Principles of Equation-Based Object-Oriented Modeling and Languages

Module B: DAEs and Algorithms in EOO Languages

Mini-course, Scuola Superiore Sant'Anna, Pisa, Italy.
December 9-10, 2014

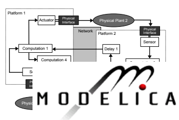
David Broman

Associate Professor, KTH Royal Institute of Technology

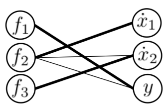
Assistant Research Engineer, University of California, Berkeley

Slides updated April 8, 2015.

Course Structure



Module A
EOO Languages and Modelica Fundamentals



Module B
DAEs and Algorithms in EOO Languages

Module C
Modelyze – Defining Equation-Based DSLs

Module D
Co-simulation and the Functional Mock-up Interface

Agenda

Part I
DAE Basics

Part II
Matching

Part III
BLT Sorting

Part IV
Pantelides

Part V
Dummy Derivatives

Part I

DAE Basics

$$\dot{x} = -x + y - z$$
$$z = x^2 + y^2$$
$$z = x + x * y$$

DAE

System of Differential algebraic equation (DAE) in general form:

$$F(x, \dot{x}, y, t) = 0$$

Alternatively, a DAE can be defined as *F(u',u) = 0*, where *F* and *u* have the same dimensions and *F* is assumed to be sufficiently differential.

$$\text{where } x, \in \mathbb{R}^n, \dot{x} \in \mathbb{R}^n, y \in \mathbb{R}^m, F : G \subseteq \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^{n+m}.$$

We have *n* number of variables that appear differentiated

We have *m* number of variables that *do not* appear differentiated

We have *n+m* number of equations

$$\boxed{\begin{aligned} \dot{x} &= -x \\ x(0) &= 1 \end{aligned}}$$

$$\boxed{\begin{aligned} \dot{x} &= -x + y \\ x^2 + y^2 &= 10 \end{aligned}}$$

1 variable *x* is differentiated
1 variable y is not differentiated
2 equations

ODE, initial value problem (IVP)

Is this an ODE or an DAE?

DAE, Example 1

$$\boxed{\begin{aligned} \dot{x} &= -x + y \\ x^2 + y^2 &= 10 \end{aligned}}$$

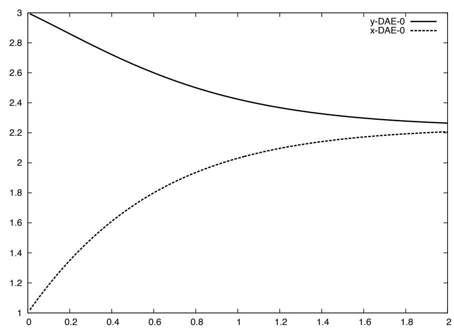
$$x(0) = 1$$

$$y(0) = 3$$

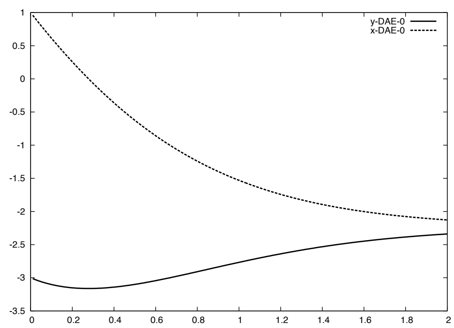
Is it an initial value problem (IVP)?

What should the initial value for y be?

We need to find consistent initial values.
Note that y(0) = -3 is also a consistent initial value.



$x(0) = 1 \quad y(0) = 3$



$x(0) = 1 \quad y(0) = -3$

DAE, Example 1

$$\dot{x} = -x + y$$
$$x^2 + y^2 = 10$$

Can we find an order that can solve these equations?

Yes, in each step:

1. Solve for *y* in equation (2). x is known.
2. Solve for x' in equation (1). Now both x and y are known.

In this case, we can actually symbolically transform this into an ODE directly.

$$\dot{x} = -x + \sqrt{10 - x^2}$$

(note that the DAE is nonlinear; we need to decide on a sign, which must be consistent with the initial values.)

DAE, Example 2

$$\dot{x} = -x + y - z$$
$$z = x^2 + y^2$$
$$z = x + x \cdot y$$

Is this an DAE?

Yes, one differentiated variable (x) and two algebraic variables (y and z)

Can we find an order that can solve these equations?

No, equations 2 and 3 are algebraically dependent on each other.

This is called the *semi-explicit form* of an DAE

$$\dot{x} = f(x, y, t)$$
$$0 = g(x, y, t)$$

One solution approach (in each time step)

1. Solve (nonlinear) algebraic equations
2. Solve differentiated variables

DAE Index

<u>Definition</u>: The index of an DAE is the minimum number of times that all or part of the DAE must be differentiated with respect to *t* in order to determine *x'* as a continuous function of *x* and *t*.

(Brenan, Campbell, Petzold, 1989)

This definition is called the differential index.

$$\begin{aligned} \dot{x} &= -x + y \\ x^2 + y^2 &= 10 \end{aligned}$$

Our first example was an index 1 DAE.

No differentiation is need to obtain an ODE. An ODE has index 0.

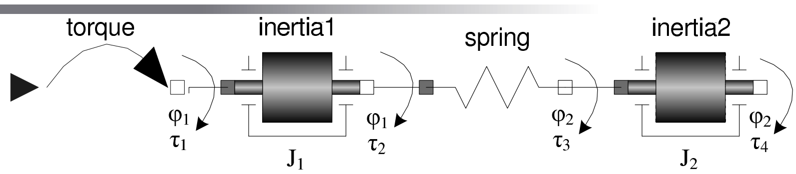
$$\begin{aligned} \dot{x} &= -x + y - z \\ z &= x^2 + y^2 \\ z &= x + x \cdot y \end{aligned}$$

Example two has an algebraic loop, and the two algebraic equations are non-singular. Example of an index 1 DAE.

Note that you can differentiate parts of the equation system once (equations (2) and (3)) to obtain an ODE. (Not recommended for numerical stability)

We will soon see examples where a system of equation is singular. These may be *higher-index DAEs (index > 1).*

Drive Shaft Example



Is this an DAE?

$$\begin{aligned} \dot{\varphi}_1 &= \omega_1 \\ \dot{\varphi}_2 &= \omega_2 \\ \dot{\omega}_1 &= \frac{\tau_1 + \tau_2}{J_1} \\ \dot{\omega}_2 &= \frac{\tau_3 + \tau_4}{J_2} \\ \tau_1 &= u \\ \tau_2 &= c \cdot (\varphi_2 - \varphi_1) \\ \tau_3 &= -c \cdot (\varphi_2 - \varphi_1) \\ \tau_4 &= 0 \end{aligned}$$

Variables: $(\varphi_1, \varphi_2, \omega_1, \omega_2, \tau_1, \tau_2, \tau_3, \tau_4)$

Appearing differentiated: $(\dot{\varphi}_1, \dot{\varphi}_2, \dot{\omega}_1, \dot{\omega}_2)$

Incidence matrix. Differentiated variables and the algebraic variables are unknown.

$$\begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} \dot{\omega}_1 & \dot{\omega}_2 & \dot{\phi}_1 & \dot{\phi}_2 & \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{array} \\ \begin{pmatrix} 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 \\ \boxed{1} & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} \end{pmatrix} \end{array}$$

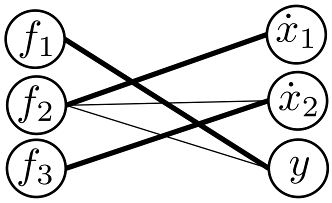
$$\begin{array}{c} \\ f_8 \\ f_7 \\ f_6 \\ f_5 \\ f_4 \\ f_3 \\ f_2 \\ f_1 \end{array} \begin{array}{c} \begin{array}{cccccccc} \tau_4 & \tau_3 & \tau_2 & \tau_1 & \dot{\omega}_2 & \dot{\omega}_1 & \dot{\phi}_2 & \dot{\phi}_1 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Matching: Find a unique mapping between variables and equations.

Sorting: Sort equations (permute matrix)

Part II

Matching



Example: Matching

System of equations

$$\boxed{\begin{aligned} &f_1(y) = 0 \\ &f_2(\dot{x}_1, \dot{x}_2, y) = 0 \\ &f_3(\dot{x}_2) = 0 \end{aligned}}$$

Construct a bipartite graph

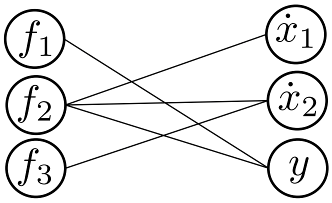
$$G = (F, V, E)$$

$$F = \{f_1, f_2, f_3\} \qquad E = \{(f_1, y), (f_2, \dot{x}_1), (f_2, \dot{x}_2), (f_2, y), (f_3, \dot{x}_2)\}$$

$$V = \{\dot{x}_1, \dot{x}_2, y\}$$

Incidence Matrix

$$\begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \end{array} \begin{array}{c} \begin{array}{ccc} \dot{x}_1 & \dot{x}_2 & y \end{array} \\ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$



Algorithm: Matching

MATCH(G)
$assign \leftarrow \emptyset$
for each $f \in G.F$
  do $C \leftarrow \emptyset$
    if not MATCH-EQUATION($G, f, \underline{C}, \underline{assign}, \emptyset$)
      then return (FALSE, $assign$)
return (TRUE, $assign$)

Color visited vertices

$$C \subseteq G.F \cup G.V$$

Assigns variables to equations

$$assign[v] = \begin{cases} f & \text{if } f \text{ matches } v \\ \text{NIL} & \text{otherwise} \end{cases}$$

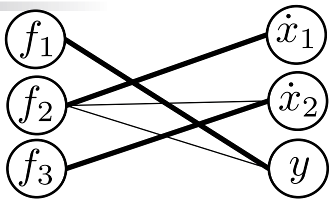
Underline means call by reference.

