Combinatorial Optimization of Matrix-Vector Multiplication

Michael Wolf
University of Illinois
6/21/2007

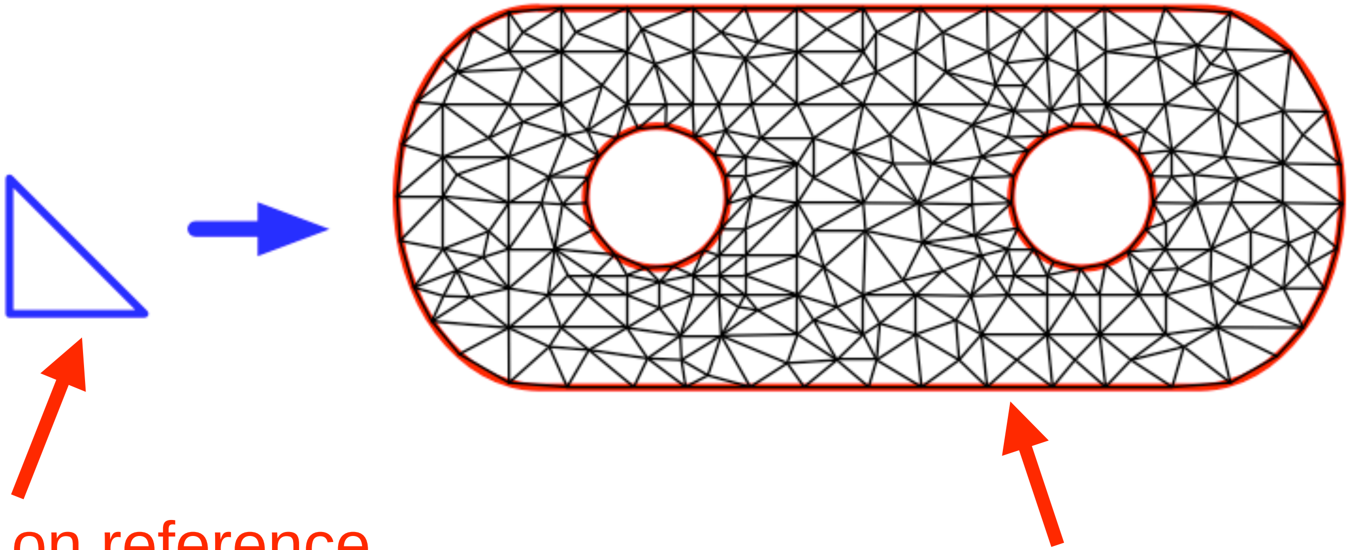
Optimization Problem

Objective: Generate set of operations for computing matrix-vector product with minimal number of multiply-add pairs (MAPs)

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \mathbf{r_1}^T \\ \hline \mathbf{r_2}^T \\ \hline \vdots \\ \hline \mathbf{r_m}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{r_1}^T\mathbf{x} \\ \mathbf{r_2}^T\mathbf{x} \\ \vdots \\ \mathbf{r_m}^T\mathbf{x} \end{bmatrix}$$

Motivation



Based on reference element, generate code to optimize construction of local stiffness matrices

Can use optimized code for every element in domain

- Reducing redundant operations in building finite element (FE) stiffness matrices
  - Reuse optimized code when problem is rerun

Related Work

- Finite element "Compilers" (FEniCS project)
  - www.fenics.org
  - FIAT (automates generations of FEs)
  - FFC (variational forms -> code for evaluation)
- Following work by Kirby, et al., Texas Tech, University of Chicago on FErari
  - Optimization of FFC generated code
  - Equivalent to optimizing matrix-vector product code

Matrix-Vector Multiplication

For 2D Laplace equation, we obtain following matrix-vector product to determine entries in local stiffness matrix

$$\mathbf{S}^e_{i,j} = y_{ni+j} = \mathbf{A}_{(ni+j,*)}\mathbf{x}$$

where

$$\mathbf{A}^T_{(ni+j,*)} = \begin{bmatrix} \left(\frac{\partial \phi_i}{\partial r}, \frac{\partial \phi_j}{\partial r}\right)_{\hat{e}} \\ \left(\frac{\partial \phi_i}{\partial r}, \frac{\partial \phi_j}{\partial s}\right)_{\hat{e}} \\ \left(\frac{\partial \phi_i}{\partial s}, \frac{\partial \phi_j}{\partial r}\right)_{\hat{e}} \\ \left(\frac{\partial \phi_i}{\partial s}, \frac{\partial \phi_j}{\partial s}\right)_{\hat{e}} \end{bmatrix} \qquad \mathbf{x} = \det(\mathbf{J}) \begin{bmatrix} \frac{\partial r}{\partial x}\frac{\partial r}{\partial x} + \frac{\partial r}{\partial y}\frac{\partial r}{\partial y} \\ \frac{\partial r}{\partial x}\frac{\partial s}{\partial x} + \frac{\partial r}{\partial y}\frac{\partial s}{\partial y} \\ \frac{\partial s}{\partial x}\frac{\partial r}{\partial x} + \frac{\partial s}{\partial y}\frac{\partial r}{\partial y} \\ \frac{\partial s}{\partial x}\frac{\partial s}{\partial x} + \frac{\partial s}{\partial y}\frac{\partial s}{\partial y} \end{bmatrix}$$

Element independent

Element dependent

Possible Optimizations - Collinear Rows

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & 0 \\ 3 & 3 & 3 & 0 \\ 2 & 2 & 2 & 0 \\ 5 & 5 & 5 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\mathbf{r_2} = 1.5\mathbf{r_1}$$

Possible Optimizations - Collinear Rows

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & 0 \\ 3 & 3 & 3 & 0 \\ 2 & 2 & 2 & 0 \\ 5 & 5 & 5 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\mathbf{r_2} = 1.5\mathbf{r_1} \Rightarrow y_2 = 1.5y_1 \quad \boxed{\text{1 MAP}}$$

Possible Optimizations - Collinear Rows

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & 0 \\ 3 & 3 & 3 & 0 \\ 2 & 2 & 2 & 0 \\ 5 & 5 & 5 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\mathbf{r_3} = \mathbf{r_1} \Rightarrow y_3 = y_1 \quad \boxed{\text{0 MAPs}}$$

