PDE Model Reduction Using SVD

https://people.sc.fsu.edu/~jburkardt/presentations/...
svd_2006_fsu.pdf

.....

John Burkardt[1] Max Gunzburger[2] Hyung-Chun Lee[3]

[1]School of Computational Science
Florida State University

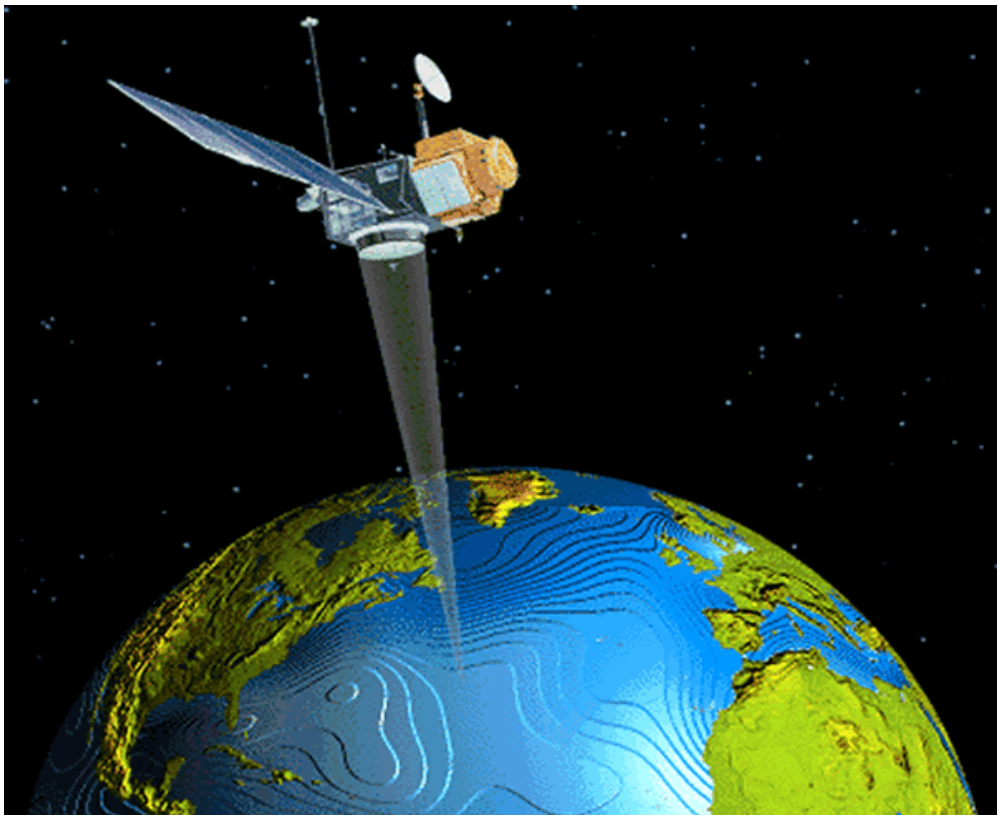
[2]Department of Mathematics and School of Computational Science
Florida State University

[3]Mathematics Department
Ajou University, Korea

SCS/FSU, 21 September 2006

Data Flood: Satellites

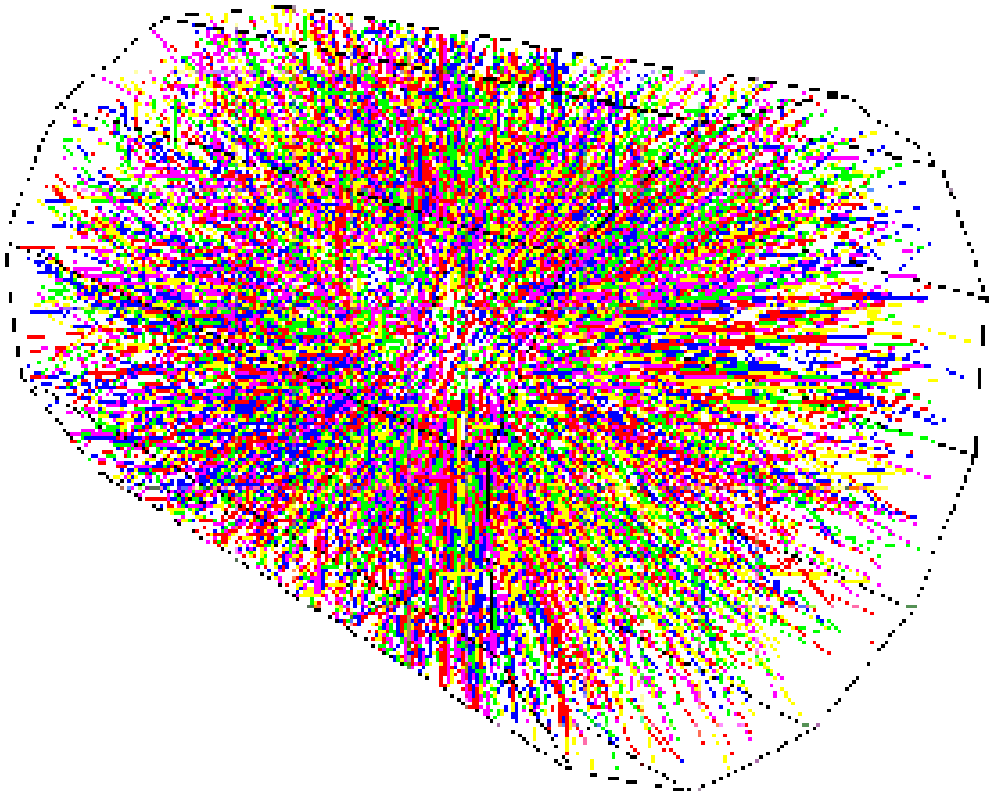
Satellites generate so much data that most is not even sent!



When it gets to earth, most is not examined!

Data Flood: Particle Accelerators

Physicists run particle accelerators that generate millions of images:



Computer programs are needed just to discard most of the data.

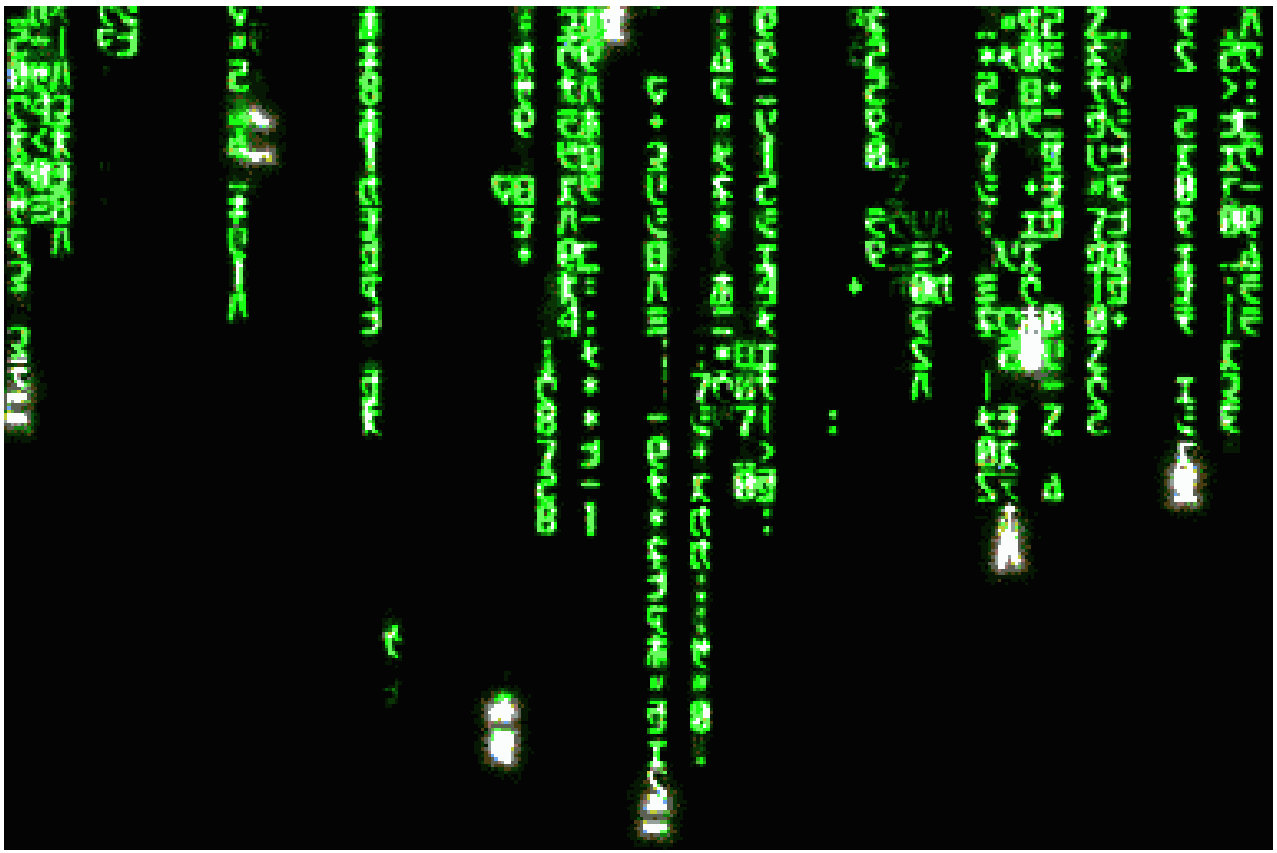
Data Flood: Fingerprints

Here is a fingerprint found after a kidnapping:

Even if a match is on file, how do you find it?

