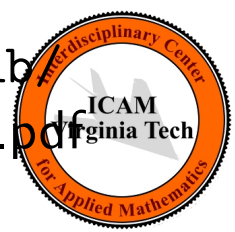
# How to make T when Ax=b: Putting Humpty Dumpty Back Together Again

John Burkardt
Mathematics Department
University of Pittsburgh

ICAM Conference on Applied and Computational Mathematics
Honoring Terry Herdman
Virginia Tech
10:30am, Wednesday

01 June 2022

T-puzzles by the Open Lab `https://teaching.pitt.edu/open-lab`
https://people.sc.fsu.edu/~jburkardt/presentations/t_puzzle_2022_vt.pdf
.....
.....



# An Inverse Problem: Humpty Dumpty

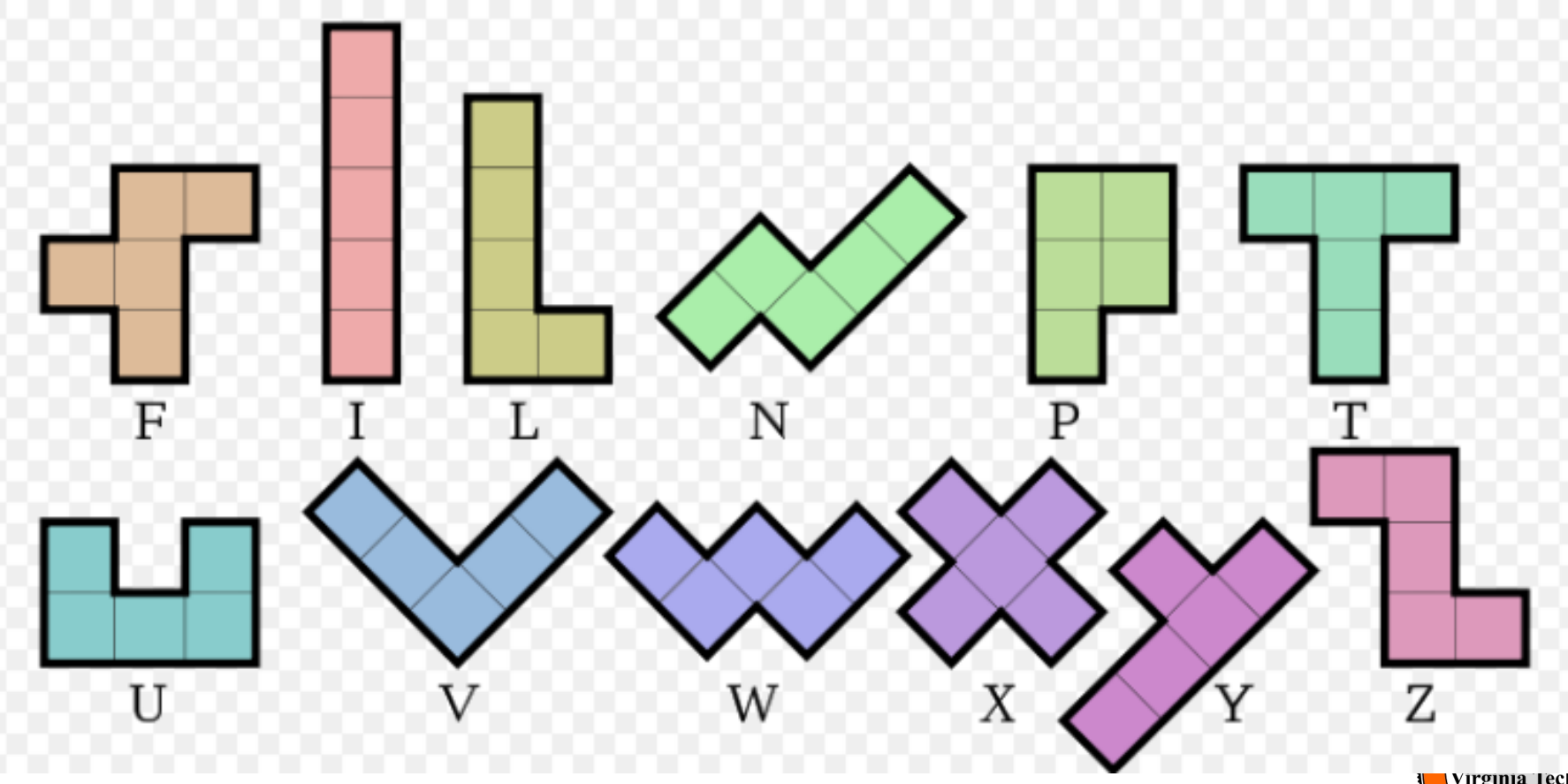
*Humpty Dumpty sat on a wall,*
*Humpty Dumpty had a great fall.*
*All the king's horses and all the king's men*
*Couldn't put Humpty together again.*

**Given pieces $b$ after transformation $A$, can we recover $x$?**

# The T Puzzle - (1800)

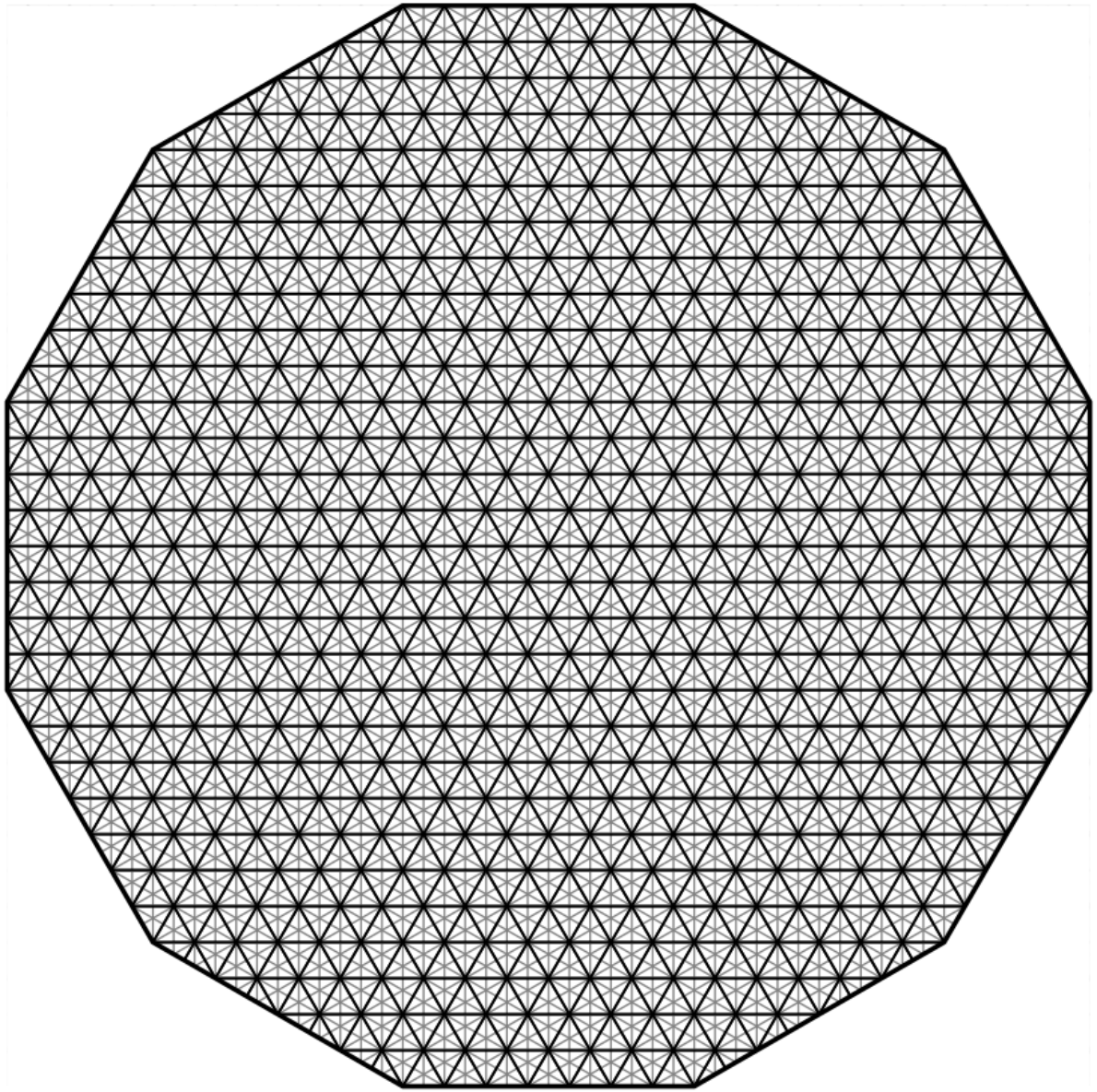
# Pentominoes & Polyominoes (Golomb, 1965)

Can we tile a 6x10 rectangle with the 12 pentominos?



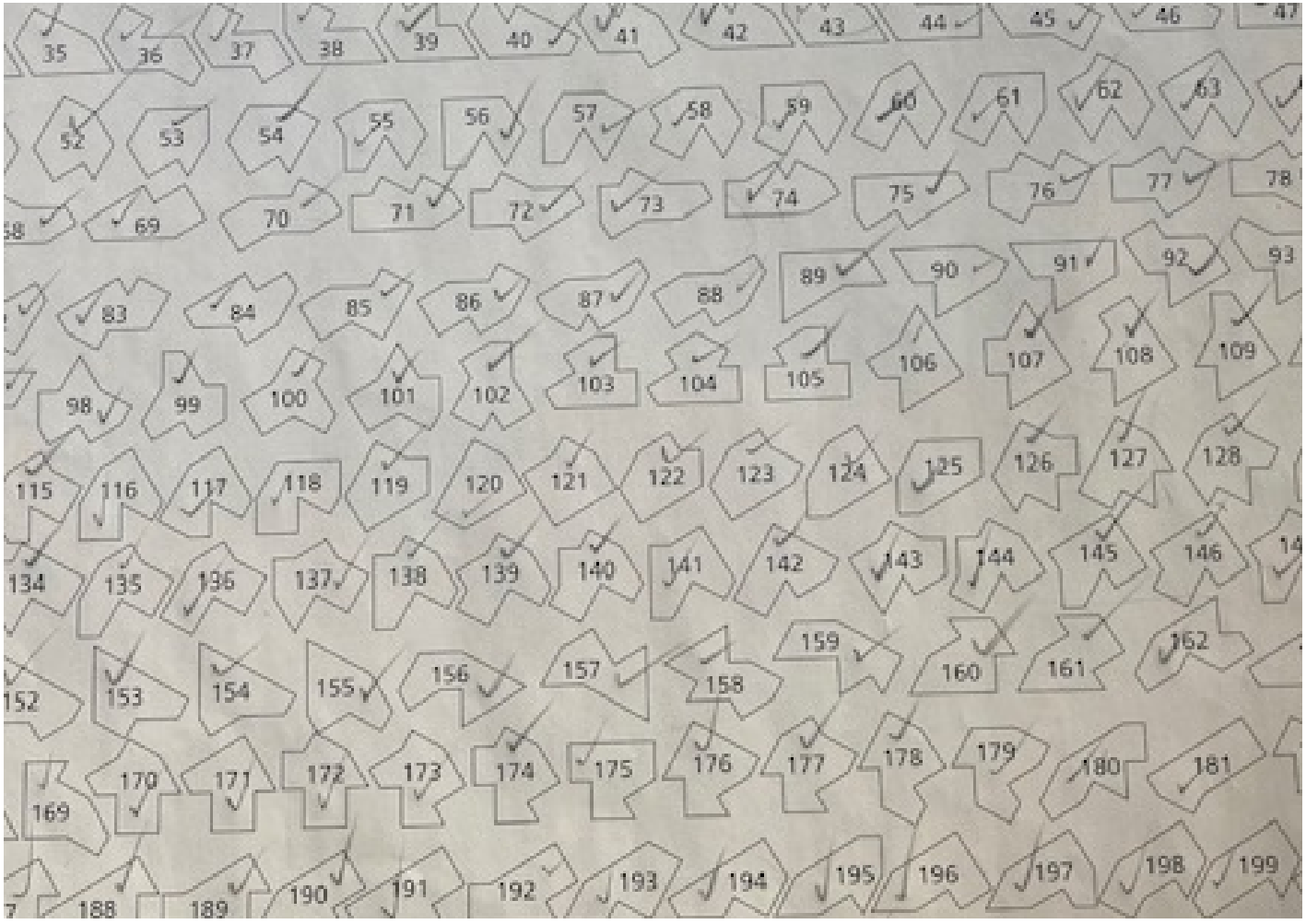
# Eternity - (Monckton, 1999)

Can we tile this shape of 7524 30-60-90 triangles using 209 tiles?



