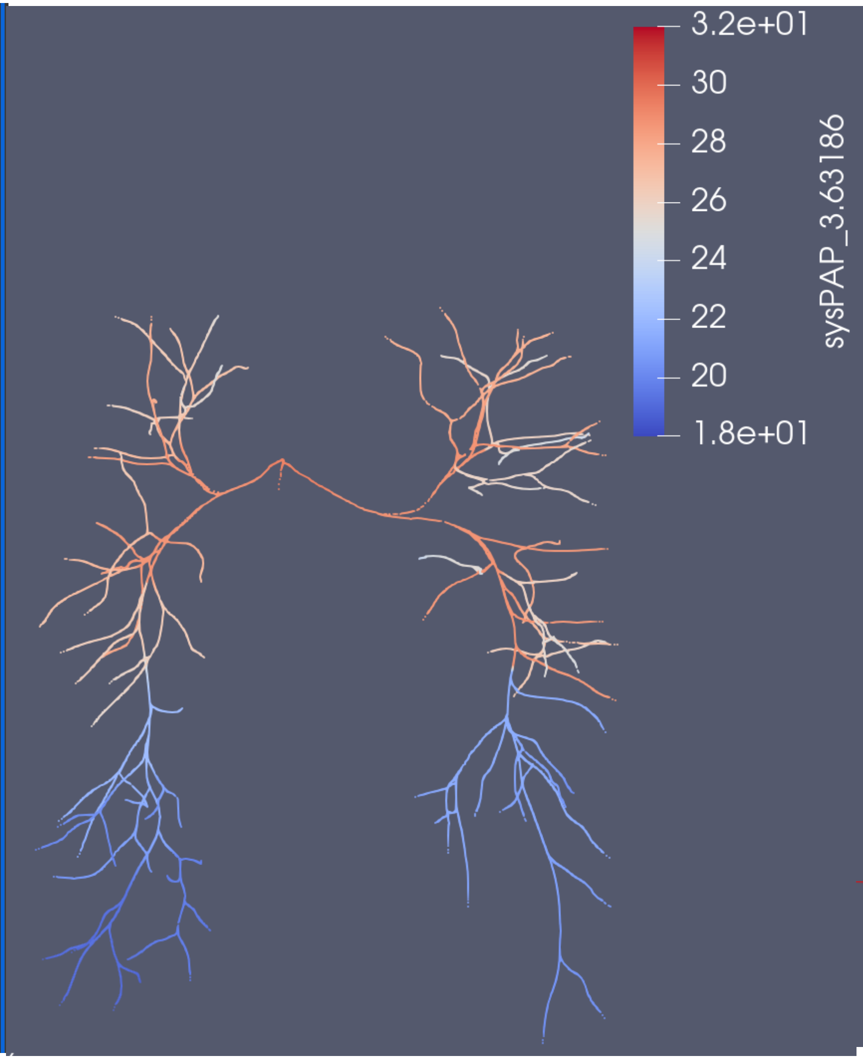
Much more consistent

0080_0001 (sysPAP)

