CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

What is chemical engineering

- Chemical engineering simply deals with producing chemicals at large-scale
- A chemist found that A and B react to give C. To do so successfully, he first mixed A and B in a test tube by shaking and then used a burner to heat the mixture. In doing so, the chemical conversion happened. C is a precipitate that needs to be separated out. He used filter paper to separate the liquid and solids. The amount of C, he made is in grams.
- If C is to be produced in Tons, what should we do?

What is chemical engineering

Chemical Engineering

Biotechnology
Medical devices/diagnostics
genetic engineering
Pharmaceuticals

Food
Agriculture
productivity
smart packaging
preservation

Environment
Catalytic converters
recycled materials
waste reduction

Materials
Polymers/plastics
ceramics
coatings

Energy
Petrochemicals
solar panels
fuel cells

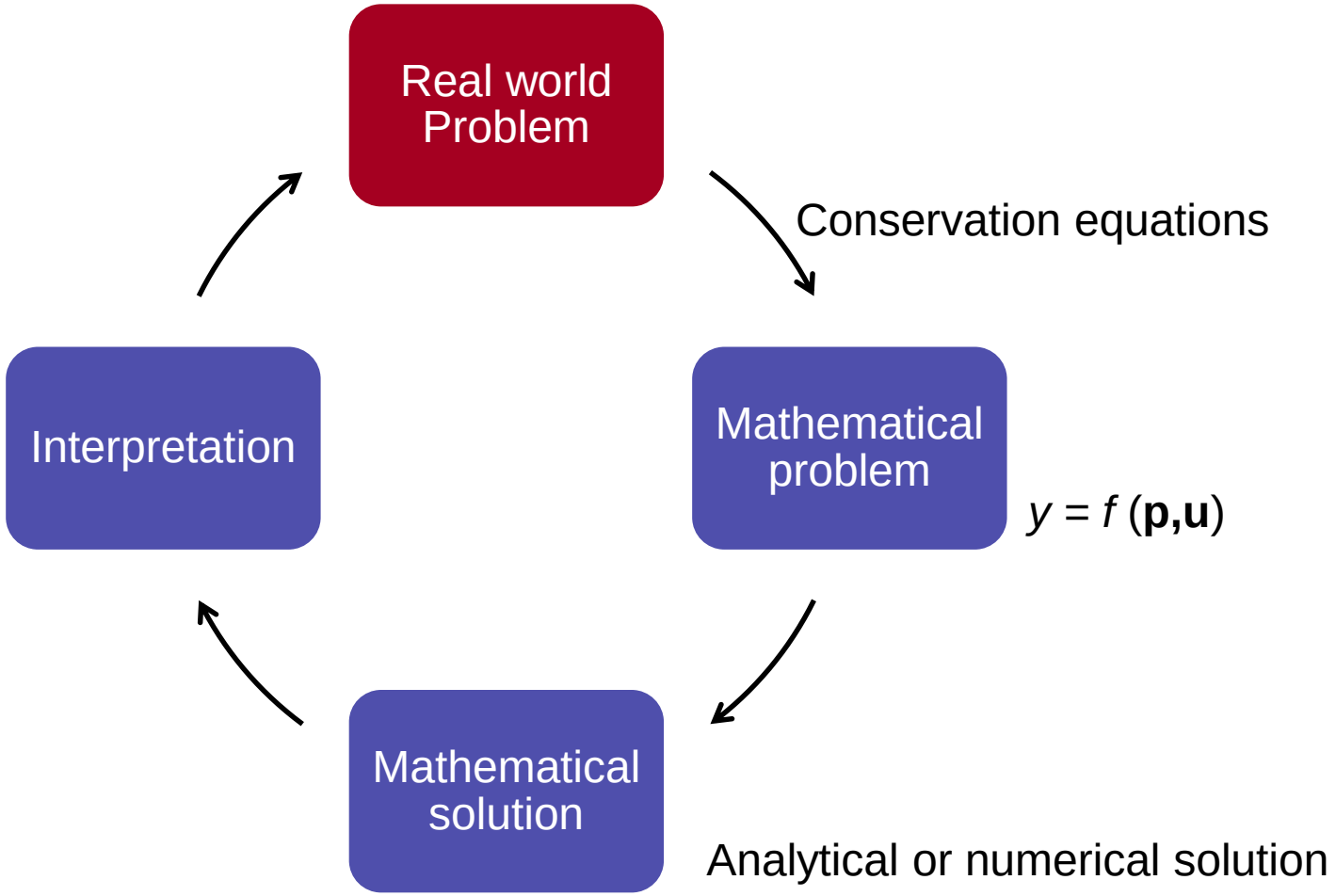
Information
Pentium – chip
manufacture
semiconductors
optical fibers

Computer applications

- Why computers are required in chemical engineering?
- Engineers convert real-life problems into mathematical problems
- Mathematical problems means set of equations
  - Mathematical modeling: writing equations that describe system of interest for a specific purpose
- Not every equation can be solved with pen and paper (analytical solution)

Computer applications

- Computers can help us to solve the equations
- Numerical solution of equations: Simulation



Example 1

- *Can I calculate the boiling time for a soft-boiled egg, given its weight and initial temperature?*

- http://newton.ex.ac.uk/teaching/CDHW/egg/

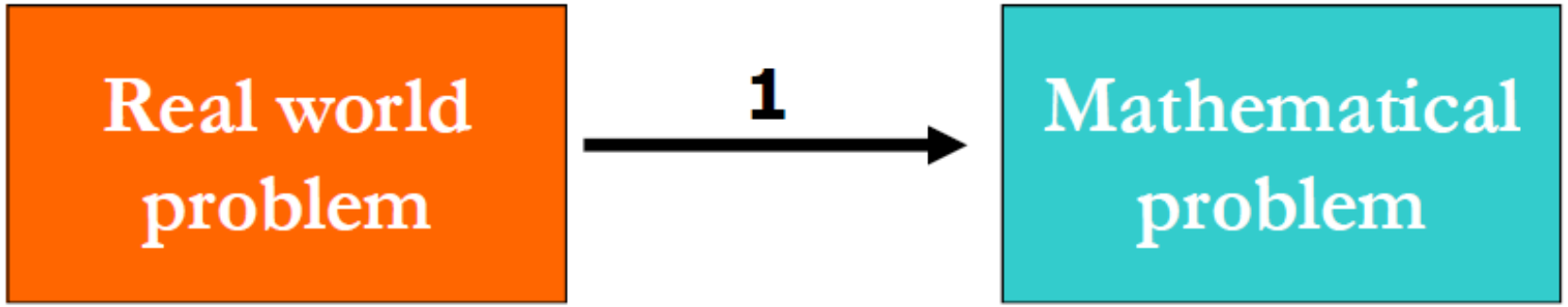
- http://newton.ex.ac.uk/teaching/CDHW/egg/CW061201-1.pdf

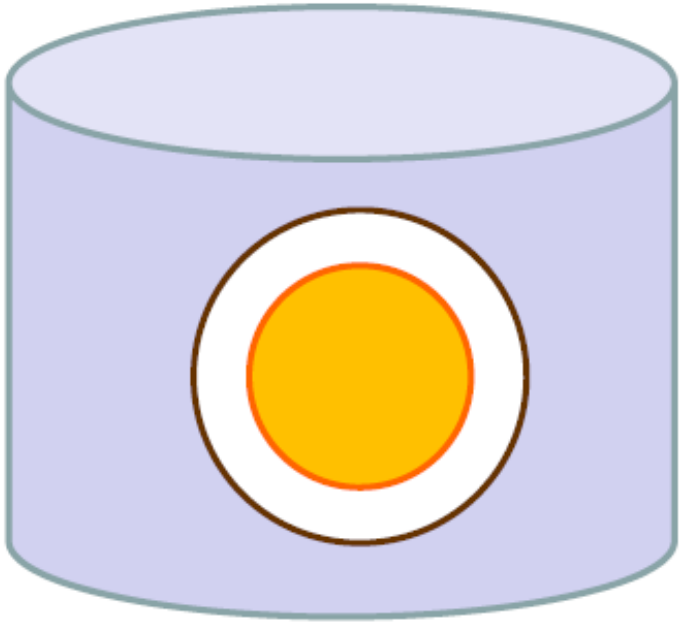
Boiling time of an egg

Real world problem

- Assumptions
  - Spherical egg
  - Egg is considered to be cooked when the T at boundary of the yolk reaches 63 °C

Boiling time of an egg

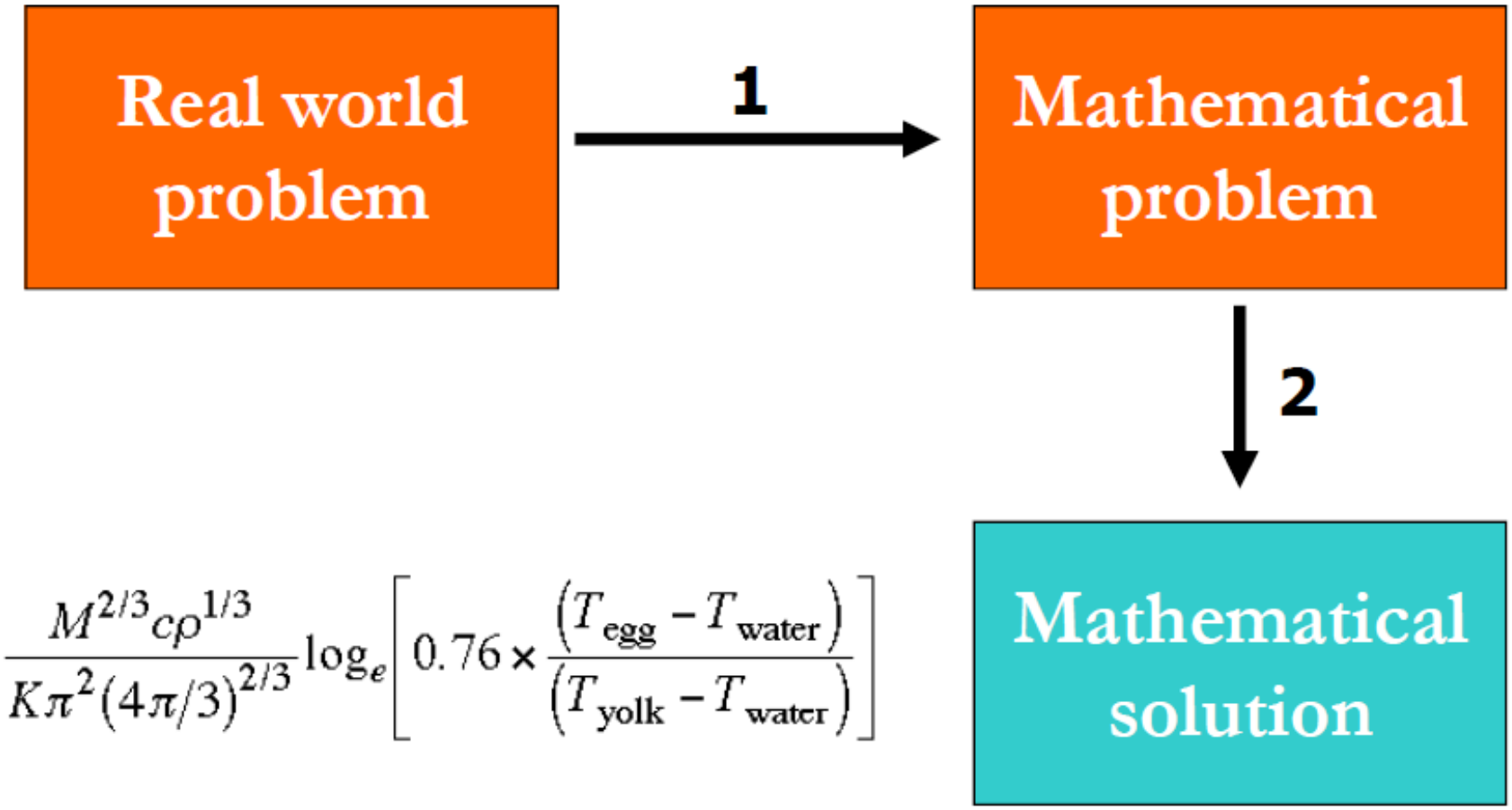




$$\frac{\partial}{\partial r}\left(r^2\frac{\partial T}{\partial r}\right)=\frac{\tau_0 r^2}{a^2}\left(\frac{\partial T}{\partial t}\right)$$

http://newton.ex.ac.uk/teaching/CDHW/egg/CW061201-1.pdf

Boiling time of an egg



$$t_{\text{cooked}} = \frac{M^{2/3} c \rho^{1/3}}{K \pi^2 (4\pi/3)^{2/3}} \log_e \left[0.76 \times \frac{(T_{\text{egg}} - T_{\text{water}})}{(T_{\text{yolk}} - T_{\text{water}})} \right]$$

Boiling time of an egg

$$\frac{\partial}{\partial r}\left(r^2\frac{\partial T}{\partial r}\right)=\frac{\tau_0 r^2}{a^2}\left(\frac{\partial T}{\partial t}\right)$$

Real world problem

1

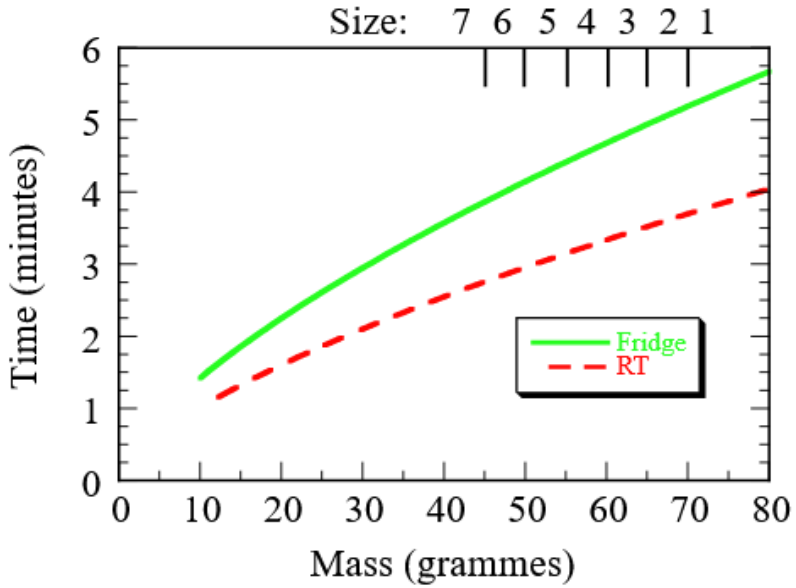
Mathematical problem

2

Mathematical solution

3

4

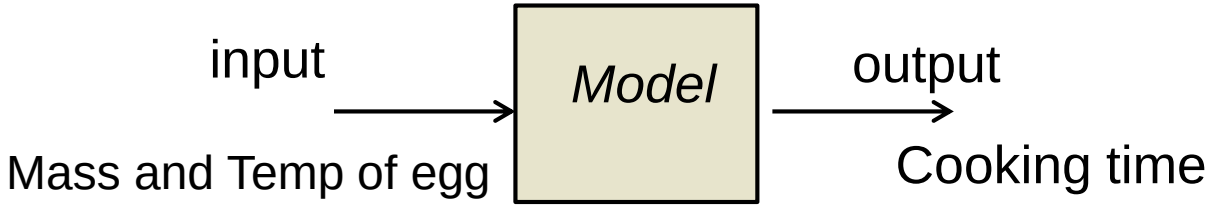


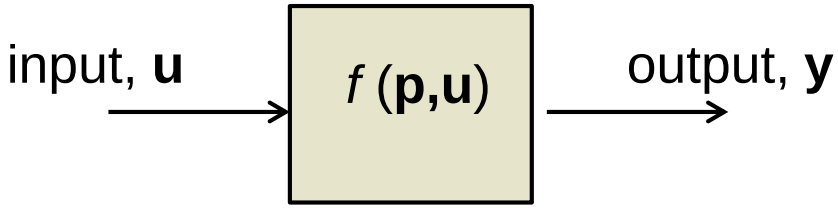
$$t_{\text{cooked}} = \frac{M^{2/3} c \rho^{1/3}}{K\pi^2 (4\pi/3)^{2/3}} \log_e \left[0.76 \times \frac{(T_{\text{egg}} - T_{\text{water}})}{(T_{\text{yolk}} - T_{\text{water}})}\right]$$

Theoretical modeling

- We just now completed a theoretical modeling problem
  - Physics-based modeling/Mechanistic modeling
  - Used theory/concept of conservation of energy to write the equation for temperature
  - Partial differential equation
  - Computers are required to simulate equations

$$\frac{\partial}{\partial r}\left(r^2 \frac{\partial T}{\partial r}\right) = \frac{\tau_0 r^2}{a^2}\left(\frac{\partial T}{\partial t}\right)$$





Empirical modeling

- Observation-based modeling
  - You can conduct experiments for different eggs and temperatures, and measure the boiling times

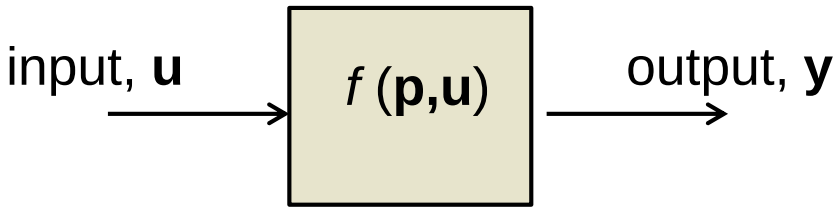
Can I calculate the boiling time for a soft-boiled egg, given its weight and initial temperature?

  - You will get some data
  - Data can be fitted to equations
  - Again, computers are needed

Observations

- Conduct experiments
- Collect data
- Data-fitting

Mass of egg (gm)	Temperature of egg (°C)	Time to cook (measured) (mins)
80	4	6
80	21	4
60	4	4.75
60	21	3.25
40	4	3.5
40	21	2.5
20	4	2.25
20	21	1.6
10	4	1.4
10	21	1



Cooking time = *f* (mass of egg, temperature of egg)

Two modeling approaches: summary

Theoretical Modeling	Process Identification
First principles, white box (chemistry, physics, etc.)	Data driven modeling, black box, empirical
Requires sufficient knowledge about the process	Requires only the process output data in response to changes in input
Provides information about the internal state of the process	Provides information only about process input-output
Typically requires fewer measurements to estimate unknown model parameters	Requires extensive measurements as accuracy relies entirely on data

Usage of mathematical modeling

- Consider a chemical reaction between A and B.
  - The conversion might depend on several parameters, e.g., c_A, c_B, T, pH, shaking rpm.
  - Let us say, we have 5 values for each of the parameters.
  - You want to optimize the process conditions
  - 5^5=3125 number of expts required

Conversion = f (c_A, c_B, T, pH, shaking rpm)

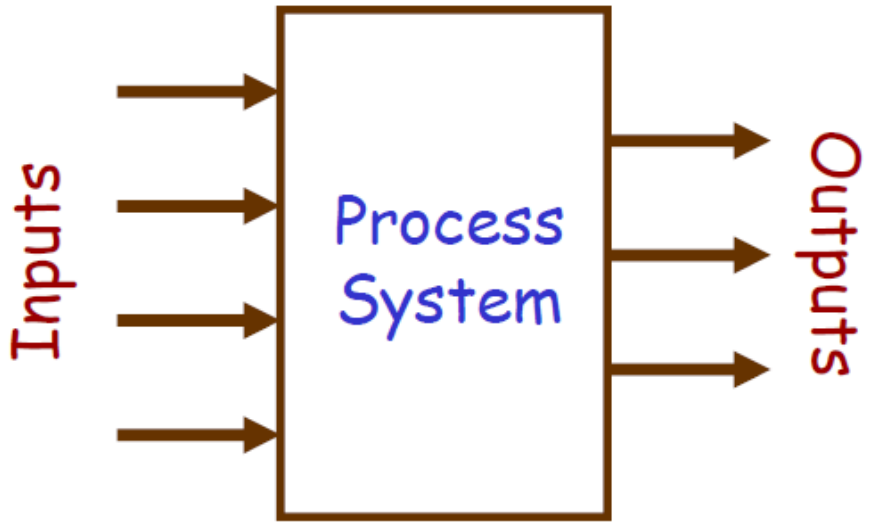
Usage of mathematical modeling

Prof. George Box once said

"All models are wrong. Some models are useful"

- In what ways, can a model be useful?
  - Process Design (known, ?, known)
  - Process Simulation (known, known, ?)
  - Control Design (?, known, known)
  - Process Optimization

  .....



Why this course?

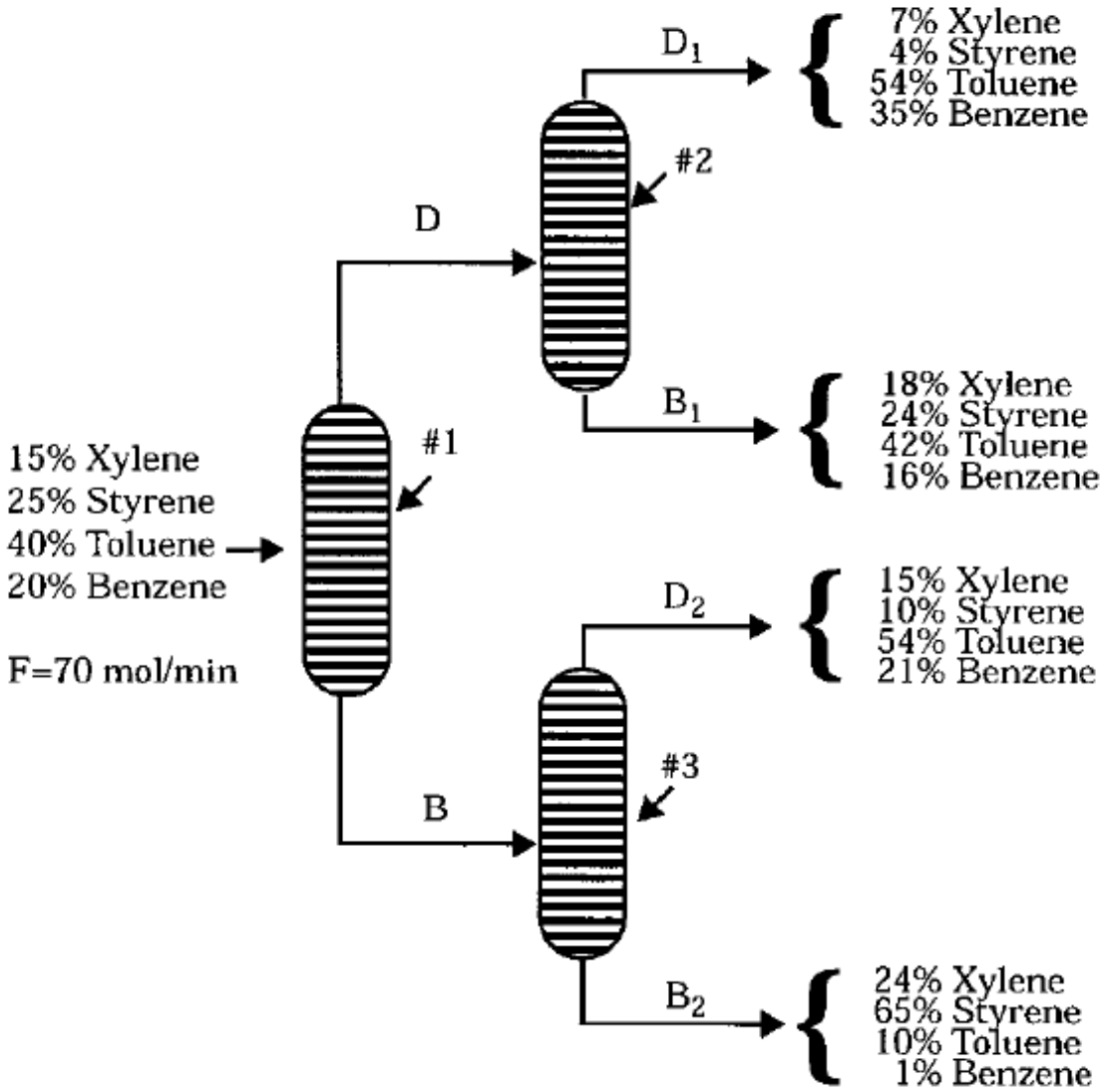
- Van der Waals equation of state

$$\left(P + \frac{a}{V^2}\right)(V - b) = RT \qquad a = \frac{27}{64}\left(\frac{R^2 T_c^2}{P_c}\right) \qquad b = \frac{RT_c}{8P_c}$$

- P (atm), V(L/gmol), R=0.08206 atm.L/(gmol. K)
- T_c=405.5 K, P_c=111.3 atm for ammonia

- Calculate the molar volume and compressibility factor for ammonia gas at a pressure of 56 atm and a temperature of 450 K.

Why this course?



Xylene: $0.07D_1 + 0.18B_1 + 0.15D_2 + 0.24B_2 = 0.15 \times 70$

Styrene: $0.04D_1 + 0.24B_1 + 0.10D_2 + 0.65B_2 = 0.25 \times 70$

Toluene: $0.54D_1 + 0.42B_1 + 0.54D_2 + 0.10B_2 = 0.40 \times 70$

Benzene: $0.35D_1 + 0.16B_1 + 0.21D_2 + 0.01B_2 = 0.20 \times 70$

Why this course?

Table A.1 Vapor Pressure of Benzene (Perry[3])

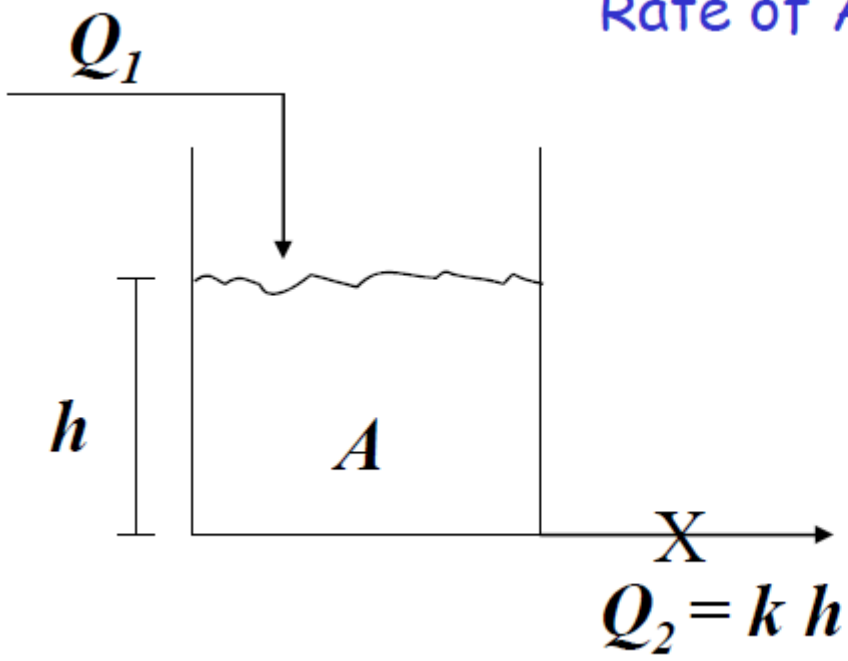
Temperature, T (°C)	Pressure, P (mmHg)
−36.7	1
−19.6	5
−11.5	10
−2.6	20
+7.6	40
15.4	60
26.1	100
42.2	200
60.6	400
80.1	760

$$P = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + \cdots + a_n T^n$$

$$\log(P) = A - \frac{B}{T + 273.15}$$

$$\log(P) = A - \frac{B}{T + C}$$

Why this course?



Rate of Accumulation = Flow Rate in - Flow Rate out

$$A\frac{dh}{dt} = Q_1 - Q_2 = Q_1 - k\,h$$

$$\Rightarrow \frac{A}{k}\frac{dh}{dt} = \frac{Q_1}{k} - h$$

$$\Rightarrow \tau\frac{dh}{dt} + h = R_1$$

Books

- Finlayson B.A., "Introduction to Chemical Engineering Computing", Second Edition, Copyright © 2012 John Wiley & Sons, Inc. https://onlinelibrary.wiley.com/doi/book/10.1002/9781118309599

- Gerald, C.F., and Wheatley, P.O., "Applied Numerical Analysis", 7th Edition, Pearson publication, 2004. http://www.cse.iitm.ac.in/~vplab/downloads/opt/Applied%20Numerical%20Analysis.pdf

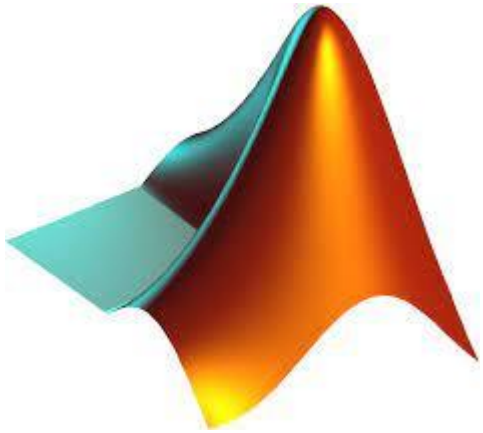
Assessment and grading

- Attendance 10%
- Assignments/projects/quizzes: 20%
- MTE: 25%
- ETE: 45%

Final Grade: f (your effort, involvement)

MATLAB

- https://in.mathworks.com/products/matlab.html



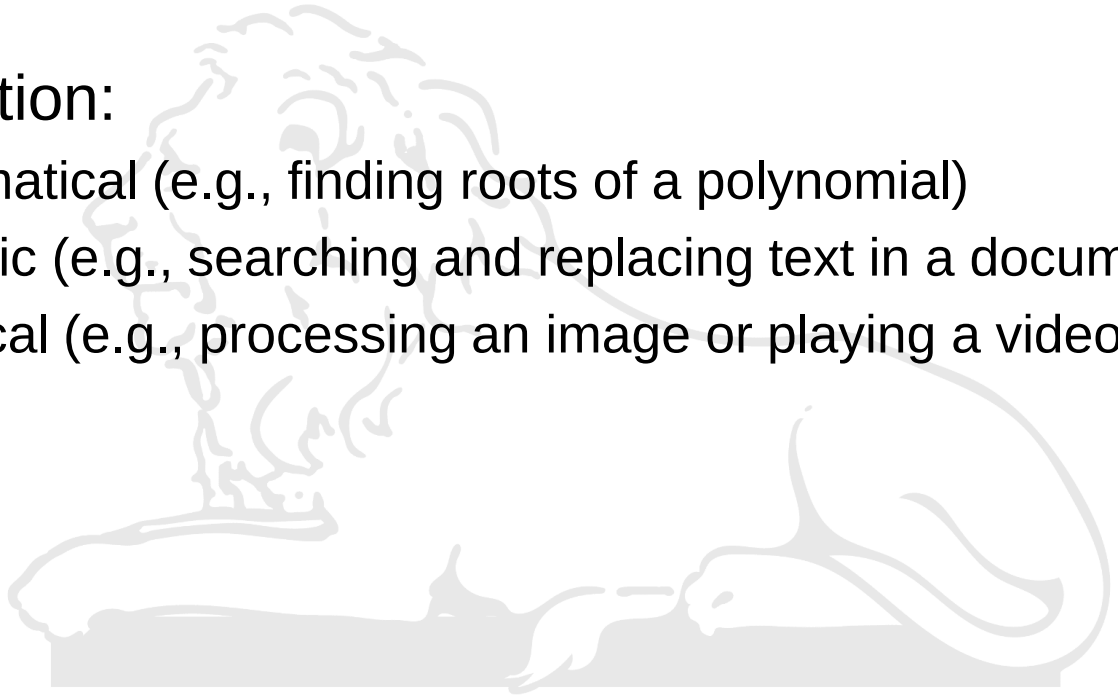
Introduction to Problem Solving with MATLAB

Department of Chemical Engineering
Indian Institute of Technology Roorkee

What is a program?

- A sequence of instructions that specifies how to perform a computation

- Computation:
  - Mathematical (e.g., finding roots of a polynomial)
  - Symbolic (e.g., searching and replacing text in a document)
  - Graphical (e.g., processing an image or playing a video)



What is a program?

- Basic instructions in any programming language
  - *Input*: Get data from keyboard, a file, the network or some device
  - *Output*: Display data on screen, save it in a file, send it over the network, etc.
  - *Math*: Perform basic mathematical operations like addition
  - *Conditional execution*: Check for certain conditions and run the appropriate code
  - *Repetition*: Perform some action repeatedly, usually with some variation

- Every program (easy/difficult) is made up of above instructions

- Programming: A process of breaking a large, complex task into smaller and smaller subtasks until the subtasks are simple enough to be performed with one of these basic instructions

Programming language?