MATCH-EQUATION($G, f, \underline{C}, \underline{assign}, vmap$)
$C \leftarrow C \cup \{f\}$
if there exits a $v \in G.V$ such that $(f, v) \in G.E$
  and $assign[v] = \text{NIL}$ and $vmap[v] = \text{NIL}$
  then $assign[v] \leftarrow f$
    return TRUE
  else for each v where $(f, v) \in G.E$ and $v \notin C$
      and $vmap[v] = \text{NIL}$
      do $C \leftarrow C \cup \{v\}$
        if MATCH-EQUATION($G, assign[v], \underline{C}, \underline{assign}, vmap$)
          then $assign[v] \leftarrow f$
            return TRUE
return FALSE

vmap and *equation coloring* is not used until in Part IV.

Example: Matching

MATCH(G)
$assign \leftarrow \emptyset$
for each $f \in G.F$
  do $C \leftarrow \emptyset$
    if not MATCH-EQUATION($G, f, \underline{C}, \underline{assign}, \emptyset$)
      then return (FALSE, $assign$)
return (TRUE, $assign$)



Exercise

Do each step of the algorithms and keep track of C and *assign*.

MATCH-EQUATION($G, f, \underline{C}, \underline{assign}, vmap$)
$C \leftarrow C \cup \{f\}$
if there exits a $v \in G.V$ such that $(f, v) \in G.E$
  and $assign[v] = \text{NIL}$ and $vmap[v] = \text{NIL}$
  then $assign[v] \leftarrow f$
    return TRUE
  else for each v where $(f, v) \in G.E$ and $v \notin C$
      and $vmap[v] = \text{NIL}$
      do $C \leftarrow C \cup \{v\}$
        if MATCH-EQUATION($G, assign[v], \underline{C}, \underline{assign}, vmap$)
          then $assign[v] \leftarrow f$
            return TRUE
return FALSE

Case A: For f2, use x1.

$$assign = \{y \mapsto f_1, \dot{x}_1 \mapsto f_2, \dot{x}_2 \mapsto f_3\}$$
$$C = \{f_1, f_2, f_3\}$$

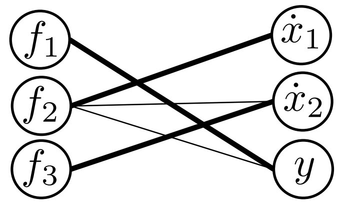
Case B: For f2, first use x2 (Reassignment of x2)

$$assign = \{y \mapsto f_1, \dot{x}_1 \mapsto f_2, \dot{x}_2 \mapsto f_3\}$$
$$C = \{f_1, f_2, f_3, \dot{x}_2, \}$$

Example: Matching

System of equations

$$
\begin{aligned}
f_1(y) &= 0 \\
f_2(\dot{x}_1, \dot{x}_2, y) &= 0 \\
f_3(\dot{x}_2) &= 0
\end{aligned}
$$



Incidence Matrix

$$
\begin{array}{c c}
 & \begin{array}{ccc} \dot{x}_1 & \dot{x}_2 & y \end{array} \\
\begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} & \begin{pmatrix} 0 & 0 & \boxed{1} \\ \boxed{1} & 1 & 1 \\ 0 & \boxed{1} & 0 \end{pmatrix}
\end{array}
$$

We may now permute the matrix

$$
\begin{array}{c c}
 & \begin{array}{ccc} \dot{x}_2 & y & \dot{x}_1 \end{array} \\
\begin{array}{c} f_3 \\ f_1 \\ f_2 \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}
\end{array}
$$

The matching problem solves the problem of finding a permutation such that the matrix has a nonzero diagonal. Also called *maximum traversal*.

Sorting into Lower Triangular Form

$$
\begin{array}{c c}
 & \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array} \\
\begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{array} & \begin{pmatrix}
0 & 0 & \boxed{1} & 0 & 0 & 1 & 0 \\
\boxed{1} & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & \boxed{1} & 1 & 1 & 0 \\
0 & \boxed{1} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & \boxed{1} & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & \boxed{1} & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & \boxed{1}
\end{pmatrix}
\end{array}
$$

Unsorted Matrix

Find a matching

$$
\begin{array}{c c}
 & \begin{array}{ccccccc} x_2 & x_6 & x_3 & x_7 & x_5 & x_4 & x_1 \end{array} \\
\begin{array}{c} f_4 \\ f_6 \\ f_1 \\ f_7 \\ f_5 \\ f_3 \\ f_2 \end{array} & \begin{pmatrix}
\boxed{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \boxed{1} & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & \boxed{1} & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & \boxed{1} & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & \boxed{1} & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & \boxed{1} & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & \boxed{1}
\end{pmatrix}
\end{array}
$$

Sorting (permutation of Matrix) into <u>Lower Triangular Matrix Form</u>

We now have a causal form; solving the equation system is straight forward.

But, we cannot always permute the matrix into lower triangular form…

Sorting into Block Lower Triangular (BLT) Form

$$
\begin{array}{c|ccccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\
\hline
f_1 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & 0 \\
f_2 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \\
f_3 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\
f_4 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
f_5 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\
f_6 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} & 0 \\
f_7 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1}
\end{array}
$$

Another unsorted Matrix

$$
\begin{array}{c|ccccccc}
 & x_2 & x_6 & x_3 & x_5 & x_1 & x_4 & x_7 \\
\hline
f_4 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
f_6 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
f_1 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 \\
f_5 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
f_2 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\
f_3 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\
f_7 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1}
\end{array}
$$

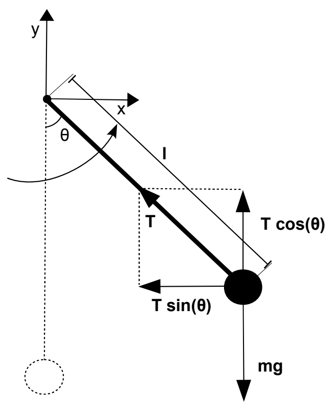
Sorting (permutation of Matrix) into Block Lower Triangular (BLT) Form

We have identified an algebraic loop.

At each time step, the algebraic loops may be solved using Guassian elimination (if linear) or a Newton's method (if nonlinear).

In part III we discuss a BLT sorting algorithm

Example: Pendulum



Pendulum in Cartesian coordinate system

$$
\begin{aligned}
-T \cdot \frac{x}{l} &= m\ddot{x} \\
-T \cdot \frac{y}{l} - mg &= m\ddot{y} \\
x^2 + y^2 &= l^2
\end{aligned}
$$

Simplified using

$$
\begin{aligned}
-T/l &= \lambda \\
m &= 1 \\
l^2 &= L
\end{aligned}
$$

Simplified

$$
\begin{aligned}
\ddot{x} &= \lambda \cdot x \\
\ddot{y} &= \lambda \cdot y - g \\
x^2 + y^2 &= L
\end{aligned}
$$

Rewritten in first order

$$
\begin{aligned}
\dot{x} &= u \\
\dot{y} &= v \\
\dot{u} &= \lambda \cdot x \\
\dot{v} &= \lambda \cdot y - g \\
x^2 + y^2 &= L
\end{aligned}
$$

Incidence Matrix

$$
\begin{array}{c|ccccc}
 & \dot{x} & \dot{y} & \dot{u} & \dot{v} & \lambda \\
\hline
f_1 & \mathbf{1} & 0 & 0 & 0 & 0 \\
f_2 & 0 & \mathbf{1} & 0 & 0 & 0 \\
f_3 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} \\
f_4 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\
f_5 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Is this an DAE?

Can we solve it?

Can we create BLT?

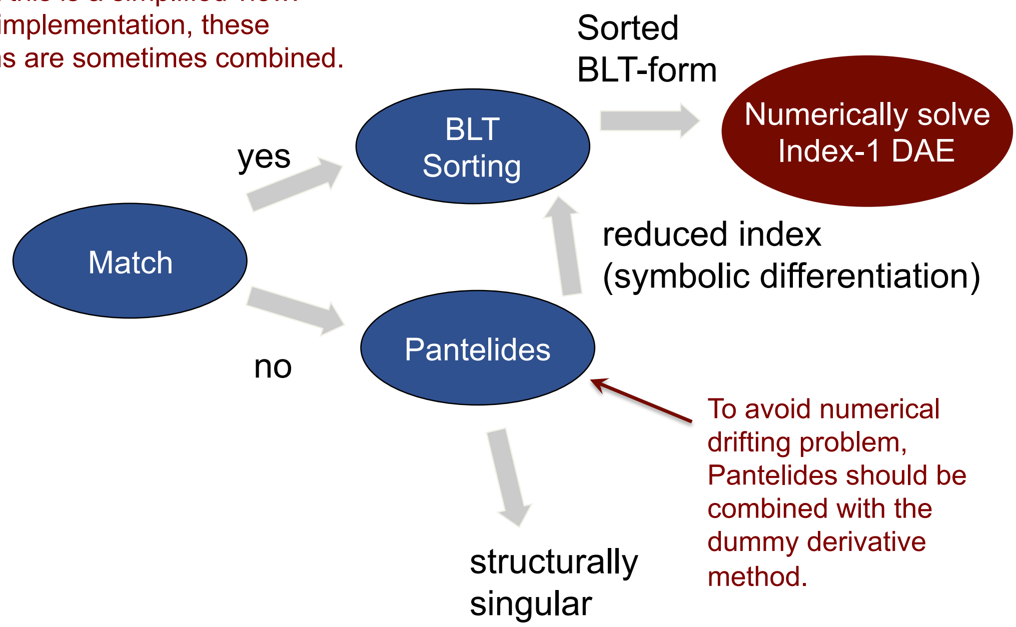
IDEA: Symbolically differentiate equations to get derivatives.

