Order Reduction of Large Scale DAE Models

J.D. Hedengren and T. F. Edgar
Department of Chemical Engineering
The University of Texas at Austin

Outline

- Motivation
- Two Step Process for DAE Model Reduction
  1. Reduction of differential equations
  2. Reduction of algebraic equations with ISAT
- Examples

DAE model size

- Small (1-100 variables)
  - Single process units (e.g., reactor models)
  - Real-time NMPC calculations are feasible
- Medium (100-10,000 variables)
  - Multiple process units
  - Multicomponent modeling, reaction networks
  - Real-time NMPC applications very difficult
- Large (10,000+ variables)
  - Plant wide dynamic models
  - Currently optimized at this level only with steady state models (RTO)

Motivation

- Plantwide NMPC control (Large scale DAE systems)
- Storage and retrieval of optimal control trajectories (small and medium scale)
- Reveal underlying structure of the model
  - Determine dynamic degrees of freedom
  - Find source of DAE initialization / convergence problems
- Automate model reduction

Adaptive DAE Model Reduction

1. <u>Reduction of differential equations</u>
2. Reduction of algebraic equations

ODE Model Reduction

- Optimally reduce the number of model variables
- Linear combination of states that retain the most important dynamics
- Methods
  - Proper Orthogonal Decomposition (POD)
  - Balanced Covariance Matrices (BCM)

Predicting DDOF

- Can model reduction be made adaptive?
- Possible error control strategies:
  - Singular values (poor predictor)
  - Solve non-reduced model at check points (inefficient)
  - Equation residuals (new approach)

Model Reduction Error

- Variable error constraint

$$\left| x_{ROM}(t) - x(t) \right| \le \varepsilon_{tol}$$

- Controlling variable error
  - ↓ model order, ↑ variable error
  - ↑ model order, ↓ variable error
- DOF
  - Total degrees of freedom (DOF) = model order
  - *Dynamic* degrees of freedom (DDOF) = reduced model order that satisfies the variable error constraint

Predicting DDOF

- Linearized system

$$\dot{x}(t) = Ax(t) + Bu(t)$$

- Galerkin projection

$$x(t) = \widetilde{P}^T \overline{x}(t) + r(t)$$

$$\dot{x}(t) = \widetilde{P}^T \dot{\overline{x}}(t) + \dot{r}(t)$$

- Substitute

$$\widetilde{P}^T \dot{\overline{x}}(t) = A\left(\widetilde{P}^T \overline{x}(t)\right) + Ar(t) - \dot{r}(t) + Bu(t)$$

- Predictor (*r(t)* = variable error, *R(t)* = equation residual)

$$R(t) = Ar(t) - \dot{r}(t) \cong Ar(t)$$

$$\boxed{r(t) \cong A^{-1}R(t) \quad \text{Variable error predictor (linearized)}}$$

Adaptive ODE Model Reduction

- Variable error constraint

$$\left| x_{ROM}(t) - x(t) \right| = \boxed{r(t) \cong A^{-1} R(t)} \le \varepsilon_{tol}$$

- Open equation format

$f(\dot{x}(t), x(t)) = 0$ Solution obtained by finding roots

$f(\widetilde{P}^T \dot{\overline{x}}(t), \widetilde{P}^T \overline{x}(t)) = R(t)$ Solution obtained by minimizing residuals

- Controlling variable error (iterative approach)
  - When $A^{-1}R(t) \le \varepsilon_{tol}$, ↓ model order
  - When $A^{-1}R(t) > \varepsilon_{tol}$, ↑ model order
- Variable error predictor can also be used to improve reduced model accuracy