Formal and natural languages

- Natural languages:
  - Languages that people speak (Hindi, English, etc.)
  - They were not designed by people; they evolved naturally
- Formal languages:
  - Languages designed by people for specific applications
  - E.g., notation used by mathematicians (denoting relationship between numbers and symbols)
  - Notation used by chemists to represent chemical structure of molecules
  - Programming languages designed to express computations

Formal and natural languages

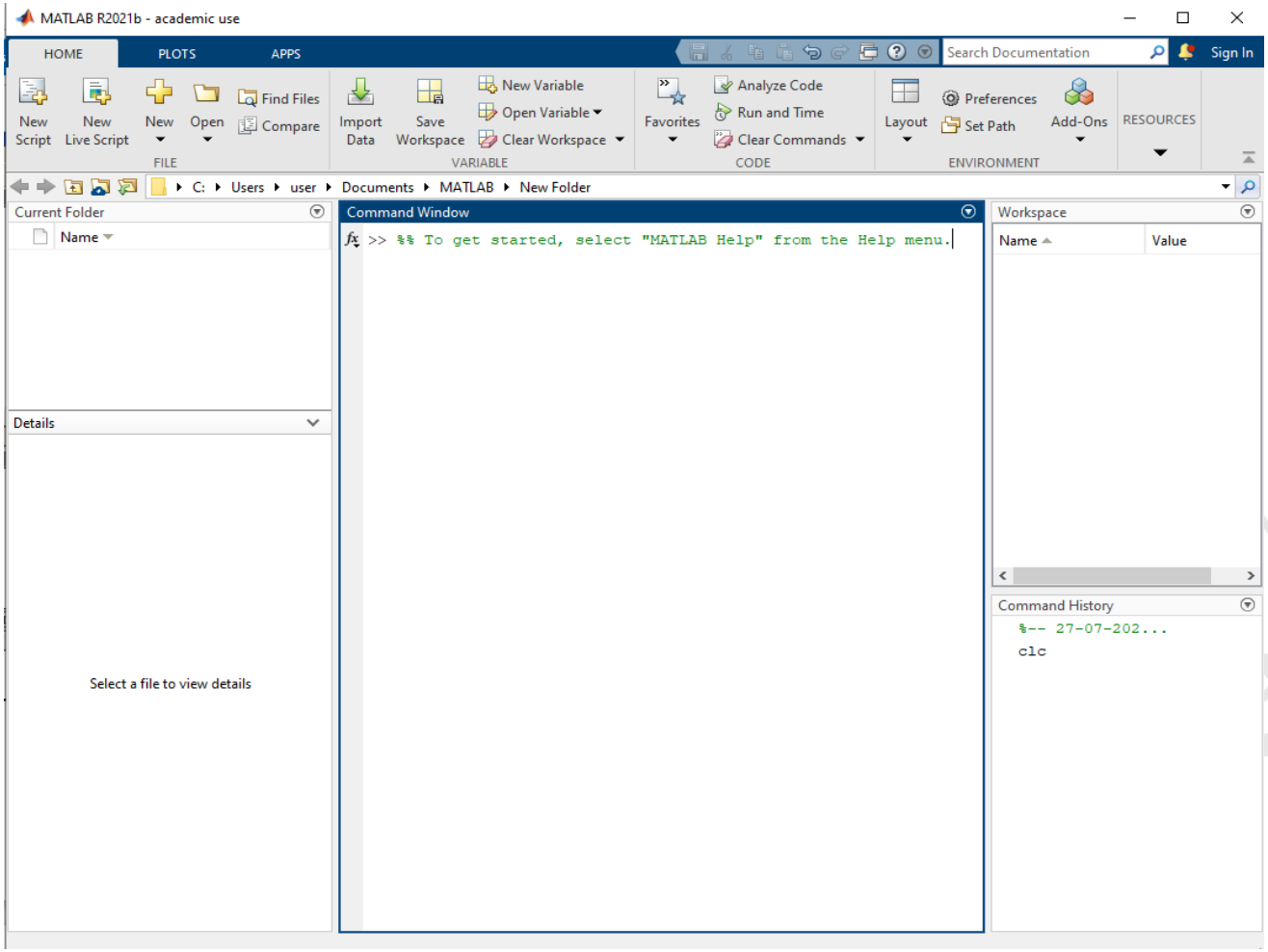
- Formal languages have strict syntax rules
- Formal languages are designed to be completely unambiguous
- Formal languages are less verbose and more concise
- Formal languages are more dense than natural languages, so it takes longer to read them.
  - The structure is important, so, not always best to read from top to bottom, left to right
- Small errors in spelling and punctuation can make a big difference in formal language

Running MATLAB

- Two possible ways
  - Install MATLAB on your computer
  - Run on browser (MATLAB online); no installation required
- Sign up is required in both the modes
- IIT Roorkee subscribed for MATLAB for all of its students and staff
- Use your IITR email and sign up

Running MATLAB

- Can be launched from the icon created on the desktop.



The desktop has the following panels –

1. Current folder
2. Command window
3. Workspace
4. Command history
5. Details

Layout can be customized

The first program

- Print or display “Hello, World!”

- In MATLAB, it looks like this:

```
>> disp('Hello, World!')
```

- *disp* is a command that displays a result on the screen. In this case, the result is the words **(press enter to execute)**:

```
Hello, World!
```

- Quotation marks
- Parentheses (small brackets)

Arithmetic operators

- MATLAB provides operators which are special symbols that represent computations like addition and multiplication
- Examples: +, -, *, /

```
>> 41+2
ans =
    43

>> 45-4
ans =
    41

>> 6*5
ans =
    30
```

```
>> 25/2
ans =
    12.5

>> 6^2
ans =
    36
```

MATLAB evaluates arithmetic expressions precisely by the rules for order of precedence. Use brackets as needed to get the correct expression or make the expression clear!

Try:

```
>> 4^3^2
>> 4^(3^2)
>> (4^3)^2
```

Write a MATLAB command to enter a fraction with 17.1 + 20.3 in the numerator and 36.5 + 41.8 in the denominator.

Values and types

- A value is one of the basic things a program work with like a letter or a number. Some values we have seen so far are `43, 12.5, 'Hello, World!'`

- The value belongs to different types: integers, floating point number, strings
  - MATLAB has different classes defined for different types (int8, int16, int32, int64, uint8, uint16, uint32, uint64, single, double)

- By default, MATLAB stores all numeric values as **double-precision floating point.**

Values and types

- Double and single
  - Space taken on computer: single: 32 bits, double: 64 bits
  - Accuracy

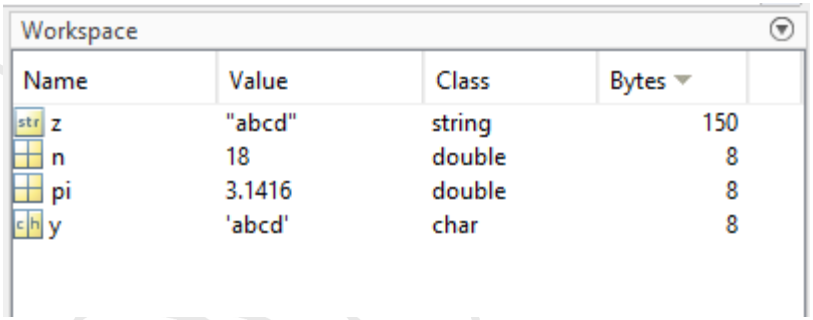
```
>> y=single(12345678)
y =
  single
    12345678

>> x=single(12345679)
x =
  single
    12345679

>> x-y
ans =
  single
     1
```

```
>> a=single(123456788)
a =
  single
    123456784

>> b=single(123456789)
b =
  single
    123456792

>> b-a
ans =
  single
     8
```

Variables

- All programming language has the ability to manipulate variables
- A variable is a name that refers to a value
- An assignment statement creates a new variables and gives it a value:

```
>> n=18
n =
     18

>> pi=3.141592653589793
pi =
   3.141592653589793

>> y='abcd'
y =
    'abcd'
```



Variable names

- Choose meaningful names
- Maximum possible length of a variable name can found by command `namelengthmax` (It is 63).
- Variable name can contain both letters and numbers, but they can't begin with a number
- Underscore is allowed in a variable name
- Same spelling, Uppercase and lowercase are considered as different variables

Expressions and statements

- An expression is a combination of values, variables, and operators

  x+5

  7*y^2/10

- A statement is a unit of code that has an effect, like creating/updating a variable or displaying a value

  x=9

  y=x+5

  z=7*y^2/10

Use of semicolon in MATLAB

- No punctuation mark is needed at the end of statement.

- However, if you want to suppress and hide the MATLAB output for an expression, add a semicolon after the expression.

- For example,

```
>> x = 3;
>> y = x + 5
y =
     8
```

Script mode

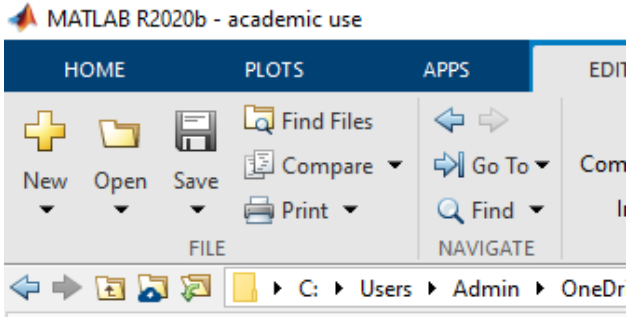
- So far, we have used MATLAB environment as a calculator. We have learned how to enter commands from the MATLAB command prompt.

- If we have more than few lines of commands to execute a particular task, using command prompt can be clumsy

- MATLAB is a powerful programming language. It allows you to write series of commands into a file and execute the file as a complete unit.

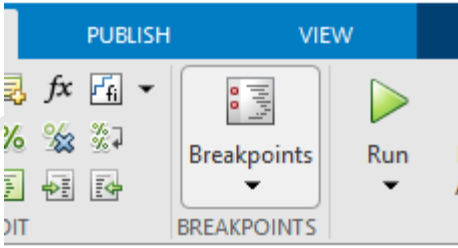
Creating, saving and running scripts

- In the MATLAB GUI, look at the toolstrip at the top
- At the LHS in toolstrip, look for a yellow +



- Clicking this will open MATLAB Editor window with a script named 'untitled'
- MATLAB scripts has .m extension

- Type all the commands in the script and save it



- To execute the file, press the run button in the toolstrip

Exercise: Script mode

- Create a script file with the following code

```
a = 5; b = 7;
c = a + b
d = c + sin(b)
e = 5 * d
f = exp(-d)
```

- When the above code is compiled and executed, it produces the following result (in the command prompt)

```
c =  12
d =  12.657
e =  63.285
f =  3.1852e-06
```

- What do you see in workspace?

Script mode: comments

- As program get bigger and complicated, it becomes more difficult to read.
- Formal/programming languages are dense. It is good idea to add notes to your programs to explain in natural language what the program is doing.
- These notes are called comments and they start with % symbol
- Shortcuts: ctrl+R and ctrl+T

- Don’t write unnecessary comments when the code is obvious

```
v=5  % assign 5 to v
```

- Comments should provide useful information that is not in the code

```
v=5  % velocity in meters/second
```

Script mode: important commands

- Two important commands:
  - `clc`
  - `clearall`
- Suggested to include both these commands in every script at the beginning
- `clc` clears the command prompt
- `clearall` clears all the variables in the workspace or in matlab memory

Exercise

- Practice in the command prompt

1. How many seconds are there in 44 minute 44 seconds?

2. If you run a 10 km race in 44 min 44 seconds, what is your average speed in miles per hour?

3. The volume of a sphere with radius r is $(4/3)\pi r^3$. What is the volume of a sphere with radius 3?

4. MRP of a book is 240 rupees, however, the bookstores get a 40% discount. Shipping cost 5 rupees for the first copy and 1 rupee for each additional copy. What is the cost price for 60 copies for a bookstore?

MATrix and MATLAB

- *MATLAB* is an abbreviation for "matrix laboratory."

- While other programming languages mostly work with numbers one at a time, MATLAB® is designed to operate primarily on whole matrices and arrays.

- All MATLAB variables are multidimensional *arrays*, no matter what type of data.

  – Is A = 9 an array in MATLAB?
  – Where can we see that? (Hint: workspace)

Vector and matrix (Mathematics)

- Vector:
  - A vector is a one-dimensional array of numbers.
  - Row vectors and column vectors

- Matrix
  - A matrix is a two-dimensional array of numbers.

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \qquad [1 \;\; 2 \;\; 3 \;\; 4] \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Vectors in MATLAB

- MATLAB allows creating both types of vectors: row and column

- Row vectors are created by enclosing the set of elements in square brackets, using space or comma to delimit the elements.

```
>> r = [7 8 9 10 11]
r =
     7     8     9    10    11
```

- Column vectors are created by enclosing the set of elements in square brackets, using semicolon to delimit the elements.

Vectors in MATLAB

```
>> r = [7; 8; 9; 10; 11]
r =
     7
     8
     9
    10
    11
```

- Referencing the elements of a vector
  - The i^{th} component of a vector `A` is referred as `A(i)`. For example

```
>> A = [ 1; 2; 3; 4; 5; 6];
>> A(3)
ans =
     3
```

Vectors in MATLAB

- MATLAB allows you to select a range of elements from a vector.

```
>> rv = [1 2 3 4 5 6 7 8 9];
>> sub_rv = rv(3:7)
sub_rv =
     3     4     5     6     7
```

- What will happen if I execute?

```
>> X=rv(:)
```

Matrix in MATLAB

- We can create a matrix by entering elements in each row as comma or space delimited numbers and using semicolons to mark the end of each row.

- Let us create a 3-by-3 matrix *A*

```
>> A = [ 1 2 3; 4 5 6; 7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
```

Matrix in MATLAB

- Referencing the elements of a matrix
  - To reference an element in the m^{th} row and n^{th} column, of a matrix A, we write

```
>> A(2,3)
ans =
    6

>> A(2,4)
Index in position 2 exceeds array bounds (must not
exceed 3).
```

  - To reference all the elements in the n^{th} column we type `A(:,n)`
  - To reference all the elements in the m^{th} row we type `A(m,:)`

Matrix in MATLAB: Exercise

- Referencing the elements of a matrix

```
>> A = [ 1 2 3; 4 5 6; 7 8 9]
```

  - How to create a smaller matrix taking the elements from the second and third columns?
  - How to create a smaller matrix taking the elements from the second and third rows?
  - How to create a smaller matrix taking the elements from the second and third rows, and second and third columns?

Matrix in MATLAB: Exercise

- Deleting a Row or a Column in a Matrix

```
>> A = [ 1 2 3; 4 5 6; 7 8 9]
```

  - [ ] denotes an empty array
  - We can delete an entire row or column of a matrix by assigning an empty set of square braces [ ] to that row or column.
  - Let us delete third row of matrix A
  - Let us delete third column of matrix A

Matrix in MATLAB: Exercise

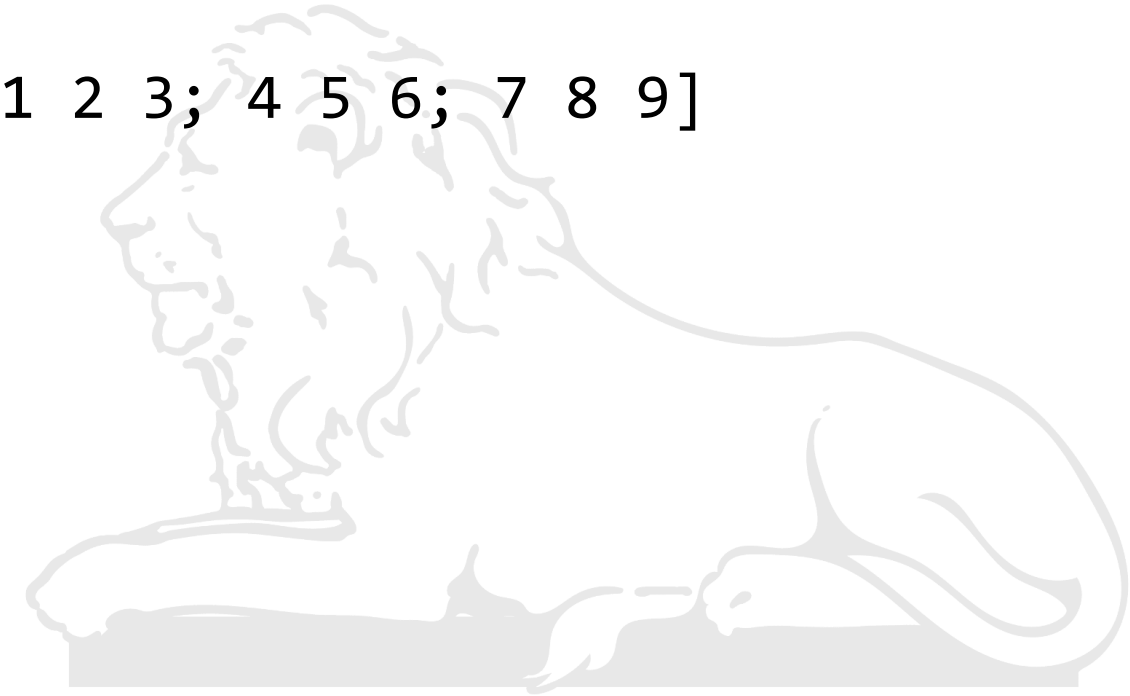
- Write a small MATLAB program to
  - Make a column vector by taking all the elements of a given matrix row-wise
  - Make a column vector by taking all the elements of a given matrix column-wise

```
>> A = [ 1 2 3; 4 5 6; 7 8 9]
```

Matrix in MATLAB: Exercise

- Write a small MATLAB program to
  - Replace the elements of a matrix which are less than or equal to 5

```
>> A = [ 1 2 3; 4 5 6; 7 8 9]
```



Matrix operations

- Addition and Subtraction of Matrices
- Division of Matrices
- Scalar Operations of Matrices
- Transpose of a Matrix
- Concatenating Matrices
- Matrix Multiplication
- Determinant of a Matrix
- Inverse of a Matrix

Addition and Subtraction of Matrices

```
>> a = [ 1 2 3 ; 4 5 6; 7 8 9];
>> b = [ 7 5 6 ; 2 0 8; 5 7 1];
>> c = a + b
>> d = a - b
```

Division of Matrices

You can divide two matrices using left (\) or right (/) division operators.

```
>> a = [ 1 2 3 ; 4 5 6; 7 8 9];
>> b = [ 7 5 6 ; 2 0 8; 5 7 1];
>> c = a / b
>> d = a \ b
```

Scalar Operations of Matrices

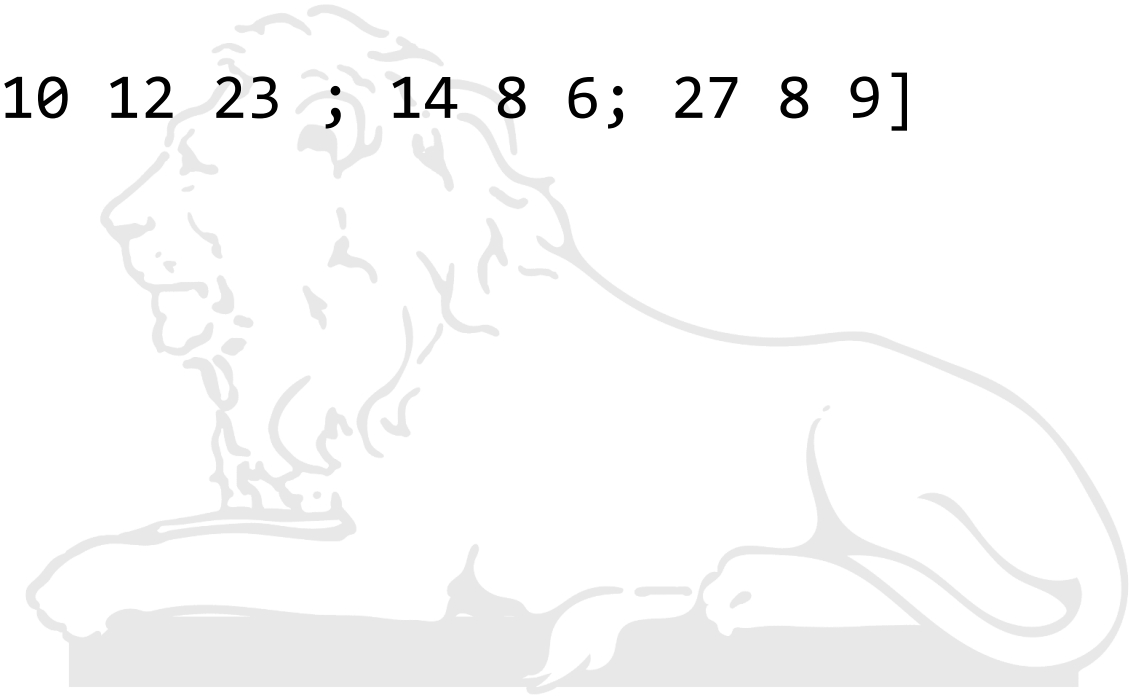
- When you add, subtract, multiply or divide a matrix by a number, this is called the **scalar operation**.
- Scalar operations produce a new matrix with same number of rows and columns with each element of the original matrix added to, subtracted from, multiplied by or divided by the number.

```
>> a = [ 10 12 23 ; 14 8 6; 27 8 9];
>> b = 2;
>> c = a + b
>> d = a - b
>> e = a * b
>> f = a / b
```

Transpose of a Matrix

- The transpose operation switches the rows and columns in a matrix. It is represented by a single quote(‘).

```
>> a = [ 10 12 23 ; 14 8 6; 27 8 9]
>> b = a'
```



Concatenating Matrices

- You can concatenate two matrices to create a larger matrix. The pair of square brackets '[]' is the concatenation operator.
- MATLAB allows two types of concatenations −
  - **Horizontal concatenation**: When you concatenate two matrices by separating those using commas, they are just appended horizontally..
  - **Vertical concatenation**: Alternatively, if you concatenate two matrices by separating those using semicolons, they are appended vertically.

```
>> a = [ 10 12 23 ; 14 8 6; 27 8 9]
>> b = [ 12 31 45 ; 8 0 -9; 45 2 11]
>> c = [a, b]
>> d = [a; b]
```

Matrix Multiplication

- Consider two matrices A and B. If A is an m x n matrix and B is an n x p matrix, they could be multiplied together to produce an m x p matrix C.
- Matrix multiplication is possible only if the number of columns in in A is equal to the number of rows n in B.

```
>> a = [ 1 2 3; 2 3 4; 1 2 5]
>> b = [ 2 1 3 ; 5 0 -2; 2 3 -1]
>> prod = a * b
```

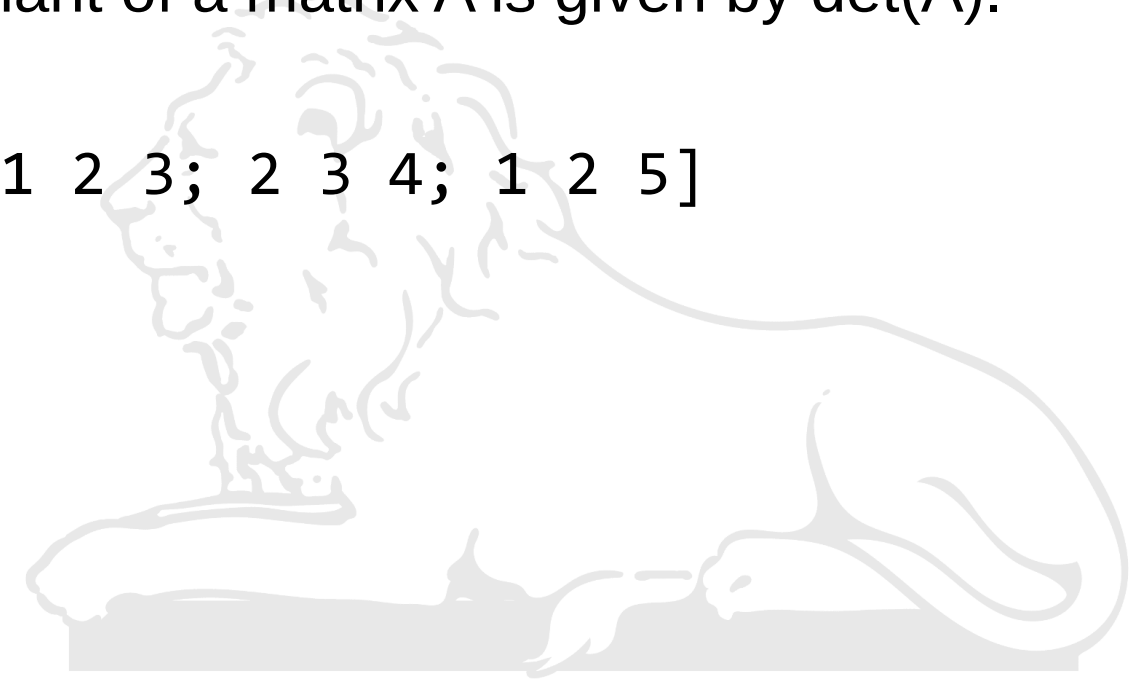
What happens if I do

```
(1) a.*b   (2) a^2   (3) a.^2
```

Determinant of a matrix

- Determinant of a matrix is calculated using the **det** function of MATLAB.
- Determinant of a matrix A is given by det(A).

```
>> a = [ 1 2 3; 2 3 4; 1 2 5]
>> det(a)
```



Inverse of a matrix

- The inverse of a matrix A is denoted by A^{-1} such that the following relationship holds −

  $$AA^{-1} = A^{-1}A = 1$$

- The inverse of a matrix does not always exist. If the determinant of the matrix is zero, then the inverse does not exist and the matrix is singular.
- Inverse of a matrix in MATLAB is calculated using the **inv** function. Inverse of a matrix A is given by inv(A).

```
>> a = [ 1 2 3; 2 3 4; 1 2 5]
>> inv(a)
```

Special Arrays in MATLAB

- Zeros: The **zeros()** function creates an array of all zeros
- The **ones()** function creates an array of all ones
- The **rand()** function creates an array of uniformly distributed random numbers on (0,1)
- Length, size, ndims, numel

What is MATLAB?

- MATLAB is a powerful, interactive software tool that can be employed with relative ease in solving science and engineering problems
- Started in the 1970s. It has now evolved into a huge system comprising of built-in functions and toolboxes
- User community size $\geq$ 500,000 (industry, government & academia)
- More than 600 built-in functions (mathematical, statistical and engineering) that are reliable, accurate and fast

What is MATLAB?

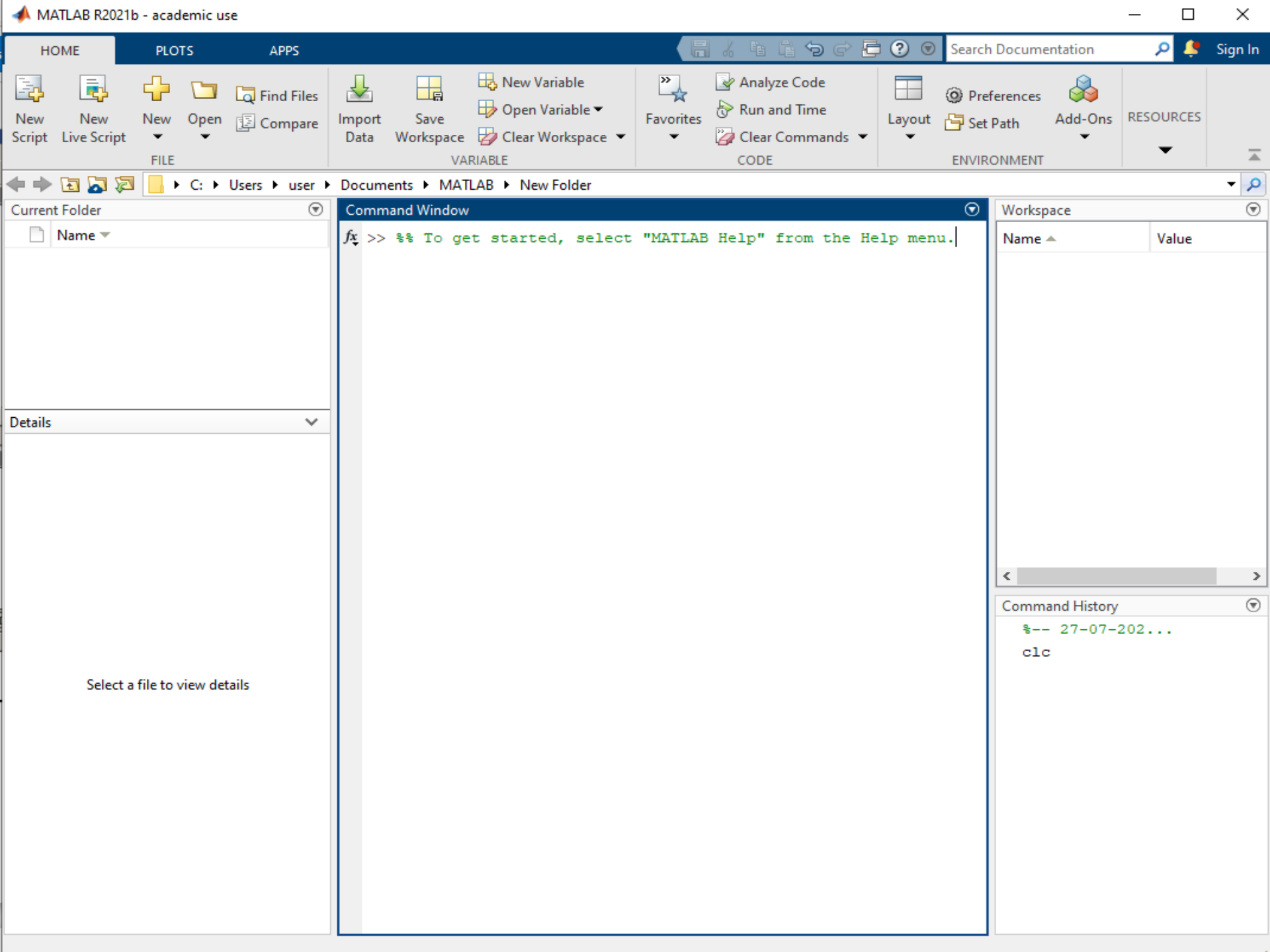
- You can use MATLAB built-in functions and other external resources to develop your own toolboxes – Third Party Products
- Excellent graphics to analyze and visualize your data. Can generate ps, eps, wmf, bmp, jpeg, tiff, …… files and insert figures into WORD, LATEX etc.
- Interfaces external languages, such as C, C++, Fortran, and Java. Can convert MATLAB to C & C++
- Support for importing data from files and external devices and for using low-level file I/O (plus access to databases and additional hardware via add-on products)