No, we cannot find a matching (see f_5). This is a higher-index problem (index > 1).

How can we determine which equations to differentiate? Solution: **Pantelides Algorithm** (other methods exist, such as Pryce's method)

How the algorithms fit together (simplified)

Note that this is a simplified view:
In a real implementation, these algorithms are sometimes combined.



Part C

BLT Sorting

Algorithm: BLT Sort

Input: a bipartite graph G

```
BLT(G)
  (match, assign) ← MATCH(G)
  if not match
     then  return error "Singular"

  D.V ← G.F
  D.E ← ∅
  for each (f, v) ∈ G.E where f ∈ G.F and assign[v] ≠ f
      do D.E ← D.E ∪ {(assign[v], f)}

  MAKEEMPTY(O)
  MAKEEMPTY(S)
  i ← 0
  lowlink ← ∅
  number ← ∅
  for each v ∈ D.V
      do if number[v] = NIL
            then STRONGCONNECT(v, D, S, i, lowlink, number, O)
  return O
```

Output: a stack of sets of equation vertices, where each set represents an equation block in the BLT matrix.

Algorithm: BLT Sort

Input: a bipartite graph G

```
BLT(G)
  (match, assign) ← MATCH(G)
  if not match
     then  return error "Singular"

  D.V ← G.F
  D.E ← ∅
  for each (f, v) ∈ G.E where f ∈ G.F and assign[v] ≠ f
      do D.E ← D.E ∪ {(assign[v], f)}

  MAKEEMPTY(O)
  MAKEEMPTY(S)
  i ← 0
  lowlink ← ∅
  number ← ∅
  for each v ∈ D.V
      do if number[v] = NIL
            then STRONGCONNECT(v, D, S, i, lowlink, number, O)
  return O
```

Step 1
Find matching

Step 2
Construct equation dependency graph

Step 3
Sort into blocks of equations using Tarjan's strongly connected component algorithm

Output: a stack of sets of equation vertices, where each set represents an equation block in the BLT matrix.

Example: BLT Sort

$$
\begin{array}{c}
\begin{array}{cccccc} \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & \dot{x}_4 & y_1 & y_2 \end{array} \\
\begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{array}
\begin{pmatrix}
0 & 0 & 1 & \boxed{1} & 0 & 1 \\
\boxed{1} & 1 & 0 & 0 & 1 & 1 \\
0 & \boxed{1} & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & \boxed{1} & 0 \\
0 & 1 & 0 & 0 & 0 & \boxed{1} \\
0 & 0 & \boxed{1} & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

$$G = (F, V, E)$$

$$F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$$
$$V = \{\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, y_1, y_2\}$$

In Part 1 of BLT - matching

$(match, assign) \leftarrow \text{MATCH}(G)$
if not $match$
  then return error "Singular"

Returns TRUE (steps omitted) with assignment

$$assign = \{\dot{x}_1 \mapsto f_2, \dot{x}_2 \mapsto f_3, \dot{x}_3 \mapsto f_6, \dot{x}_4 \mapsto f_1, y_1 \mapsto f_4, y_2 \mapsto f_5\}$$

Algorithm: BLT Sort

BLT(G) — Input: a bipartite graph G

$(match, assign) \leftarrow \text{MATCH}(G)$
if not $match$
  then return error "Singular"

$D.V \leftarrow G.F$
$D.E \leftarrow \emptyset$
for each $(f, v) \in G.E$ where $f \in G.F$ and $assign[v] \neq f$
  do $D.E \leftarrow D.E \cup \{(assign[v], f)\}$

Step 2
Construct equation dependency graph

$\text{MAKEEMPTY}(O)$
$\text{MAKEEMPTY}(S)$
$i \leftarrow 0$
$lowlink \leftarrow \emptyset$
$number \leftarrow \emptyset$
for each $v \in D.V$
  do if $number[v] = \text{NIL}$
    then $\text{STRONGCONNECT}(v, D, \underline{S}, \underline{i}, \underline{lowlink}, \underline{number}, \underline{O})$
return O

Output: a stack of sets of equation vertices, where each set represents an equation block in the BLT matrix.

Example: BLT Sort

$$\begin{array}{c|cccccc} & \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & \dot{x}_4 & y_1 & y_2 \\ \hline f_1 & 0 & 0 & 1 & \boxed{1} & 0 & 1 \\ f_2 & \boxed{1} & 1 & 0 & 0 & 1 & 1 \\ f_3 & 0 & \boxed{1} & 1 & 0 & 0 & 0 \\ f_4 & 1 & 0 & 1 & 0 & \boxed{1} & 0 \\ f_5 & 0 & 1 & 0 & 0 & 0 & \boxed{1} \\ f_6 & 0 & 0 & \boxed{1} & 0 & 0 & 1 \end{array}$$

When solving x'_4, variables x'_3 and y_2 need to be known. Hence, f_1 is dependent on f_5 and f_6.

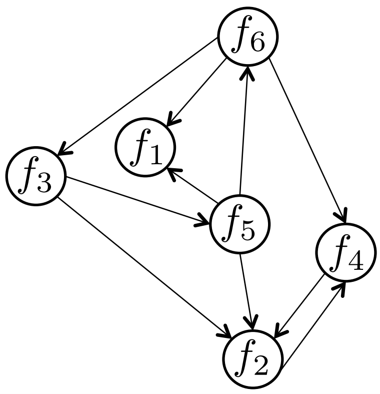
$$G = (F, V, E) \qquad F = \{f_1, f_2, f_3, f_4, f_5, f_6\} \quad V = \{\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, y_1, y_2\}$$

Matching tells us that x'_3 is supposed to be solved using f_6

$$assign = \{\dot{x}_1 \mapsto f_2, \dot{x}_2 \mapsto f_3, \dot{x}_3 \mapsto f_6, \dot{x}_4 \mapsto f_1, y_1 \mapsto f_4, y_2 \mapsto f_5\}$$

In Part 2 of BLT – construct equation dependency graph (digraph)

```
D.V ← G.F
D.E ← ∅
for each (f, v) ∈ G.E where f ∈ G.F and assign[v] ≠ f
    do D.E ← D.E ∪ {(assign[v], f)}
```



$$D = (V, E)$$
$$V = \{f_1, f_2, f_3, f_4, f_5, f_6\}$$
$$E = \{f_2 \mapsto f_4, f_3 \mapsto f_2, f_3 \mapsto f_5, f_4 \mapsto f_2, f_5 \mapsto f_1, f_5 \mapsto f_2, f_5 \mapsto f_6, f_6 \mapsto f_1, f_6 \mapsto f_3, f_6, \mapsto f_4\}$$

Exercise
Create D graphically

Algorithm: BLT Sort

Input: a bipartite graph G

```
BLT(G)
(match, assign) ← MATCH(G)
if not match
   then return error "Singular"

D.V ← G.F
D.E ← ∅
for each (f, v) ∈ G.E where f ∈ G.F and assign[v] ≠ f
    do D.E ← D.E ∪ {(assign[v], f)}

MAKEEMPTY(O)
MAKEEMPTY(S)
i ← 0
lowlink ← ∅
number ← ∅
for each v ∈ D.V
    do if number[v] = NIL
          then STRONGCONNECT(v, D, S, i, lowlink, number, O)
return O
```

Step 3
Sort into blocks of equations using Tarjan's strongly connected component algorithm

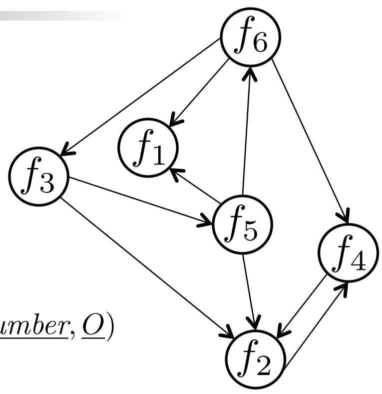
Output: a stack of sets of equation vertices, where each set represents an equation block in the BLT matrix.

Algorithm: StrongConnect (Tarjan)

```
StrongConnect(v, D, S, i, lowlink, number, O)
  i ← i + 1
  lowlink[v] ← i
  number[v] ← i
  PUSH(S, v)
  for each w ∈ D.V where (v, w) ∈ D.E
      do if number[w] = NIL
            then StrongConnect(w, D, S, i, lowlink, number, O)
                 lowlink[v] ← MIN(lowlink[v], lowlink[w])
            else  if w ∈ S and number[w] < number[v]
                     then lowlink[v] ← MIN(lowlink[v], number[w])
  if lowlink[v] = number[v]
     then eqset ← ∅
          while not ISEMPTY(S) and number[TOP(S)] ≥ number[v]
              do eqset ← eqset ∪ {POP(S)}
          PUSH(O, eqset)
  return
```


Algorithm: StrongConnect (Tarjan)

```
StrongConnect(v, D, S, i, lowlink, number, O)
  i ← i + 1
  lowlink[v] ← i
  number[v] ← i
  PUSH(S, v)
  for each w ∈ D.V where (v, w) ∈ D.E
      do if number[w] = NIL
            then StrongConnect(w, D, S, i, lowlink, number, O)
                 lowlink[v] ← MIN(lowlink[v], lowlink[w])
            else  if w ∈ S and number[w] < number[v]
                     then lowlink[v] ← MIN(lowlink[v], number[w])
  if lowlink[v] = number[v]
     then eqset ← ∅
          while not ISEMPTY(S) and number[TOP(S)] ≥ number[v]
              do eqset ← eqset ∪ {POP(S)}
          PUSH(O, eqset)
  return
```