Special case when rows identical

Possible Optimizations - Partial Collinear Rows

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & 0 \\ 3 & 3 & 3 & 0 \\ 2 & 2 & 2 & 0 \\ 5 & 5 & 5 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\mathbf{r_4} = 2.5\mathbf{r_1} + 8\mathbf{e_4}$$

Possible Optimizations - Partial Collinear Rows

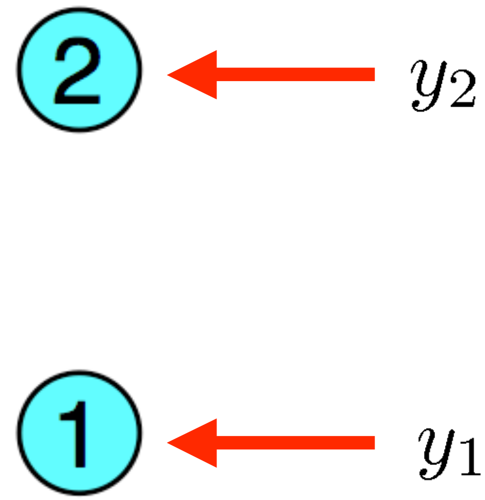
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & 0 \\ 3 & 3 & 3 & 0 \\ 2 & 2 & 2 & 0 \\ 5 & 5 & 5 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\mathbf{r_4} = 2.5\mathbf{r_1} + 8\mathbf{e_4} \Rightarrow y_4 = 2.5 y_1 + 8 x_4$$

$\boxed{\text{2 MAPs}}$

Graph Model - Resulting Vector Entry Vertices

$$
\begin{array}{c|c:c:c:c|}
\mathbf{r_1}^T & 8 & 4 & 4 & 8 \\
\mathbf{r_2}^T & 0 & -2 & -2 & -4
\end{array}
$$

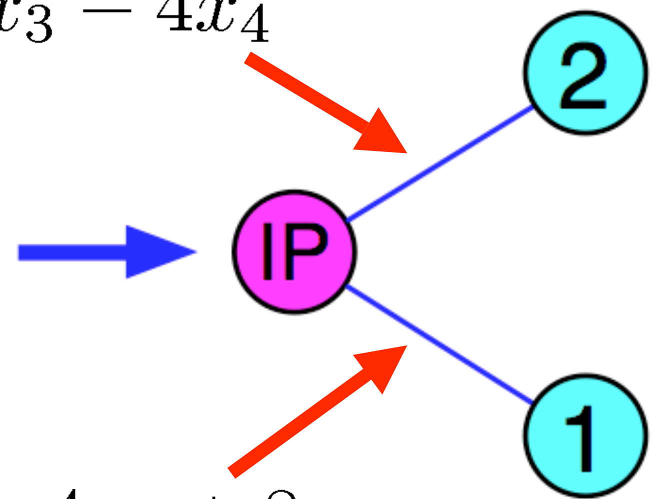


- Entries in resulting vector represented by vertices in graph model

Graph Model - Inner-Product Vertex and Edges

$$y_2 = -2x_2 - 2x_3 - 4x_4$$

$$\begin{array}{c|c:c:c:c|}
\mathbf{r_1}^T & 8 & 4 & 4 & 8 \\
\hline
\mathbf{r_2}^T & 0 & -2 & -2 & -4
\end{array}$$

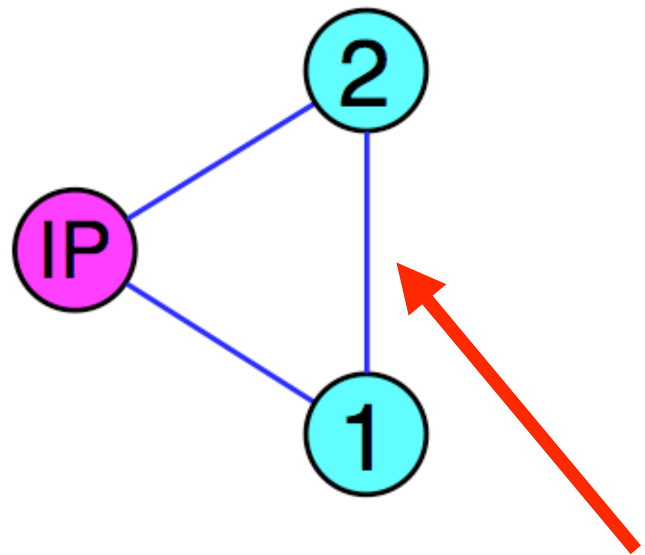


$$y_1 = 8x_1 + 4x_2 + 4x_3 + 8x_4$$

- Additional inner-product (IP) vertex
- Edges connect IP vertex to every other vertex, representing inner-product operation

Graph Model - Row Relationship Edges

$$\begin{array}{c|cccc|} \mathbf{r_1}^T & 8 & 4 & 4 & 8 \\ \mathbf{r_2}^T & 0 & -2 & -2 & -4 \end{array}$$



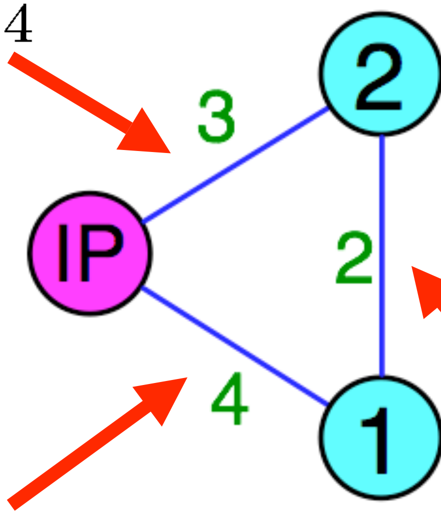
$$y_1 = -2y_2 + 8x_1$$
$$y_2 = -0.5y_1 + 4x_1$$

- Operations resulting from relationships between rows represented by edges between corresponding vertices

