Grid Algebra in Finite Difference Code

Heather K. Hardeman

&

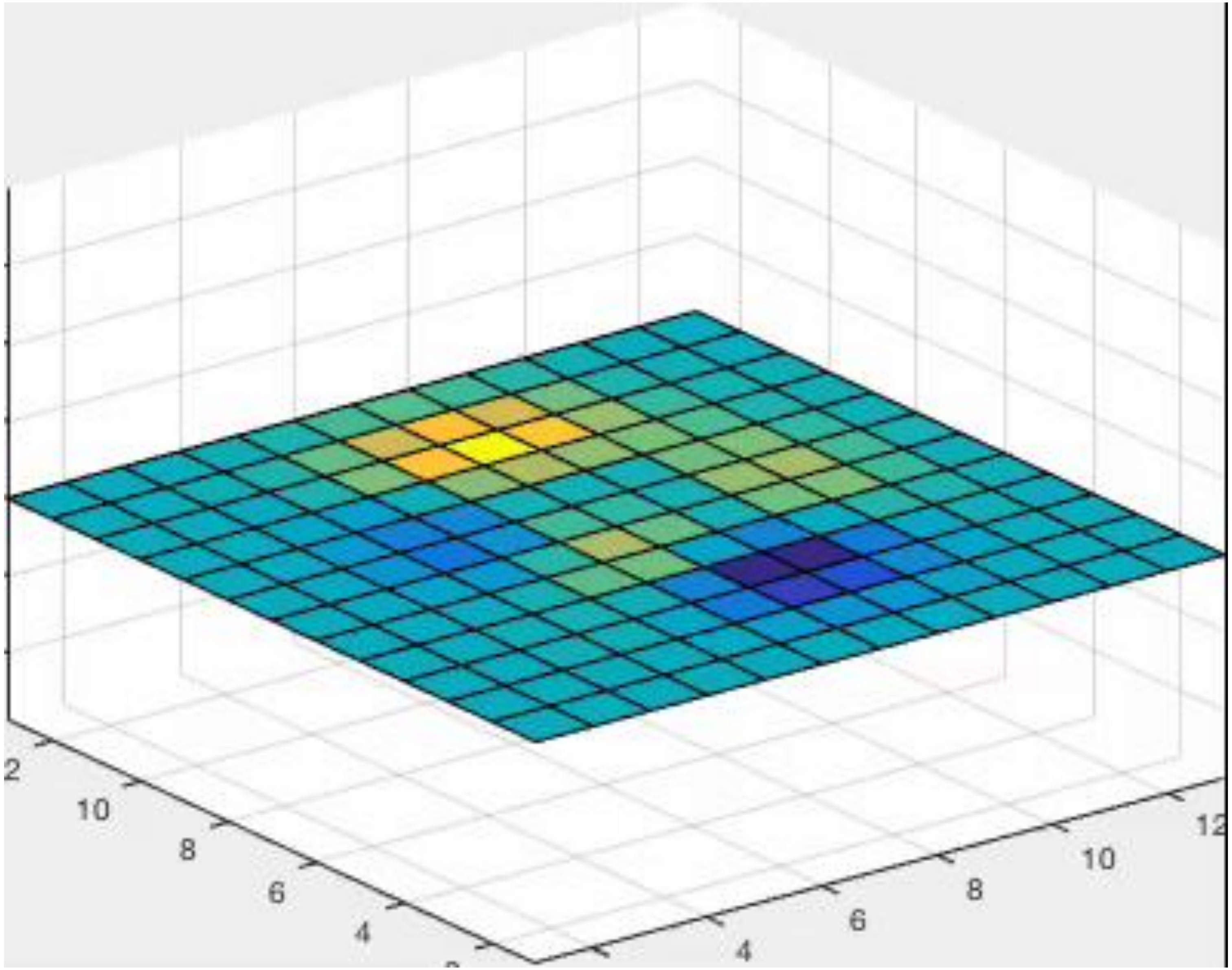
Michael P. Lamoureux

December 2, 2016

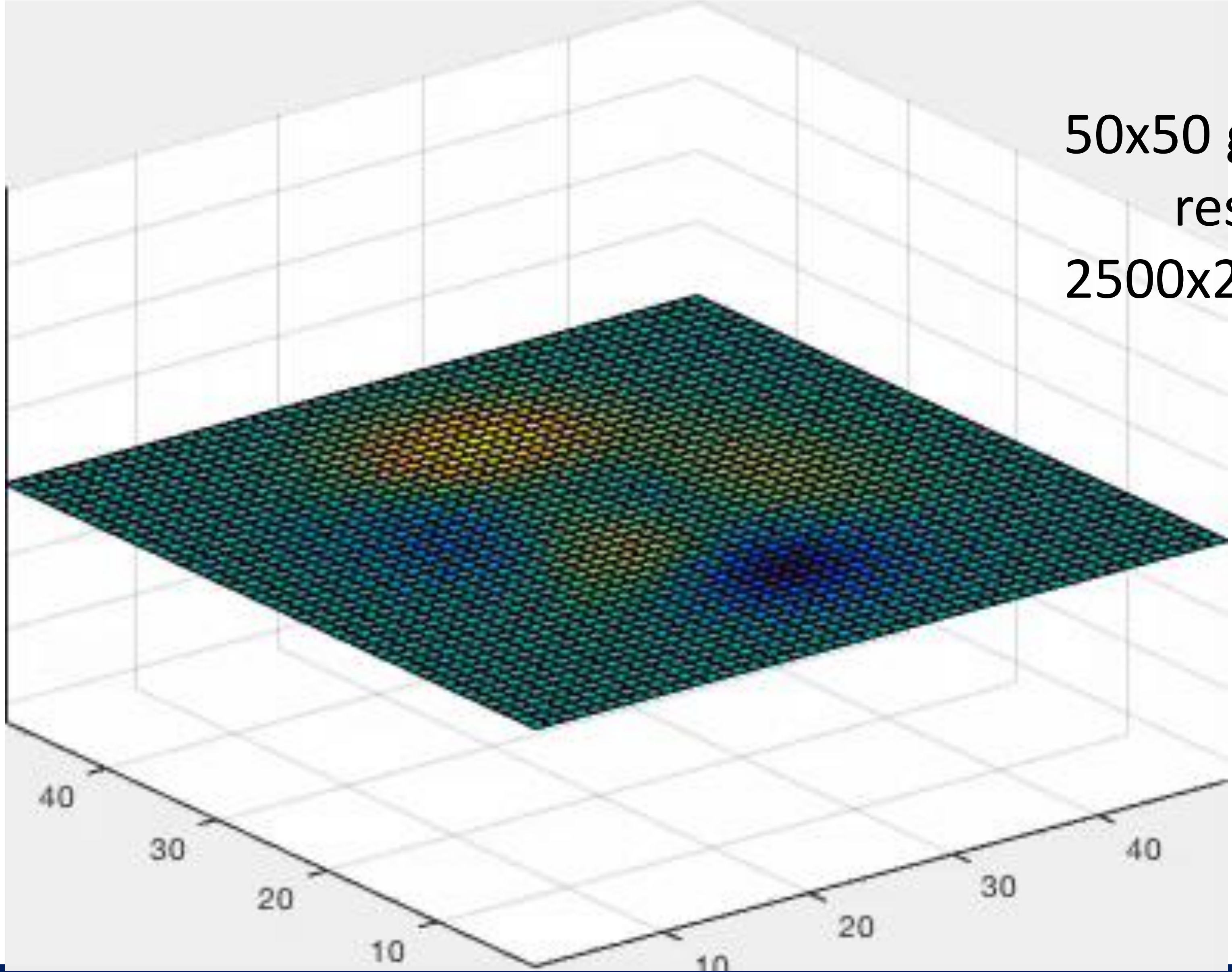
Outline

- Motivation
- Grid representation of operators
- Linear algebra on a grid
- Factoring the Laplacian
- Numerics for the wave equation in grid form
- Conclusions

Motivation – computing on a grid