Exercise
Construct stack O

Top of the stack is to the left

$O = [\{f_3, f_5, f_6\}, \{f_2, f_4\}, \{f_1\}]$

Example: BLT Sort

$$
\begin{array}{c|cccccc}
 & \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & \dot{x}_4 & y_1 & y_2 \\
\hline
f_1 & 0 & 0 & 1 & \boxed{1} & 0 & 1 \\
f_2 & \boxed{1} & 1 & 0 & 0 & 1 & 1 \\
f_3 & 0 & \boxed{1} & 1 & 0 & 0 & 0 \\
f_4 & 1 & 0 & 1 & 0 & \boxed{1} & 0 \\
f_5 & 0 & 1 & 0 & 0 & 0 & \boxed{1} \\
f_6 & 0 & 0 & \boxed{1} & 0 & 0 & 1
\end{array}
$$

$$G = (F, V, E)$$

$$F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$$

$$V = \{\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, y_1, y_2\}$$

$$assign = \{\dot{x}_1 \mapsto f_2, \dot{x}_2 \mapsto f_3, \dot{x}_3 \mapsto f_6, \dot{x}_4 \mapsto f_1, y_1 \mapsto f_4, y_2 \mapsto f_5\}$$

$$O = [\{f_3, f_5, f_6\}, \{f_2, f_4\}, \{f_1\}]$$

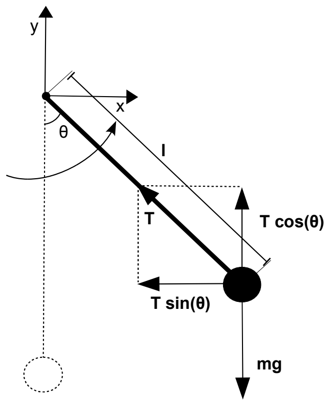
We can now create the sorted BLT matrix

$$
\begin{array}{c|cccccc}
 & \dot{x}_2 & y_2 & \dot{x}_3 & \dot{x}_1 & y_1 & \dot{x}_4 \\
\hline
f_3 & 1 & 0 & 1 & 0 & 0 & 0 \\
f_5 & 1 & 1 & 0 & 0 & 0 & 0 \\
f_6 & 0 & 1 & 1 & 0 & 0 & 0 \\
f_2 & 1 & 1 & 0 & 1 & 1 & 0 \\
f_4 & 0 & 0 & 1 & 1 & 1 & 0 \\
f_1 & 0 & 1 & 1 & 0 & 0 & 1
\end{array}
$$

Part IV

Pantelides

Example: Pendulum



Pendulum in Cartesian coordinate system

$$
\begin{aligned}
-T \cdot \frac{x}{l} &= m\ddot{x} \\
-T \cdot \frac{y}{l} - mg &= m\ddot{y} \\
x^2 + y^2 &= l^2
\end{aligned}
$$

Simplified using

$$
\begin{aligned}
-T/l &= \lambda \\
m &= 1 \\
l^2 &= L
\end{aligned}
$$

Simplified

$$
\begin{aligned}
\ddot{x} &= \lambda \cdot x \\
\ddot{y} &= \lambda \cdot y - g \\
x^2 + y^2 &= L
\end{aligned}
$$

Rewritten in first order

$$
\begin{aligned}
\dot{x} &= u \\
\dot{y} &= v \\
\dot{u} &= \lambda \cdot x \\
\dot{v} &= \lambda \cdot y - g \\
x^2 + y^2 &= L
\end{aligned}
$$

Incidence Matrix

$$
\begin{array}{c c}
 & \begin{array}{ccccc} \dot{x} & \dot{y} & \dot{u} & \dot{v} & \lambda \end{array} \\
\begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} &
\begin{pmatrix}
\mathbf{1} & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & \mathbf{1} \\
0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

Pendulum: Graph Construction

System of equations

$$
\begin{aligned}
\dot{x} &= u \\
\dot{y} &= v \\
\dot{u} &= \lambda \cdot x \\
\dot{v} &= \lambda \cdot y - g \\
x^2 + y^2 &= L
\end{aligned}
$$

$$
\begin{aligned}
f_1(\dot{x}, u) &= 0 \\
f_2(\dot{y}, v) &= 0 \\
f_3(\dot{u}, \lambda, x) &= 0 \\
f_4(\dot{v}, \lambda, y) &= 0 \\
f_5(x, y) &= 0
\end{aligned}
$$

Note that we include both differentiated and not differentiated variables.

Construct a bipartite graph
$G = (F, V, E)$

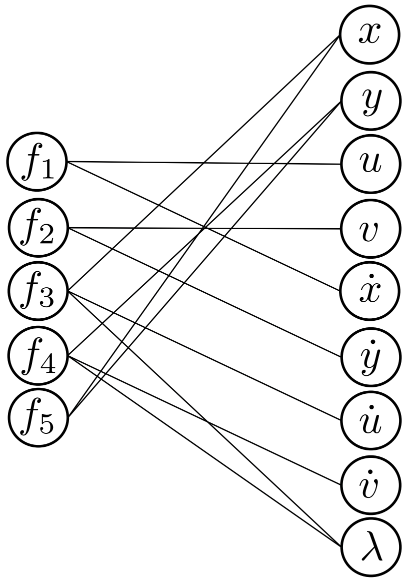
$$F = \{f_1, f_2, f_3, f_4, f_5\}$$

$$V = \{x, y, u, v, \dot{x}, \dot{y}, \dot{u}, \dot{v}, \lambda\}$$

$$
\begin{aligned}
E = \{&(f_1, \dot{x}), (f_1, u), \\
&(f_2, \dot{y}), (f_2, v), \\
&(f_3, \dot{u}), (f_3, \lambda), (f_3, x), \\
&(f_4, \dot{v}), (f_4, \lambda), (f_4, y), \\
&(f_5, x), (f_5, y)\}
\end{aligned}
$$

Pendulum

$$f_1(\dot{x}, u) = 0$$
$$f_2(\dot{y}, v) = 0$$
$$f_3(\dot{u}, \lambda, x) = 0$$
$$f_4(\dot{v}, \lambda, y) = 0$$
$$f_5(x, y) = 0$$



Algorithm: Pantelides

```
PANTELIDES(G, vmap, eqmap)
  assign ← ∅
  for each e ∈ G.F
      do f ← e
         repeat
                C ← ∅
                match ← MATCH-EQUATION(G, f, C, assign, vmap)
                if not match
                  then for each v ∈ C where v ∈ G.V
                           do let v' be a vertex, such that v' ∉ G.V
                              vmap[v] ← v'
                              G.V ← G.V ∪ {v'}
                       for each f ∈ C where f ∈ G.F
                           do let f' be a vertex, such that f' ∉ G.F
                              eqmap[f] ← f'
                              G.F ← G.F ∪ {f'}
                              for each v ∈ G.V where (f, v) ∈ G.E
                                  do G.E ← G.E ∪ {(f', v), (f', vmap[v])}
                       for each v ∈ C where v ∈ G.V
                           do assign[vmap[v]] ← eqmap[assign[v]]
                       f ← eqmap[f]
         until match
  return assign
```

Mapping variables to differentiated variables

$$vmap[v] = \begin{cases} v' & \text{if } \frac{dv}{dt} = v' \\ \text{NIL} & \text{otherwise} \end{cases}$$

Mapping equations to their differentiated version

$$eqmap[f] = \begin{cases} f' & \text{if } \frac{df}{dt} = f' \\ \text{NIL} & \text{otherwise} \end{cases}$$

Assigns variables to equations

$$assign[v] = \begin{cases} f & \text{if } f \text{ matches } v \\ \text{NIL} & \text{otherwise} \end{cases}$$

Pendulum

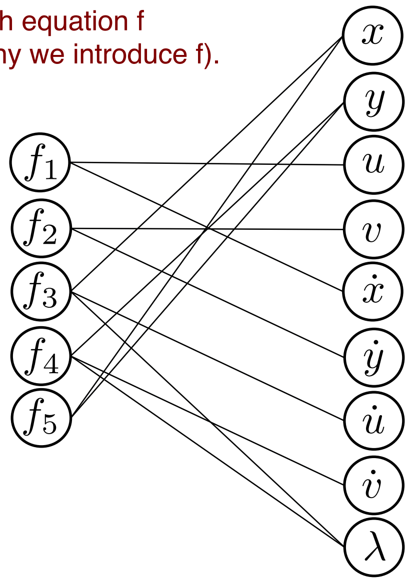
We start with no variable to equation assignments.

Iterate over each equation f (we see later why we introduce f).

PANTELIDES(G, $\underline{vmap}$, $\underline{eqmap}$)

1 $assign \leftarrow \emptyset$
2 **for** each $e \in G.F$
3 **do** $f \leftarrow e$
4 **repeat**
5 $C \leftarrow \emptyset$

Preparation for matching algorithm.
Set all vertices to be uncolored.