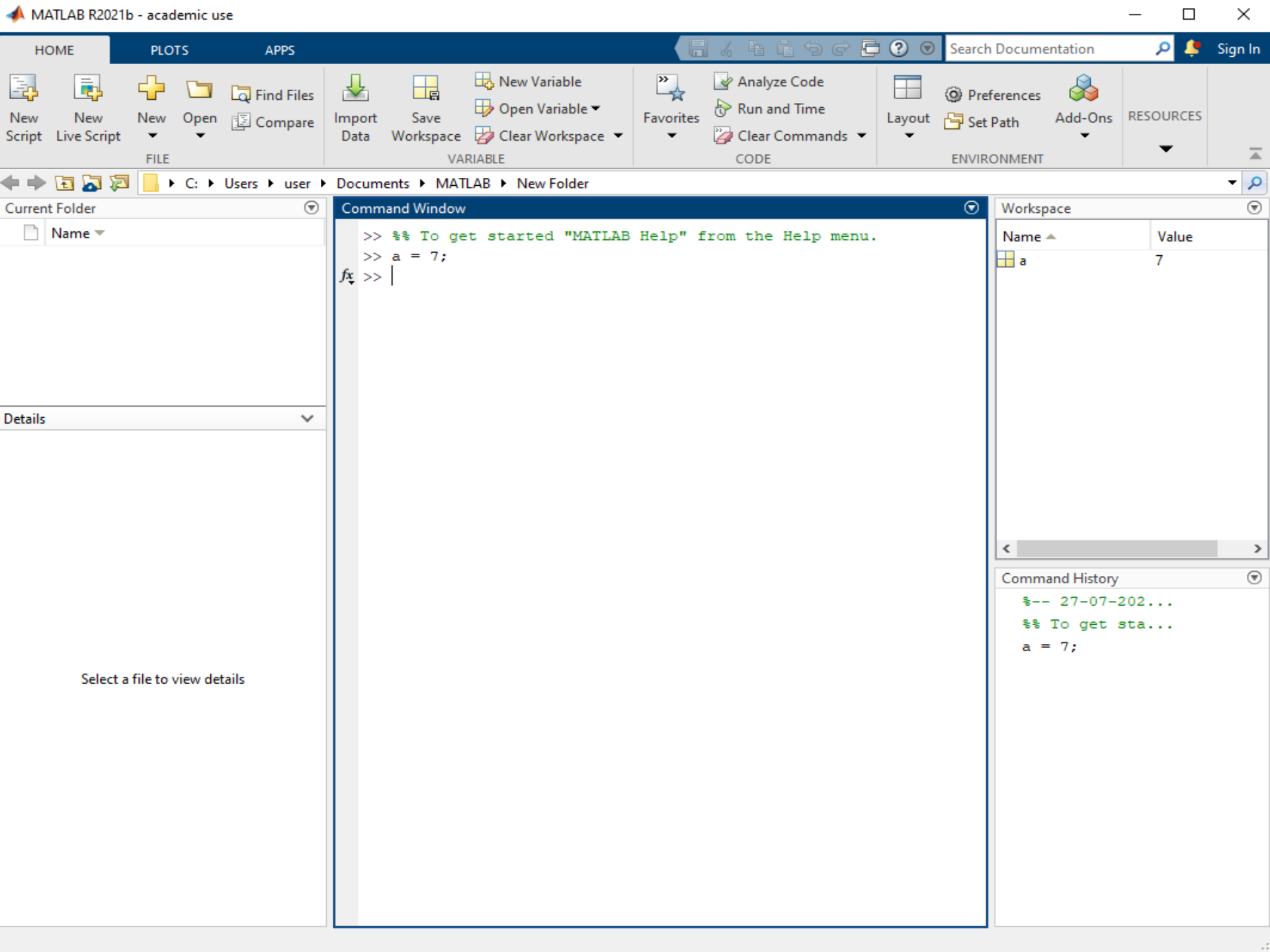
Disadvantages of MATLAB

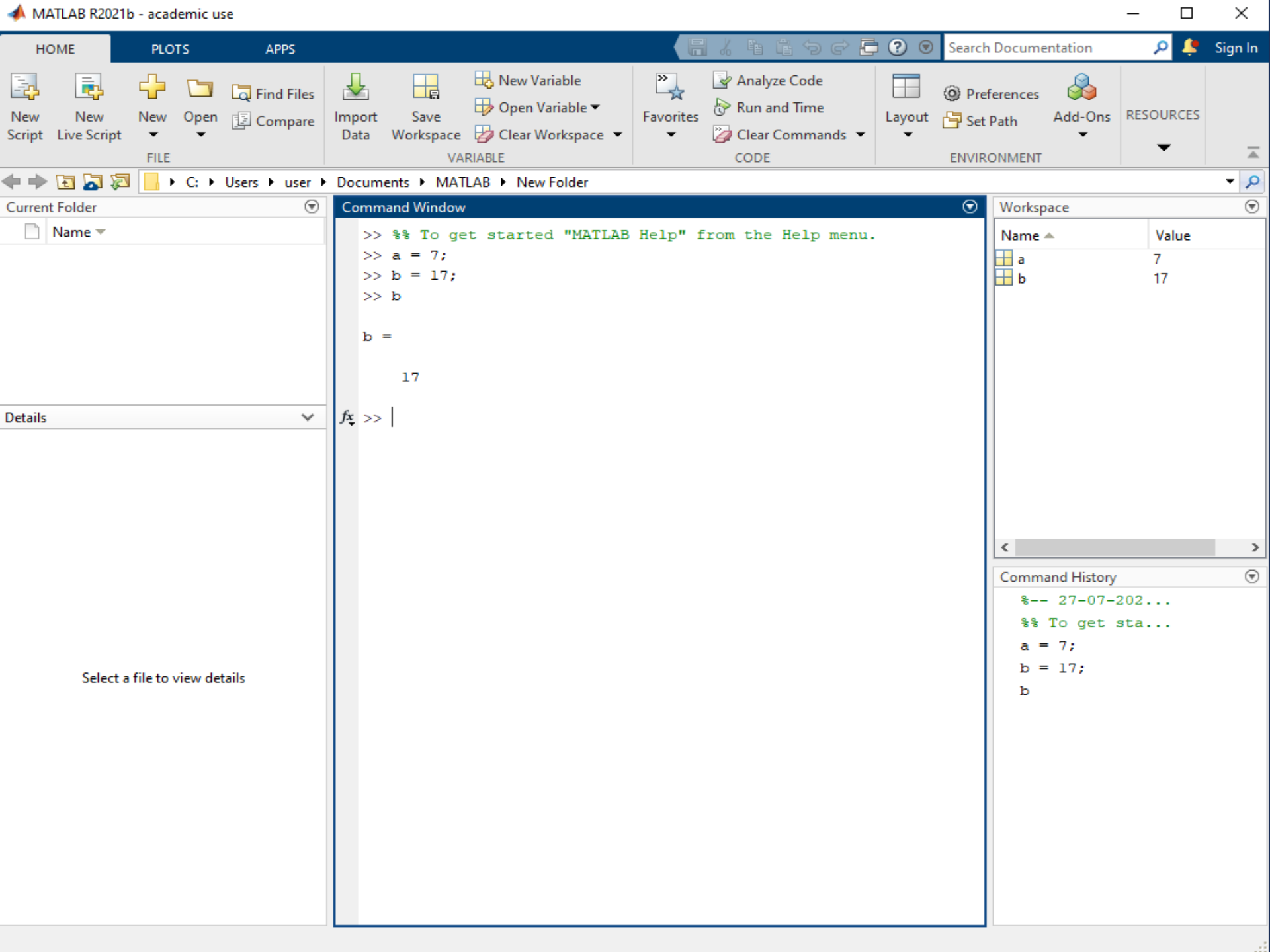
- Cost – about 5~10 times more expensive than conventional Fortran or C compiler

- It is an interpreter-based language and not compiler-based. So, it executes programs line by line and not as a compiled object.
  - Can make MATLAB extremely slow, particularly with "loops"

- Can be overcome by properly structuring the MATLAB program or by using a MATLAB to C compiler to create an "executable".
  - This has the additional advantage that your source code can be protected


MATLAB R2021b - academic use
HOME
PLOTS
APPS
Search Documentation
Sign In
New Script
New Live Script
New
Open
Find Files
Compare
FILE
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
VARIABLE
Favorites
Analyze Code
Run and Time
Clear Commands
CODE
Layout
Preferences
Set Path
Add-Ons
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder
Current Folder
Name
Details
Select a file to view details
Command Window
>> %% To get started, select "MATLAB Help" from the Help menu.
Workspace
Name
Value
Command History
%-- 27-07-202...
clc


MATLAB R2021b - academic use
HOME
PLOTS
APPS
Search Documentation
Sign In
New Script
New Live Script
New
Open
Find Files
Compare
FILE
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
VARIABLE
Favorites
Analyze Code
Run and Time
Clear Commands
CODE
Layout
Preferences
Set Path
Add-Ons
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder
Current Folder
Name
Details
Select a file to view details
Command Window
>> %% To get started "MATLAB Help" from the Help menu.
>> a = 7;
>>
Workspace
Name
Value
a
7
Command History
%-- 27-07-202...
%% To get sta...
a = 7;


MATLAB R2021b - academic use
HOME
PLOTS
APPS
Search Documentation
Sign In
New Script
New Live Script
New
Open
Find Files
Compare
FILE
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
VARIABLE
Favorites
Analyze Code
Run and Time
Clear Commands
CODE
Layout
Preferences
Set Path
Add-Ons
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder
Current Folder
Name
Details
Select a file to view details
Command Window
>> %% To get started "MATLAB Help" from the Help menu.
>> a = 7;
>> b = 17;
>> b
b =
17
>>
Workspace
Name
Value
a
7
b
17
Command History
%-- 27-07-202...
%% To get sta...
a = 7;
b = 17;
b

MATLAB R2021b - academic use

HOME PLOTS APPS

Search Documentation

Sign In

New Script | New Live Script | New | Open | Find Files | Compare — FILE

Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace — VARIABLE

Favorites | Analyze Code | Run and Time | Clear Commands — CODE

Layout | Preferences | Set Path | Add-Ons — ENVIRONMENT

RESOURCES

C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder

Current Folder

Name

Details

Select a file to view details

Command Window

```
>> %% To get started "MATLAB Help" from the Help menu.
>> a = 7;
>> b = 17;
>> b

b =

    17

>> c = a + b

c =

    24

>>
```

Workspace

Name	Value
a	7
b	17
c	24

Command History

```
%-- 27-07-202...
%% To get sta...
a = 7;
b = 17;
b
c = a + b
```

MATLAB R2021b - academic use

HOME PLOTS APPS

Search Documentation

Sign In

New Script | New Live Script | New | Open | Find Files | Compare — FILE

Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace — VARIABLE

Favorites | Analyze Code | Run and Time | Clear Commands — CODE

Layout | Preferences | Set Path | Add-Ons — ENVIRONMENT

RESOURCES

C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder

Current Folder

Name

Details

Select a file to view details

Command Window

```
>> %% To get started "MATLAB Help" from the Help menu.
>> a = 7;
>> b = 17;
>> b

b =

    17

>> c = a + b

c =

    24

>> whos
  Name      Size              Bytes  Class     Attributes

  a         1x1                   8  double
  b         1x1                   8  double
  c         1x1                   8  double

>> %% Grand total is 3 elements using 24 bytes
>>
```

Workspace

Name	Value
a	7
b	17
c	24

Command History

```
%-- 27-07-202...
%% To get sta...
a = 7;
b = 17;
b
c = a + b
whos
%% Grand tota...
```

MATLAB R2021b - academic use

HOME PLOTS APPS

Search Documentation

Sign In

New Script | New Live Script | New | Open | Find Files | Compare — FILE

Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace — VARIABLE

Favorites | Analyze Code | Run and Time | Clear Commands — CODE

Layout | Preferences | Set Path | Add-Ons — ENVIRONMENT

RESOURCES

C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder

Current Folder

Name

Details

Select a file to view details

Command Window

```
>> %% To get started "MATLAB Help" from the Help menu.
>> a = 7;
>> b = 17;
>> b

b =

    17

>> c = a + b

c =

    24

>> whos
  Name        Size              Bytes  Class     Attributes

  a           1x1                   8  double
  b           1x1                   8  double
  c           1x1                   8  double

>> %% Grand total is 3 elements using 24 bytes
>> clear
>>
```

Workspace

Name	Value

Command History

```
%-- 27-07-202...
%% To get sta...
a = 7;
b = 17;
b
c = a + b
whos
%% Grand tota...
clear
```

MATLAB R2021b - academic use

HOME PLOTS APPS

Search Documentation

Sign In

New Script | New Live Script | New | Open | Find Files | Compare — FILE

Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace — VARIABLE

Favorites | Analyze Code | Run and Time | Clear Commands — CODE

Layout | Preferences | Set Path | Add-Ons — ENVIRONMENT

RESOURCES

C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder

Current Folder

Name

Details

Select a file to view details

Command Window

```
>> A = [9 8 7;6 5 4;3 2 1]

A =

     9     8     7
     6     5     4
     3     2     1

>> B = [18 17 16;15 14 13;12 11 10]

B =

    18    17    16
    15    14    13
    12    11    10

>> C = A + B

C =

    27    25    23
    21    19    17
    15    13    11

>> whos
  Name        Size              Bytes  Class     Attributes

  A           3x3                  72  double
  B           3x3                  72  double
  C           3x3                  72  double

>> %% Grand total is 27 using 216 bytes
```

Workspace

Name	Value
A	[9,8,7;6,5,4;3,2,1]
B	[18,17,16;15,14,13;
C	[27,25,23;21,19,17;

Command History

```
%-- 27-07-202...
A = [9 8 7;6 ...
B = [18 17 16...
C = A + B
whos
```

MATLAB R2021b - academic use

HOME PLOTS APPS

Search Documentation | Sign In

New Script | New Live Script | New | Open | Find Files | Compare — FILE

Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace — VARIABLE

Favorites | Analyze Code | Run and Time | Clear Commands — CODE

Layout | Preferences | Set Path | Add-Ons — ENVIRONMENT

RESOURCES

C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder

Current Folder

Name

Details

Select a file to view details

Command Window

```
    27    25    23
    21    19    17
    15    13    11

>> whos
  Name      Size              Bytes  Class     Attributes

  A         3x3                  72  double
  B         3x3                  72  double
  C         3x3                  72  double

>> %% Grand total is 27 using 216 bytes
>> A*B

ans =

   366   342   318
   231   216   201
    96    90    84

>> F = inv(A)
Warning: Matrix is close to singular or badly scaled. Results
may be inaccurate. RCOND =  3.083953e-18.


F =

   1.0e+15 *

    2.2518   -4.5036    2.2518
   -4.5036    9.0072   -4.5036
    2.2518   -4.5036    2.2518

fx >>
```

Workspace

Name	Value
A	[9,8,7;6,5,4;3,2,1]
ans	[366,342,318;231,2
B	[18,17,16;15,14,13;
C	[27,25,23;21,19,17;
F	[2.2518e+15,-4.50

Command History

```
%-- 27-07-202...
A = [9 8 7;6 ...
B = [18 17 16...
C = A + B
whos
%% Grand tota...
A*B
F = inv(A)
```

MATLAB R2021b - academic use

HOME PLOTS APPS

Search Documentation

Sign In

New Script | New Live Script | New | Open | Find Files | Compare — FILE

Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace — VARIABLE

Favorites | Analyze Code | Run and Time | Clear Commands — CODE

Layout | Preferences | Set Path | Add-Ons — ENVIRONMENT

RESOURCES

C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder

Current Folder

Name

Details

Select a file to view details

Command Window

```
>> A = [9 8 7;6 5 4;3 2 1]

A =

     9     8     7
     6     5     4
     3     2     1

>> A'

ans =

     9     6     3
     8     5     2
     7     4     1

>> A^2

ans =

   150   126   102
    96    81    66
    42    36    30

>> A.^2

ans =

    81    64    49
    36    25    16
     9     4     1

fx >>
```

Workspace

Name	Value
A	[9,8,7;6,5,4;3,2,1]
ans	[81,64,49;36,25,16;
B	[18,17,16;15,14,13;
C	[27,25,23;21,19,17;
F	[2.2518e+15,-4.50

Command History

```
%-- 27-07-202...
A = [9 8 7;6 ...
B = [18 17 16...
C = A + B
whos
%% Grand tota...
A*B
F = inv(A)
clc
A = [9 8 7;6 ...
A'
A^2
A.^2
```

MATLAB R2021b - academic use

HOME PLOTS APPS

Search Documentation

Sign In

New Script | New Live Script | New | Open | Find Files | Compare — FILE

Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace — VARIABLE

Favorites | Analyze Code | Run and Time | Clear Commands — CODE

Layout | Preferences | Set Path | Add-Ons — ENVIRONMENT

RESOURCES

C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder

Current Folder

Name

Details

Select a file to view details

Command Window

```
>> A(1:2,2:3)

ans =

     8     7
     5     4

>> [rank(A) det(A)]

ans =

    2.0000   -0.0000

>> [V,D] = eig(A)

V =

   -0.8187   -0.6123    0.4082
   -0.5253    0.0868   -0.8165
   -0.2320    0.7858    0.4082


D =

   16.1168         0         0
         0   -1.1168         0
         0         0   -0.0000

>>
```

Workspace

Name	Value
A	[9,8,7;6,5,4;3,2,1]
ans	[2,-1.3323e-15]
B	[18,17,16;15,14,13;
C	[27,25,23;21,19,17;
D	[16.1168,0,0;0,-1.1
F	[2.2518e+15,-4.50
V	[-0.8187,-0.6123,0.

Command History

```
F = inv(A)
clc
A = [9 8 7;...
A'
A^2
A.^2
clc
A(4:5,6:8)          0.20 sec
clc
A(3:4,4:5)
clc
A(1:2,2:3)
[rank(A) de...
[V,D] = eig(A)
```

A short interactive session

```
>> A=[1 2 3;4 5 6;7 8 9];
>> for i=1:3
for j=1:3
if A(i,j) <= 5
A(i,j)=0;
end
end
end
>> A

A =

     0     0     0
     0     0     6
     7     8     9
```

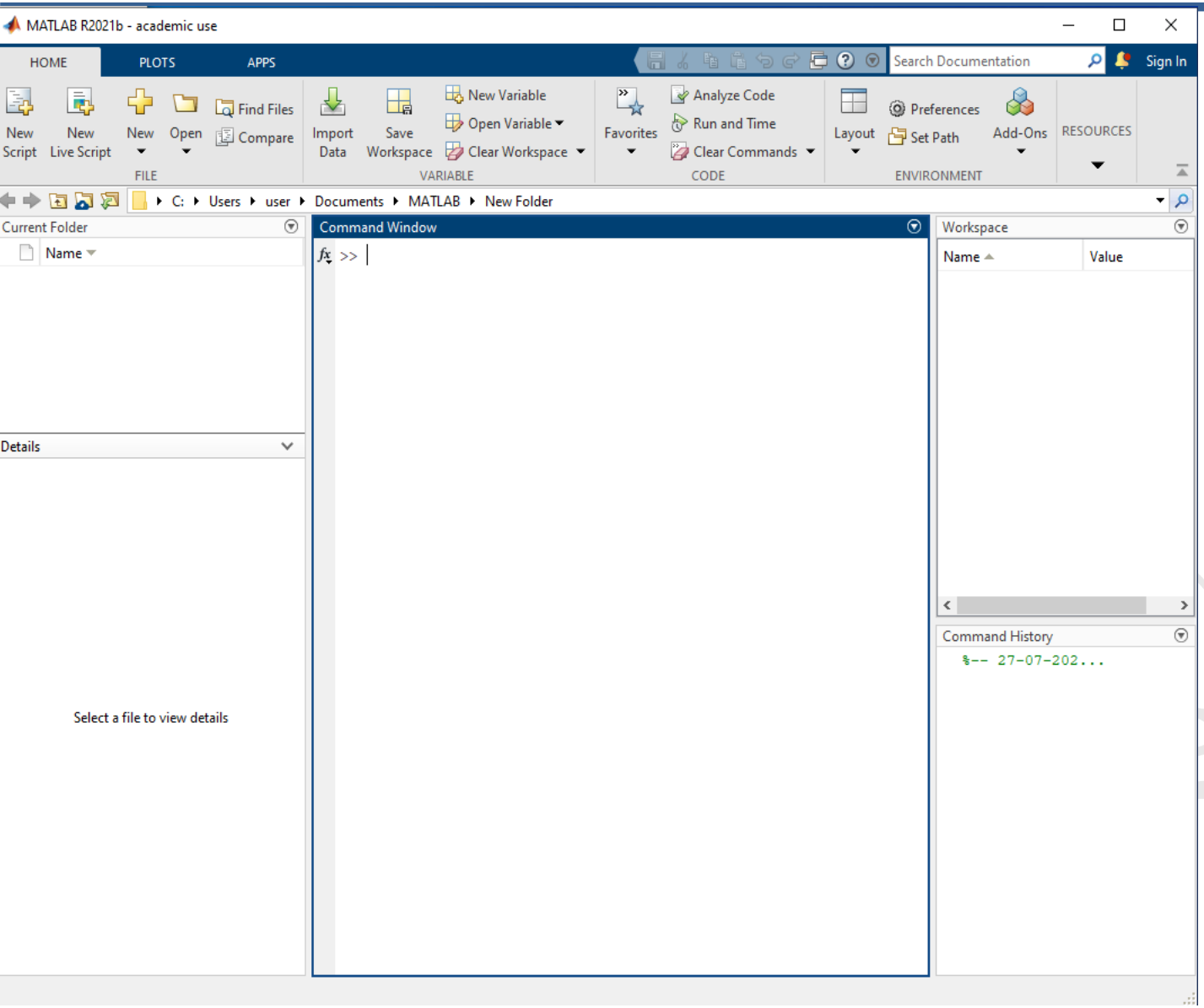
Another short interactive session

```
>> A=[1 2 3;4 5 6;7 8 9];
>> B=[ ];
>> for i=1:3
B=[B;A(i,:)'];
end
>> B

B =

   1
   2
   3
   4
   5
   6
   7
   8
   9
```

Writing M-Files

- Interactive Programming is useful only in the initial stages – when you are learning the syntax, commands etc.
- Very soon you should graduate into writing what are called m-files.
- M-files are text files that contain a sequence of MATLAB commands to achieve the required goals.
- You can enter the commands into a file and save it. Then just type in the file name inside the MATLAB command window. All the commands will be executed and the results will be available in the MATLAB workspace

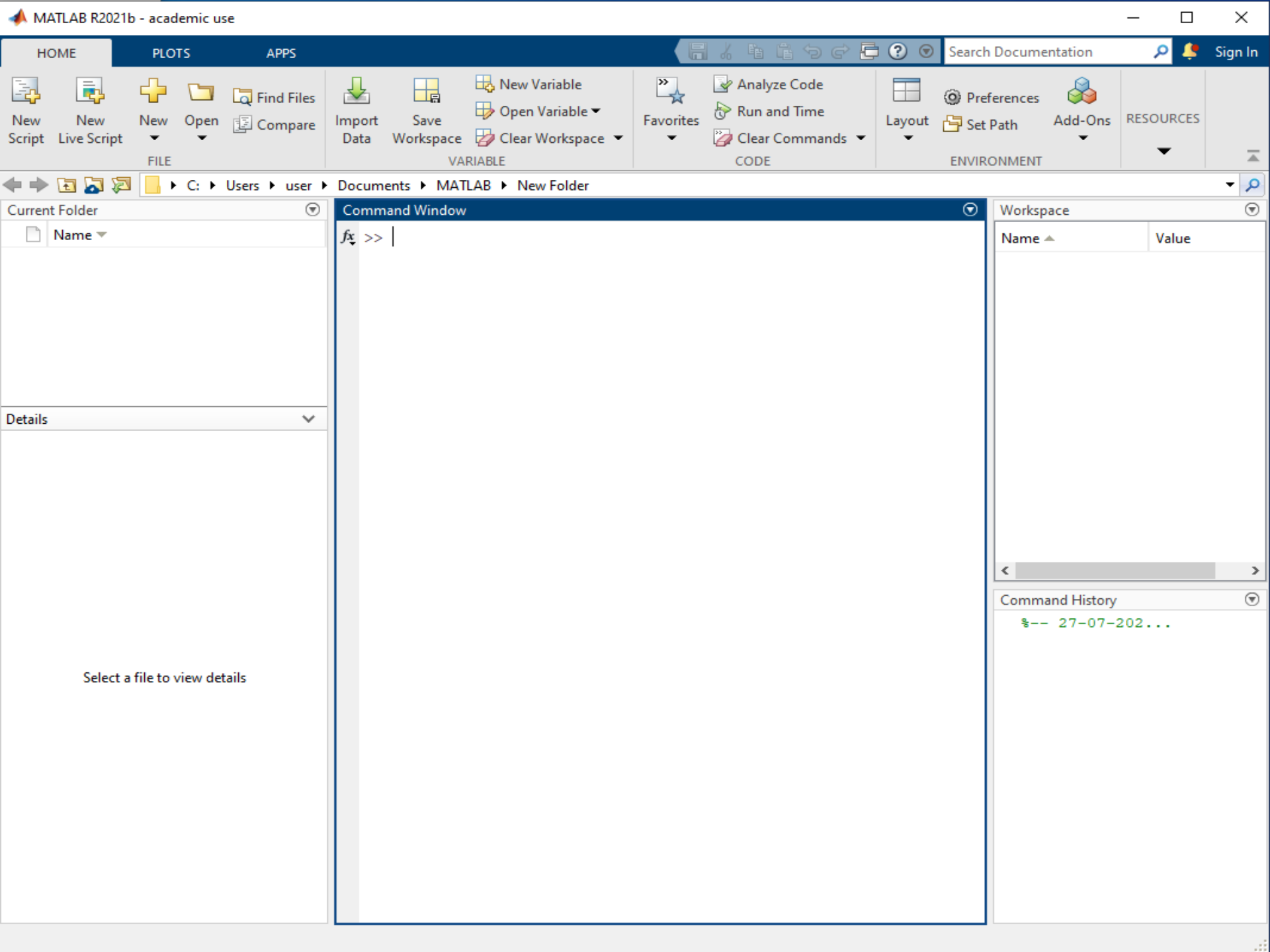
Writing M-Files



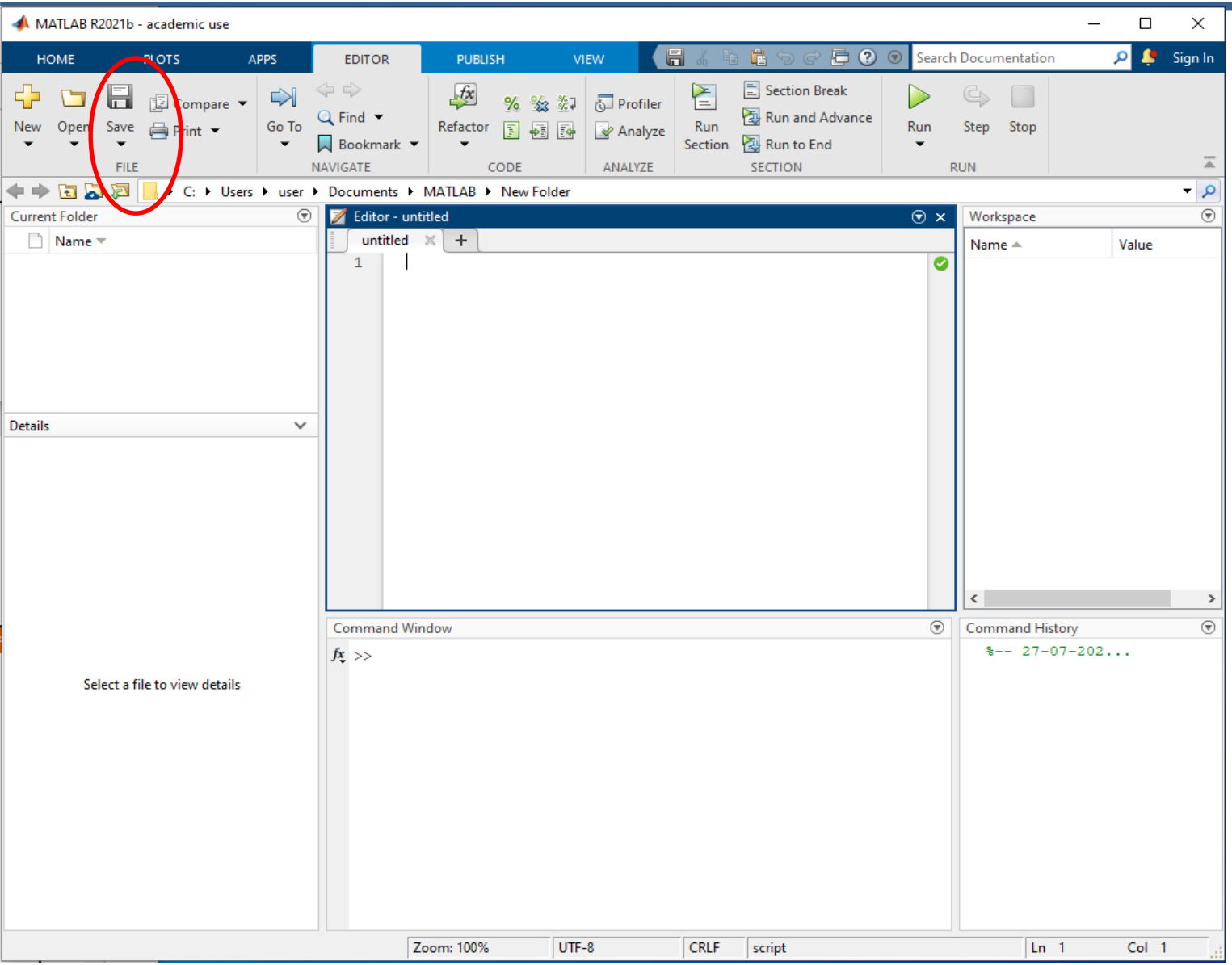
Choose

Files
↓
New
↓
M-file

This will open up a new window for entering the MATLAB commands


MATLAB R2021b - academic use
HOME
PLOTS
APPS
Search Documentation
Sign In
New Script
New Live Script
New
Open
Find Files
Compare
FILE
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
VARIABLE
Favorites
Analyze Code
Run and Time
Clear Commands
CODE
Layout
Preferences
Set Path
Add-Ons
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder
Current Folder
Name
Details
Select a file to view details
Command Window
>>
Workspace
Name
Value
Command History
%-- 27-07-202...

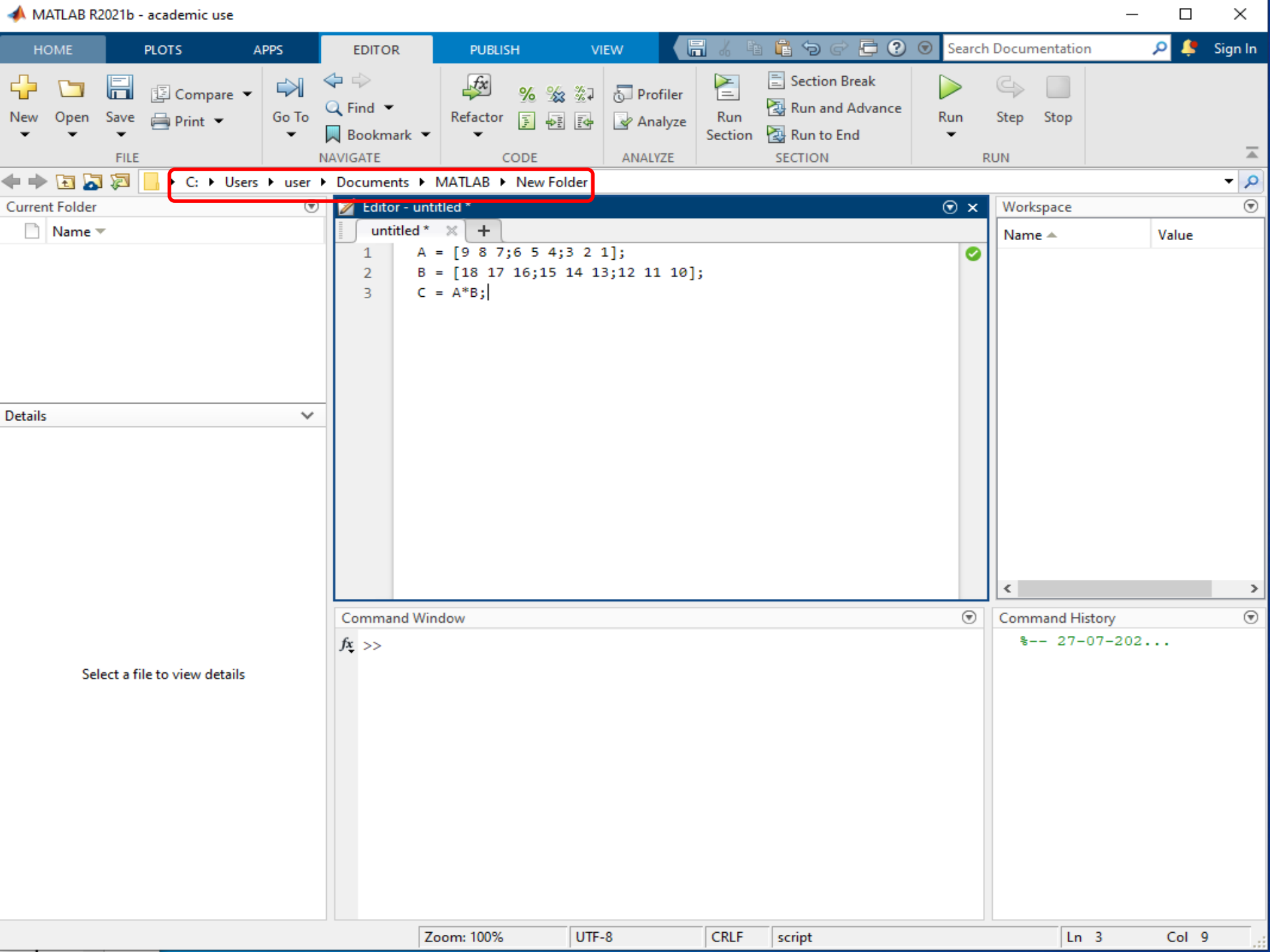
Writing M-Files

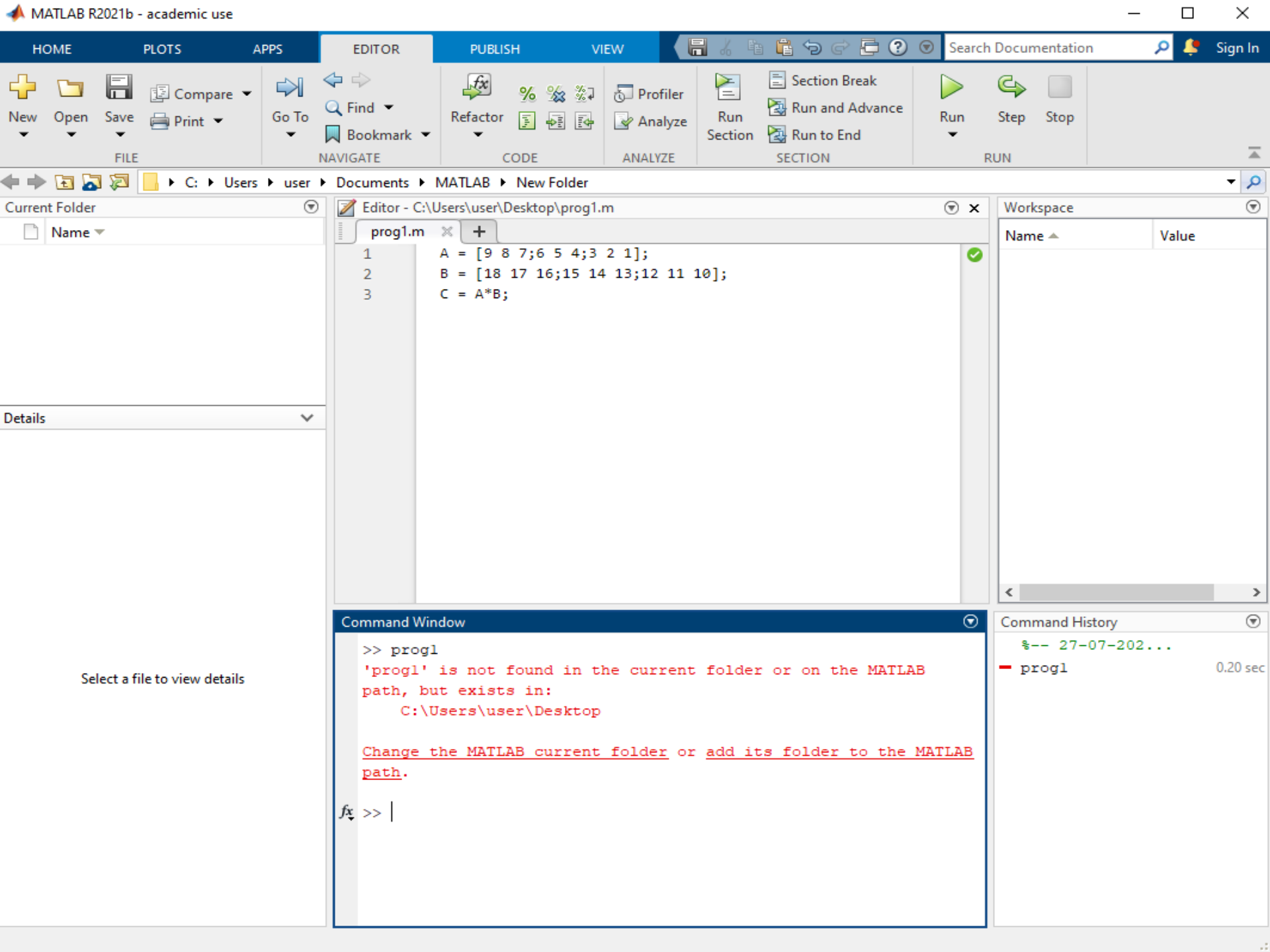


Choose

File
↓
Save

and specify the directory and a name for the file


MATLAB R2021b - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder
Current Folder
Name
Details
Select a file to view details
Editor - untitled *
untitled *
1 A = [9 8 7;6 5 4;3 2 1];
2 B = [18 17 16;15 14 13;12 11 10];
3 C = A*B;
Workspace
Name
Value
Command Window
>>
Command History
%-- 27-07-202...
Zoom: 100%
UTF-8
CRLF
script
Ln 3
Col 9