Motivation – big grids, and big operators



50x50 grid
results in
2500x2500 matrices

Motivation

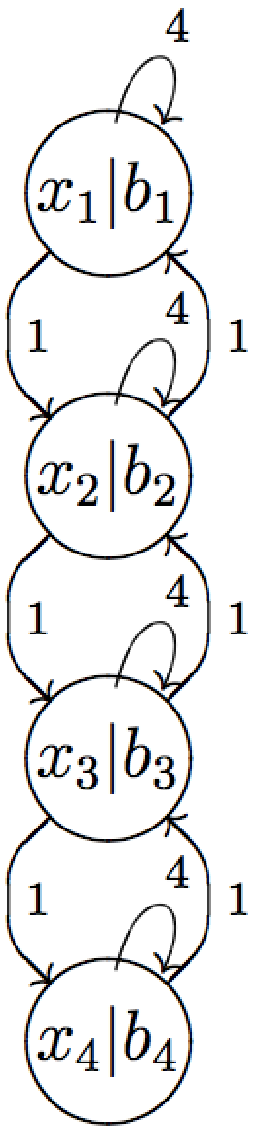
- Finite difference methods are an effective, efficient method for solving many differential equations.
- PDEs in 2D and 3D lead to large, sparse matrices.
- Implicit methods require the solution of these large matrices.
- We want faster solution methods, built on the grid geometry.

