Class 28. Topics Not Covered

- A one-semester course is not nearly long enough to touch on all the variety of topics in computational astrophysics, nor to go into any great detail on very many of them.
- Below is an incomplete list of topics we did not get a chance to cover that are of possible interest. Hopefully the list will reveal new ideas not previously considered, or at least serve as a starting point for trying out other computational courses.

Algorithms

Sorting

- Cf. *NRiC* §8.
- Well-known algorithms:
  1. Bubble Sort — $\mathcal{O}(N^2)$
  2. Quicksort — $\mathcal{O}(N \log_2 N)$
     - Recursive method (divide and conquer).
     - Implemented on most Unix systems as `qsort()`:
       ```
       void qsort(void *base,size_t nmemb,size_t size,
                  int(*compar)(const void*,const void *));
       ```
     - The comparison function returns an integer $<$, $==$, or > 0 if the first argument is considered to be respectively $<$, $==$, or $>$ the second.
  3. Heap Sort — $\mathcal{O}(N \log_2 N)$
     - Sorts in place; no auxilliary storage required.
- Hashing

Interpolation and extrapolation

- Cf. *NRiC* §3.
- Many applications, including fitting unevenly spaced data to a regular grid (recall FFT).

Minimization and maximization (optimization)

- Cf. *NRiC* §10.
- We saw this in the context of modeling data.
- New cool method: simulated annealing.
  - Good for finding global minima (e.g., travelling salesman problem, which is NP-complete and therefore has $\sim \exp(\text{const.} \times N)$ possible combinations, where N is the number of cities to visit—Cf. *NRiC* §10.9).

  - Exploit how nature finds the minimum energy state for a cooling fluid.
  - Sometimes take an uphill step (at low probability, e.g. Boltzmann: $\text{Prob}(E) \sim \exp(-E/kT)$).

Eigensystems

- Cf. *NRiC* §11.
- E.g., principal axes of a rigid body are the eigenvectors of its inertia tensor. Expressions such as total energy and angular momentum become simple in the frame described by the body axes (no matrix multiplication required).

Methods

Adaptive methods

- We saw in *NRiC* that adapability in time (and space) offers advantages for solving differential equations.
- General rule: take a large step (or use coarser grid spacing) when the local derivatives are small. Take a small step (or use finer grid spacing) when the local derivatives are large.
- Adaptive methods are tricky to implement because of increased bookkeeping, difficulty synchronizing data (i.e., timesteps or meshes), and quality control (loss of time reversibility, reflections off mesh boundaries, etc.)

Neural nets

- Used primarily for pattern recognition, e.g., images, sounds, etc.
- Generates *output* depending on given *input* based on *weights* that relate the input to the output (neural paths).
- Train the net with specific inputs of known output. This determines the weights. The more training, the better the predictive power of the net.
- Demo: name generator!
  - Train with real English words, e.g., from a dictionary or an actual text.
  - Divide input into syllables using simple rules.
  - Each syllable is an "output" for the net. The input is the sequence of letters in the syllable. The weights are the probability that a particular letter in a syllable will be followed by another particular letter. E.g., 'e' can appear after a variety of letters, but never after 'q'.

- To generate syllables randomly, assign probabilities of a letter (or "end-of-syllable") based on the weights in the net. E.g., the likelihood of selecting 'q' as the first letter is low, but if you do, the likelihood of selecting 'u' as the next letter is 100%.
- String together the random syllables to form words (or "names.")

Principal Component Analysis (PCA)

- Used to discover or reduce dimensionality of data set and identify new meaningful underlying variables.
- Goal is to define the components (eigenvectors) and strengths of the components (eigenvalues) that best describe the data.
- E.g., with no *a priori* knowledge of the data, can compute eigenvectors and eigenvalues of the covariance matrix.
  - Recall,
  $$\mathrm{var(X)} \equiv \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}.$$
  - The covariance is similar except measured across dimensions:
  $$\mathrm{covar(X, Y)} \equiv \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}.$$
  - Positive covariance $\Longrightarrow$ correlated data, negative $\Longrightarrow$ anti-correlated, zero $\Longrightarrow$ uncorrelated.
  - In 3-D, all possible covariance pairs form 3×3 symmetric matrix (with variances in each dimension along diagonal).
  - Eigenvals and eigenvectors of covariance matrix determine principal relationships between the data, with the vectors being the components and the values being the weights.
- For astronomy, can use PCA to analyze spectra by *training* the analyzer with synthetic/laboratory spectra (to determine eigenvectors) and then applying the algorithm to real data.
- Also used for face recognition and data compression.

Parallel methods

- More of an art than a science!
- Goal is to divide problem among N_p processors, ideally achieving a speedup of factor N_p (rarely attained in practice).
- Requires method of communicating between processors (e.g., low-level library like MPI) and arbitrating collisions (simultaneous requests for same resource).

- Algorithm challenge is to *balance* work between N_p processors. Sometimes straightforward (e.g., rigid mesh), sometimes very hard (e.g., particle methods).

Applications

Particle collisions

- Need to a way to determine whether a collision has occured, or is about to occur.
- Problems arise when collisions are treated as instantaneous events (easiest approach, but can lead to phenomena like *inelastic collapse*).

Radiative transfer

- A type of raytracing with attenuation and source terms.
- The frequency-dependent radiative transfer equation is

$$\frac{dI_\nu}{ds} = J_\nu - K_\nu I_\nu,$$

  where I is the intensity in W m^{-2} sr^{-1}, J is the emissivity (W m^{-3} sr^{-1}), K is the extinction coefficient (m^{-1}), and s is measured along the radiation direction (m).
- Evidently this is an ODE with potentially complicated dependencies, but the usual solution methods still apply.

Tools

Software tools

1. Makefiles
2. Debuggers
3. Revision Control