MATLAB R2021b - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Compare
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Profiler
Analyze
ANALYZE
Run Section
Section Break
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder
Current Folder
Name
Details
Select a file to view details
Editor - C:\Users\user\Desktop\prog1.m
prog1.m
1 A = [9 8 7;6 5 4;3 2 1];
2 B = [18 17 16;15 14 13;12 11 10];
3 C = A*B;
Workspace
Name
Value
Command Window
>> prog1
'prog1' is not found in the current folder or on the MATLAB path, but exists in:
C:\Users\user\Desktop
Change the MATLAB current folder or add its folder to the MATLAB path.
>>
Command History
%-- 27-07-202...
prog1
0.20 sec

MATLAB R2021b - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

Search Documentation

Sign In

New Open Save Compare Print FILE

Go To Find Bookmark NAVIGATE

Refactor CODE

Profiler Analyze ANALYZE

Run Section Section Break Run and Advance Run to End SECTION

Run Step Stop RUN

C: ▸ Users ▸ user ▸ Documents ▸ MATLAB ▸ New Folder

Current Folder

Name

Details

Select a file to view details

Editor - C:\Users\user\Desktop\prog1.m

prog1.m

```
1  A = [9 8 7;6 5 4;3 2 1];
2  B = [18 17 16;15 14 13;12 11 10];
3  C = A*B;
```

Workspace

Name	Value
ans	'C:\Users\user\Do

Command Window

```
path, but exists in:
    C:\Users\user\Desktop

Change the MATLAB current folder or add its folder to the MATLA
path.

>> pwd

ans =

    'C:\Users\user\Documents\MATLAB\New Folder'

>>
```

Command History

```
%-- 27-07-202...
prog1                0.20 sec
pwd
```

matlab:matlab.internal.language.commandline.executeCode('cd ''C:\Users\user\Desktop''')

CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

Set of equations

- Determine the values $x_1, x_2, x_3, \ldots, x_n$ that simultaneously satisfy a set of equations

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &= 0 \\
f_2(x_1, x_2, \ldots, x_n) &= 0 \\
&\vdots \\
f_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}
$$

- Equations can be linear/nonlinear

Set of linear algebraic equations

$$
\begin{aligned}
a_{11}\,x_1 + a_{12}\,x_2 + \cdots + a_{1n}\,x_n &= b_1 \\
a_{21}\,x_1 + a_{22}\,x_2 + \cdots + a_{2n}\,x_n &= b_2 \\
&\vdots \\
a_{n1}\,x_1 + a_{n2}\,x_2 + \ldots + a_{nn}\,x_n &= b_n
\end{aligned}
$$

- a's are constant coefficients
- b's are constants
- x's are unknowns
- n is the number of equations

Importance of linear equations

- Linear equations are the basis for mathematical models of
  - economics,
  - weather prediction,
  - heat and mass transfer,
  - statistical analysis, and
  - a myriad of other applications.
- The methods for solving ordinary and partial differential equations depend on them

Set of linear equations and matrix algebra

$$
\begin{aligned}
a_{11}\,x_1 + a_{12}\,x_2 + \cdots + a_{1n}\,x_n &= b_1 \\
a_{21}\,x_1 + a_{22}\,x_2 + \cdots + a_{2n}\,x_n &= b_2 \\
&\vdots \\
a_{n1}\,x_1 + a_{n2}\,x_2 + \ldots + a_{mn}\,x_n &= b_n
\end{aligned}
$$

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} =
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}
$$

$$AX = B$$

where $\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$, $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, $\mathrm{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

Matrix algebra

- Square matrix
- Symmetric matrix
- Diagonal matrix
- Identity matrix
- Upper triangular matrix
- Lower triangular matrix

Matrix algebra

- Matrix addition and subtraction
  - Commutative $[A]+[B]= [B]+[A]$
  - Associative $([A]+[B])+[C]= [A]+([B]+[C])$

- Multiplication of a matrix with a scalar

- Multiplication of two matrices
  - Associative $([A][B])[C]= [A]([B][C])$
  - Distributive $[A]([B]+[C])= [A][B]+[A][C]$
  - Not commutative $[A][B] \neq [B][A]$

Determinant of a matrix

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$|\mathrm{A}| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = (ad - bc)$$

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

$$= a(ei - fh) - b(di - fg) + c(dh - eg)$$

Solving linear equations in MATLAB

- $AX = B$

```
>> a=[1 2; 3, 4];
>> b=[3, 5];
>> b=b';
>> x=a\b

x =

    -1
     2
```

```
>> x=inv(A)*b
Unrecognized function or variable 'A'.

Did you mean:
>> x=inv(a)*b

x =

   -1.0000
    2.0000
```

Solving linear equation systems

- Three types of methods
  - Direct methods
  - Iterative methods
  - Decomposition methods

Solving linear equation systems

- A is a nxn matrix

$$[A][A]^{-1} = [A]^{-1}[A] = [I]$$

- Our system, $AX = B$
- Multiply both sides by A^{-1}

$$A^{-1}AX = A^{-1}B$$

$$X = A^{-1}B$$

$$A^{-1} = \frac{1}{|A|} \cdot \text{Adj } A$$

- Adj A means the adjoint matrix of A
- The adjoint of a matrix is the transpose of the cofactor element matrix of the given matrix.

Cramer's rule

- Our system $AX = B$
- Solution is given by

$$x_j = \frac{\det(A_j)}{\det(A)} \quad ; \det(A) \neq 0 \; ; \; j = 1, \cdots, n$$

- A_j is obtained by replacing the j^{th} column of A by B

Cramer's rule

- Total number of determinants required = n+1
- Number of operations to calculate determinant of a nxn matrix = (n-1) (n!)
- Total operations in Cramer's rule= (n+1) (n-1) (n!)

  $\sim n^2 n!$

For a 100x100 matrix: $100^2\ 100! = 10^{162}$ calculations

Gauss elimination

- Consider the system

$$x_1+x_2+2x_3 = 3$$
$$2x_1+3x_2+x_3 = 2$$
$$3x_1-x_2-x_3 = 6$$

- Let us perform the following row operations

Old Eq (2) – 2 Eq (1) → New Eq (2): $x_2 - 3x_3 = -4$

Old Eq (3) – 3 Eq (1) → New Eq (3): $-4x_2 - 7x_3 = -3$

- The equivalent system of equations is

$$x_1+x_2+2x_3 = 3$$
$$x_2-3x_3 = -4$$
$$-4x_2-7x_3 = -3$$

We have eliminated x_1 from Eq. (2) and (3).

Gauss elimination contd…

- Let us perform another elementary row operation

  Old Eq (3) + 4 Eq (2) → New Eq (3): $-19x_3 = -19$

- The new equivalent system of equations is

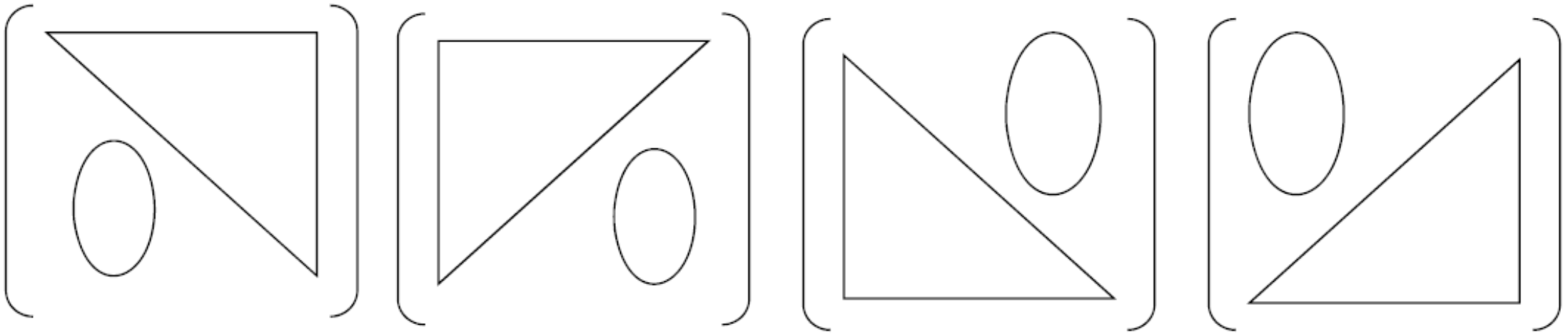
$$x_1 + x_2 + 2x_3 = 3$$
$$0x_1 + x_2 - 3x_3 = -4$$
$$0x_1 + 0x_2 - 19x_3 = -19$$

We have eliminated x_2 from Eq. (3).

- Now, if we write in matrix format, the coefficient matrix is upper triangular matrix
  - That can be easily solved
  - Solve Eq. (3) to get $x_3=1$
  - Substitute $x_3=1$ in Eq. (2) to get $x_2= -1$
  - Substitute $x_3=1$ and $x_2= -1$ in Eq. (1) to get $x_1= 2$

Gauss elimination contd…

- Idea behind: Convert "full" coefficient matrix into a lower or upper triangular matrix (Note: the constant matrix on RHS will also be modified)



Upper Triangular Structure

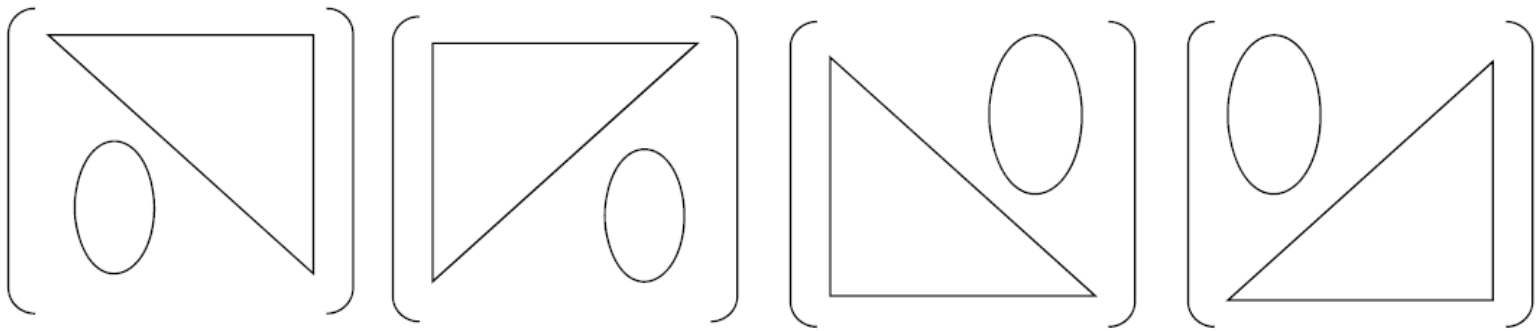
Reverse Upper Triangular Structure

Lower Triangular Structure

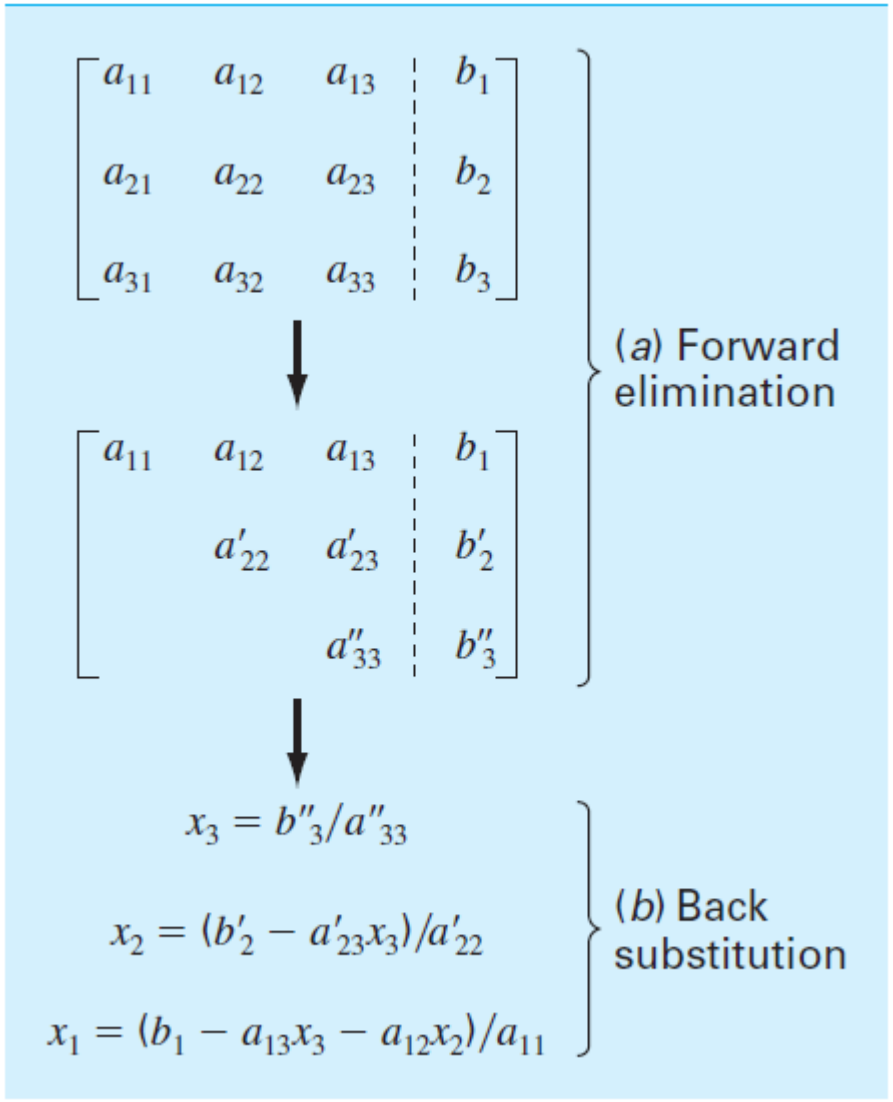
Reverse Lower Triangular Structure

Gauss elimination contd…

- If coefficient matrix is
  - Upper triangular, the system can be solved in sequence x_n, x_{n-1}, …, x_1 using equations in order n, n-1, …, 1
  - Reverse upper triangular, the system can be solved in sequence x_1, x_2, …, x_n using equations in order n, n-1, …, 1
  - Lower triangular, the system can be solved in sequence x_1, x_2, …, x_n using equations in order 1, 2, …, n
  - Reverse lower triangular, the system can be solved in sequence x_n, x_{n-1}, …, x_1 using equations in order 1, 2, …, n



Two stages of Gauss elimination



Forward elimination: generalization

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

- First, we eliminate x_1 from Eq. (2) and (3). To do so, we perform the following row operations

  New Eq (2) = Old Eq (2) – (a_{21}* Old Eq (1) / a_{11})

  New Eq (3) = Old Eq (3) – (a_{31}* Old Eq (1) / a_{11})

- a_{11} is the pivot element in the elimination of x_1 from Eq (2)&(3)

Forward elimination: generalization

- The new equivalent system is

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$\left(a_{22} - \left(\frac{a_{21}a_{12}}{a_{11}}\right)\right)x_2 + \left(a_{23} - \left(\frac{a_{21}a_{13}}{a_{11}}\right)\right)x_3 = b_2 - \frac{a_{21}b_1}{a_{11}}$$

$$\left(a_{32} - \left(\frac{a_{31}a_{12}}{a_{11}}\right)\right)x_2 + \left(a_{33} - \left(\frac{a_{31}a_{13}}{a_{11}}\right)\right)x_3 = b_3 - \frac{a_{31}b_1}{a_{11}}$$

- In other words,

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a'_{22}x_2 + a'_{23}x_3 = b'_2$$

$$a'_{32}x_2 + a'_{33}x_3 = b'_3$$

Diagonal elements are pivot elements

- Let us perform another elementary row operation

New Eq (3) = Old Eq (3) – ($a'32$ * Old Eq (2) / $a'22$)

$a'22$ is the pivot element in the elimination of x_2 from Eq (3)

Forward elimination: generalization

- The new equivalent system is

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a'_{22}x_2 + a'_{23}x_3 = b'_2$$

$$\left(a'_{33} - \left(\frac{a'_{32}a'_{23}}{a'_{22}}\right)\right)x_3 = b'_3 - \frac{a_{32}b_2}{a_{22}}$$

- In other words,

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a'_{22}x_2 + a'_{23}x_3 = b'_2$$

$$a''_{33}x_3 = b''_3$$

This is a upper triangular system and can be solved in sequential order x_3, x_2 and x_1

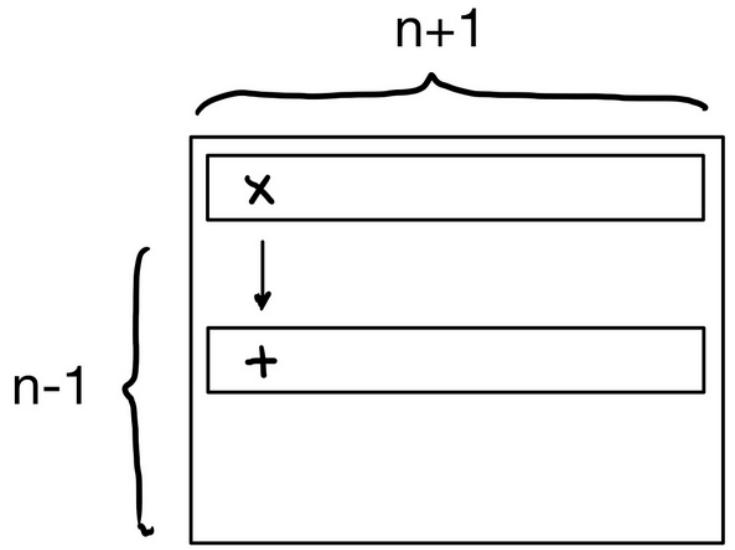
Forward elimination: Number of calculations

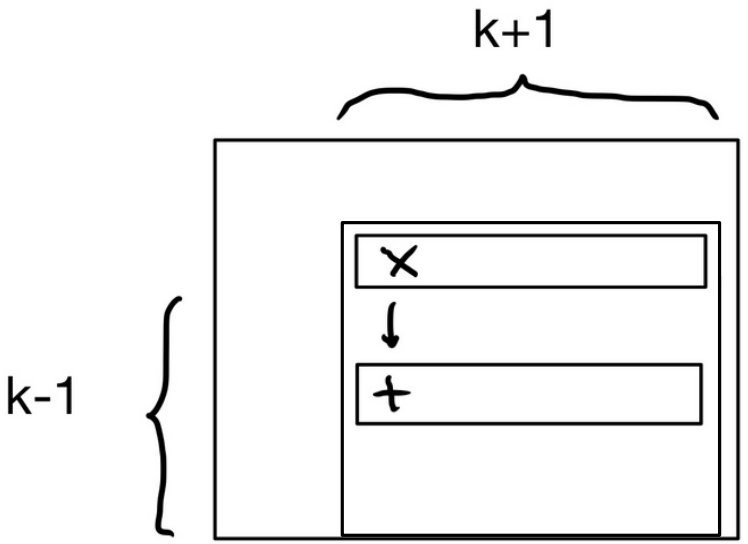
- For n equations in n unknowns, the augmented matrix is a nx(n+1) matrix.
- For each row *i*
  - Add a multiple of i^{th} row to all rows below it

New Eq (2) = Old Eq (2) – (a_{21}* Old Eq (1) / a_{11})

- For row 1, this process will require
  - There are (n-1) rows below row 1, each of those has (n+1) elements
  - Each element requires 1 multiplication, and 1 subtraction
  - Total calculations = 2 (n-1) (n+1) = $2n^2-2$

- When operating on i^{th} row, there are k = n-i+1 unknowns, so there are $2k^2-2$ calcs required





Forward elimination: Number of calculations

- k ranges from n down to 1
- So, the total number of arithmetic opns required

$$\sum_{k=1}^{n}(2k^2-2) = 2\left(\sum_{k=1}^{n}k^2 - \sum_{k=1}^{n}1\right)$$

$$= 2\left(\frac{n(n+1)(2n+1)}{6} - n\right)$$

$$= \frac{2}{3}n^3 + n^2 - \frac{5}{3}n$$

Back-substitution: generalization

- Consider the system as Lx=b

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\det(L) = l_{11} * l_{22} * \cdots l_{nn} = \prod_{i=1}^{n} l_{nn}$$

For a non-singular L, all l_{ii} should be non-zero

Back-substitution: generalization

$$l_{11}x_1 = b_1 \rightarrow x_1 = \frac{b_1}{l_{11}}$$

$$l_{21}x_1 + l_{22}x_2 = b_2 \rightarrow x_2 = \frac{b_2 - l_{21}x_1}{l_{22}}$$

And so on

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$l_{n1}x_1 + l_{n2}x_2 + \cdots + l_{nn}x_n = bn \rightarrow$$

$$x_n = \frac{b_n - l_{n1}x_1 - l_{n2}x_2 - \cdots . - l_{n,n-1}x_{n-1}}{l_{nn}}$$

In general, $$x_k = \frac{b_k - l_{k1}x_1 - l_{k2}x_2 - \cdots . - l_{k,k-1}x_{k-1}}{l_{kk}}$$

Back-substitution: No. of calculations

$$x_1 = \frac{b_1}{l_{11}} \qquad x_2 = \frac{b_2 - l_{21}x_1}{l_{22}} \qquad x_k = \frac{b_k - lk_1x_1 - l_{k2}x_2 - \cdots . - l_{k,k-1}x_{k-1}}{l_{kk}}$$

To get x_1 → one division

To get x_2 → one division + one subtraction + one multiplication

To get x_3 → one division + two subtractions + two multiplications

….

To get x_n → one division + (n-1) subtractions + (n-1) multiplications

Total calculations to get $x_1, x_2, \ldots, x_n$ →

n divisions + (1+2+…+(n-1)) subtractions + (1+2+…+(n-1) multiplications

i.e., approximately n^2 calculations

Gauss elimination: with matrix format

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

$$\left[\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

Augmented matrix

$$\left[\begin{array}{ccc:c} 1 & 1 & 2 & 3 \\ 2 & 3 & 1 & 2 \\ 3 & -1 & -1 & 6 \end{array}\right] \rightarrow \left[\begin{array}{ccc:c} 1 & 1 & 2 & 3 \\ 0 & 1 & -3 & -4 \\ 0 & -4 & -7 & -3 \end{array}\right] \rightarrow \left[\begin{array}{ccc:c} 1 & 1 & 2 & 3 \\ 0 & 1 & -3 & -4 \\ 0 & 0 & -19 & -19 \end{array}\right]$$

$$\rightarrow \quad x = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

Example: linearly dependent system

$$\begin{bmatrix} 1 & 2 & -1 & -2 & -1 \\ 2 & 1 & 1 & -1 & 4 \\ 1 & -1 & 2 & 1 & 5 \\ 1 & 3 & -2 & -3 & -3 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & -1 & -2 & -1 \\ 0 & -3 & 3 & 3 & 6 \\ 0 & -3 & 3 & 3 & 6 \\ 0 & 1 & -1 & -1 & -2 \end{bmatrix}$$

The **rank** of a matrix is defined as (a) the maximum number of linearly independent *row* vectors in the matrix or (b) the maximum number of linearly independent *column* vectors in the matrix.

Rank of augment matrix = rank of coefficient matrix = 2

Number of unknowns = 4

MATLAB Program

```
clear all
clc
A=[1 1 2; 2 3 1; 3 -1 -1];
B=[3; 2; 6];
C=[A,B];
[m,n]=size(C);
for i=1:m-1
    for j=i+1:m
        C(j,:)= C(j,:)-(C(j,i)* C(i,:)/ C(i,i));
    end
end

C
```

```
x=zeros(1,m);
for i=m:-1:1
    k=0;
    for j=2:m
        k=k+c(i,j)*x(j);
    end
    x(i)=(c(i,n)-k)/c(i,i);
end
```

Example: pivot element is zero

$$A=\begin{bmatrix}0 & 1 & 1\\ 1 & 1 & 2\\ 4 & 5 & 6\end{bmatrix}$$

a_{11} = 0, cannot be used for pivoting

Perform row interchange. Interchange rows (1) and (2)

$$A=\begin{bmatrix}1 & 1 & 2\\ 0 & 1 & 1\\ 4 & 5 & 6\end{bmatrix}\Rightarrow\begin{bmatrix}1 & 1 & 2\\ 0 & 1 & 1\\ 0 & 1 & -2\end{bmatrix}\Rightarrow\begin{bmatrix}1 & 1 & 2\\ 0 & 1 & 1\\ 0 & 0 & -3\end{bmatrix}$$

Coefficient Matrix is Upper Triangular

Rule: If pivot element = 0, then interchange that row with any of the later rows which will have a non-zero pivot element.

Example: pivot element is too small

- Sometimes, the candidate pivot element is quite small but not zero. Though, this does not lead to the "mathematical breakdown" of the algorithm, it does have a detrimental effect on the quality of the obtained solution when implemented on a finite precision computer.

$$\begin{bmatrix} 0.0001 & 0.5 \\ 0.4 & -0.3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix}$$

Our PC does 4-digit rounded decimal arithmetic

$$\left[\begin{array}{cc|c} 0.0001 & 0.5 & 0.5 \\ 0.4 & -0.3 & 0.1 \end{array}\right] \rightarrow \left[\begin{array}{cc|c} 1 & 5000 & 5000 \\ 0 & -2000 & -2000 \end{array}\right]$$

Rounded value of -2000.3

Rounded value of -1999.9

$x_1 = 0$
$x_2 = 1$ Wrong Solution

Rule: Avoid not only "zero" for the pivot element but also small pivots.

Pivoting

- Partial Pivoting: Make the element that is largest in magnitude from the remaining rows as the pivot element
- Threshold Pivoting: Perform interchange of rows only if the natural pivot element is significantly smaller than the competitor
- Complete Pivoting: Largest element in the entire remaining matrix is made the pivot element. This strategy therefore involves interchange of both rows and columns.

Example 1

- A civil engineer involved in construction requires 4800, 5800, and 5700 m^3 of sand, fine gravel, and coarse gravel, respectively, for a building project. There are three pits from which these materials can be obtained. The composition of these pits is

	Sand %	Fine Gravel %	Coarse Gravel %
Pit1	52	30	18
Pit2	20	50	30
Pit3	25	20	55

- How many cubic meters must be hauled from each pit in order to meet the engineer's needs?

Example 2

- An electrical engineer supervises the production of three types of electrical components. Three kinds of material. metal, plastic, and rubber.are required for production. The amounts needed to produce each component are

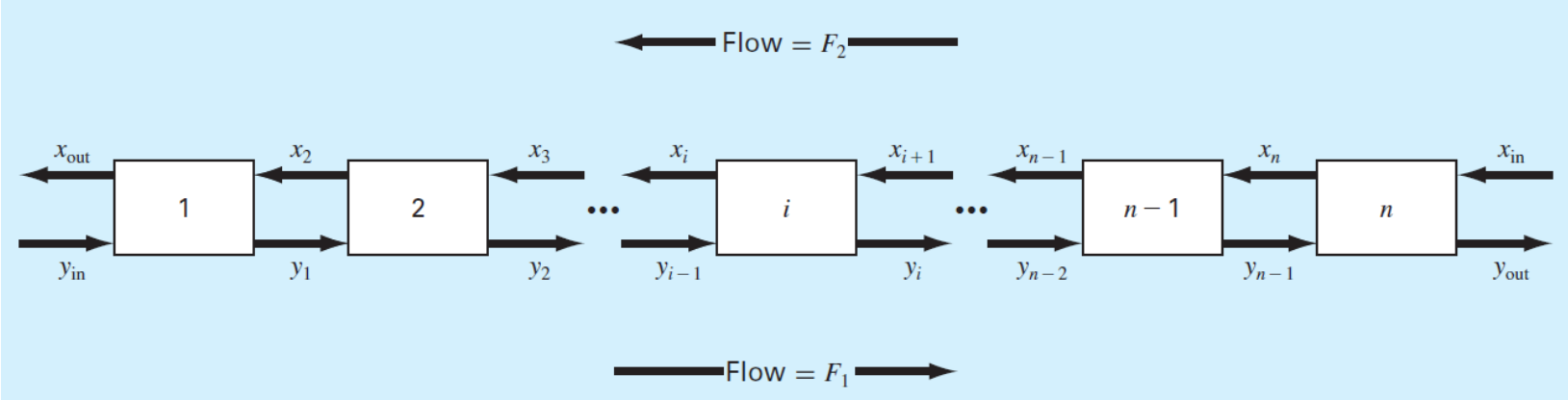
Component	Metal (g/ component)	Plastic (g/ component)	Rubber (g/ component)
1	15	0.25	1.0
2	17	0.33	1.2
3	19	0.42	1.6

- If totals of 2.12, 0.0434, and 0.164 kg of metal, plastic, and rubber, respectively, are available each day, how many components can be produced per day?

Example 3

A stage extraction process is depicted in Figure below. In such systems, a stream containing a weight fraction y_{in} of a chemical enters from the left at a mass flow rate of F_1. Simultaneously, a solvent carrying a weight fraction x_{in} of the same chemical enters from the right at a flow rate of F_2. Thus, for stage i, a mass balance can be represented as $F_1 y_{i-1} + F_2 x_{i+1} = F_1 y_i + F_2 x_i$

At each stage, an equilibrium is assumed to be established between y_i and x_i as $y_i = K x_i$. Typically, K is obtained experimentally and it is assumed to be known here. If $F_1 = 400$ kg/h, $y_{in} = 0.1$, $F_2 = 800$ kg/h, $x_{in} = 0$, and $K = 5$, determine the values of y_{out} and x_{out} if a five-stage reactor is used. Note that mass balance must be modified to account for the inflow weight fractions when applied to the first and last stages.



CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

Gauss-Jordon method

- Key idea: Converting the coefficient matrix to the identity matrix

Example:

$$x_1+2x_2-x_3-2x_4 = -1$$
$$2x_1+x_2+x_3-x_4 = 4$$
$$10x_1-5x_2+2x_3+ x_4 = 5$$
$$-x_1+5x_2-2x_3-3x_4 = -3$$

Example

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & -2 & -1 \\ 2 & 1 & 1 & -1 & 4 \\ 10 & -5 & 2 & 1 & 5 \\ -1 & 5 & -2 & -3 & -3 \end{array}\right] \rightarrow \left[\begin{array}{cccc|c} 1 & 2 & -1 & -2 & -1 \\ 0 & -3 & 3 & 3 & 6 \\ 0 & -25 & 12 & 21 & 15 \\ 0 & 7 & -3 & -5 & -4 \end{array}\right]$$

$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & -2 & -1 \\ 0 & 1 & -1 & -1 & -2 \\ 0 & -25 & 12 & 21 & 15 \\ 0 & 7 & -3 & -5 & -4 \end{array}\right] \rightarrow \left[\begin{array}{cccc|c} 1 & 0 & -1 & 0 & -1 \\ 0 & 1 & -1 & -1 & -2 \\ 0 & 0 & -13 & -4 & -35 \\ 0 & 0 & 4 & 2 & 10 \end{array}\right]$$

Example

$$\left[\begin{array}{cccc|c} 1 & 0 & -1 & 0 & -1 \\ 0 & 1 & -1 & -1 & -2 \\ 0 & 0 & 1 & 4/13 & 35/13 \\ 0 & 0 & 4 & 2 & 10 \end{array}\right] \rightarrow \left[\begin{array}{cccc|c} 1 & 0 & 0 & -4/13 & 4/13 \\ 0 & 1 & 0 & -9/13 & 9/13 \\ 0 & 0 & 1 & 4/13 & 35/13 \\ 0 & 0 & 0 & 10/13 & -10/13 \end{array}\right]$$

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & -4/13 & 4/13 \\ 0 & 1 & 0 & -9/13 & 9/13 \\ 0 & 0 & 1 & 4/13 & 35/13 \\ 0 & 0 & 0 & 1 & -1 \end{array}\right] \rightarrow \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & -1 \end{array}\right] \Rightarrow x = \begin{bmatrix} 0 \\ 0 \\ 3 \\ -1 \end{bmatrix}$$

Tridiagonal systems

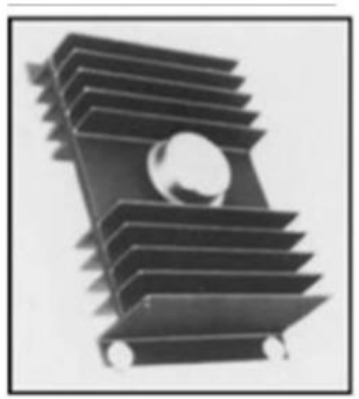
$$
\begin{aligned}
2x_1 - x_2 &= -1 \\
-x_1 + 2x_2 - x_3 &= 0 \\
-x_2 + 2x_3 - x_4 &= 0 \\
-x_3 + 2x_4 &= 0
\end{aligned}
$$

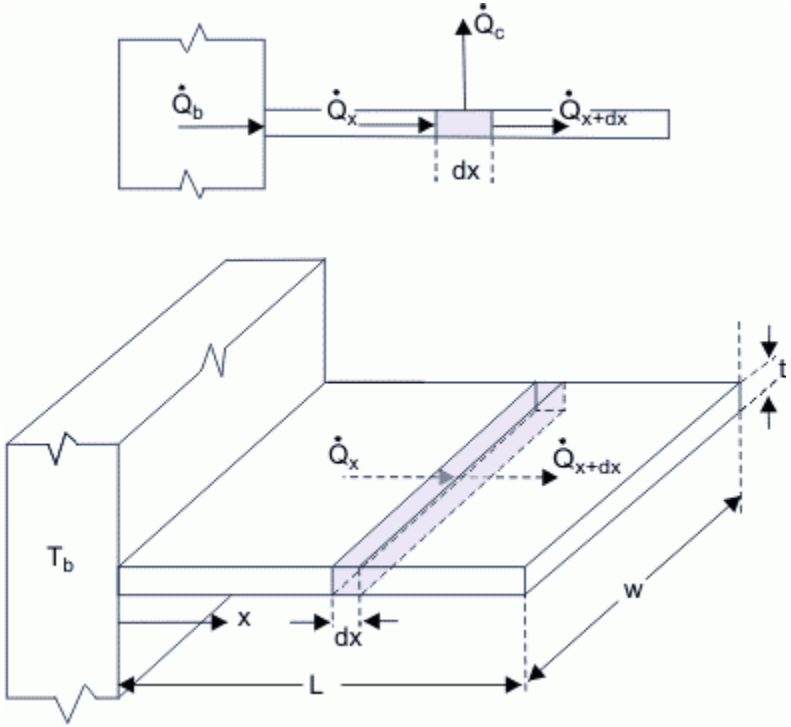
Coefficient matrix

$$
A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}
$$

Example

- Heat transfer through a fin
- https://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node128.html





Example: fin

- Equation governing the heat transfer in the fin

$$\frac{d^2\theta}{d\xi^2} - \frac{hPL^2}{kA}\theta = 0$$

- Boundary conditions:

$$\theta(\xi = 0) = 1$$
$$\theta(\xi = 1) = 0$$

$$\theta = \frac{T - T_\infty}{T_0 - T_\infty}$$

- Let $\frac{hPL^2}{kA} = 1$. This gives

$$\frac{d^2\theta}{d\xi^2} - \theta = 0$$

Example: fin

- Finite-difference approximations provide a means to transform derivatives into algebraic form

Forward Difference Formula (1st order accurate)

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Backward Difference Formula (1st order accurate)

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

Centered Difference Formula (2nd order accurate)

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

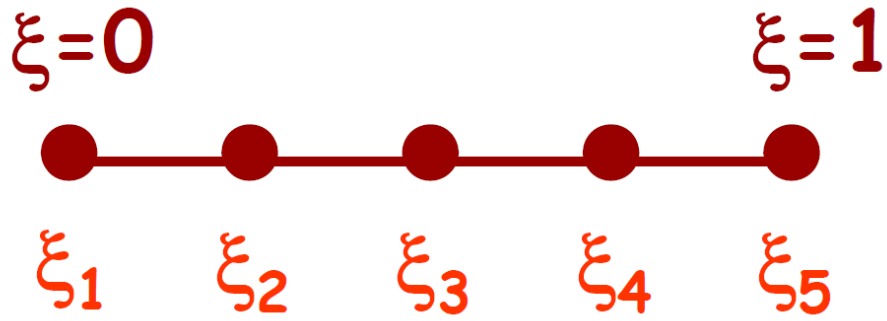
Centered Difference Formula for 2nd Derivative (2nd order accurate)

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Example: fin

$$\frac{d^2\theta}{d\xi^2} - \theta = 0$$

- Divide the fin into 5 equispaced nodes
- 5 grid points; 3 internal grid points; 2 boundary grid points
- For internal grid points



$$\frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{\Delta\xi^2} - \theta_i = 0$$

$$\Delta\xi = 1/4$$

$$\theta_{i+1} - 2\theta_i + \theta_{i-1} - 0.0625\theta_i = 0$$

We know that $\theta_1 = 1, \theta_5 = 0$

$$\theta_{i-1} - 2.0625\theta_i + \theta_{i+1} = 0$$

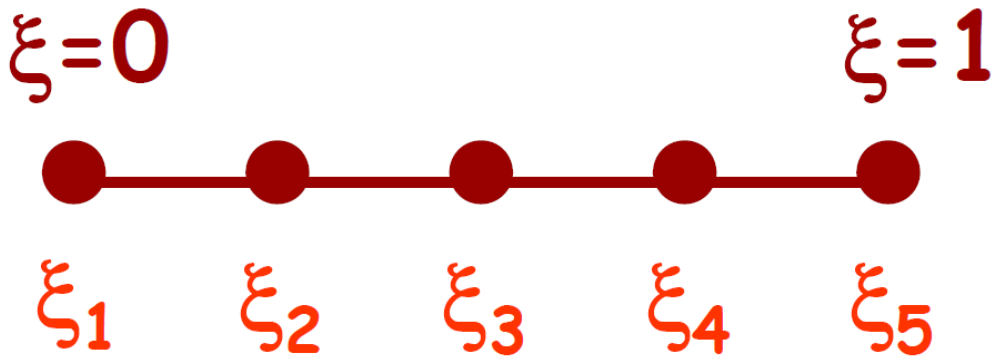
Example: fin

$$\theta_{i-1} - 2.0625\theta_i + \theta_{i+1} = 0$$

$\theta_1 - 2.0625\theta_2 + \theta_3 = 0$ (for node 2)

$\theta_2 - 2.0625\theta_3 + \theta_4 = 0$ (for node 3)

$\theta_3 - 2.0625\theta_4 + \theta_5 = 0$ (for node 4)



$\Delta\xi = 1/4$

We know that $\theta_1 = 1, \theta_5 = 0$

$-2.0625\theta_2 + \theta_3 = -1$

$\theta_2 - 2.0625\theta_3 + \theta_4 = 0$

$\theta_3 - 2.0625\theta_4 = 0$

Example: fin

$$\begin{bmatrix} -2.0625 & 1 & 0 \\ 1 & -2.0625 & 1 \\ 0 & 1 & -2.0625 \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

A tridiagonal structure

Solving tridiagonal systems

$$b_1x_1+c_1x_2 = d_1$$
$$a_2x_1+ b_2x_2 +c_2x_3 = d_2$$
$$a_3x_2+ b_3x_3 +c_3x_4 = d_3$$

.....

$$a_{n-1}x_{n-2}+ b_{n-1}x_{n-1} +c_{n-1}x_n = d_{n-1}$$
$$a_nx_{n-1}+b_nx_n = d_n$$

Sparse systems

Solving tridiagonal systems

- Coefficient of x_1 is nonzero only in Eq. 2
- If x_1 is eliminated from eq. 2, eq. 2 will look like

$$b'_2 x_2 + c'_2 x_3 = d'_2$$

- Same structure as Eq. 1, only two unknowns

- Thus, at elimination stage, we will work with only two equations (of this form)

$$x_k + h_k x_{k+1} = p_k \quad \text{(Eq. A)}$$

$$a_{k+1} x_k + b_{k+1} x_{k+1} + c_{k+1} x_{k+2} = d_{k+1} \quad \text{(Eq. B)}$$

- After eliminating x_k from Eq. B, it will look like

$$x_{k+1} + h_{k+1} x_{k+2} = p_{k+1}$$

Solving tridiagonal systems

- Can we represent h_{k+1} and p_{k+1} in terms of b_{k+1}, d_{k+1}, c_{k+1}, a_{k+1}, h_k and p_k?

- Multiply Eq. *A* by a_{k+1} and subtract it from Eq. B

$$(b_{k+1}-a_{k+1}h_k)x_{k+1} + c_{k+1}x_{k+2} = d_{k+1} - a_{k+1}p_k$$

$$x_{k+1} + \frac{c_{k+1}}{b_{k+1} - a_{k+1}\, h_k}\, x_{k+2} = \frac{d_{k+1} - a_{k+1}\, p_k}{b_{k+1} - a_{k+1}\, h_k}$$

Thus, $h_{k+1} = \dfrac{c_{k+1}}{b_{k+1} - a_{k+1}\, h_k}$ **&** $p_{k+1} = \dfrac{d_{k+1} - a_{k+1}\, p_k}{b_{k+1} - a_{k+1}\, h_k}$

Solving tridiagonal systems

- The system will now look like

$$x_k + h_k x_{k+1} = p_k \quad (\text{for } k = 1,2,3,\ldots,n-1)$$

$$x_n = p_n \quad (\text{for } k = n)$$

- We can now work backwards to get

$$x_k = p_k - h_k x_{k+1} \quad (\text{for } k = n-1,\ldots,3,2,1)$$

Note that x_{k+1} is already known so that all RHS terms are all available during the computation of x_k from the above equation

CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

Interpretation

- What will be f(3.5)?

x	$f(x)$
2	8
3	27
4	64
5	125
6	216

Difference operators

- Let us say we have a data set

$$\{(f_j, x_j) | \quad j = 1, 2, \ldots, m\}$$

- Data is equally spaced

$$x_{i+1} = x_i + h$$
$$where\ i = 1, 2, \ldots, m-1$$

- h: step size, discretization interval

- $\Delta f_i = f_{i+1} - f_i$ Forward difference operator
- $\nabla f_i = f_i - f_{i-1}$ Backward difference operator
- $E f_i = f_{i+1}$ Shift operator

Difference operators: higher order

- $\Delta^2 f_i = \Delta(\Delta f_i) = \Delta(f_{i+1} - f_i) = \Delta(f_{i+1}) - \Delta(f_i)$

$$= (f_{i+2} - f_{i+1}) - (f_{i+1} - f_i)$$

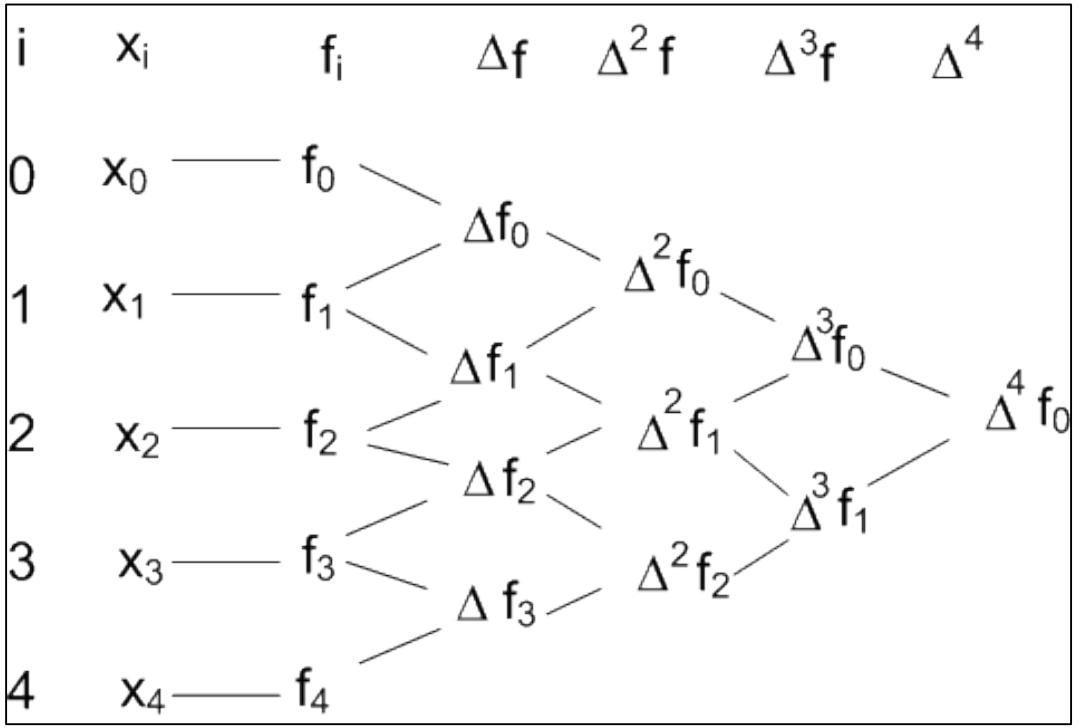
$$= f_{i+2} - 2f_{i+1} + f_i$$

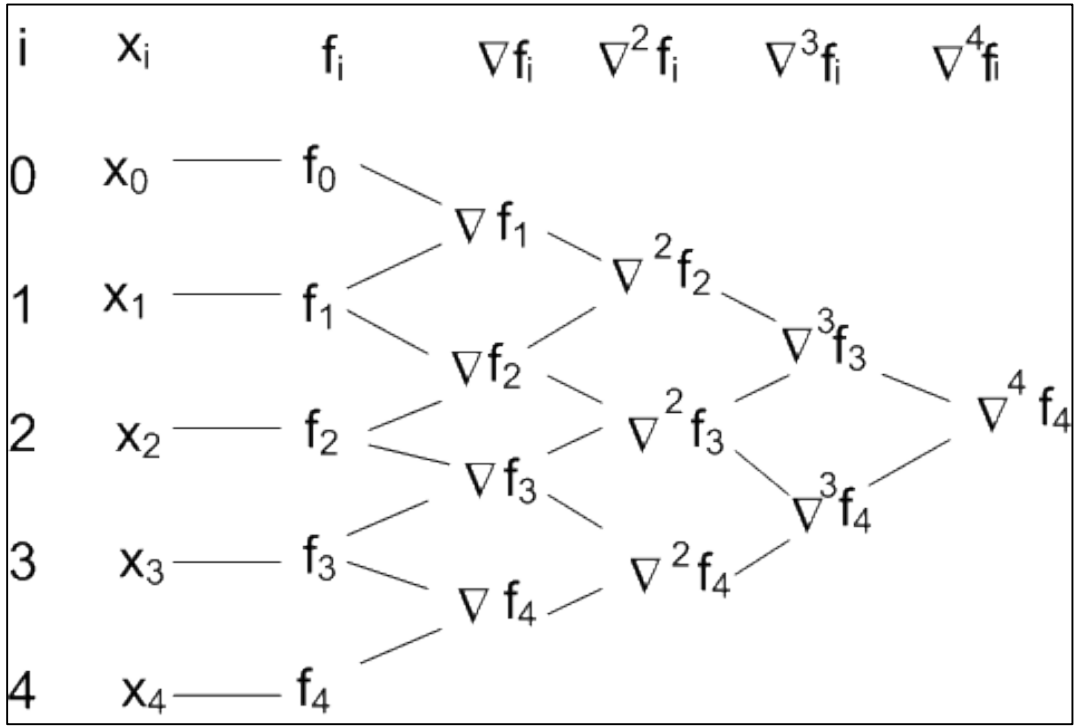
- $\Delta^3 f_i = \Delta^2(\Delta f_i) = \Delta^2(f_{i+1} - f_i) = \Delta^2(f_{i+1}) - \Delta^2(f_i)$

$$= (f_{i+3} - 2f_{i+2} + f_{i+1}) - (f_{i+2} - 2f_{i+1} + f_i)$$

$$= f_{i+3} - 3f_{i+2} + 3f_{i+1} - f_i$$

Difference tables





Shift operator

- $Ef_i = f_{i+1}$
- $E^2 f_i = f_{i+2}$

what are f_i and f_{i+1}?

$$Ef(x_i) = f(x_i + h)$$

- We know that $\Delta f_i = f_{i+1} - f_i$
- This can be rewritten as $\Delta f_i = Ef_i - f_i = (E-1)f_i$

$$\Delta = (E-1)$$

Or $E = (1 + \Delta)$

- Generalizing this, $E^\alpha = (1+\Delta)^\alpha$

- Exercise: Prove that $E^2 = (1+\Delta)^2$

Polynomial approximation of dataset

- Our data set

$$\{(f_j, x_j) | \quad j = 1, 2, \ldots, m\}$$

This can be approximated as

$$f(x) = P_n(x) + R(x)$$

Where $P_n(x)$ is the polynomial approximation of f(x) and R(x) is the residual error.

- What will be f(3.5)?

x	*f(x)*
2	8
3	27
4	64
5	125
6	216

- In other words, we need to find $f(3+0.5)$ or $f(2+1.5)$ or $f(4-0.5)$.

Or $f(x_i + \alpha h)$

- $f(x_i + \alpha h) = E^{\alpha} f(x_i) = (1+\Delta)^{\alpha} f(x_i)$

- Using binomial theorem, we can write

$$f(x_i + \alpha h) = \left[1 + \alpha\Delta + \frac{\alpha(\alpha-1)}{2!}\Delta^2 + \frac{\alpha(\alpha-1)(\alpha-2)}{3!}\Delta^3 + \cdots\right] f(x_i)$$

- Can choose α to be 0, 1, …, m-1 and indeed other non-integer real values to reach any x value in $[x_1, x_m]$

Numerical differentiation

- We need to estimate $\frac{df}{dx}$
- We have already written

$$f(x) = f(x_i + \alpha h) = \left[1 + \alpha\Delta + \frac{\alpha(\alpha-1)}{2!}\Delta^2 + \frac{\alpha(\alpha-1)(\alpha-2)}{3!}\Delta^3 + \cdots\right] f(x_i)$$

- $\frac{df}{dx} = \frac{df}{d\alpha} * \frac{d\alpha}{dx} = \frac{df}{d\alpha} * \frac{1}{h}$ Since $x = x_i + \alpha h \rightarrow \frac{d\alpha}{dx} = \frac{1}{h}$
- Thus we can write

$$f'(x) \approx \frac{1}{h}\left[\Delta + \frac{\alpha+(\alpha-1)}{2}\Delta^2 + \frac{\{\alpha(\alpha-1)+(\alpha-1)(\alpha-2)+\alpha(\alpha-2)\}}{6}\Delta^3 + \cdots\right] f(x_i)$$

- For x_1, $\alpha = 0$

$$f'(x_1) \approx \frac{1}{h}\left[\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 + \cdots \pm \frac{1}{m-1}\Delta^{m-1}\right] f(x_1) + O(h^{m-1})$$

- Consider only the first term

$$f'(x_1) \approx \frac{1}{h}\Delta f(x_1) + O(h)$$
$$\approx \frac{1}{h}\{f_2 - f_1\} + O(h)$$

Two-points (2P)
First order accurate (FOA)
Forward difference (FDA)

Generalized form: $f'(x_i) \approx \frac{1}{h}\{f_{i+1} - f_i\} + O(h)$

- Consider the first two terms

$$f'(x_1) \approx \frac{1}{h}\left[\Delta f(x_1) - \frac{1}{2}\Delta^2 f(x_1)\right] + O(h^2)$$
$$\approx \frac{1}{h}\left[(f_2 - f_1) - \frac{1}{2}(f_1 - 2f_2 + f_3)\right] + O(h^2)$$
$$\approx \frac{1}{2h}[-3f_1 + 4f_2 - f_3] + O(h^2)$$

3P, SOA, FDA

Generalized form: $f'(x_i) \approx \frac{1}{2h}[-3f_i + 4f_{i+1} - f_{i+2}] + O(h^2)$

- For x_2, $\alpha = 1$

$$f'(x_2) \approx \frac{1}{h}\left[\Delta + \frac{1}{2}\Delta^2 - \frac{1}{6}\Delta^3 + \cdots\right] f(x_1) + O(h^{m-1})$$

- Consider the first two terms

$$f'(x_2) \approx \frac{1}{h}\left[\Delta f(x_1) + \frac{1}{2}\Delta^2 f(x_1)\right] + O(h^2)$$

$$\approx \frac{1}{h}\left[(f_2 - f_1) + \frac{1}{2}(f_1 - 2f_2 + f_3)\right] + O(h^2)$$

$$\approx \frac{1}{2h}[f_3 - f_1] + O(h^2)$$

2P, SOA, CDA

- Generalizing this, we can write another formula for $f'(x_i)$

$$f'(x_i) \approx \frac{1}{2h}\{f_{i+1} - f_{i-1}\} + O(h^2)$$

First Derivative		
Method	**Formula**	**Truncation Error**
Two-point forward difference	$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
Three-point forward difference	$f'(x_i) = \frac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2})}{2h}$	$O(h^2)$
Two-point central difference	$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$
Two-point backward difference	$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
Three-point backward difference	$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h}$	$O(h^2)$

- Second order derivatives
- We have already written

$$f(x) = f(x_i + \alpha h) = \left[1 + \alpha\Delta + \frac{\alpha(\alpha-1)}{2!}\Delta^2 + \frac{\alpha(\alpha-1)(\alpha-2)}{3!}\Delta^3 + \cdots\right] f(x_i)$$

- $\frac{d^2 f}{dx^2} = \frac{d}{dx}\left[\frac{df}{d\alpha} * \frac{d\alpha}{dx}\right] = \frac{d}{d\alpha}\left[\frac{df}{d\alpha} * \frac{d\alpha}{dx}\right]\frac{d\alpha}{dx} = \frac{1}{h^2}\frac{d^2 f}{d\alpha^2}$

- Thus we can write

$$f''(x) \approx \frac{1}{h^2}\left[\Delta^2 + \frac{\{\alpha+(\alpha-1)+(\alpha-1)+(\alpha-2)+\alpha+(\alpha-2)\}}{6}\Delta^3 + \cdots\right] f(x_i)$$

- For x_1, $\alpha = 0$

$$f''(x_1) \approx \frac{1}{h^2}\left[\Delta^2 - \Delta^3 + \frac{11}{12}\Delta^4 + \cdots\right] f(x_1)$$

- Consider only the first term

$$f''(x_1) \approx \frac{1}{h^2}[\Delta^2 f(x_1)] + O(h)$$

$$\approx \frac{1}{h^2}\{f_1 - 2f_2 + f_3\} + O(h)$$

3P, FOA, FDA

Generalized form: $f''(x_i) \approx \frac{1}{h^2}\{f_i - 2f_{i+1} + f_{i+2}\} + O(h)$

- Consider the first two terms

$$f''(x_1) \approx \frac{1}{h^2}[\Delta^2 f(x_1) - \Delta^3 f(x_1)] + O(h^2)$$

$$\approx \frac{1}{h^2}\{2f_1 - 5f_2 + 4f_3 - f_4\} + O(h^2)$$

4P, SOA, FDA

Generalized form: $f''(x_i) \approx \frac{1}{h^2}\{2f_i - 5f_{i+1} + 4f_{i+2} - f_{i+3}\} + O(h^2)$

- For x_2, $\alpha = 1$

$$f''(x_2) \approx \frac{1}{h^2}[\Delta^2 + 0\Delta^3 + \cdots]f(x_1)$$

- Consider the first two terms

$$f''(x_2) \approx \frac{1}{h^2}[\Delta^2 f(x_1)] + O(h^2)$$

$$\approx \frac{1}{h^2}\{f_1 - 2f_2 + f_3\} + O(h^2)$$

3P, SOA, CDA

- Generalizing this, we can write another formula for $f''(x_i)$

$$f''(x_i) \approx \frac{1}{h^2}\{f_{i-1} - 2f_i + f_{i+1}\} + O(h^2)$$

Second Derivative		
Method	**Formula**	**Truncation Error**
Three-point forward difference	$f''(x_i) = \frac{f(x_i) - 2f(x_{i+1}) + f(x_{i+2})}{h^2}$	$O(h)$
Four-point forward difference	$f''(x_i) = \frac{2f(x_i) - 5f(x_{i+1}) + 4f(x_{i+2}) - f(x_{i+3})}{h^2}$	$O(h^2)$
Three-point central difference	$f''(x_i) = \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1})}{h^2}$	$O(h^2)$
Three-point backward difference	$f''(x_i) = \frac{f(x_{i-2}) - 2f(x_{i-1}) + f(x_i)}{h^2}$	$O(h)$
Four-point backward difference	$f''(x_i) = \frac{-f(x_{i-3}) + 4f(x_{i-2}) - 5f(x_{i-1}) + 2f(x_i)}{h^2}$	$O(h^2)$

Example

- What is f'(3.5)?
  - Using 2P, FOA, FDA
  - Using 2P, FOA, BDA
  - Using 2P, SOA, CDA

- What is f''(3.5)?
  - Using 3P, SOA, CDA

x	f(x)
3.4	0.294118
3.5	0.285714
3.6	0.277778

Indian Institute of Technology Roorkee

CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

Acknowledgements: Prof. Laksh, NUS

Solving nonlinear algebraic equations

- Single equation

$$f(x) = 0$$
$$f(x, p) = 0$$

p = parameter

- Set of equations

$$f_1(x_1, x_2) = 0$$
$$f_2(x_1, x_2) = 0$$

- Generalized

$$\underline{f}(\underline{x}) = 0$$
$$\underline{f}\left(\underline{x}, \underline{p}\right) = 0$$

Single nonlinear equation

$$f(x) = 0$$
$$f(x, p) = 0$$

x = unknown variable, p = parameter

- Example: $ax^2 + bx + c = 0$

$$f(x) = ax^2 + bx + c$$

- Solving this equation means finding the roots
  - Values of x that make $f(x)$ equal to zero
  - Also called the zeros of $f(x)$
- The roots depends on the values of parameters (a, b and c)

Example 1

- Van der Waals equation of state

$$\left(P + \frac{a}{V^2}\right)(V - b) = RT \qquad a = \frac{27}{64}\left(\frac{R^2 T_c^2}{P_c}\right) \qquad b = \frac{RT_c}{8P_c}$$

- P (atm), V(L/gmol), R=0.08206 atm.L/(gmol. K)
- T_c=405.5 K, P_c=111.3 atm for ammonia

- Calculate the molar volume and compressibility factor for ammonia gas at a pressure of 56 atm and a temperature of 450 K.

Example 2

- Second law of motion,
  - rate of change of momentum of a body is equal to the resultant force acting on it

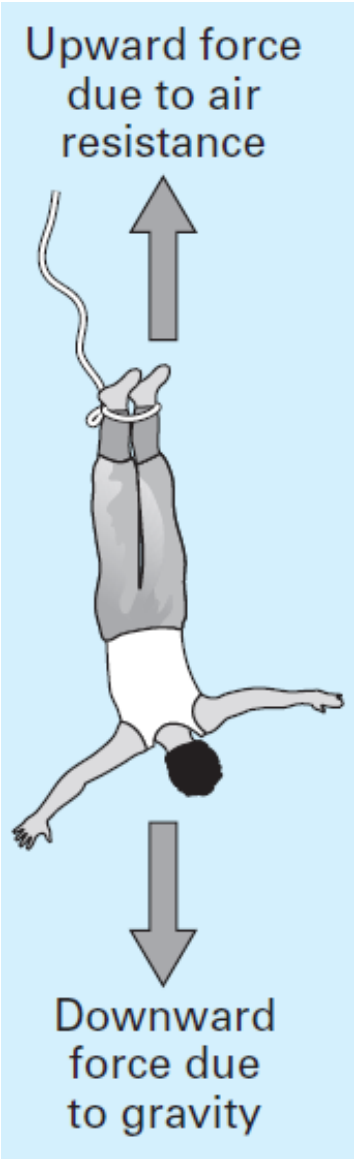
$$\frac{d(mv)}{dt} = F_{net}$$

- The net force on the body is

$$F_{net} = F_g - F_r = mg - c_d v^2$$

- Substituting, we get

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$



Example 2 continued…

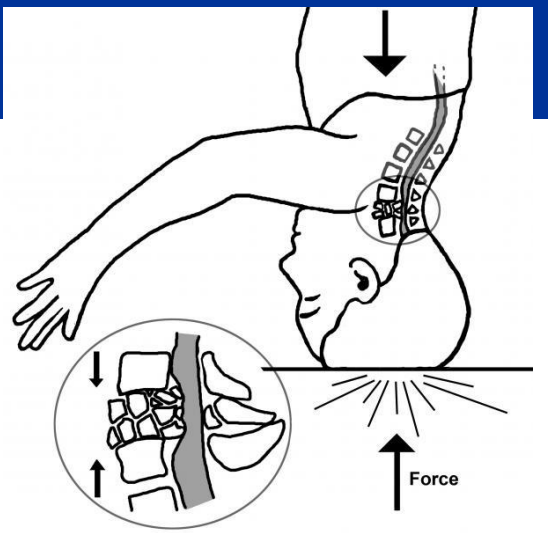
- Analytical solution

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}\, t\right)$$

- Model for the bungee jumper's velocity
- You can predict the jumper's velocity
- Such computations can be performed directly because *v* is expressed ***explicitly*** as a function of the model parameters. That is, it is isolated on one side of the equal sign.

