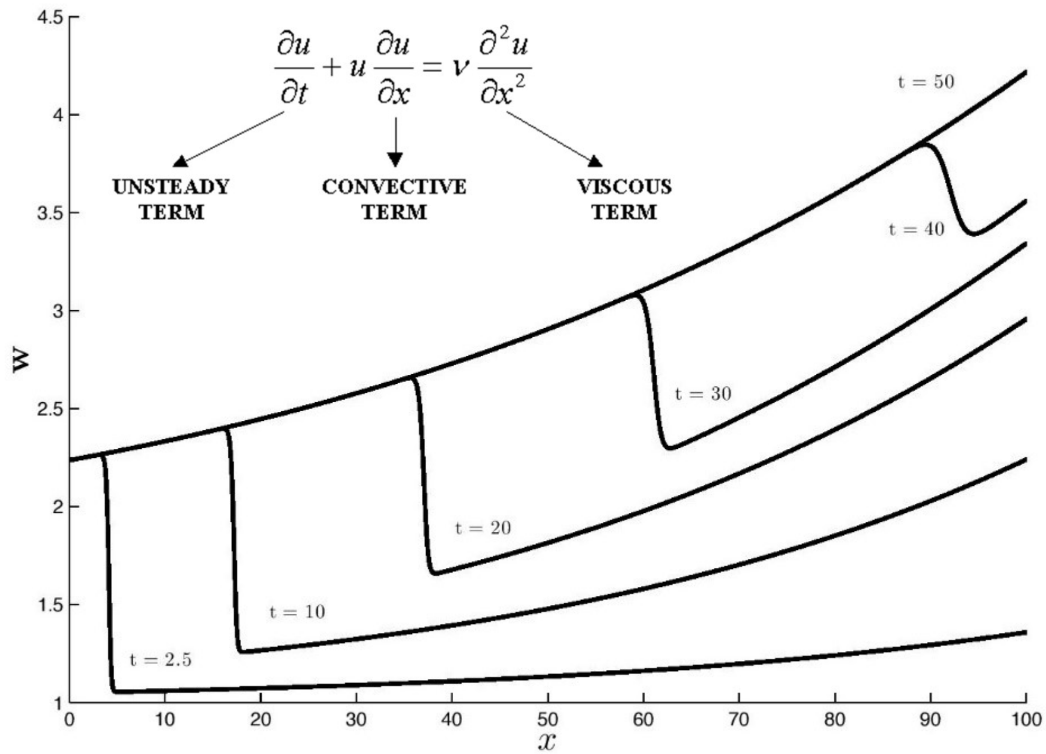
A point selection method for hyper reduction in 2D

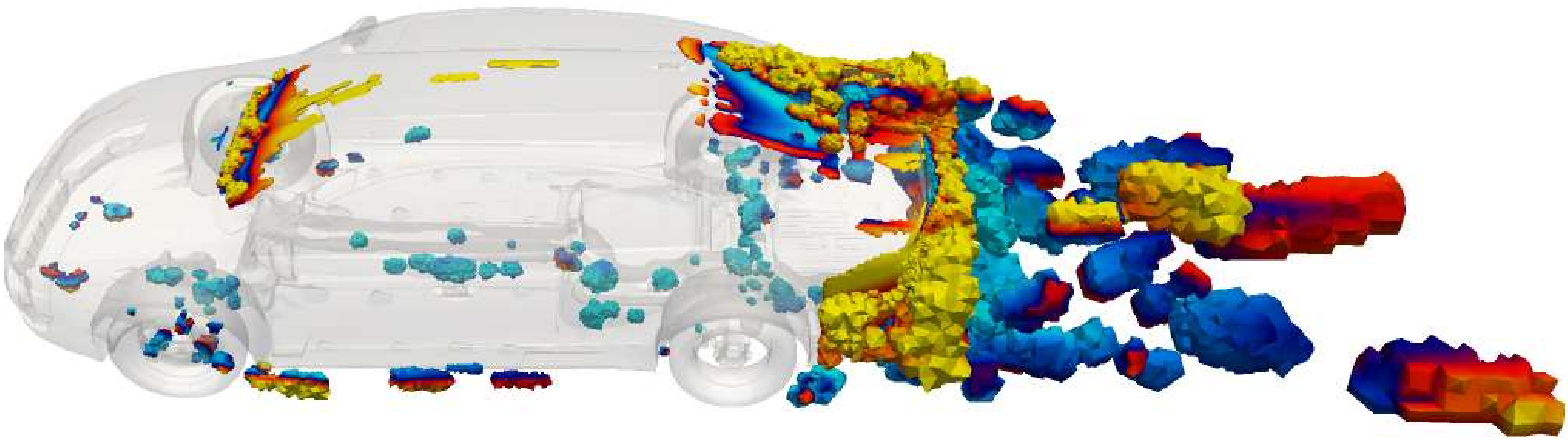
1/9/2017

TINA WHITE

Burgers' Equation Solutions

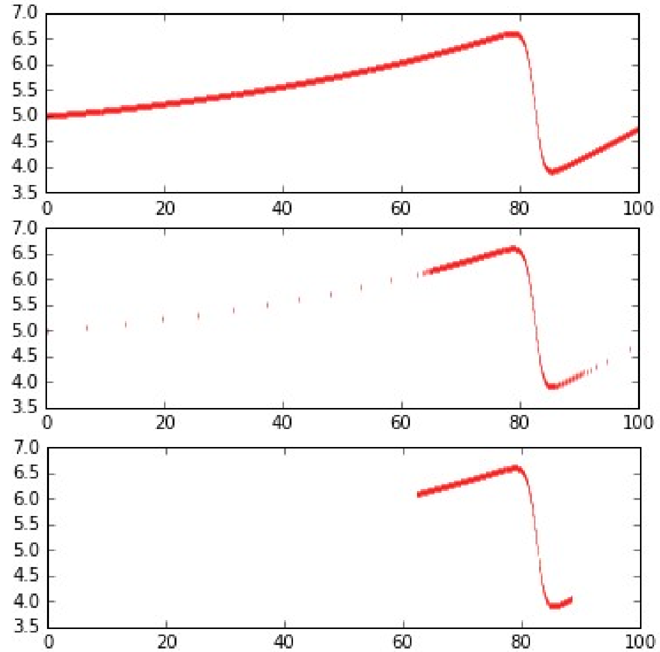


Hyper Reduction



Volkswagen Passat

Motivation



Original Points

Method 1?