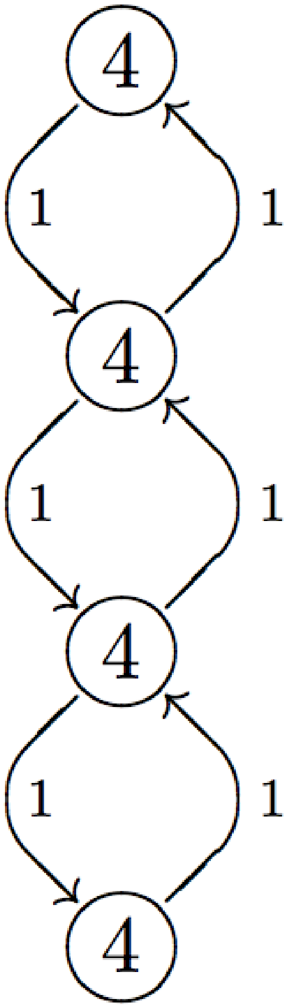
- Speed of order O(#grid points), per time step.

- Grid algebra is linear algebra, directly represented on a 2D or 3D grid.

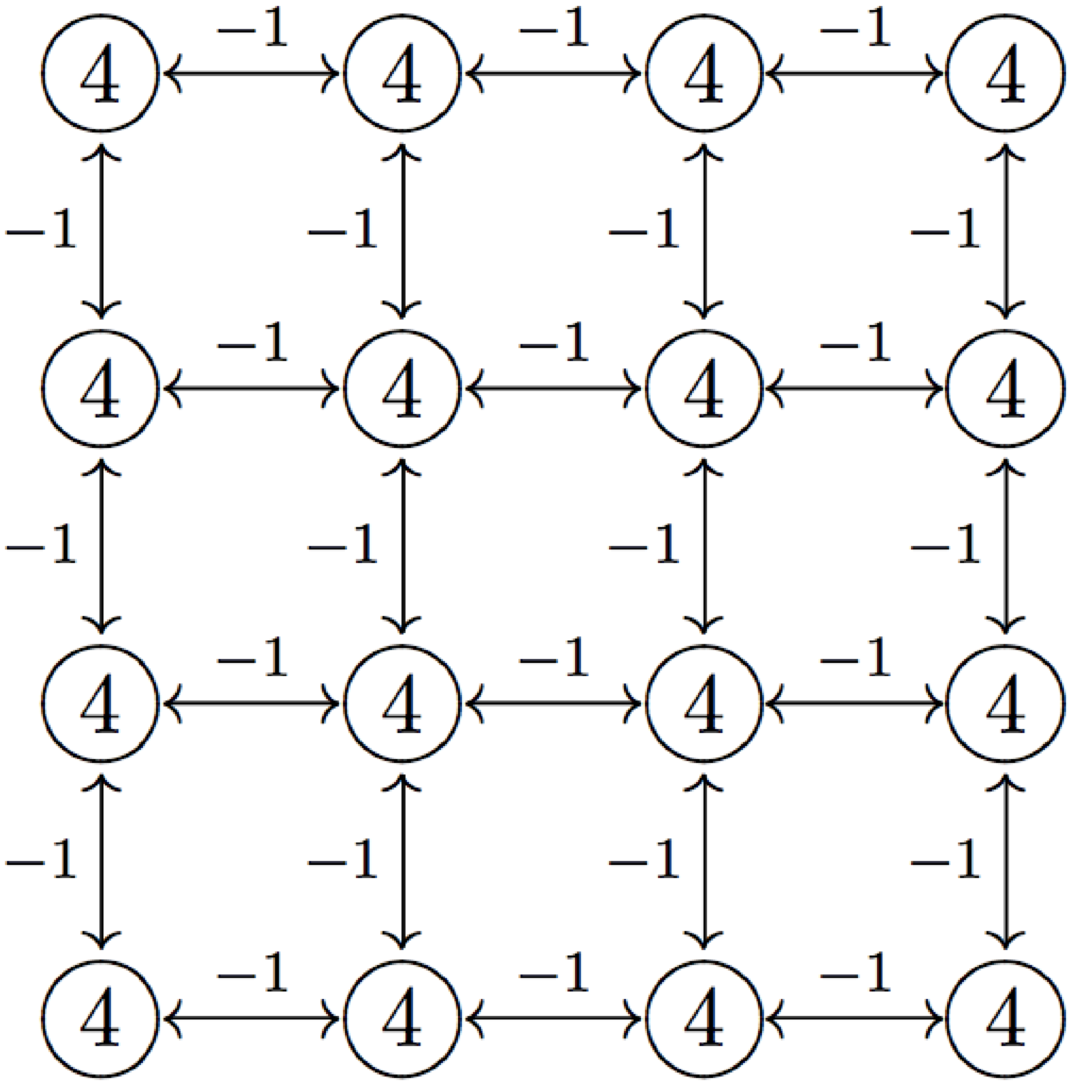
Matrix Ax = b. As an equation. As a graph. As a reduced graph.