Example: Adaptive ODE Reduction

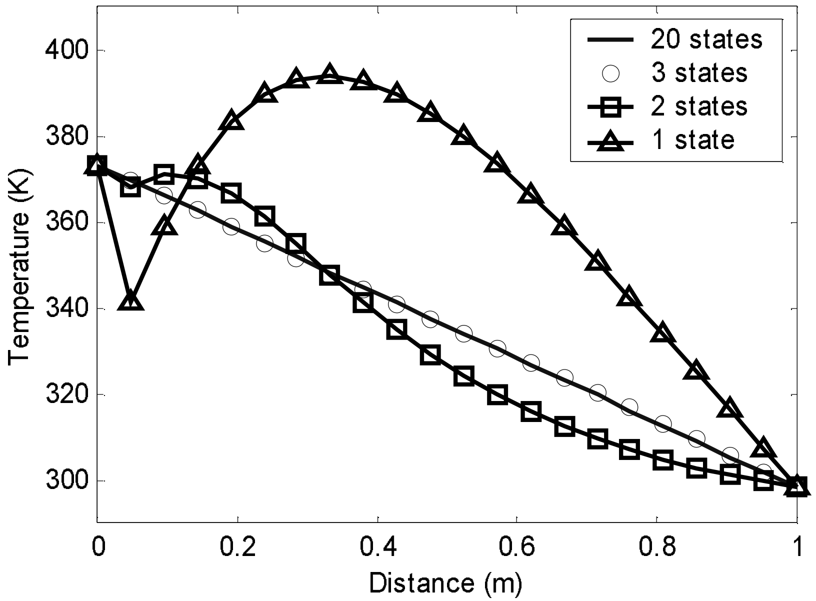
- 1-D unsteady heat conduction

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right)$$

- Discretized the PDE to give a set of 20 ODEs
- Simulation:
  - Aluminum slab with thickness 1 m
  - Initially at 25 ºC
  - At t = 0 the left boundary is changed to 100 ºC
  - Tolerance set ε_{tol} = 1 ºC

Example Results

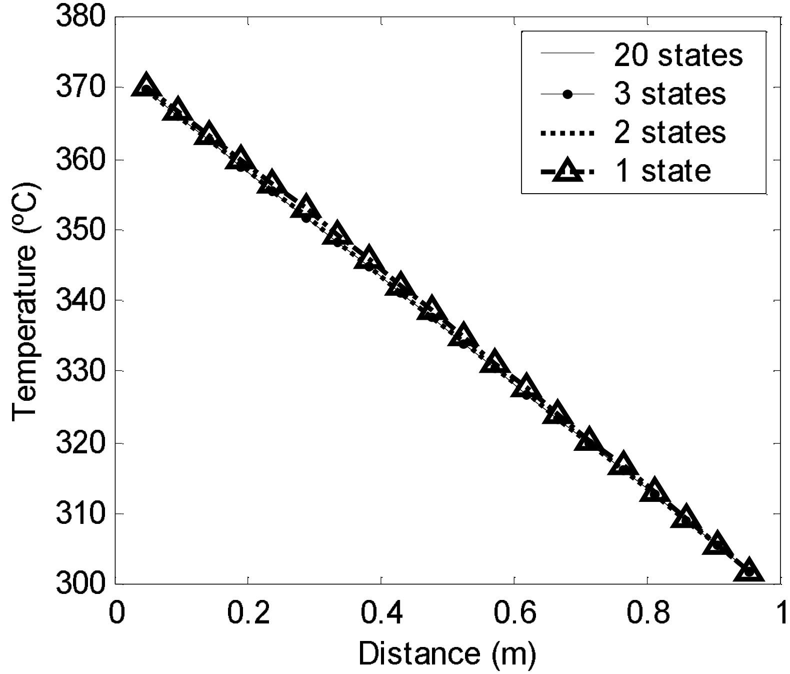
- After 100 minutes the temperature profile approaches steady state



- Variable error predictor indicates that at least 3 states are required to meet error tolerance of 1 ºC

Example Results

- Variable error predictor can also be used to improve the reduced model accuracy (1 state required with correction)



- Excellent prediction because the model is nearly linear and approaching steady state

