Building Computational Skills for Mathematical Modeling in Science and Engineering through an Interdisciplinary Elective Course

Dr. Ashlee N. Ford Versypt, Ph.D.

Assistant Professor

Oklahoma State University

College of Engineering, Architecture & Technology

School of Chemical Engineering

tinyurl.com/ashleefv

@FordVersyptLab

ashleefv@okstate.edu

Motivation for CHE 4753/5753 elective course (1st day)

- Teaching best practices for using and building state-of-the-art computer simulations for scientific research
- Providing hands-on experience in applying content from more theoretical numerical methods or modeling courses
- Promoting reproducible science with emphasis on version control and documentation
- Sharing passion for computational science & engineering, knowledge, and tips gained over 15+ years of training and practice

Inspiration for course design: Software Carpentry

software-carpentry.org

software carpentry

Teaching basic lab skills
for research computing

Our Workshops

A Software Carpentry workshop is a hands-on, two-day event that covers the core skills needed to be productive in a small research team. Short tutorials alternate with practical exercises, and all instruction is done via live coding. *All workshop participants are required to abide by our code of conduct to ensure that all attendees to have an enjoyable and fulfilling experience.*

About Us

Since 1998, Software Carpentry has been teaching researchers the computing skills they need to get more done in less time and with less pain. Our volunteer instructors have run hundreds of events for more than 34,000 researchers since 2012. All of our lesson materials are freely reusable under the Creative Commons - Attribution license.

The Software Carpentry Foundation and its sibling lesson project, Data Carpentry, have merged to become The Carpentries, a fiscally sponsored project of Community Initiatives, a 501(c)3 non-profit incorporated in the United States. See the staff page for The Carpentries.

Course emphasis

- NOT to replicate content from 1^{st} programming course or Numerical Methods
- Accelerate on boarding of new grad students (ChE, MAE, Chem, Math, CIVE, Physics)
- Computational skill development
  - solving realistic continuum scale science and engineering problems
  - managing open source code projects, and
  - disseminating computational research results through scientific documentation and publications.

Course learning objectives

1. utilize **Git** for version control using common commands: status, add, commit, push

Git

MINGW64:/c/research/arthritis

```
Ashlee@Sunshine MINGW64 ~
$ cd /c/research/

Ashlee@Sunshine MINGW64 /c/research
$ cd arthritis

Ashlee@Sunshine MINGW64 /c/research/arthritis (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    Combined_Bangsgaard_and_Stepensky_Model.asv

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Tb.png

no changes added to commit (use "git add" and/or "git commit -a")

Ashlee@Sunshine MINGW64 /c/research/arthritis (master)
$ git add Tb.png

Ashlee@Sunshine MINGW64 /c/research/arthritis (master)
$ git commit -m "immage file added"
[master 2a995dc] immage file added
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Tb.png

Ashlee@Sunshine MINGW64 /c/research/arthritis (master)
$ git pull origin master
From https://bitbucket.org/ashleefv/arthritis
 * branch            master     -> FETCH_HEAD
Already up-to-date.

Ashlee@Sunshine MINGW64 /c/research/arthritis (master)
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 48.14 KiB | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://bitbucket.org/ashleefv/arthritis.git
   59664c1..2a995dc  master -> master
```

master

Filter files

/

Name	Size	Last commit	Message
CompAssn1		2018-10-02	assignment 3 submission
CompAssn2		2018-09-09	assignment 2 submission
CompAssn3		2018-10-02	assignment 3 submission
CompAssn4		2018-11-14	removed unneeded file
CompAssn5		2018-11-14	assignment 5 submission
CompAssn6		2018-12-11	assignment 6 submission
README.md	2.56 KB	2018-08-22	Initial commit

Commits

All branches

Find commits

Author	Commit	Message	Date
Natalia DaSilva	4c74944	assignment 6 submission	2018-12-11
Natalia DaSilva	ad5ada7	GUIs complete. need to verify	2018-12-10
Natalia DaSilva	ad41605	all guis created. not working on solving the ODEs and plotting	2018-12-09
Natalia DaSilva	2853858	all Gui windows completed. I can inout any number of equations and they will display in list box.	2018-12-09
Natalia DaSilva	1956bb1	comp assignment 6 started. main gui and equations gui created	2018-12-07
Natalia DaSilva	e20a1f7	assignment 5 submission	2018-11-14
Natalia DaSilva	c370d75	removed unneeded file	2018-11-14
Natalia DaSilva	4204f54	final files	2018-11-14
Natalia DaSilva	b9d0861	final stages completed. Just need to package app	2018-11-14
Natalia DaSilva	f9c1810	finished all code. Need to add title and documentation	2018-11-14
Natalia DaSilva	9a1b915	just need to change the length of the simulation.	2018-11-14
Natalia DaSilva	24966ef	creating files and starting the GUI	2018-11-13
Natalia DaSilva	8599cfe	Assignment 4 submission	2018-10-24

CompAssn6/mainGUIcomassn6.fig **MODIFIED** View fi

Binary file modified.