Example 2 continued…

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}\, t\right)$$



- According to medical studies, bungee jumper's chances of sustaining a significant vertebrae injury increase significantly if the free fall velocity exceeds 36 m/s after 4 s of free fall
- Determine the critical mass at which this criterion is exceeded given a drag coefficient of 0.25 kg/m

Example 2 continued…

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}\, t\right)$$

- All unknowns, except m
- We have an equations (to determine m), but it cannot be solved explicitly for m
- m is said to be **implicit**
- To solve for m, we can write

$$f(m) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}\, t\right) - v(t) = 0$$

$$\sqrt{\frac{9.81m}{0.25}} \tanh\left(\sqrt{\frac{9.81(0.25)}{m}}\, 4\right) - 36 = 0$$

Single nonlinear equation

$$ax^2 + bx + c = 0$$

- Direct solution

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- There are many equations that could not be solved directly → Approximate solution techniques
  1. Graphical method
  2. Trial and error method
  3. Numerical methods: systematic strategies to arrive at the roots

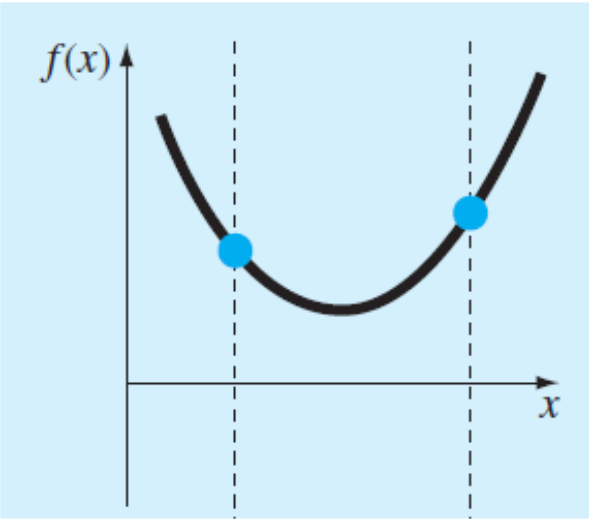
Graphical method

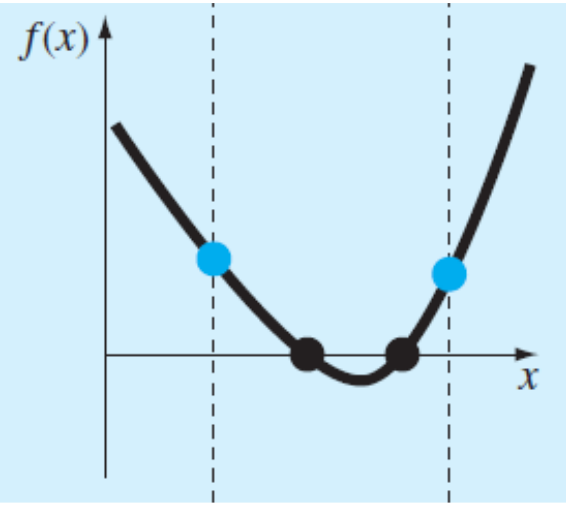
- Make a plot of the function
- Observe where it crosses the x axis
- This point, which represents the x value for which $f(x) = 0$, provides a rough approximation of the root

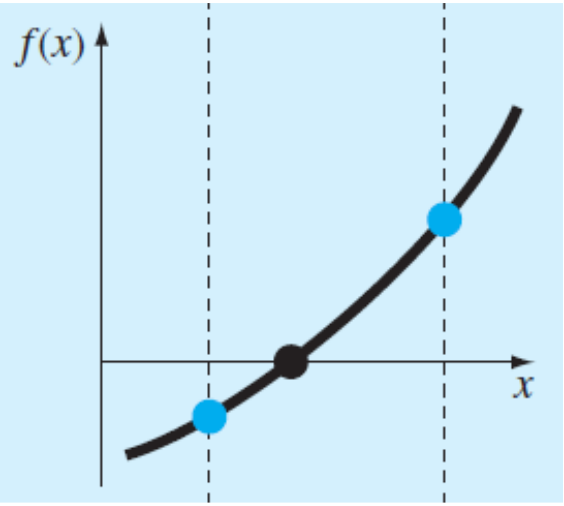
Solve example 2 by graphical method in MATLAB

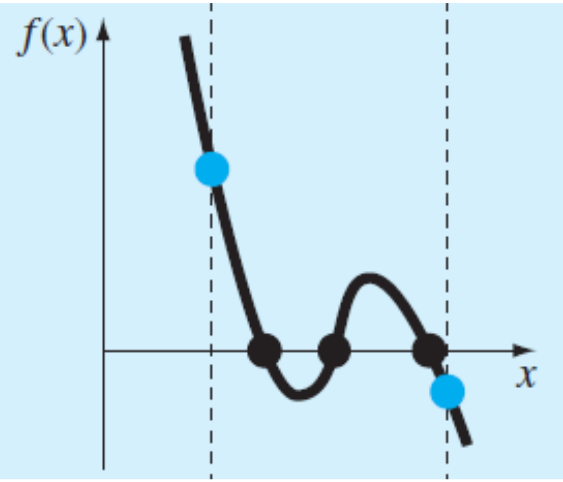
Graphical method

- Number of ways in which roots can occur (or be absent) in an interval prescribed by a lower and upper bound









Trial and error method

- Guess a value of x and evaluate whether $f(x)$ is zero
- If not, make another guess for x
- Again evaluate $f(x)$ to check if the new value of x provides a better estimate of the root
- Repeat until a guess results in an $f(x)$ that is close to zero.

Numerical methods

- Bracketing methods: Based on two initial guesses that bracket the root
  - Bisection method
  - False position method
- Open methods: Can involve one or more initial guesses, but there is no need for them to bracket the root.
  - Secant method
  - Newton's method
  - Muller's method
  - Fixed point iteration method

Bisection method

- Key idea: if $f(x)$ is continuous and it changes signs at two x-values, there must be at least one root between these x-values
- To determine a root of $f(x) = 0$ given values x_0 and x_1 such that $f(x_0) * f(x_1) < 0$

Repeat
  Set $x_2 = (x_0 + x_1)/2$
  If $f(x_2) * f(x_0) < 0$ Then
    Set $x_1 = x_2$
    Else Set $x_0 = x_2$
  End if
Until $(|x_0 - x_1|) < 2 * tolerance$

Example

- Solve in MATLAB by bisection method

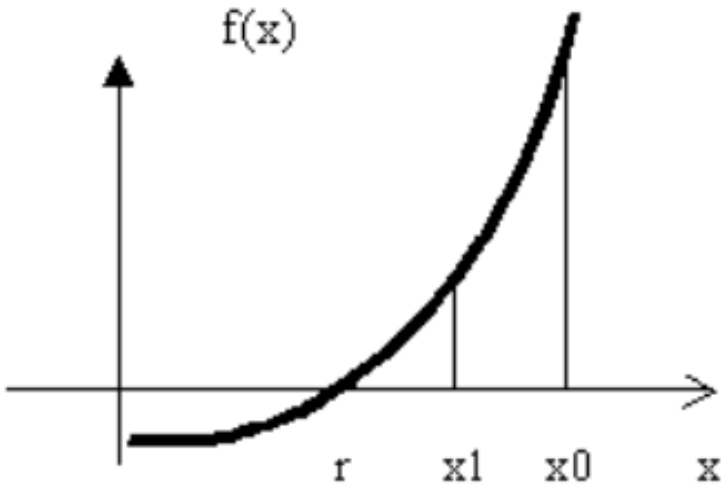
$$f(x) = 3x + \sin x - e^x = 0$$

$$x0 = 0 \text{ and } x1 = 1$$

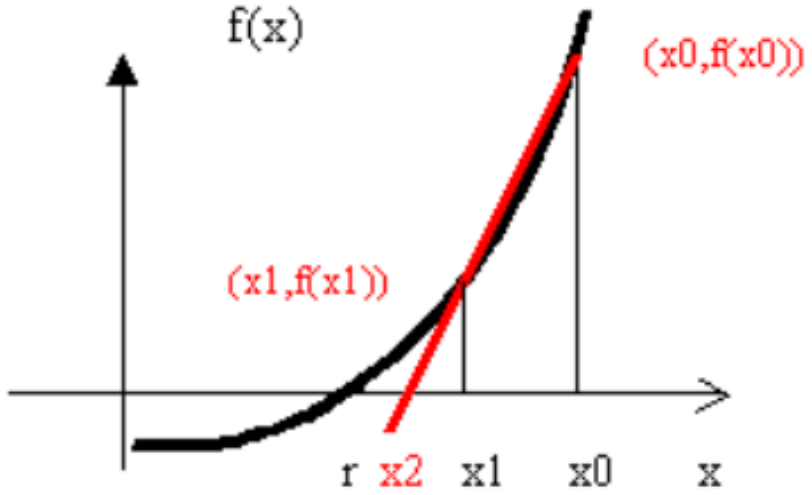
$$tolerance = 1 \times 10^{-4}$$

Secant method

- Key idea: approximate the curve with a straight line for x between the values of x_0 and r.



The straight line is assumed to be the secant which connects the two points (x_0, $f(x_0)$) and (x_1, $f(x_1)$)



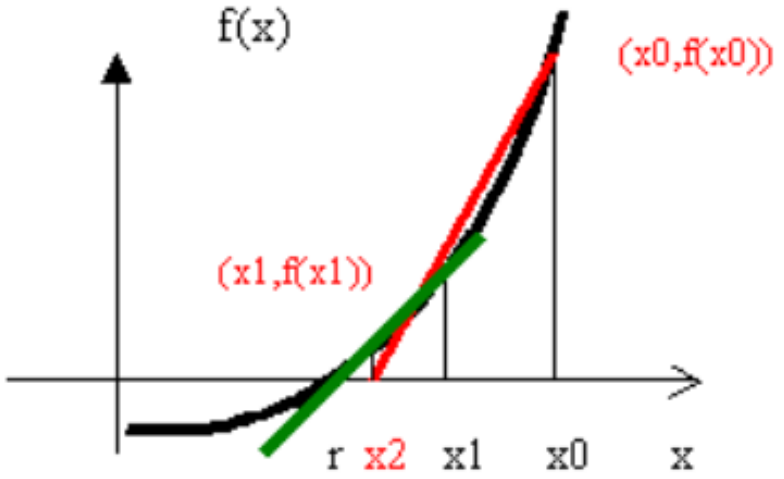
Both the secant line and the new root estimate x_2 are shown in red

x_2 is closer to the root r than either x_1 or x_0

https://www.lehigh.edu/~ineng2/clipper/notes/secant.htm

Secant method

- Repeat this process (green line)
- Getting closer to r



Secant method

- Slope of red line $m = \frac{f(x_1)-f(x_0)}{x_1-x_0}$
- Equation of red line considering one point as $(x_1, f(x_1))$

$$y - f(x_1) = m(x - x_1)$$

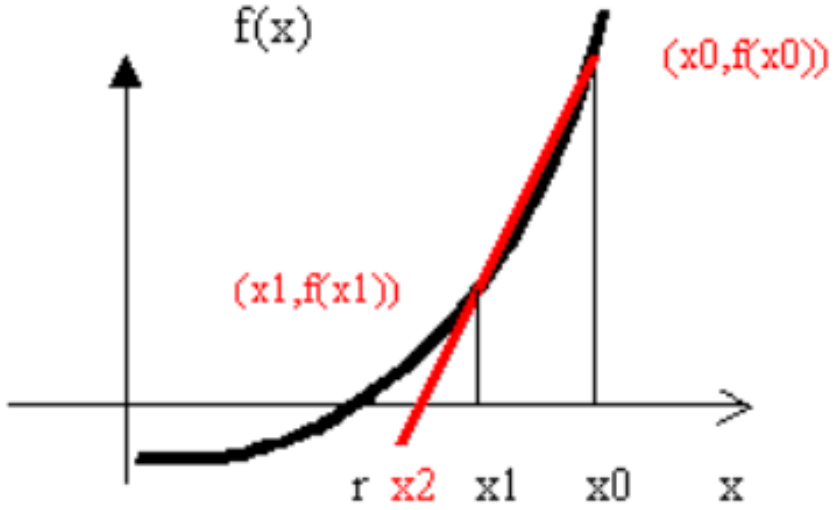
- The point $x = x_2$ corresponds to the point of the straight line where $y = 0$

$$-f(x_1) = m(x_2 - x_1)$$

- Thus

$$x_2 = x_1 - \frac{f(x_1)}{m}$$

$$x_2 = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$$



Secant method

- Generalizing, we get

$$x_{i+1} = x_i - f(x_i)\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}$$

- To determine a root of $f(x) = 0$, given two values, x_0 and x_1, that are near the root,

If $|f(x_0)| < |f(x_1)|$ Then
  Swap x_0 with x_1
Repeat
  Set $x_2 = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$
  Set $x_0 = x_1$
  Set $x_1 = x_2$
Until $|f(x_2)| < tolerance$

Example

- Solve in MATLAB by secant method

$$f(x) = 3x + \sin x - e^x = 0$$

$$x_0 = 0 \text{ and } x_1 = 1$$

$$tolerance = 1 \times 10^{-4}$$

False position (linear interpolation method)

- Key idea: mix of bisection and secant method.

Repeat

Set $x_2 = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$

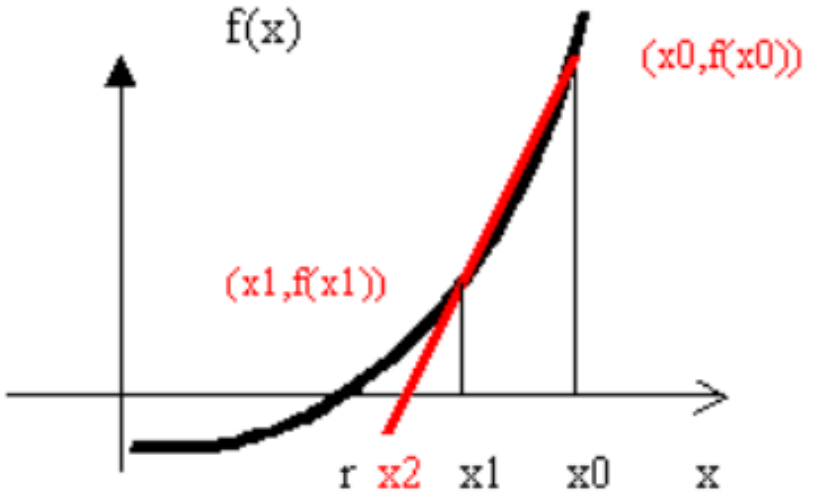
If $f(x_2) * f(x_0) < 0$ Then

Set $x_1 = x_2$

Else Set $x_0 = x_2$

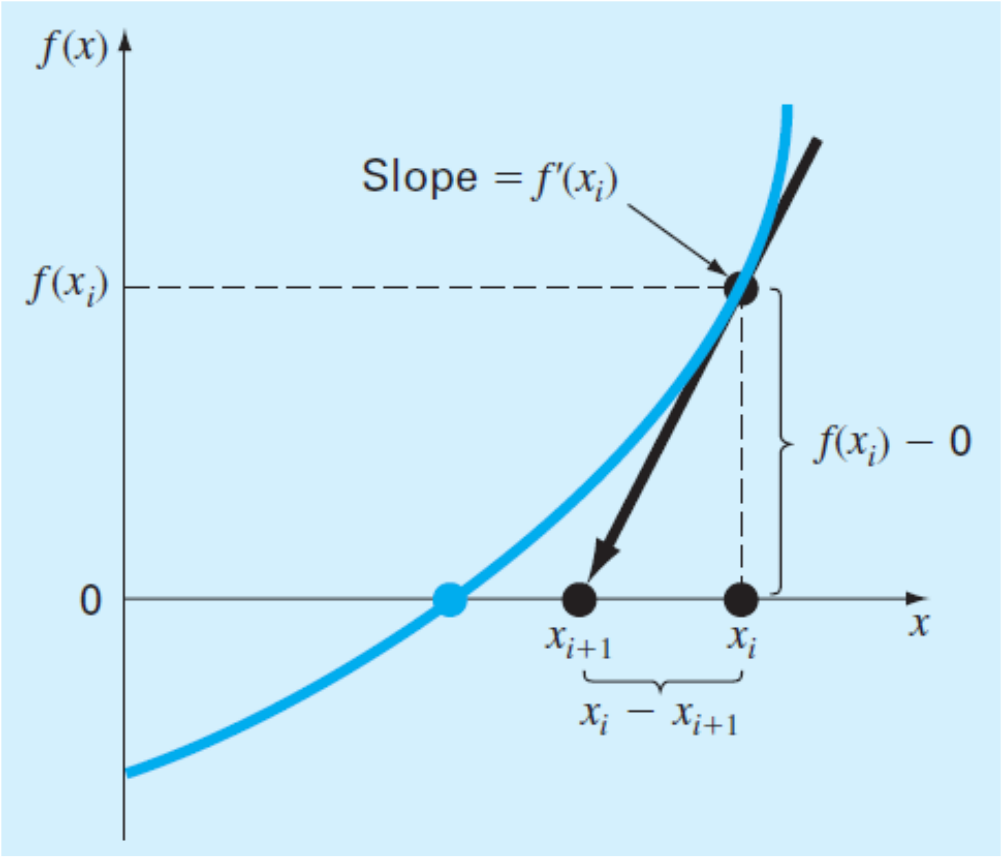
End if

Until $|f(x_2)| < tolerance$

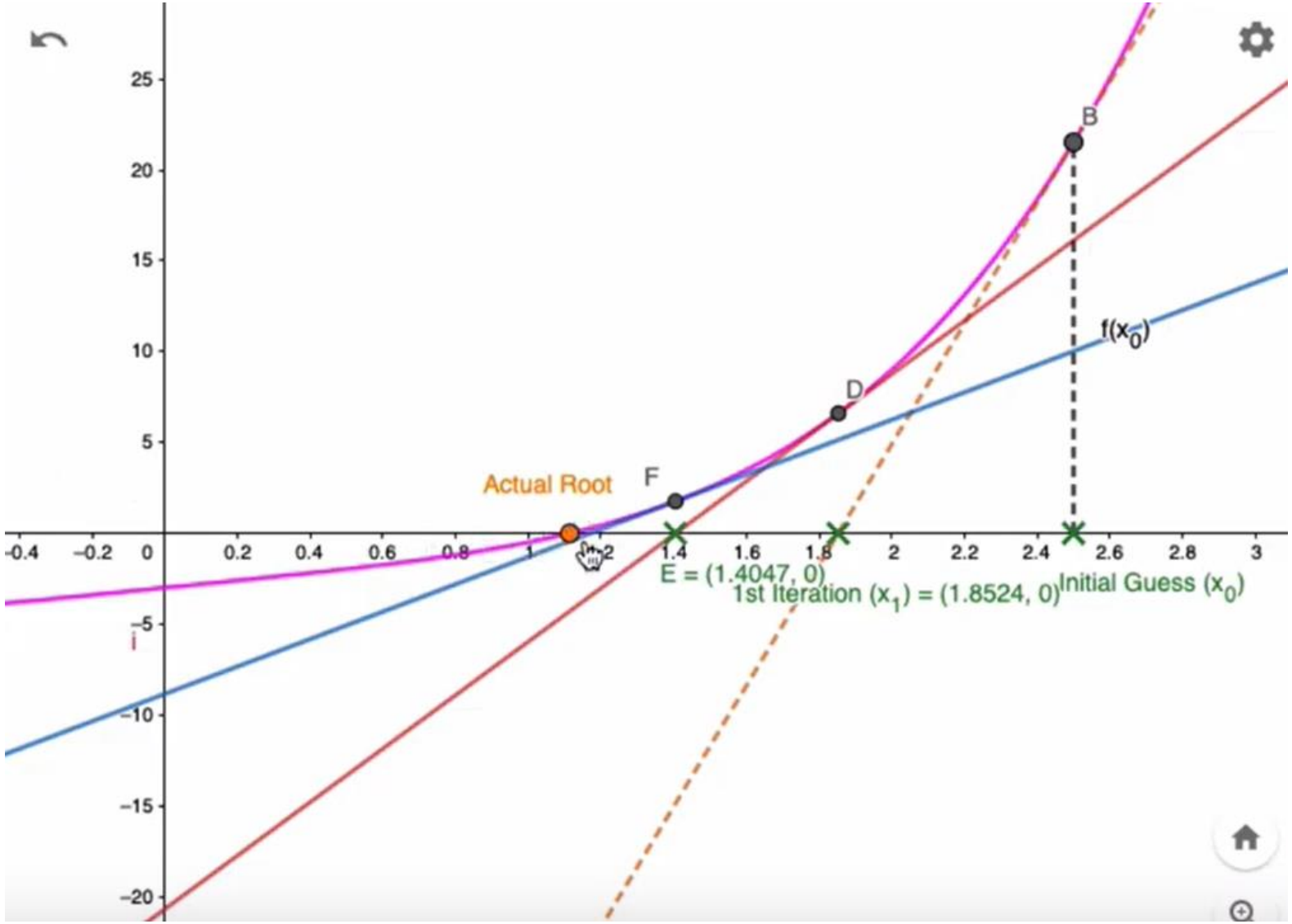


Newton's method

- **Key idea**
  - Make a initial guess, say x_i
  - A tangent can be drawn from the point $[x_i, f(x_i)]$
  - The point where this tangent crosses the x axis usually represents an improved estimate of the root



Newton's method



https://www.youtube.com/watch?v=R0no1yo-ckQ

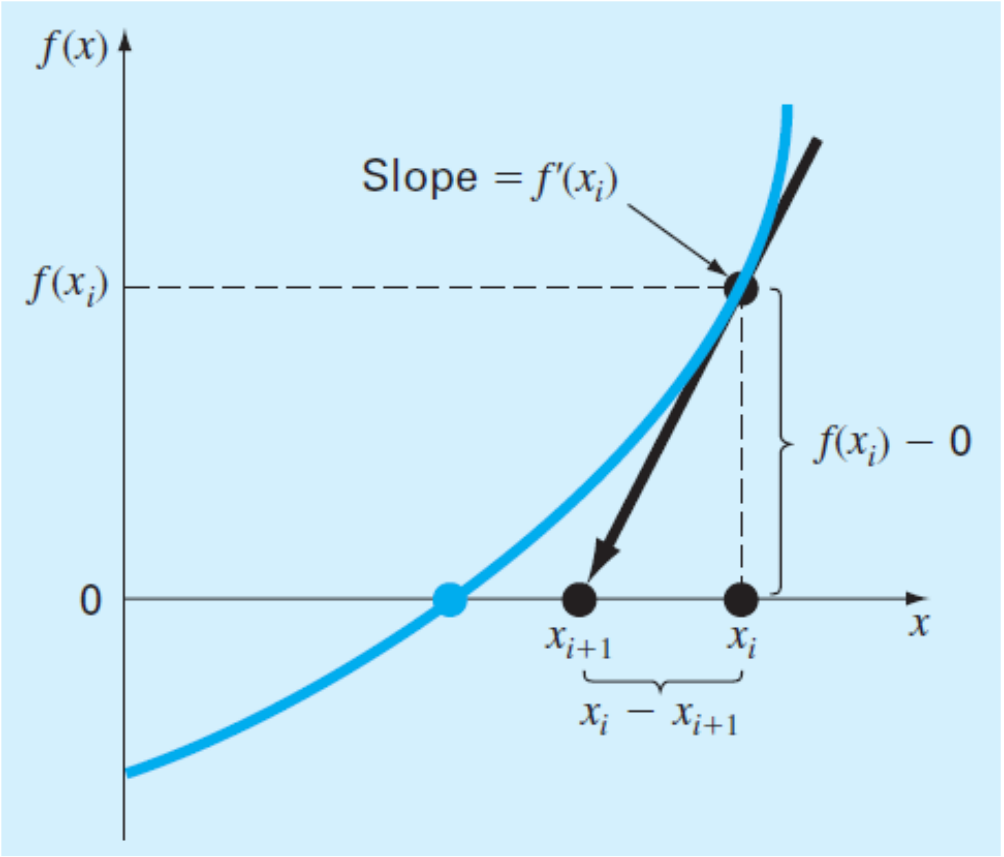
Newton's method

- We know that

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

Rearranging this gives

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$



Newton's method by Taylor Series Interpretation

- Let us write Taylor series expansion of $f(x)$ around the point x_i

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{f''(x_i)(x - x_i)^2}{2!} + \cdots$$

- Retaining only the first order term

$$f(x) = f(x_i) + f'(x_i)(x - x_i)$$

Good approximation of $f(x)$ at x_i and in the vicinity of x_i

- Consider x = r (root)

$$f(r) = f(x_i) + f'(x_i)(r - x_i)$$

Newton's method by Taylor Series Interpretation

- But $f(r) = 0$

$$0 = f(x_i) + f'(x_i)(r - x_i)$$

- This implies

$$r = x_i - \frac{f(x_i)}{f'(x_i)}$$

- In general

$$x^{[n+1]} = x^{[n]} - \frac{f\left(x^{[n]}\right)}{f'(x^{[n]})}$$

Newton's method: algorithm

- To determine a root of $f(x) = 0$, given x_0 reasonably close to the root,

Compute $f(x_0), f'(x_0)$.
If $f(x_0) \neq 0$ And $f'(x_0) \neq 0$ Then
Repeat
Set $x_1 = x_0$.
Set $x_0 = x_0 - \frac{f(x_0)}{f'(x_0)}$
Until $(|x_1 - x_0|) <* tol1$ Or $|f(x_1)| < tol2$.
End If.

Extended Newton's method

- When we derived Newton's method from Taylor series expansion, we retained only the first order term
- We can retain both linear and quadratic terms, it is called extended Newton's method

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{f''(x_i)(x - x_i)^2}{2}$$

- Consider x = r (root)

$$f(r) = f(x_i) + f'(x_i)(r - x_i) + \frac{f''(x_i)(r - x_i)^2}{2}$$

Extended Newton's method

- But $f(r) = 0$

$$0 = f(x_i) + f'(x_i)(r - x_i) + \frac{f''(x_i)(r - x_i)^2}{2}$$

$$\rightarrow \quad f(x_i) + (r - x_i)\left[f'(x_i) + \frac{f''(x_i)(r - x_i)}{2}\right] = 0$$

The $(r - x_i)$ term inside the square braces may be replaced by the results of the Newton Method

$$r = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$\rightarrow \quad f(x_i) + (r - x_i)\left[f'(x_i) - \frac{f''(x_i)\, f(x_i)}{2f'(x_i)}\right] = 0$$

Extended Newton's method

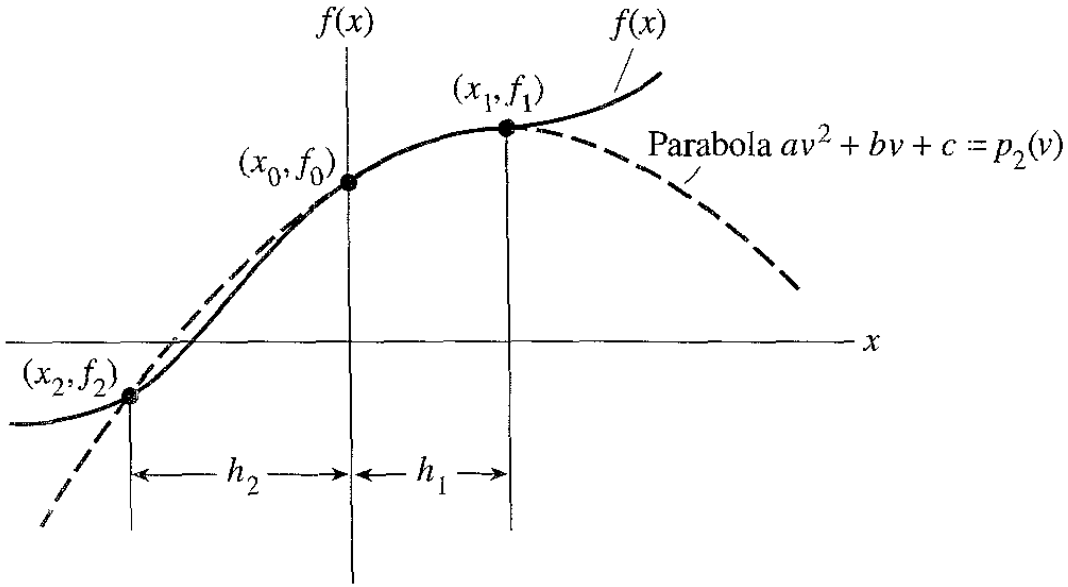
$$(r - x_i) = \frac{-f(x_i)}{\left[f'(x_i) - \frac{f''(x_i)\, f(x_i)}{2f'(x_i)}\right]}$$

$$\rightarrow \quad r = x_i - \frac{f(x_i)}{\left[f'(x_i) - \frac{f''(x_i)\, f(x_i)}{2f'(x_i)}\right]}$$

$$x^{[n+1]} = x^{[n]} - \frac{f\left(x^{[n]}\right)}{\left[f'(x^{[n]}) - \frac{f''(x^{[n]})\, f(x^{[n]})}{2f'(x^{[n]})}\right]}$$

Muller's method

- Key idea: approximate the function $f(x)$ with a quadratic polynomial for x.
- We need three initial guesses
- Similar to secant method

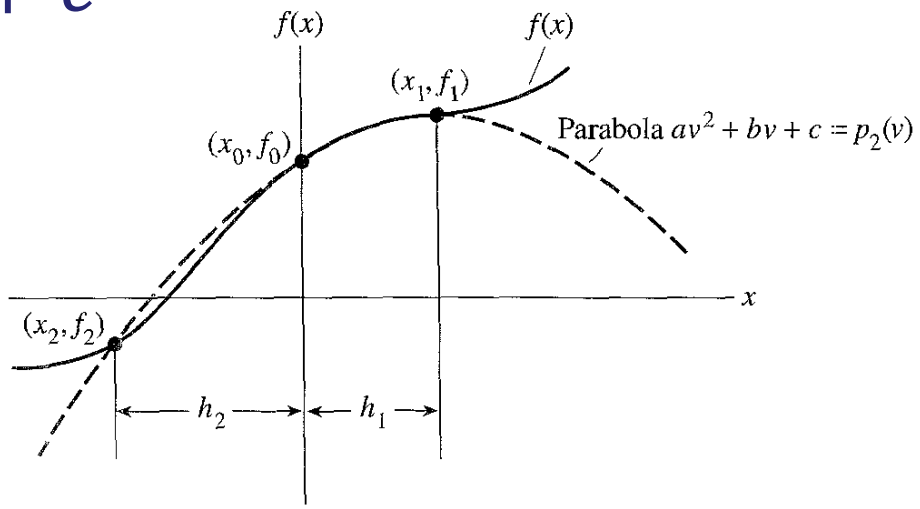


Muller's method

- Let the three initial guesses be (x_0, x_1, x_2). Let the function values at these three points be denoted by (f_0, f_1, f_2).
- Define $v = x - x_0$
- We construct a 2nd degree polynomial as

$$P_2(v) = av^2 + bv + c$$

- Let $h_1 = x_1 - x_0$ and $h_2 = x_0 - x_2$



Muller's method

- We estimate the coefficients (a, b and c) by evaluating $P_2(v)$ at the 3 points

$$v = 0: \quad a(0)^2 + b(0) + c = f_0;$$

$$v = h_1: \quad ah_1^2 + bh_1 + c = f_1;$$

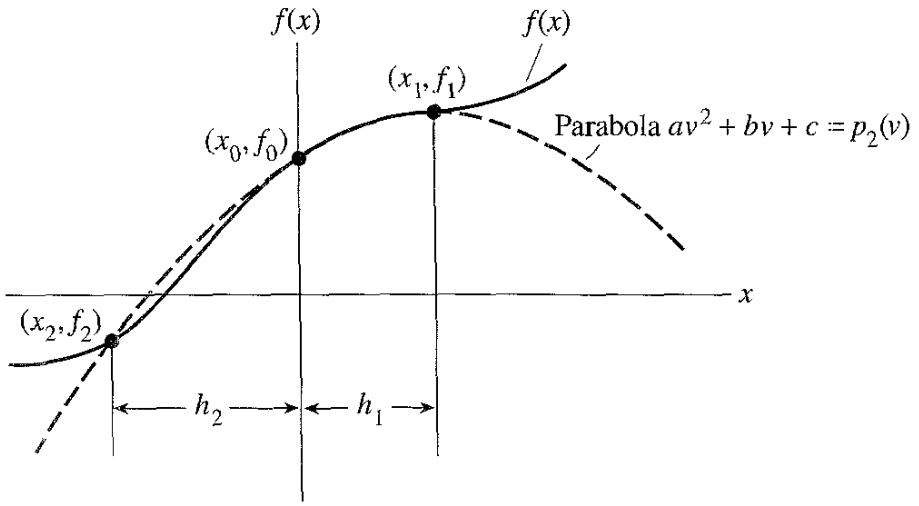
$$v = -h_2: \quad ah_2^2 - bh_2 + c = f_2.$$

$c = f_0$

2 eqns and 2 unknowns

$$a = \frac{f_2 + \gamma f_1 - f_0 (1+\gamma)}{\gamma h_1^2 (1+\gamma)} \quad \text{and} \quad b = \frac{f_1 - f_0 - ah_1^2}{h_1}$$

where $\gamma = \frac{h_2}{h_1}$



Muller's method

- We have now obtained the values of a, b & c

- The roots of the equation $P_2(v) = 0$ are given by (analytical expression for quadratic equation)

$$v = \tilde{r} - x_0 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\tilde{r} = x_0 + \left[\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right]$$

$\tilde{r}$ is a better guess for the root of $f(x) = 0$

CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

Acknowledgements: Prof. Laksh, NUS

Fixed point iteration method

- Key idea: Rearrange $f(x) = 0$ as $x = g(x)$ and then perform iterations as $x^{[n+1]} = g(x^{[n]})$
- Example: $f(x) = x^2 - 2x - 3 = 0$
  - Rearranging: $x^2 = 2x + 3 \rightarrow x = \sqrt{2x + 3}$
  - Iteration: Let us take $x = 4$ as initial guess

$$
\begin{aligned}
x_0 &= 4, \\
x_1 &= \sqrt{11} = 3.31662, \\
x_2 &= \sqrt{9.63325} = 3.10375, \\
x_3 &= \sqrt{9.20750} = 3.03439, \\
x_4 &= \sqrt{9.06877} = 3.01144, \\
x_5 &= \sqrt{9.02288} = 3.00381,
\end{aligned}
$$

Fixed point iteration method

- Rearranging in the form of $x = g(x)$ can be done in multiple ways

- Example: $f(x) = x^2 - 2x - 3 = 0$
  - $x^2 = 2x + 3 \rightarrow x = \sqrt{2x + 3}$
  - $x = (x^2 - 3\ )/2$
  - $x(x - 2) = 3 \rightarrow x = 3/(x - 2)$
  - $x^2 = 2x - 3 \rightarrow x = 2 - (3/x)$