Data Flood: Where's the Information?

We've got all this data, why aren't we smarter?



Data is not information!

SVD: The Singular Value Decomposition

We introduce the **singular value decomposition**:.

The SVD of an M by N (real) matrix A:

$$A = U \cdot \Sigma \cdot V'$$

- U is M by M orthogonal;
- Σ is an M by N (nonnegative) diagonal matrix.
- V is N by N orthogonal;

SVD: What the SVD Tells Us

For a system with patterns, the SVD decomposition tells us where the information is concentrated.

$$A = U \cdot \Sigma \cdot V'$$

- The leading columns of U are the "preferred behaviors";
- The diagonal of Σ is an energy or importance weight;
- The entries of Σ are positive, and sorted in decreasing order;
- For information with patterns, the entries of Σ are rapidly decreasing.

SVD: Can Tell Us About A Mapping

We think of A as mapping unit vectors x to an ellipsoid of result vectors Ax:

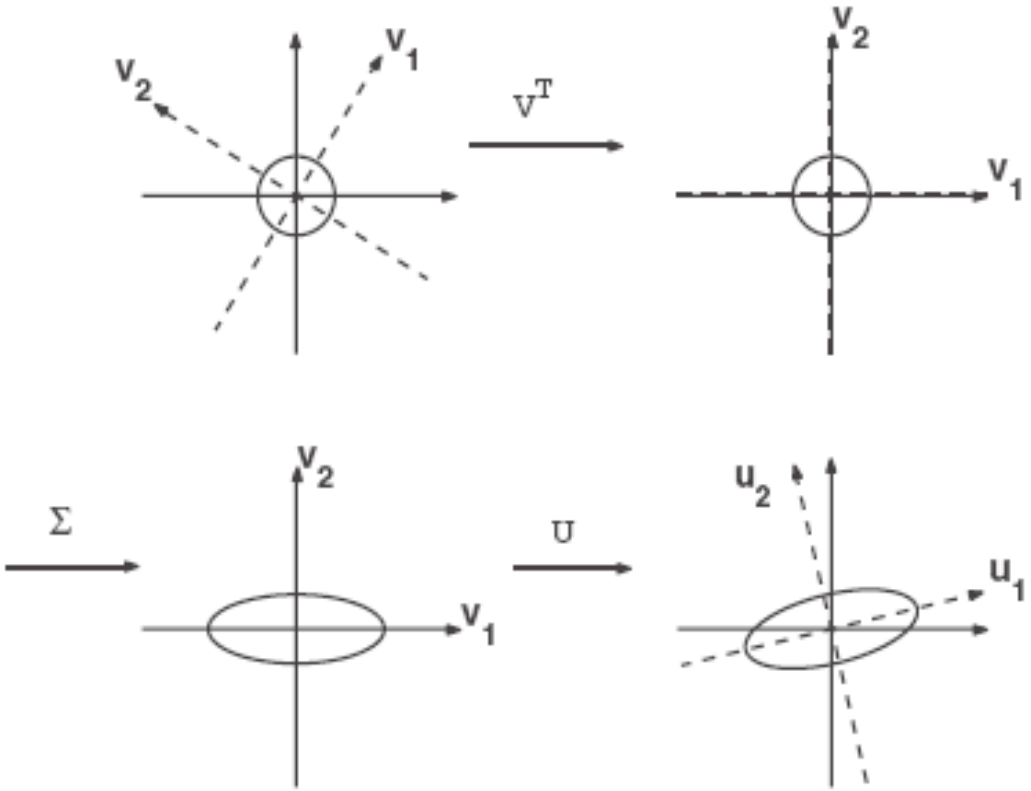


Image from Muller, Magaia, Herbst

SVD: Summing Rank-1 Matrices

The SVD also represents A as the sum of rank 1 matrices.

$$A = \sigma_1 u_1 \cdot v_1' + \sigma_2 u_2 \cdot v_2' ... + \sigma_r u_r \cdot v_r'.$$

This representation is done "optimally".

A truncated approximation is the best of that rank, in the Frobenius norm.

SVD: From Data, We Get Information

- The singular values tell us "how singular" a matrix is;
- The singular value ratios tell us how "balanced" the matrix is;
- The rightmost V vectors give a null space basis;
- The leftmost U vectors give the dominant outputs;
- Solve underdetermined or overdetermined systems;
- Can find an orthonormal basis for vector set (Gram-Schmidt);
- We may be able to approximate a matrix by very low rank;

SVD: Software Sources

- **EISPACK**, an "ancient" eigenvalue package, "SVD()"
- **LINPACK**, an "ancient" linear system package, "SSVDC()"
- **LAPACK**, a modern linear algebra package, "SGESVD()"
- **SCALAPACK**, parallel LAPACK, "PSGESVD()"
- **ESSL**, IBM linear algebra package, "SGESVF()"
- **ACM TOMS**, Algorithm #581, "GRSVD()"
- **MAPLE**, function "SingularValues()"
- **MATLAB**, function "svd()"
- **MATHEMATICA**, function "SingularValues[]"
- **Numerical Recipes**. "SVDCMP()"

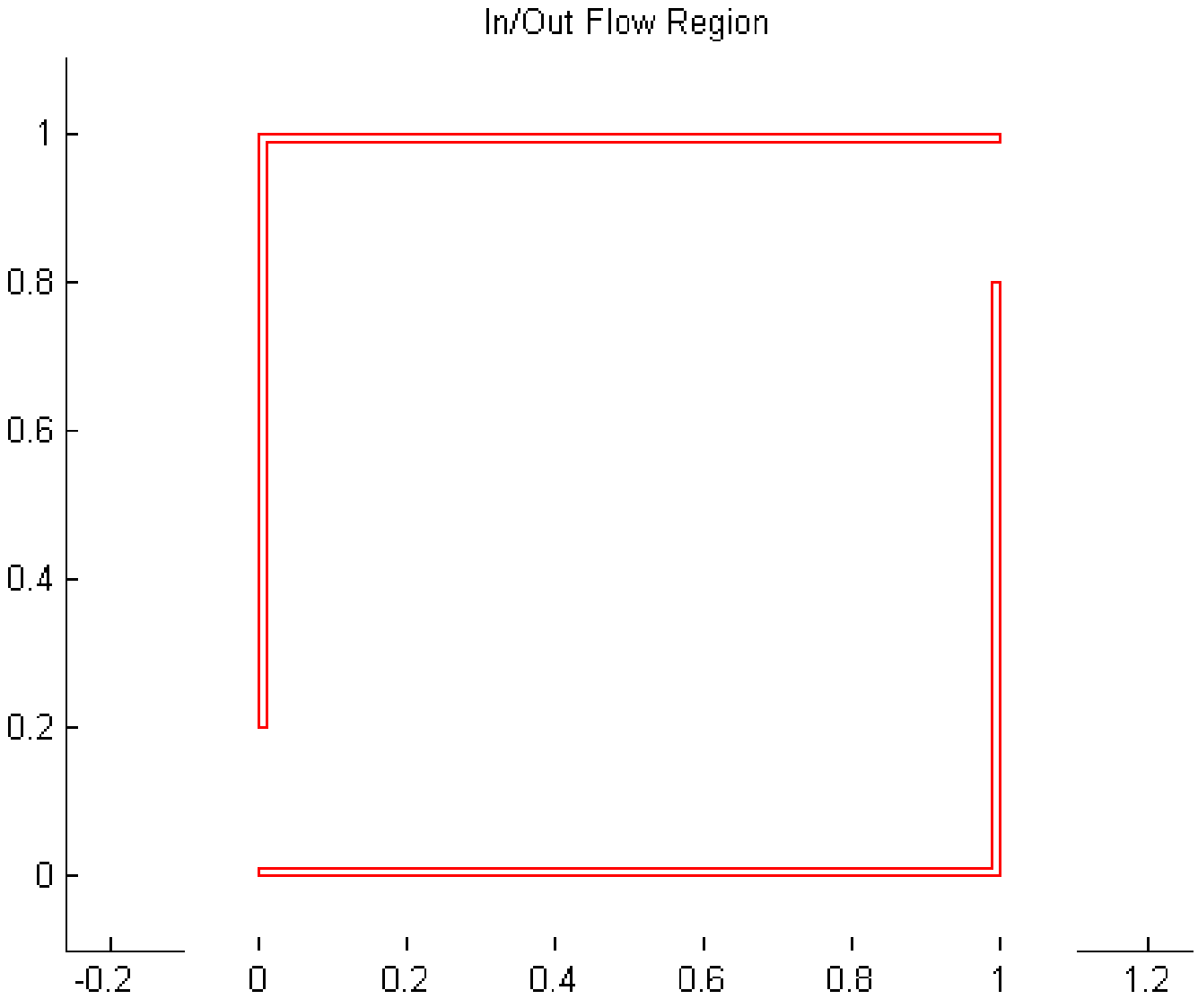
The IN/OUT Flow, Governing Equations

Incompressible viscous Navier-Stokes equations:

- $u(x, y)$ velocity vector;
- $p(x, y)$ pressure;
- ρ, known constant density;
- μ, known viscosity.

