Using graph algorithms to debug equation-oriented chemical process models

Background, and tutorial on `incidence_analysis` package

Robert Parker

Advised by Lorenz Biegler

February 25, 2022

PSE Seminar

Tutorial overview

Roadmap:

- Background on graph algorithms and algebraic modeling (these slides)
- Two examples using a model from IDAES (`incidence_examples` repo & Jupyter)
- Tutorial applying algorithms to small model using `incidence_analysis` (`incidence_examples` repo & Jupyter)

Goals:

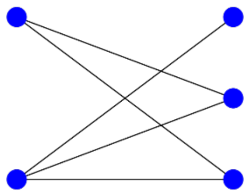
- You will know how to use incidence_analysis to apply graph algorithms to your models
- You will know why you might want to

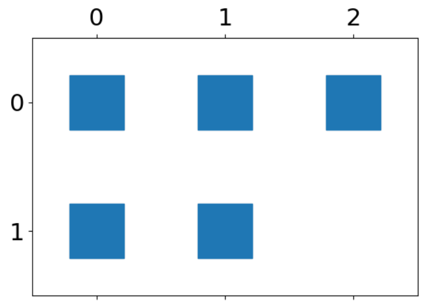
Every equation-oriented algebraic model has an associated bipartite graph of variables and equations

The model and graph have an associated incidence matrix

$$x^2 + y^2 + z^2 = 1$$

$$x + 2y = 3$$





This graph can tell us a lot about our process model:

- Whether any equations are redundant
- Which variables are *not* appropriate degrees of freedom
- Whether the model is structurally singular
- Where in the model a singularity is coming from

And can lead us towards decompositions that help us initialize, simulate, or optimize our process model

The first algorithm that helps us is for computing a maximum matching

Matching — A set of pairs of adjacent nodes such that no node appears more than once

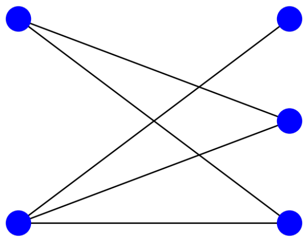
Maximum matching — The largest possible matching (most pairs of nodes) in a given graph

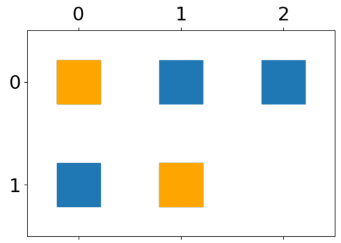
Perfect matching — A matching that contains every node (variable and equation

Unmatched variables are fine (just degrees of freedom)

$$x^2 + y^2 + z^2 = 1$$

$$x + 2y = 3$$



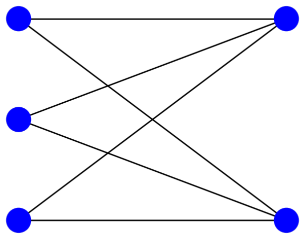


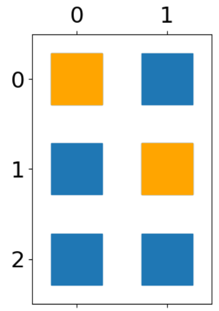
Unmatched equations are usually a problem

$$x^2 + y^2 = 1$$

$$x + 2y = 3$$

$$xy = 7$$





- A maximum matching in a bipartite graph can be computed in polynomial time
- If our model doesn't have a perfect matching, its Jacobian is guaranteed to be singular (cannot have a zero-free diagonal)
- Matching is not unique, so unmatched variables/equations may not tell us what is wrong

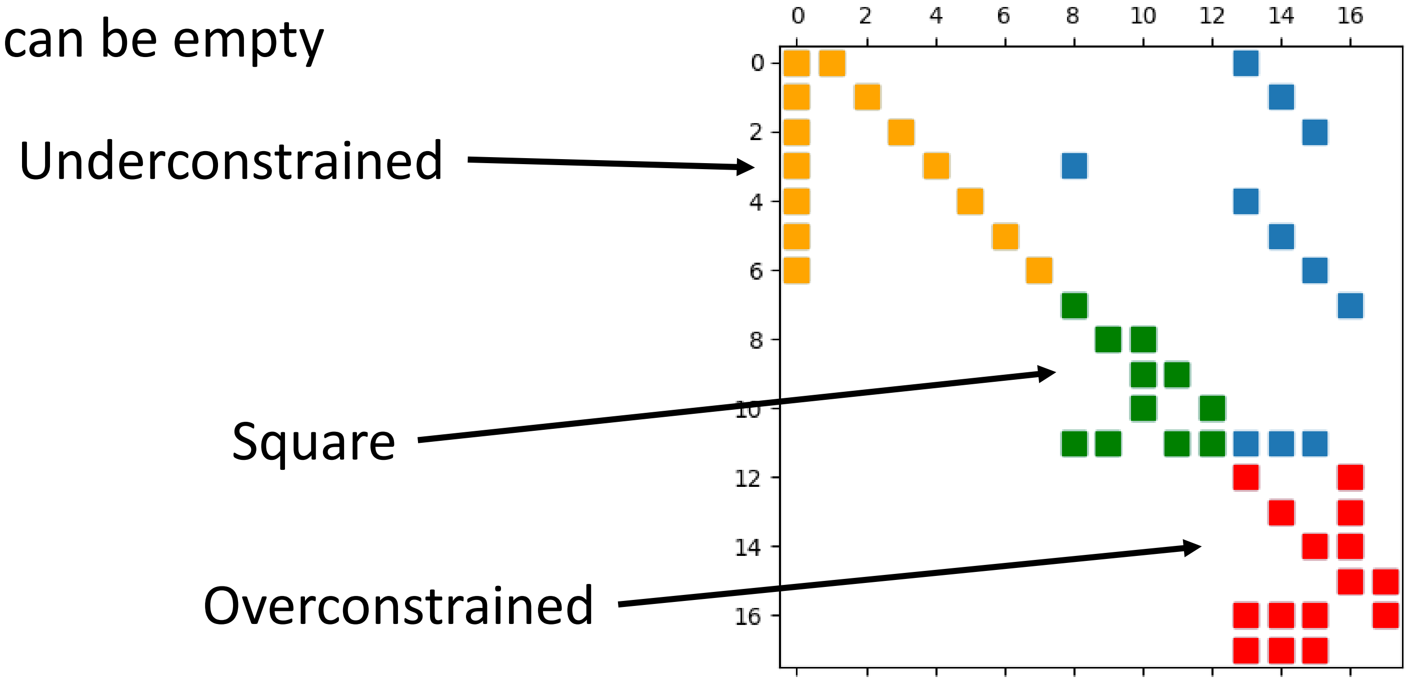
Duff, I.S., On algorithms for obtaining a maximum transversal. ACM Trans. Math. Soft., 1981