Method 2?

Baseline sample points selection method

Baseline method (greedy algorithm from Kyle's thesis, based on Carlberg's 2009 paper)

– favors individual energetic points, doesn't consider their location in space

Algorithm 5 Greedy algorithm for computing sample indices.

Input: $\Phi_R(n_{x_R}, X_R)$, $\Phi_J(n_{x_J}, X_J)$, n_R, n_J, n_{i}
Output: $\mathcal{I}$

$\mathcal{I} = \emptyset$, $\bar{n}_{\mathrm{i}} = 0$, $m = 1$
$\mathsf{R} \leftarrow \phi_R^1$, $\mathsf{J} \leftarrow \phi_J^1$
while $\bar{n}_{\mathrm{i}} < n_{\mathrm{i}}$ **do**
  $i \leftarrow \arg \max_{l \in \{1,\dots,N\} \setminus \mathcal{I}} \left((\mathsf{R}_l)^2 + (\mathsf{J}_l)^2 \right)$
  $\mathcal{K} \leftarrow \{k \in \{1, \dots, N\} \setminus (\mathcal{I} + i) \mid \mathcal{J}(k) = \mathcal{J}(i)\}$
  $\mathcal{I} \leftarrow \mathcal{I} + i + \mathcal{K}$
  $\bar{n}_{\mathrm{i}} \leftarrow \bar{n}_{\mathrm{i}} + 1 + \dim \mathcal{K}$
  $m \leftarrow m + 1$
  $p_R = \min(m-1, n_R)$, $p_J = \min(m-1, n_J)$
  $\mathsf{R} \leftarrow \phi_R^m - \left[\phi_R^1 \ \cdots \ \phi_R^{p_R}\right] \phi_{Rr}^m$, where $\phi_{Rr}^m = \arg \min_{a \in \mathbb{R}^{\bar{n}_{\mathrm{i}}}} \left\| \left[\hat{\phi}_R^1 \ \cdots \ \hat{\phi}_R^{p_R}\right] a - \hat{\phi}_R^m \right\|_2$
  $\mathsf{J} \leftarrow \phi_J^m - \left[\phi_J^1 \ \cdots \ \phi_J^{p_J}\right] \phi_{Jr}^m$, where $\phi_{Jr}^m = \arg \min_{a \in \mathbb{R}^{\bar{n}_{\mathrm{i}}}} \left\| \left[\hat{\phi}_J^1 \ \cdots \ \hat{\phi}_J^{p_J}\right] a - \hat{\phi}_J^m \right\|_2$
end while

Carlberg's GNAT 2009 paper

Algorithm 4 Selection of the masks.

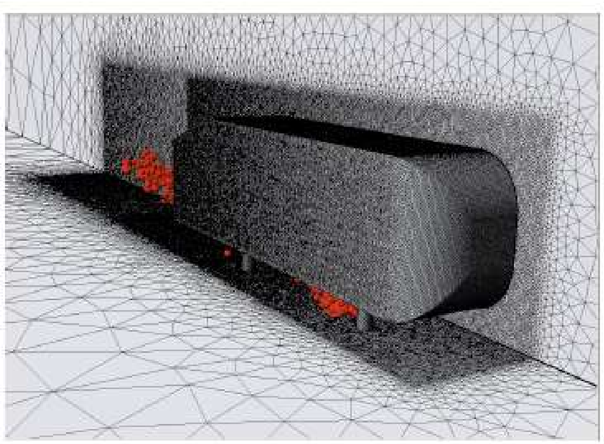
Input: Desired number of sampled nodes n_{SN}, and the ROB for the nonlinear terms, $\mathbf{\Psi} = [\psi_1, \dots, \psi_k] \in \mathbb{R}^{n \times k}$
Outputs: $\mathcal{E}$, $\mathcal{E}'$

Find $\xi = \mathrm{nodeWithMax}\left(|\psi_1|\right)$
Identify the degrees of freedom $\{e_{(\xi, i_{\mathrm{DOF}})}\}_{i_{\mathrm{DOF}}=1}^{n_{\mathrm{DOF}}}$ associated with node ξ
Set $\mathcal{E} = \{e_{(\xi,1)}, \cdots, e_{(\xi, n_{\mathrm{DOF}})}\}$
$n_{\mathrm{nodesToAdd}} = \mathrm{ceil}\left(n_{\mathrm{SN}}/k\right)$
for $i_{\mathrm{vec}} = 2, \cdots, k$ **do**
  Set $\mathbf{U} = [\psi_1, \cdots, \psi_{i_{\mathrm{vec}}-1}]$
  for $i_{\mathrm{node}} = 1, \cdots, n_{\mathrm{nodesToAdd}}$ **do**
    Compute masked quantities $\overline{\overline{\psi}}_{i_{\mathrm{vec}}}$ and $\overline{\overline{\mathbf{U}}}$ corresponding to $\mathcal{E}$
    Compute gappy reconstruction $\widetilde{\psi_{i_{\mathrm{vec}}}} = \mathbf{U} \overline{\overline{\mathbf{U}}}^{+} \overline{\overline{\psi}}_{i_{\mathrm{vec}}}$
    Find $\xi = \mathrm{nodeWithMax}\left(|\psi_{i_{\mathrm{vec}}} - \widetilde{\psi_{i_{\mathrm{vec}}}}|\right)$
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{e_{(\xi,1)}, \cdots, e_{(\xi, n_{\mathrm{DOF}})}\}$
  end for
end for
Identify $\mathcal{E}'$, the degrees of freedom necessary to evaluate the residual and Jacobian at $\mathcal{E}$.