Adaptive DAE Model Reduction

1. Reduction of differential equations (model reduction)
2. Reduction of algebraic equations with ISAT

Partitioning and Precedence Ordering

- DAE model

$$f_{DAE}(\dot{z}, z, t) = 0 \quad \Longleftrightarrow \quad \begin{aligned} f_{ODE}(\dot{x}, x, y, t) &= 0 \\ f_{AE}(x, y, t) &= 0 \end{aligned}$$

x = ODE state; y = algebraic state

- Sparsity matrix

$$J_{ij} = \begin{cases} 1 & \text{if } y_j \text{ or } \dot{x}_j \text{ appears in } f_{DAE\,i} \\ 0 & \text{otherwise} \end{cases}$$

- Pairing equations and variables
  - Obtain a maximum transversal (largest diagonal via rearrangement)
  - Zero-free diagonal means that each variable is uniquely paired with an equation

Partitioning and Precedence Ordering

- Lower triangular block form
  - Each successive block of variables and equations can be solved independently
  - Inverting the sparsity matrix shows global variable dependencies
  - Binary distillation example (230 x 230 system):

<u>Original sparsity</u>

$$
\left[\begin{array}{cc|cccccccc}
X & 0 & X & 0 & 0 & X & 0 & 0 & 0 & 0 \\
0 & X & 0 & 0 & 0 & X & X & X & 0 & 0 \\
\hline
0 & 0 & X & 0 & 0 & 0 & 0 & 0 & X & 0 \\
0 & 0 & 0 & X & 0 & 0 & X & X & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & X \\
0 & 0 & 0 & X & 0 & X & 0 & 0 & 0 & 0 \\
0 & 0 & X & 0 & X & 0 & X & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & X & 0 & 0 & X & 0 & 0 \\
0 & 0 & 0 & 0 & X & 0 & 0 & 0 & X & 0 \\
0 & 0 & 0 & 0 & X & 0 & 0 & 0 & 0 & X
\end{array}\right]
\left[\begin{array}{c}
\dot{\mathbf{x}}_A \\
\dot{\mathbf{h}} \\
\hline
\mathbf{y}_A \\
\mathbf{x}_L \\
\mathbf{T} \\
\dot{\mathbf{n}}_V \text{ or } \dot{\mathbf{n}}_L \\
\mathbf{h}_V \\
\mathbf{h}_L \\
\mathbf{P}_A^{sat} \\
\mathbf{P}_B^{sat}
\end{array}\right]
$$

<u>Lower triangular block form</u>

$$
\left[\begin{array}{cccccccc|cc}
X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & X & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 0 & X & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 0 & X & X & 0 & 0 & 0 & 0 \\
X & X & X & X & X & X & X & 0 & 0 & 0 \\
X & X & X & X & X & X & X & X & 0 & 0 \\
\hline
X & X & X & X & X & X & X & X & X & 0 \\
X & X & X & X & X & X & X & X & 0 & X
\end{array}\right]
\left[\begin{array}{c}
\mathbf{T} \\
\mathbf{P}_A^{sat} \\
\mathbf{P}_B^{sat} \\
\mathbf{h}_L \\
\mathbf{y}_A \\
\mathbf{h}_V \\
\mathbf{x}_L \\
\dot{\mathbf{n}}_V \text{ or } \dot{\mathbf{n}}_L \\
\hline
\dot{\mathbf{x}}_A \\
\dot{\mathbf{h}}
\end{array}\right]
$$

Scalability to Large Systems

- n – number of algebraic equations
- τ – number of non-zeros in the sparsity matrix
- The maximum transversal algorithm has a worst case bound of $O(n\tau)$ although typical examples are more like $O(n) + O(\tau)$ (Duff, 1981)
- The lower triangular block algorithm also exhibits excellent scaling for large problems with an upper bound of $O(n) + O(\tau)$ (Duff and Reid, 1978)
- Similar to approaches for solving process design equations (1970s)