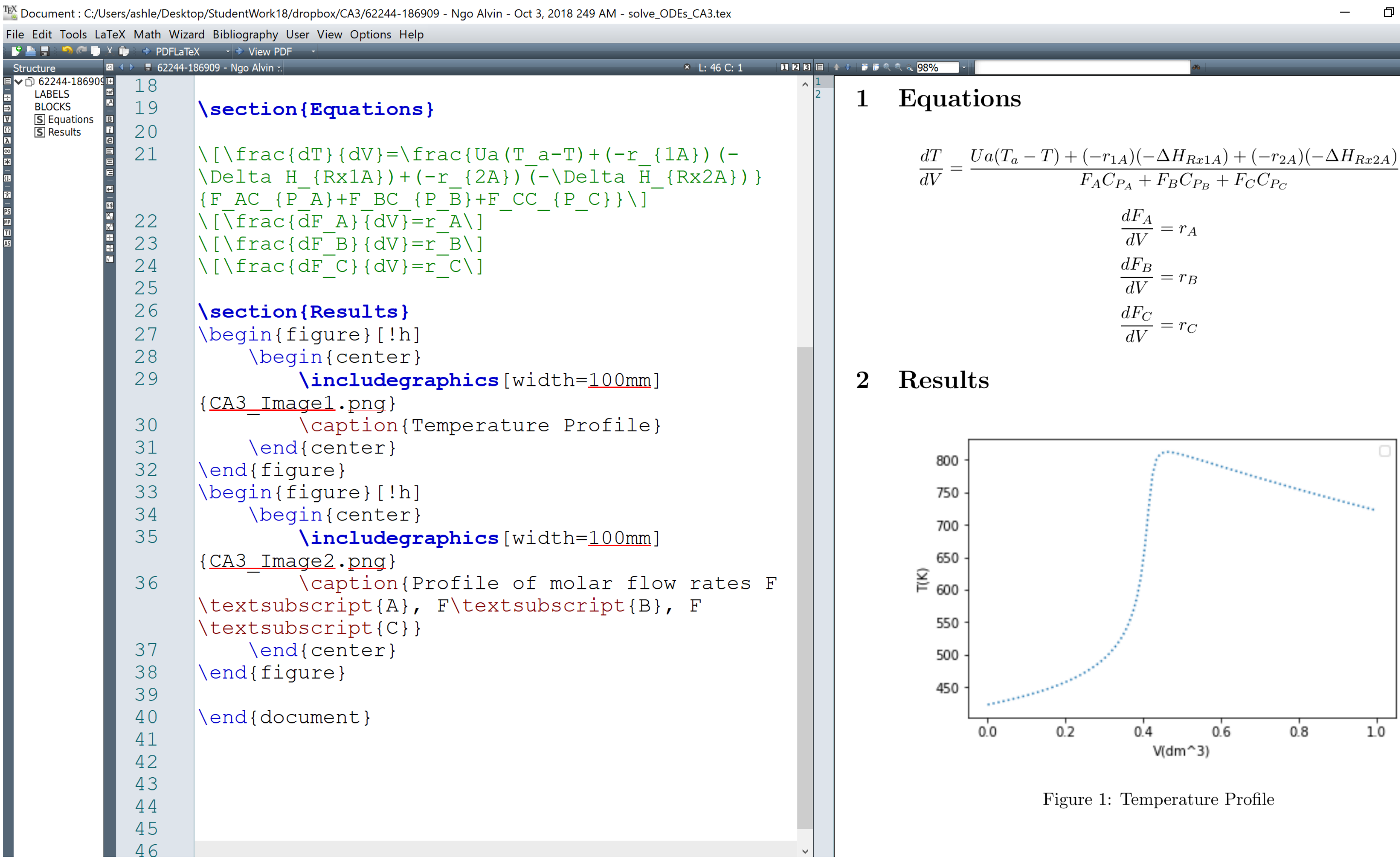
CompAssn6/mainGUIcomassn6.m **MODIFIED** Side-by-side diff View fi

```
...

 % Edit the above text to modify the response to help mainGUIcomassn6

-% Last Modified by GUIDE v2.5 09-Dec-2018 15:58:23
+% Last Modified by GUIDE v2.5 10-Dec-2018 16:40:51

 % Begin initialization code - DO NOT EDIT
 gui_Singleton = 1;
...
 % hObject    handle to pushbutton1 (see GCBO)
 % eventdata  reserved - to be defined in a future version of MATLAB
 % handles    structure with handles and user data (see GUIDATA)
+clear handles.num_algEquations handles.num_diffEquations handles.numerator handles.denominator handles.initial_value handles.diff_equation handles.explicit_equation
 enterequationsGUI
 uiwait
 load('amountofequations.mat');
...
 for k = 1:num_algEquations
     enterALGgui
     uiwait
-    load('ALG_equation.mat')
+    load('ALG_equation.mat');
     handles.explicit_equation{k} = explicit_equation;
 end

...
     DIFFequationstring{i} = strcat('d',handles.numerator{i}, '/d', handles.denominator{i}, ' = ', handles.diff_equation{i});
 end

-set(handles.listbox1, 'String', DIFFequationstring)
+set(handles.listbox1, 'String', DIFFequationstring);

```

Course learning objectives

2. write scientific reports and similar documents in the **LaTeX** typesetting language with TeXMaker as the editor using an article template and include equations, figures, tables, document hierarchy, cross referencing, and citations (using BibTeX) in the documents

LaTeX & TeXMaker



Course learning objectives

3. use basic Unix commands to run programs, navigate and organize a file system, and **access the university's supercomputing cluster**

New User Tutorial

Welcome to the OSU HPCC New User Tutorial. This online tutorial is divided into two parts.

- Part 1 will cover how to log in to Cowboy and the basics of navigating a linux system.
- Part 2 will go over how to schedule your jobs on Cowboy and submit them.

Download a PDF copy of the New User Tutorial HERE.

To schedule one-on-one training or ask a question email hpcc@okstate.edu.

Part 1: Navigating Linux

Course learning objectives

4. use **best practices** for computational problem solving and research and scientific computing as described in publications provided as **assigned readings**