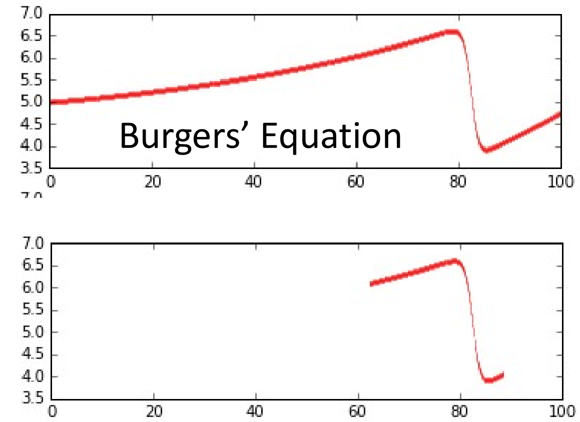
Kyle's Thesis Chapter 2

Baseline sample points selection method

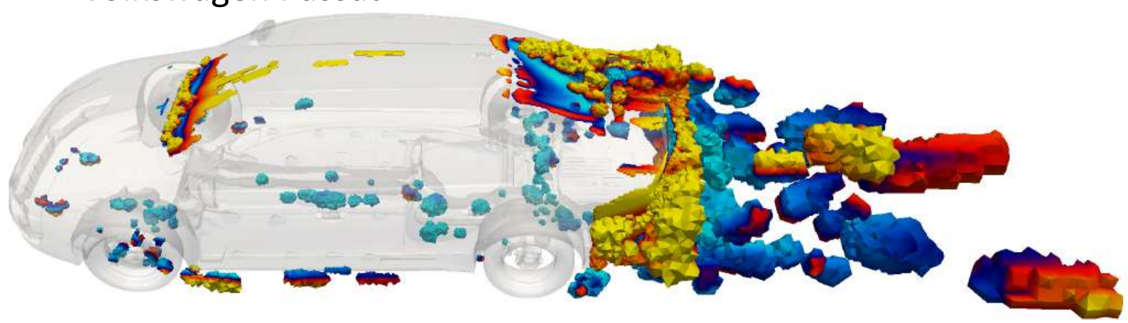
Baseline method – visualized



Ahmed body mesh

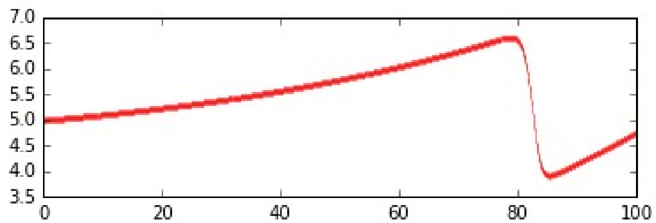


Volkswagen Passat

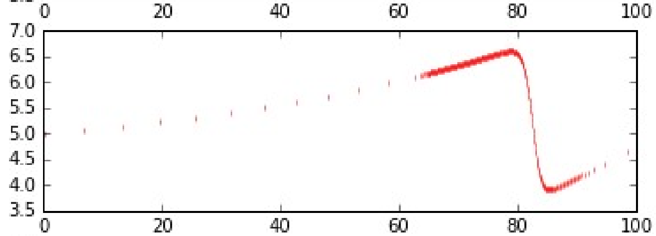


New sample points selection method

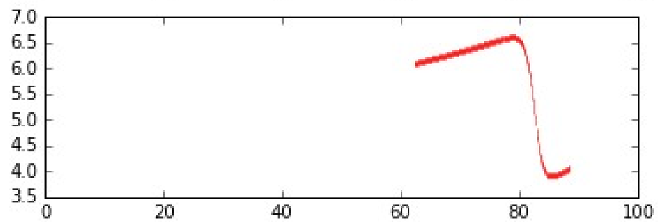
Motivation – it’s better to represent the remainder of the flow field sparsely than not at all