Initial state after step 5.

$$
\begin{aligned}
vmap &= \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\} \\
eqmap &= \{\} \\
assign &= \{\} \\
C &= \{\}
\end{aligned}
$$

Algorithm: Pantelides

PANTELIDES(G, $\underline{vmap}$, $\underline{eqmap}$)

1 $assign \leftarrow \emptyset$
2 **for** each $e \in G.F$
3 **do** $f \leftarrow e$
4 **repeat**
5 $C \leftarrow \emptyset$
6 $match \leftarrow$ MATCH-EQUATION($G, f, \underline{C}, \underline{assign}, vmap$)
7 **if** not $match$
8 **then for** each $v \in C$ where $v \in G.V$
9 **do** let v' be a vertex, such that $v' \notin G.V$
10 $vmap[v] \leftarrow v'$
11 $G.V \leftarrow G.V \cup \{v'\}$
12 **for** each $f \in C$ where $f \in G.F$
13 **do** let f' be a vertex, such that $f' \notin G.F$
14 $eqmap[f] \leftarrow f'$
15 $G.F \leftarrow G.F \cup \{f'\}$
16 **for** each $v \in G.V$ where $(f, v) \in G.E$
17 **do** $G.E \leftarrow G.E \cup \{(f', v), (f', vmap[v])\}$
18 **for** each $v \in C$ where $v \in G.V$
19 **do** $assign[vmap[v]] \leftarrow eqmap[assign[v]]$
20 $f \leftarrow eqmap[f]$
21 **until** $match$
22 **return** $assign$

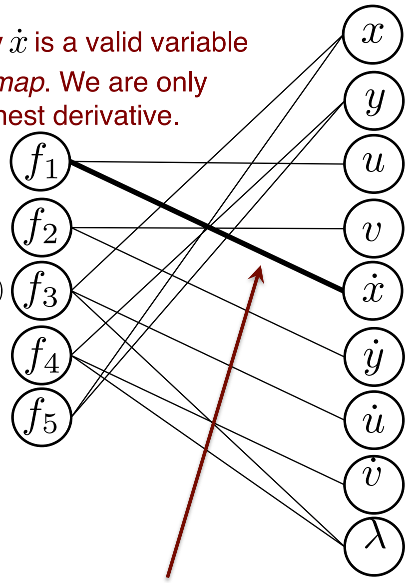
Try to find a match for equation f_1.

Pendulum

MATCH-EQUATION($G, f, \underline{C}, assign, vmap$)

$C \leftarrow C \cup \{f\}$
if there exits a $v \in G.V$ such that $(f, v) \in G.E$
  and $assign[v] =$ NIL and $vmap[v] =$ NIL
  then $assign[v] \leftarrow f$
    return TRUE
  else **for** each v where $(f, v) \in G.E$ and $v \notin C$
    and $vmap[v] =$ NIL
    do $C \leftarrow C \cup \{v\}$
      if MATCH-EQUATION($G, assign[v], \underline{C}, \underline{assign}, vmap$)
        then $assign[v] \leftarrow f$
          return TRUE
return FALSE

Note that only $\dot{x}$ is a valid variable because of *vmap*. We are only match for highest derivative.



State when returning from Match-Equation.

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$$
$$eqmap = \{\}$$
$$assign = \{\dot{x} \mapsto f_1\}$$
$$C = \{f_1\}$$

Matched variable to equation

Colored one equation

Algorithm: Pantelides

PANTELIDES($G, \underline{vmap}, \underline{eqmap}$)

$assign \leftarrow \emptyset$
for each $e \in G.F$
  do $f \leftarrow e$
    repeat
      $C \leftarrow \emptyset$
      $match \leftarrow$ MATCH-EQUATION($G, f, \underline{C}, assign, vmap$)
      if not $match$
        then for each $v \in C$ where $v \in G.V$
          do let v' be a vertex, such that $v' \notin G.V$
            $vmap[v] \leftarrow v'$
            $G.V \leftarrow G.V \cup \{v'\}$
        for each $f \in C$ where $f \in G.F$
          do let f' be a vertex, such that $f' \notin G.F$
            $eqmap[f] \leftarrow f'$
            $G.F \leftarrow G.F \cup \{f'\}$
            for each $v \in G.V$ where $(f, v) \in G.E$
              do $G.E \leftarrow G.E \cup \{(f', v), (f', vmap[v])\}$
        for each $v \in C$ where $v \in G.V$
          do $assign[vmap[v]] \leftarrow eqmap[assign[v]]$
        $f \leftarrow eqmap[f]$
    until $match$
return $assign$

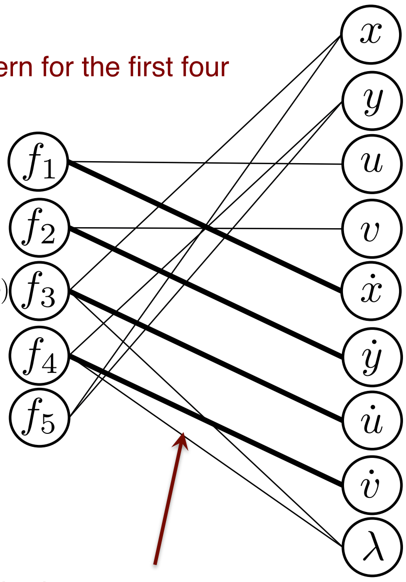
Function Match-Equation returns TRUE. Consequently, we break out of the repeat-until loop and proceeds with the next equation.

Pendulum

$\textsc{Match-Equation}(G, f, \underline{C}, assign, vmap)$

$C \leftarrow C \cup \{f\}$
if there exits a $v \in G.V$ such that $(f, v) \in G.E$
  and $assign[v] = \text{NIL}$ and $vmap[v] = \text{NIL}$
  then $assign[v] \leftarrow f$
    return TRUE
  else **for each** v where $(f, v) \in G.E$ and $v \notin C$
    and $vmap[v] = \text{NIL}$
      do $C \leftarrow C \cup \{v\}$
        if $\textsc{Match-Equation}(G, assign[v], \underline{C}, \underline{assign}, vmap)$
          then $assign[v] \leftarrow f$
            return TRUE
return FALSE

Same pattern for the first four equations.



State after matching for f_1, f_2, f_3, f_4

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$$
$$eqmap = \{\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$$
$$C = \{f_4\}$$

Matched 4 equations (could also have matched lambda).

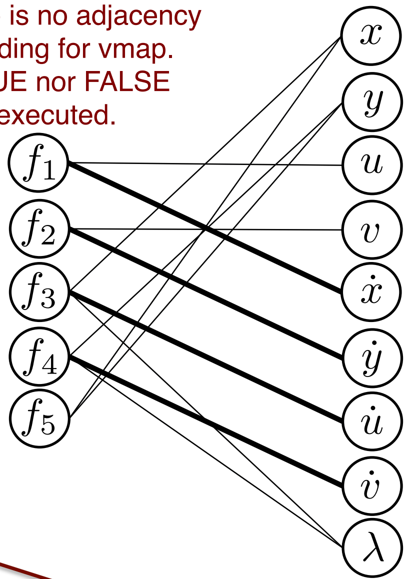
Note that only the last equation is colored because colors are cleared before matching.

Pendulum

$\textsc{Match-Equation}(G, f, \underline{C}, \underline{assign}, vmap)$

$C \leftarrow C \cup \{f\}$
if there exits a $v \in G.V$ such that $(f, v) \in G.E$
  and $assign[v] = \text{NIL}$ and $vmap[v] = \text{NIL}$
  then $assign[v] \leftarrow f$
    return TRUE
  else **for each** v where $(f, v) \in G.E$ and $v \notin C$
    and $vmap[v] = \text{NIL}$
      do $C \leftarrow C \cup \{v\}$
        if $\textsc{Match-Equation}(G, assign[v], \underline{C}, \underline{assign}, vmap)$
          then $assign[v] \leftarrow f$
            return TRUE
return FALSE

For f_5, there is no adjacency variable holding for vmap. Neither TRUE nor FALSE branch are executed.



State after matching f_5

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$$
$$eqmap = \{\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$$
$$C = \{f_5\}$$

Algorithm Match-Equation returns FALSE and returns with f_5 colored.

Algorithm: Pantelides

```
PANTELIDES(G, vmap, eqmap)
assign ← ∅
for each e ∈ G.F
    do f ← e
        repeat
                C ← ∅
                match ← MATCH-EQUATION(G, f, C, assign, vmap)
                if not match
                  then for each v ∈ C where v ∈ G.V
                            do let v′ be a vertex, such that v′ ∉ G.V
                                vmap[v] ← v′
                                G.V ← G.V ∪ {v′}
                       for each f ∈ C where f ∈ G.F
                            do let f′ be a vertex, such that f′ ∉ G.F
                                eqmap[f] ← f′
                                G.F ← G.F ∪ {f′}
                                for each v ∈ G.V where (f, v) ∈ G.E
                                    do G.E ← G.E ∪ {(f′, v), (f′, vmap[v])}
                       for each v ∈ C where v ∈ G.V
                            do assign[vmap[v]] ← eqmap[assign[v]]
                       f ← eqmap[f]
          until match
return assign
```

We have match = FALSE

No colored variables.

But we have one colored equation.

No colored variables.

Pendulum

State after matching f_5

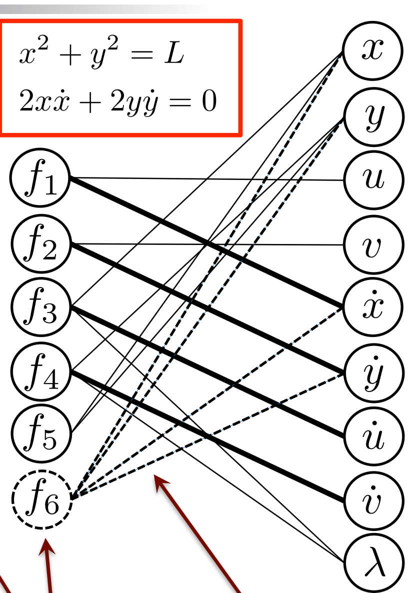
$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$$
$$eqmap = \{\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$$
$$C = \{f_5\}$$

```
for each f ∈ C where f ∈ G.F
    do let f′ be a vertex, such that f′ ∉ G.F
        eqmap[f] ← f′
        G.F ← G.F ∪ {f′}
        for each v ∈ G.V where (f, v) ∈ G.E
            do G.E ← G.E ∪ {(f′, v), (f′, vmap[v])}
```

State after creating differentiated equation.

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$$
$$eqmap = \{f_5 \mapsto f_6\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$$
$$C = \{f_5\}$$

$$x^2 + y^2 = L$$
$$2x\dot{x} + 2y\dot{y} = 0$$



Create a new equation node f_6 by differentiating f_5.