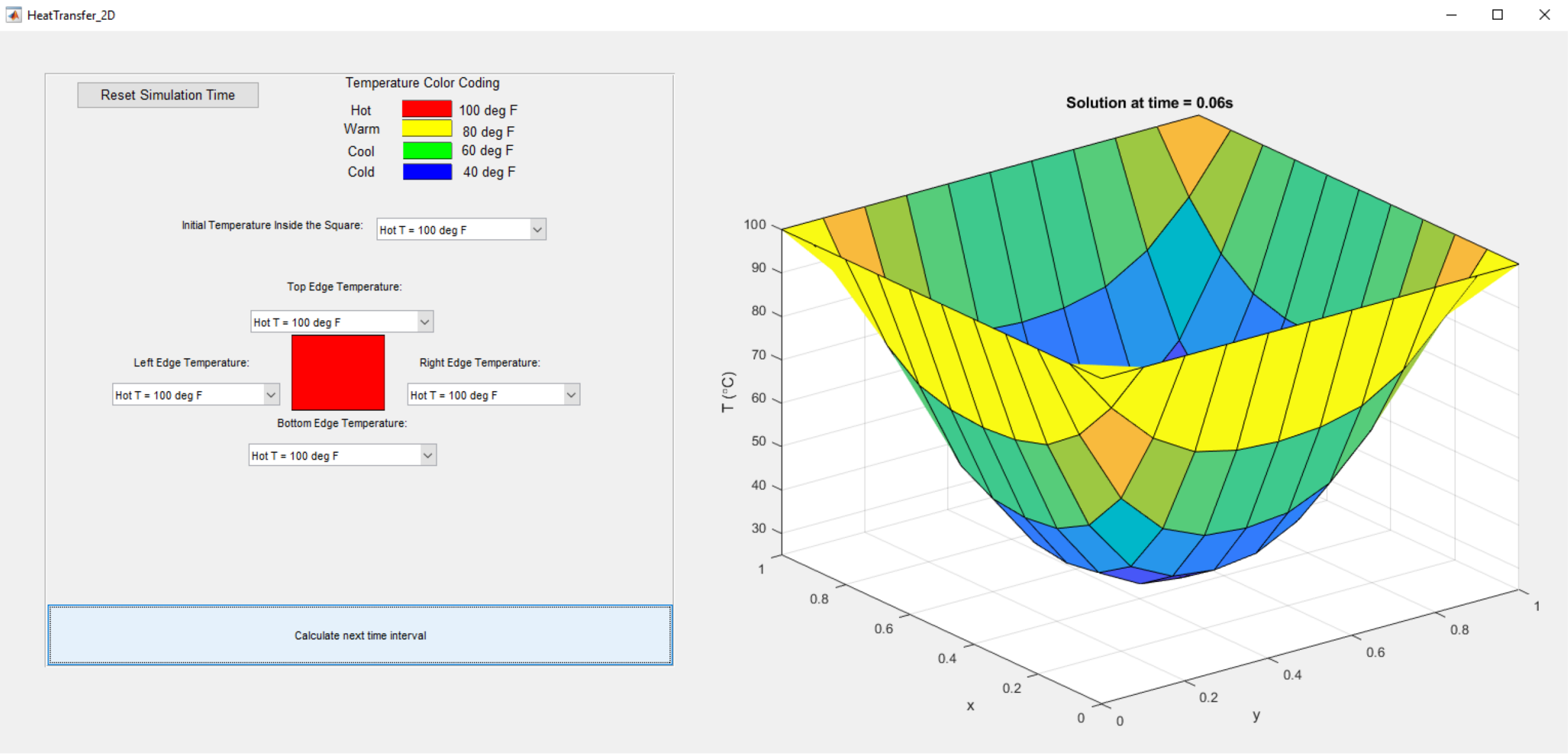
10% of grade

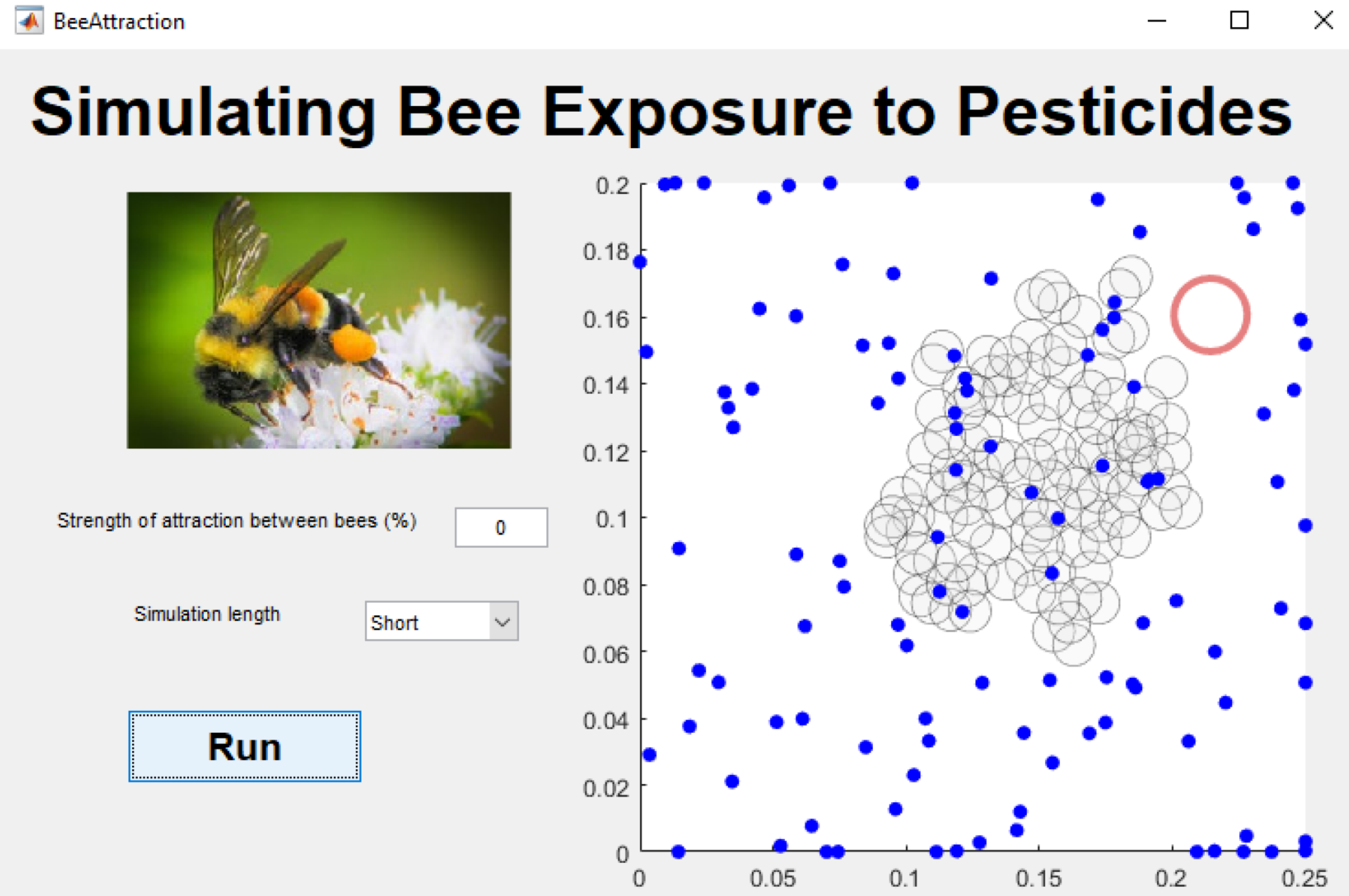
Readings from textbooks, websites, and articles to flip the lecture portion

1. Background and overview information on LaTeX for mathematical typesetting
2. Best practices for software engineering in scientific computing
3. Basics of Python programming
4. Using built-in functions in MATLAB and Python for solving common classes of problems in scientific computing with established numerical methods, focusing on nonlinear equations, numerical integration, and ordinary differential equations (ODEs)
5. Python modules NumPy and SciPy
6. Parameter estimation by linear and nonlinear least squares regression
7. Sensitivity analysis
8. Graphical user interfaces (GUIs) for scientific computing in MATLAB and Python
9. Verification and validation in scientific computing
10. Reproducible research computing and other tips for sharing figures, code, and documentation from computational projects