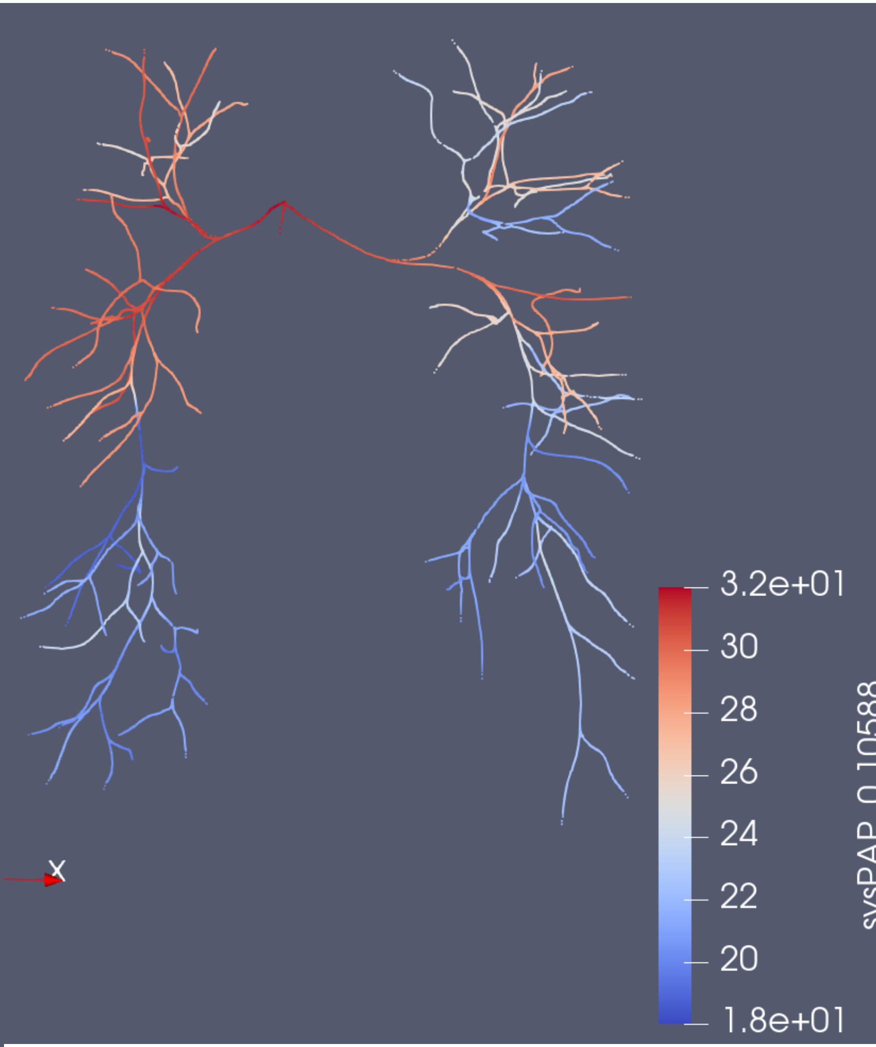
Original Solver



Junction Coefficient Solver

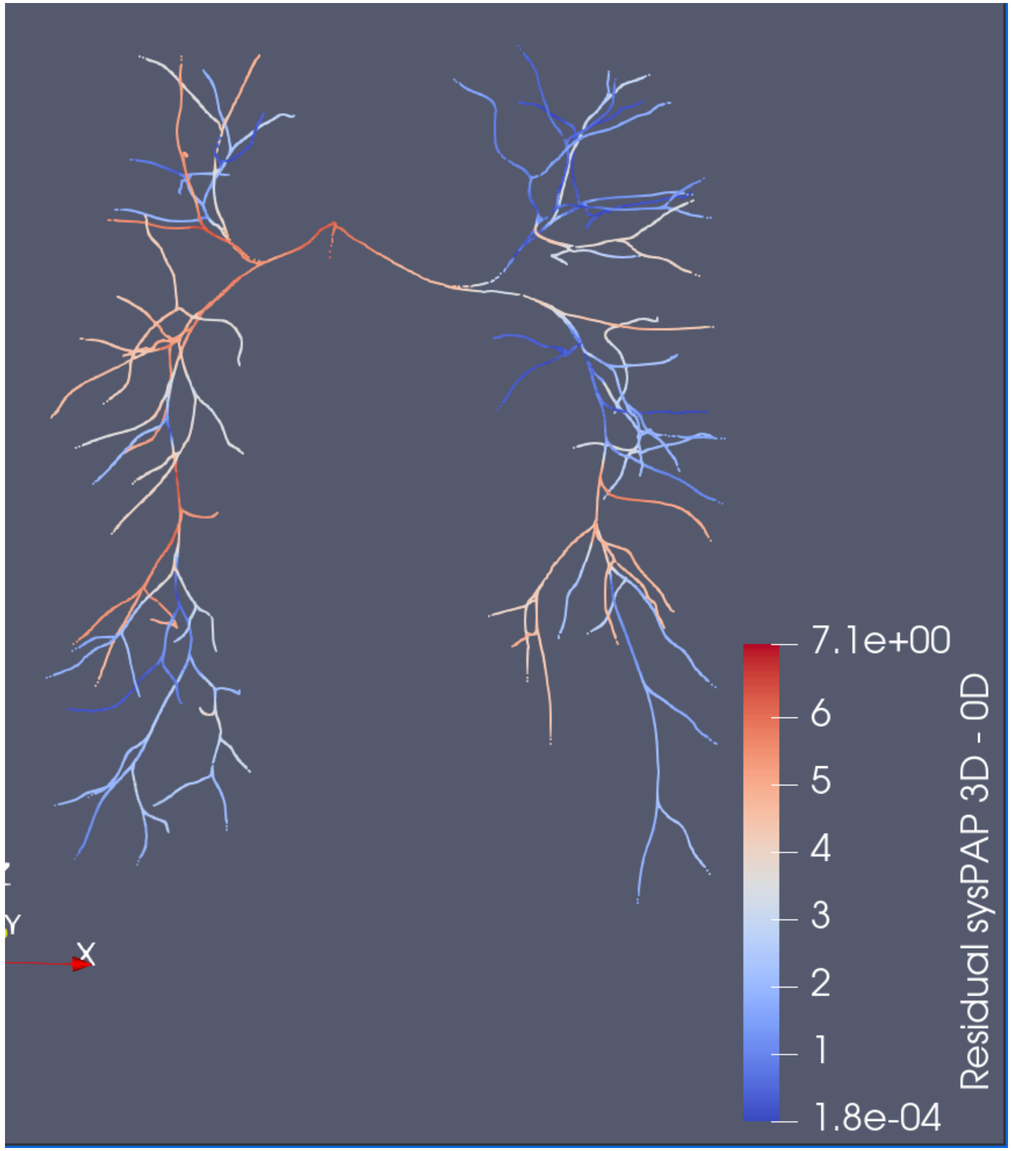


3D

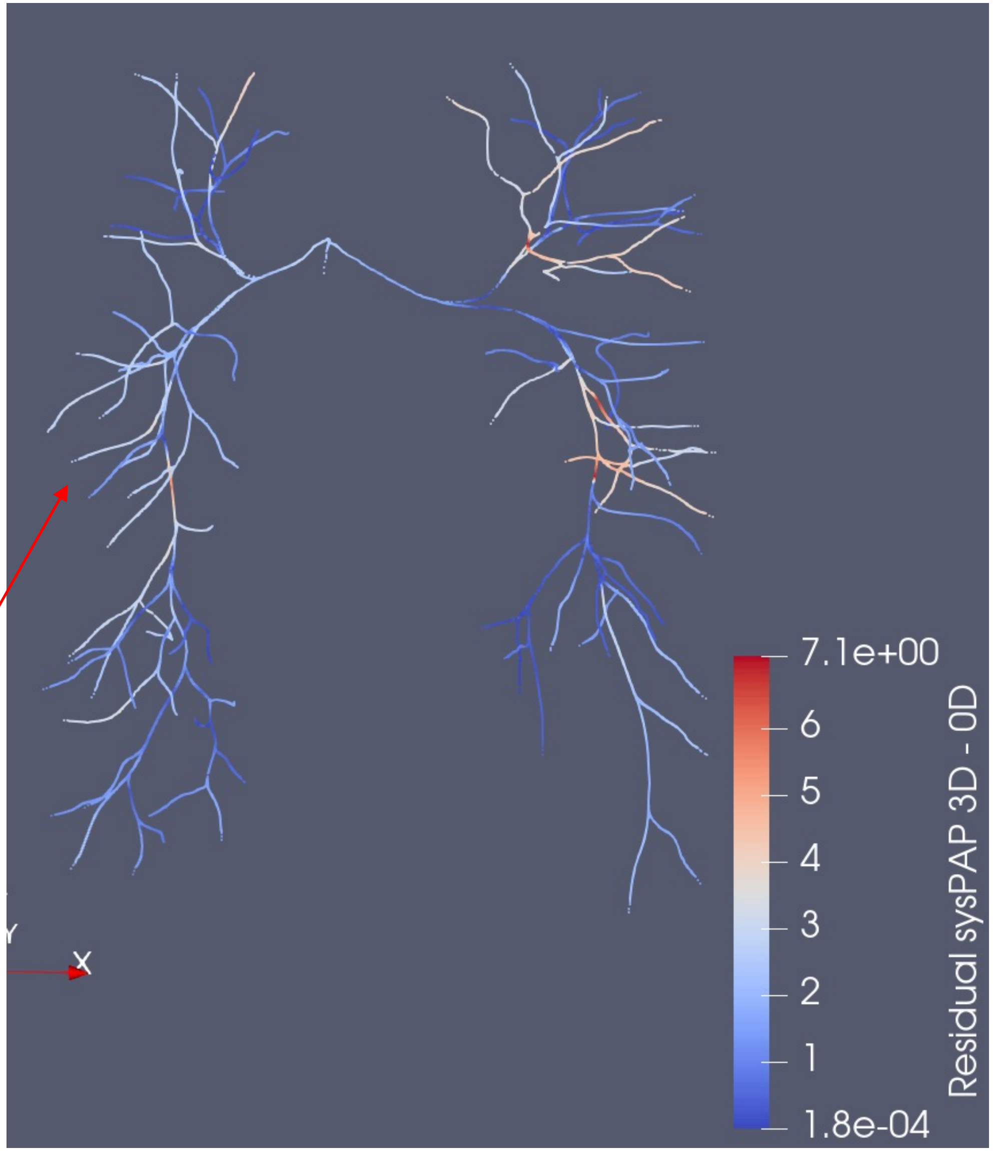


0080_0001 (Residual sysPAP)

