PETR 5313: CRN 38950, Fall 2017
Numerical Application in Petroleum Engineering,
Lesson 00: Syllabus

Ekarit Panacharoensawad, PhD
Terry Fuller Petroleum Engineering Research Building Room 236
ekarit.panacharoensawad@ttu.edu

Cite as: Panacharoensawad, E. (2017) "Numerical application in petroleum engineering, Lesson 01: Introduction", presentation slide for Texas Tech PETR 5313 Fall 2017

Safety moment

- Emergency exit
- Tornado
- Fire
- Active shooter

Course Introduction

Objective

- After this course, you should be able to understand the numerical method behind various algorithm and implement them properly.

Importance of the numerical method

- Need for conducting research
- Need for complicated calculation
- Allow the improvement of the existing algorithm
- Programming becomes a must in the future world

Course Contents

- Basic Programming
  - Summation calculation (Excel / VBA / numpy / python)
  - Print number 1 to 100 to screen / file / Excel / Excel VBA / python / jupyter notebook, LibreOffice coding
  - Matrix multiplication (in Excel, LibreOffice, jupyter notebook)
  - Symbolic computation (python with sympy)
  - Quick integration with sympy
  - Ei function calculation with sympy
  - List comprehension
  - From dictionary to series and dataframe (python: pandas)

Course Contents...

- Basic Data Visualization
  - Automated data visualization (python: matplotlib)
  - Contour plot
  - 2D heat map of a user defined function
  - Quick look at seaborn
- Root of non-linear equation
  - Bi-section method
  - Newton method

Course Contents...

- System of linear equations
  - AX = b
  - A is a matrix
  - Usage of library
- System of non-linear equation
  - Linearization with Jacobian
- Linear Regression
  - Underfit / Overfit, regression with polynomial
  - Lasso regularization
  - Manually / Excel / python library

Course Contents...

- Interpolation
- Integration
  - Trapezoidal, Simpson'rules
- Differentiation
  - 2, 3, 4, 5, .. methods
- Ordinary Differential Equation (ODB)
  - Explicit Euler, Implicit Euler, RK4, RKF45
- Partial Different Equation (PDE)
  - Elliptic PDE (Poisson equation)
  - Parabolic PDE (Heat equation)

Course Contents...

- Optimization 1D and n-D problems
  - Gradient Descent
  - Levenberg Marquardt (LM)
  - LM for a black-box model
    - Finding fitting parameters in model / non-linear equation

Weekly Schedule (we have 15 weeks, roughly)

- W1: Basic Programming
- W2: Basic Programming
- W3: Basic Data Visualization
- W4: Root finding (1 unknown)
- W5: System of linear equations
- W6: Exam 1 (close-book with 2 pages (1side each) info sheet)
- W7: System of non-linear equations
- W8: Regression
- W9: Interpolation / Integration

Weekly Schedule...

- W10: Differentiation / ODE
- W11: Exam 2 (close-book with 4 pages (1side each) info sheet). The first 2 pages of info sheet can be from the Exam 1.
- W12: PDE
- W13: PDE
- W14: Optimization
- W15: Course Project

Detailed Schedule

- 11 - 12:20 MW, room 204 Aug 28 - Dec 13
- Contact hours = 45 hours (face to face / exam)
- No classes day
  - Labor day: Sep 4
  - Thanksgiving: Nov 22 - 26
- Last day to add a course: Aug 31
- No exam (except makeup/lab): Nov 30 - Dec 6
- Last day of class: Dec 6

Project: Programming Challenge

- Individual assignment
- Start 11/30/17 after the last lecture
- Deadline = Dec 13 before midnight

This programming challenge project consists of each and every parts in this course. You may think about it as a 20% homework, but this time, F is given for cheating (plus contact the Office of Student Conduct).

This project needs about 18 hours to do. 18 hours without an instructor is equivalent to 6 contact hours of lecture.

Date	Subject	contact hour	accumulative contact hours	Hw assigned	Hw due
08/28/17	Basic Prog	1.5	1.5		
08/30/17	Basic Prog	1.5	3		
09/04/17	no class	0	3		
09/06/17	Basic Prog	1.5	4.5		
09/11/17	Basic Prog	1.5	6	1: B Prog	
09/13/17	Basic DatVis	1.5	7.5		
09/18/17	Basic DatVis	1.5	9	2: DatVis	1
09/20/17	Root finding	1.5	10.5		
09/25/17	Root finding	1.5	12	Exam1 Prep	2
09/27/17	Sys Lin	1.5	13.5		
10/02/17	Sys Lin	1.5	15		
10/04/17	Exam 1	1.5	16.5	3: Root find	
10/09/17	Sys nonLin	1.5	18		
10/11/17	Sys nonLin	1.5	19.5	4: Sys lin	3
10/16/17	Regress	1.5	21		
10/18/17	Regress	1.5	22.5	5: nonLin	4