Course learning objectives

5. develop **graphical user interfaces** for interactive model reuse


HeatTransfer_2D
Reset Simulation Time
Temperature Color Coding
Hot 100 deg F
Warm 80 deg F
Cool 60 deg F
Cold 40 deg F
Initial Temperature Inside the Square:
Hot T = 100 deg F
Top Edge Temperature:
Hot T = 100 deg F
Left Edge Temperature:
Hot T = 100 deg F
Right Edge Temperature:
Hot T = 100 deg F
Bottom Edge Temperature:
Hot T = 100 deg F
Calculate next time interval
Solution at time = 0.06s
T (°C)
x
y


BeeAttraction
Simulating Bee Exposure to Pesticides
Strength of attraction between bees (%)
0
Simulation length
Short
Run
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
0.05
0.1
0.15
0.2
0.25

Course learning objectives

6. program well-documented, readable code in the high-level languages of **Python and MATLAB** that uses libraries, built-in functions, and user-defined functions
   – to solve systems of linear and nonlinear equations,
   – to numerically integrate functions and data,
   – to solve systems of ordinary and partial differential equations,
   – to estimate parameters for mathematical models using optimization and data fitting tools,
   – to calculate statistics of data and stochastic processes, and
   – to create publication quality figures

Class periods

- 75 minutes twice weekly
- Discussion of readings
- Paired programming challenges (ungraded but related to key concepts for the assignments)
- Code samples/worked examples
- Demonstrating my process for writing code in real time, debugging, and looking through documentation

Homeworks & video assignment

- Context driven for engr and science research needs
- Not artificial canonical, “toy” problems
  - **If you have problems you want GUIs for, my class may be able to design them**
- Complexity increases with topics scaffolded onto previous assignment objectives