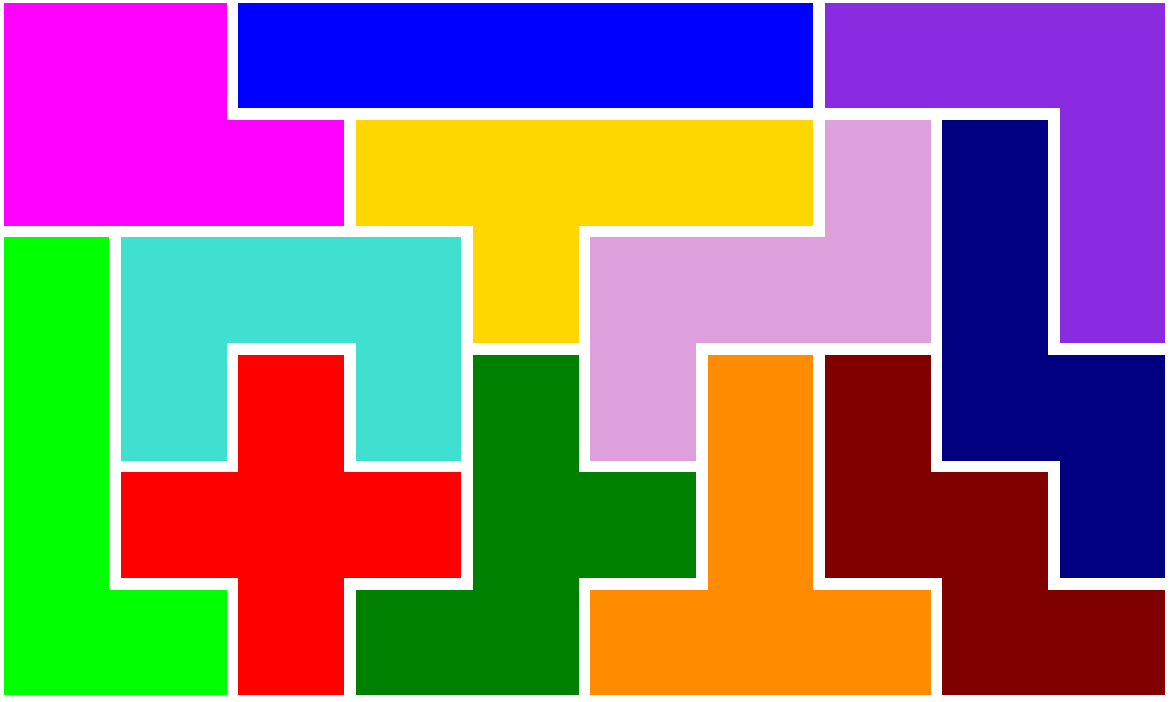
# Eternity - (Monckton, 1999)

We are to use 209 tiles, each formed of 36 30-60-90 triangles.



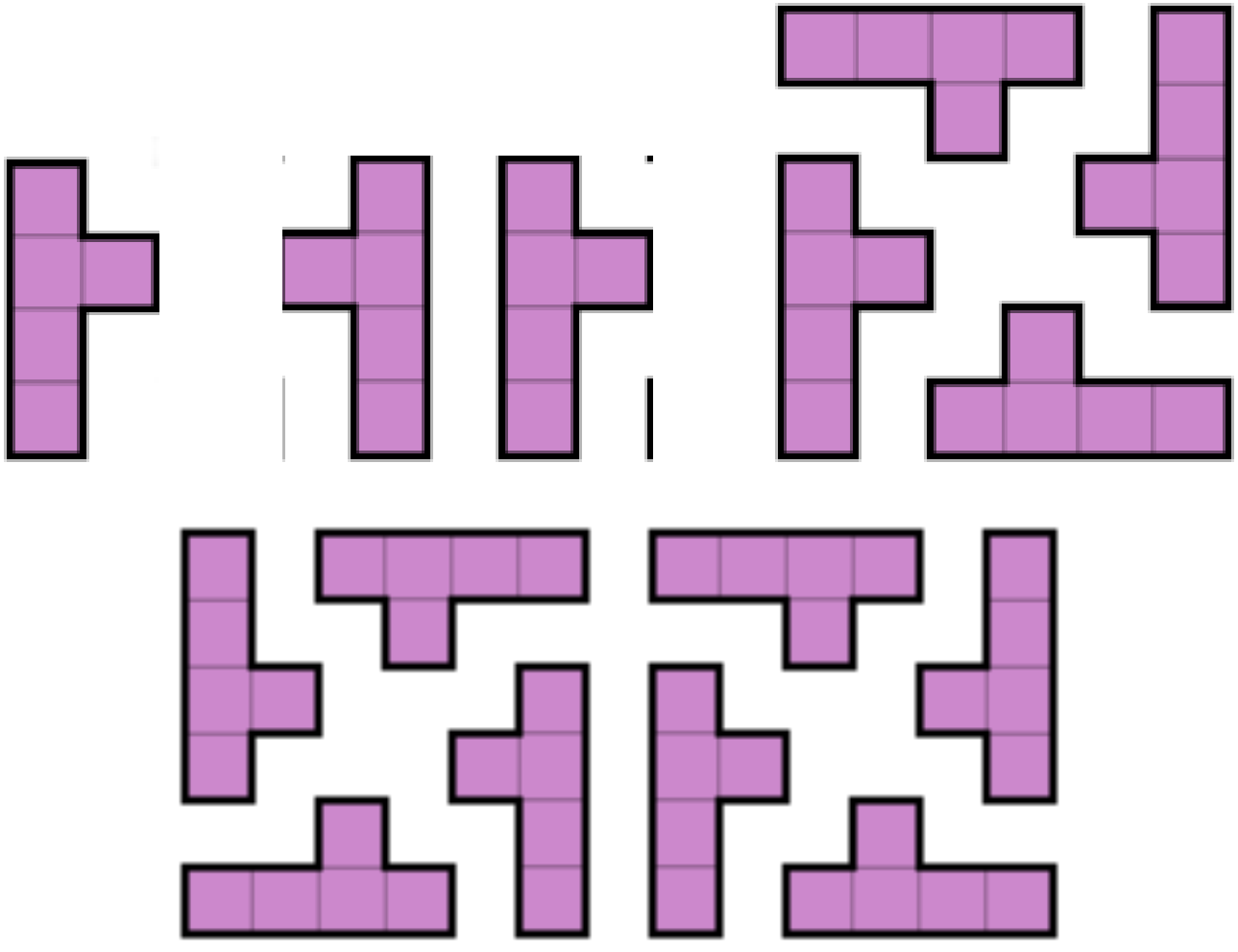
# OK, But Are Tiling Problems Math?

*"One can guess that there are several tilings of a* $6 \times 10$ *rectangle using the twelve pentominoes. However, one might not predict just how many there are. An exhaustive computer search has found that there are 2339 such tilings. These questions make nice puzzles, but are* **not the kind of interesting mathematical problem** *that we are looking for."*
"Tilings" - Federico Ardila, Richard Stanley



Interdisciplinary Center ICAM Virginia Tech for Applied Mathematics

# Tile : 2 Reflections: 4 Rotations = 8 Orientations



# Simultaneous Solution: Linear Algebra?

Linear Algebra can set multiple objects that satisfy multiple requirements.

# Tiling $\rightarrow$ Exact Cover

Donald Knuth: *"Tiling is a version of the exact cover problem."*

Select columns of a 0/1 matrix so every row has a single 1:

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

# Exact Cover $\rightarrow$ Linear Algebra

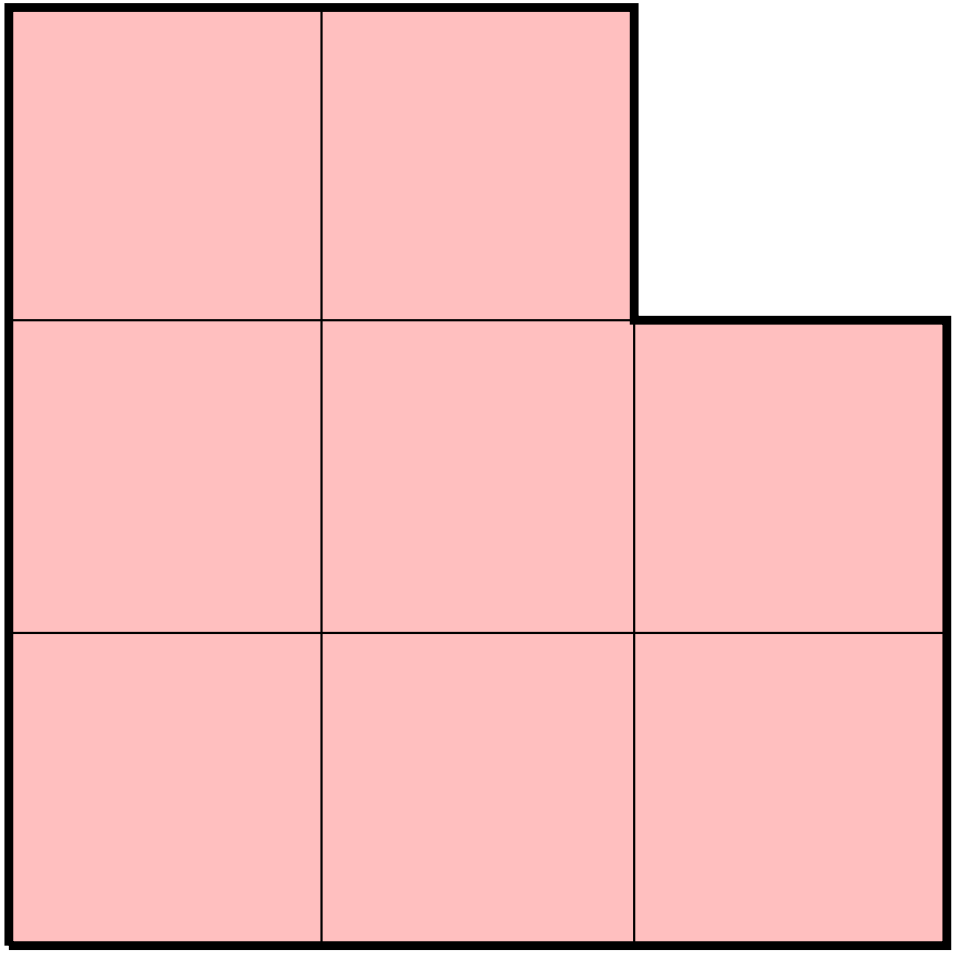
Select columns of A so every row has a single 1.