Original Solver



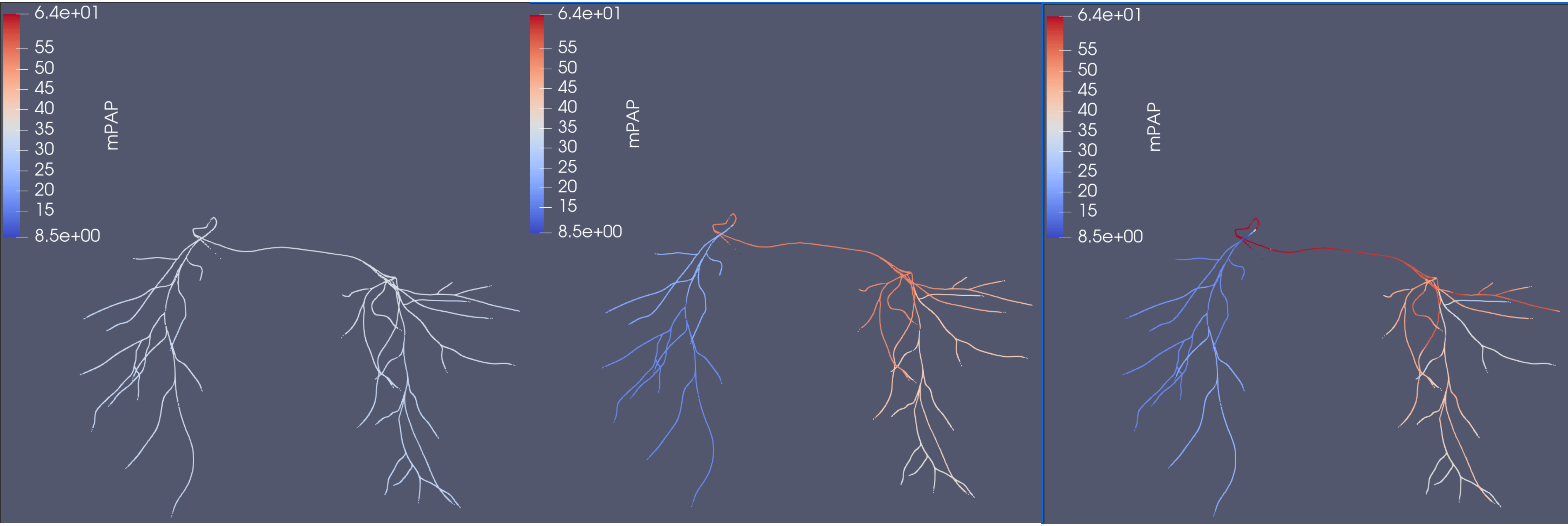
Junction Coefficient Solver



Much more consistent

0118_1000 (meanPAP)

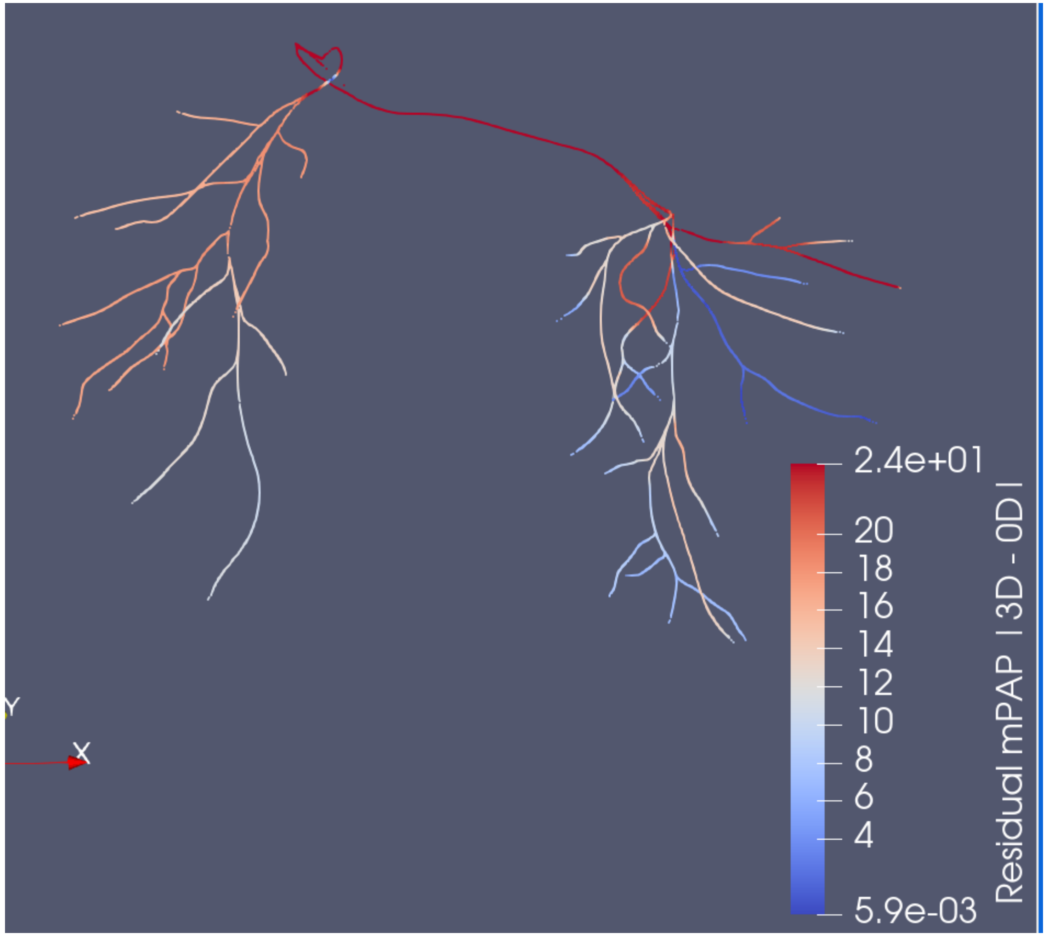
Original Solver | Junction Coefficient Solver | 3D



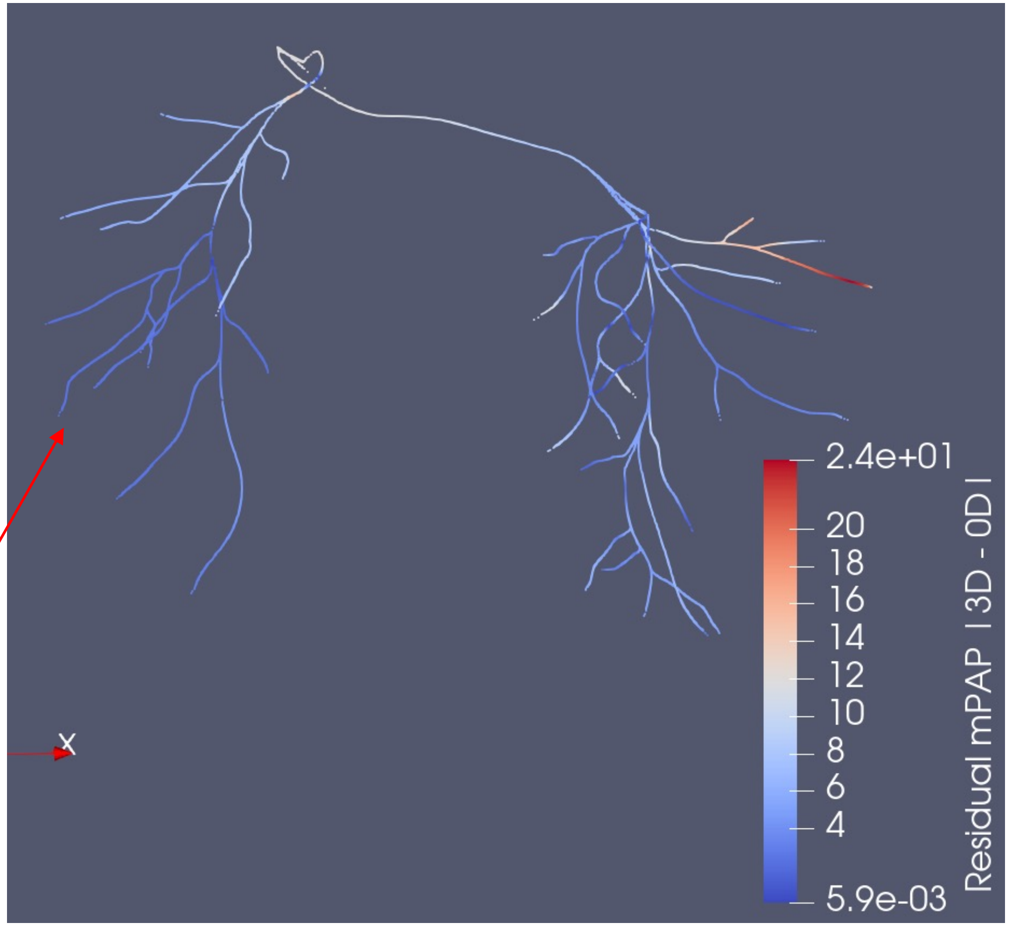
- Evident that the flow distribution and pressure drop is much more realistic