Graph Model - Edge Weights

$$y_2 = -2x_2 - 2x_3 - 4x_4$$

$\mathbf{r_1}^T$	8	4	4	8
$\mathbf{r_2}^T$	0	-2	-2	-4



$$y_1 = 8x_1 + 4x_2 + 4x_3 + 8x_4$$

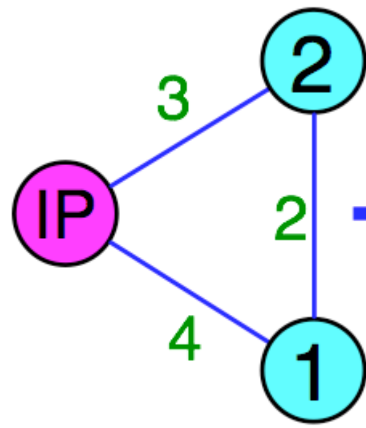
$$y_1 = -2y_2 + 8x_1$$
$$y_2 = -0.5y_1 + 4x_1$$

- Edge weights are MAP costs for operations

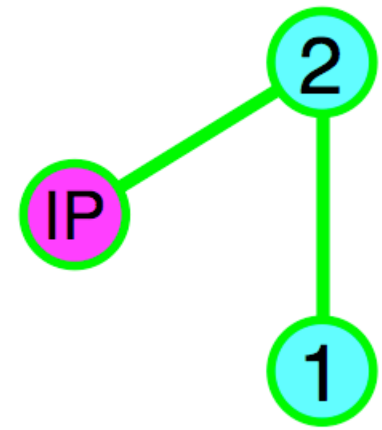
Graph Model Solution

$\mathbf{r_1}^T$	8	4	4	8
$\mathbf{r_2}^T$	0	-2	-2	-4

Matrix



Graph

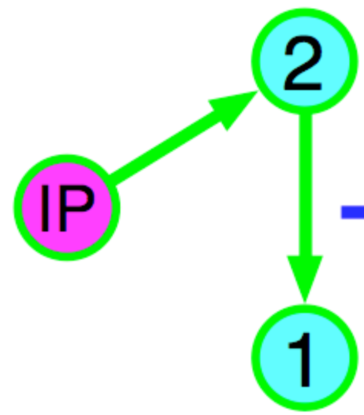


MST(5)

- Solution is minimum spanning tree (MST)
  - Minimum subgraph
  - Connected and spans vertices
  - Acyclic

Graph Model Solution

$$\begin{array}{c|c|c|c|c|} \hline \mathbf{r_1}^T & 8 & 4 & 4 & 8 \\ \hline \mathbf{r_2}^T & 0 & -2 & -2 & -4 \\ \hline \end{array}$$



$$\begin{aligned} y_2 &= -2x_2 - 2x_3 - 4x_4 \\ y_1 &= -2y_2 + 8x_1 \end{aligned}$$

Matrix MST Traversal Instructions

- Prim's algorithm to find MST (polynomial time)
- MST traversal yields operations to optimally compute (for these relationships) matrix-vector product

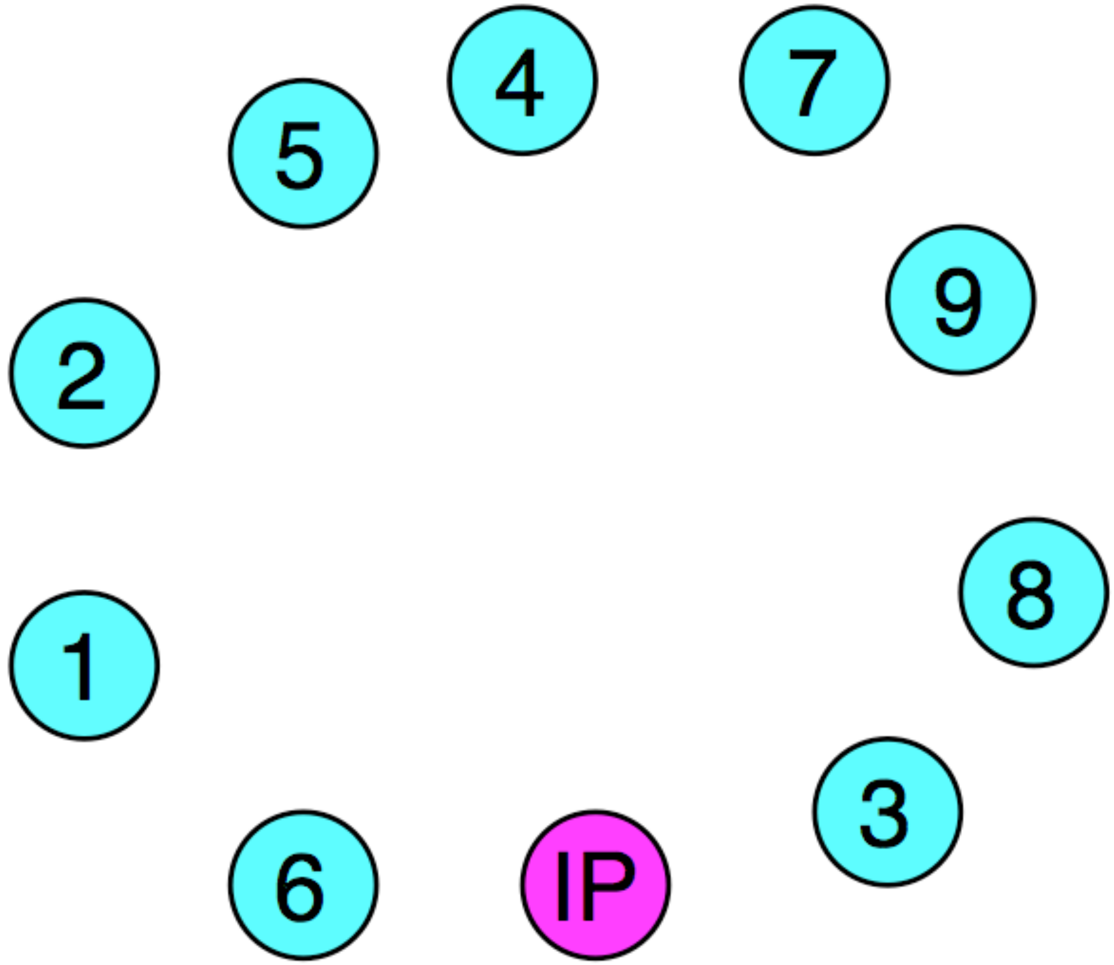
Graph Model Example

$\mathbf{r_1}^T$	0	4/3	0
$\mathbf{r_2}^T$	0	0	1/2
$\mathbf{r_3}^T$	1/2	0	0
$\mathbf{r_4}^T$	1/6	1/6	0
$\mathbf{r_5}^T$	0	1/6	1/6
$\mathbf{r_6}^T$	0	-2/3	-2/3
$\mathbf{r_7}^T$	-4/3	-4/3	0
$\mathbf{r_8}^T$	0	-4/3	-4/3
$\mathbf{r_9}^T$	4/3	4/3	4/3