Original Points, n = 1000 points

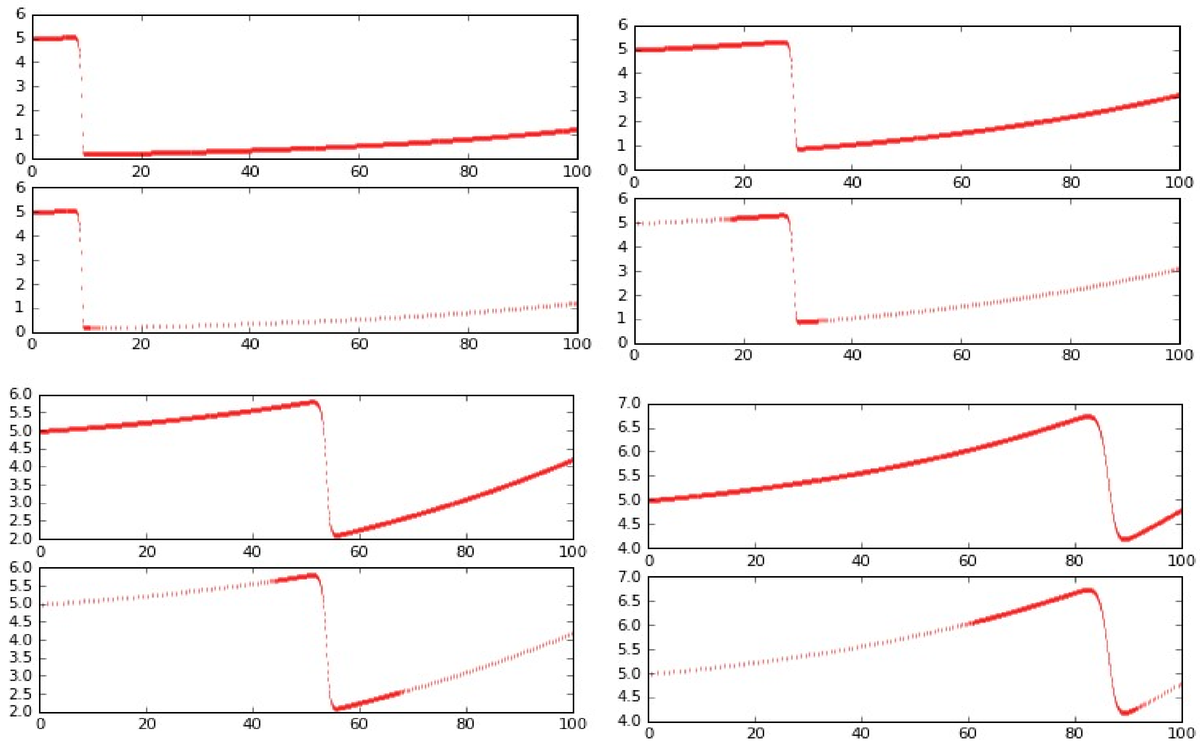


NEW Selection Method, n = 250 points



Baseline Selection Method, n = 250 points

New sample points selection method



New sample points selection method

Summary of the two point selection method changes

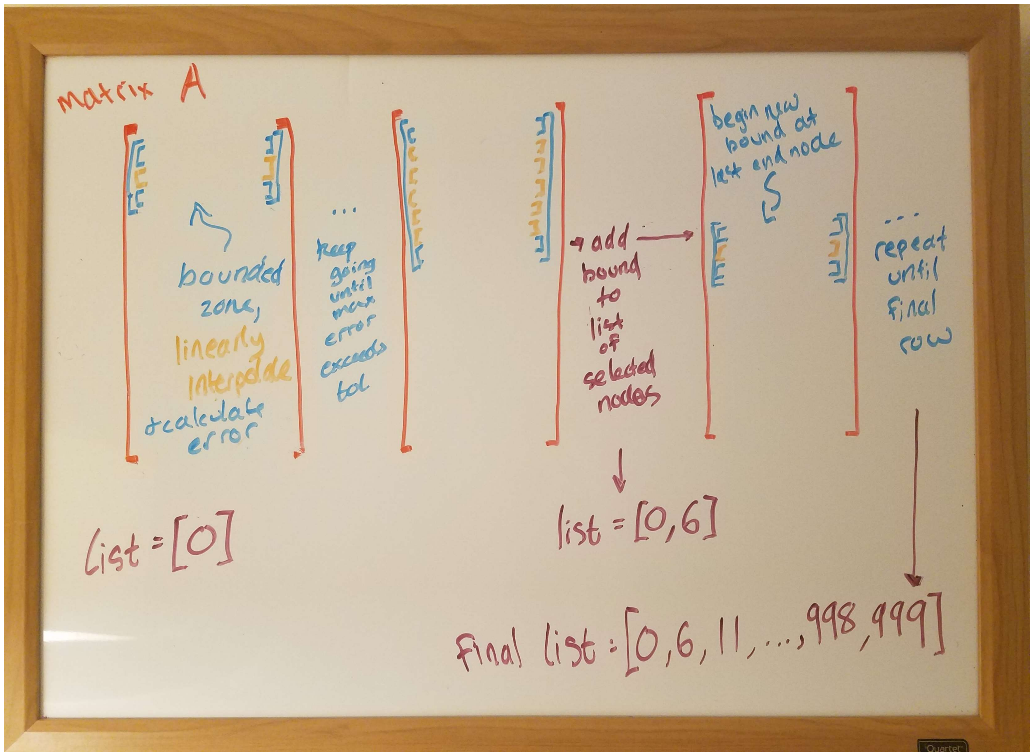
- First change
  - The baseline method uses the matrix of singular vectors to select points
  - The new method uses the matrix of snapshots to select points
- Second change
  - The baseline method chooses the individual points with the highest energy
  - The new method chooses location-based zones of points with the highest energy

It's unclear which change results in the increases in speed and accuracy, or whether both are necessary to see the reported gains.

The two changes cannot be evaluated separately because the baseline selection method is designed to work with a matrix of singular vectors, and the new selection method is designed to work with a matrix of snapshots. Given the wrong type of input matrix, both methods perform worse.

New sample points selection method

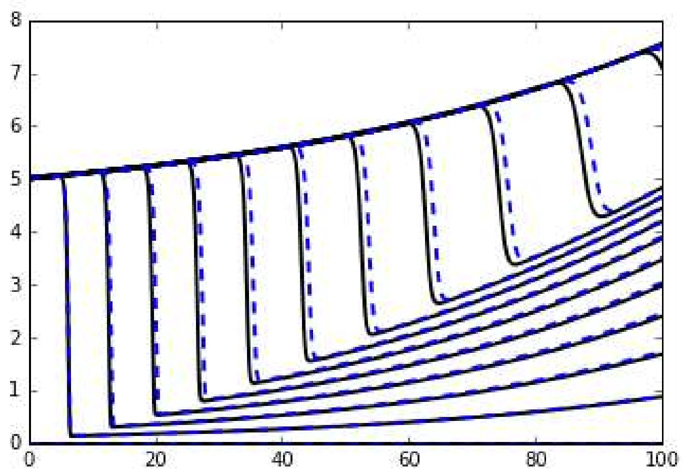
Visualized



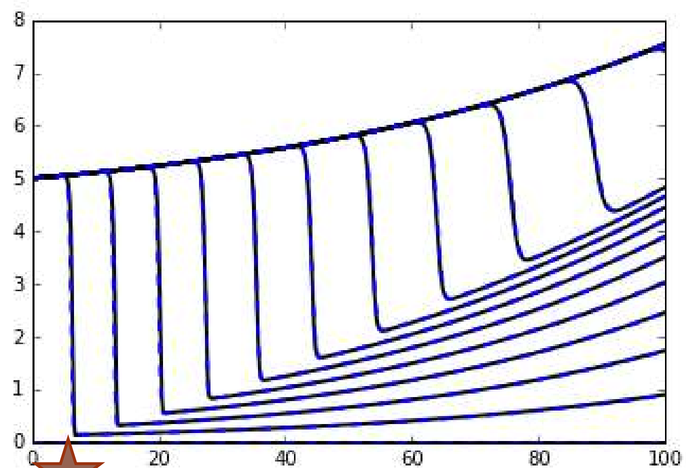
New sample points selection method

The new sampling method is faster and more accurate (3-6x)