0118_1000 (Residual meanPAP)

Original Solver



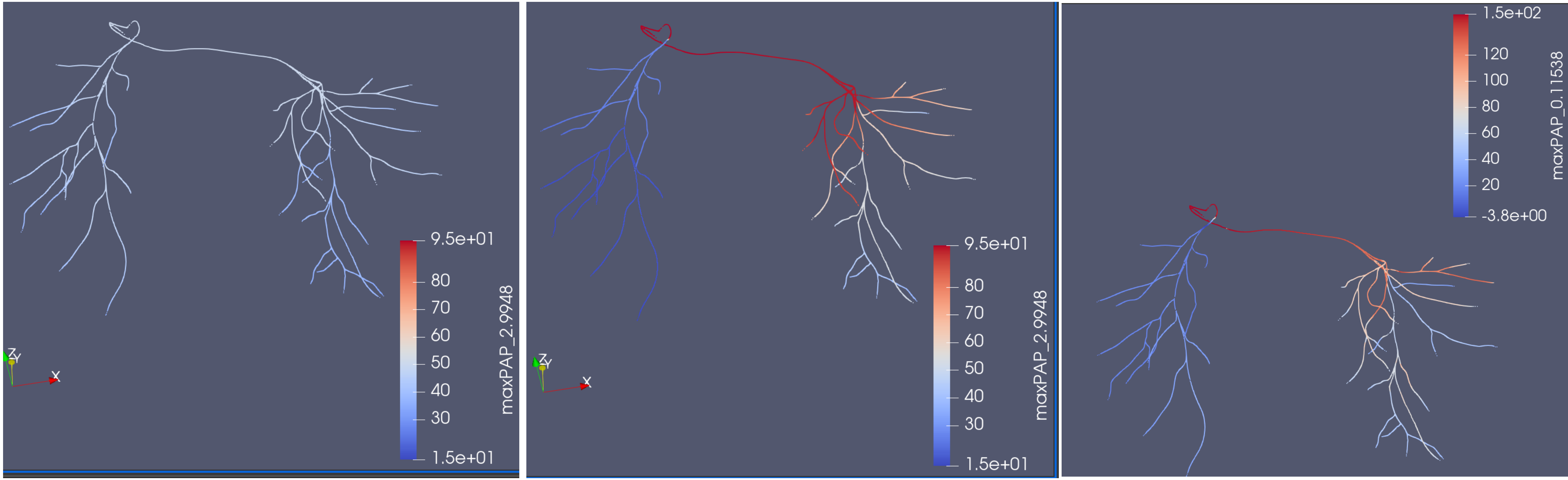
Junction Coefficient Solver



Much more consistent

0118_1000 (sysPAP)