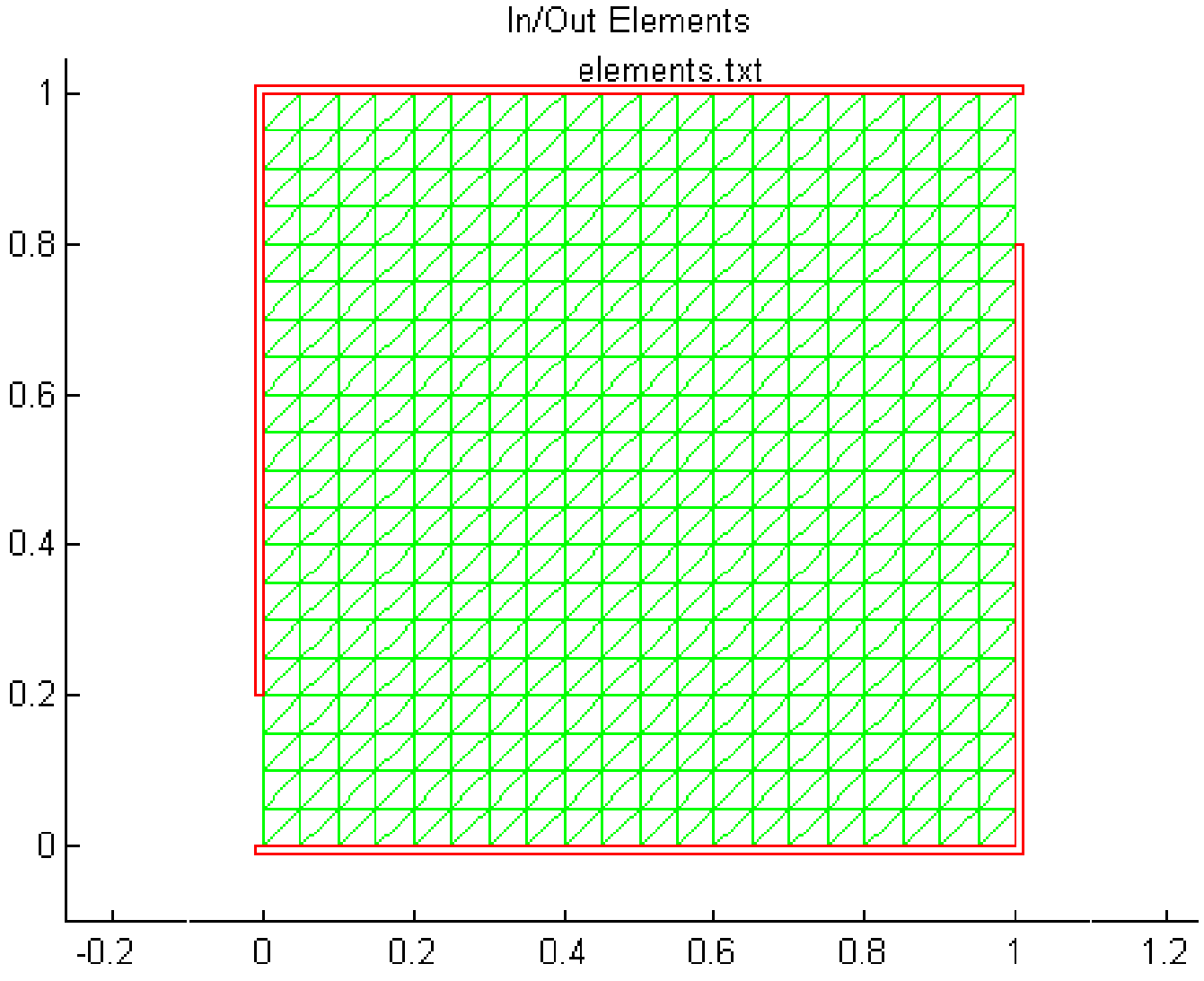
$$\begin{aligned} \rho \mathrm{u}_t - \mu \Delta \mathrm{u} + \rho(\mathrm{u} \cdot \nabla)\mathrm{u} + \nabla p &= 0 \\ \rho \nabla \cdot \mathrm{u} &= 0 \end{aligned}$$