- Example result 1



Baseline Selection Method, n = 151
Residual = 0.04149
5.2068 seconds



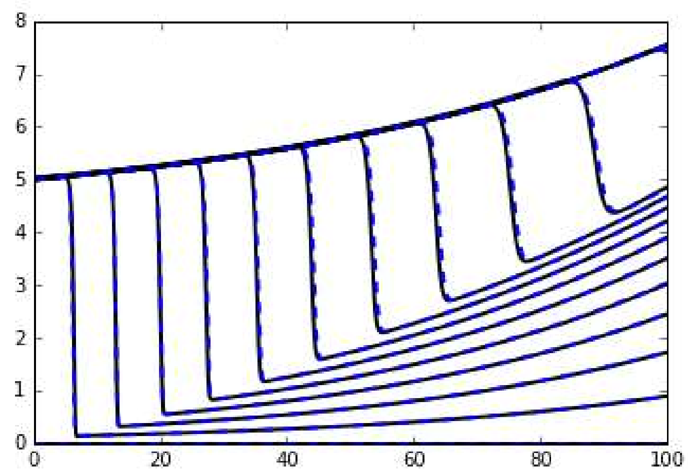
NEW Selection Method n ~ 151
Residual = 0.00486
5.0471 seconds

IMPROVED

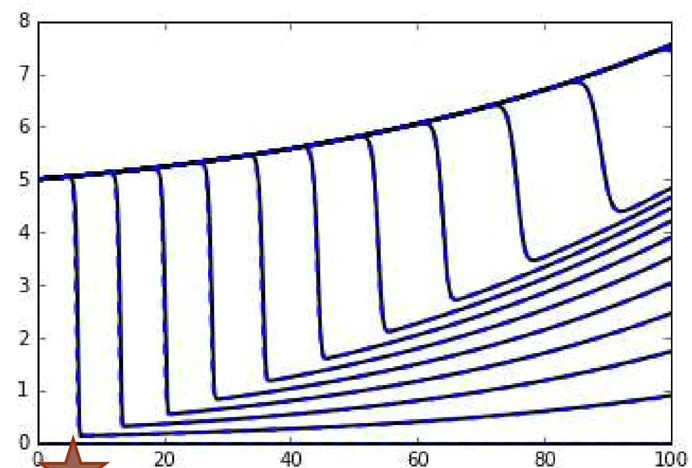
New sample points selection method

The new sampling method is faster and more accurate (3-6x)

- Example result 2:



Baseline Selection Method, n = 151
Residual = 0.01603
5.2253 seconds



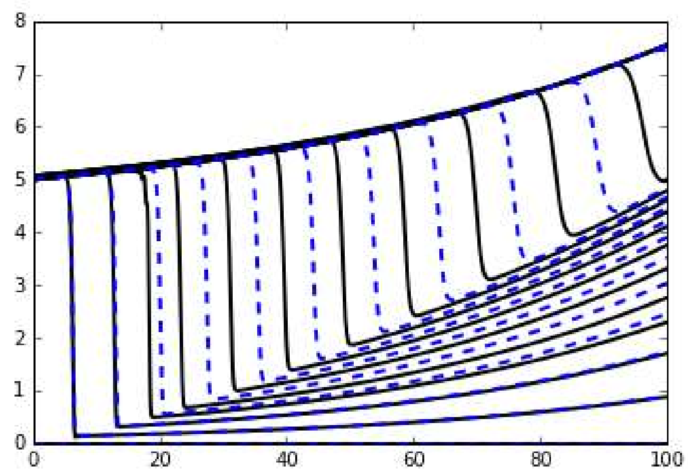
NEW Selection Method n ~ 151
Residual = 0.00339
5.1512 seconds

IMPROVED

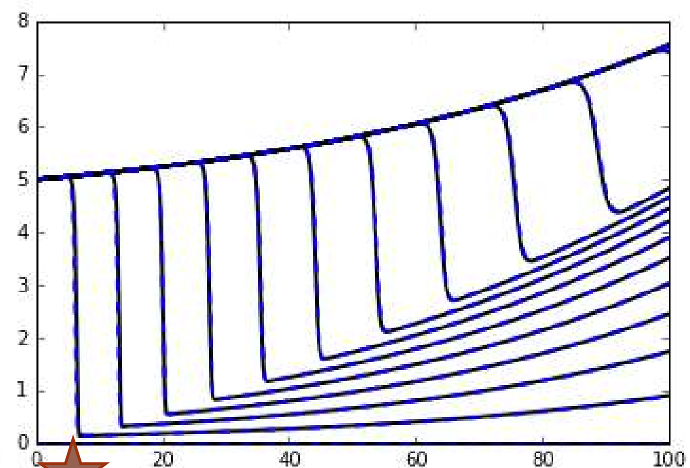
New sample points selection method

The new sampling method is faster and more accurate (3-6x)

- Example result 3



Baseline Selection Method, n = 151
Residual = 0.15968
5.2676 seconds



NEW Selection Method n ~ 151
Residual = 0.00562
5.1748 seconds

IMPROVED

Overview of 1D Results

The new point selection method

- Improves online speed and accuracy over baseline method
- Slows offline speed due partly to lack of code optimization (this has been improved)
- Appears more robust to clustering variation over baseline methods
- Has two non-standard implementation details
  - It uses the matrix of snapshots to select points instead of the matrix of singular vectors
  - It chooses points based on error within nearby zones of points instead of individual point energy