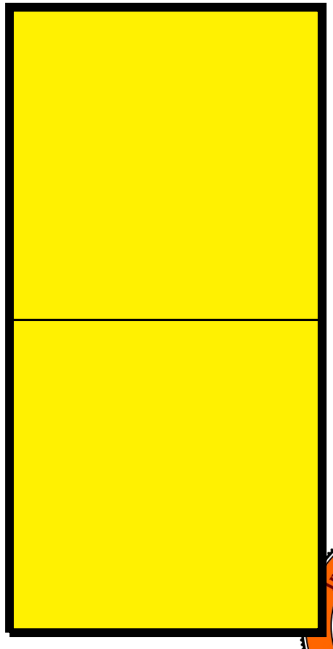
This is equivalent to
*Find x so that A*x=b where b is a vector of 1's.*

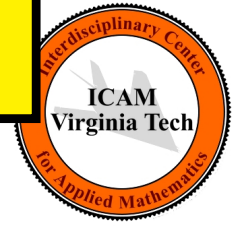
Now this is linear algebra!

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

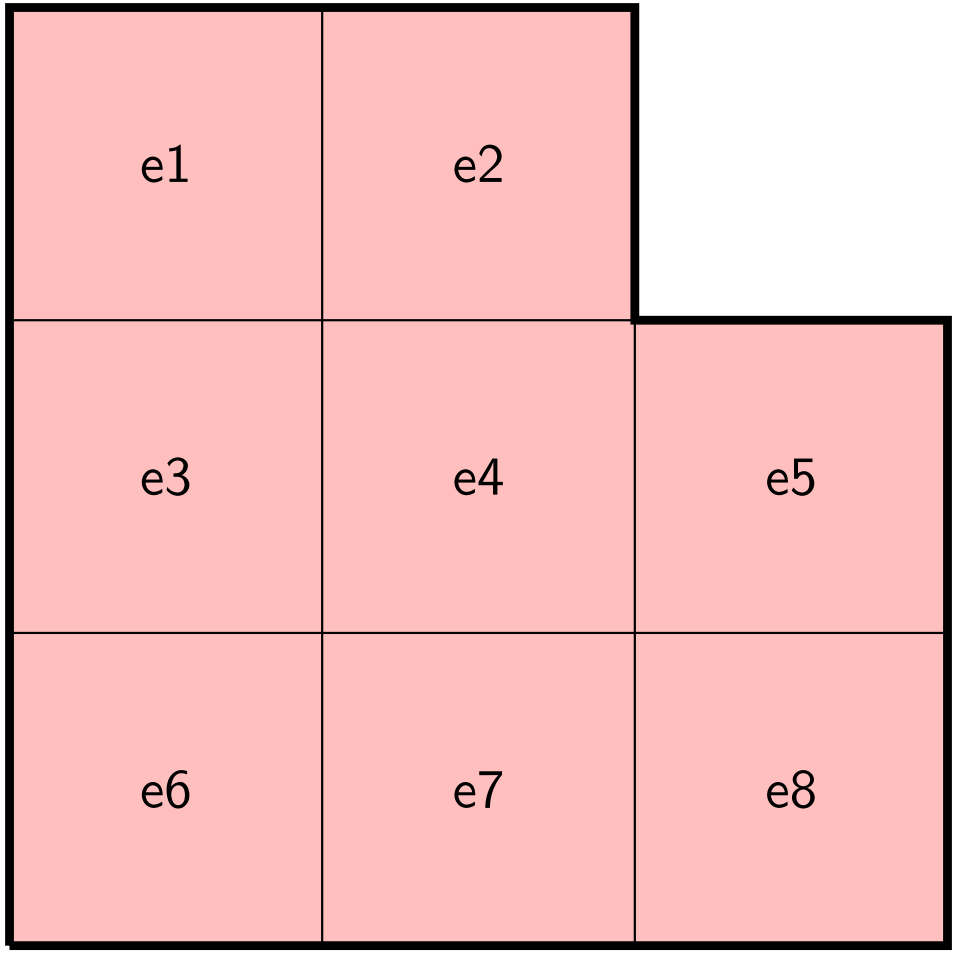
# Example: Tile the Reid Polygon with 4 Dominoes



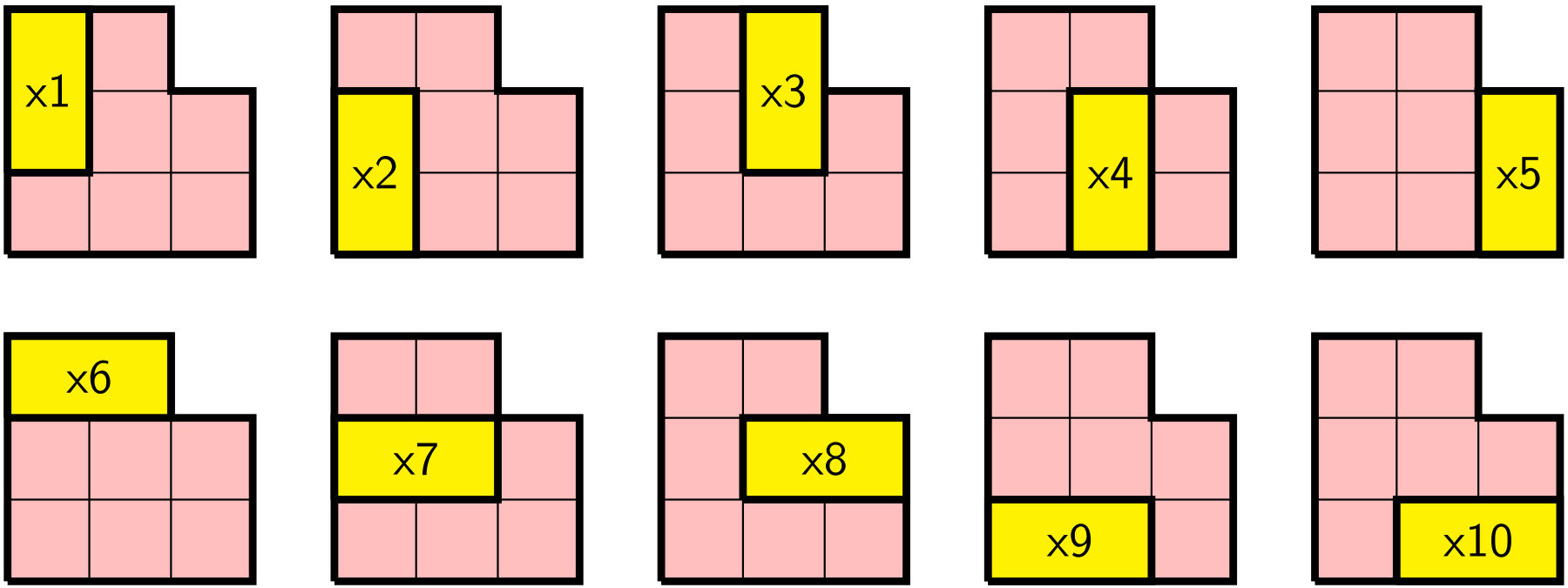




# The Reid Equations: $e_1 : e_8$



# The Reid Variables $x_1 : x_{10}$

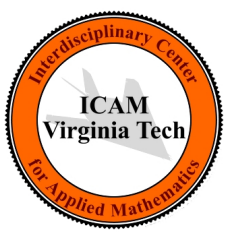



We start to see how the equations and variables combine:

$$x_1 + x_6 = 1 \text{ Cell 1 must be covered once}$$
$$x_3 + x_6 = 1 \text{ Cell 2 must be covered once}$$
$$x_1 + x_2 + x_7 = 1 \text{ Cell 3 must be covered once}$$
$$\ldots \quad \ldots$$

# The Reid Linear System: 8 Equations, 10 Unknowns

$$
\begin{array}{lrrrrrrrrrrcc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & & b \\
e_1: & x_1 & & & & & +x_6 & & & & & = & 1 \\
e_2: & & & +x_3 & & & +x_6 & & & & & = & 1 \\
e_3: & x_1 & +x_2 & & & & & +x_7 & & & & = & 1 \\
e_4: & & & x_3 & +x_4 & & & +x_7 & +x_8 & & & = & 1 \\
e_5: & & & & & x_5 & & & +x_8 & & & = & 1 \\
e_6: & & x_2 & & & & & & & +x_9 & & = & 1 \\
e_7: & & & & x_4 & & & & & +x_9 & +x_{10} & = & 1 \\
e_8: & & & & & x_5 & & & & & +x_{10} & = & 1
\end{array}
$$



# The Reid Linear System: 8 Equations, 10 Unknowns

$$A\,x = b$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

# Reduced Row Echelon Form of Reid Linear System

Notice that variables 7, 9 and 10 are free!

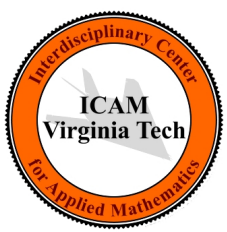
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ : | 1 | | | | | | 1 | | $-1$ | | $=$ | 0 |
| $e_2$ : | | 1 | | | | | | | 1 | | $=$ | 1 |
| $e_3$ : | | | 1 | | | | 1 | | $-1$ | | $=$ | 0 |
| $e_4$ : | | | | 1 | | | | | 1 | 1 | $=$ | 1 |
| $e_5$ : | | | | | 1 | | | | | 1 | $=$ | 1 |
| $e_6$ : | | | | | | 1 | $-1$ | | 1 | | $=$ | 0 |
| $e_7$ : | | | | | | | | 1 | | 1 | $=$ | 0 |
| $e_8$ : | | | | | | | | | | | $=$ | 0 |

Equation 8 disappears because once we have covered the first 7 cells, cell 8 is guaranteed to be covered.

# Drop Zero Row, Add Degrees of Freedom

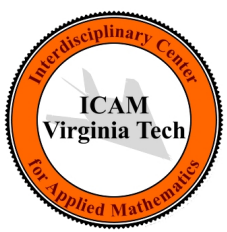
We add placeholder equations for variables 7, 9 and 10.

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ : | 1 | | | | | | 1 | | $-1$ | | $=$ | 0 |
| $e_2$ : | | 1 | | | | | | | 1 | | $=$ | 1 |
| $e_3$ : | | | 1 | | | | 1 | | $-1$ | | $=$ | 0 |
| $e_4$ : | | | | 1 | | | | | 1 | 1 | $=$ | 1 |
| $e_5$ : | | | | | 1 | | | | | 1 | $=$ | 1 |
| $e_6$ : | | | | | | 1 | $-1$ | | 1 | | $=$ | 0 |
| $f_1$ : | | | | | | | 1 | | | | $=$ | 0?/1? |
| $e_7$ : | | | | | | | | 1 | | 1 | $=$ | 0 |
| $f_2$ : | | | | | | | | | 1 | | $=$ | 0?/1? |
| $f_3$ : | | | | | | | | | | 1 | $=$ | 0?/1? |



# Solutions for 8 Binary Right Hand Sides

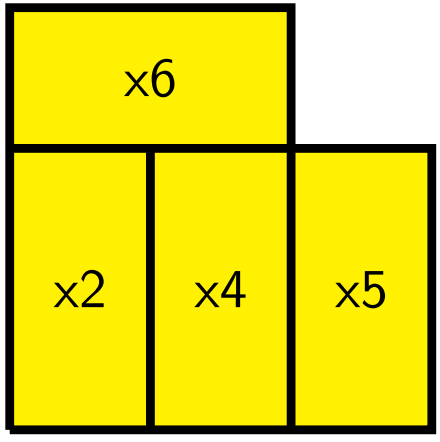
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $x_1$ : | 0 | 0 | 1 | 1 | $-1$ | $-1$ | 0 | 0 |
| $x_2$ : | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| $x_3$ : | 0 | 0 | 1 | 1 | $-1$ | $-1$ | 0 | 0 |
| $x_4$ : | 1 | 0 | 0 | $-1$ | 1 | 0 | 0 | $-1$ |
| $x_5$ : | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| $x_6$ : | 1 | 1 | 0 | 0 | 2 | 2 | 1 | 1 |
| $x_7$ : | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $x_8$ : | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $x_9$ : | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $x_{10}$ : | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