- Matrix used for building FE local stiffness matrices
  - 2D Laplace Equation
  - 2nd order Lagrange polynomial basis
- Simplified version of matrix
  - Identical rows removed
  - Several additional rows removed

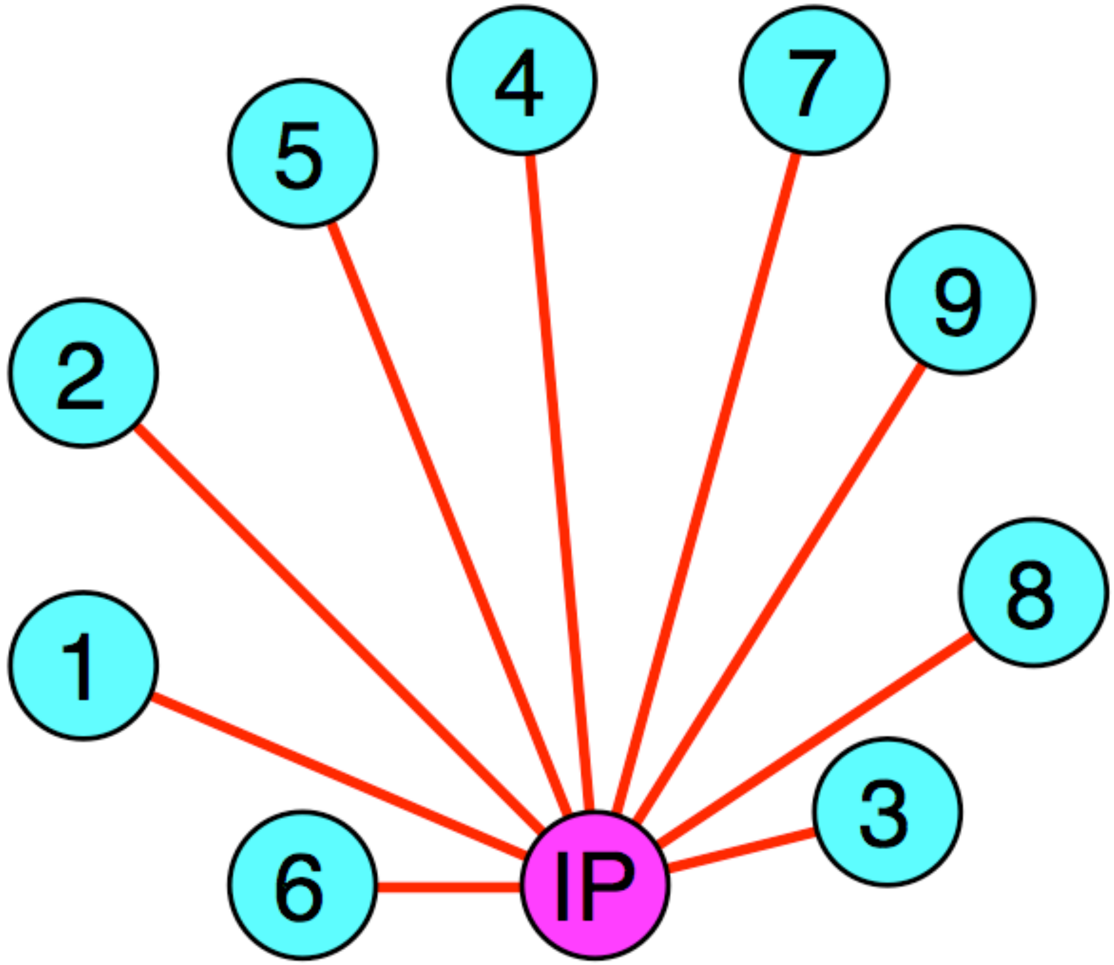
Graph Model Example - Vertices

$\mathbf{r}_1^T$	0	4/3	0
$\mathbf{r}_2^T$	0	0	1/2
$\mathbf{r}_3^T$	1/2	0	0
$\mathbf{r}_4^T$	1/6	1/6	0
$\mathbf{r}_5^T$	0	1/6	1/6
$\mathbf{r}_6^T$	0	-2/3	-2/3
$\mathbf{r}_7^T$	-4/3	-4/3	0
$\mathbf{r}_8^T$	0	-4/3	-4/3
$\mathbf{r}_9^T$	4/3	4/3	4/3