The IN/OUT Flow: region

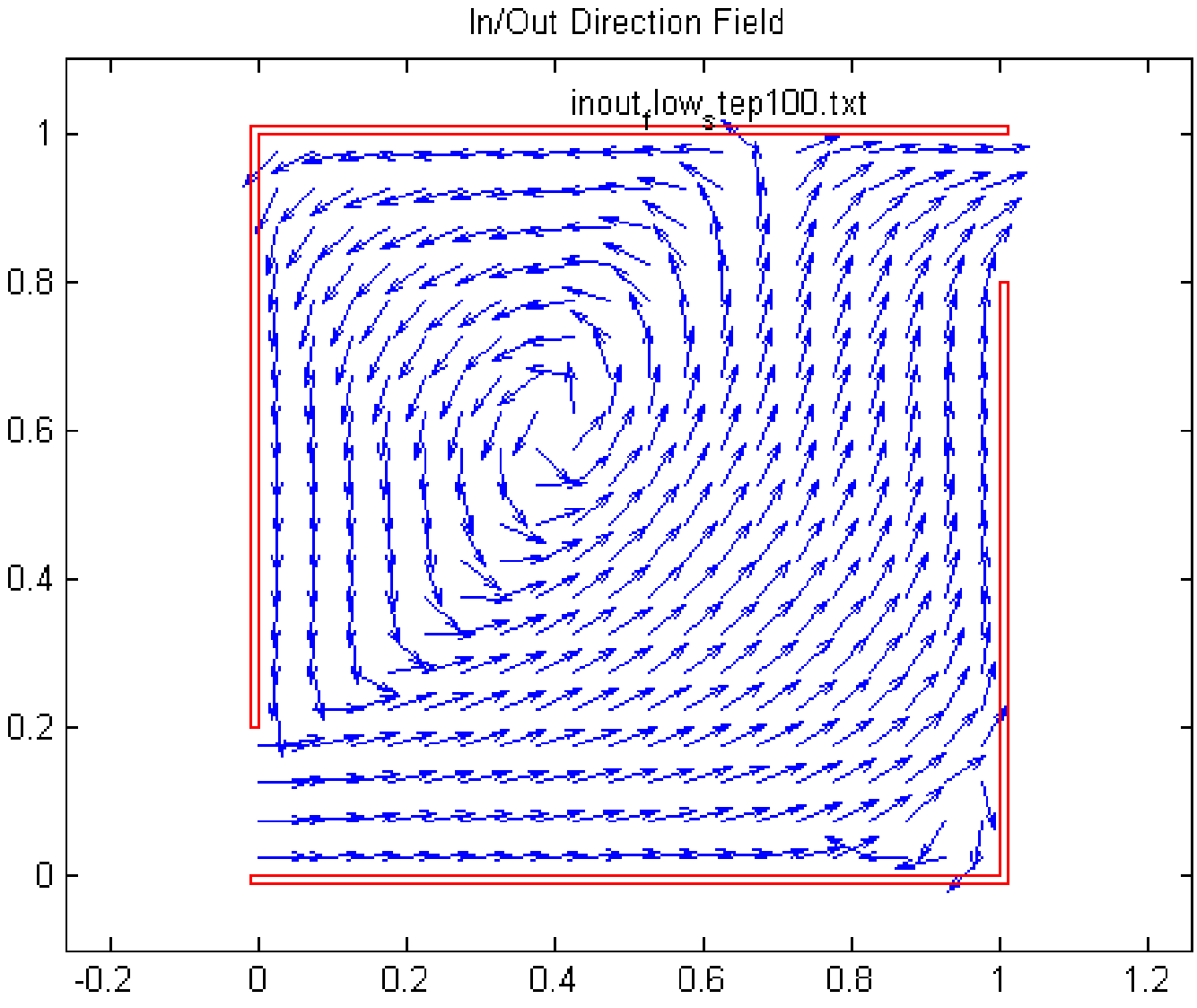


Finite Elements: The Elements

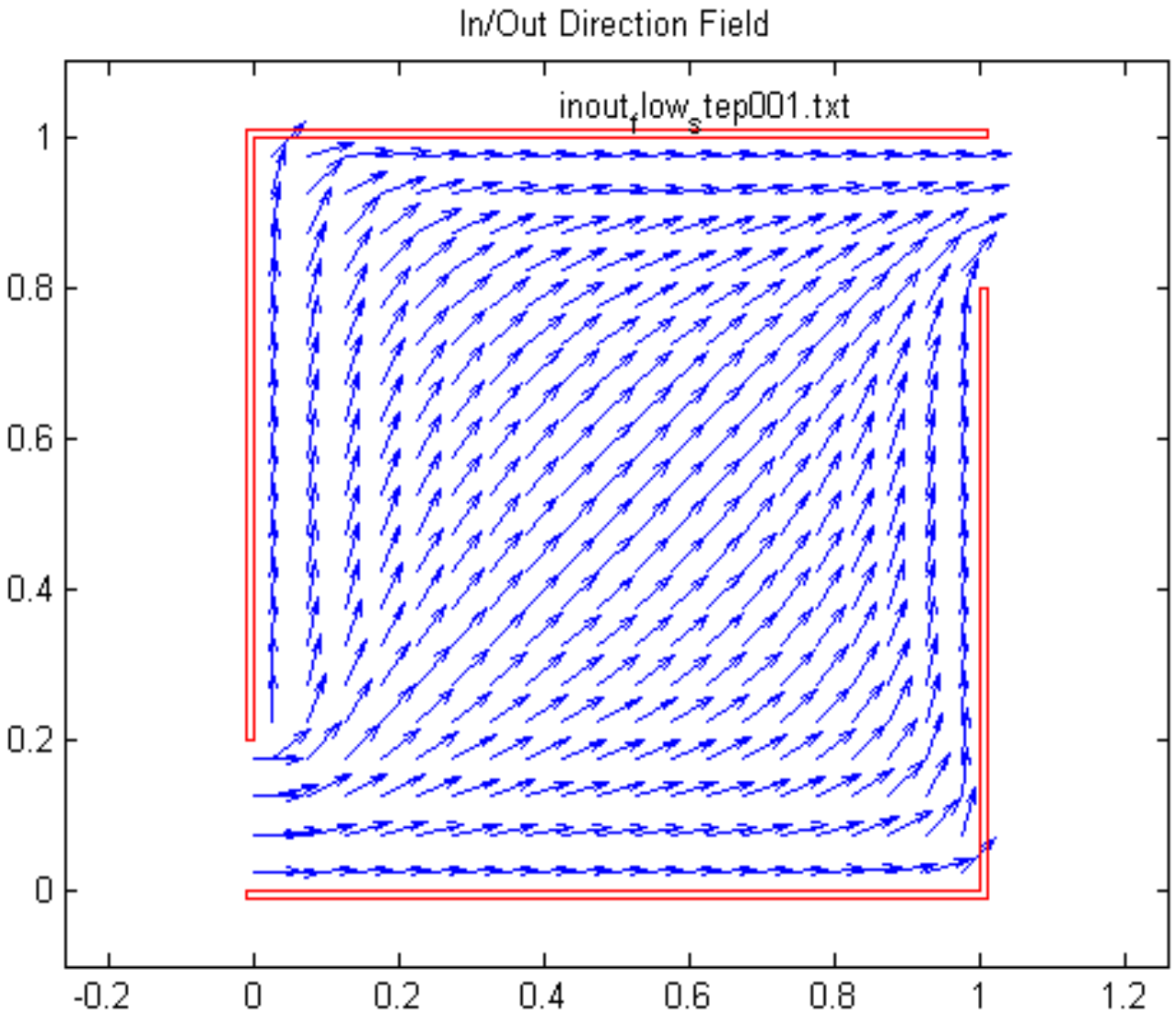
We create a mesh of "elements".