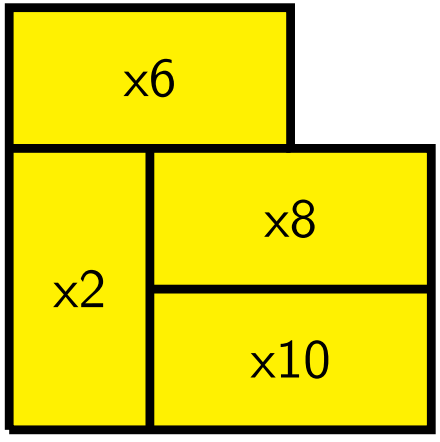


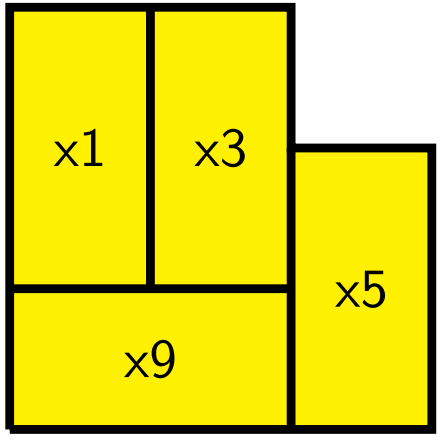
# 4 Acceptable Solutions (only 0 and 1 values)

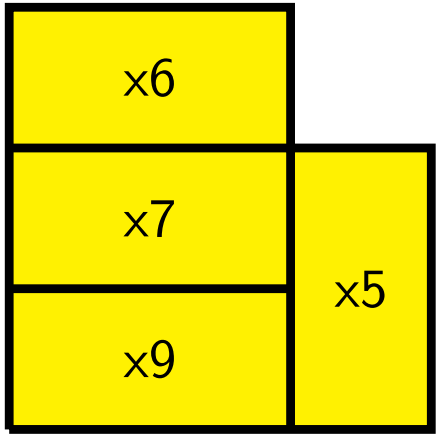
| | ✓ | ✓ | ✓ | × | × | × | ✓ | × |
|---|---|---|---|---|---|---|---|---|
| $x_1$ : | 0 | 0 | 1 | 1 | −1 | −1 | 0 | 0 |
| $x_2$ : | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| $x_3$ : | 0 | 0 | 1 | 1 | −1 | −1 | 0 | 0 |
| $x_4$ : | 1 | 0 | 0 | −1 | 1 | 0 | 0 | −1 |
| $x_5$ : | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| $x_6$ : | 1 | 1 | 0 | 0 | 2 | 2 | 1 | 1 |
| $x_7$ : | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $x_8$ : | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $x_9$ : | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $x_{10}$ : | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

# The Reid Tilings (Labeled)



# Bigger Problems Need a Better Solver

The Reid linear system $A * x = b$ was $M = 8$ equations in $N = 10$ unknowns. There were $S = 4$ copies of $T = 1$ tile type with $U = 2$ orientations. It was easy to write a code to reduce $A$ and $b$, via reduced row-echelon form; then to deal with the free variables, and then to eliminate solutions with unacceptable values. But for larger problems, this approach won't work.

- The row-reduced echelon form (RREF) is very sensitive to roundoff.
- We can't rely on MATLAB's `rref(A)` command, (real arithmetic).
- A "hand-made" integer code can only handle small problems.
- Tiling regions can have thousands of cells (equations/rows $= M$).
- Tiling problems can have hundreds of distinct tiles $= T$.
- Each distinct tile may have roughly $M * U$ configurations, (orientations followed by placement in grid) so variables/columns $N \approx T * M * U$.
- The linear system may have many degrees of freedom $D$.
- The number of possible solutions we need to check rises like $2^D$.
- To solve interesting problems, need accurate, efficient integer solver;

# Accurate & Efficient $A * x = b$ Solvers!

Solving underdetermined integer problems $A * x = b$ turns out to be an activity of enormous interest, especially in the linear programming community, in which the problem can include the request to optimize a corresponding cost function $\phi(x)$.

MATLAB Optimization Toolbox includes **optimvar()**.

Fast and efficient solvers are freely available: CPLEX, Gurobi, SCIP.

Moreover, the linear programming community uses a simple **LP** file format to describe such problems. So our task can be simplified:

- Set up the problem, write it to an LP file;
- Call an appropriate integer linear programming solver;
- Retrieve the solutions, count, plot, analyze;

# Cleve Moler's Online T Puzzler

ebook: https://www.mathworks.com/moler/exm/index.html

# The Humpty Dumpty Problem

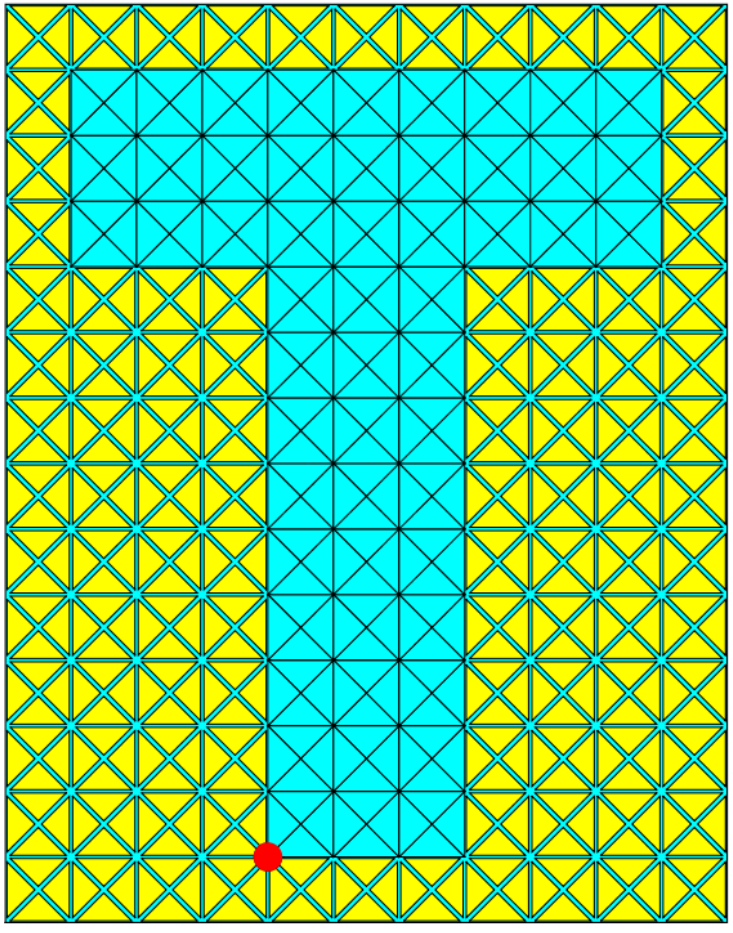
If the T puzzle is similar to the Reid problem, then presumably we can construct a procedure that will automatically find all possible solutions.

But:

- Each tile is a different shape;
- The tiles are not simple rectangular shapes;
- Some tiles have more reflections and rotations than others;
- A rectangular grid of cells won't work; we have diagonal lines too.
- The resulting linear system will be much larger than for Reid.

# T Puzzle Grid

To accommodate various rotations, reflections, translations of the tiles, we need a grid of $6 \times 9 \times 4$ isosceles right triangles: 6 big squares, each containing 9 little squares, each containing 4 triangles.



# Simulation Tile Placement

In order to automate the problem definition, we used a "boundary word" description of each tile, in a standard configuration, a string of letters N, S, E, W, NE, NW, SE, SW, that outline the shape.

Simple transformations of the boundary word correspond to rotation and reflection. Translation is also easy to model.

We must verify that each tranformation results in a configuration that is entirely inside the puzzle region.

# Linear System for T Puzzle

Now we have the tools we need to build the linear system $A * x = b$.

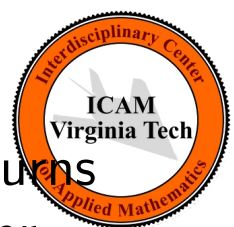
There are $6 \times 9 \times 4 = 216$ triangular elements. Each of them must be covered exactly once. There are 4 tiles, each must be used exactly once. This gives a total of 216+4=220 equations,

Each tile configuration (rotation + reflection + translation remaining in grid) is a variable. The tiles vary in their number of configurations:

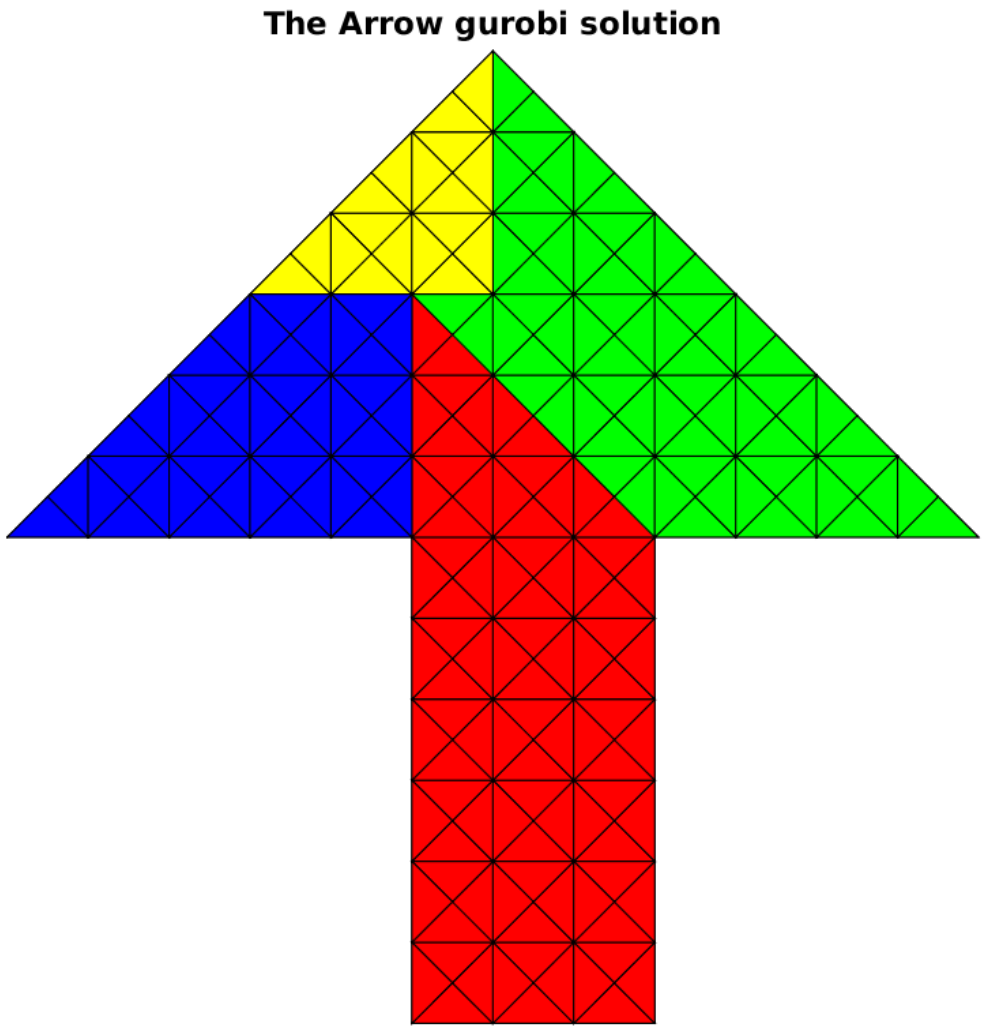
| | | |
|---|---|---|
| 1 | 20 | long |
| 2 | 2 | weird |
| 3 | 56 | trapezoid |
| 4 | 70 | triangle |
| Total | 148 | The T |

Solutions $x$ of our $220 \times 148$ linear system must be "binary".