Original Solver | Junction Coefficient Solver | 3D

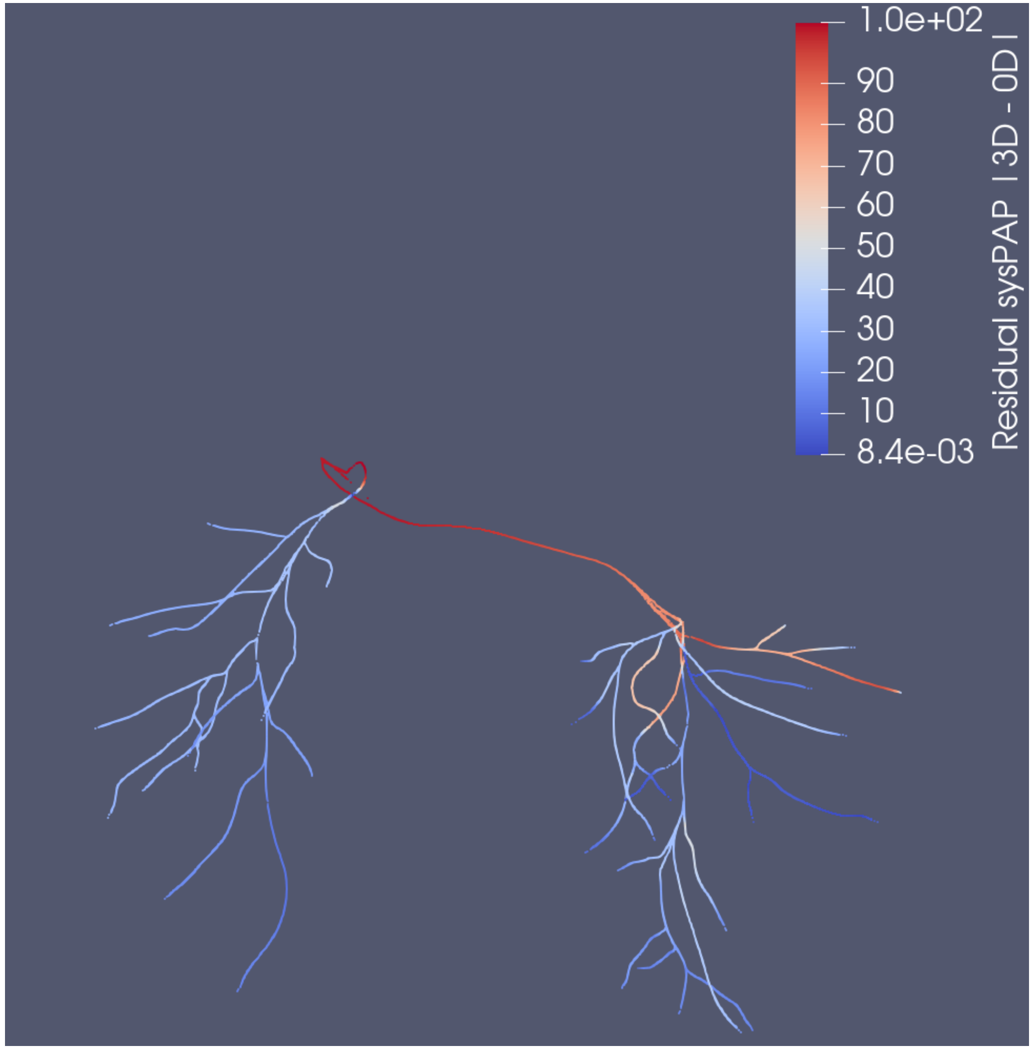


Not to scale

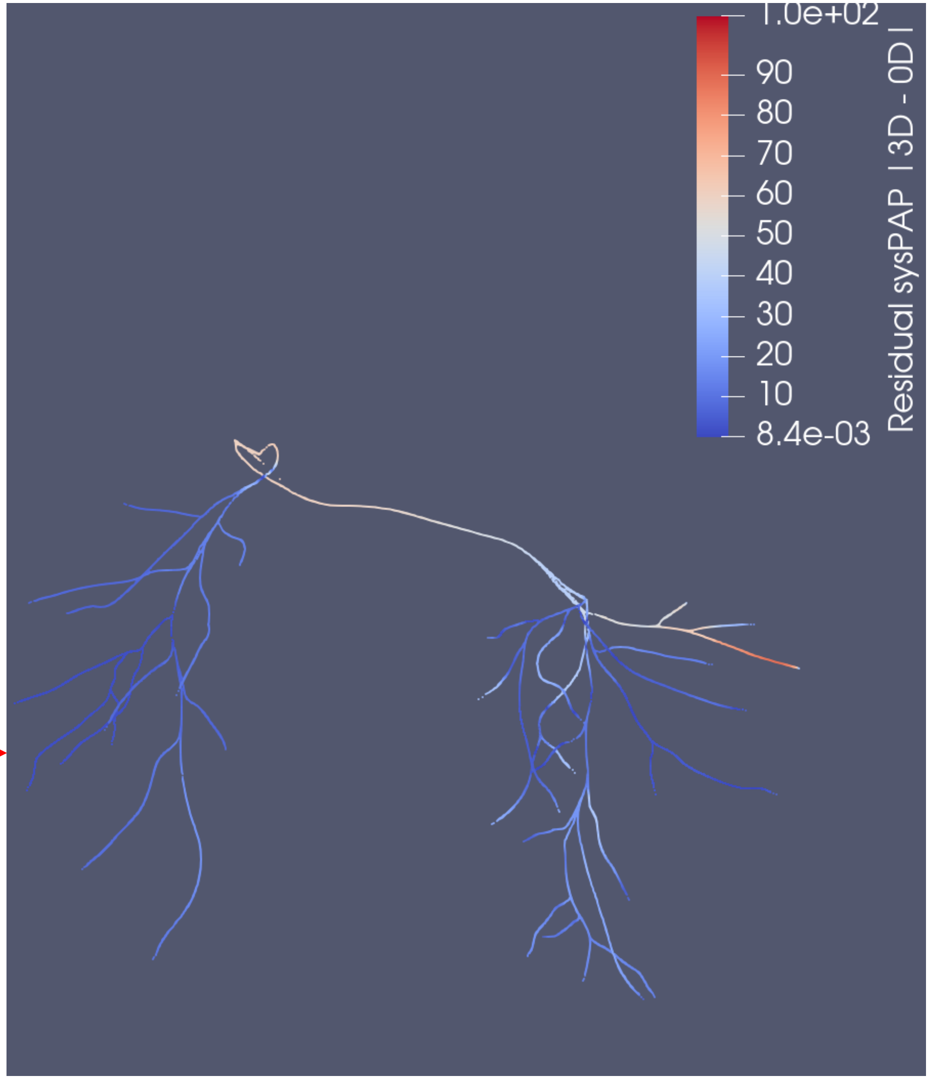
- Although JS Solver is unable to fully capture systole, it is far closer.

0118_1000 (Residual sysPAP)

Original Solver



Junction Coefficient Solver



Much more consistent

Formulas

Inlet/Outlet Errors

$$\epsilon_{P,avg} = \frac{1}{n_{cap}} \sum_{i=1}^{n_{cap}} \frac{\sum_{t=1}^{n_t} |P_{t,i}^{0D} - P_{t,i}^{3D}|}{\sum_{t=1}^{n_t} P_{t,i}^{3D}}$$

$$\epsilon_{P,sys} = \frac{n_t}{n_{cap}} \sum_{i=1}^{n_{cap}} \frac{|P_{t_{sys},i}^{0D} - P_{t_{sys},i}^{3D}|}{\sum_{t=1}^{n_t} P_{t,i}^{3D}}$$

$$\epsilon_{P,dia} = \frac{n_t}{n_{cap}} \sum_{i=1}^{n_{cap}} \frac{|P_{t_{sys},i}^{0D} - P_{t_{sys},i}^{3D}|}{\sum_{t=1}^{n_t} P_{t,i}^{3D}}$$

$$t_{dia} = arg\ min_t Q_{t,inlet}^{3D}$$

$$t_{sys} = arg\ max_t Q_{t,inlet}^{3D}$$

Pressure Drop (Inlet – Outlet) Errors

$$\epsilon_{\Delta P,avg} = \frac{1}{n_{out}} \sum_{i=1}^{n_{out}} \frac{\sum_{t=1}^{n_t} |\Delta P_{t,i}^{0D} - \Delta P_{t,i}^{3D}|}{\sum_{t=1}^{n_t} \Delta P_{t,i}^{3D}}$$

$$\epsilon_{P,sys} = \frac{n_t}{n_{cap}} \sum_{i=1}^{n_{cap}} \frac{|\Delta P_{t_{sys},i}^{0D} - \Delta P_{t_{sys},i}^{3D}|}{\sum_{t=1}^{n_t} \Delta P_{t,i}^{3D}}$$

$$\epsilon_{P,dia} = \frac{n_t}{n_{cap}} \sum_{i=1}^{n_{cap}} \frac{|\Delta P_{t_{sys},i}^{0D} - \Delta P_{t_{sys},i}^{3D}|}{\sum_{t=1}^{n_t} \Delta P_{t,i}^{3D}}$$

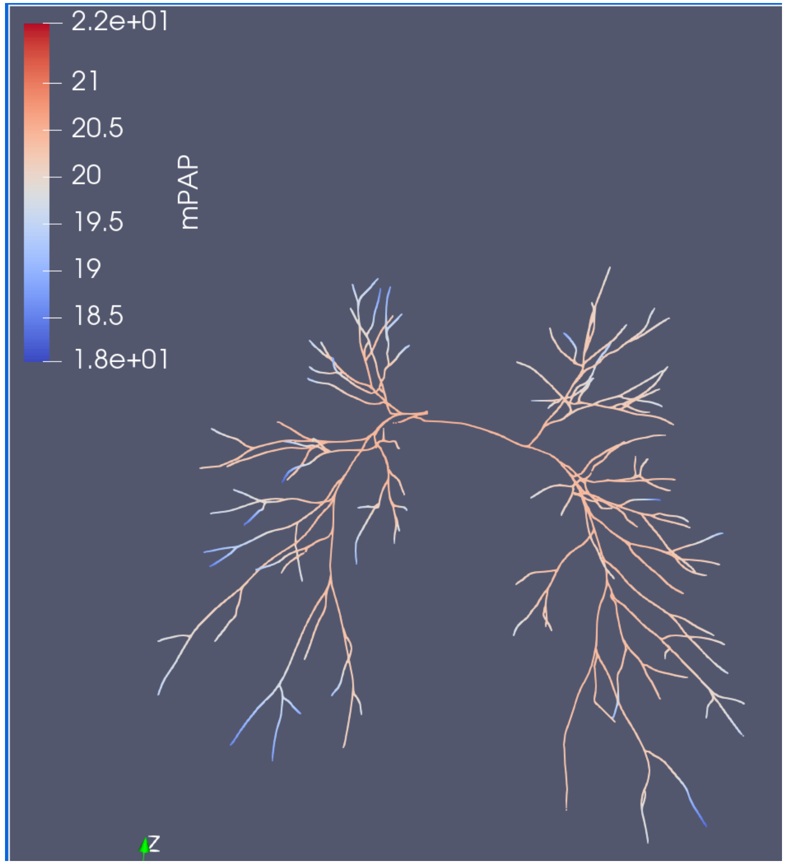
$$\Delta P_{t,i}^{dD} = P_{t,inlet}^{dD} - P_{t,i}^{dD}$$

Quantitative Metrics (3D Relative Errors)