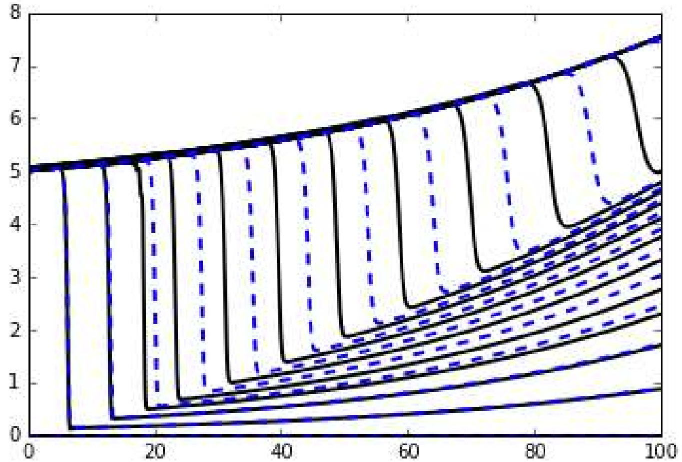
Autumn 2016 Plans

Improvements to point selection method

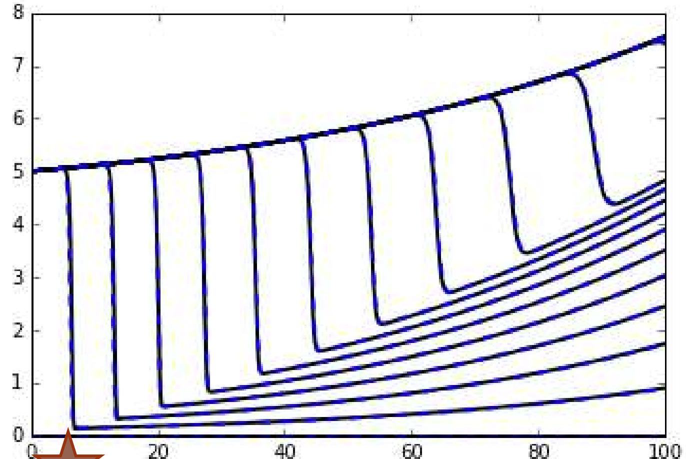
- Write more general form of selection algorithm for 2D data - DONE
- Optimize offline point selection for efficiency in 1D - DONE

Point sampling offline speed increase in 1D

The new sampling method is faster online and more accurate (3-6x) – slightly slower offline



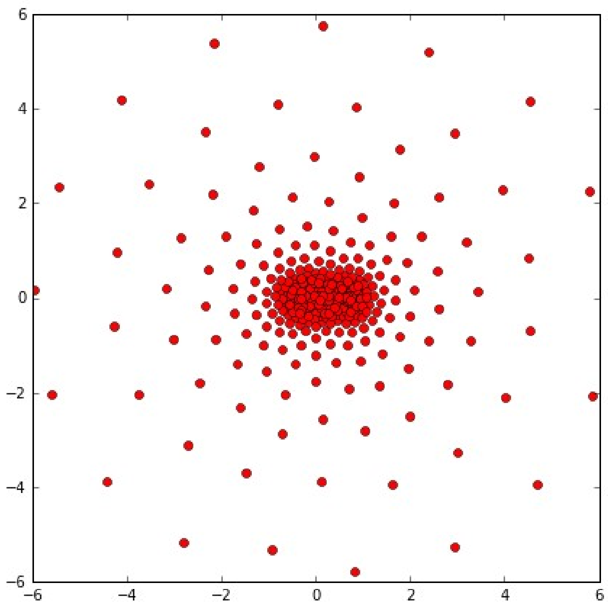
Baseline Selection Method, n = 151
Residual = 0.15968
5.2676 seconds (online)
0.2207 seconds (offline)



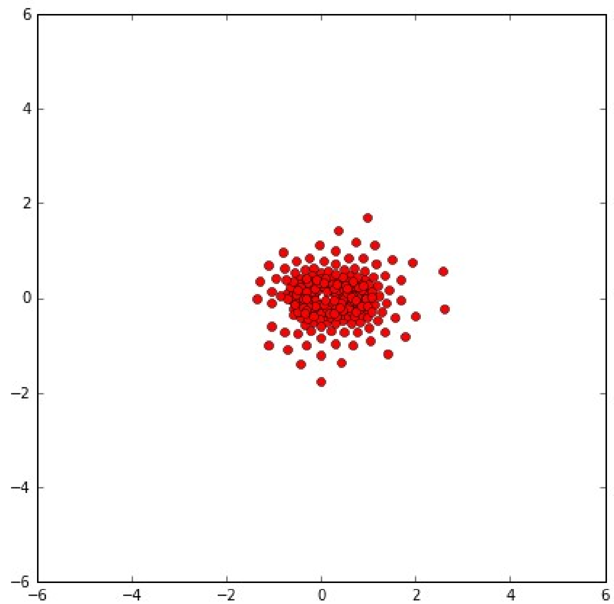
NEW Selection Method n ~ 151
Residual = 0.00562
5.1748 seconds (online)
1.9023 -> 0.5099 seconds (offline)