Create edges to variables and their derivatives.

Algorithm: Pantelides

PANTELIDES(G, $\underline{vmap}$, $\underline{eqmap}$)

$assign \leftarrow \emptyset$
for each $e \in G.F$
  do $f \leftarrow e$
    repeat
      $C \leftarrow \emptyset$
      $match \leftarrow$ MATCH-EQUATION($G, f, \underline{C}, \underline{assign}, vmap$)
      if not $match$
        then for each $v \in C$ where $v \in G.V$
            do let v' be a vertex, such that $v' \notin G.V$
              $vmap[v] \leftarrow v'$
              $G.V \leftarrow G.V \cup \{v'\}$
          for each $f \in C$ where $f \in G.F$
            do let f' be a vertex, such that $f' \notin G.F$
              $eqmap[f] \leftarrow f'$
              $G.F \leftarrow G.F \cup \{f'\}$
              for each $v \in G.V$ where $(f, v) \in G.E$
                do $G.E \leftarrow G.E \cup \{(f', v), (f', vmap[v])\}$
          for each $v \in C$ where $v \in G.V$
            do $assign[vmap[v]] \leftarrow eqmap[assign[v]]$
          $f \leftarrow eqmap[f]$
      until $match$
return $assign$

Repeat again (match was FALSE), but now with the differentiated equation f_6.

$\boxed{eqmap = \{f_5 \mapsto f_6\}}$

Pendulum

State before calling Match-Equation

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$
$eqmap = \{f_5 \mapsto f_6\}$
$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$
$C = \{\}$

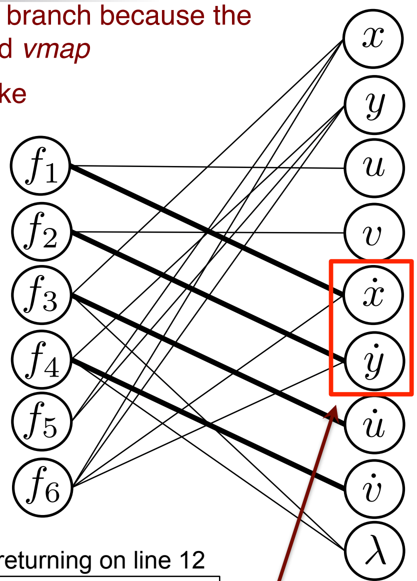
Cannot take TRUE branch because the states of *assign* and *vmap*

Color and make recursive call.

MATCH-EQUATION($G, f, \underline{C}, \underline{assign}, vmap$)

1 $C \leftarrow C \cup \{f\}$
2 **if** there exits a $v \in G.V$ such that $(f, v) \in G.E$
3   and $assign[v] =$ NIL and $vmap[v] =$ NIL
4   **then** $assign[v] \leftarrow f$
5   **return** TRUE
6   **else** **for** each v where $(f, v) \in G.E$ and $v \notin C$
7   and $vmap[v] =$ NIL
8   **do** $C \leftarrow C \cup \{v\}$
9   **if** MATCH-EQUATION($G, assign[v], \underline{C}, \underline{assign}, vmap$)
10   **then** $assign[v] \leftarrow f$
11   **return** TRUE
12 **return** FALSE

Returns false.



Two variables are applicable for the FALSE branch.

Before first recursive call at line 9

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$
$eqmap = \{f_5 \mapsto f_6\}$
$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$
$\boxed{C = \{f_6, \dot{x}\}}$

After all recursion when returning on line 12

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$
$eqmap = \{f_5 \mapsto f_6\}$
$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$
$\boxed{C = \{f_6, \dot{x}, f_1, \dot{y}, f_2\}}$

Algorithm: Pantelides

$\textsc{Pantelides}(G, \underline{vmap}, \underline{eqmap})$

$assign \leftarrow \emptyset$
for each $e \in G.F$
  do $f \leftarrow e$
    repeat
      $C \leftarrow \emptyset$
      $match \leftarrow \textsc{Match-Equation}(G, f, \underline{C}, \underline{assign}, vmap)$
      if not $match$
        then **for** each $v \in C$ where $v \in G.V$
          do let v' be a vertex, such that $v' \notin G.V$
            $vmap[v] \leftarrow v'$
            $G.V \leftarrow G.V \cup \{v'\}$
        for each $f \in C$ where $f \in G.F$
          do let f' be a vertex, such that $f' \notin G.F$
            $eqmap[f] \leftarrow f'$
            $G.F \leftarrow G.F \cup \{f'\}$
            for each $v \in G.V$ where $(f, v) \in G.E$
              do $G.E \leftarrow G.E \cup \{(f', v), (f', vmap[v])\}$
        for each $v \in C$ where $v \in G.V$
          do $assign[vmap[v]] \leftarrow eqmap[assign[v]]$
        $f \leftarrow eqmap[f]$
    until $match$
return $assign$

$$\boxed{C = \{f_6, \dot{x}, f_1, \dot{y}, f_2\}}$$

Differentiating equation two times

$$x^2 + y^2 = L$$
$$2x\dot{x} + 2y\dot{y} = 0$$
$$2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 = 0$$

First step: create new differentiated variables

Pendulum

State before creating new variables

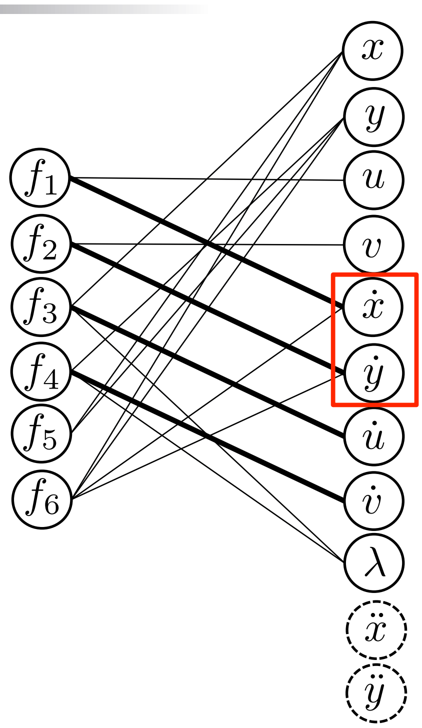
$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}\}$$
$$eqmap = \{f_5 \mapsto f_6\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$$
$$C = \{f_6, \boxed{\dot{x}}, f_1, \boxed{\dot{y}}, f_2\}$$

for each $v \in C$ where $v \in G.V$
  do let v' be a vertex, such that $v' \notin G.V$
    $vmap[v] \leftarrow v'$
    $G.V \leftarrow G.V \cup \{v'\}$

New variables and mapping

After adding new variables

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \boxed{\dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}}\}$$
$$eqmap = \{f_5 \mapsto f_6\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$$
$$C = \{f_6, \dot{x}, f_1, \dot{y}, f_2\}$$



Algorithm: Pantelides

```
PANTELIDES(G, vmap, eqmap)
assign ← ∅
for each e ∈ G.F
    do f ← e
        repeat
                C ← ∅
                match ← MATCH-EQUATION(G, f, C, assign, vmap)
                if not match
                  then for each v ∈ C where v ∈ G.V
                            do let v′ be a vertex, such that v′ ∉ G.V
                               vmap[v] ← v′
                               G.V ← G.V ∪ {v′}
                        for each f ∈ C where f ∈ G.F
                            do let f′ be a vertex, such that f′ ∉ G.F
                               eqmap[f] ← f′
                               G.F ← G.F ∪ {f′}
                               for each v ∈ G.V where (f, v) ∈ G.E
                                   do G.E ← G.E ∪ {(f′, v), (f′, vmap[v])}
                        for each v ∈ C where v ∈ G.V
                            do assign[vmap[v]] ← eqmap[assign[v]]
                        f ← eqmap[f]
          until match
return assign
```

Second step: create new differentiated equation nodes

Pendulum

State before creating equation nodes

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$

$eqmap = \{f_5 \mapsto f_6\}$

$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$

$C = \{\boxed{f_6}, \dot{x}, f_1, \dot{y}, f_2\}$

```
for each f ∈ C where f ∈ G.F
    do let f′ be a vertex, such that f′ ∉ G.F
       eqmap[f] ← f′
       G.F ← G.F ∪ {f′}
       for each v ∈ G.V where (f, v) ∈ G.E
           do G.E ← G.E ∪ {(f′, v), (f′, vmap[v])}
```

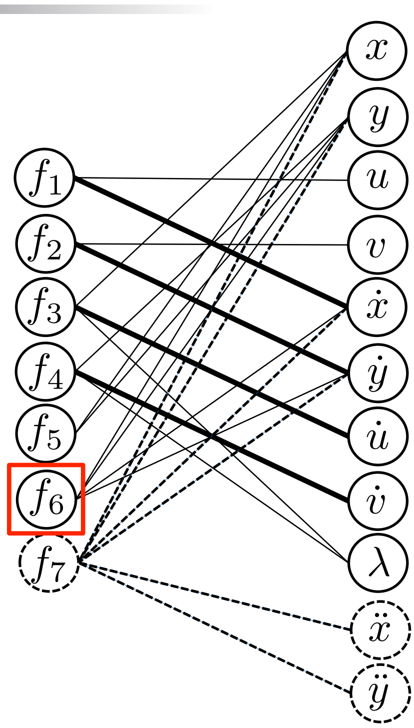
After adding equation f_6

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$

$eqmap = \{f_5 \mapsto f_6, \boxed{f_6 \mapsto f_7}\}$

$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$

$C = \{f_6, \dot{x}, f_1, \dot{y}, f_2\}$



Pendulum

State before creating equation nodes

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$

$eqmap = \{f_5 \mapsto f_6\}$

$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$

$C = \{\boxed{f_6}, \dot{x}, \boxed{f_1}, \dot{y}, \boxed{f_2}\}$

for each $f \in C$ where $f \in G.F$
    do let f' be a vertex, such that $f' \notin G.F$
        $eqmap[f] \leftarrow f'$
        $G.F \leftarrow G.F \cup \{f'\}$
        for each $v \in G.V$ where $(f, v) \in G.E$
            do $G.E \leftarrow G.E \cup \{(f', v), (f', vmap[v])\}$