Date	Subject	contact hour	accumulative contact hours	Hw assigned	Hw due
10/23/17	IntPol/Integr	1.5	24		
10/25/17	IntPol/Integr	1.5	25.5	6: Regress	5
10/30/17	ODE	1.5	27		
11/01/17	ODE	1.5	28.5	Exam2 Prep	6
11/06/17	Exam 2	1.5	30		
11/08/17	PDE	1.5	31.5	7: intPol/Intg	
11/13/17	PDE	1.5	33		
11/15/17	PDE	1.5	34.5	8-9: O/PDE	7
11/20/17	PDE	1.5	36		
11/22/17	no class	0	36		
11/27/17	Optimize	1.5	37.5		
11/29/17	Optimize	1.5	39		
12/04/17	Project/no class	1.5	40.5	Project	8-9
12/06/17	Project/no class	1.5	42		
12/11/17	Project/no class	1.5	43.5		
12/13/17	Project/no class	1.5	45		Project

Requirement / Textbook

- Windows / Ubuntu / Linux / OSX (Ubuntu is encouraged)
- Calculator: Use during exam.
  - Programmable calculator is recommended ($42.00)
- PC/laptop ($600 - $1500). Otherwise, you need to have access to a computer with Excel + Python. Python via Citrix can be used but it is not recommended.

Required Textbook: None

Reference Textbook

- Burden, R. Faires, D.J., Burden, A. (2015) “Numerical Analysis”, 10th Ed., Cengage Learning
- Robert Johansson, "Numerical Python: A Practical Techniques Approach for Industry", APress ($25.00 via MyCopy SpringerLink)
- Chapra S.C., Canale, R.P. "Numerical Methods for Engineers", 7th Ed. McGrawHill
- Beazley D., Jones, B.K. "Python Cookbook", 3rd Ed. O'Reilly
- Python documentation https://docs.python.org/3/download.html
- Numpy documentation https://docs.scipy.org/doc/numpy/numpy-user-1.13.0.pdf
- Scipy documentation https://docs.scipy.org/doc/scipy/scipy-ref-0.19.1.pdf
- Sympy documentation https://github.com/sympy/sympy/releases/download/sympy-1.1.1/sympy-docs-pdf-1.1.1.pdf
- Pandas documentation https://pandas.pydata.org/pandas-docs/stable/pandas.pdf
- Matplotlib documentation https://matplotlib.org/Matplotlib.pdf

Syllabus Highlights (total % = 110)

- Exam 1: 10% Exam 2: 10% Quiz: 10%
- Programming Project: 20% Homework: 60%
- Class attendance (lecture + exam): mandatory*
  - Miss 0 time without the university excuse = get A+ or lower
  - Miss 1 times without the university excuse = get A or lower
  - Miss 2 times without the university excuse = get A or lower
  - Miss 3 times without the university excuse = get A- or lower
  - Miss 4 times without the university excuse = get B+ or lower
  - See syllabus for 6+

*** This policy starts after the last day to add the course**

Late for Class / Quiz policies

- Late between 5 to 30 minutes = 1 Late for class
- 3 Lates = 1 missing class
- Late more than 30 minutes of 1 class = missing class
  - You are still allowed to come in and study
- Late less than 5 minutes is not considered as late
- **Late policy starts after the last day to add the course**
- **No quiz before the last day of adding the course**
- With the university excuse, missing quiz score will be excluded from the grade calculation only if you submit the quiz answer (otherwise, get zero).

Missing Homework Policy

- Each homework take about 0-3 hours to review the material and 3-6 hours to do it (roughly 6 hours totally)

In addition to getting zero for any missing homework,

- missing 1 assignment = Get A or lower (cannot get A+)
- missing 2 assignments = Get B+ or lower (cannot get A- or above)
- missing 3 assignments = Get C+ or lower (cannot get B- or above)
- missing 4 assignments = Get D+ or lower (cannot get C- or above)
- missing 5 assignments or more = F

Assignment means homework + project.

Missing Class with the University Excuse

For this case, it is your responsibility to submit the report on time and conform to the requirement of the report. If you fail to submit the report on time or your report get the "fail" grade, you will be considered as missing from that day of class, and it is counted toward the penalty of the letter grade (previous slide).