$$\begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$





Laplace operator on 4x4 grid



Laplace operator, as a 16x16 matrix

$$
\begin{bmatrix}
4 & -1 & & & -1 & & & & & & & & & & & \\
-1 & 4 & -1 & & & -1 & & & & & & & & & & \\
 & -1 & 4 & -1 & & & -1 & & & & & & & & & \\
 & & -1 & 4 & & & & -1 & & & & & & & & \\
-1 & & & & 4 & -1 & & & -1 & & & & & & & \\
 & -1 & & & -1 & 4 & -1 & & & -1 & & & & & & \\
 & & -1 & & & -1 & 4 & -1 & & & -1 & & & & & \\
 & & & -1 & & & -1 & 4 & & & & -1 & & & & \\
 & & & & -1 & & & & 4 & -1 & & & -1 & & & \\
 & & & & & -1 & & & -1 & 4 & -1 & & & -1 & & \\
 & & & & & & -1 & & & -1 & 4 & -1 & & & -1 & \\
 & & & & & & & -1 & & & -1 & 4 & & & & -1 \\
 & & & & & & & & -1 & & & & 4 & -1 & & \\
 & & & & & & & & & -1 & & & -1 & 4 & -1 & \\
 & & & & & & & & & & -1 & & & -1 & 4 & -1 \\
 & & & & & & & & & & & -1 & & & -1 & 4
\end{bmatrix}
$$

We hide the zeros for simplicity.
This is a large, structured matrix.

A grid operator, “lower triangular” form

$$
\begin{tikzcd}
2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[d, "-1"'] \\
2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[d, "-1"'] \\
2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[r, "-1"] \arrow[d, "-1"'] & 2 \arrow[d, "-1"'] \\
2 \arrow[r, "-1"] & 2 \arrow[r, "-1"] & 2 \arrow[r, "-1"] & 2
\end{tikzcd}
$$

Note the arrows all go down and to the right.

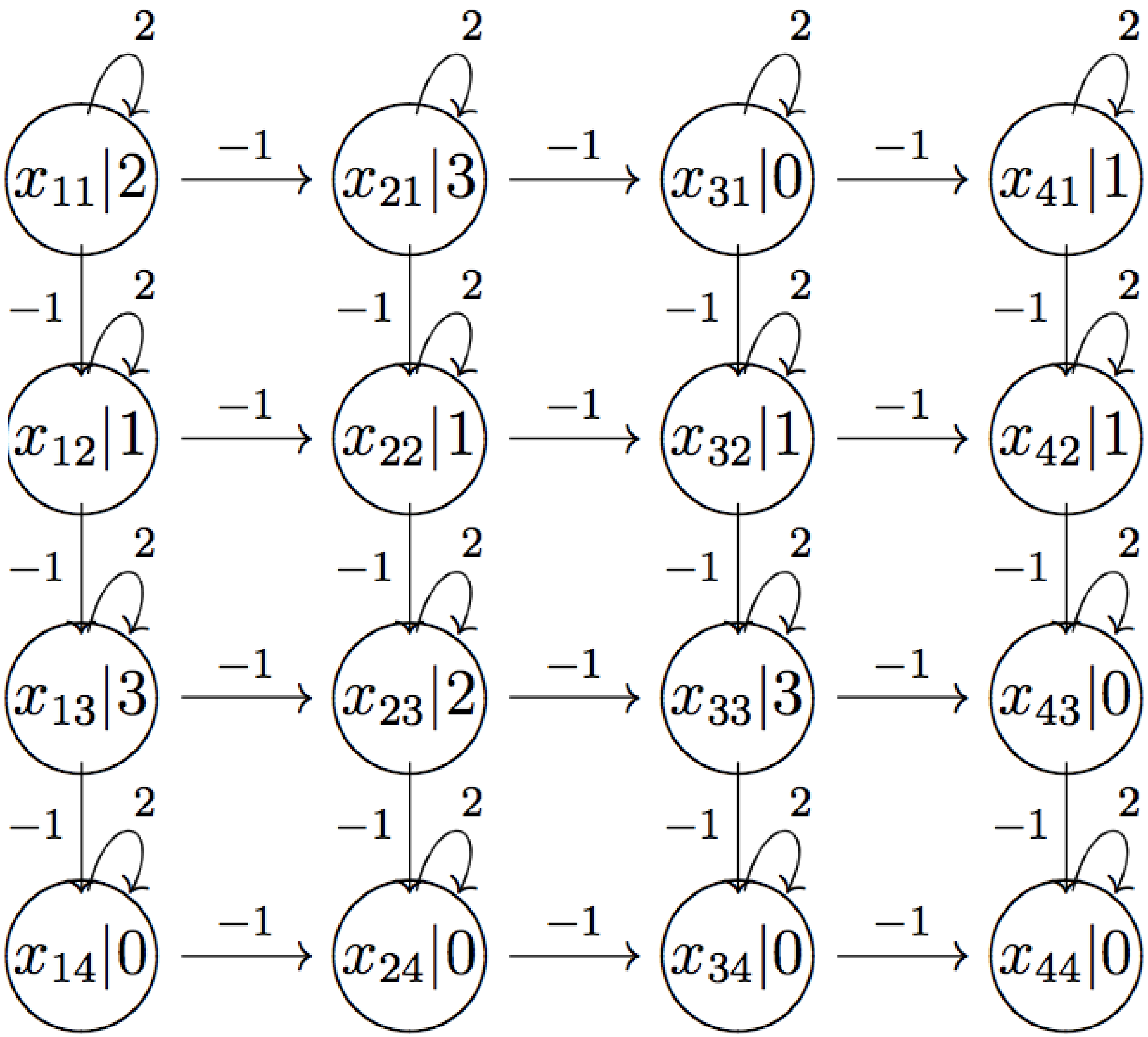
Solving triangular form, by back substitution