- https://flipgrid.com/s/f1d267ea79e8

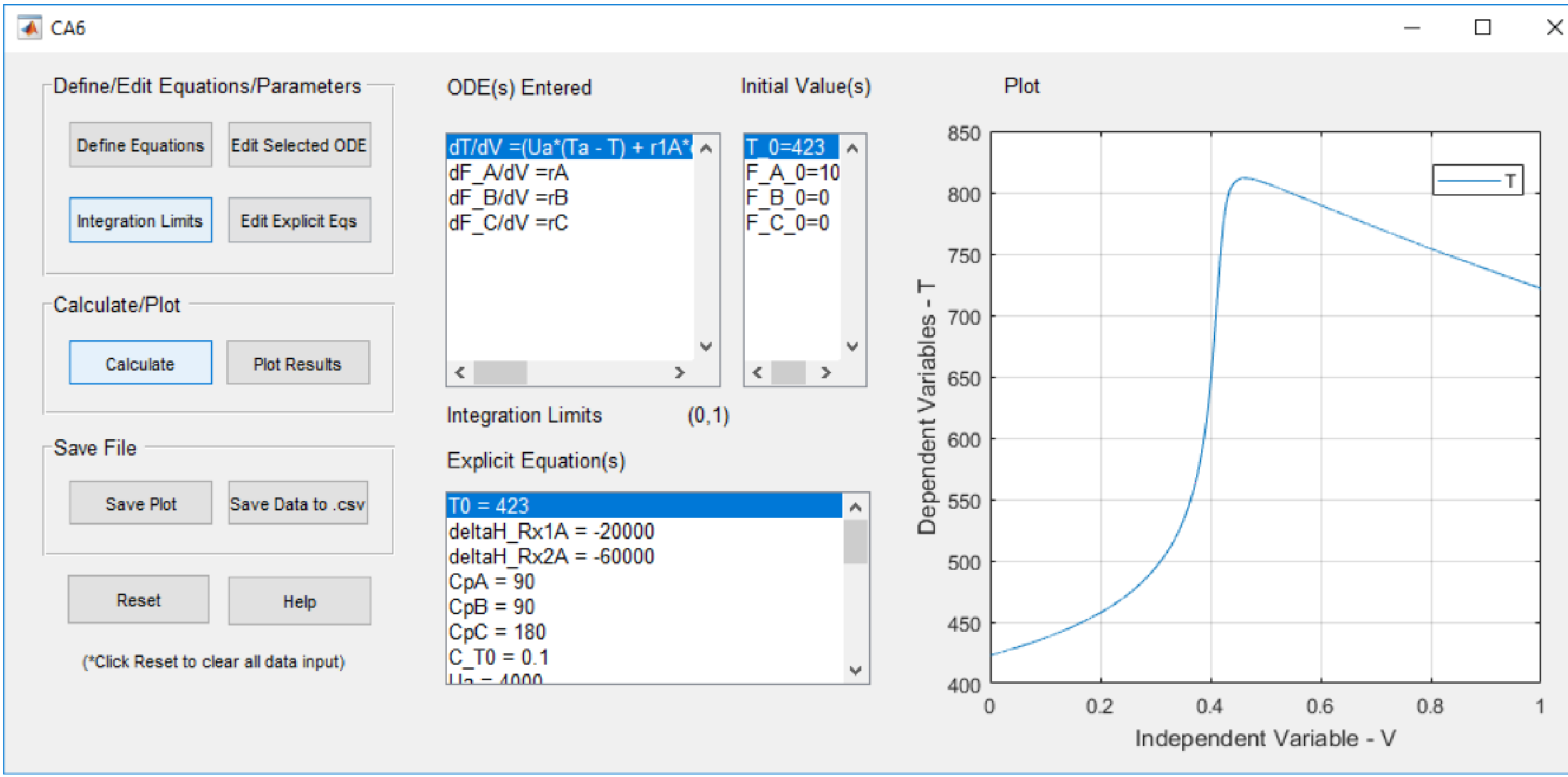
  Starting at 3:41

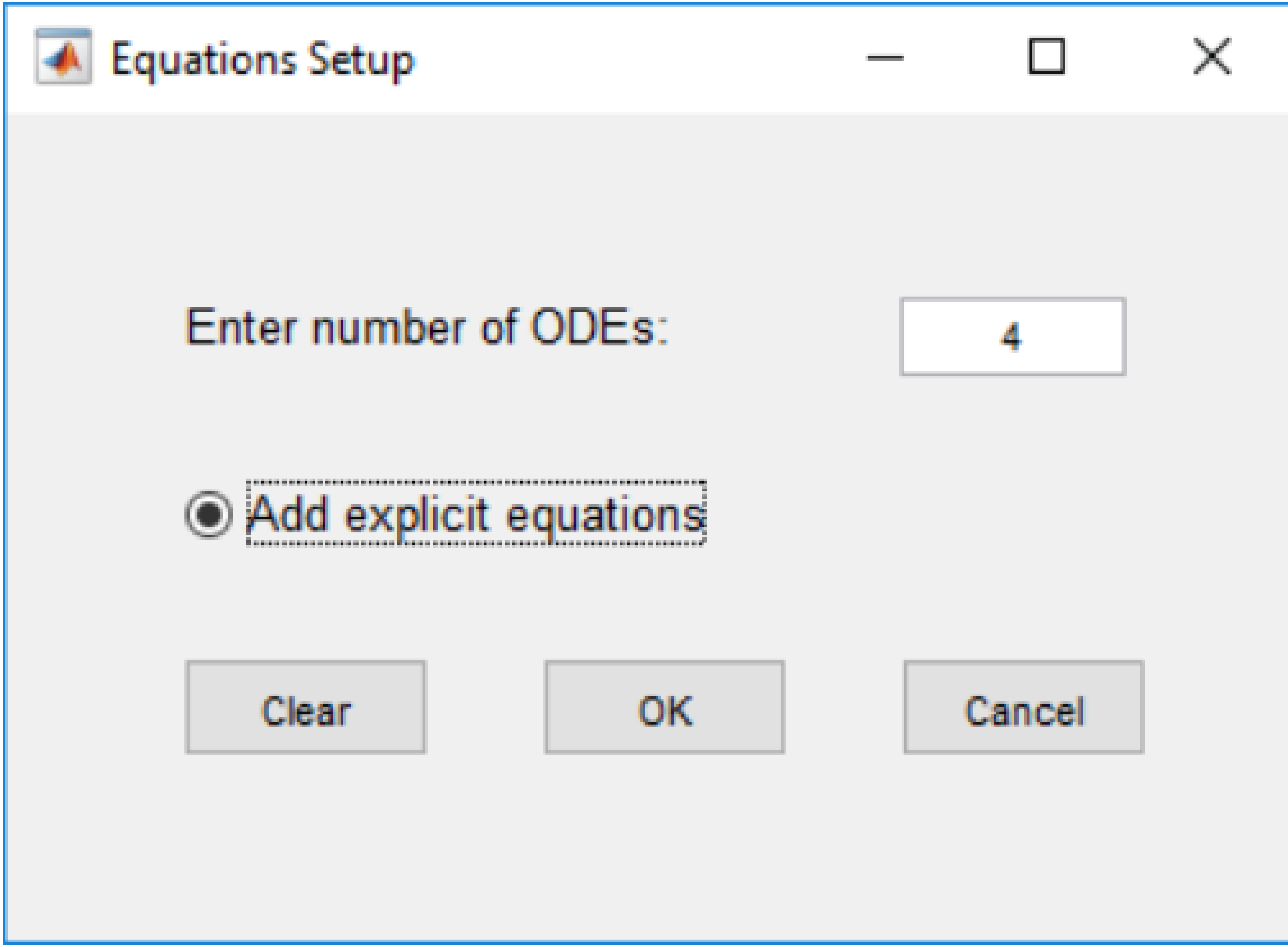
Final assignment/course project

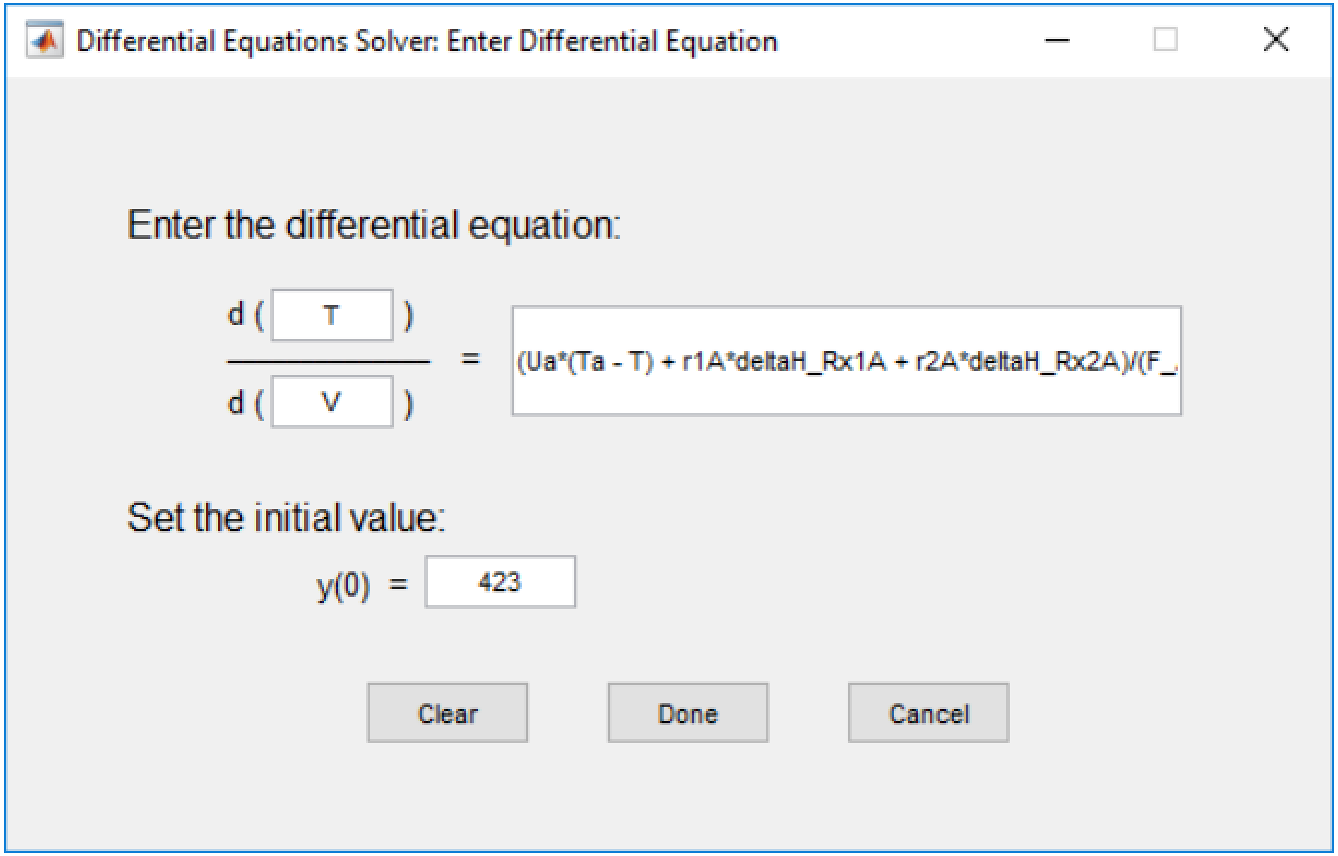
- Verification with old homeworks (ODE models from CA 3 & 4 with optimal parameter values estimated in CA 4)
- JoCSE paper = Ref. 34