Reduction of Algebraic Equations

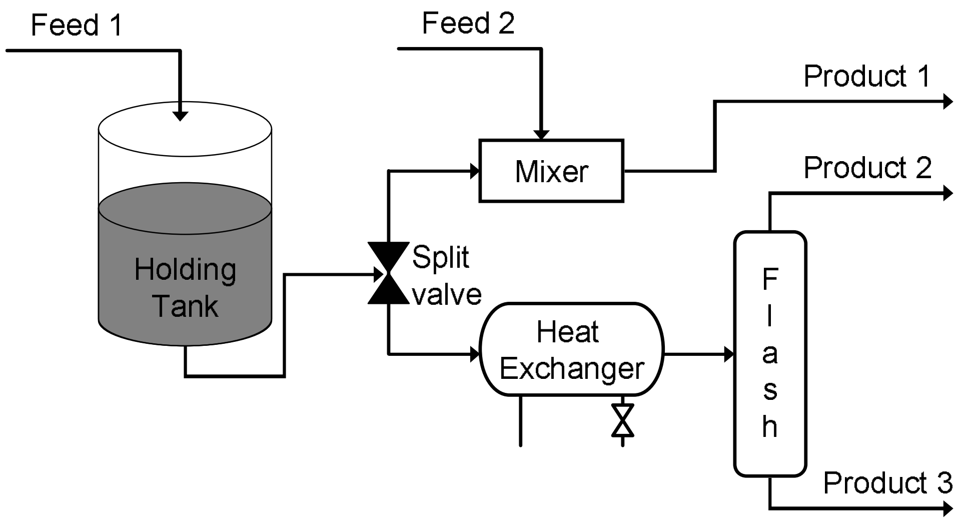
- Explicit transformation of algebraic equations
  - Transform model equations into an explicit form
    - Apply Tarjan's algorithm for precedence ordering
  - Model equations can be proprietary (not available to user, e.g. commercial simulator)
  - Neural networks
    - Extrapolation problems
    - No reliable error control strategy
  - In situ adaptive tabulation (ISAT)
    - Dynamic database with error control
    - Replacement for neural nets?

Example: Flowsheet Modeling and Model Reduction

- Multicomponent, multiphase object-oriented simulator
- FORTRAN 90 routines for fast execution
- DIPPR database with properties for >1700 compounds
- DASPK 3.0 for numerical integration and sensitivity analysis
- Current models are a compressor, splitter, mixer, vessel, heat exchanger, and flash column

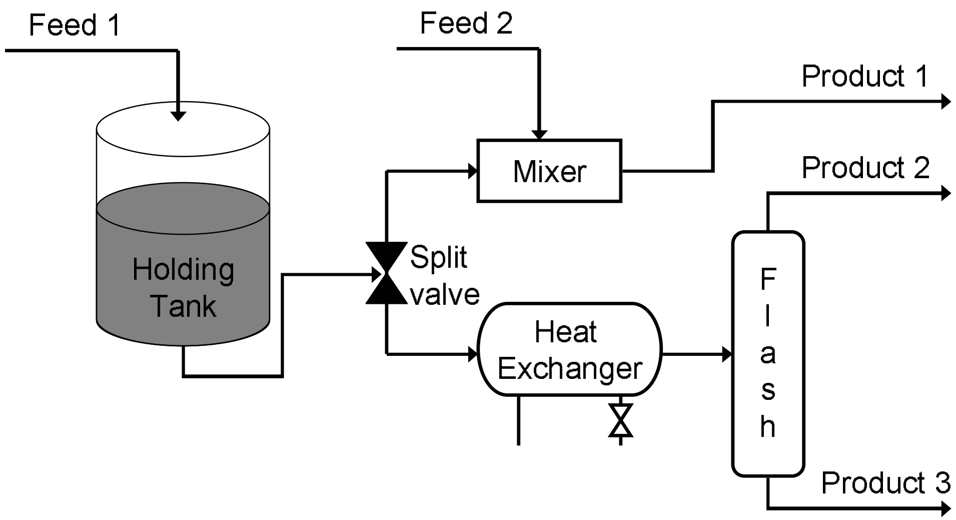
Example: Flowsheet Model

- Blending and separation
  - Feed streams: butane, pentane, hexane, heptane, and octane
- DAE model
  - 12 differential equations
  - 217 algebraic equations