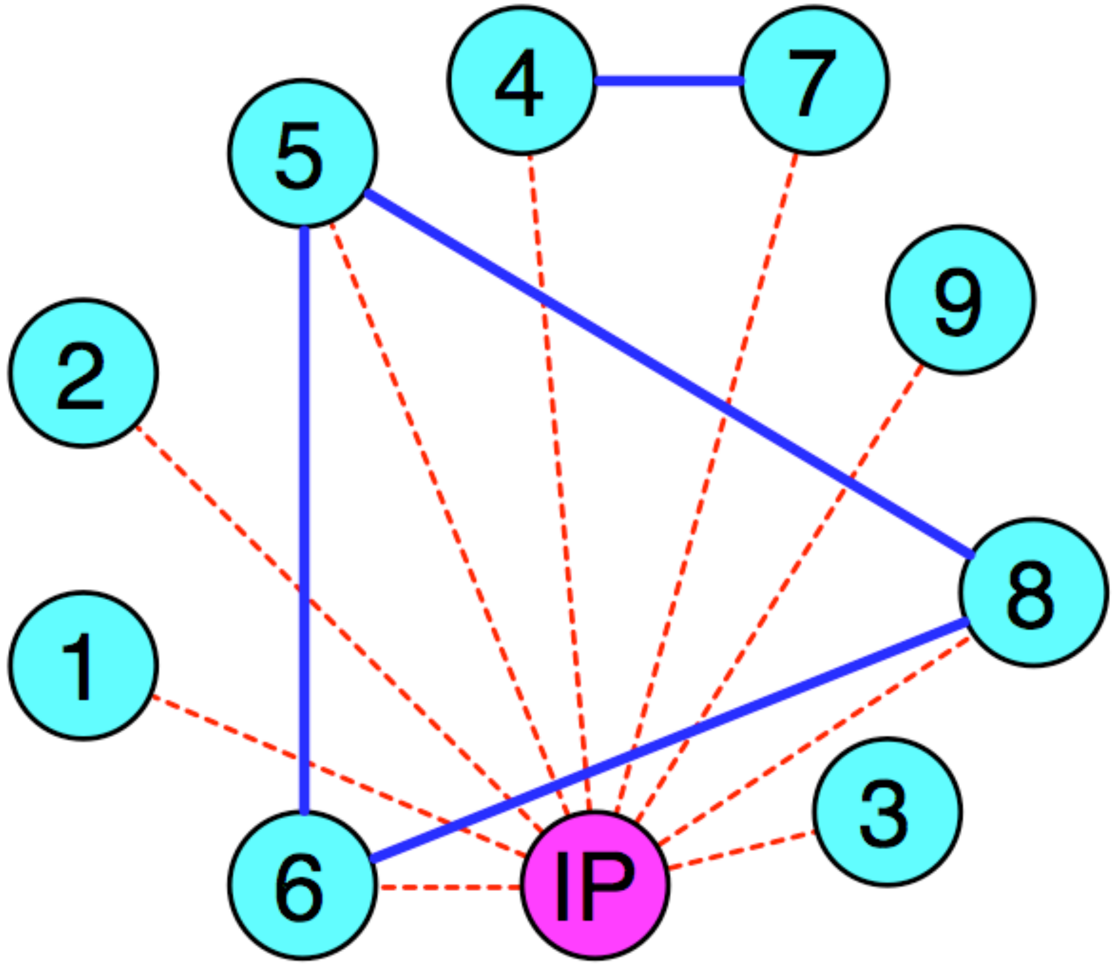
Graph Model Example - Inner Product Edges

${\mathbf{r}_1}^T$	0	4/3	0
${\mathbf{r}_2}^T$	0	0	1/2
${\mathbf{r}_3}^T$	1/2	0	0
${\mathbf{r}_4}^T$	1/6	1/6	0
${\mathbf{r}_5}^T$	0	1/6	1/6
${\mathbf{r}_6}^T$	0	-2/3	-2/3
${\mathbf{r}_7}^T$	-4/3	-4/3	0
${\mathbf{r}_8}^T$	0	-4/3	-4/3
${\mathbf{r}_9}^T$	4/3	4/3	4/3



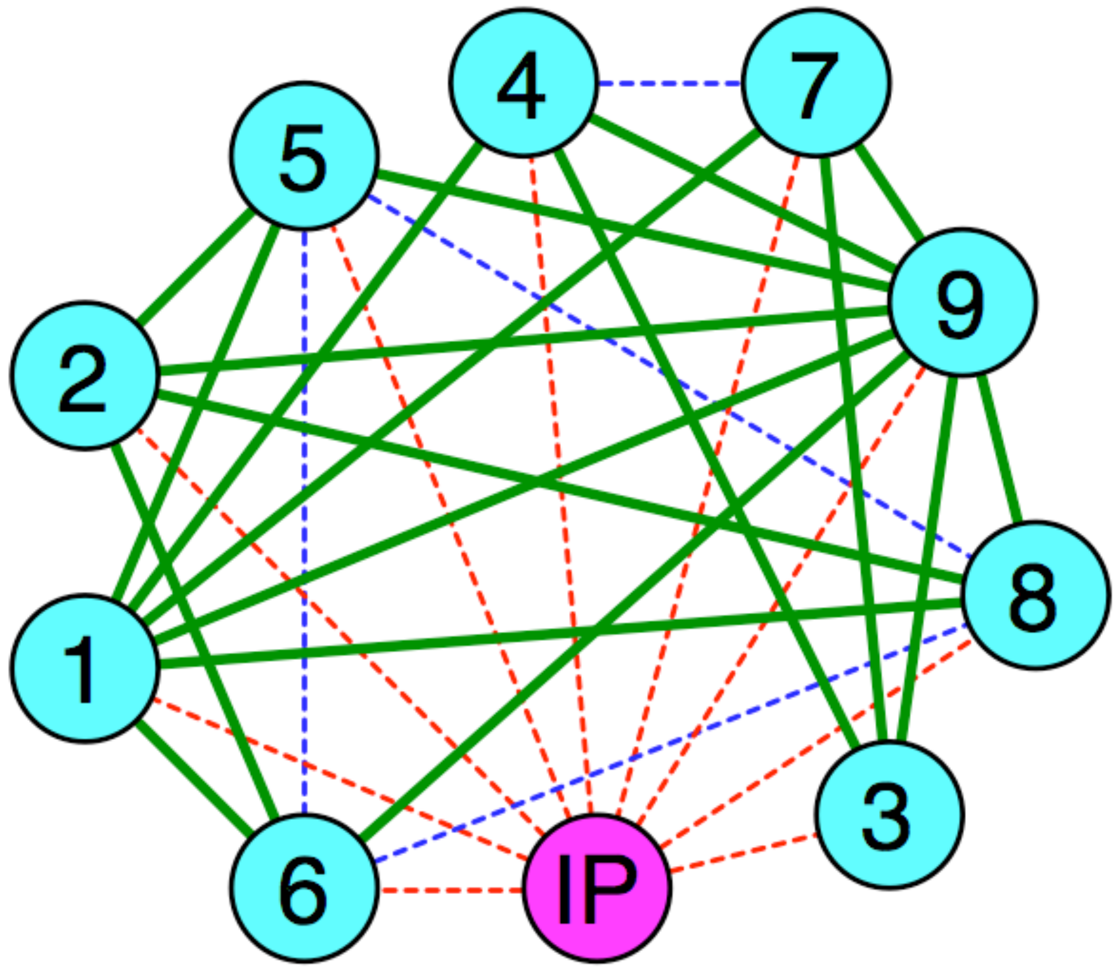
Graph Model Example - Collinear Edges

$\mathbf{r_1}^T$	0	4/3	0
$\mathbf{r_2}^T$	0	0	1/2
$\mathbf{r_3}^T$	1/2	0	0
$\mathbf{r_4}^T$	1/6	1/6	0
$\mathbf{r_5}^T$	0	1/6	1/6
$\mathbf{r_6}^T$	0	-2/3	-2/3
$\mathbf{r_7}^T$	-4/3	-4/3	0
$\mathbf{r_8}^T$	0	-4/3	-4/3
$\mathbf{r_9}^T$	4/3	4/3	4/3