IMPROVED

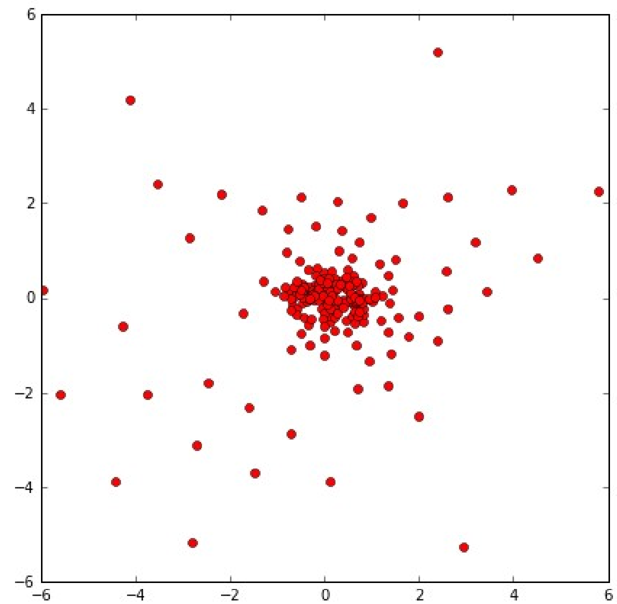
Point sampling extension to 2D



Original Points, n = 513



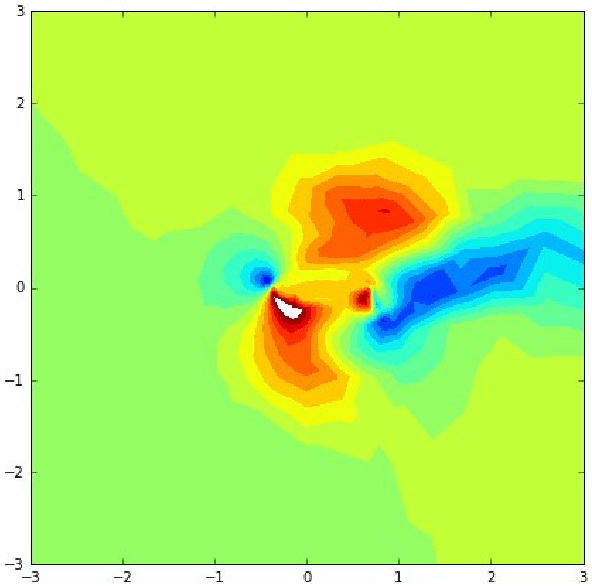
Baseline Selection Method, n = 273



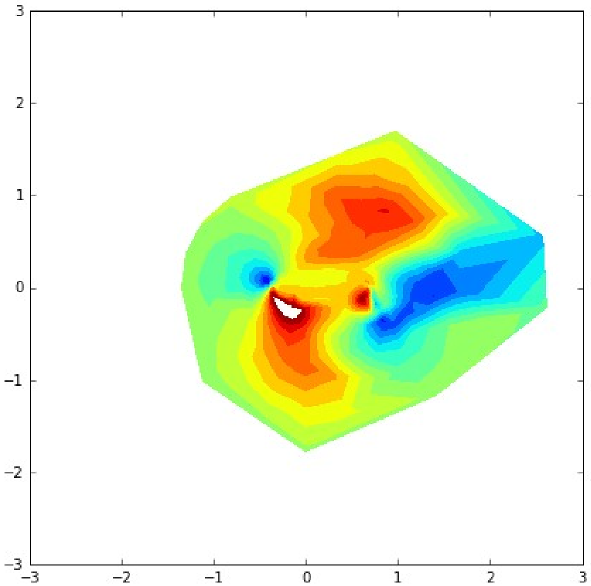
NEW Selection Method n = 272

Point sampling extension to 2D

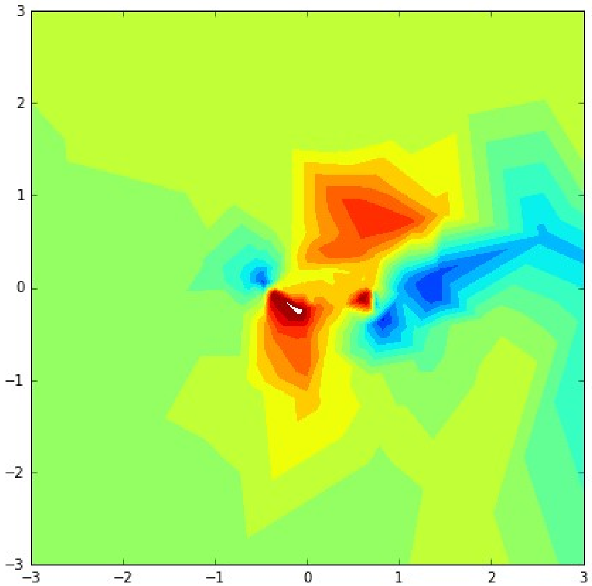
Contours for chosen points – projections only, not solutions