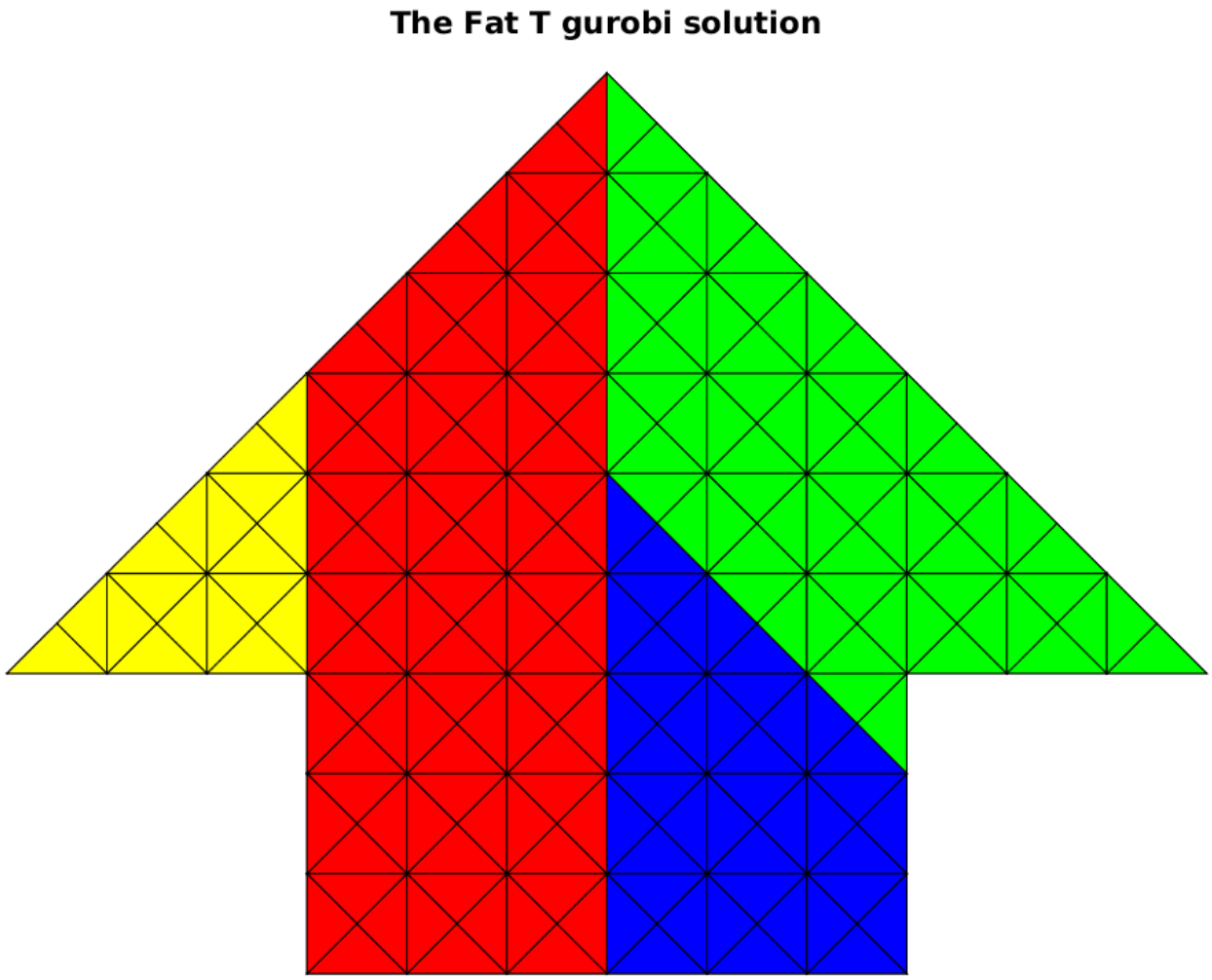
We write the linear system as an LP file. In a few seconds, Gurobi returns 2 possible solutions. One is simply the left/right reflection of the other.