Fixed point iteration method

- Let us try another rearrangement
- Example: $f(x) = x^2 - 2x - 3 = 0$
  - Rearranging: $x(x-2) = 3 \rightarrow x = 3/(x-2)$
  - Iteration: Let us take $x = 4$ as initial guess

$$x_0 = 4,$$
$$x_1 = 1.5,$$
$$x_2 = -6,$$
$$x_3 = -0.375,$$
$$x_4 = -1.263158,$$
$$x_5 = -0.919355,$$
$$x_6 = -1.02762,$$
$$x_7 = -0.990876,$$
$$x_8 = -1.00305,$$

Fixed point iteration method

- Let us try another rearrangement
- Example: $f(x) = x^2 - 2x - 3 = 0$
  - Rearranging: $x = (x^2 - 3)/2$
  - Iteration: Let us take $x = 4$ as initial guess

$$x_0 = 4,$$

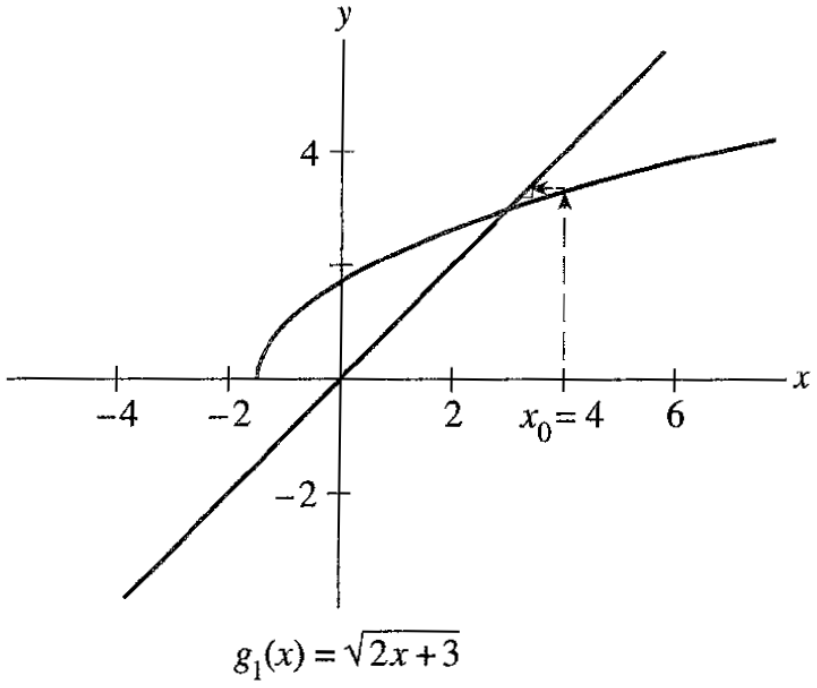
$$x_1 = 6.5,$$

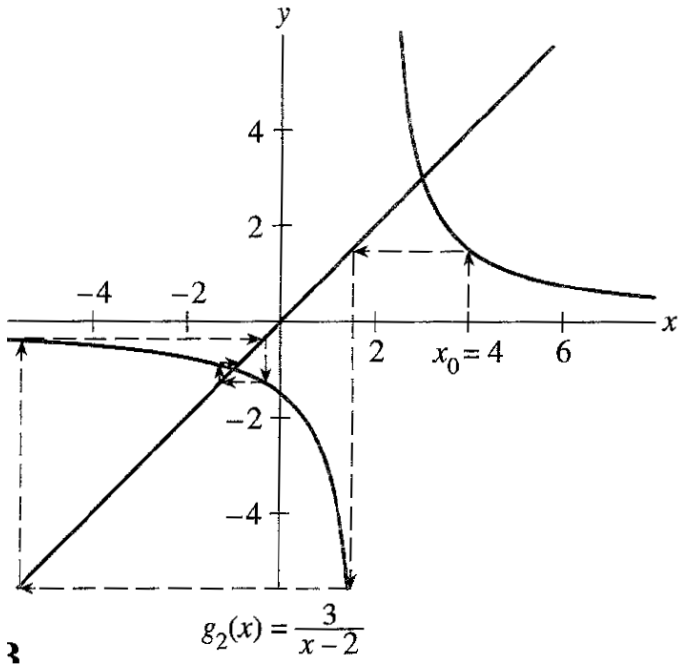
$$x_2 = 19.625,$$

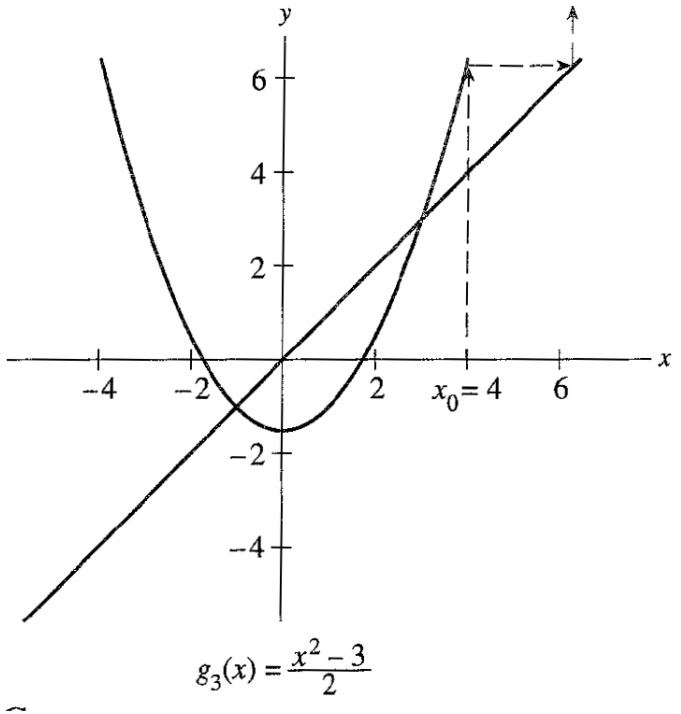
$$x_3 = 191.070,$$

Diverged!

Fixed point iteration method







- It appears that the different behaviors depend on whether the slope of the curve is greater, less, or of opposite sign to the slope of the line (which equals + 1).

Fixed point iteration method

- Convergence is an issue
  - Not all rearrangements lead to convergence

- Consider fixed point iteration:
  - $x^{[n+1]} = g\left(x^{[n]}\right)$
  - At convergence, $r = g(r)$

$$\rightarrow x^{[n+1]} - r = g\left(x^{[n]}\right) - g(r)$$

$$\rightarrow e^{[n+1]} = g\left(x^{[n]}\right) - g(r)$$

Fixed point iteration method

- Multiply and divide RHS with $x^{[n]} - r$
  - $e^{[n+1]} = \left(\frac{g(x^{[n]}) - g(r)}{x^{[n]} - r}\right)\left(x^{[n]} - r\right)$
- With mean value theorem

$$e^{[n+1]} = g'(\xi)\left(x^{[n]} - r\right)$$

where ξ lies between $x^{[n]}$ and r

$$\rightarrow e^{[n+1]} = g'(\xi)e^{[n]}$$

1. The error will decrease with every iteration if $|g'(\xi)|<1$
2. The rate of convergence is linear since $e^{[n+1]} \propto e^{[n]}$

Error analysis for Newton's method

- $x^{[n+1]} = x^{[n]} - \frac{f(x^{[n]})}{f'(x^{[n]})}$
- Comparing with the fixed point iteration method (i.e., $x = g(x)$), we can say

$$g(x) = x - \frac{f(x)}{f'(x)}$$

- We have found out that the method will converge if $|g'(\xi)| < 1$

$$g'(x) = 1 - \frac{(f')^2 - ff''}{(f')^2} = \frac{ff''}{(f')^2}$$

- At root $(x = r)$, $f(r) = 0$ and $f'(r) \neq 0$

Thus, $g'(r) = 0$

For Newton scheme, we have |g'(r)| < 1. Therefore, with good initial guess, the Newton Scheme will converge.

System of nonlinear algebraic equations

$$
\begin{aligned}
f_1\left(x_1, x_2, \cdots, x_n\right) &= 0 \\
f_2\left(x_1, x_2, \cdots, x_n\right) &= 0 \\
&\vdots \\
f_n\left(x_1, x_2, \cdots, x_n\right) &= 0
\end{aligned}
$$

- In short form, $\underline{f}(\underline{x}) = \underline{0}$
- The equations are satisfied at the root ($\underline{x} = \underline{r}$), i.e., $x_1 = r_1, x_2 = r_2, \ldots., x_n = r_n$

Newton method for set of equations

- Let us consider two nonlinear equations with two unknowns

$$f_1(x_1, x_2) = 0$$
$$f_2(x_1, x_2) = 0$$

$$\underline{f}(\underline{x}) = \underline{0}$$

- Assuming that the roots $\underline{r}$ are {r1, r2}

Newton method for set of equations

- Let us revise Taylor series expansion of $f(x)$ around the point $x^{[0]}$ (for a single variable)

$$f(x) = f\left(x^{[0]}\right) + f'\left(x^{[0]}\right)\left(x - x^{[0]}\right) + \frac{f''(x^{[0]})(x - x^{[0]})^2}{2!} + \cdots$$

- Taylor series expansion of $\underline{f}(\underline{x})$ around the point $\left(x_1^{[0]}, x_2^{[0]}\right)$ (for two variables)

$$f_1(x_1, x_2) = f_1\left(x_1^{[0]}, x_2^{[0]}\right) + \left[\frac{\partial f_1}{\partial x_1}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)}\left(x_1 - x_1^{[0]}\right) + \left[\frac{\partial f_1}{\partial x_2}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)}\left(x_2 - x_2^{[0]}\right) + \cdots$$

$$f_2(x_1, x_2) = f_2\left(x_1^{[0]}, x_2^{[0]}\right) + \left[\frac{\partial f_2}{\partial x_1}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)}\left(x_1 - x_1^{[0]}\right) + \left[\frac{\partial f_2}{\partial x_2}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)}\left(x_2 - x_2^{[0]}\right) + \cdots$$

Newton method for set of equations

- Consider $\underline{x} = \underline{r}$ (roots)

$$f_1(r_1, r_2) = f_1\left(x_1^{[0]}, x_2^{[0]}\right) + \left[\frac{\partial f_1}{\partial x_1}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \left(r_1 - x_1^{[0]}\right) + \left[\frac{\partial f_1}{\partial x_2}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \left(r_2 - x_2^{[0]}\right)$$

$$f_2(r_1, r_2) = f_2\left(x_1^{[0]}, x_2^{[0]}\right) + \left[\frac{\partial f_2}{\partial x_1}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \left(r_1 - x_1^{[0]}\right) + \left[\frac{\partial f_2}{\partial x_2}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \left(r_2 - x_2^{[0]}\right)$$

- But $\underline{f}(\underline{r}) = \underline{0}$

$$0 = f_1\left(x_1^{[0]}, x_2^{[0]}\right) + \left[\frac{\partial f_1}{\partial x_1}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \left(r_1 - x_1^{[0]}\right) + \left[\frac{\partial f_1}{\partial x_2}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \left(r_2 - x_2^{[0]}\right)$$

$$0 = f_2\left(x_1^{[0]}, x_2^{[0]}\right) + \left[\frac{\partial f_2}{\partial x_1}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \left(r_1 - x_1^{[0]}\right) + \left[\frac{\partial f_2}{\partial x_2}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \left(r_2 - x_2^{[0]}\right)$$

Newton method for set of equations

- Let us introduce $\delta_1 = r_1 - x_1^{[0]}$ and $\delta_2 = r_2 - x_2^{[0]}$

$$0 = f_1\left(x_1^{[0]}, x_2^{[0]}\right) + \left[\frac{\partial f_1}{\partial x_1}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \delta_1 + \left[\frac{\partial f_1}{\partial x_2}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \delta_2$$

$$0 = f_2\left(x_1^{[0]}, x_2^{[0]}\right) + \left[\frac{\partial f_2}{\partial x_1}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \delta_1 + \left[\frac{\partial f_2}{\partial x_2}\right]_{\left(x_1^{[0]}, x_2^{[0]}\right)} \delta_2$$

- In matrix form

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}_{\left(x_1^{[0]}, x_2^{[0]}\right)} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_{\left(x_1^{[0]}, x_2^{[0]}\right)} \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix}$$

- In compact form

$$\underline{0} = \underline{f}^{[0]} + \underline{\underline{J}}^{[0]} \underline{\delta} \qquad \rightarrow \underline{\delta} = -\left[\underline{\underline{J}}^{[0]}\right]^{-1} \underline{f}^{[0]}$$

The improved guess $\underline{x}^{[1]}$ is obtained as $\underline{x}^{[0]} + \underline{\delta}$. Continue iterations until convergence.

Newton method for set of equations

- In general, we can write

$$\underline{\delta}^{[n]} = -\left[\underline{\underline{J}}^{[n]}\right]^{-1} \underline{f}^{[n]}$$

$$\underline{x}^{[n+1]} = \underline{x}^{[n]} + \underline{\delta}^{[n]}$$

$$\text{Or } \underline{x}^{[n+1]} = \underline{x}^{[n]} - \left[\underline{\underline{J}}^{[n]}\right]^{-1} \underline{f}^{[n]}$$

- Solve

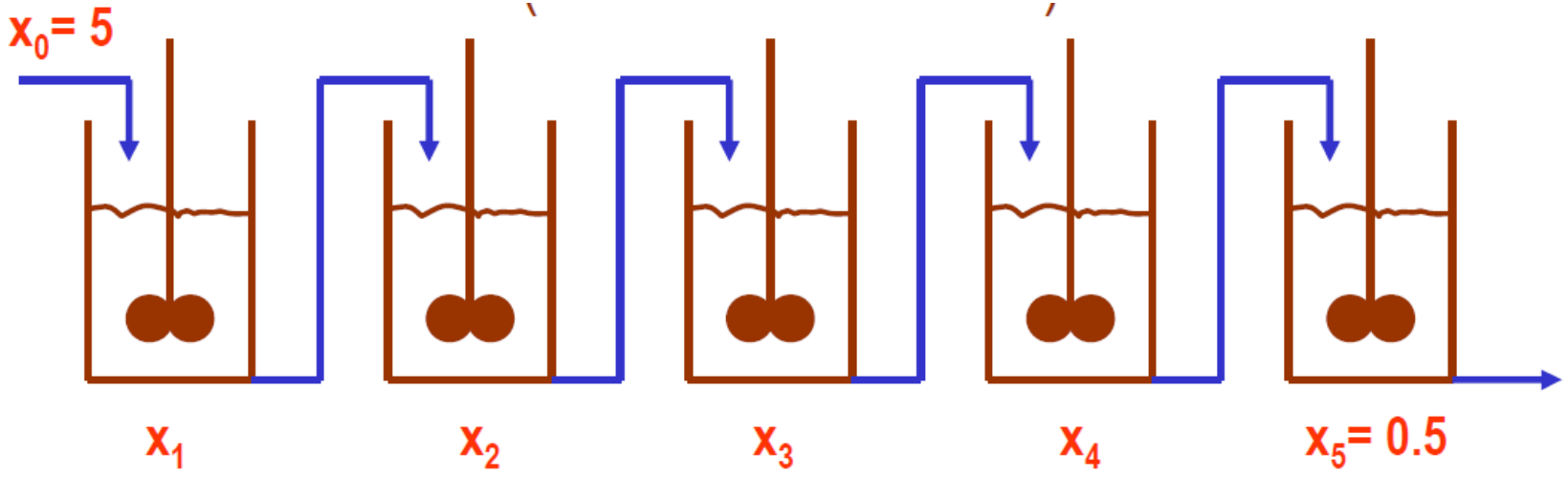
$$F_1(\boldsymbol{y}) = 4 - 8y_2 + 4y_3 - 2y_2^3 = 0$$
$$F_2(\boldsymbol{y}) = 1 - 4y_2 + 3y_3 + y_3^2 = 0$$

Using the Newton's method, starting with

$y^{[0]} = \left[y_2^{[0]}\ y_3^{[0]}\right] = [0.5\ 0.5]$.

Example

- 5 CSTR Of equal volume operating in series at steady state. Reaction ($A \rightarrow B$) in CSTR is second order, reaction rate constant is unity. Volumetric flow rate at inlet of 1st reactor is 200 m^3/s. In the figure below, x denotes concentration of A. Find Volume of each reactor.



Example

- The combustion of a hydrocarbon in air leads to the formation of ten different products. This combustion process can be represented by the following system of non-linear algebraic equations.

$$f_1 = n_1 + n_4 - 3 = 0$$
$$f_2 = 2n_1 + n_2 + n_4 + n_7 + n_8 + n_9 + 2n_{10} - R = 0$$
$$f_3 = 2n_2 + 2n_5 + n_6 + n_7 - 8 = 0$$
$$f_4 = 2n_3 + 2n_9 - 4R = 0$$
$$f_5 = K_5 n_2 n_4 - n_1 n_5 = 0$$
$$f_6 = K_6 n_1^{1/2} n_4^{1/2} - n_1^{1/2} n_6 (p/n_T)^{1/2} = 0$$
$$f_7 = K_7 n_1^{1/2} n_2^{1/2} - n_4^{1/2} n_7 (p/n_T)^{1/2} = 0$$
$$f_8 = K_8 n_1 - n_4 n_8 (p/n_T) = 0$$
$$f_9 = K_9 n_1^{1/2} n_3^{1/2} - n_4 n_9 (p/n_T)^{1/2} = 0$$
$$f_{10} = K_{10} n_1^2 - n_4^2 n_{10} (p/n_T) = 0$$

where n_1, n_2, n_3 ..... n_{10} are the number of moles of each product formed per mole of hydrocarbon undergoing combustion and n_T is total moles of product formed. R is a parameter expressing the relative amounts of air and the hydrocarbon. The value of various parameters for the above mentioned equations are as follows:

Solve the above system of algebraic equations to find the number of moles of each product produced per mole of the reactant. Physical considerations dictate that all the $\mathbf{n_i}$ are real and positive.

$K_5 = 0.193$, $K_6 = 0.002597$, $K_7 = .003448$, $K_8 = 1.799 \times 10^{-5}$; $K_9 = 1.255 \times 10^{-4}$, $K_{10} = 3.3846 \times 10^{-5}$; $p = 40$ and $R = 15$.

Example

The mass balance equations for the following reactions taking place in a CSTR [18]:

$$A \xrightarrow{k_1} 2B$$

$$A \underset{k_3}{\overset{k_2}{\rightleftharpoons}} C$$

$$B \xrightarrow{k_4} D + C$$

is given by

$$F_1 = -C_A + C_{Ao} + \left[-k_1 C_A - k_2 C_A^{3/2} + k_3 C_C^2\right]\theta = 0$$

$$F_2 = -C_B + \left(2k_1 C_A - k_4 C_B^2\right)\theta = 0$$

$$F_3 = -C_C + \left(k_2 C_A^{3/2} - k_3 C_C^2 + k_4 C_B^2\right)\theta = 0$$

$$F_4 = -C_D + \left(k_4 C_B^2\right)\theta = 0$$

$k_1 = 1.0\ 1/s$

$k_2 = 0.2\ \text{lit}^{1/2}/\text{mol}^{1/2} - s$

$k_3 = 0.05\ \text{lit} / \text{mol} - s$

$k_4 = 0.4\ \text{lit} / \text{mol} - s$

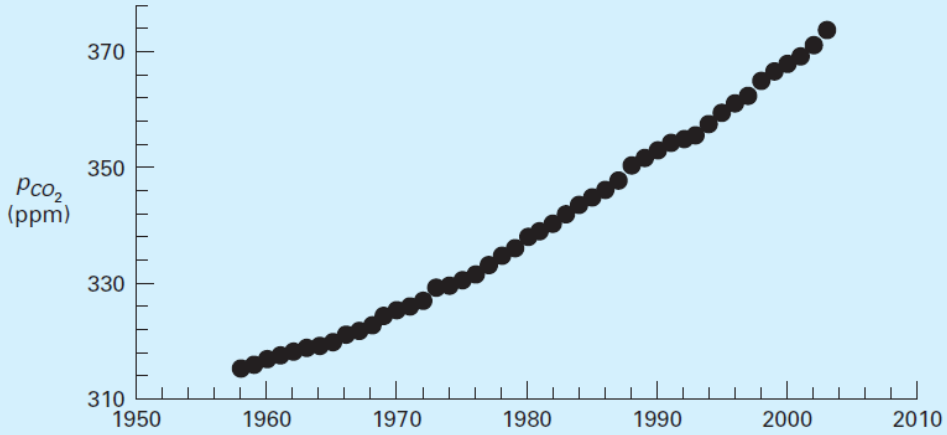
$\theta = 2\ s$

$C_{Ao} = 1\ \text{mol} / \text{lit}$

Example

It is well documented that carbon dioxide is the primary determinant of the pH of the rain. How this increasing CO2 trend is affecting the pH of rainwater?

Average annual partial pressures of atmospheric carbon dioxide (ppm) measured at Mauna Loa, Hawaii.



$$K_1 = 10^6 \frac{[H^+][HCO_3^-]}{K_H p_{CO_2}}$$

$$K_2 = \frac{[H^+][CO_3^{-2}]}{[HCO_3^-]}$$

$$K_w = [H^+][OH^-]$$

$$c_T = \frac{K_H p_{CO_2}}{10^6} + [HCO_3^-] + [CO_3^{-2}]$$

$$0 = [HCO_3^-] + 2[CO_3^{-2}] + [OH^-] - [H^+]$$

where K_H = Henry's constant, and K_1, K_2, and K_w are equilibrium coefficients. The five unknowns are c_T = total inorganic carbon, $[HCO_3^-]$ = bicarbonate, $[CO_3^{-2}]$ = carbonate, $[H^+]$ = hydrogen ion, and $[OH^-]$ = hydroxyl ion. Notice how the partial pressure of CO_2 shows up in Eqs. (5.9) and (5.12).

Use these equations to compute the pH of rainwater given that $K_H = 10^{-1.46}$, $K_1 = 10^{-6.3}$, $K_2 = 10^{-10.3}$, and $K_w = 10^{-14}$. Compare the results in 1958 when the p_{CO_2} was 315 and in 2008 when it was 386 ppm. When selecting a numerical method for your computation, consider the following:

- You know with certainty that the pH of rain in pristine areas always falls between 2 and 12.
- You also know that pH can only be measured to two places of decimal precision.

https://personal.ems.psu.edu/~brune/m532/m532_ch5_aqueous_phase.htm

Raoult's law

- The partial vapor pressure of any volatile component of a solution is the product of vapour pressure of that pure component and the mole fraction of the component in the solution.

$$p_i = p_i^{sat} x_i \rightarrow y_i = K_i x_i$$

$$K_i = p_i^{\mathrm{sat}}(T)/p$$

Bubble point calculations

- The bubble point of a liquid is the point where the liquid just starts to evaporate (boil), that is, when the first vapor bubble is formed.
  - If the temperature is given, then we must lower the pressure until the first bubble is formed.
  - If the pressure is given, then we must increase the temperature until the first bubble is formed.
- In both cases, this corresponds to adjusting T or p until the computed sum of vapor fractions is just 1, $\Sigma y_i = 1$ or

$$\Sigma_i K_i x_i = 1$$

$$\Sigma_i x_i p_i^{\mathrm{sat}}(T) = p$$

Bubble point calculations

$$\sum_i x_i p_i^{\mathrm{sat}}(T) = p$$

- The liquid-phase composition x_i is known and

$$\log\left(p_i^{sat}\right) = A_i - \frac{B_i}{C_i + T}$$

- Thus, we can write

$$\sum_i x_i 10^{\left(A_i - \frac{B_i}{C_i + T}\right)} = p$$

- If p is Known, we get an implicit equation for T which needs to be solved numerically

Example

- Consider a liquid mixture with 50% pentane (1), 30% hexane (2) and 20% cyclohexane (3) (all in mol-%). At p = 5 bar, the temperature is gradually increased. What is the bubble temperature and composition of the first vapor that is formed?

```
% log10(psat[bar])=A-B/(T[K]+C)          Tb[K]         dvapHb [J/mol]
A1=3.97786;   B1=1064.840;  C1=-41.136;  Tb1=309.22;  dvapHb1=25790;  % pentane   C5H12
A2=4.00139;   B2=1170.875;  C2=-48.833;  Tb2=341.88;  dvapHb2=28850;  % hexane    C6H14
A3=3.93002;   B3=1182.774;  C3=-52.532;  Tb3=353.93;  dvapHb3=29970;  % cyclohex  C6H12
A4=5.20277;   B4=1580.080;  C4=-33.650;  Tb4=337.69;  dvapHb4=35210;  % methanol  CH3OH
A5=5.11564;   B5=1687.537;  C5=-42.98;   Tb5=373.15;  dvapHb5=40660;  % water     H2O
A6=4.48540;   B6= 926.132;  C6=-32.98;   Tb6=239.82;  dvapHb6=23350;  % ammonia   NH3
A7=3.92828;   B7= 803.997;  C7=-26.11;   Tb7=231.02;  dvapHb7=19040;  % propane   C3H8
A8=4.05075;   B8=1356.360;  C8=-63.515;  Tb8=398.82;  dvapHb8=34410;  % octane    C8H18
A9=4.12285;   B9=1639.270;  C9=-91.310;  Tb9=489.48;  dvapHb9=43400;  % dodecane  C12H26
A10=3.98523; B10=1184.24;  C10=-55.578; Tb10=353.24; dvapHb11=30720;  % benzene   C6H6
A11=4.05043; B11=1327.62;  C11=-55.525; Tb11=383.79; dvapHb11=33180;  % toluene   C7H8
```

Dew point calculations

- In this case the vapor-phase composition y_i is given
- The dew point of a vapor (gas) is the point where the vapor just begins to condense, that is, when the first liquid drop is formed.
  - If the temperature is given, then we must increase the pressure until the first liquid is formed.
  - If the pressure is given, then we must decrease the temperature until the first liquid is formed.
- In both cases, this corresponds to adjusting T or p until $\Sigma x_i = 1$ or

$$\Sigma_i y_i / K_i = 1 \quad \text{or} \quad \Sigma_i \frac{y_i}{p_i^{\text{sat}}(T)} = \frac{1}{p}$$

Dew point calculations

$$\sum_i \frac{y_i}{p_i^{\mathrm{sat}}(T)} = \frac{1}{p}$$

- The vapor-phase composition y_i is known and

$$\log\left(p_i^{sat}\right) = A_i - \frac{B_i}{C_i + T}$$

- Thus, we can write

$$\sum_i \frac{y_i}{10^{\left(A_i - \frac{B_i}{C_i + T}\right)}} = \frac{1}{p}$$

- If p is Known, we get an implicit equation for T which needs to be solved numerically

Example

- Consider a vapor mixture with 50% pentane (1), 30% hexane (2) and 20% cyclohexane (3). At p = 5 bar, the temperature is gradually decreased. What is the dew point temperature and the composition of the first liquid that is formed?

Flash calculations

- Consider a flash where a feed F (with composition zi) is split into a vapor product V (with composition y_i) and a liquid product (with composition x_i)
- For each of the components, we can write a material balance

$$FZ_i = Lx_i + Vy_i$$

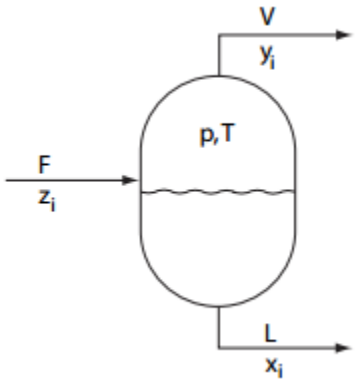
- In addition, the vapor and liquid is assumed to be in equilibrium

$$y_i = K_i x_i$$

- K_i can be evaluated using Antoine equation

$$\log\left(p_i^{sat}\right) = A_i - \frac{B_i}{C_i + T}$$

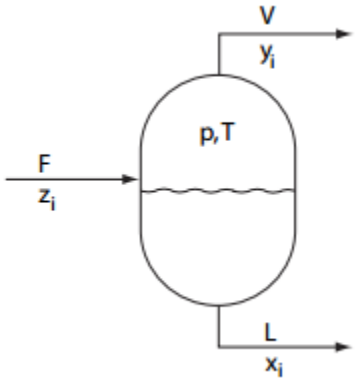
- In addition, we have the two relationships $\Sigma x_i = 1$ and $\Sigma y_i = 1$.
- The feed is given (F, z_i). Unknowns: x_i, y_i, K_i, T, P, L, V
- If there are N-components, no. of unknowns = 3N+4
- Number of equations we have= 3N+2



The simplest flash is usually to specify p and T (pT -flash),

https://folk.ntnu.no/skoge/bok/mer/flash_english_edition_2009

pT-Flash calculations



- Mass balance for ith component

$$FZ_i = Lx_i + Vy_i$$

- Substituting $y_i = Kx_i$ from Raoult's law

$$FZ_i = Lx_i + VK_ix_i$$

- We can rewrite the above

$$x_i = (Fz_i/(L + VK_i)$$

- Introduce L=F-V (total mass balance)

$$x_i = \frac{z_i}{1 + \frac{V}{F}(K_i - 1)}$$

https://folk.ntnu.no/skoge/bok/mer/flash_english_edition_2009

$$x_i = \frac{z_i}{1 + \frac{V}{F}(K_i - 1)}$$

- We cannot directly calculate x_i because the vapor split V/F is not known

- To calculate V/F, we exploit the fact that

$$\Sigma_i (y_i - x_i) = 0$$

$$\Sigma_i \frac{z_i(K_i - 1)}{1 + \frac{V}{F}(K_i - 1)} = 0$$

Rachford-Rice flash equation

$$\Sigma_i \frac{z_i(K_i - 1)}{1 + \frac{V}{F}(K_i - 1)} = 0 \qquad K_i = p_i^{\text{sat}}(T)/p$$

- With T and p specified, we know K_i and the Rachford-Rice equation can be solved for V/F.

- Later on, we can calculate x_i and y_i

$$x_i = \frac{z_i}{1 + \frac{V}{F}(K_i - 1)}$$

- A feed F is split into a vapor product V and a liquid product L in a flash tank. The feed is 50% pentane, 30% hexane and 20% cyclohexane (all in mol-%). In the tank, T = 390K and p = 5 bar. For example, we may have a heat exchanger that keeps constant temperature and a valve on the vapor product stream that keeps constant pressure. We want to find the product split and product compositions. Assume ideal liquid mixture and ideal gas (Raoult's law)

CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

Boundary value problem (BVP)

- Diffusion followed by 1st order Rxn in a Slab

$$\frac{d^2y}{dx^2} - y = 0$$

- Boundary conditions

1. $\frac{dy}{dx} = 0$ at $x = 0$
2. $y = 1$ at $x = 1$

Finite Difference Method

- Key idea:

  Convert the ODE's and boundary conditions into a set of algebraic equations (linear or nonlinear) using Numerical differentiation formulas.

  Solve the resulting algebraic equation system using the methods studied earlier.

Steps

- Step 1: Break the solution domain into equal sized sub-domains using equidistant grid points.

Steps

- Step 2: Replace the derivatives in the ODE's and B.C.'s by appropriate finite difference approximations.
  - In doing so, make sure that the truncation error of all approximations are of the same order
  - This converts the DAE system into an algebraic system.