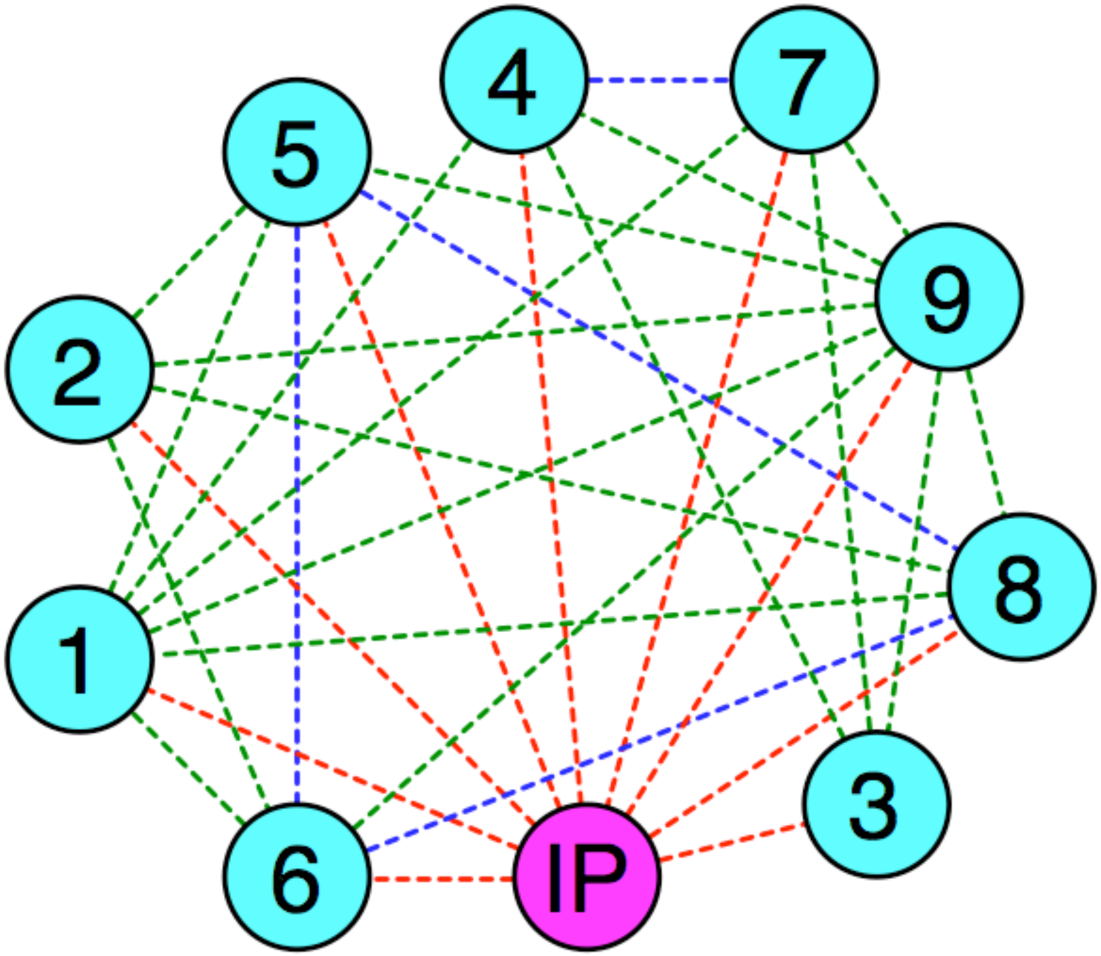
Graph Model Example - Partial Collinear Edges

$\mathbf{r_1}^T$	0	4/3	0
$\mathbf{r_2}^T$	0	0	1/2
$\mathbf{r_3}^T$	1/2	0	0
$\mathbf{r_4}^T$	1/6	1/6	0
$\mathbf{r_5}^T$	0	1/6	1/6
$\mathbf{r_6}^T$	0	-2/3	-2/3
$\mathbf{r_7}^T$	-4/3	-4/3	0
$\mathbf{r_8}^T$	0	-4/3	-4/3
$\mathbf{r_9}^T$	4/3	4/3	4/3