More details on what kind of report, deadline, etc to substitute for missing the class with the university excuse can be found on the syllabus.

Missing Exam

- Get 0 (zero) for missing an exam without the university excuse
- Get the grade of "i" (incomplete) for missing any exam with the university excuse.
- The correction for the grade of "i" can be done according to the university policy (after December)
- Student who submit the intent to graduate in this semester and miss the exam with the university excuse will be allowed to take the makeup exam. Other students will get the grade of "i"
- Specific requirements on the makeup exam deadline is in the syllabus

Copyleft Policy of this Course

- This course is made with the copyleft policy. You / anyone can make / adjust any part of the material / homework / exam to the full extent with no need to ask for my permission (if I have the right in such material). For the code / example / etc that is from other sources, it is your responsibility to obtain the 'right to copy' in the right way.

Copy Homework Policy

- If you found responsible for copying homework / let other people copying your work, you will get
- minus 5 points of the total grade for the first attempt and the report to the Office of Student Conduct
- The academic penalty (not related to the grade) will be at the discretion of the Office of Student Conduct
- For the second attempt you get "F" and the report to the Office of Student Conduct

Other Policy

Please see the syllabus for other policy

- Late for Class Policy
- Cheating on Exam
- Use other people programming code without citation
- Student Disability Services
- Video Recording
- Voice Recording

Before We Start (academic integrity)

- Why you as a student should care if other student cheat on the exam?
- Why you as a student should care if other student cheat on the homework?
- What is the impact to the university if we let the one who cheat graduate?
- What is the impact on the value of your degree, if someone who know nothing but pass by cheating get the same degree?

Before We Start (who care about this)

- How can numerical method and programming (or calculation) help me to get paid more than a technician with 20 years of experience?

- What if, what I learned has never appeared again when I work?

- Do I need to know programming already before taking this course?
- No you don't need to. Just do homework, it will be fine.

What to learn and why?

- We will learn the numerical method, the algorithm behind the method, how to make such algorithm, and how to use the algorithm from libraries.
- If we already have functions ready-to-go, why do we care about making it ourself? do we re-invent the wheel?
- We need to know to the point where we can
  - Use it, Make it, Modify it, Improve other people code/procedure so we know it completely / in-depth (if not complete).
  - It is not about re-inventing the wheel, it is about training you to completely know the material (or most of it)

Numerical method Application Examples

- Calculation of Ei function used in reservoir engineering
- Basic in reservoir simulation / history matching calculation
- Hydrodynamics calculation of drilling fluid
- Multiphase flow hydrodynamics calculation
- Heat transfer calculation in subsea pipeline
- Solving various governing equation in ODEs form
- And... more for any simulation that you may have seen already.
- Required for furture study in Finite Element / Finite Volume / Etc.

What to learn and why?..

- Needed for doing PhD / MS research efficiently / in many cases
- Especially for experimentalist, doing experiments just for the sake of doing it is the wrong thing to do and expensive. A less expensive numerical simulation / calculation is needed to guide the direction of the experiment
- **Getting MS/PhD without knowing any programming just does not fit with the future world and you don't want that, in my opinion.**

Before We Start: Python Installation

This course use Python 3.6

- Option A: Windows > Anaconda > Jupyter notebook
- Option B: Windows > Ubuntu on Virtual Machine > Anaconda > Jupyter notebook
- Option C: Ubuntu > Anaconda > Jupyter notebook
- Option D: Mac OS > Anaconda > Jupyter notebook
  - https://docs.continuum.io/anaconda/install/mac-os
- https://docs.continuum.io/anaconda/install/
- http://jupyter.readthedocs.io/en/latest/install.html

Before We Start: Python Installation

Option E: Use it via Citrix (Not recommend, slow and very slow)

- Some libraries that are needed may not be available.
- We need
- sklearn PIL (pillows)
- matplotlib scipy pandas
- numpy sympy seaborn
- Citrix does not have Python 3.6 (only 3.4 and 2.7)
- Frequently, you need an administer privilege to install require library / package (may be able to get around but not convenient)

Before We Start: Python Installation ...

Windows/Anaconda

- Some less popular libraries are difficult to be installed on , (you still can but need to google a lot, a lot of headache)
- Try this one first (add a new library will work, most of the time)

Dual OS: Windows/Ubuntu is possible

  - Take several hours to do so: Backup -> use Gparted to re-partition the drive -> install Ubuntu -> take some time to get used to the system / terminal
  - A lot less headache when import new library + Free OS

Windows with virtual machine Ubuntu: trade the performance for the quick installation of python library