- Step 3: Set up the algebraic equation system and solve it

Numerical differentiation formula

- Finite-difference approximations provide a means to transform derivatives into algebraic form

Forward Difference Formula (1st order accurate)

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Backward Difference Formula (1st order accurate)

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

Centered Difference Formula (2nd order accurate)

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Centered Difference Formula for 2nd Derivative (2nd order accurate)

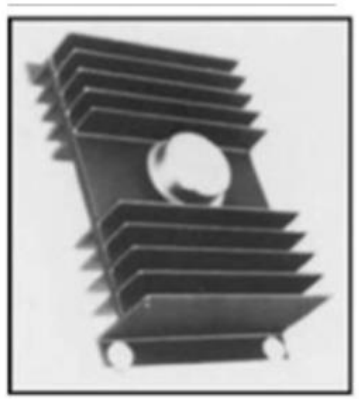
$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

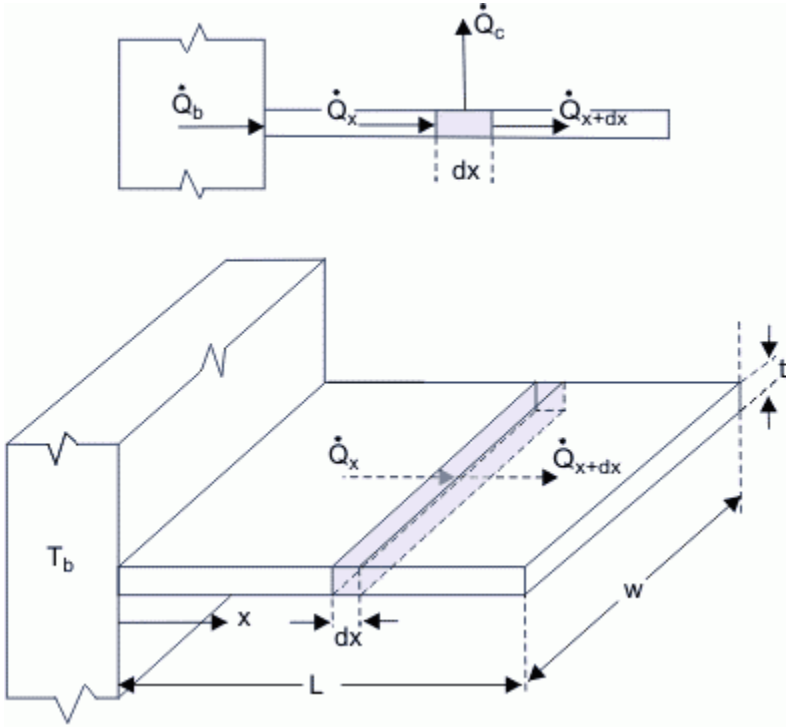
First Derivative		
Method	**Formula**	**Truncation Error**
Two-point forward difference	$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
Three-point forward difference	$f'(x_i) = \frac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2})}{2h}$	$O(h^2)$
Two-point backward difference	$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
Three-point backward difference	$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h}$	$O(h^2)$
Two-point central difference	$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$

Second Derivative		
Method	**Formula**	**Truncation Error**
Three-point forward difference	$f''(x_i) = \frac{f(x_i) - 2f(x_{i+1}) + f(x_{i+2})}{h^2}$	$O(h)$
Four-point forward difference	$f''(x_i) = \frac{2f(x_i) - 5f(x_{i+1}) + 4f(x_{i+2}) - f(x_{i+3})}{h^2}$	$O(h^2)$
Three-point backward difference	$f''(x_i) = \frac{f(x_{i-2}) - 2f(x_{i-1}) + f(x_i)}{h^2}$	$O(h)$
Four-point backward difference	$f''(x_i) = \frac{-f(x_{i-3}) + 4f(x_{i-2}) - 5f(x_{i-1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Three-point central difference	$f''(x_i) = \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1})}{h^2}$	$O(h^2)$

Example

- Heat transfer through a fin
- https://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node128.html





Example: fin

- Equation governing the heat transfer in the fin

$$\frac{d^2\theta}{d\xi^2} - \frac{hPL^2}{kA}\theta = 0$$

- Boundary conditions:

$$\theta(\xi = 0) = 1; \theta(\xi = 1) = 0$$

$$\theta = \frac{T - T_\infty}{T_0 - T_\infty}$$

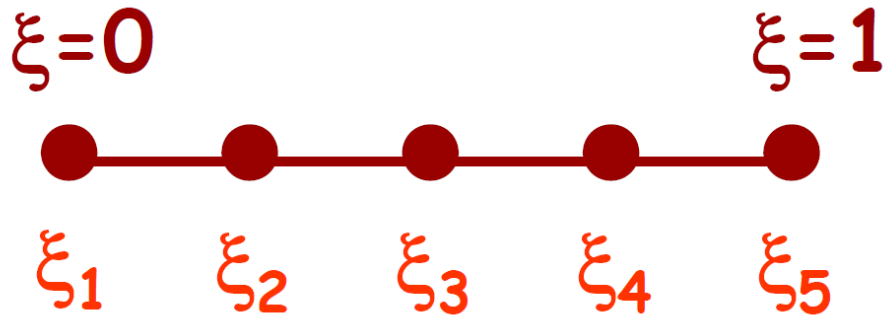
- Let $\frac{hPL^2}{kA} = 1$. This gives

$$\frac{d^2\theta}{d\xi^2} - \theta = 0$$

Example: fin

$$\frac{d^2\theta}{d\xi^2} - \theta = 0$$

- Divide the fin into 5 equispaced nodes
- 5 grid points; 3 internal grid points; 2 boundary grid points
- For internal grid points



$$\frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{\Delta\xi^2} - \theta_i = 0$$

$$\Delta\xi = 1/4$$

$$\theta_{i+1} - 2\theta_i + \theta_{i-1} - 0.0625\theta_i = 0$$

We know that $\theta_1 = 1, \theta_5 = 0$

$$\theta_{i-1} - 2.0625\theta_i + \theta_{i+1} = 0$$

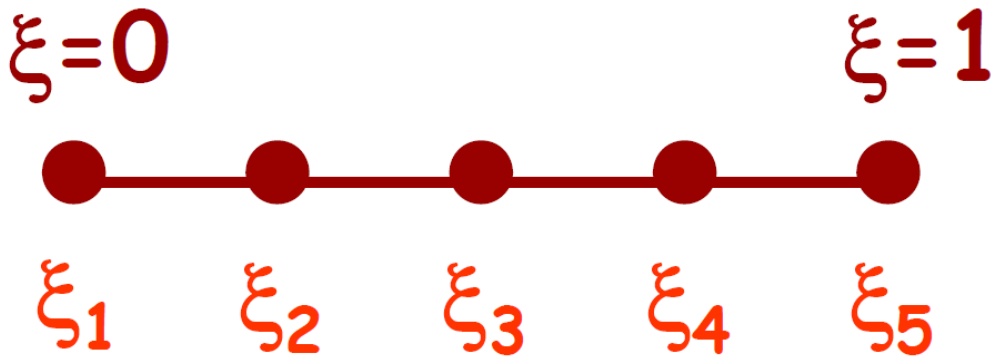
Example: fin

$$\theta_{i-1} - 2.0625\theta_i + \theta_{i+1} = 0$$

$\theta_1 - 2.0625\theta_2 + \theta_3 = 0$ (for node 2)

$\theta_2 - 2.0625\theta_3 + \theta_4 = 0$ (for node 3)

$\theta_3 - 2.0625\theta_4 + \theta_5 = 0$ (for node 4)



$\Delta\xi = 1/4$

We know that $\theta_1 = 1, \theta_5 = 0$

$-2.0625\theta_2 + \theta_3 = -1$

$\theta_2 - 2.0625\theta_3 + \theta_4 = 0$

$\theta_3 - 2.0625\theta_4 = 0$

Example: fin

$$\begin{bmatrix} -2.0625 & 1 & 0 \\ 1 & -2.0625 & 1 \\ 0 & 1 & -2.0625 \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

A tridiagonal structure

Example 2

- Heat Transfer through a variable Cross section area Fin

$$A(\xi)\frac{d^2\theta}{d\xi^2}+\frac{dA(\xi)}{d\xi}\frac{d\theta}{d\xi}-\frac{hPL^2}{k}\theta=0$$

$$\theta\left(\xi=0\right)=1;\quad \left.\frac{d\theta}{d\xi}\right|_{\xi=1}=0$$

$$Let\ \frac{hPL^2}{k}=2 \quad \textbf{and } A(\xi) = 5 - 4\xi$$

Solving BVPs in MATLAB

- Solver name: bvp4c

Syntax

```
sol = bvp4c(odefun,bcfun,solinit)
sol = bvp4c(odefun,bcfun,solinit,options)
```

Description

`sol = bvp4c(odefun,bcfun,solinit)` integrates a system of differential equations of the form $y' = f(x,y)$ specified by `odefun`, subject to the boundary conditions described by `bcfun` and the initial solution guess `solinit`. Use the `bvpini` function to create the initial guess `solinit`, which also defines the points at which the boundary conditions in `bcfun` ar enforced.

Example

- Consider $y'' + y = 0$

The equation is defined on the interval $[0, \pi/2]$ subject to the boundary conditions

$$y(0) = 0,$$

$$y(\pi/2) = 2.$$

- First, we rewrite the equation as a system of first order equations (substituting $y_1 = y$ and $y_2 = y'$)

$$y_1' = y_2$$

$$y_2' = -y_1$$

Solution using bvp4c

```
sol = bvp4c(odefun,bcfun,solinit)
```

- First input to bvp4c is odefun

```
function dydx = bvpfcn(x,y)
dydx = zeros(2,1);
dydx = [y(2)
        -y(1)];
end
```

$$y_1' = y_2,$$
$$y_2' = -y_1.$$

- Second input is bcfun

```
function res = bcfcn(ya,yb)
res = [ya(1)
       yb(1)-2];
end
```

$$y(0) = 0,$$
$$y(\pi/2) = 2.$$

- Third input is solinit
  - Use the bvpinit function to create the node points and an initial guess for the solution of the equation.

```
xmesh = linspace(0,pi/2,5);
solinit = bvpinit(xmesh, [2,2]);
```

```
sol = bvp4c(@bvpfcn,@bcfcn,solinit);

plot(sol.x, sol.y, '-o')
```

Example 2

- Solve using bvp4c,
- Heat Transfer through a variable Cross section area Fin

$$A(\xi)\frac{d^2\theta}{d\xi^2}+\frac{dA(\xi)}{d\xi}\frac{d\theta}{d\xi}-\frac{hPL^2}{k}\theta=0$$

$$\theta\left(\xi=0\right)=1;\quad \left.\frac{d\theta}{d\xi}\right|_{\xi=1}=0$$

$$Let\ \frac{hPL^2}{k}=2 \quad \textbf{and } A(\xi) = 5 - 4\xi$$

Example: IVP

- Consider a plug flow reactor in which A and C are fed in equimolar amounts (2 mol/m^3) and the following reaction takes place $2A \rightarrow B$. Reaction takes place in liquid phase and the volumetric flow rate remains constant even when reaction occurs. Flow velocity in the reactor is 0.5 m/s, reaction rate constant is 0.3 m^3/(kmol.s) and the total reactor length is 2.4 m. Plot the concentration profiles of A, B and C along the length of the reactor.

- Problem 24.13 in chapra. Axial dispersed PFR reactor.

CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

Acknowledgements: Prof. Laksh, NUS

Differential equations

- Ordinary differential equations
  - Initial value problem (IVP)
  - Boundary value problem (BVP)

- Partial differential equations

$$\frac{dy}{dx} = \sin x\, e^{y}$$

$$\frac{d^2y}{dx^2} + 10\frac{dy}{dx} + 9y = 0$$

$$\frac{d^4y}{dx^4} + \frac{d^3y}{dx^3} + \frac{d^2y}{dx^2} + \frac{dy}{dx} + y = \cos x$$

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + u = e^{x-y}$$

$$3\frac{\partial^2 u}{\partial x^2} + 7\frac{\partial u}{\partial x \partial y} + 6\frac{\partial^2 u}{\partial y^2} = 0$$

$$\frac{\partial v}{\partial t} - k\frac{\partial v}{\partial x^2} = v$$

IVP and BVP

- Differential equations always require some conditions to achieve unique solutions
- When all of the conditions on the problem are specified at the same value for the independent variable, the problem is termed an *initial-value problem*.
- If the conditions on the problem are at two different values for the independent variable, usually at the boundaries of some region of interest, it is called a *boundary-value problem*.

$$\begin{aligned} y'' + y' + y &= f(t) \\ y(a) &= \alpha \\ y'(a) &= \beta \\ t &\in [a, b]. \end{aligned}$$

$$\begin{aligned} y'' + y' + y &= f(t) \\ y(a) &= \alpha \\ y(b) &= \beta \\ t &\in [a, b], \end{aligned}$$

Identify (IVP or BVP)

- Prob 1: $y''-3y'+2y=4e^x$ subject to $y(0)=2$ and $y'(0)=-1$
- Prob 2: $\frac{dy}{dt} = 4e^{0.8t} - 0.5y$ subject to $y(t=0) = 2$
- Prob 3: $\frac{d^2y}{dx^2} - y = 0$ subject to (i) $\frac{dy}{dx} = 0$ at $x = 0$ and (ii) $y = 1$ at $x = 1$

Finite-difference approximations

- Finite-difference approximations provide a means to transform derivatives into algebraic form

Forward Difference Formula (1st order accurate)

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Backward Difference Formula (1st order accurate)

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

Centered Difference Formula (2nd order accurate)

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Centered Difference Formula for 2nd Derivative (2nd order accurate)

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Euler's method (IVP)

- Based on forward difference
- Consider the following ODE

$$\frac{dy}{dt} = f(t, y)$$

- Let us take any time t_i, corresponding y is y_i. The ODE should be valid at that point (t_i, y_i)

$$\left.\frac{dy}{dt}\right|_{(t_i, y_i)} = f(t_i, y_i)$$

- *Taking* finite-difference approximation of the derivative at time t_i

$$\frac{y(t_{i+1}) - y(t_i)}{t_{i+1} - t_i} = f(t_i, y_i)$$

- Let $t_{i+1} - t_i = \Delta t$

$$\frac{y(t_{i+1}) - y(t_i)}{\Delta t} = f(t_i, y_i)$$

- Rearranging, we get

$$y(t_{i+1}) = y(t_i) + f(t_i, y_i).\Delta t$$

Example

$$\frac{dy}{dt} = 4e^{0.8t} - 0.5y$$

Initial condition, $y(t = 0) = 2$

- Solve the above ODE using Euler method from t = 0 to 4 with a step size of 1.

- Compare the solution with the analytical solution given below

$$y = \frac{4}{1.3}(e^{0.8t} - e^{-0.5t}) + 2e^{-0.5t}$$

Taylor series expansion

- A Taylor series is a series expansion of a function about a point.
- A one-dimensional Taylor series is an expansion of a real function $g(t)$ about a point $t = a$ is given by

$$g(t) = g(a) + g'(a)(t-a) + \frac{g"(a)}{2!}(t-a)^2 + \frac{g^{(3)}(a)}{3!}(t-a)^3 + \cdots + \frac{g^{(n)}(a)}{n!}(t-a)^n + \cdots$$

Derivation of Euler & Runge-Kutta methods

- Consider the following ODE

$$\frac{dy}{dt} = f(t, y)$$

- Let us say that the solution of this ODE is

$$y = g(t)$$

- Taylor series expansion of $g(t)$ about point $t = t_i$

$$g(t) = g(t_i) + g'(t_i)(t - t_i) + \frac{g"(t_i)}{2!}(t - t_i)^2 + \frac{g^{(3)}(t_i)}{3!}(t - t_i)^3 + \cdots + \frac{g^{(n)}(t_i)}{n!}(t - t_i)^n + \cdots$$

- As a numerical solution, we are interested to find y value (or $g(t)$) at time $t = t_{i+1}$.

$$g(t_{i+1}) = g(t_i) + g'(t_i)(t_{i+1} - t_i) + \frac{g"(t_i)}{2!}(t_{i+1} - t_i)^2 + \frac{g^{(3)}(t_i)}{3!}(t_{i+1} - t_i)^3 + \cdots + \frac{g^{(n)}(t_i)}{n!}(t_{i+1} - t_i)^n + \cdots$$

- As $y = g(t)$ and $\frac{dy}{dt} = f(t, y)$, we can write

$$g'(t_i) = f(t_i, y_i),\ g"(t_i) = f'(t_i, y_i)$$

$$g(t_{i+1}) = y_{i+1},\quad g(t_i) = y_i$$

- Let us define $t_{i+1} - t_i = h$

- Incorporating the above, we can write

$$y_{i+1} = y_i + f(t_i, y_i)h + \frac{f'(t_i, y_i)}{2!}h^2 + \frac{f"(t_i, y_i)}{3!}h^3 + \cdots + \frac{f^{(n)}(t_i, y_i)}{n!}h^n + \cdots$$

Euler's method

- If we consider the first two terms only

$$y_{i+1} = y_i + f(t_i, y_i)h$$

- This is Euler's method; it is referred to as Runge-Kutta 1st order method.

- The true error in the approximation is given by

$$\frac{f'(t_i, y_i)}{2!}h^2 + \frac{f''(t_i, y_i)}{3!}h^3 + \cdots + \frac{f^{(n)}(t_i, y_i)}{n!}h^n + \cdots$$

Runge-Kutta 2nd order method

- If we consider the first three terms

$$y_{i+1} = y_i + f(t_i, y_i)h + \frac{f'(t_i, y_i)}{2!}h^2$$

- Example: $\frac{dy}{dt} = e^{-2t} - 3y,\ y(0) = 5$

$$f'(t,y) = \frac{df(t,y)}{dt} = \frac{\partial f(t,y)}{\partial t} + \frac{\partial f(t,y)}{\partial y}\frac{dy}{dt}$$

*Need to find the derivative of $f(t_i, y_i)$ symbolically

*Calculation of $f'(t_i, y_i)$ can be challenging sometimes

Runge-Kutta 2nd order method

Runge-Kutta proposed that

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$

where

$$k_1 = f(t_i, y_i)$$
$$k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h)$$

- This form allows one to take advantage of the 2nd order method without having to calculate $f'(t_i, y_i)$
- However, there are 4 unknown constants, i.e., a_1, a_2, p_1, q_{11}

- Going back to the Taylor series expansion with three terms

$$y_{i+1} = y_i + f(t_i, y_i)h + \frac{f'(t_i, y_i)}{2!}h^2$$

- We know that $f'(t,y) = \frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt}$ thus

$$y_{i+1} = y_i + f(t_i, y_i)h + \frac{1}{2!}h^2\left[\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt}\right]_{(t_i,y_i)}$$

- As $\frac{dy}{dt} = f(t,y)$, we can write

$$y_{i+1} = y_i + f(t_i, y_i)h + \frac{1}{2!}h^2\left.\frac{\partial f}{\partial t}\right|_{(t_i,y_i)} + \frac{1}{2!}h^2\left.\frac{\partial f}{\partial y}\right|_{(t_i,y_i)} f(t_i, y_i)$$

- We have written $k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h)$
- Let us revise Taylor series expansion of $f(x)$ around the point $x = a$ (for a single variable)

$$f(x) = f(a) + f'(a)(x-a) + \frac{f"(a)(x-a)^2}{2!} + \cdots$$

- Similarly, Taylor series expansion of $f(t, y)$ around the point (t_i, y_i) (for two variables)

$$f(t,y) = f(t_i, y_i) + \left.\frac{\partial f}{\partial t}\right|_{(t_i,y_i)} (t - t_i) + \left.\frac{\partial f}{\partial y}\right|_{(t_i,y_i)} (y - y_i) + \cdots$$

- Taking $t = t_i + p_1 h$ and $y = y_i + q_{11} k_1 h$, we can write

$$k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h) = f(t_i, y_i) + \left.\frac{\partial f}{\partial t}\right|_{(t_i,y_i)} p_1 h + \left.\frac{\partial f}{\partial y}\right|_{(t_i,y_i)} q_{11} k_1 h + \cdots$$

- The solution has been proposed as

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$

- Substituting for k_1 and k_2

$$k_1 = f(t_i, y_i)$$

$$y_{i+1} = y_i + \left[a_1 f(t_i, y_i) + a_2 \left\{ f(t_i, y_i) + \left.\frac{\partial f}{\partial t}\right|_{(t_i, y_i)} p_1 h + \left.\frac{\partial f}{\partial y}\right|_{(t_i, y_i)} q_{11} k_1 h \right\} \right] h$$

$$y_{i+1} = y_i + (a_1 + a_2) f(t_i, y_i)h + a_2 p_1 \left.\frac{\partial f}{\partial t}\right|_{(t_i, y_i)} h^2 + a_2 q_{11} \left.\frac{\partial f}{\partial y}\right|_{(t_i, y_i)} f(t_i, y_i)h^2$$

- Comparing

$$y_{i+1} = y_i + (a_1 + a_2) f(t_i, y_i) h + a_2 p_1 \left.\frac{\partial f}{\partial t}\right|_{(t_i, y_i)} h^2 + a_2 q_{11} \left.\frac{\partial f}{\partial y}\right|_{(t_i, y_i)} f(t_i, y_i) h^2$$

$$y_{i+1} = y_i + f(t_i, y_i) h + \frac{1}{2!} h^2 \left.\frac{\partial f}{\partial t}\right|_{(t_i, y_i)} + \frac{1}{2!} h^2 \left.\frac{\partial f}{\partial y}\right|_{(t_i, y_i)} f(t_i, y_i)$$

- We can get values the following equations

$$a_1 + a_2 = 1$$
$$a_2 p_1 = 1/2$$
$$a_2 q_{11} = 1/2$$

3 equations, 4 unknowns -> infinite choices of a_1, a_2, p_1, q_{11}

One of the solutions is
$a_1 = a_2 = 1/2,\ \ p_1 = q_{11} = 1$

Substituting the suggested values of constants, we can define 2nd order Runge-Kutta formulae as

$$y_{i+1} = y_i + \frac{h}{2}(k_1 + k_2)$$

where

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + h, y_i + k_1 h)$$

Runge-Kutta 4th order method

$$\frac{d\boldsymbol{y}}{dt} = \boldsymbol{f}(t, \boldsymbol{y})$$

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\left(\frac{\boldsymbol{k_1}}{6} + \frac{\boldsymbol{k_2}}{3} + \frac{\boldsymbol{k_3}}{3} + \frac{\boldsymbol{k_4}}{6}\right)$$

$$\boldsymbol{k}_1 = \boldsymbol{f}(t_i, \boldsymbol{y}_i)$$
$$\boldsymbol{k}_2 = \boldsymbol{f}\left(t_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{k}_1\right)$$
$$\boldsymbol{k}_3 = \boldsymbol{f}\left(t_i + \frac{h}{2}, \boldsymbol{y}_i + \frac{h}{2}\boldsymbol{k}_2\right)$$
$$\boldsymbol{k}_4 = \boldsymbol{f}(t_i + h, \boldsymbol{y}_i + h\boldsymbol{k}_3)$$

Example

- Solve

$$\frac{dy}{dt} = 4e^{0.8t} - 0.5y$$

- Initial condition, $y(t = 0) = 2$

- Solve this problem by Euler method, RK-2, and RK-4 and compare the solutions.

Example

Let us consider a simple isothermal reaction $A \rightarrow B$ taking place in a MFR with constant holdup V. The kinetics of the reaction is given by second order reaction. The material balance for the reactor is written as

$$V \frac{dC_A}{dt} = F\left(C_{AF} - C_A\right) - k\, C_A^2\, V$$

where C_{AF} is the concentration of A in the reactor inlet stream (mol/m^3), C_A is the instantaneous concentration of A in the reactor (the exit stream composition is equal to the reactor composition for an ideal MFR), F is the inlet flow rate (also the effluent flow rate) in m^3/min, k is the reaction rate constant in m^3/(mol*min) and V is the reactor holdup in m^3. The nominal values are: F = 9 m^3/min, C_{AF} = 5 mol/m3, k = 2 m^3/(mol*min) and V = 1 m^3. If the feed concentration changes from 5 mol/m^3 to 6 mol/m^3, how does the concentration of A in the reactor evolve with time?

Example continued…

- Due to some constraints, it is required to reduce the effluent concentration of species A below 1 mol/m^3. The operation team suggests the use of a train of MFRs of similar size (same holdup). How many reactors will be needed to achieve the desired operation?
- For the above train of MFRs, Determine how the concentration of reactant A changes in the N^{th} reactor when the feed concentration to the first reactor changes from 5 to 6 mol/m^3.

Indian Institute of Technology Roorkee

CHN-323
Computer Applications in Chemical Engineering

Ashwini Kumar Sharma

Department of Chemical Engineering
Indian Institute of Technology Roorkee
Email: ashwini.fch@iitr.ac.in

Acknowledgements: Prof. Laksh, NUS

Example 1

An object is being projected upward at a specified velocity. It is subject to linear drag and its altitude as a function of time can be computed as

$$z = z_0 + \frac{m}{c}\left(v_0 + \frac{mg}{c}\right)\left(1 - e^{-(c/m)t}\right) - \frac{mg}{c}t$$

where

z = altitude (m) above the earth's surface (defined as $z = 0$)

t = time (s)

z_0 = the initial altitude (m) =100 m

m = mass (kg) =80 kg

c = a linear drag coefficient (kg/s) = 15 kg/s

v_0 = initial velocity (m/s)=55 m/s

Plot z vs t till the object returns to the ground.

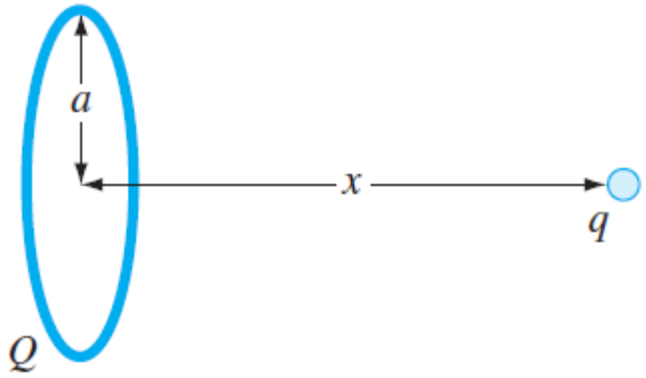
Find the time at which the maximum elevation is achieved?

- With the graph
- With optimization principles

Example 2

A total charge Q is uniformly distributed around a ring-shaped conductor with radius a. A charge q is located at a distance x from the center of the ring. The force exerted on the charge by the ring is given by

$$F = \frac{1}{4\pi e_0}\frac{qQx}{(x^2 + a^2)^{3/2}}$$



where $e_0 = 8.85 \times 10^{-12}\ C^2/(N.m^2)$, $q = Q = 2 \times 10^{-5}\ C$, and $a = 0.9$ m.

Determine the distance x where the force is a maximum.

Example 3

A train (series) of four well mixed reactors operate isothermally. The species whose concentration is designated by c reacts according to the mechanism: r = -kc^n in each tank. The fluid flow rate is at a fixed value q. Each tank has a volume V_i (i = 1, ..., 4).

The material balance for any reactor i is written as

$$\frac{d(V_i\, c_i)}{dt} = q\, c_{i-1} - q\, c_i - V_i\, k\, c_i^n$$

Determine the volume of each reactor such that the *steady state* yield of the product is maximum. The total volume of the four tanks is 20 m^3. The values of the parameters and operating variables are n = 2.5, q = 71 m^3/h, inlet concentration into first reactor c_0 = 20 kg mol/m^3 and k = 6.25×10^{-3} [m^3/kg mol]$^{1.5}$(s)$^{-1}$.

Minimize: outlet reactant concentration (or maximize product concentration)

w.r.t v_i, i=1,..,4

Bounds: $v_i > 0$

Constraints: $\sum v_i = 20$

A clothing manufacturer earns a profit of $10 on a T-shirt and $15 on a sweater. Producing a T-shirt requires 2 minutes on the cutting machine and 1 minute on the stitching machine, while a sweater requires 2 minutes of cutting and 4 minutes of stitching. How many T-shirts and sweaters should be produced per hour to maximize profit?

- The decision variables are the number of T-shirts (x) and the number of sweaters (y) to produce per hour. The manufacturer earns $10 per T-shirt, so the total earned from this product type if x are made is $10x$. Similarly, a profit of $15 per sweater on y units produced is a total of $15y$.
- The target function is the total profit, across both product types:

$$f(x, y) = 10x + 15y$$

- Since x and y count items of clothing, we can assume the implicit constraints $x, y \geq 0$.

- The maximum number of items that can be produced is also limited by the time available on the cutting and stitching machines. During a one-hour period, meaning 60 minutes, if x T-shirts and y sweaters all take up 2 minutes each on the cutting machine, the total length of time the machine is running is

$$2x + 2y \leq 60$$

- Similarly, since T-shirts take 1 minute and sweaters take 4 minutes on the stitching machine, the total length of time this machine is running per hour is

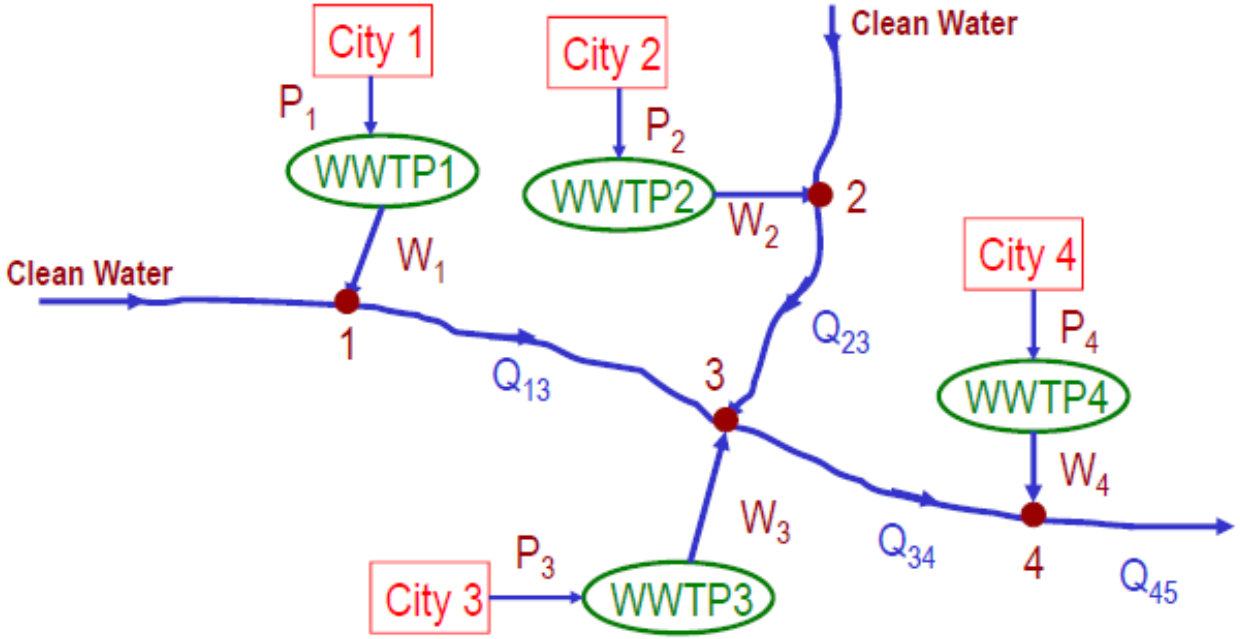
$$x + 4y \leq 60$$

- The optimization problem can be restated in full as follows:

$$\text{maximize} \quad f(x,y) = 10x + 15y \quad \text{subject to} \quad \begin{cases} x, y \geq 0 \\ 2x + 2y \leq 60 \\ x + 4y \leq 60 \end{cases}$$

Example 3

Minimize the cost of waste-water treatment in a city consortium while maintaining the pollutant concentration in any stream not greater than 20 mg/L.



WWTP – Waste Water Treatment Plant

P_i = waste generated by City i (mg / day); i = 1, 2, 3, 4

W_i = waste discharged by City i (mg / day); i = 1, 2, 3, 4

x_i = fraction waste removed by WWTPi; i = 1, 2, 3, 4

c_i = concentration of pollutant at junction i (mg / L); i = 1, 2, 3, 4

Q_{ij} = Vol. flow rate between junction i and junction j (L / day)

R_{ij} = fraction of pollution remaining as the river flows downstream from junction i to junction j (pollutant level is reduced due to chemical & biological decomposition processes)

d_i = cost of waste treatment in WWTPi ($/mg)

City	P_i (mg/day)	d_i ($10^{-6}/mg)	Segment	Q (L/day)	R
1	$1x10^9$	2	1-3	$1x10^7$	0.5
2	$2x10^9$	2	2-3	$5x10^7$	0.35
3	$4x10^9$	4	3-4	$11x10^7$	0.6
4	$2.5x10^9$	4	4-5	$25x10^7$	

Solution

Total cost of wastewater treatment in a day

$$\text{Cost} = d_1 P_1 x_1 + d_2 P_2 x_2 + d_3 P_3 x_3 + d_4 P_4 x_4$$

Pollutant concentrations at every junction

$$c_1 = \frac{(1 - x_1) P_1}{Q_{13}}$$

$$c_2 = \frac{(1 - x_2) P_2}{Q_{23}}$$

$$c_3 = \frac{Q_{13} R_{13} c_1 + Q_{23} R_{23} c_2 + (1 - x_3) P_3}{Q_{34}}$$

$$c_4 = \frac{Q_{34} R_{34} c_3 + (1 - x_4) P_4}{Q_{45}}$$

Example: fmincon

- $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ **Rosenbrock function**

- Find the minimum value starting from the point [-1,2], constrained to have $x_1 + 2x_2 \leq 1$.
- Express this constraint in the form Ax <= b by taking A = [1,2] and b = 1.
- Notice that this constraint means that the solution will not be at the unconstrained solution (1,1), because at that point $x_1 + 2x_2 = 3 > 1$.

- fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;

- x0 = [-1,2];
- A = [1,2];
- b = 1;
- x = fmincon(fun,x0,A,b)

- $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

- Find the minimum value starting from the point [0.5,0], constrained to have $x_1 + 2x_2 \leq 1$ and $2x_1 + x_2 = 1$.

- Express the linear inequality constraint in the form A*x <= b by taking A = [1,2] and b = 1.

- Express the linear equality constraint in the form Aeq*x = beq by taking Aeq = [2,1] and beq = 1.

- fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
- x0 = [0.5,0];
- A = [1,2];
- b = 1;
- Aeq = [2,1];
- beq = 1;
- x = fmincon(fun,x0,A,b,Aeq,beq)