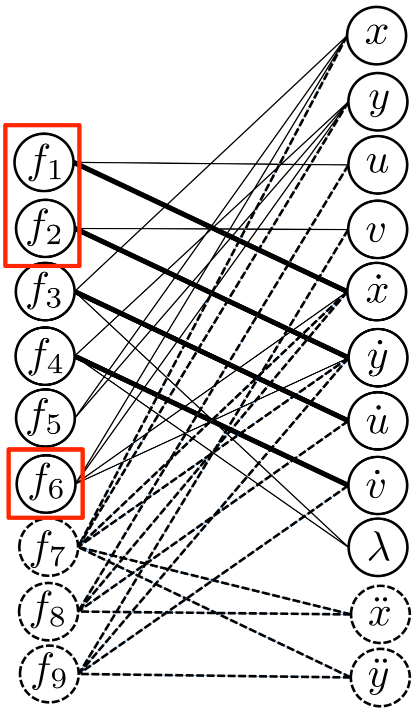
After adding all equations

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$

$eqmap = \{f_5 \mapsto f_6 \boxed{f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\}}$

$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$

$C = \{f_6, \dot{x}, f_1, \dot{y}, f_2\}$



Algorithm: Pantelides

PANTELIDES$(G, vmap, eqmap)$

```
assign ← ∅
for each e ∈ G.F
    do f ← e
        repeat
            C ← ∅
            match ← MATCH-EQUATION(G, f, C, assign, vmap)
            if not match
              then for each v ∈ C where v ∈ G.V
                        do let v′ be a vertex, such that v′ ∉ G.V
                           vmap[v] ← v′
                           G.V ← G.V ∪ {v′}
                   for each f ∈ C where f ∈ G.F
                        do let f′ be a vertex, such that f′ ∉ G.F
                           eqmap[f] ← f′
                           G.F ← G.F ∪ {f′}
                           for each v ∈ G.V where (f, v) ∈ G.E
                              do G.E ← G.E ∪ {(f′, v), (f′, vmap[v])}
                   for each v ∈ C where v ∈ G.V
                        do assign[vmap[v]] ← eqmap[assign[v]]
                   f ← eqmap[f]
        until match
return assign
```

Third step: assign variables to equations for new variables.

Pendulum

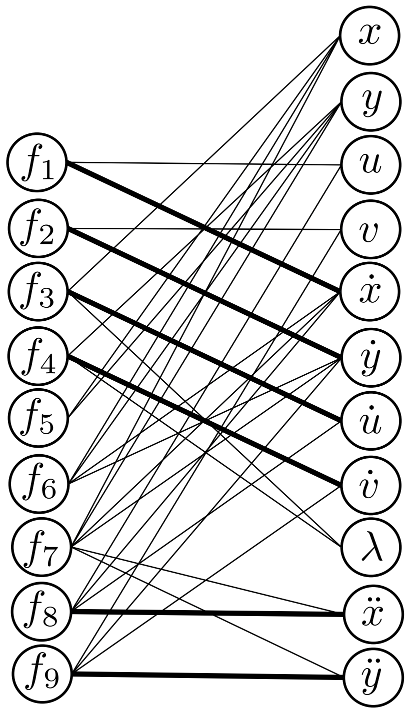
After adding all equations

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$$
$$eqmap = \{f_5 \mapsto f_6, f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4\}$$
$$C = \{f_6, \boxed{\dot{x},} f_1, \boxed{\dot{y},} f_2\}$$

for each $v \in C$ where $v \in G.V$
do $assign[vmap[v]] \leftarrow eqmap[assign[v]]$

After adding new assignments

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$$
$$eqmap = \{f_5 \mapsto f_6, f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4, \boxed{\ddot{x} \mapsto f_8, \ddot{y} \mapsto f_9\}}$$
$$C = \{f_6, \dot{x}, f_1, \dot{y}, f_2\}$$



Algorithm: Pantelides

```
PANTELIDES(G, vmap, eqmap)
  assign ← ∅
  for each e ∈ G.F
      do f ← e
         repeat
               C ← ∅
               match ← MATCH-EQUATION(G, f, C, assign, vmap)
               if not match
                 then for each v ∈ C where v ∈ G.V
                           do let v' be a vertex, such that v' ∉ G.V
                              vmap[v] ← v'
                              G.V ← G.V ∪ {v'}
                      for each f ∈ C where f ∈ G.F
                           do let f' be a vertex, such that f' ∉ G.F
                              eqmap[f] ← f'
                              G.F ← G.F ∪ {f'}
                              for each v ∈ G.V where (f, v) ∈ G.E
                                  do G.E ← G.E ∪ {(f', v), (f', vmap[v])}
                      for each v ∈ C where v ∈ G.V
                           do assign[vmap[v]] ← eqmap[assign[v]]
                      f ← eqmap[f]
           until match
  return assign
```

Repeat again with second differentiated version of equation five.

Pendulum

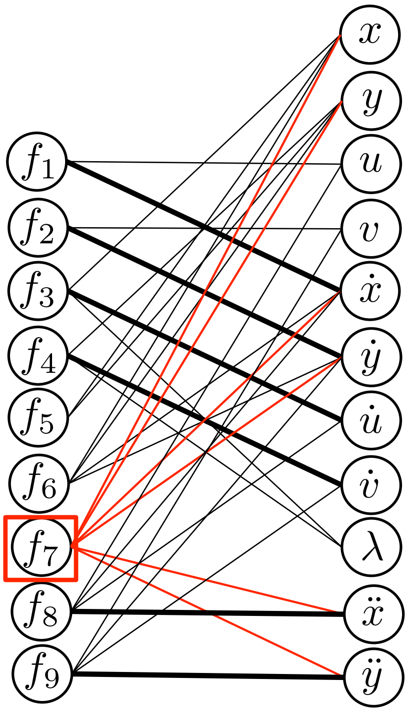
Before matching

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$$
$$eqmap = \{f_5 \mapsto f_6, f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4, \ddot{x} \mapsto f_8, \ddot{y} \mapsto f_9\}$$

MATCH-EQUATION$(G, f, \underline{C}, \underline{assign}, vmap)$

$C \leftarrow C \cup \{f\}$
if there exits a $v \in G.V$ such that $(f, v) \in G.E$
  and $assign[v] = \text{NIL}$ and $vmap[v] = \text{NIL}$
  then $assign[v] \leftarrow f$
    return TRUE
  else **for** each v where $(f, v) \in G.E$ and $v \notin C$
    and $vmap[v] = \text{NIL}$
      do $C \leftarrow C \cup \{v\}$
        if MATCH-EQUATION$(G, assign[v], \underline{C}, \underline{assign}, vmap)$
          then $assign[v] \leftarrow f$
            return TRUE
return FALSE

For clarity: view variables and edges where vmap[v] = NIL



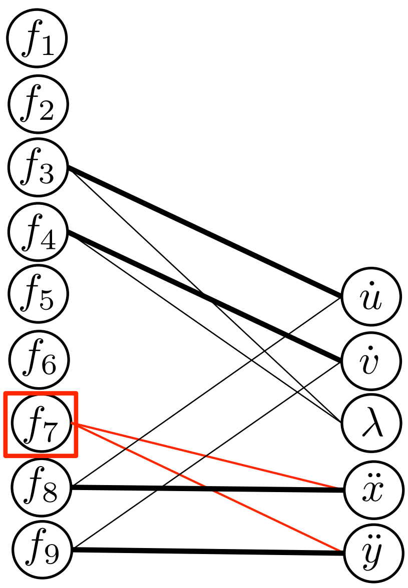
Pendulum

Before matching

$$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$$
$$eqmap = \{f_5 \mapsto f_6, f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\}$$
$$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4, \ddot{x} \mapsto f_8, \ddot{y} \mapsto f_9\}$$

MATCH-EQUATION$(G, f, \underline{C}, \underline{assign}, vmap)$

$C \leftarrow C \cup \{f\}$
if there exits a $v \in G.V$ such that $(f, v) \in G.E$
  and $assign[v] = \text{NIL}$ and $vmap[v] = \text{NIL}$
  then $assign[v] \leftarrow f$
    return TRUE
  else **for** each v where $(f, v) \in G.E$ and $v \notin C$
    and $vmap[v] = \text{NIL}$
      do $C \leftarrow C \cup \{v\}$
        if MATCH-EQUATION$(G, assign[v], \underline{C}, \underline{assign}, vmap)$
          then $assign[v] \leftarrow f$
            return TRUE
return FALSE



Pendulum

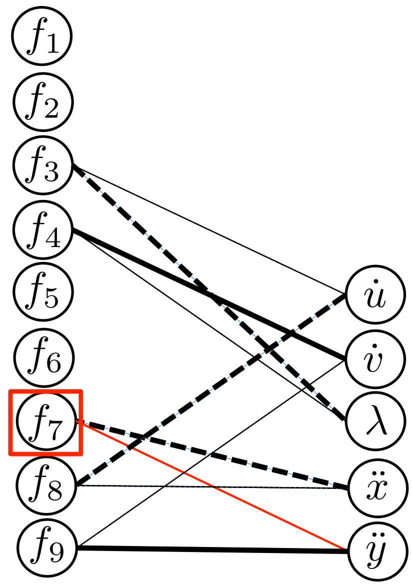
Before matching

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$
$eqmap = \{f_5 \mapsto f_6, f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\}$
$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_3, \dot{v} \mapsto f_4,$
$\ddot{x} \mapsto f_8, \ddot{y} \mapsto f_9\}$

$\text{MATCH-EQUATION}(G, f, \underline{C}, \underline{assign}, vmap)$
$C \leftarrow C \cup \{f\}$
if there exits a $v \in G.V$ such that $(f, v) \in G.E$
  and $assign[v] = \text{NIL}$ and $vmap[v] = \text{NIL}$
  then $assign[v] \leftarrow f$
    return TRUE
  else **for** each v where $(f, v) \in G.E$ and $v \notin C$
    and $vmap[v] = \text{NIL}$
      do $C \leftarrow C \cup \{v\}$
        if $\text{MATCH-EQUATION}(G, assign[v], \underline{C}, \underline{assign}, vmap)$
          then $assign[v] \leftarrow f$
            return TRUE
return FALSE

$vmap = \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\}$
$eqmap = \{f_5 \mapsto f_6, f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\}$
$assign = \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_8, \dot{v} \mapsto f_4,$
$\lambda \mapsto f_3, \ddot{x} \mapsto f_7, \ddot{y} \mapsto f_9\}$
$C = \{f_7, \ddot{x}, f_8, \dot{u}, f_3\}$

Successful match!