Start at upper left corner.

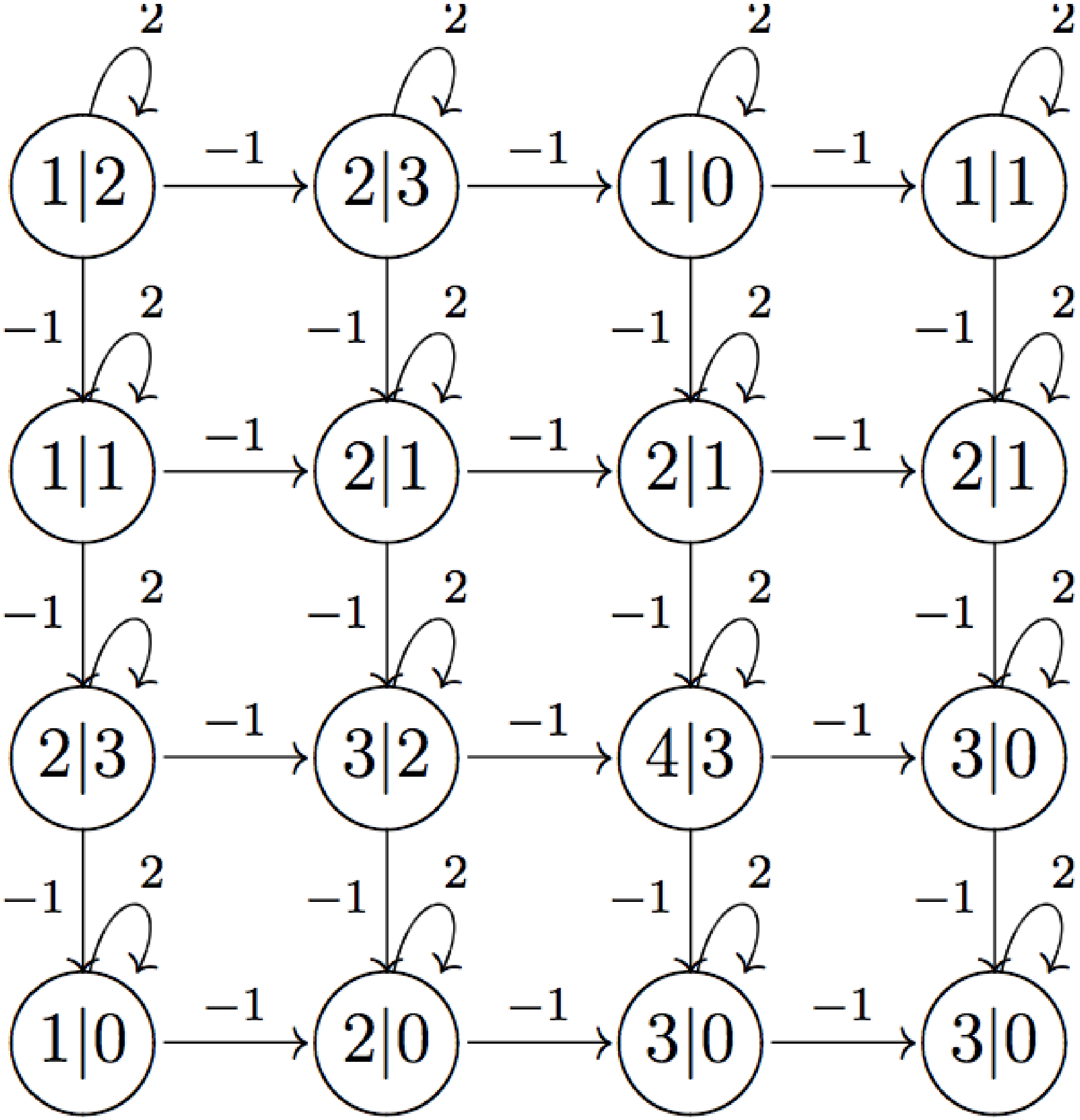
$$2x_{11} = 2$$

$$2x_{12} - x_{11} = 1$$

Etc.