Example: Fluid Flow with Varying Input



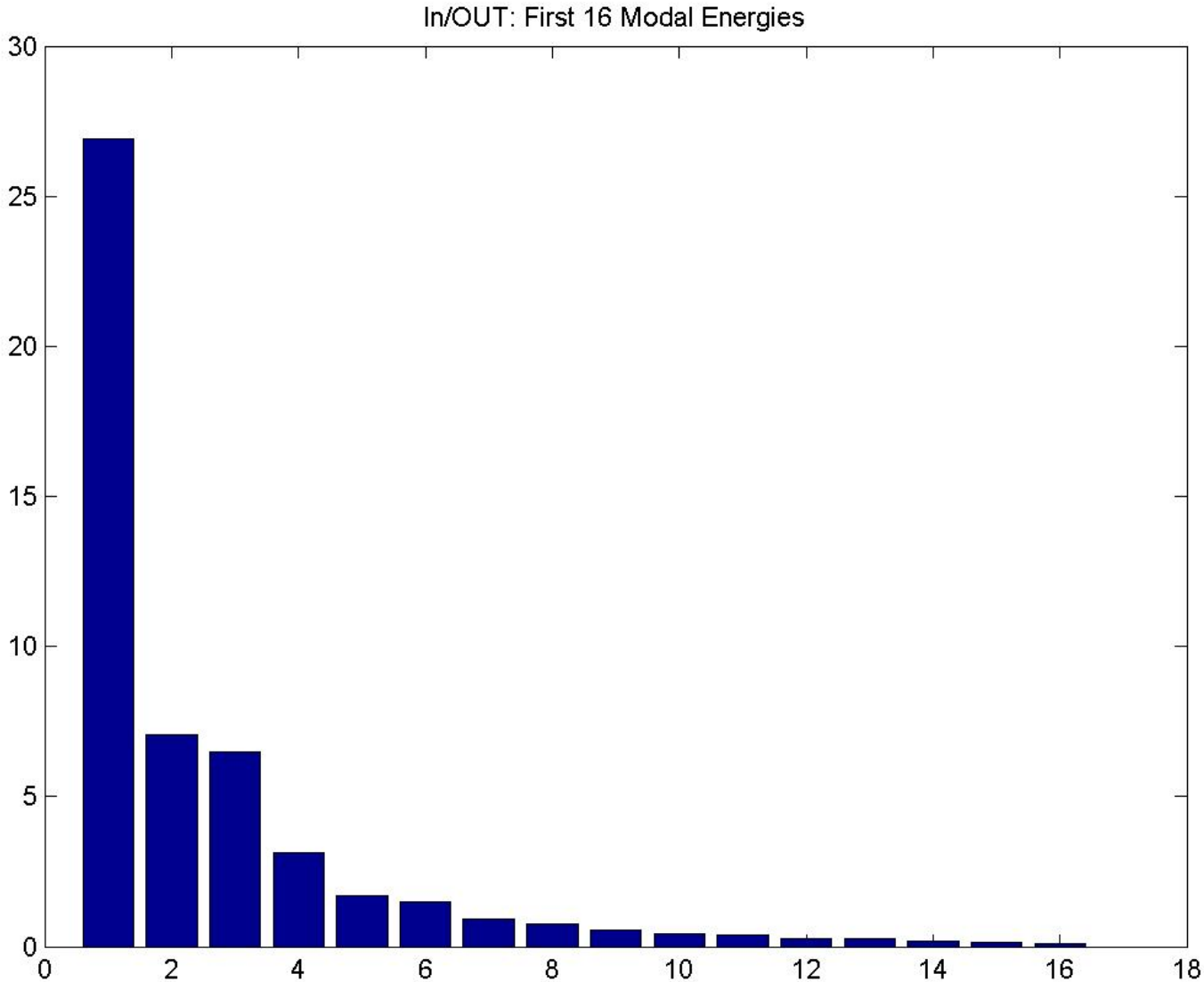
The IN/OUT Flow: Animation



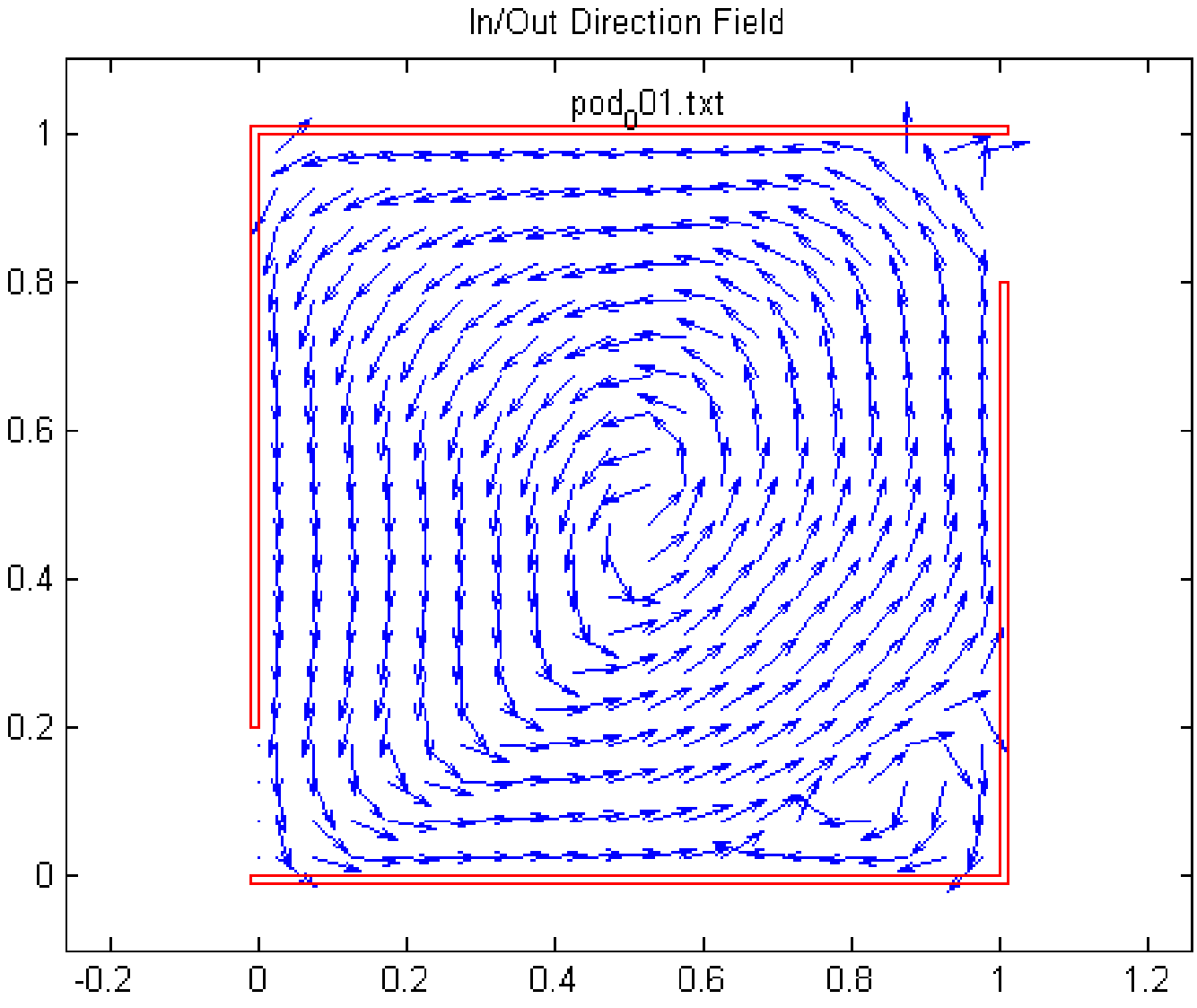
SVD Model Reduction for IN/OUT: Vector "Importance"

Index	Value	Relative	Cumulative
1	26.9107	0.527	0.527
2	7.0878	0.138	0.666
3	6.5015	0.127	0.794
4	3.1420	0.061	0.855
5	1.6973	0.033	0.889
6	1.4947	0.029	0.918
7	0.9253	0.018	0.936
8	0.7592	0.014	0.951
9	0.5738	0.011	0.962
10	0.4570	0.008	0.971
...	...	...	...
16	0.0994	0.001	0.997

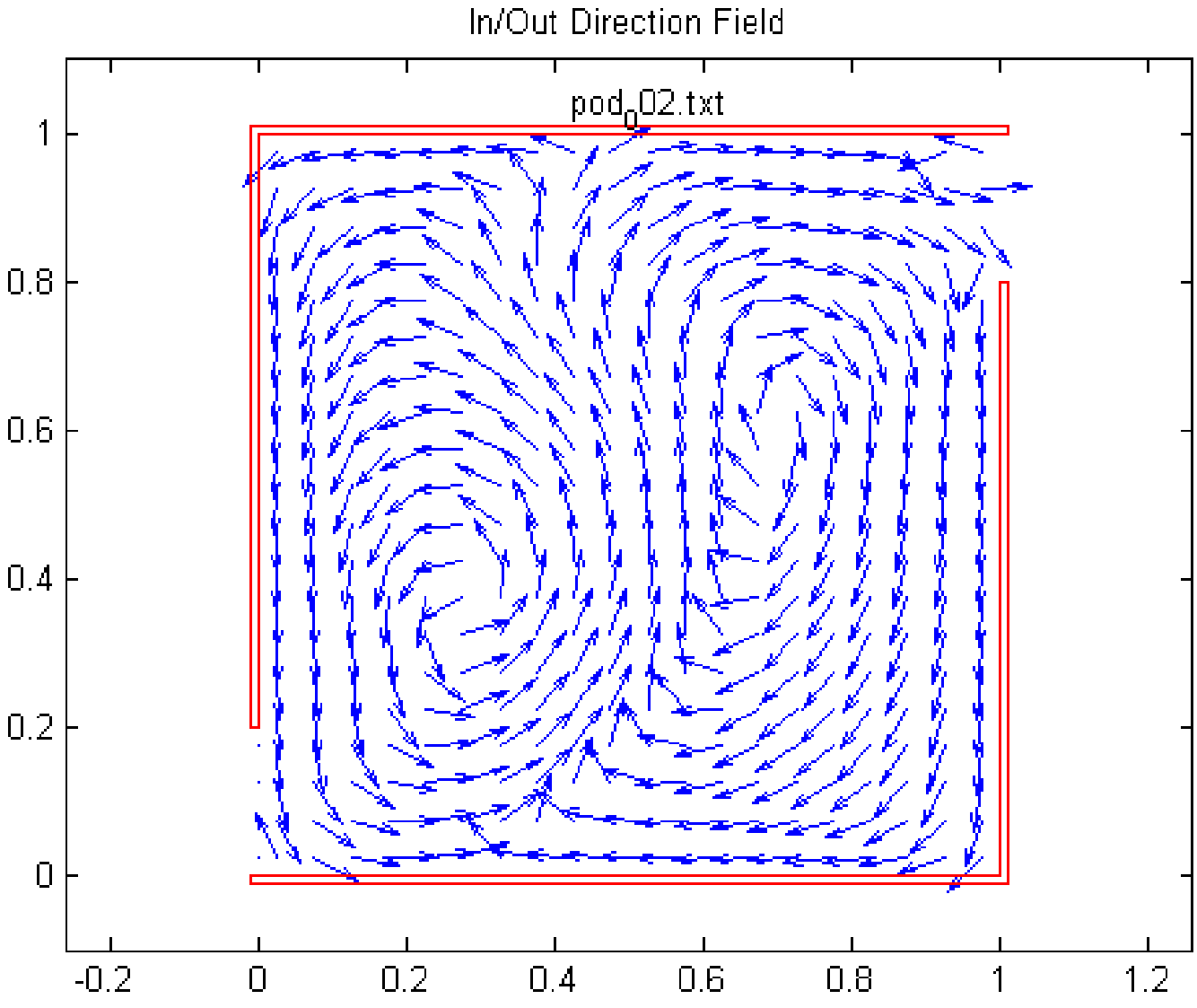
SVD Model Reduction for IN/OUT: Vector "Importance"