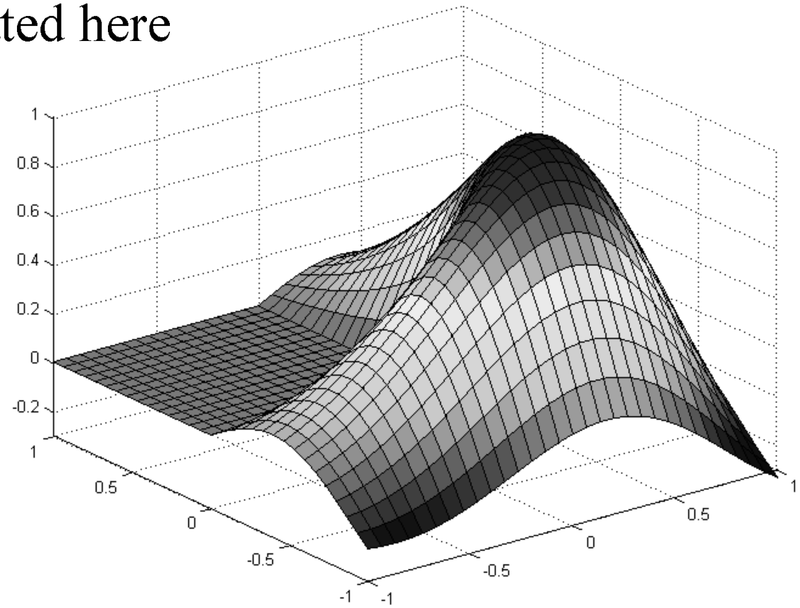
Example: Reduced Flowsheet Model Results

- Algebraic equation decomposition
  - 202 successively independent sets of variables and equations
  - One implicit set: 16 equations (flash column)
  - Model reduced from 229 to 28 states
  - 12 ODEs / 16 AEs



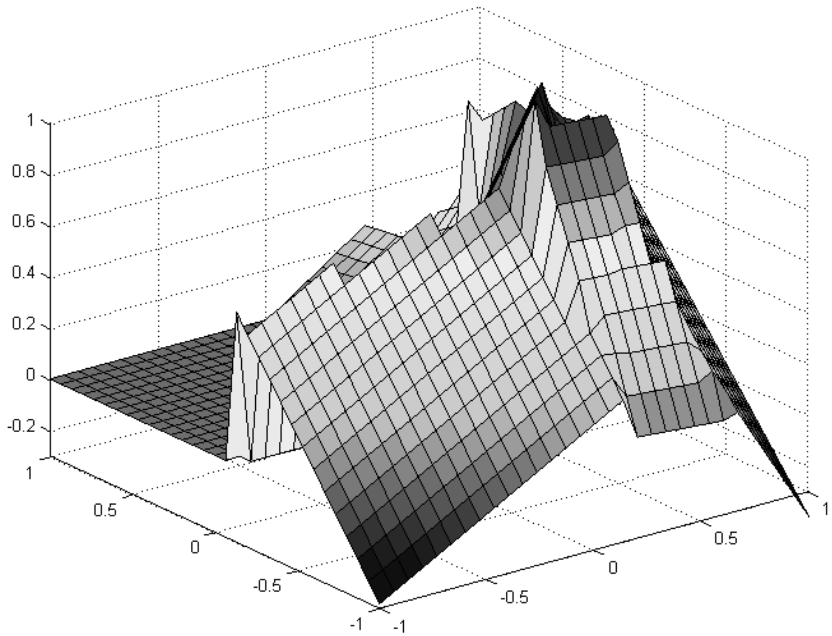
Example: ISAT vs. Neural Nets

- Nonlinear function test case (2 independent variables)
  - 1st eigenfunction of an L-shaped membrane
  - 2nd and 3rd eigenfunctions also appear on Mathworks' publications
  - Linear and nonlinear regions
  - Points that are not continuously differentiable
  - ISAT also handles function discontinuities, although that capability is not demonstrated here