Original Points, n = 513

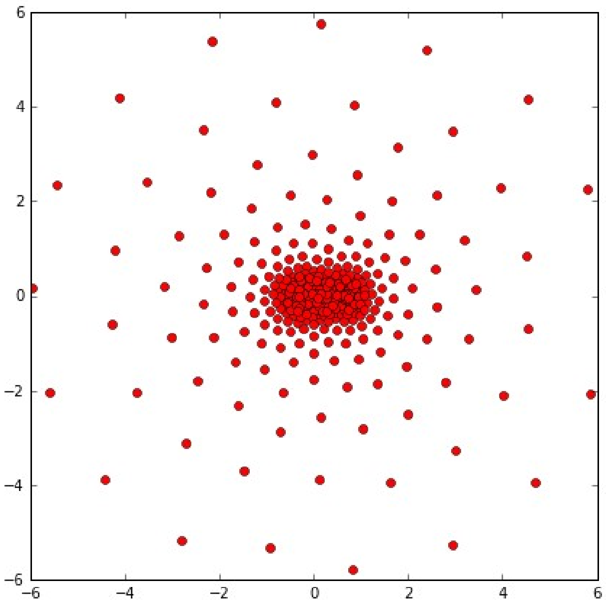


Baseline Selection Method, n = 273

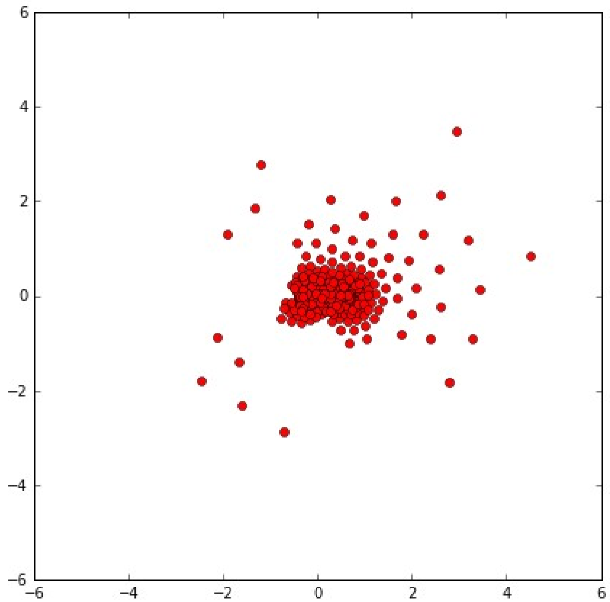


NEW Selection Method n = 272

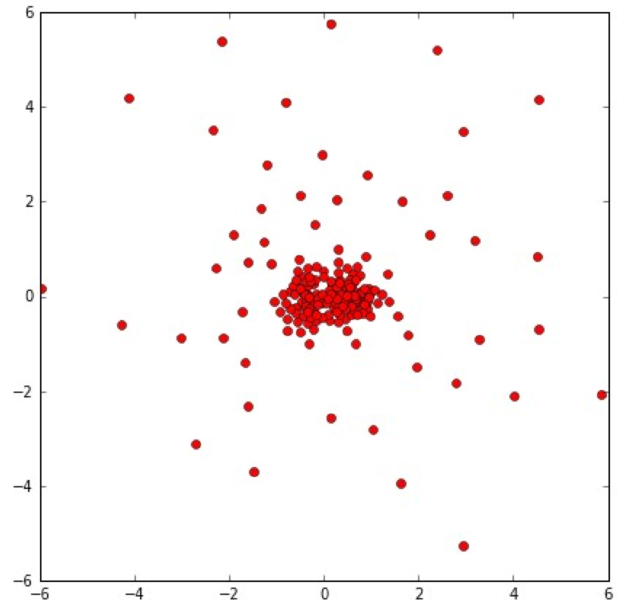
Point sampling extension to 2D



Original Points, n = 513



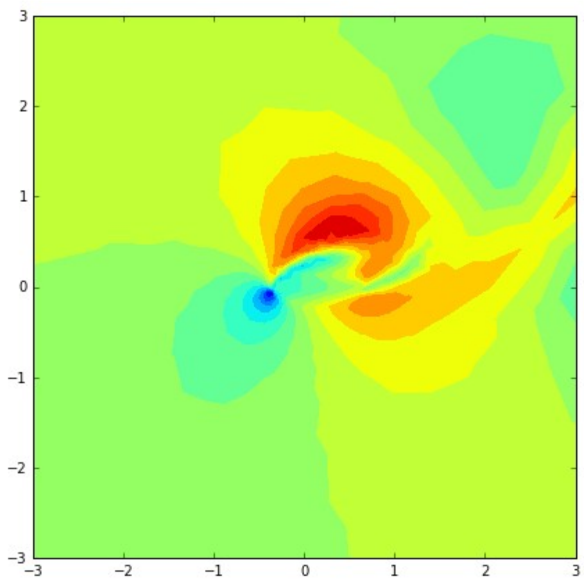
Baseline Selection Method, n = 273



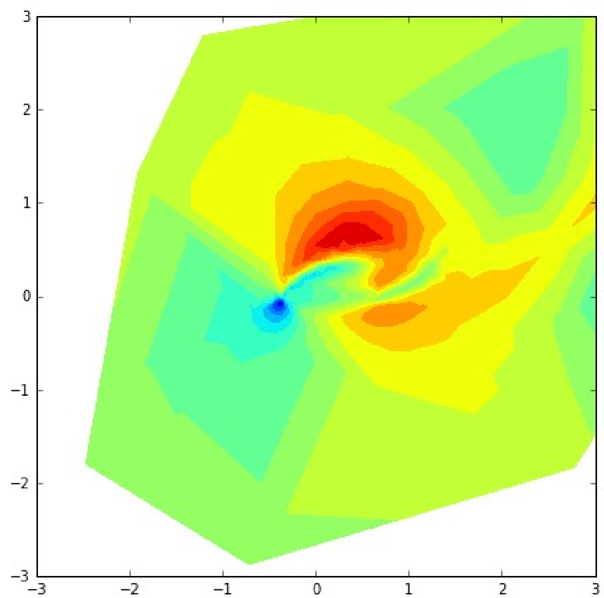
NEW Selection Method n = 272

Point sampling extension to 2D

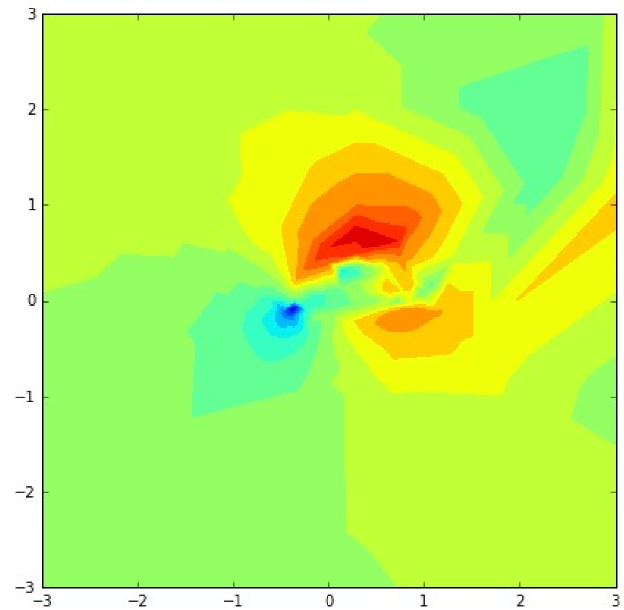
Contours for chosen points – projections only, not solutions



Original Points, n = 513



Baseline Selection Method, n = 273



NEW Selection Method n = 272