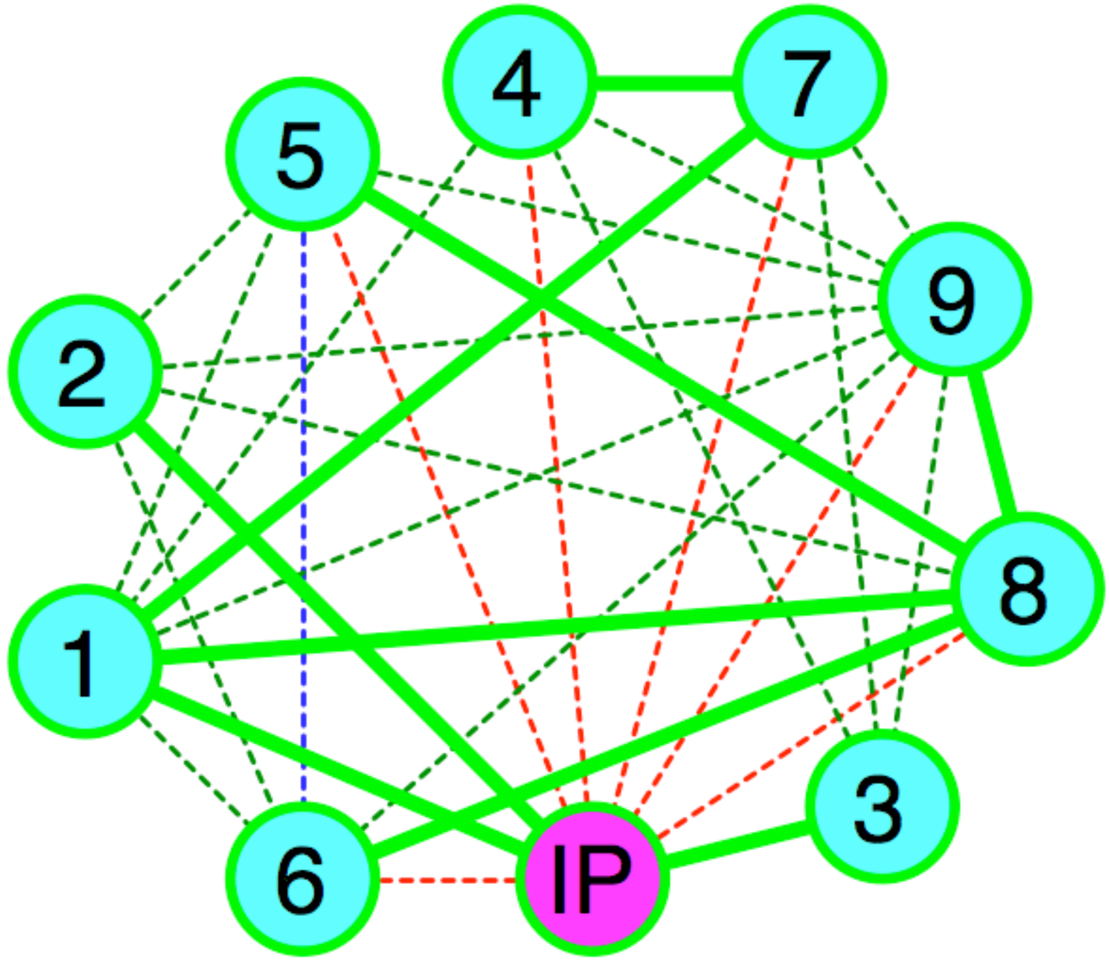
Graph Model Example - Final Graph

$\mathbf{r_1}^T$	0	4/3	0
$\mathbf{r_2}^T$	0	0	1/2
$\mathbf{r_3}^T$	1/2	0	0
$\mathbf{r_4}^T$	1/6	1/6	0
$\mathbf{r_5}^T$	0	1/6	1/6
$\mathbf{r_6}^T$	0	-2/3	-2/3
$\mathbf{r_7}^T$	-4/3	-4/3	0
$\mathbf{r_8}^T$	0	-4/3	-4/3
$\mathbf{r_9}^T$	4/3	4/3	4/3



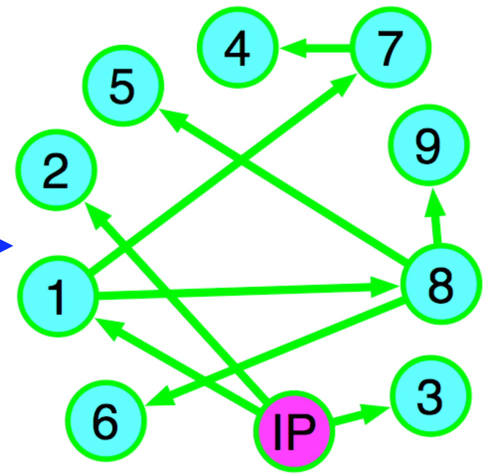
Graph Model Example - Solution (MST)

$\mathbf{r_1}^T$	0	4/3	0
$\mathbf{r_2}^T$	0	0	1/2
$\mathbf{r_3}^T$	1/2	0	0
$\mathbf{r_4}^T$	1/6	1/6	0
$\mathbf{r_5}^T$	0	1/6	1/6
$\mathbf{r_6}^T$	0	-2/3	-2/3
$\mathbf{r_7}^T$	-4/3	-4/3	0
$\mathbf{r_8}^T$	0	-4/3	-4/3
$\mathbf{r_9}^T$	4/3	4/3	4/3



Graph Model Example - Instructions Generated

$\mathbf{r_1}^T$	0	4/3	0
$\mathbf{r_2}^T$	0	0	1/2
$\mathbf{r_3}^T$	1/2	0	0
$\mathbf{r_4}^T$	1/6	1/6	0
$\mathbf{r_5}^T$	0	1/6	1/6
$\mathbf{r_6}^T$	0	-2/3	-2/3
$\mathbf{r_7}^T$	-4/3	-4/3	0
$\mathbf{r_8}^T$	0	-4/3	-4/3
$\mathbf{r_9}^T$	4/3	4/3	4/3



$$
\begin{aligned}
y_3 &= 0.5x_1 \\
y_2 &= 0.5x_3 \\
y_1 &= (4/3)x_2 \\
y_8 &= -y_1 - (4/3)x_3 \\
y_7 &= -y_1 - (4/3)x_1 \\
y_9 &= -y_8 + (4/3)x_1 \\
y_6 &= 0.5y_8 \\
y_5 &= (-1/8)y_8 \\
y_4 &= (-1/8)y_7
\end{aligned}
$$

Matrix (16 nz) | MST traversal | Instructions (9 MAPs)

Graph Model Results - 2D Laplace Equation

Order	Unoptimized MAPs	Graph MAPs
1	10	**7**
2	34	**14**
3	108	**43**
4	292	**152**
5	589	**366**
6	1070	**686**

60% decrease

- Graph model shows significant improvement over unoptimized algorithm

Graph Model Results - 3D Laplace Equation

Order	Unoptimized MAPs	Graph MAPs
1	21	**17**
2	177	**79**
3	789	**342**
4	2586	**1049**
5	7125	**3592**
6	16749	**8835**

59% decrease

- Again graph model requires significantly fewer MAPs than unoptimized algorithm

Limitation of Graph Model

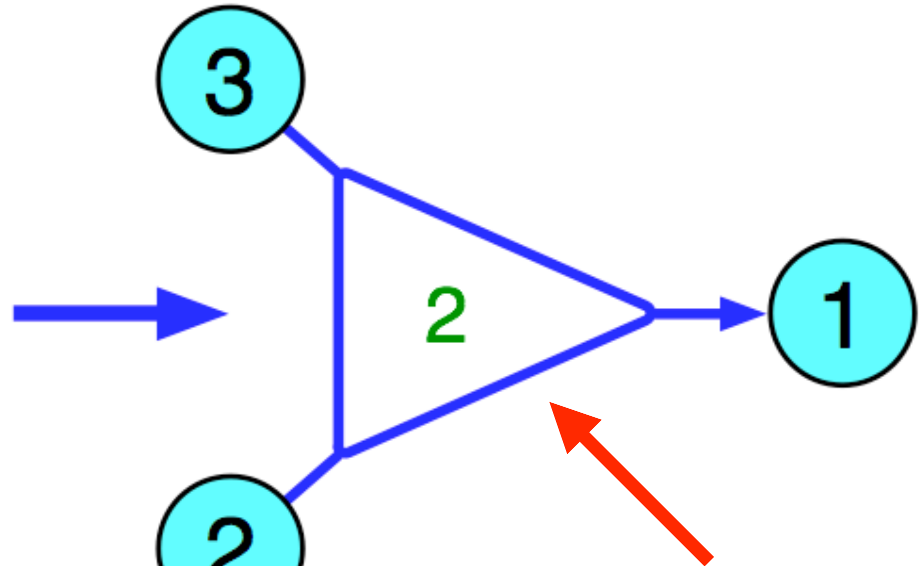
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 4 & 4 & 4 & 4 \\ 0 & 0 & 2 & 2 \\ 2 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\mathbf{r_2} = 2\mathbf{r_3} + 2\mathbf{r_4} \Rightarrow y_2 = 2y_3 + 2y_4$$