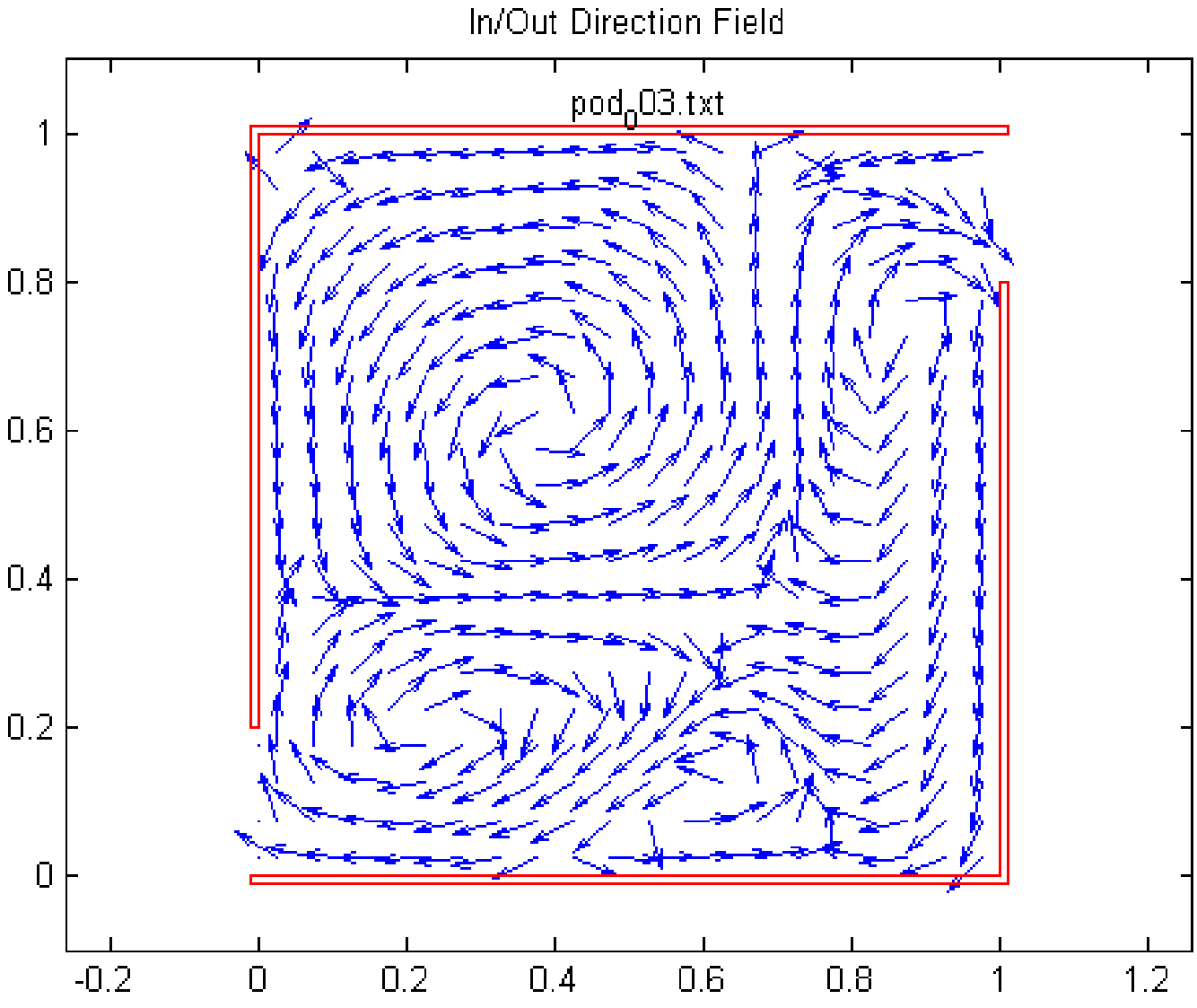
SVD Model Reduction for IN/OUT: Vector 1



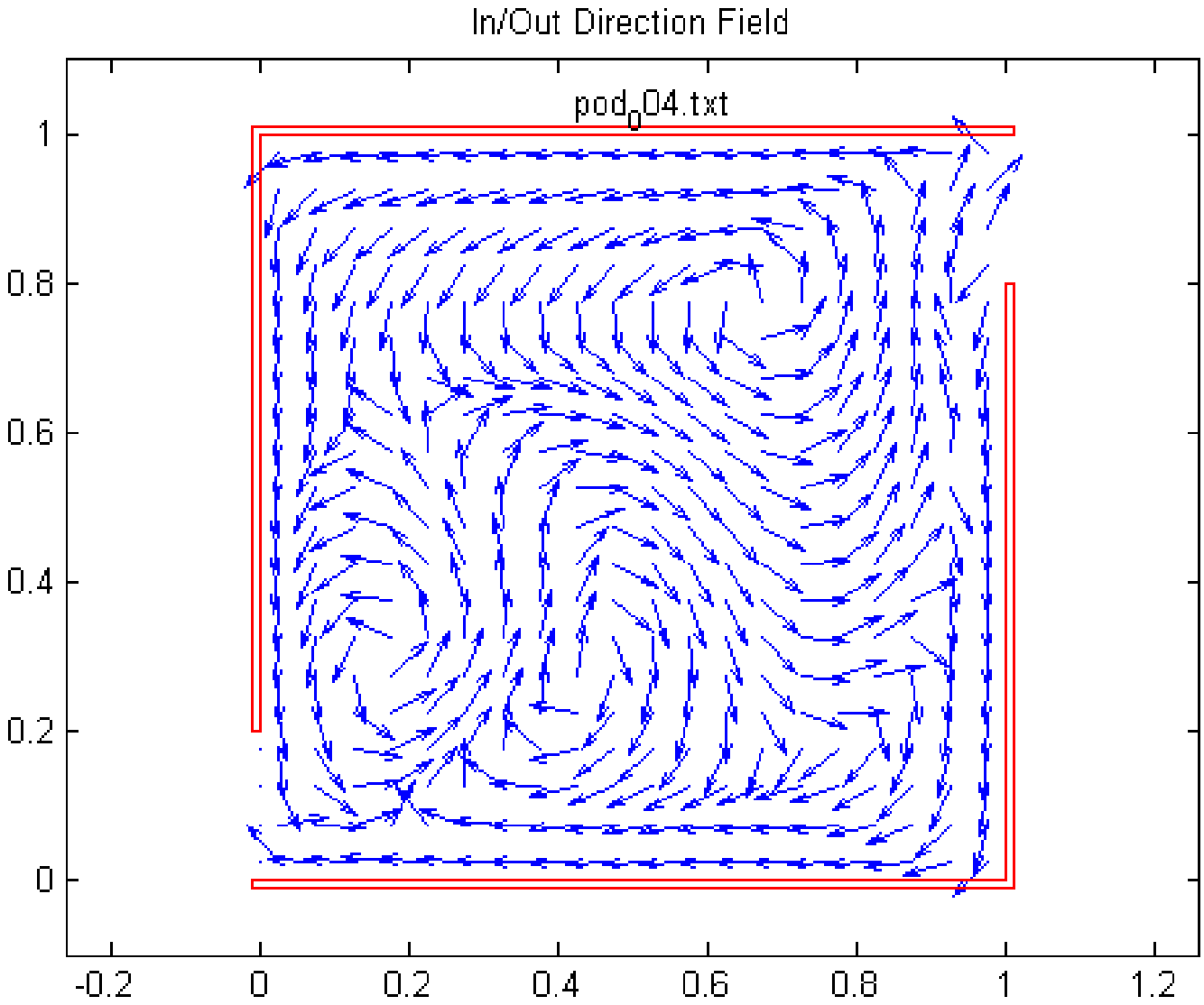
SVD Model Reduction for IN/OUT: Vector 2



SVD Model Reduction for IN/OUT: Vector 3



SVD Model Reduction for IN/OUT: Vector 4



Finite Elements: Function approximation

FEM approximates velocity and pressure using basis functions generated by mesh:

$$u(x,y) \approx U(x,y) = \sum_{i=1}^{NV} a_i \quad \phi_i(x,y)$$

$$p(x,y) \approx P(x,y) = \sum_{j=1}^{NP} b_j \quad \psi_i(x,y)$$

ROM: Reduced Order Modeling

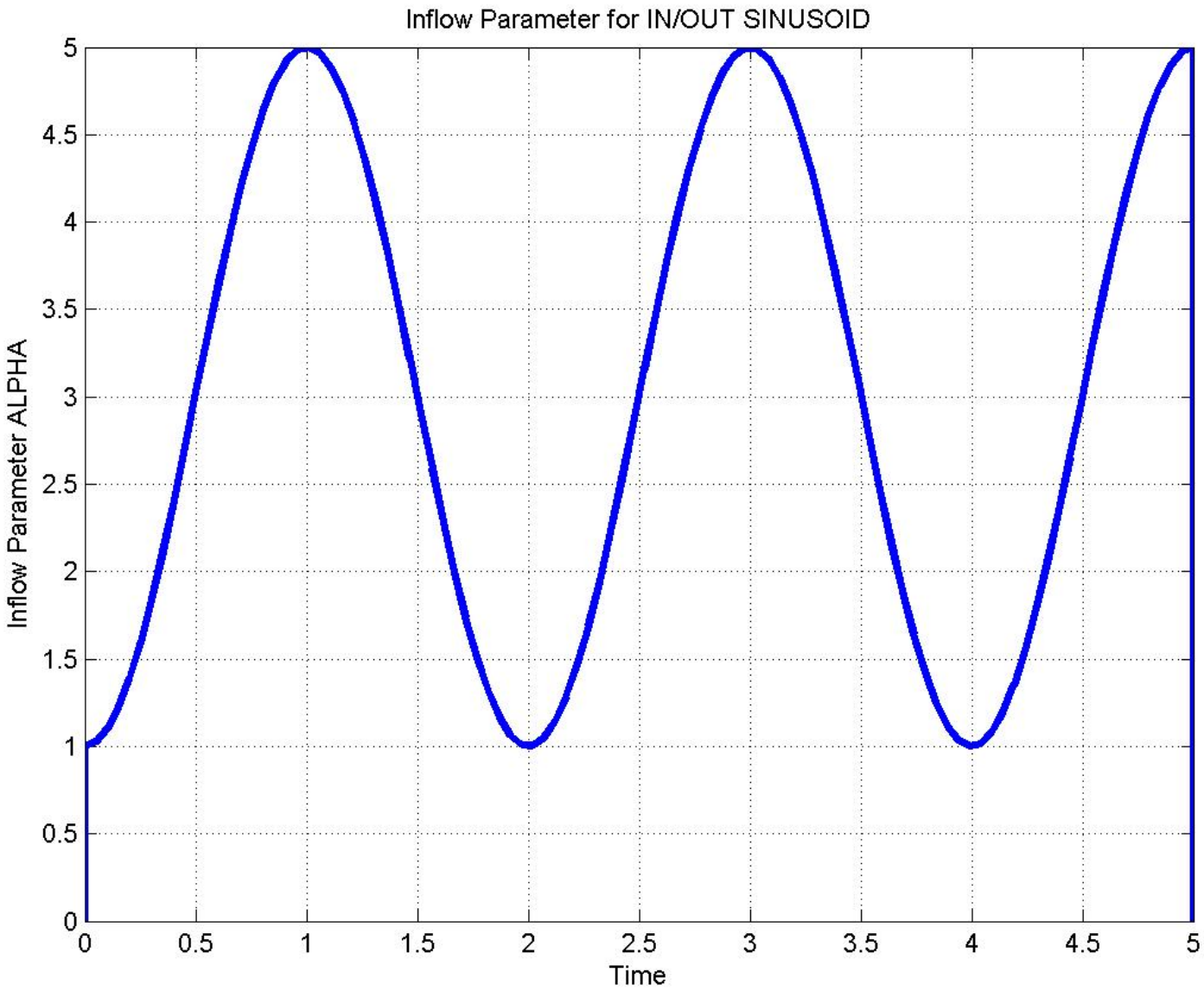
SVD applied to the data extracts an small orthogonal basis of "important" vectors.

Using basis functions from the finite element mesh, we had to define 3,362 basis functions, and then solve for the corresponding coefficients.

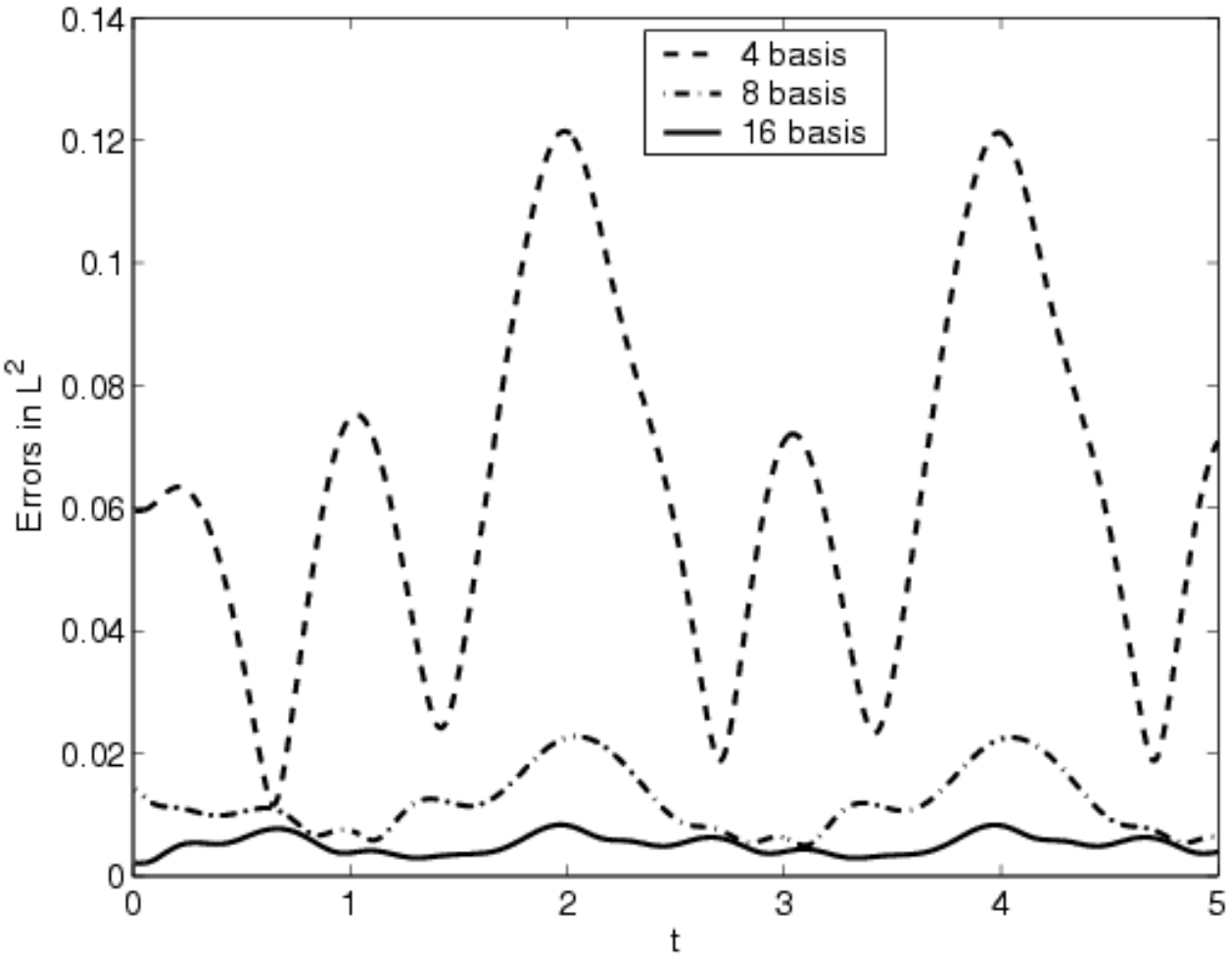
The SVD tells us that every velocity vector we've seen so far could be well approximated by our 16 SVD basis vectors.

Why not use them as a new finite element basis?

ROM: Solving New IN/OUT Problems



ROM: Solving New IN/OUT Problems



L2 difference between ROM and full FEM solutions.

SVD and Data Analysis

Navier Stokes equations still hide lots of information.

SVD extracts useful data (singular vectors).

SVD also gives "importance" of each vector.

SVD did not need to "know" the Navier Stokes equations.

We can apply SVD to data with no known governing equations.

FACES: (Muller, Magaia, Herbst)

Image from Muller, Magaia, Herbst

FACES: Creating Data

200 people were each photographed in 3 poses for N=600 images.

If each image used 400 * 800 pixels, this makes about 300,000 pixels, each with R, G and B values, for about M=1,000,000 numeric values.

FACES: SVD Analaysis (Singular Values)