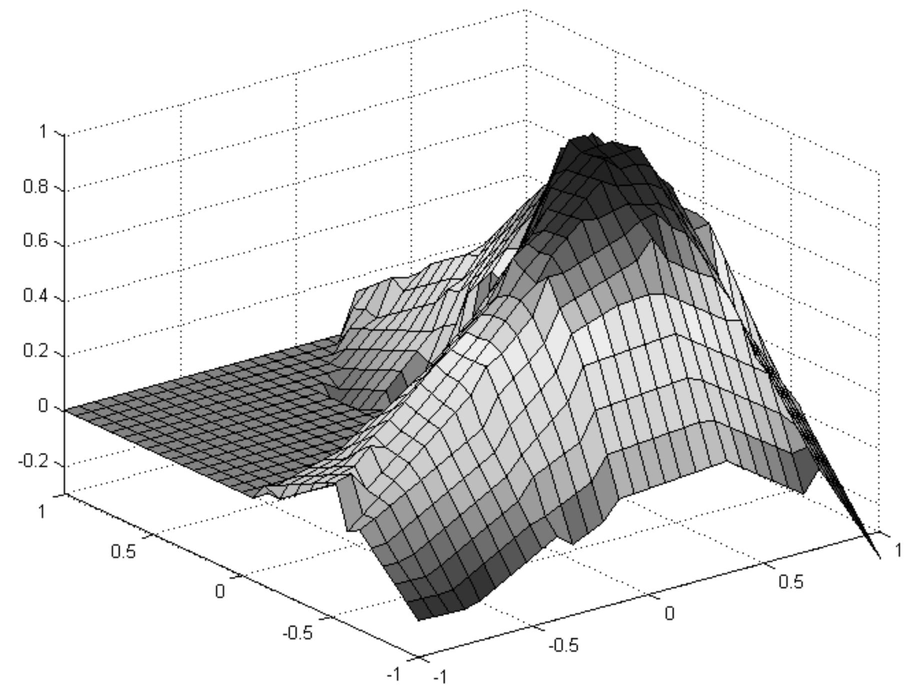
ISAT

- Principal tuning parameter (ε_{tol})
  - Set to ε_{tol} = 0.5 (extremely coarse)
  - Intuitive adjustable parameter – in this case little accuracy is required
  - ISAT created 12 linear regions (x_1, x_2, f)