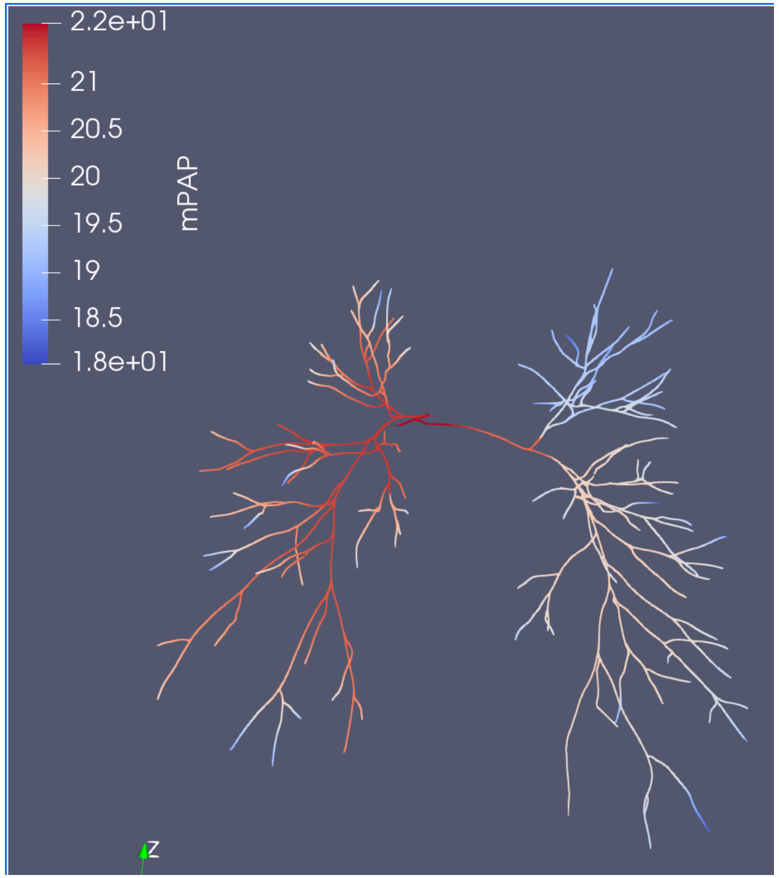
Models	Version	$\epsilon_{P,avg}$	$\epsilon_{P,sys}$	$\epsilon_{P,dia}$	$\epsilon_{\Delta P,avg}$	$\epsilon_{\Delta P,sys}$	$\epsilon_{\Delta P,dia}$
0080_0001	Original	6.69%	21.67%	2.65%	71.21%	205.29%	9.95%
	JC	4.90%	12.31%	2.69%	54.49%	82.33%	9.79%
0118_100	Original	45.19%	82.68%	17.02%	88.13%	322.63%	12.93%
	JC	19.39%	43.10%	6.00%	48.03%	196.50%	5.15%

SU0238 (mPAP)