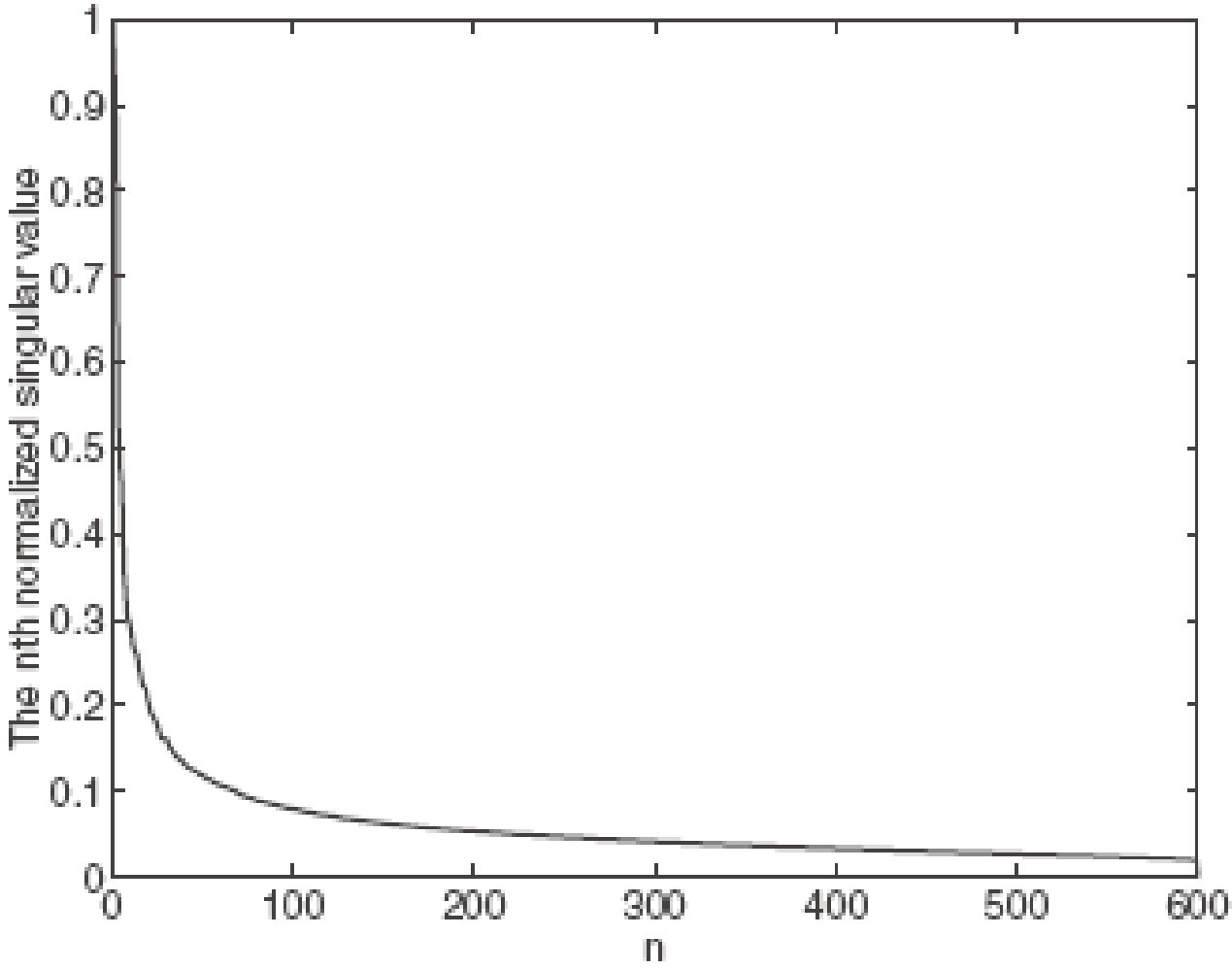
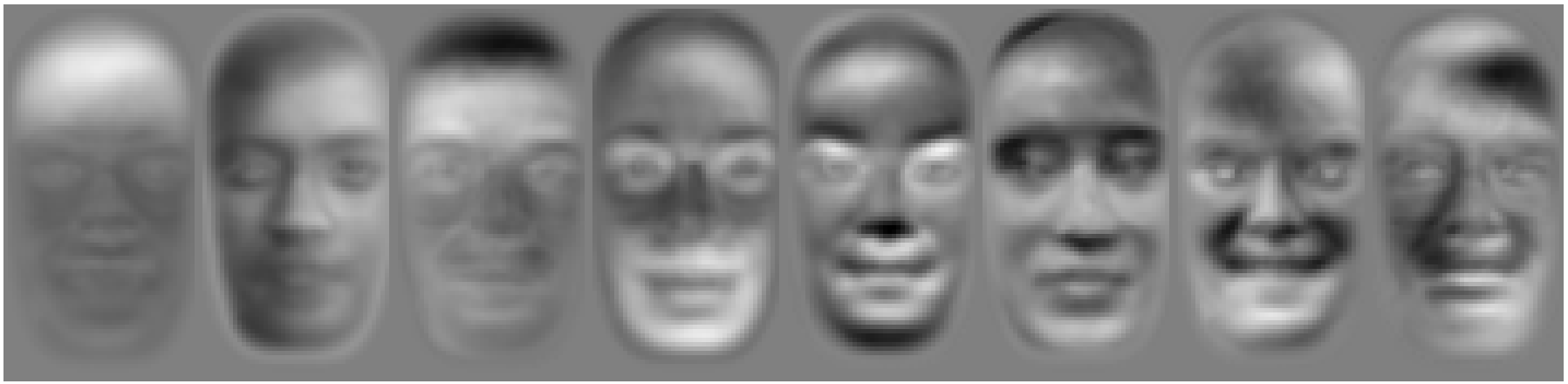


Image from Muller, Magaia, Herbst

FACES: SVD Analysis (Singular Vectors)



Using similar SVD methods on the face data, here are first 8 modes or "eigenfaces".

Usually 100 or 150 "eigenfaces" are enough to store almost all the facial information.

Image from Muller, Magaia, Herbst

FACES: "Recognizing" a New Face

Reconstruction using 40, 100, 450 eigenfaces.

Image from Muller, Magaia, Herbst

FACES: The Covariance Matrix

The SVD includes information that can be used to recognize or reject new data. If we "gently" preprocess the data, we get usable covariance information.

- Subtract average from all data;
- Scale, dividing by $\sqrt{N}$;
- U vectors are maximum variation directions
- Σ contains standard deviations

FACES: The Covariance Matrix

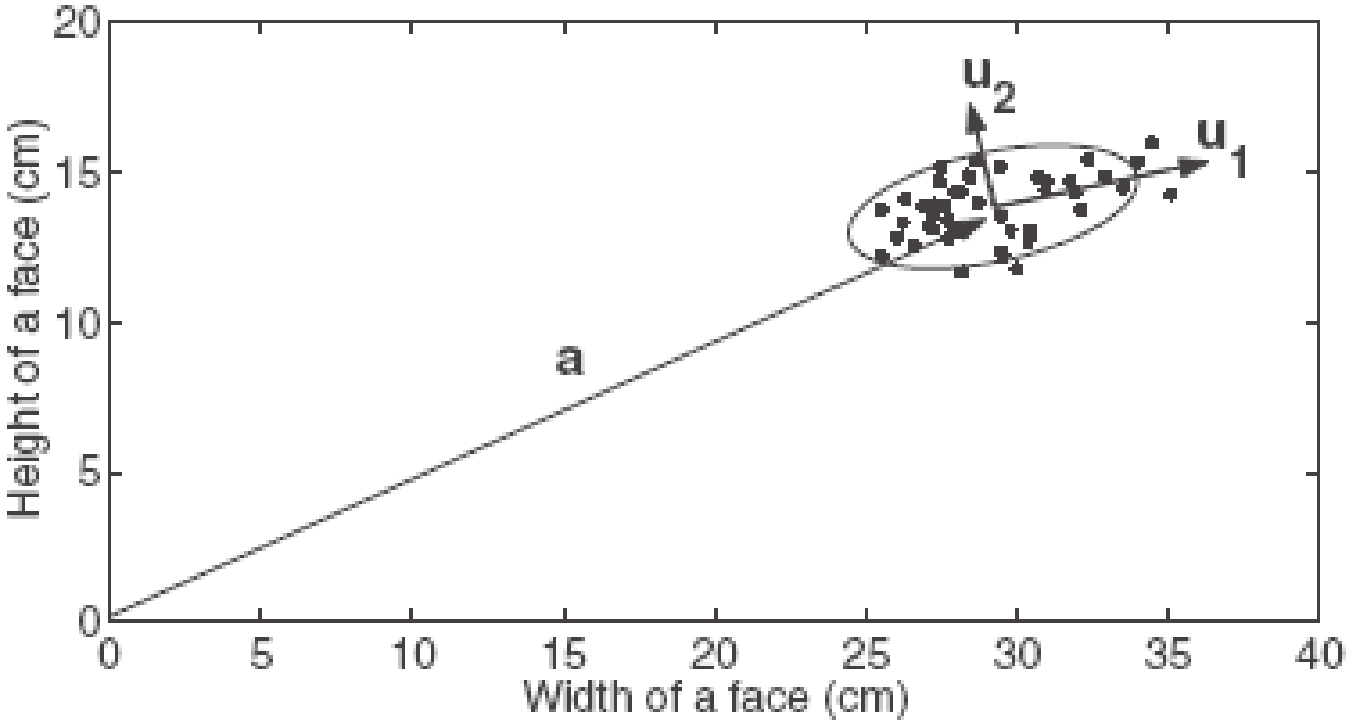


Image from Muller, Magaia, Herbst

FACES: The Covariance Matrix

Suppose we have a new item. Is it a face?

Subtract the average, and project it onto the U vectors.

From the SVD information, we can compute the probability that an item of data would fall this far from the average.

Summary of SVD

- SVD is a natural analysis of data.
- knowledge of governing equations (if any!) not necessary
- unknown information and patterns discovered.
- compact representation of huge dataset;
- a natural (secondary) basis for FEM calculations.
- projection can be used to "recognize" new data.

SVD is one way to turn **data** *into* **information**.

References

Burkardt, Gunzburger, Lee, *Centroidal Voronoi Tessellation-Based Reduced-Order Modeling of Complex Systems*, SIAM Journal on Scientific Computing, Volume 28, Number 2, 2006.

Muller, Magaia, Herbst, *Singular Value Decomposition, Eigenfaces, and 3D Reconstructions*, SIAM Review, Volume 46, Number 3, 2004.

Trefethen, Bau, *Numerical Linear Algebra*, SIAM, 1997.