The Dulmage-Mendelsohn partition determines which variables and equations *can possibly be* unmatched

Every matrix has a Dulmage-Mendelsohn partition

Partitions rows and columns into three subsets

Any of these subsets can be empty



Underconstrained

Square

Overconstrained

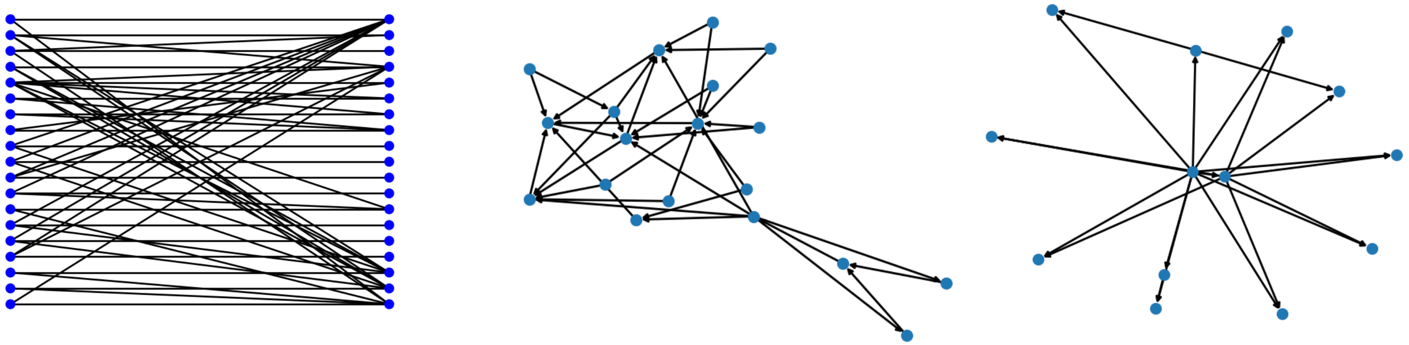
Dulmage, A.L. and Mendelsohn, N.S., Coverings of bipartite graphs. Can. J. Math., 1958.

Pothen, A. and Fan, C.J., Computing the block triangular form of a sparse matrix, ACM Trans. Math. Soft., 1990

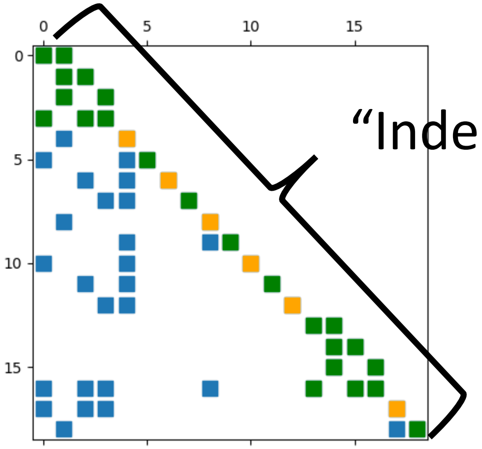
Block triangularization can show what subsystem is causing singularity or ill-conditioning

Block triangularization:

- Perfect matching
- Strongly connected components
- Topological order



Duff, I.S. and Reid, J.K., An implementation of Tarjan’s algorithm for the block triangularization of a matrix. ACM Trans. Math. Soft., 1978



“Independent” diagonal blocks

- Solving diagonal subblocks in order is equivalent to solving entire system
- Nonsingularity is equivalent to nonsingularity of all diagonal blocks

We will motivate these algorithms with two examples, then walk through a tutorial

```
github.com/robbybp/incidence_examples
```

- `incidence_examples/example1/clc_dm_example.ipynb`
- `incidence_examples/example2/clc_scc_example.ipynb`
- `incidence_examples/tutorial/tutorial_filled_in.ipynb`

```
jupyter notebook <filename>
```