Ruggiero, Zhao, Ford Versypt, JOCSE, 2018

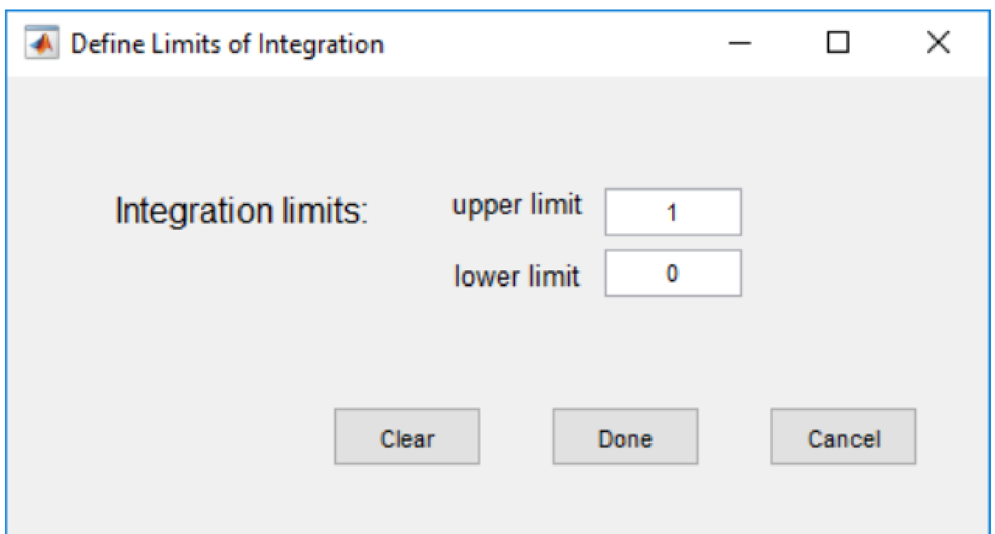
- Recreate Polymath ODE solver functionality in MATLAB
- Individual with open discussion encouraged
- 30% of grade & at least 4 weeks to complete


CA6
Define/Edit Equations/Parameters
Define Equations
Edit Selected ODE
Integration Limits
Edit Explicit Eqs
Calculate/Plot
Calculate
Plot Results
Save File
Save Plot
Save Data to .csv
Reset
Help
(*Click Reset to clear all data input)
ODE(s) Entered
dT/dV =(Ua*(Ta - T) + r1A*
dF_A/dV =rA
dF_B/dV =rB
dF_C/dV =rC
Initial Value(s)
T_0=423
F_A_0=10
F_B_0=0
F_C_0=0
Integration Limits
(0,1)
Explicit Equation(s)
T0 = 423
deltaH_Rx1A = -20000
deltaH_Rx2A = -60000
CpA = 90
CpB = 90
CpC = 180
C_T0 = 0.1
Plot
Dependent Variables - T
Independent Variable - V
T
850
800
750
700
650
600
550
500
450
400
0
0.2
0.4
0.6
0.8
1


Equations Setup
Enter number of ODEs:
4
Add explicit equations
Clear
OK
Cancel



Differential Equations Solver: Enter Differential Equation
Enter the differential equation:
d (T)
d (V)
=
(Ua*(Ta - T) + r1A*deltaH_Rx1A + r2A*deltaH_Rx2A)/(F_
Set the initial value:
y(0) =
423
Clear
Done
Cancel



Define Limits of Integration
Integration limits:
upper limit
1
lower limit
0
Clear
Done
Cancel

Kinetics ex. System of nonlinear ODEs

Note from Dr. Ford Versypt

Be cautious when following these example equations:

1. The program used here allows equations to be entered in any order. MATLAB and Python both require variables to be defined before they can be used.
2. The differential equations list and the explicit equations for k1a and k2a have parameter values plugged in. Instead you should leave the parameter names in the formulas and separately provide parameter values so that the values can be changed easily.

$$F_T = F_A + F_B + F_C \tag{E12-5.13}$$

$$k_{1A} = 10 \exp\left[4000\left(\frac{1}{300} - \frac{1}{T}\right)\right] \mathrm{s}^{-1}$$

$(T \text{ in K})$

$$k_{2A} = 0.09 \exp\left[9000\left(\frac{1}{300} - \frac{1}{T}\right)\right] \frac{\mathrm{dm}^3}{\mathrm{mol \cdot s}}$$

Energy balance:

$$\frac{dT}{dV} = \frac{4000(373 - T) + (-r_{1A})(20{,}000) + (-r_{2A})(60{,}000)}{90F_A + 90F_B + 180F_C} \tag{E12-5.14}$$

The Polymath program and its graphical outputs are shown in Table E12-5.1 and Figures E12-5.1 and E12-5.2.

TABLE E12-5.1 POLYMATH PROGRAM

Differential equations

```
1 d(Fa)/d(V) = r1a+r2a
2 d(Fb)/d(V) = -r1a
3 d(Fc)/d(V) = -r2a/2
4 d(T)/d(V) = (4000*(373-T)+(-r1a)*20000+(-r2a)*60000)/(90*Fa+90*Fb+180*Fc)
```

Explicit equations

```
1  k1a = 10*exp(4000*(1/300-1/T))
2  k2a = 0.09*exp(9000*(1/300-1/T))
3  Cto = 0.1
4  Ft = Fa+Fb+Fc
5  To = 423
6  Ca = Cto*(Fa/Ft)*(To/T)
7  Cb = Cto*(Fb/Ft)*(To/T)
8  Cc = Cto*(Fc/Ft)*(To/T)
9  r1a = -k1a*Ca
10 r2a = -k2a*Ca^2
```

Calculated values of DEQ variables

	Variable	Initial value	Final value
1	Ca	0.1	2.069E-09
2	Cb	0	0.0415941
3	Cc	0	0.016986
4	Cto	0.1	0.1
5	Fa	100.	2.738E-06
6	Fb	0	55.04326
7	Fc	0	22.47837
8	Ft	100.	77.52163
9	k1a	482.8247	2.426E+04
10	k2a	553.0557	3.716E+06
11	r1a	-48.28247	-5.019E-05
12	r2a	-5.530557	-1.591E-11
13	T	423.	722.0882
14	To	423.	423.
15	V	0	1.