# Arrow solution

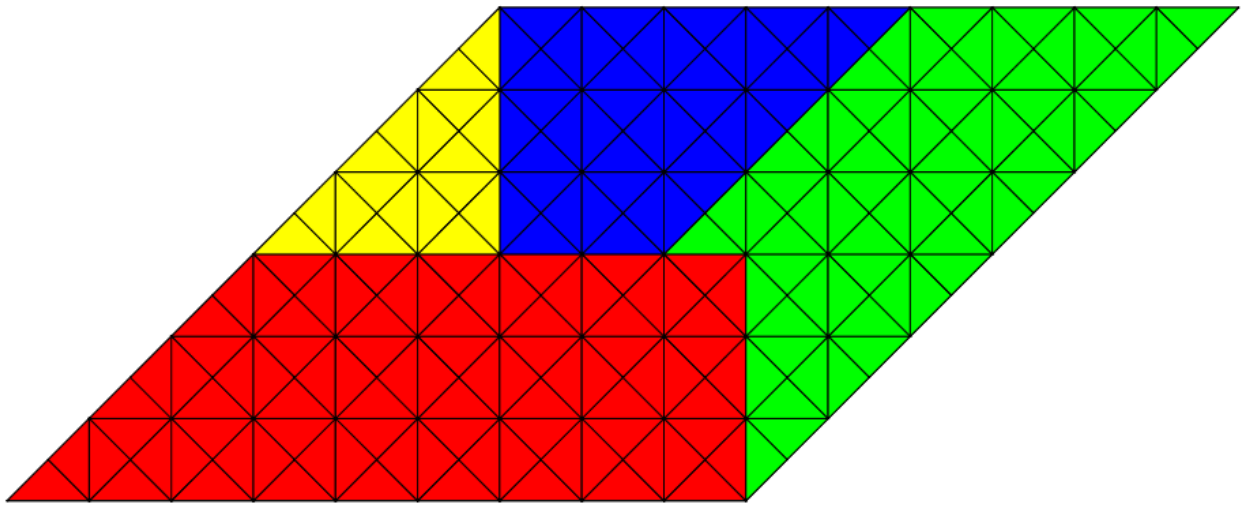


# Fat Arrow solution

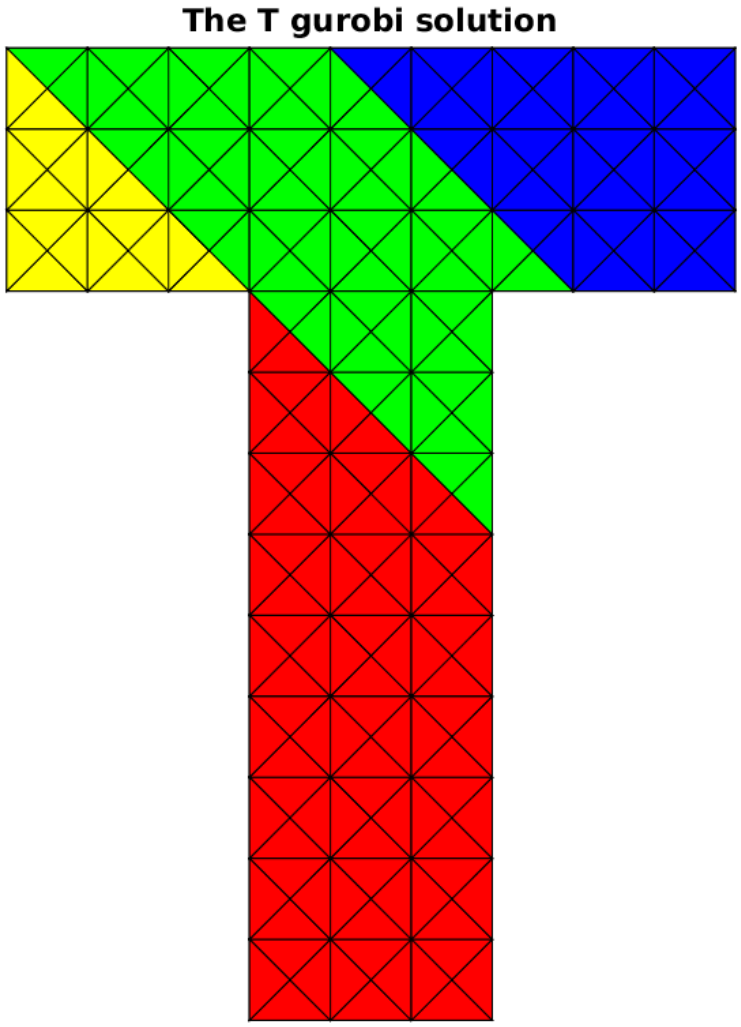


# Rhombus solution

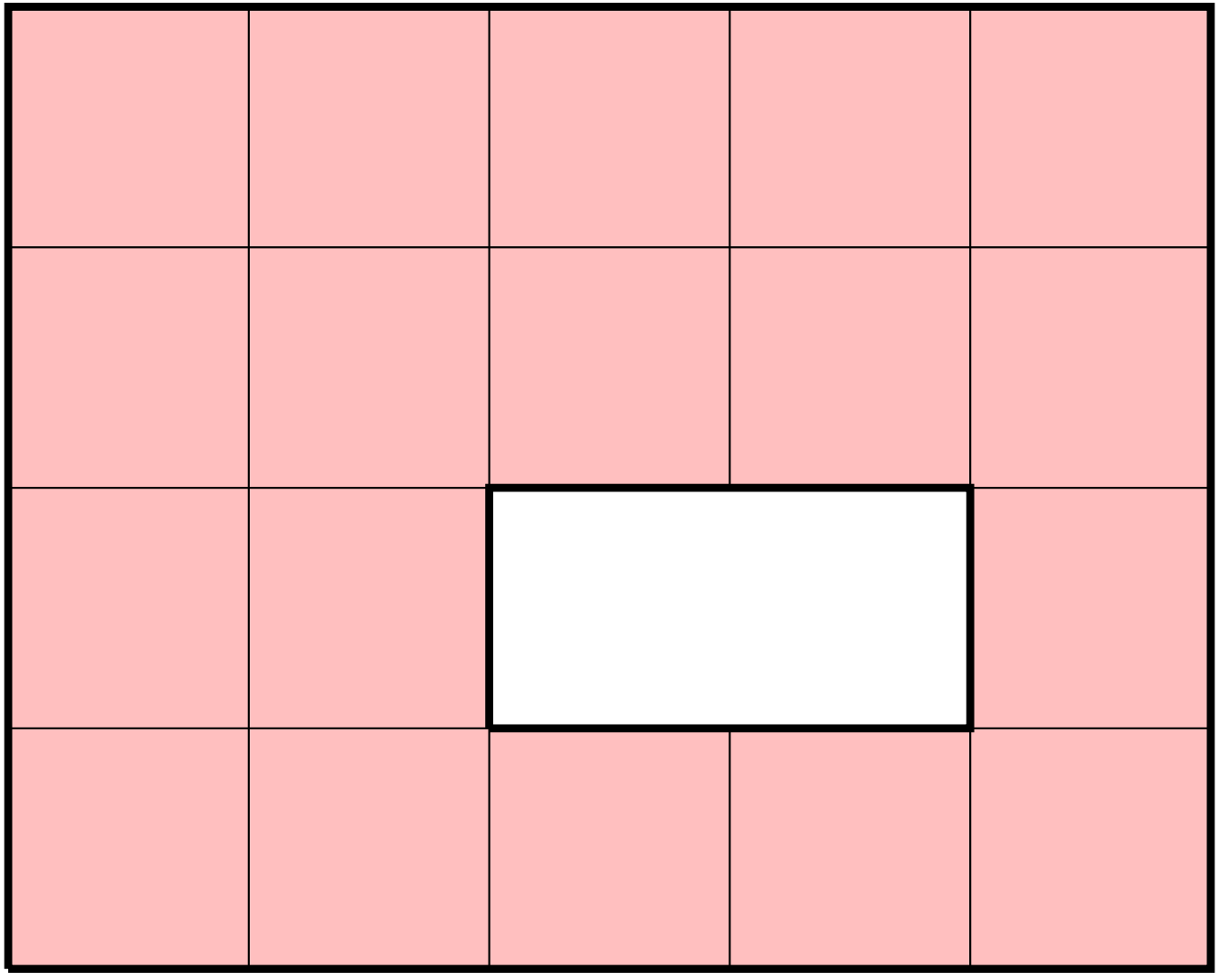
**The Rhombus gurobi solution**



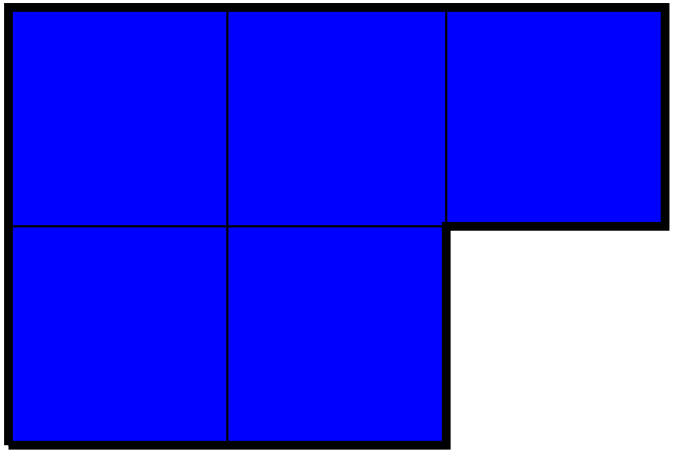
# T solution

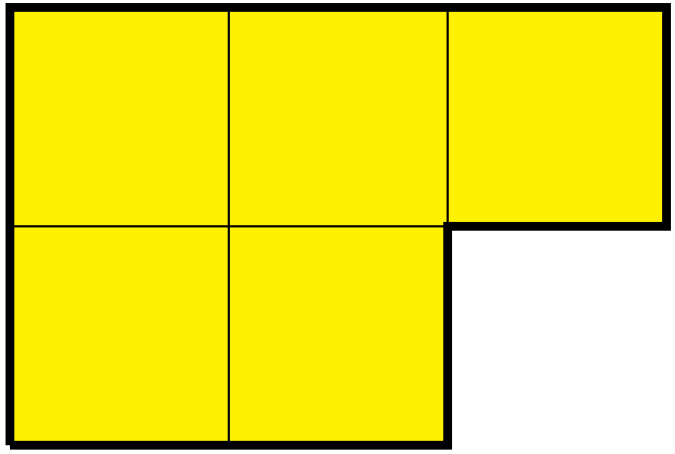


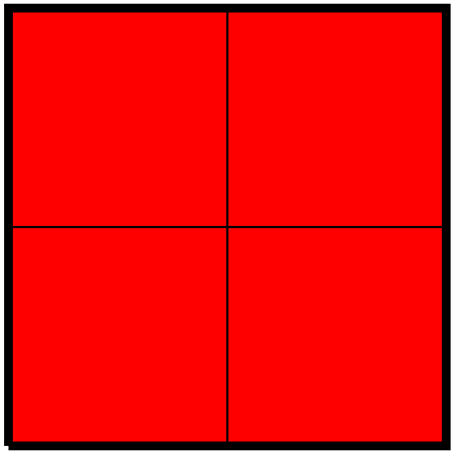
# 4x5 Example: Region R

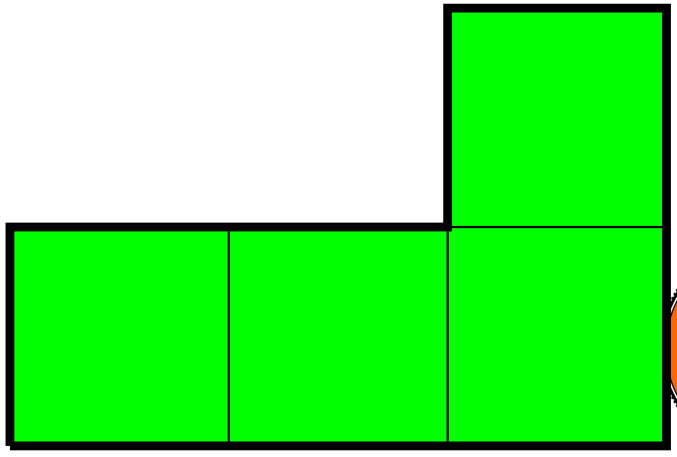


# 4x5 Example: Tiles



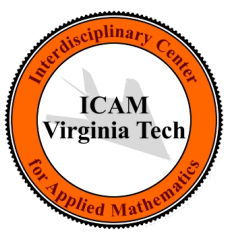




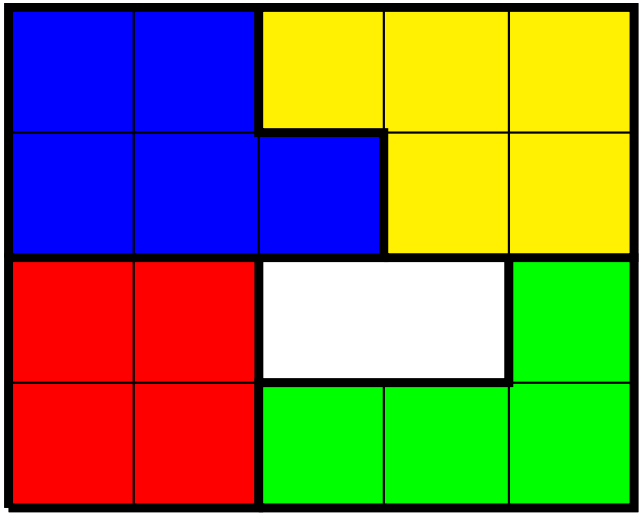


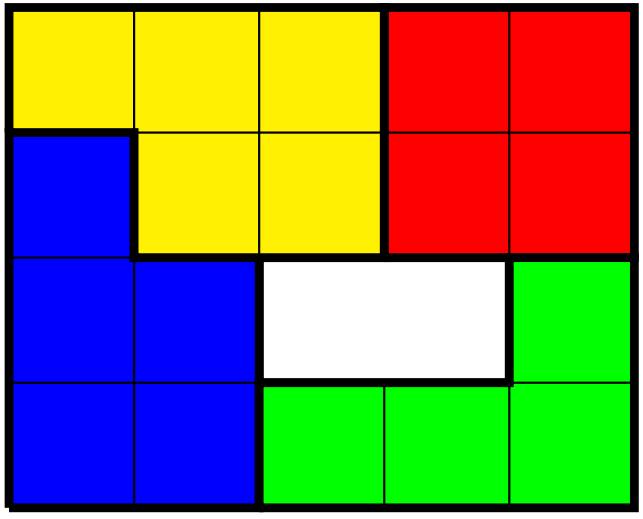
# 4x5 Example: Degrees of Freedom Go Crazy

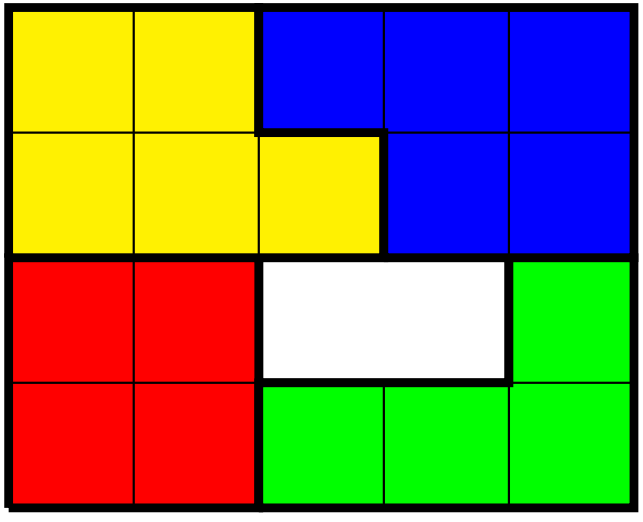
- RREF system has 23 rows and 62 columns;
- Augmented system has 42 degrees of freedom;
- ALL binary right hand sides is $2^{42}$, on the order of a **trillion**;
- Only check binary RHS with at most 4 degrees of freedom set to 1: $1 + 42 + 42 * 41/2 + 42 * 41 * 40/6 + 42 * 41 * 40 * 39/24 = 124,314$;
- Generated and solved all 123,314 right hand sides, found 4 binary solutions in less than 7 seconds.