Solved grid system



Linear algebra on a grid

- Nodes on grid index the rows and columns of a matrix
- Weights on arrows are matrix coefficients
- All arrows going towards one node is equivalent to a matrix row

- Matrix row operations correspond to operations on arrows
- Multiply row by a constant = multiply all arrow weights, pointing to one node
- Adding one row to another = add weights on arrows pointing to one node, to weights on arrows pointing to another node
- Exchanging rows = exchange arrows pointing to two nodes

Linear algebra on a grid

- Composing operators = chasing arrows, tails to heads, taking sum and products of weights
- Factoring operators = finding arrows, to give a composition
- Operators of special form are easy to invert, solve

Factoring the Laplacian

Analytic form:

$$\nabla^2 = (\frac{\partial}{\partial x})^2 + (\frac{\partial}{\partial y})^2 = (\frac{\partial}{\partial x} - i\frac{\partial}{\partial y})(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y})$$

Discrete form:

$$\nabla^2 = [\frac{(R-I)}{\Delta x} - i\frac{(U-I)}{\Delta y}][\frac{(I-L)}{\Delta x} + i\frac{(I-D)}{\Delta y}]$$

This factor has only right and up arrows.

This factor has only left and down arrows.

Both are solvable by back substitution.

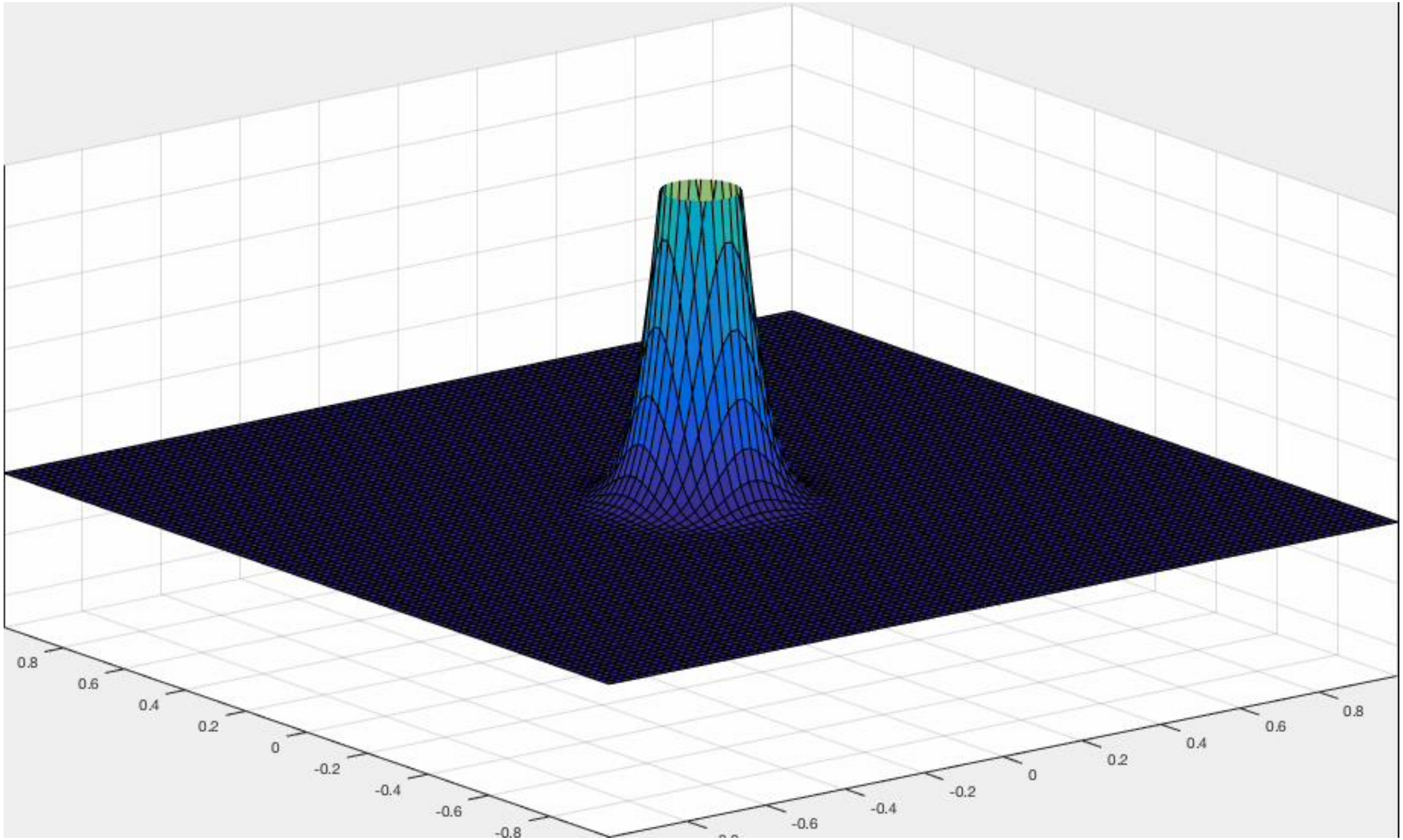
Grid factoring, Cholesky factorization

A better form:

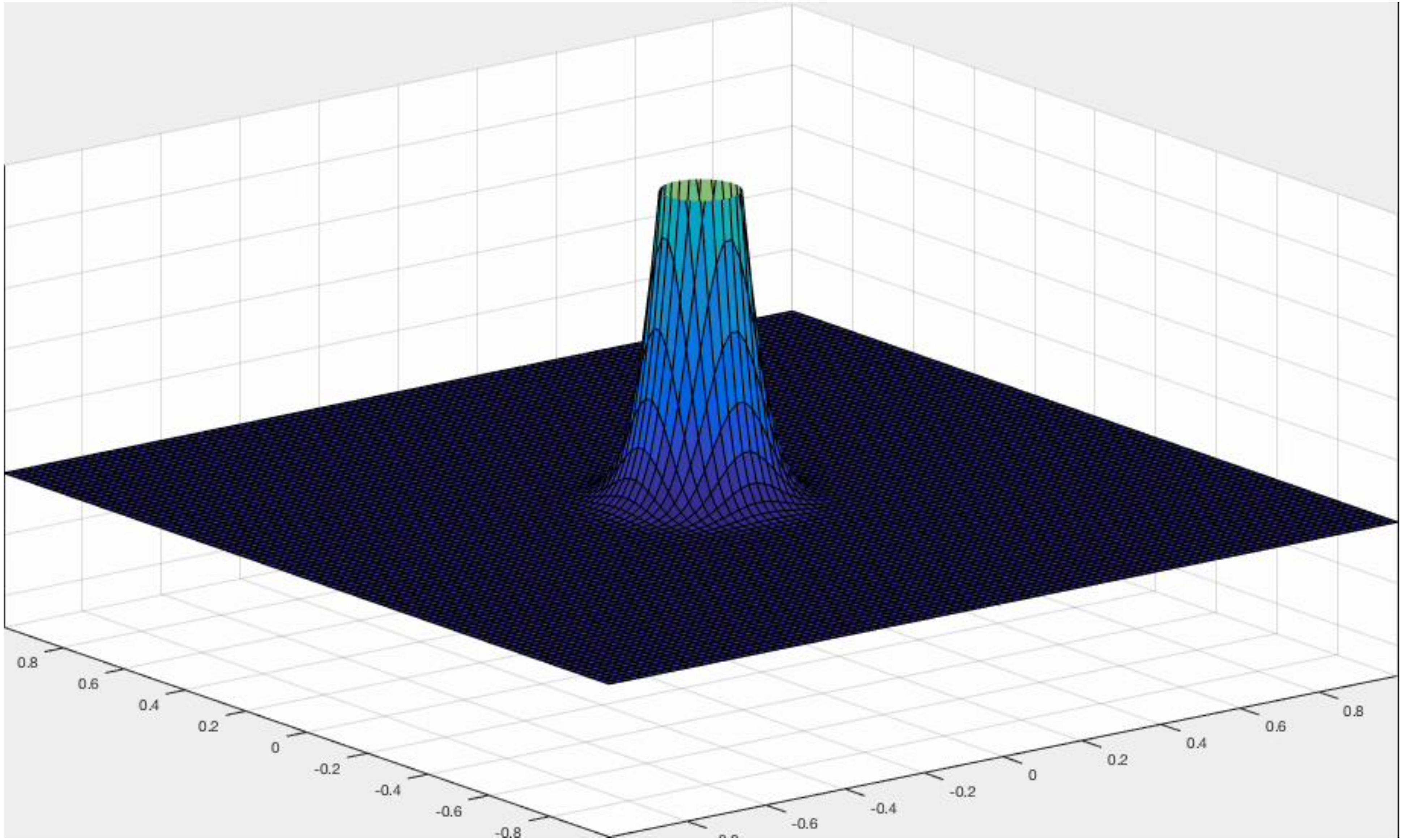
Cholesky factorization as bi-diagonal matrices in x, y directions.