Algorithm: Pantelides

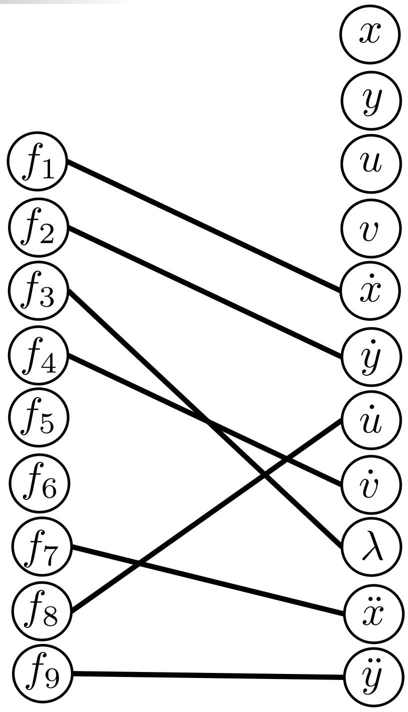
$\text{PANTELIDES}(G, \underline{vmap}, \underline{eqmap})$
$assign \leftarrow \emptyset$
for each $e \in G.F$
  do $f \leftarrow e$
    repeat
      $C \leftarrow \emptyset$
      $match \leftarrow \text{MATCH-EQUATION}(G, f, \underline{C}, \underline{assign}, vmap)$
      if not $match$
        then **for** each $v \in C$ where $v \in G.V$
            do let v' be a vertex, such that $v' \notin G.V$
              $vmap[v] \leftarrow v'$
              $G.V \leftarrow G.V \cup \{v'\}$
          for each $f \in C$ where $f \in G.F$
            do let f' be a vertex, such that $f' \notin G.F$
              $eqmap[f] \leftarrow f'$
              $G.F \leftarrow G.F \cup \{f'\}$
              for each $v \in G.V$ where $(f, v) \in G.E$
                do $G.E \leftarrow G.E \cup \{(f', v), (f', vmap[v])\}$
          for each $v \in C$ where $v \in G.V$
            do $assign[vmap[v]] \leftarrow eqmap[assign[v]]$
          $f \leftarrow eqmap[f]$
    until $match$
return $assign$

Last equation and successful match.
Algorithm terminates.

Result of Pantelides on Pendulum

$$
\begin{aligned}
vmap &= \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\} \\
eqmap &= \{f_5 \mapsto f_6, f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\} \\
assign &= \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_8, \dot{v} \mapsto f_4, \\
&\qquad \lambda \mapsto f_3, \ddot{x} \mapsto f_7, \ddot{y} \mapsto f_9\}
\end{aligned}
$$

(1)	$\dot{x} = u$	$f_1(\dot{x}, u) = 0$
(2)	$\dot{y} = v$	$f_2(\dot{y}, v) = 0$
(3)	$\dot{u} = \lambda \cdot x$	$f_3(\dot{u}, \lambda, x) = 0$
(4)	$\dot{v} = \lambda \cdot y - g$	$f_4(\dot{v}, \lambda, y) = 0$
(5)	$x^2 + y^2 = L$	$f_5(x, y) = 0$
(6)	$2x\dot{x} + 2y\dot{y} = 0$	$f_6(x, \dot{x}, y, \dot{y}) = 0$
(7)	$2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 = 0$	$f_7(x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}) = 0$
(8)	$\ddot{x} = \dot{u}$	$f_8(\ddot{x}, \dot{u}) = 0$
(9)	$\ddot{y} = \dot{v}$	$f_9(\ddot{y}, \dot{v}) = 0$



Result of Pantelides on Pendulum

$$
\begin{aligned}
vmap &= \{x \mapsto \dot{x}, y \mapsto \dot{y}, u \mapsto \dot{u}, v \mapsto \dot{v}, \dot{x} \mapsto \ddot{x}, \dot{y} \mapsto \ddot{y}\} \\
eqmap &= \{f_5 \mapsto f_6, f_6 \mapsto f_7, f_1 \mapsto f_8, f_2 \mapsto f_9\} \\
assign &= \{\dot{x} \mapsto f_1, \dot{y} \mapsto f_2, \dot{u} \mapsto f_8, \dot{v} \mapsto f_4, \\
&\qquad \lambda \mapsto f_3, \ddot{x} \mapsto f_7, \ddot{y} \mapsto f_9\}
\end{aligned}
$$

Is the system of equations solvable if we replace the old equations with their differentiated version?

(1) $\dot{x} = u$

(2) $\dot{y} = v$

(3) $\dot{u} = \lambda \cdot x$

(4) $\dot{v} = \lambda \cdot y - g$

(5) $x^2 + y^2 = L$

(6) $2x\dot{x} + 2y\dot{y} = 0$

(7) $2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 = 0$

(8) $\ddot{x} = \dot{u}$

(9) $\ddot{y} = \dot{v}$

By substituting (8) and (9) we have

$$\ddot{x} = \lambda \cdot x$$
$$\ddot{y} = \lambda \cdot y - g$$
$$2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 = 0$$

Which is solvable for highest derivative

$$
\begin{array}{c|ccc}
 & \lambda & \ddot{y} & \ddot{x} \\
\hline
f_1 & \mathbf{1} & 0 & \mathbf{1} \\
f_2 & \mathbf{1} & \mathbf{1} & 0 \\
f_3 & 0 & \mathbf{1} & \mathbf{1}
\end{array}
$$

Same result if converted into order one equation

$$
\begin{array}{c|ccccc}
 & \dot{y} & \dot{x} & \dot{u} & \lambda & \dot{v} \\
\hline
f_2 & \mathbf{1} & 0 & 0 & 0 & 0 \\
f_1 & 0 & \mathbf{1} & 0 & 0 & 0 \\
f_5 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} \\
f_3 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 \\
f_4 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1}
\end{array}
$$

Part IV

Dummy Derivatives

Index Reduction

Should differentiated equations from Pantelides be used for index reduction?

$$\begin{aligned} \ddot{x} &= \lambda \cdot x \\ \ddot{y} &= \lambda \cdot y - g \\ 2x\ddot{x} + 2\dot{x}^2 + 2y\ddot{y} + 2\dot{y}^2 &= 0 \end{aligned}$$

The reduced problem (index-1) is mathematically correct, but since equation

$$x^2 + y^2 = L$$

is not present, numerical approximation gives a "drifting problem". In our example, the pendulum's length will grow...

Dummy Derivative

Basic Idea:

- Include all differentiated equations
- For each equation, introduce a “dummy derivative” variable.

$$
\begin{aligned}
&\ddot{x} = \lambda \cdot x \\
&y'' = \lambda \cdot y - g \\
&x^2 + y^2 = L \\
&2x\dot{x} + 2yy' = 0 \\
&2x\ddot{x} + 2\dot{x}^2 + 2yy'' + 2y'^2 = 0
\end{aligned}
$$

All constraints are present and the number of equations and unknowns match.

The actual algorithm is presented by Mattson and Söderlind (1993)

References and Further Reading

Iain S. Duff. On Algorithms for Obtaining a Maximum Transversal. ACM Transactions on Mathematical Software, 7(3):315–330, 1981.

Iain S. Duff and John K. Reid. An Implementation of Tarjan’s Algorithm for the Block Triangularization of a Matrix. ACM Transactions on Mathe- matical Software, 4(2):137–147, 1978.

S. E. Mattsson, H. Olsson, and H. Elmqvist. Dynamic selection of states in fymola. In Proceedings of the Modelica Workshop, pages 61–67, 2000.

S. E. Mattsson and G. Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. SIAM Journal on Scientific Computing, 14(3):677–692, 1993.

C. C. Pantelides. The Consistent Initialization of Differential-Algebraic Sys- tems. SIAM Journal on Scientific and Statistical Computing, 9(2):213–231, 1988.

R. Tarjan. Depth-first search and linear graph algorithms. SIAM Journal on Computing, 1(2):146–160, 1972.

Summary and Conclusions

Summary and Conclusions

Some key take away points:

- **Matching** finds a mapping between variables and equations. Used both in BLT sorting and Pantelides algorithm.
- **BLT** sorts blocks of equation, where each block represents an algebraic loop. Uses matching and Tarjan’s algorithm.
- **Pantelides algorithm** determine the subset of equations that needs to be differentiated.
- The **Dummy Derivative method** perform correct index reduction and avoids the drifting problem.

Thanks for listening!