Course Syllabus
ASTR 21100
Winter 2019
Computational Techniques in Astrophysics
Classes: Wed, Fri 13:30-14:50, KPTC 309

Instructor: Andrey Kravtsov

Contact info: office=ERC 415, campus phone = 2-4249, email = kravtsov@uchicago.edu

Office hours: Tue, 10:30-11:30am (I will notify you if this changes in certain weeks)

Teaching assistants: Cory Cotter (cotter@astro.uchicago.edu),
Dimitrios Tanoglidis (dtanoglidis@uchicago.edu)

TA office hours: Cory Cotter: Tuesday 3-4pm, Hale room on the 5th floor of ERC
Dimitrios Tanoglidis: Monday, 10-11am, Hale room on the 5th fl ERC

Pre-requisites: familiarity with basic calculus and working knowledge of python. We will be using Jupyter notebooks for class materials and assignments.

Your laptop and power cord are required in each class. If you don't have a laptop available before a particular class, you can borrow one from the TechBar at the Regenstein Library.

Reading materials: there is no required textbook, relevant material will be distributed and/or presented in class. Check the website above for the general list of useful materials.

Website via which class materials and assignments will be distributed (you will need to sign up for a github account, if you don't have one): https://github.com/a-kravtsov/a211w19

Slack workspace: https://a211w19.slack.com

Slack workspace will be used for communications between you and instructor and TAs outside classes and during group work in classes.

Important: when you create your github and slack accounts, use your CNetID as your user name, if possible. This will make it easier for us to recognize you.

Class format: each class will start with a review of assignment results and introduction of new concepts and material (approximately half of the class time). The second half of the class will be dedicated to group work in groups of 3-4 students on the current assignments, assisted by instructor and TA. We will assign you to a group and groups will be shuffled every couple of weeks. Although you will work in groups in class, you will need to submit your assignment work individually.

Assignments: this is a heavily practice-based class and there will be assignments almost every week of the course. Assignments will come in two flavors: exercises that require coding, computation, and analyses, and small sets of conceptual questions. The former will be distributed on Wednesdays and due the following Tuesday at 5pm and will contribute to 70% of your class grade. The exercises will have points associated with them, which will count towards the total point count for the entire assignment. Grading will be based on how fully an exercise is completed. Partial points will be given for partly completed exercise.

The sets of questions will be distributed on Wednesday and due on Thursday at 9pm and will contribute to 20% of your class grade.

Late policy: the format of the course relies on prompt reviews of the assignments. Thus, no late submissions will be accepted.

Class attendance and participation: we expect you to attend every class. *10% of your grade will be based on your participation in the in-class group work and activities.* If you will have a serious reason (e.g., illness), please contact course instructor and TA well ahead of time to arrange to be excused.

Coding style and good practices: we will discuss good coding style and practices in class (see here for useful guidelines). We expect you to attempt to follow good coding practices and may take points off for code that violates style guidelines and is especially difficult to read.

Exams: there will be no exams.

Honor code: during group discussions you will discuss strategies to do a particular exercise or calculation or how to write a code. However, we expect you to then personally write code and obtain your own results, which you will submit as your individual assignment.

Topics to be covered:

This is a tentative list of topics we will be covering in this class, approximately in the order in which they will be introduced and discussed. Most topics will be introduced via practical exercises that will illustrate their use in specific realistic astronomical applications. Many of the applications will use more than a single method and thus we will discuss multiple methods when we consider a particular application. At the same time, certain methods will be encountered multiple times in different applications.

Although not listed explicitly, certain practical methods and topics, such as floating point accuracy, truncation errors, coding style and good practices, I/O for ASCII, spreadsheet tables, and FITS files and dataset manipulation, plotting and data visualization, debugging, code testing, useful data structures, object-oriented programming, creating software projects in python, etc. will also be introduced and discussed as part of specific exercises.

Approximation of functions: interpolation, least squares approximation using analytic functions (or series of functions), approximation/interpolation using a series of orthogonal basis functions, Fourier transform and FFT, wavelets.

Reconstruction of functions based on sampled point distribution: histograms and adaptive histograms, kernel density estimation. Smoothing noisy data.

Numerical integration methods. Romberg integration.

Monte Carlo methods. Methods for sampling random distributions in low dimensions. Estimation of uncertainties using bootstrap method.

Markov Chain Monte Carlo (MCMC) method for sampling distributions in many dimensions. MCMC chain convergence criteria. Overview of other modern MCMC methods and codes.

Using MCMC for modelling data using Bayesian framework. Finding credible intervals for the model parameters from the MCMC-sampled posterior pdf. Numerical estimates of Bayesian evidence for model comparison.

Ordinary differential equations: overview of explicit and implicit methods and their optimal applications. Adaptive time stepping. Symplectic integration methods.

Machine learning methods and their uses overview. Clustering and classification algorithms.

Wed, Jan 9. Intro. The meaning of numerical computing. Solving equations by bisection method.

Fri, Jan 11. Reconstructing functions from samples. Kernel Density Estimates (KDE).

Wed, Jan 16. Reconstructing functions and KDE, contd. KDE in 2D.

Fri, Jan 18. Polymomial interpolation. Barycentric interpolation method

Wed, Jan 23. Piecewise interpolation with splines. Polynomial regression using Gram-Schmidt procedure. Savitzky-Golay smoothing.

Fri, Jan 25. Trigonometric interpolation.

Wed, Jan 30. Class cancelled due to extremely cold temperatures.

Fri, Feb 1. Discrete Fourier Transform. FFT.

Wed, Feb 6. FFT contd.

Fri, Feb 8. Applications of FFT. Spectral analysis. Wiener filtering. Lomb-Scargle periodogram.

Wed, Feb 13. Monte Carlo method. Pseudo-random number generators.

Fri, Feb 15. College break.

Wed, Feb 20. Sampling methods 1d. MCMC.

Fri, Feb 22. Using MCMC in Bayesian analyses to sample posteriors and model comparisons.

Wed, Feb 27. Numerical integration

Fri, Mar 1. Numerical methods of ODE solution. Explicit methods. Runge-Kutta (RK) methods.

Wed, Mar 6. Numerical methods of ODE solution (contd.) Stiff ODEs, implicit methods.

Fri, Mar 8. Machine learning: unsupervised learning, clustering.

Wed, Mar 13. Machine learning: supervised learning. Classification methods.

Fri, Mar 15. Deep learning methods.

Mon, Mar 18. 1:30-3:30pm (location TBD). Meet to present results of the final assignment/project (this is the official time for the final exam of this class).