$$\nabla^2 = -\left[\frac{Q_x^T}{\Delta x} - i\frac{Q_y^T}{\Delta y}\right]\left[\frac{Q_x}{\Delta x} + i\frac{Q_y}{\Delta y}\right];$$

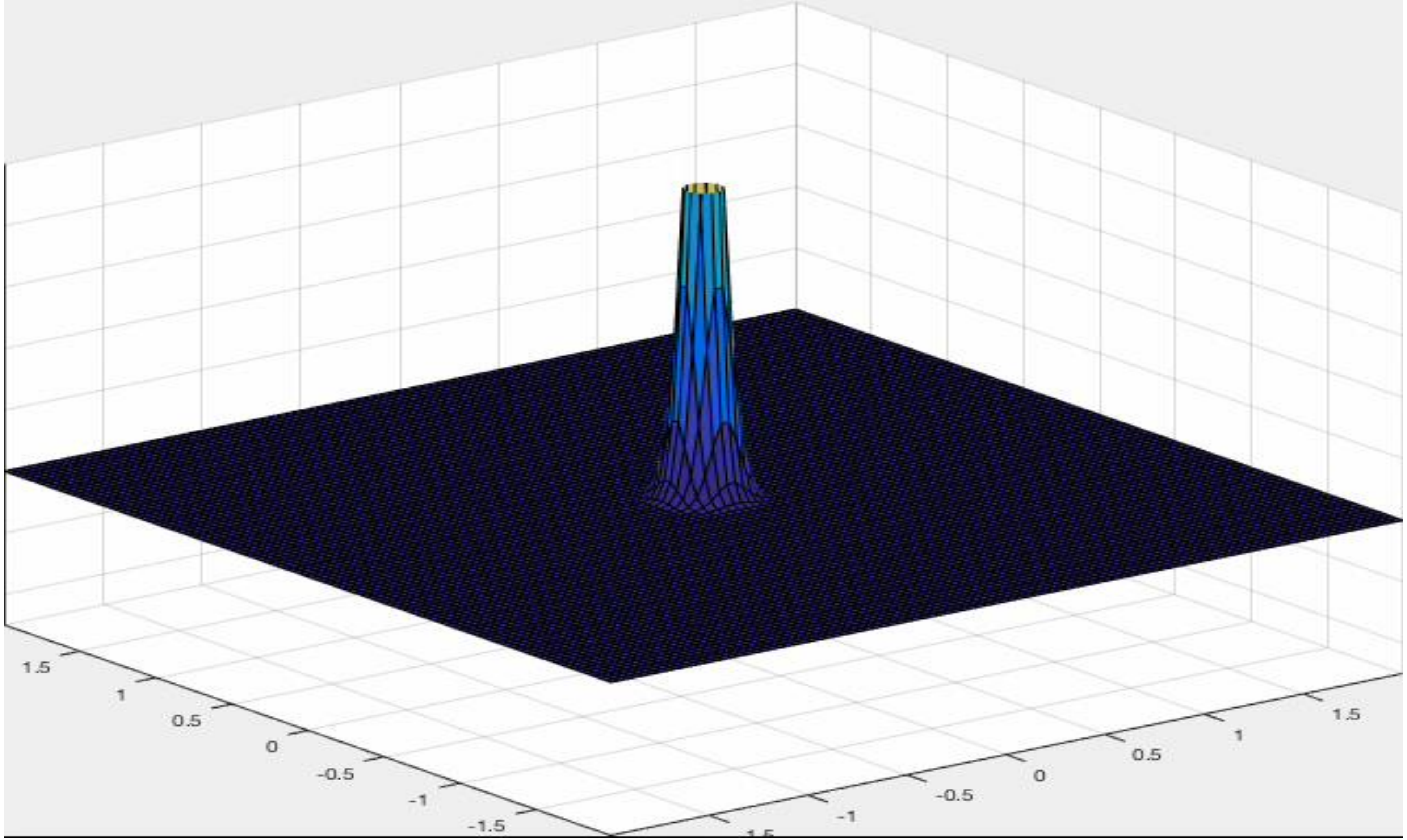
Coding example – explicit FD solution, grid algebra



Coding example – implicit FD solution, grid algebra



Sanity check on implicit – move out the x,y boundary



Conclusions

- Large, sparse matrices are a challenge in FD implicit methods.
- Linear algebra can be done directly on grid representations.
- Efficient representation of operators in computer memory.
- Early tests indicate this works, and is fast, for numerical solution of the wave equation.

Acknowledgements

- Thank you to:

- CREWES sponsors
- CREWES staff and students
- NSERC funding through CRD and Discovery grants

For financial and moral support.

Erroneous coding in paper (note 10^6 power)