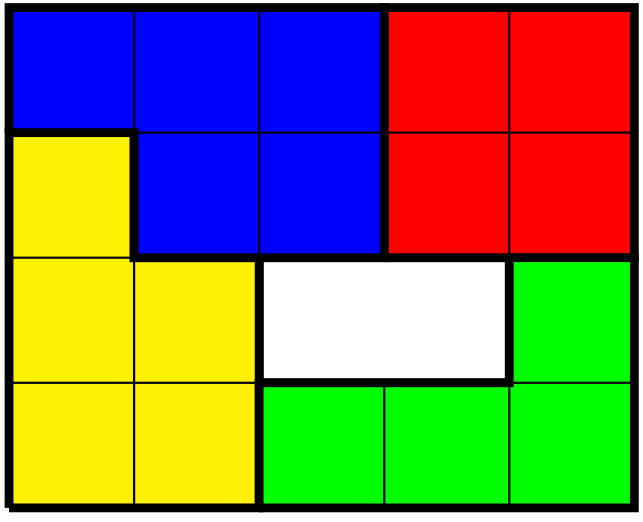


# 4x5 Example: Four Solutions

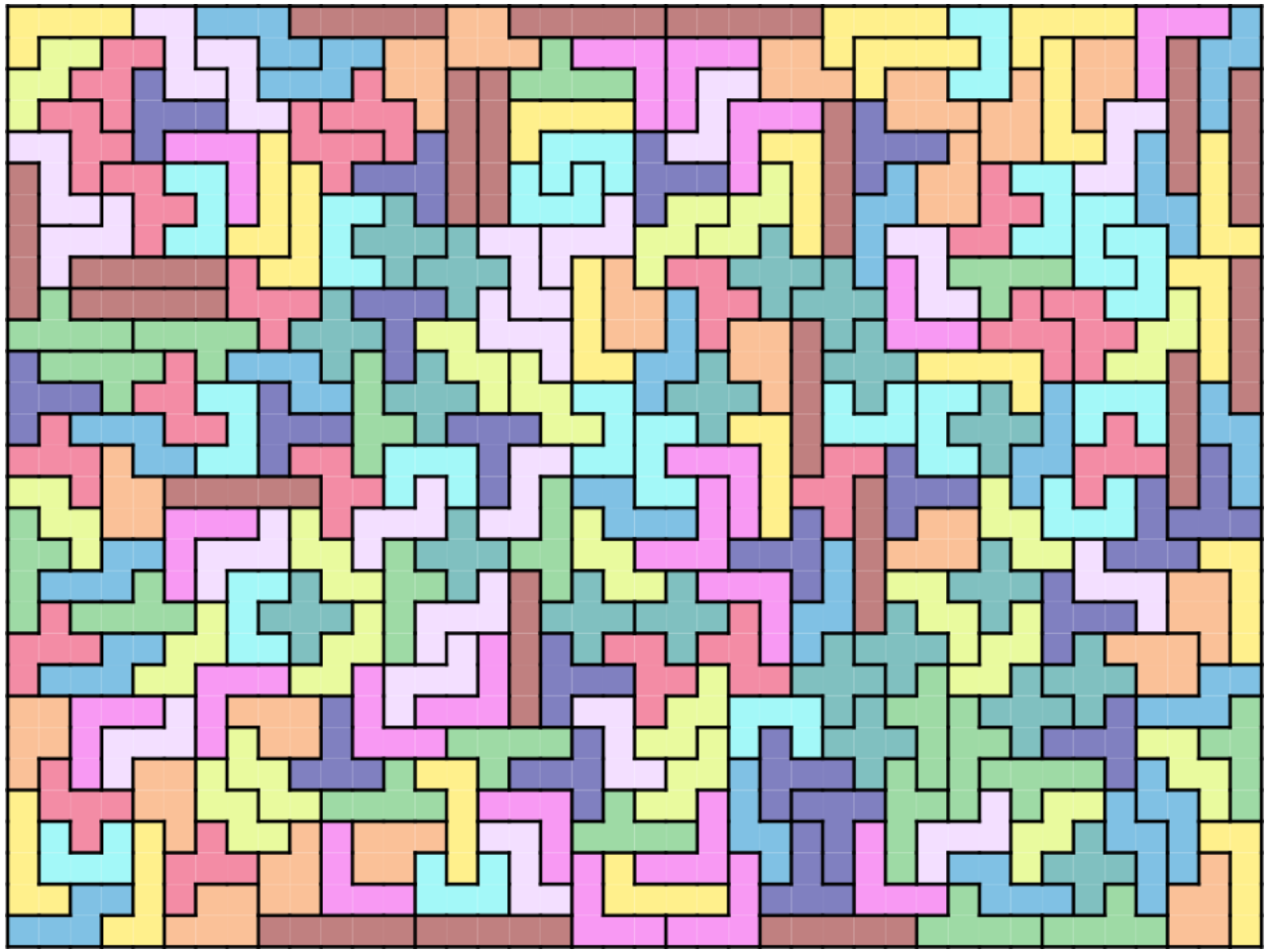








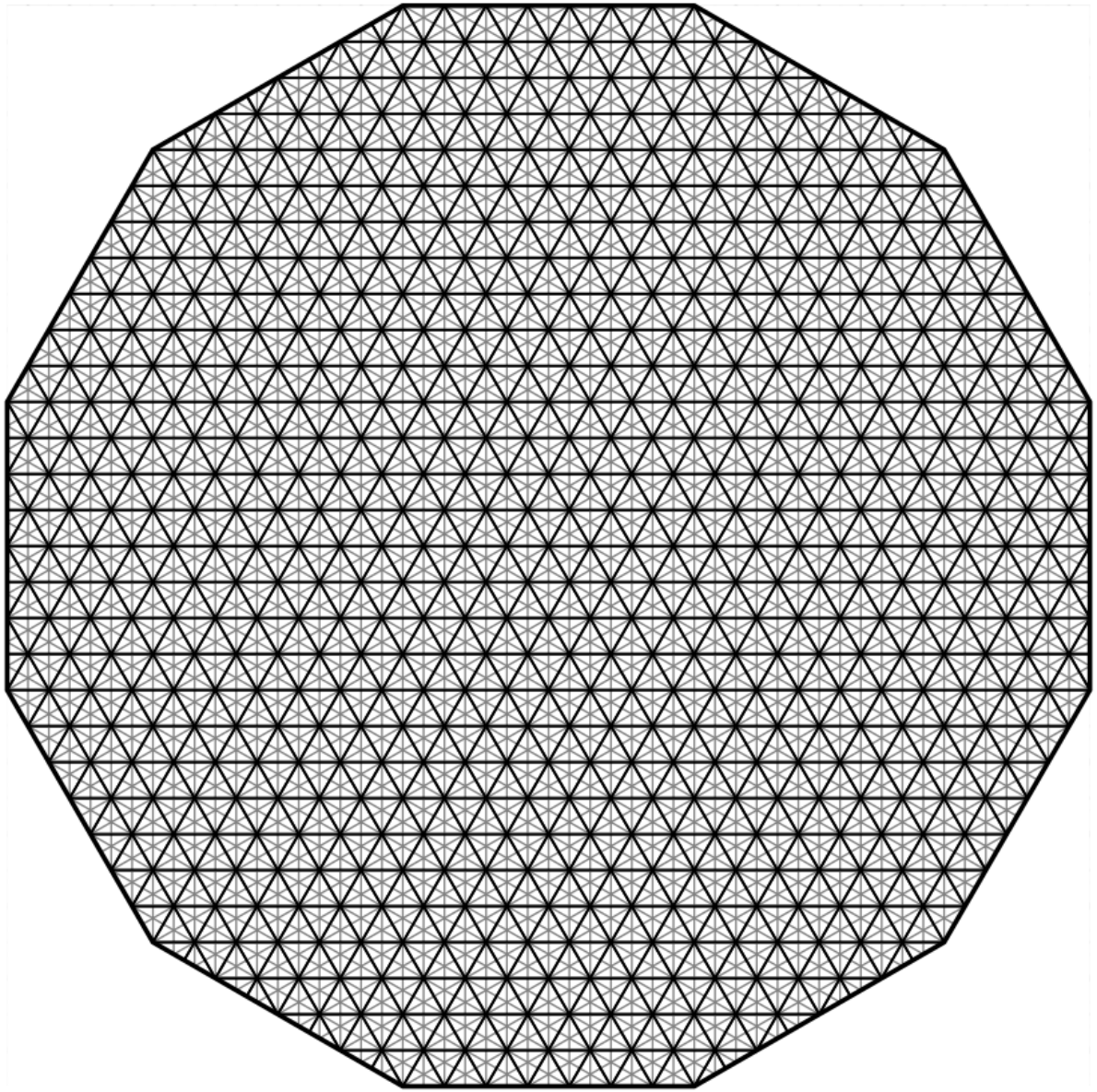
# A Large Pentomino Problem



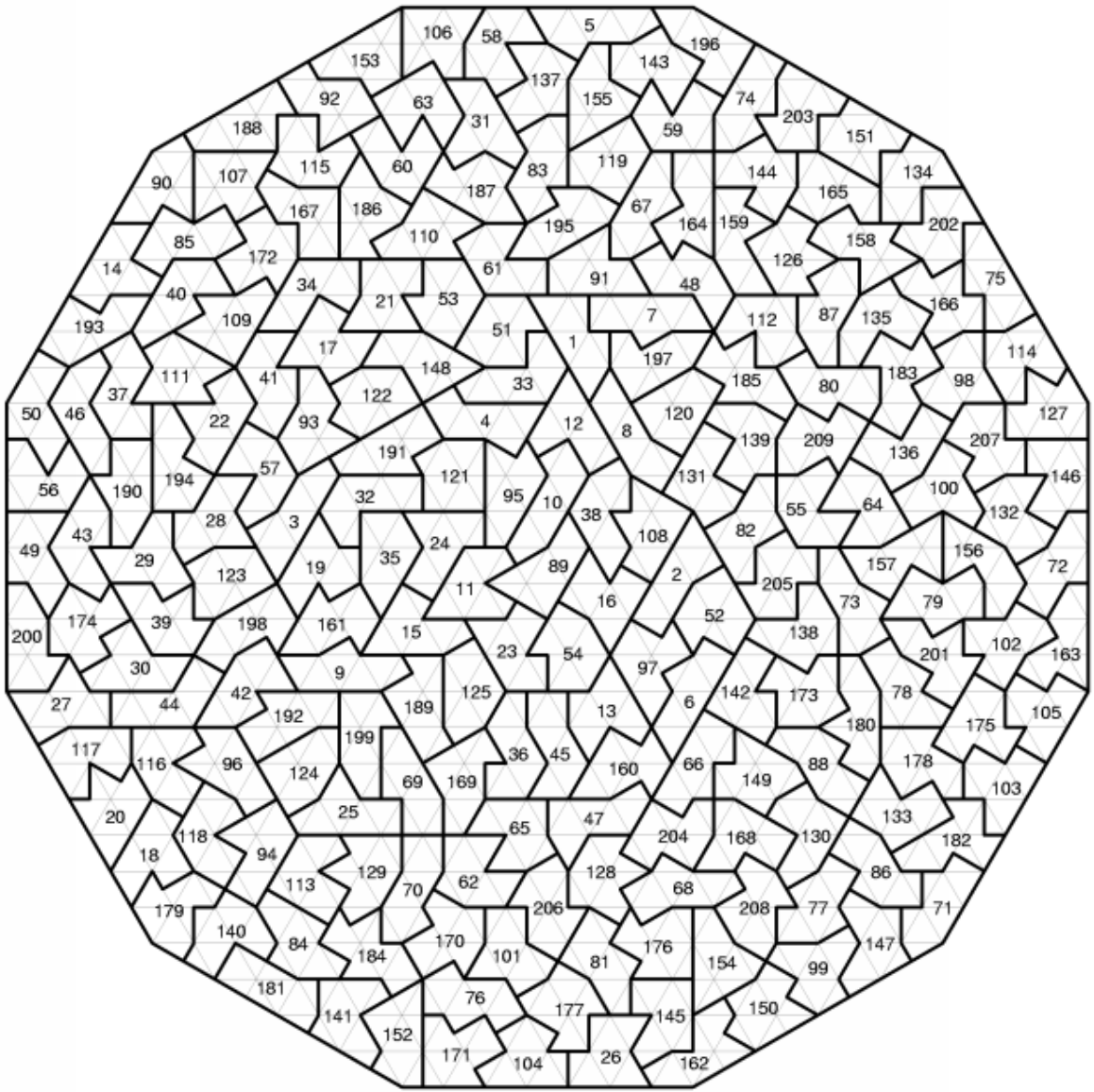
12 pentominoes, 20 copies each, 1,213 equations, 67,396 variables

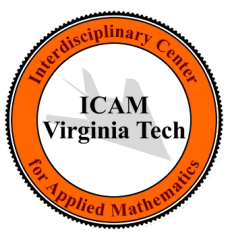
8 (equivalent) solutions computed by CPLEX in 9.5 minutes.