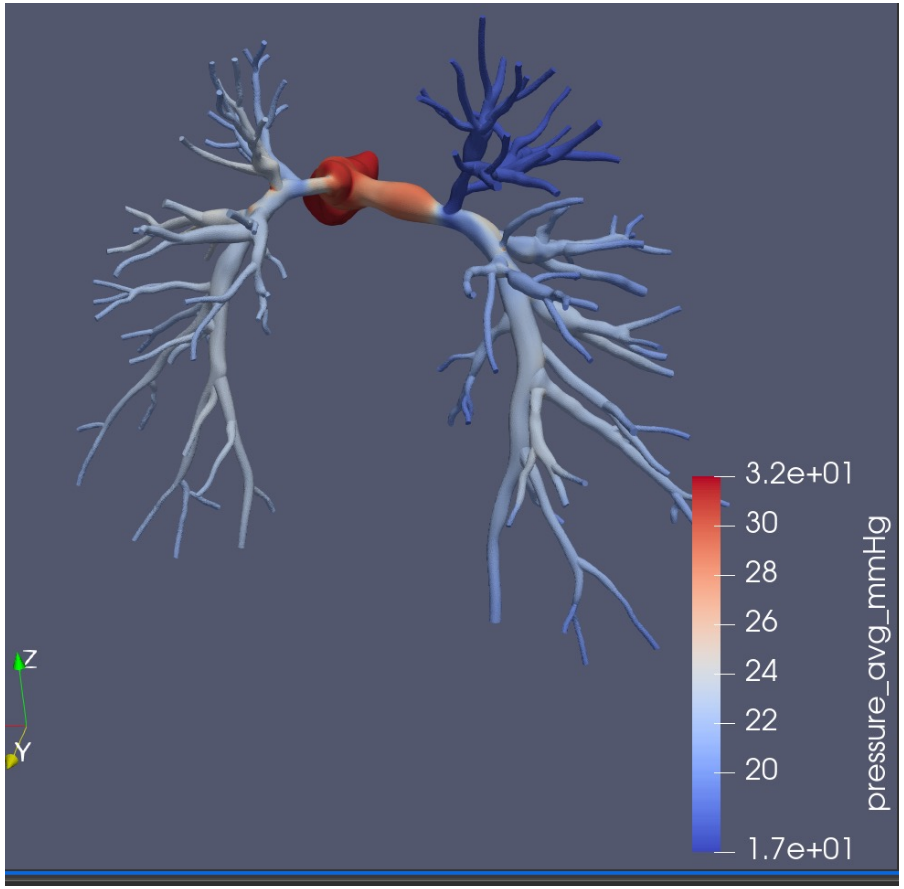
Original Solver



Junction Coefficient Solver



3D

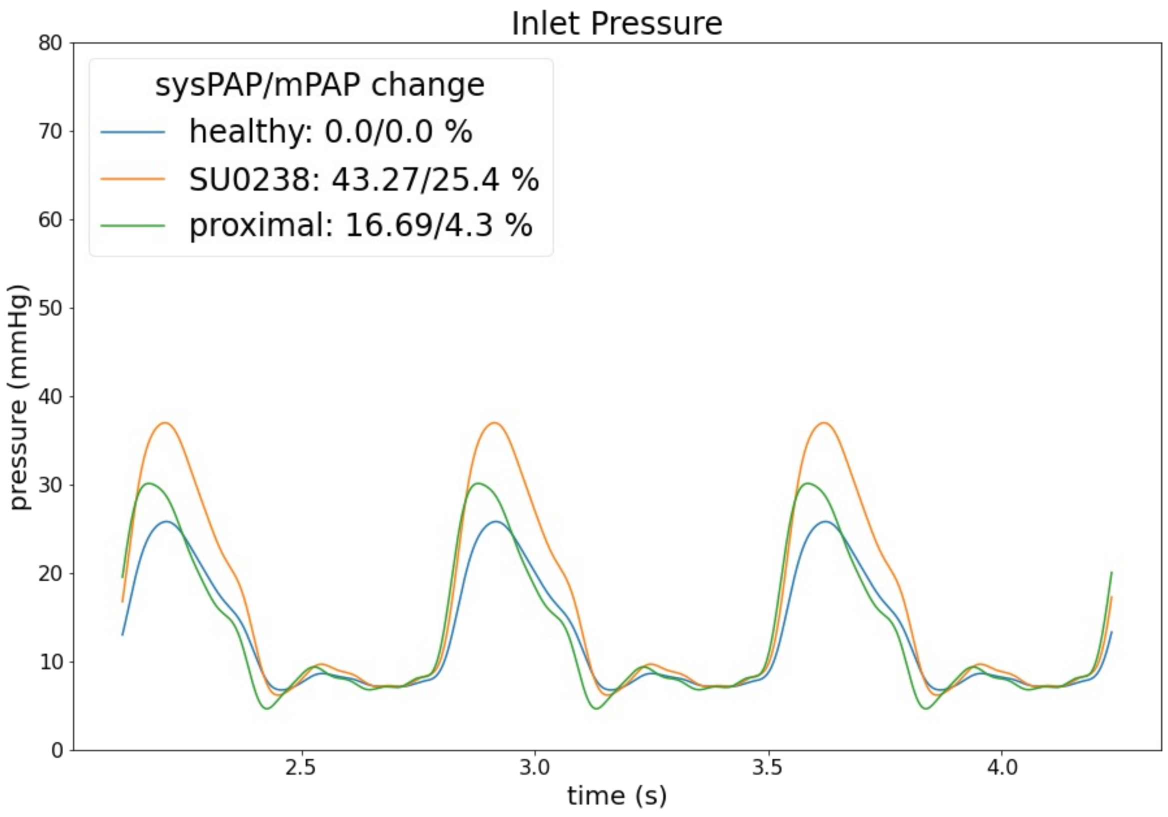


Not to scale

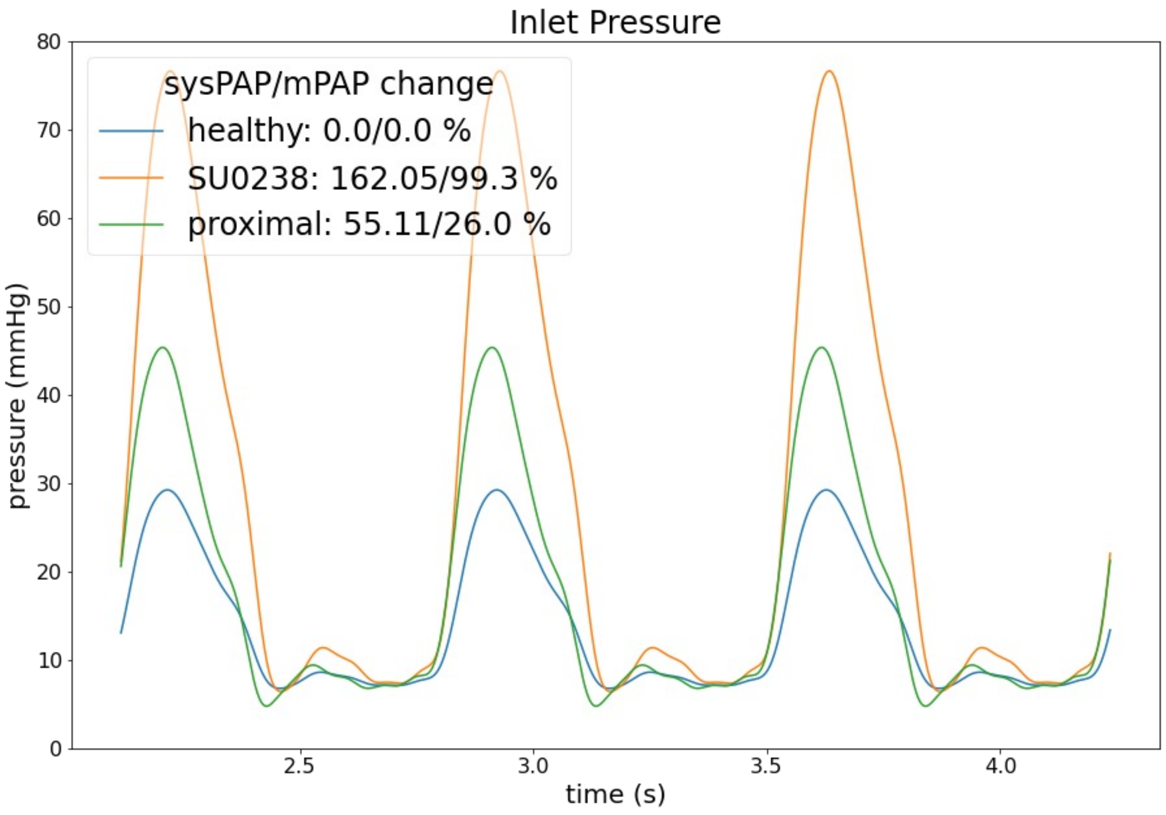
- JC Solver alone is not enough to resolve all problems
- Suspect either insufficient number of vessels to capture rapid shrinkage in MPA radius (not a JC Solver problem)

Artificial Stenosis (0080_0001)