- Edges connect 2 vertices
- Can represent only binary row relationships
- Cannot exploit linear dependency of more than two rows
- Thus, hypergraphs needed

Hypergraph Model

$\mathbf{r_1}^T$	3	3	3	3
$\mathbf{r_2}^T$	1	1	0	0
$\mathbf{r_3}^T$	0	0	1	1

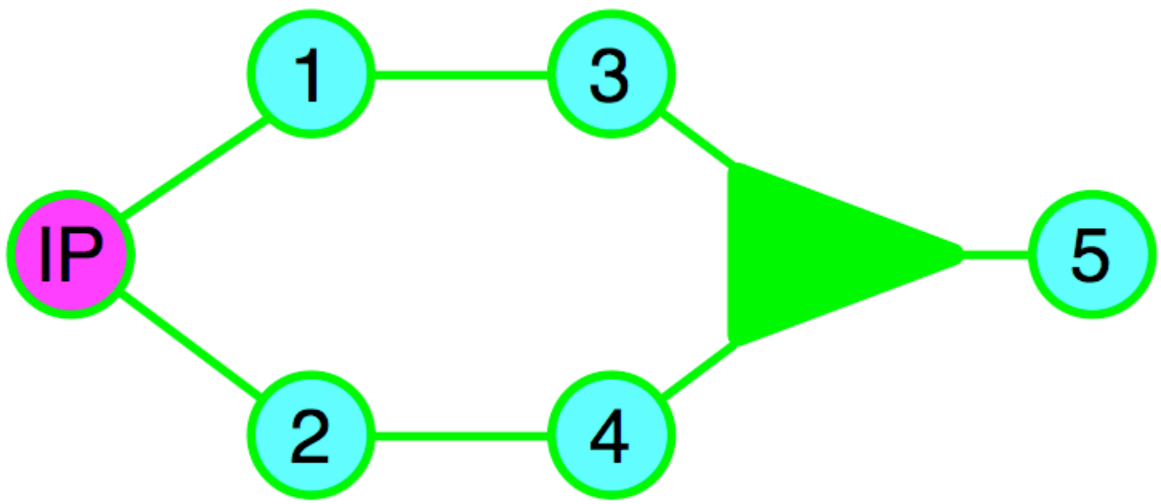


$$y_1 = 3y_2 + 3y_3$$

- Same edges (2-vertex hyperedges) as graph model
- Additional higher cardinality hyperedges for more complicated relationships
  - Limiting to 3-vertex linear dependency hyperedges for this talk

Hypergraph Model

- Extended Prim's algorithm to include hyperedges
- Polynomial time algorithm
- Solution not necessarily a tree
  - {IP,1,3,5}
  - {IP,2,4,5}
- No guarantee of optimum solution
- Finding optimum solution to hypergraph problem NP-hard



Hypergraph Model Results - 2D Laplace Equation

Order	Unoptimized MAPs	Graph MAPs	HGraph MAPs
1	10	7	**6**
2	34	**14**	**14**
3	108	**43**	**43**
4	292	152	**150**
5	589	366	**363**
6	1070	**686**	**686**

- Hypergraph solution slightly better for some orders but not significantly better
- Graph algorithm close to optimal?
  - 3 Columns
  - Binary relationships may be good enough

Hypergraph Model Results - 3D Laplace Equation

Order	Unoptimized MAPs	Graph MAPs	HGraph MAPs
1	21	**17**	**17**
2	177	79	**68**
3	789	342	**297**
4	2586	1049	**852**
5	7125	3592	**3261**
6	16749	8835	**8340**

19% additional decrease

- Hypergraph solution significantly better than graph solution for many orders

Future Work

- Higher cardinality hyperedges
  - Perhaps useful for 3D FE problems
  - Implemented 4, 5, 6 vertex hyperedges
  - Hyperedge explosion
  - Need efficient hyperedge pruning algorithms
- More complicated hyperedge relationships
  - Similar to partial collinear row relationships for edges
- Optimal and more nearly optimal solution methods
  - Combinatorial optimization formulation
- Other matrices

Acknowledgements/Thanks

- DOE and Krell Institute
  - Financial support
  - Collaborative opportunities
- Professor Michael Heath, advisor
- Professor Robert Kirby, Texas Tech University
- Dr. Erik Boman, Sandia National Laboratories

2D Laplace Equation Matrices

Order	Rows	Entries	Nonzeros
1	6	18	10
2	21	63	34
3	55	165	108
4	120	360	292
5	231	693	589
6	406	1218	1070

- 3 Columns

3D Laplace Equation Matrices

Order	Rows	Entries	Nonzeros
1	10	60	21
2	55	330	177
3	210	1260	789
4	630	3780	2586
5	1596	9576	7125
6	3570	21420	16749

- 6 Columns

Accuracy

Relative Error 2D Laplace

Order	GPCR Error	HGraph Error
1	0	0
2	2.53565e-09	2.55594e-09
3	6.40668e-09	2.44340e-09
4	2.47834e-10	9.30090e-09
5	4.95544e-09	5.87721e-09
6	4.28141e-09	4.28166e-09

Relative Error 3D Laplace

Order	GPCR Error	HGraph Error
1	0	0
2	9.33830e-09	7.35996e-09
3	2.60053e-08	3.51190e-08
4	8.31206e-09	1.47134e-08
5	4.22496e-08	6.30277e-08
6	1.07992e-06	1.41391e-06

- Single precision input matrices
- Single precision code generation