# Eternity - (Monckton, 1999)

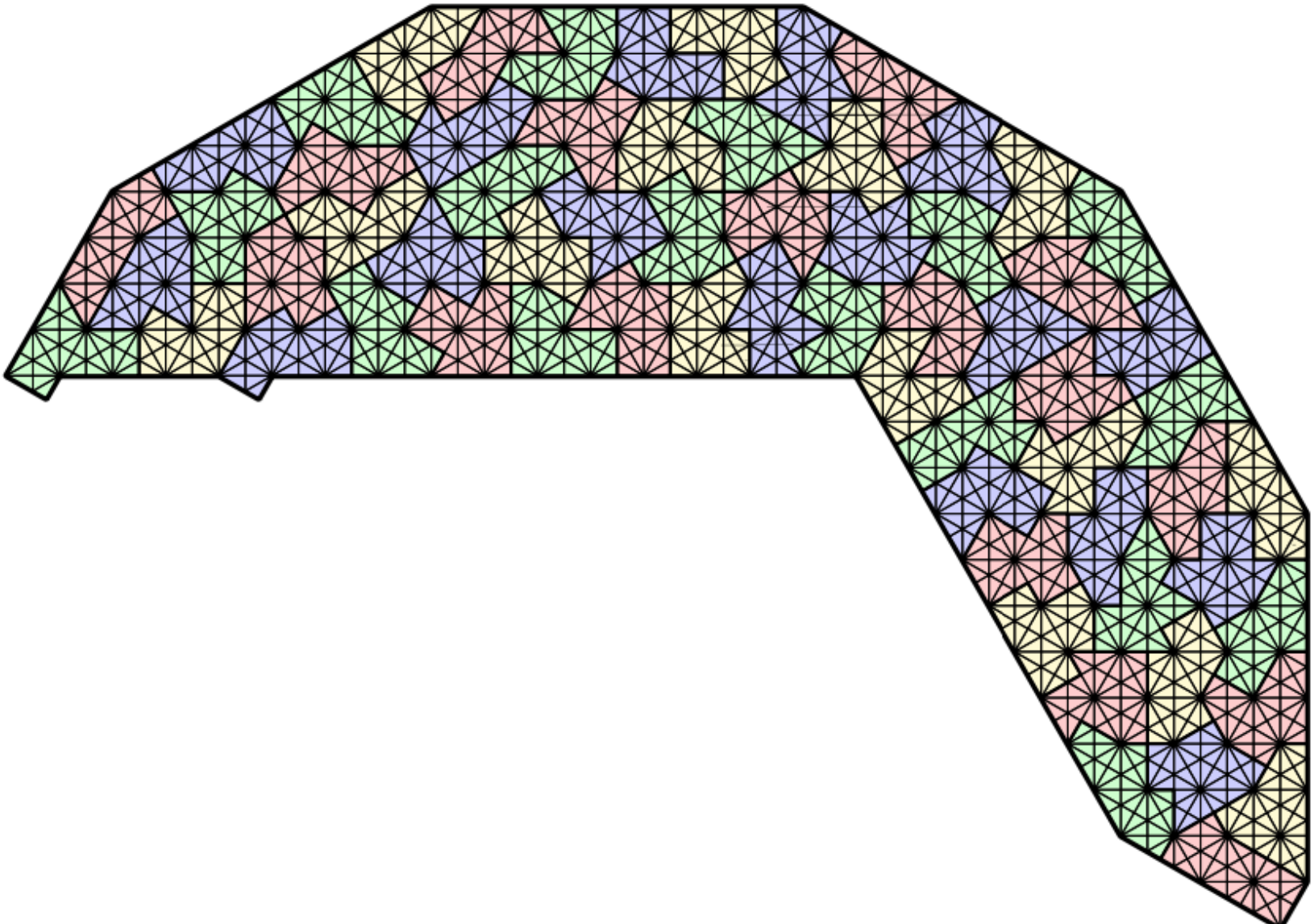


# Eternity - solved for £1,000,000 prize (Selby, 2000)

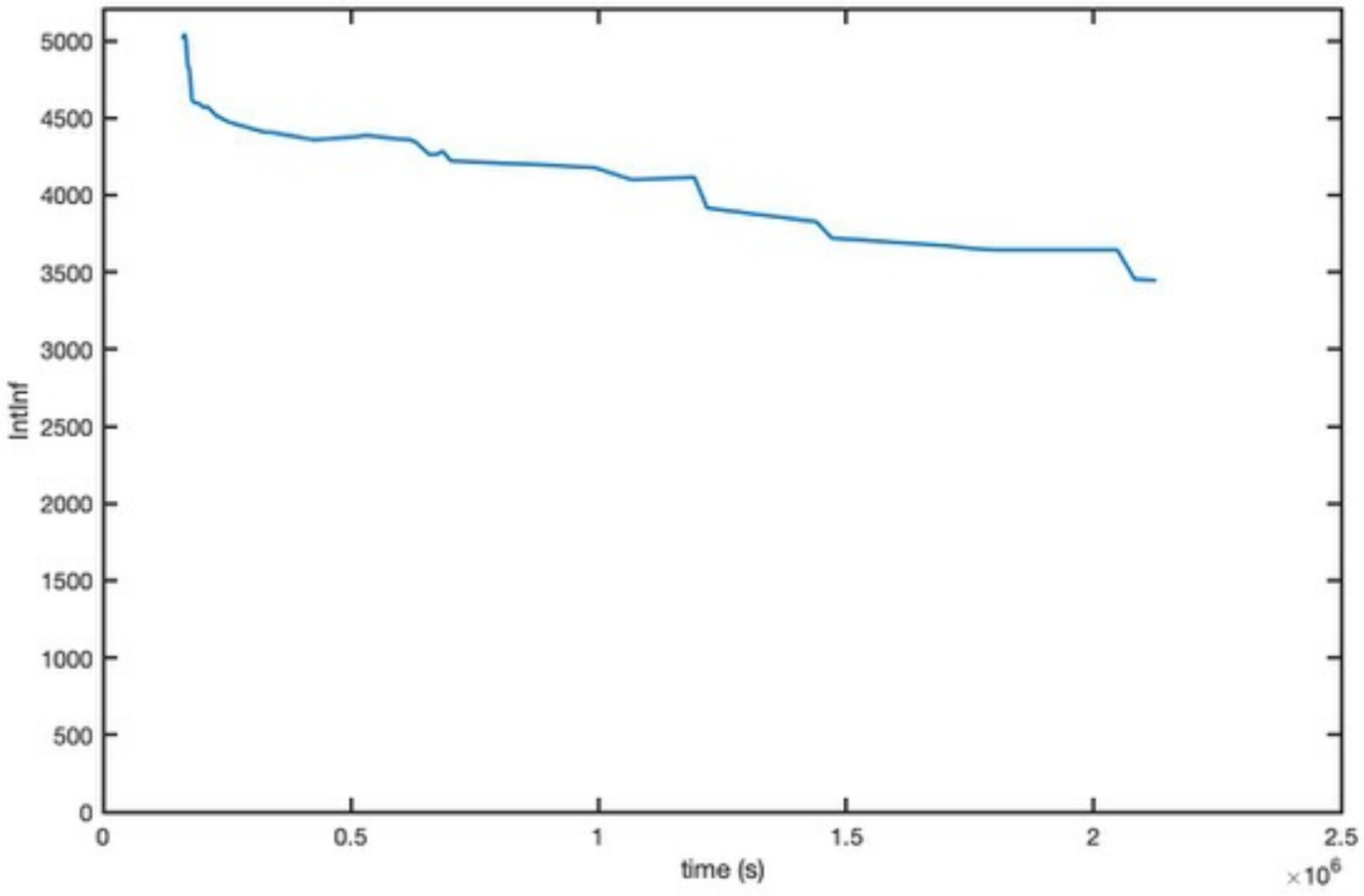


# Boomerang: A piece of Eternity (66/209 tiles)



# Trying the whole Eternity puzzle (209/209 tiles)



# Trying the whole Eternity puzzle (209/209 tiles)

We are running the Eternity problem on Gurobi.
Gurobi reduced our LP problem to a kernel of about 5,000 constraints.
While it works on them, it can display a plot of its progress.

1500 constraints done in 2,000,000 seconds
rate = 1333 seconds per constraint
5000 constraints * 1333 seconds = 6,665,000 seconds total.
6,665,000 seconds / 86400 seconds per day = 77 days.

Optimism: *Could get a positive result in 6 more weeks.*

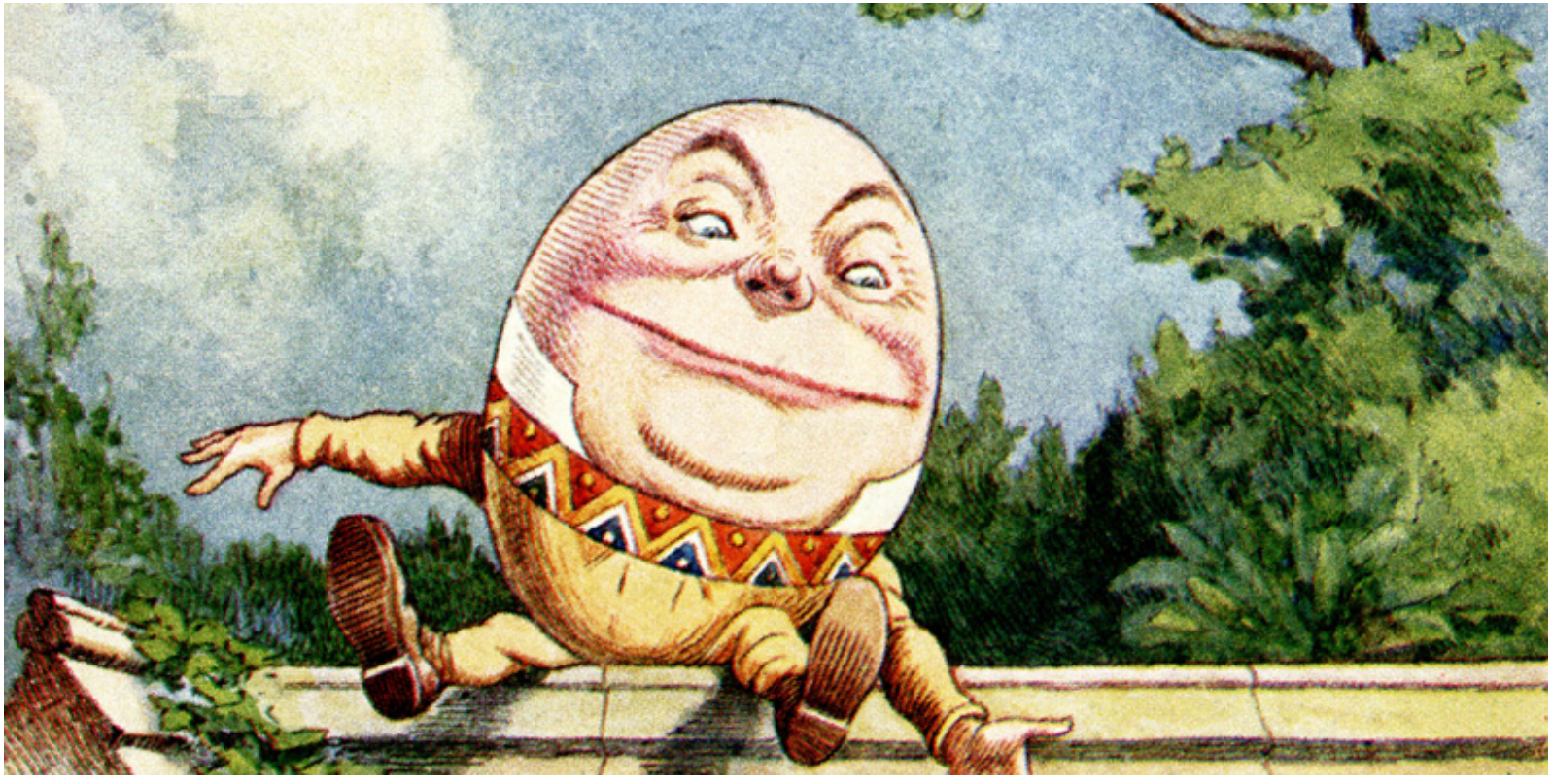
# Conclusion: Rebuilding with Linear Algebra

To solve a tiling problem, we look for an underlying grid of cells that define both the region and the tiles. *This isn't always possible!*

- Equations: Each region cell must be covered, just once.
- Equations: Each tile must be used, just once.
- Variables: Each rotated, reflected, translated tile remaining in region
- Equations + Variables: **underdetermined linear system** $Ax = b$.
- **Reduced Row Echelon Form** lets us analyze the system.
- **Linear Programming Software** solves big systems.
- We seek **binary** vectors $x$ whose entries are only 0 or 1.
- There may be no such solutions at all.
- If there are **free variables**, we may have multiple solutions.

Any solution $x$ tells us exactly how to use the pieces so we can put a broken object back together...

# The Humpty Dumpty Solution



*Humpty Dumpty thought he was through,*
*But linear algebra knew what to do.*
*It set up the system and solved it and then*
*Neatly put Humpty together again.*

# References

Cleve Moler's T Puzzle:

https://people.sc.fsu.edu/~jburkardt/m_src/t_puzzle_gui/t_puzzle_gui.m

https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/moler/exm/chapters/puzzle.pdf

Online references:

https://people.sc.fsu.edu/~jburkardt/presentations/t_puzzle_2022_pitt.pdf

https://people.sc.fsu.edu/~jburkardt/m_src/t_puzzle/t_puzzle.html

Journal article:

Marcus Garvie, John Burkardt,
*A New Mathematical Model for Tiling Finite Regions of the Plane with Polyominoes*,
Contributions to Discrete Mathematics,
Volume 15, Number 2, July 2020, pages 95-131.