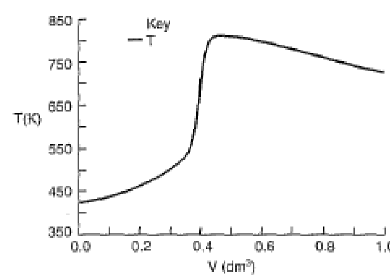


Figure E12-5.1 Temperature profile.

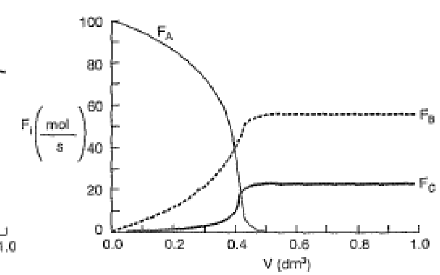
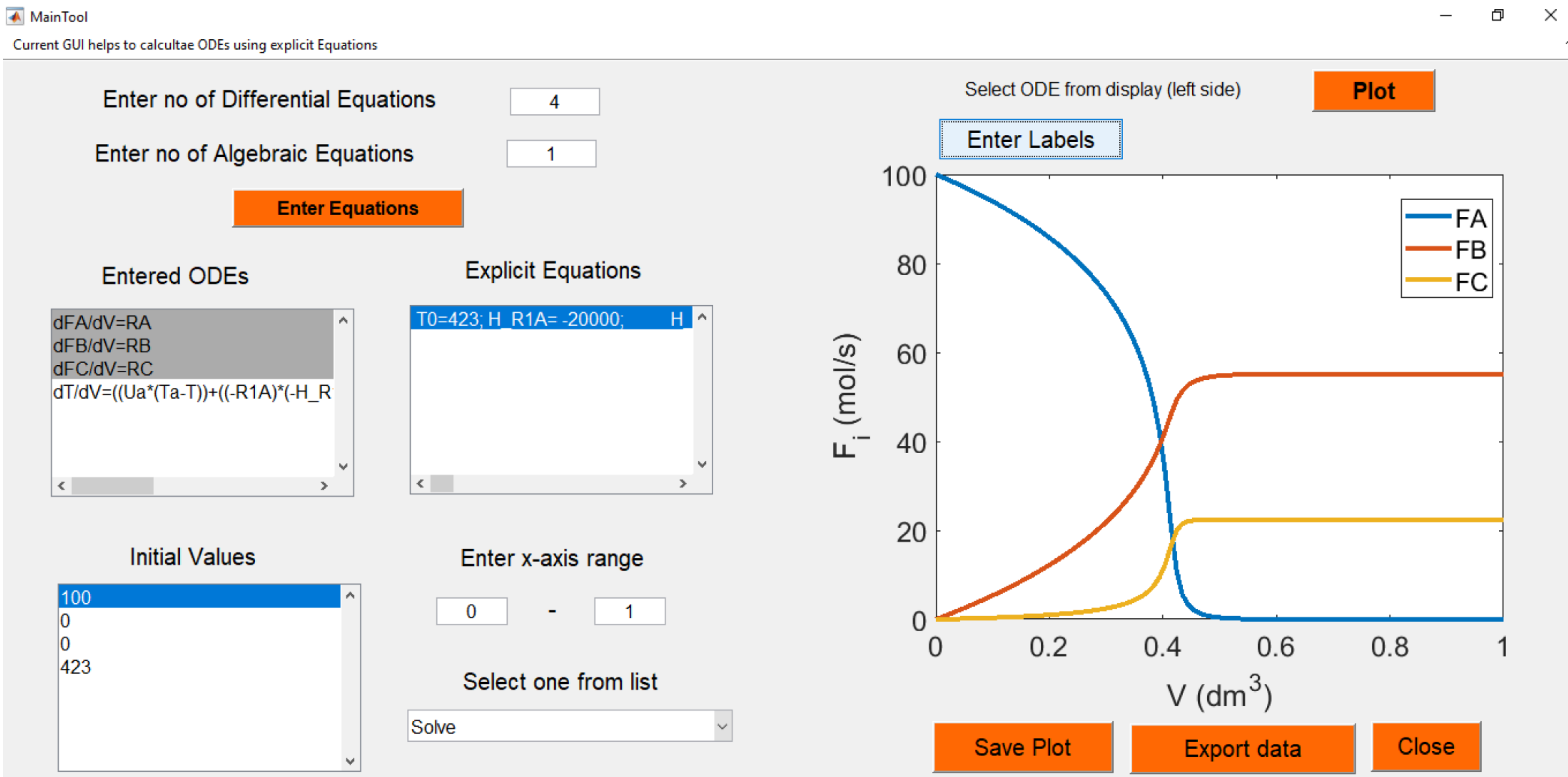
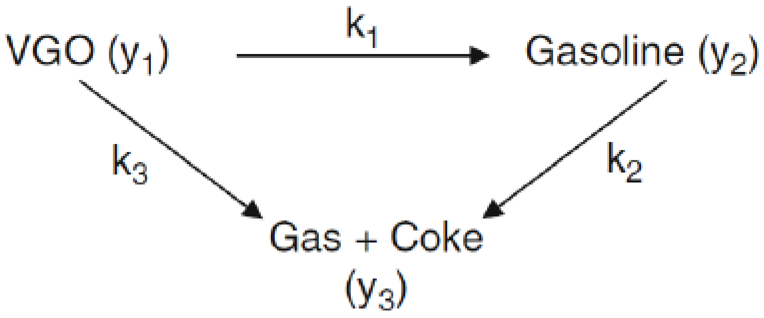


Figure E12-5.2 Profile of molar flow rates F_A, F_B, and F_C.