Original

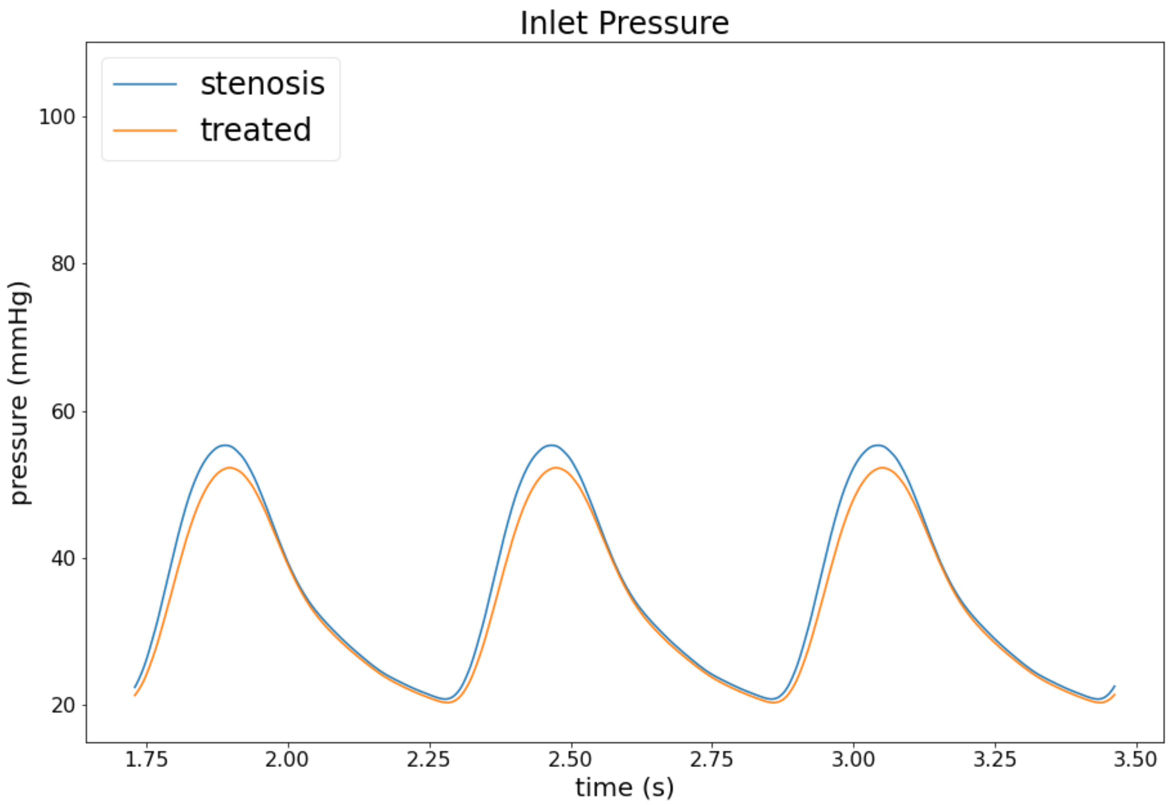


JC

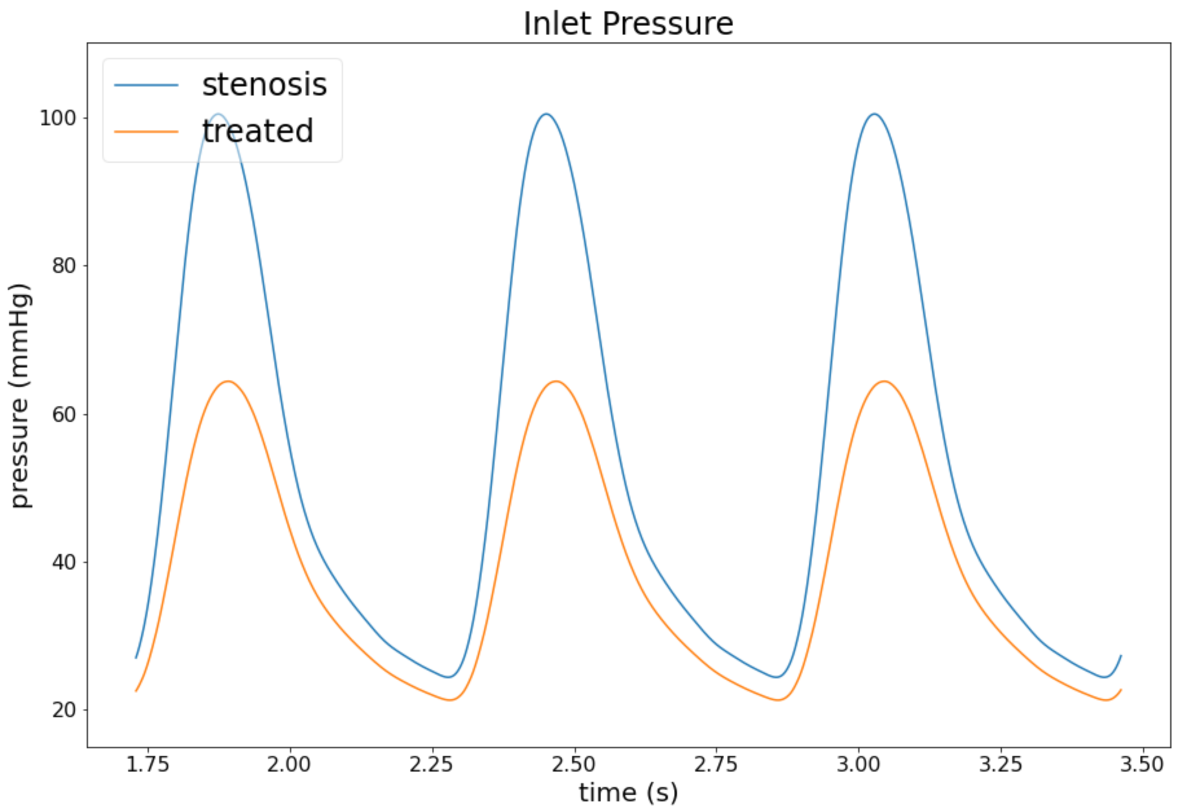


Treatment (0118_0001)

Original



JC



- Significant change
- Similar to previous literature's ~47% reduction

Goal 1&2 – Next Steps

- Temporarily halting progress on stenosis models & focus on Healthy models.
  - Healthy models offer a target treatment for uncertainty quantification, as well as guarantee that stenosis is in the vessels
- Fix up unusable models for more data

Goal 3-6 – Next Steps

- Put Clinician meeting on hold
- Set up a Neural Network
  - Inputs:
    - Vessel Diameter Changes
  - Outputs:
    - Tentatively the pressure/flow outs after the diameter change
    - Discuss with a clinician for more precise outputs
- Sobol Sampling to achieve a more even/desired distribution when sampling diameter changes
- Training on Notre Dame’s CRC computing cluster

Training Set Generation

Stenosis file


```
{
    "all_changed_vessels": [
        14,
        15,
        16,
        159,
        124,
        125,
        126,
        96,
        122,
        8,
        168,
        71,
        48,
```


- Sobol Sampling (Must be in orders of 2) w/ dimension **n**
  - How many data points do I need
    - This determines if I need to rewrite the code for memory efficiency, since I currently immediately load all samples in
- Save order: (According to txt file ->)
  - Alternating Pressure, Flow
  - 2 files, input.npy/output.npy files
- Potential for speedup. Write directly to some sort of database. Optimizations for multiprocessing.


```
array([[1.30957191, 1.95651537, 1.14492055, 1.46177554, 1.70763011,
        2.56526171, 2.48410722, 2.04747433, 1.460812  , 2.23970645,
        2.57515242, 1.71248038, 1.99972515, 1.91396862, 2.55745468,
        1.63051493, 2.33558233, 2.16137826, 2.32593687, 1.41377184,
        2.02099743, 1.61231466, 1.00603265, 1.19018237],
       [2.2799739 , 1.40145287, 2.01269072, 2.03171296, 2.27736571,
        1.61287331, 1.15708365, 1.20901873, 2.09791105, 1.66577677,
        1.34327466, 1.86295623, 1.43332623, 1.38979374, 1.45835443,
        1.8555532 , 1.11471767, 1.34757066, 1.3598813 , 2.02448589,
        1.3731172 , 2.04179576, 2.52589537, 2.31361528],
       [2.12543237, 2.28900871, 1.67807155, 1.00513652, 1.94987218,
        1.03638735, 1.950886  , 1.70605503, 2.19374059, 1.02683705,
        1.70916083, 2.35355593, 2.48740801, 2.52040739, 1.03523926,
        1.09119565, 1.651111  , 2.29533936, 1.51121843, 1.34091386,
        2.48709416, 2.33501978, 1.4700775 , 1.70256523],
```



```
array([[1.34413966e+04, 2.58828629e+01, 1.32451301e+04, 2.58830679e+01,
        1.15694245e+04, 2.58831227e+01, 1.15666446e+04, 1.35998353e+00,
        1.15694137e+04, 5.40391445e-01, 1.15694117e+04, 5.40406179e-01,
        1.15694094e+04, 5.40419994e-01, 1.09598818e+04, 1.66421990e+00,
        1.32259117e+04, 1.48782918e+00, 1.23002564e+04, 1.41668200e+01,
        1.04059840e+04, 7.72576728e-01, 1.11480964e+04, 1.81914472e+00,
        1.09674587e+04, 1.13407907e+00, 1.06082570e+04, 1.13410907e+00,
        1.04956330e+04, 1.13415762e+00, 1.15036568e+04, 5.92004767e-01,
        1.13850535e+04, 5.66466527e-01, 1.13067127e+04, 1.16534112e+00,
        1.13662459e+04, 5.75753878e-01, 1.12149799e+04, 4.27149589e-01,
        1.11572948e+04, 4.27158814e-01, 1.10147626e+04, 4.27164363e-01,
        1.03891058e+04, 9.08086919e-01, 1.11613367e+04, 1.23953131e+00],
       [1.39056345e+04, 2.95142976e+01, 1.29146198e+04, 2.95143990e+01,
        1.26817990e+04, 2.95145903e+01, 1.26809049e+04, 1.66719388e+00,
        1.26817950e+04, 6.60912801e-01, 1.26817783e+04, 6.60919804e-01,
        1.26817142e+04, 6.60923408e-01, 9.68343823e+03, 1.17967921e+00,
        1.36402321e+04, 1.58020545e+00, 1.07272038e+04, 1.04680435e+01,
        1.07179610e+04, 8.35539620e-01, 1.13678237e+04, 1.90655350e+00
```


Input.npy (top)
Output.npy (bottom)

Neural Network

- Input (n)
- Hidden (10 layers * 30 Neurons per layer)
- Output (2n)
- ReLU after each layer
- No BatchNorms yet.

- Batch size = 128
- SGD, lr = 1e-5, momentum = .9

Thank You!