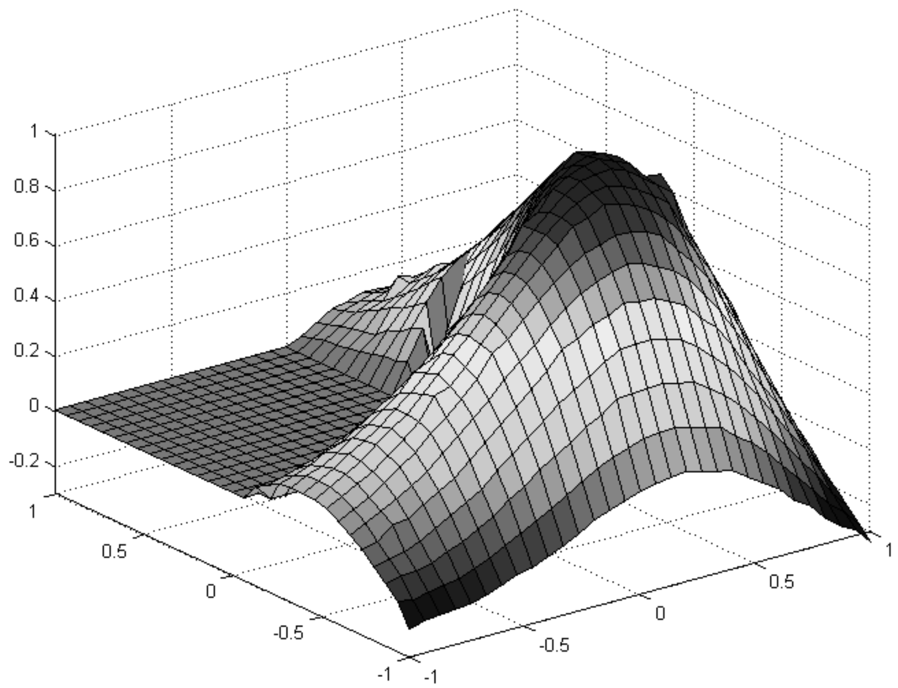
ISAT

- Principal tuning parameter
  - Set to $\varepsilon_{tol} = 0.1$
  - Moderate accuracy is required
  - ISAT created 48 linear regions



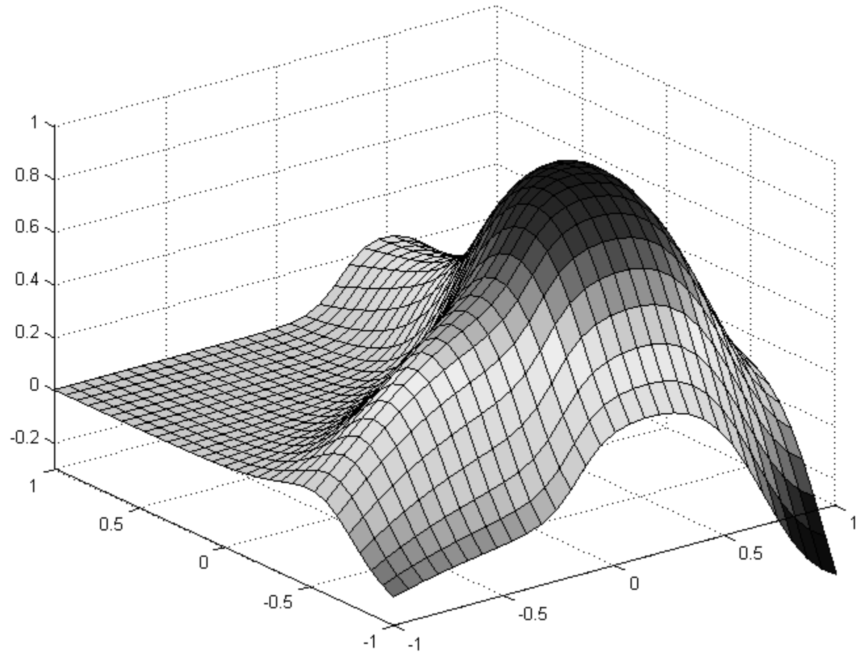
ISAT

- Principal tuning parameter
  - Set to $\varepsilon_{tol} = 0.01$
  - High accuracy is required
  - ISAT created 206 linear regions



Neural Net

- Principal tuning parameters
  - Structure: 2 layers
    - Hidden layer: 4 neurons, tangent function
    - Output layer: 1 neuron, linear function
  - Optimization tolerances
- Generated with MATLAB's neural net toolbox



Example: Conclusions

- ISAT advantages
  - Fewer tuning parameters
  - More intuitive tuning parameters
  - Approximates discontinuous and non continuously differentiable functions
  - Builds *in situ,* with no global optimizations

Conclusions

- Two step DAE model reduction process

  1. Reduction of differential equations
     - Adaptive ODE reduction
     - Predictor can also be used as a corrector (from example: 3 states → 1 state)
     - Substantial decreases in the number of ODEs are possible

Conclusions

- Two step DAE model reduction process

  2. Reduction of Algebraic Equation with ISAT
     - Example demonstrates ~10 times reduction in number of variables
     - Successful reduction of object-oriented flowsheet models with multicomponent processes
     - ISAT explicitly transforms sets of nonlinear equations with a given error tolerance
     - ISAT suggested as a replacement for neural networks