MainTool
Current GUI helps to calcultae ODEs using explicit Equations
Enter no of Differential Equations
4
Enter no of Algebraic Equations
1
Enter Equations
Entered ODEs
dFA/dV=RA
dFB/dV=RB
dFC/dV=RC
dT/dV=((Ua*(Ta-T))+((-R1A)*(-H_R
Explicit Equations
T0=423; H_R1A= -20000; H
Initial Values
100
0
0
423
Enter x-axis range
0
-
1
Select one from list
Solve
Select ODE from display (left side)
Plot
Enter Labels
FA
FB
FC
F_i (mol/s)
V (dm^3)
Save Plot
Export data
Close

FCC ex.
Parameter estimation of complex models



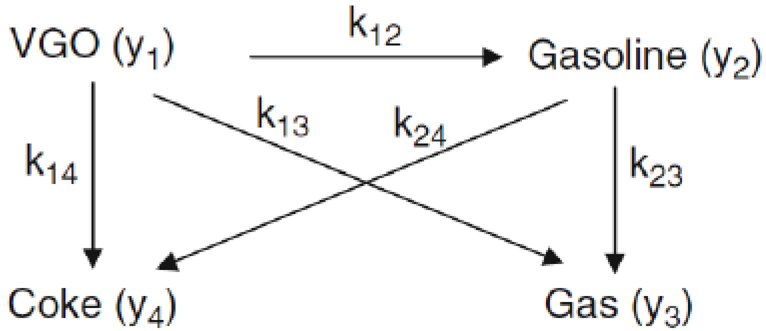
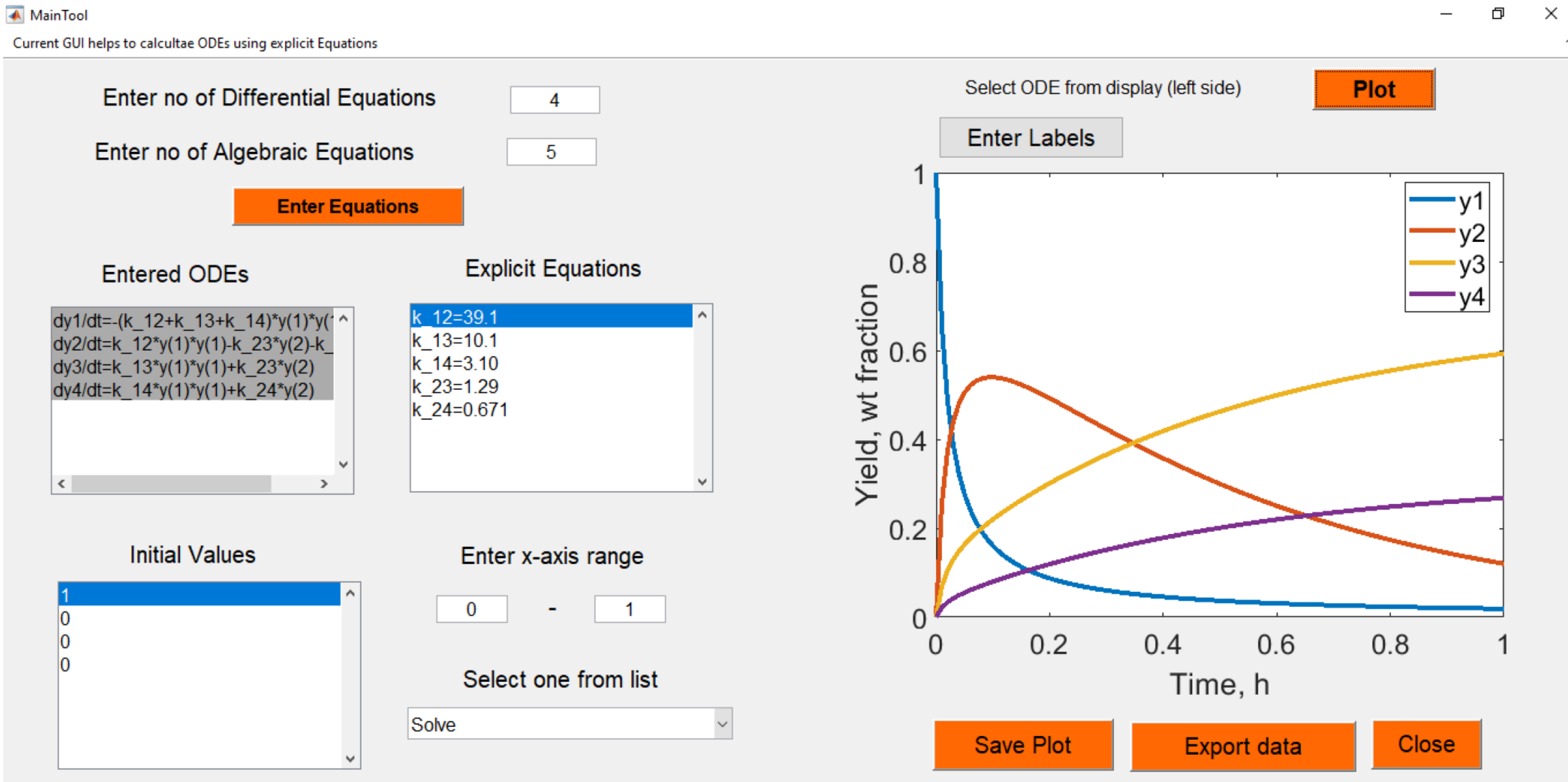


Table 1: Data for FCC process.

Time (h)	Conversion	Weight Fraction VGO	Gasoline	Gas	Coke
1/60	0.4926	0.5074	0.3767	0.0885	0.0274
1/30	0.6204	0.3796	0.4385	0.136	0.0459
1/20	0.7118	0.2882	0.4865	0.1681	0.0572
1/10	0.8238	0.1762	0.5416	0.2108	0.0714


MainTool
Current GUI helps to calcultae ODEs using explicit Equations
Enter no of Differential Equations
4
Enter no of Algebraic Equations
5
Enter Equations
Entered ODEs
dy1/dt=-(k_12+k_13+k_14)*y(1)*y(
dy2/dt=k_12*y(1)*y(1)-k_23*y(2)-k_
dy3/dt=k_13*y(1)*y(1)+k_23*y(2)
dy4/dt=k_14*y(1)*y(1)+k_24*y(2)
Explicit Equations
k_12=39.1
k_13=10.1
k_14=3.10
k_23=1.29
k_24=0.671
Initial Values
1
0
0
0
Enter x-axis range
0
-
1
Select one from list
Solve
Select ODE from display (left side)
Plot
Enter Labels
Yield, wt fraction
0
0.2
0.4
0.6
0.8
1
Time, h
0
0.2
0.4
0.6
0.8
1
y1
y2
y3
y4
Save Plot
Export data
Close

Building Computational Skills for Mathematical Modeling in Science and Engineering through an Interdisciplinary Elective Course

Dr. Ashlee N. Ford Versypt, Ph.D.

Assistant Professor

Oklahoma State University

College of Engineering, Architecture & Technology

School of Chemical Engineering

tinyurl.com/ashleefv

@FordVersyptLab

ashleefv@okstate.edu

Steal this course (or share GUI ideas)