Numerical methods notes

Jacopo Tissino, Giorgio Mentasti

December 2, 2019

The course is held by Michela Mapelli: a team leader in GW astronomy.

The first lessons are about basic Linux and Python, I will not take notes now, I will start later on.

Thu Nov 07 2019

1 Sorting

Useful for astrophysics since we often deal with large files.

Bubble sort We loop through the file, and swap each pair if it is not ordered.

It is $O(n^2)$ in general.

Selection sort We look for the minimum and put it at the beginning, then scan the remaining array.

It is $O(n^2)$ in general.

Quicksort

1. We pick an element, the *pivot*;
2. we reorder the array so that all elements less than the pivot come before it;
3. we do this recursively to the subarrays to the left and right of the pivot.

It is $O(n^2)$ in the worst case, $O(n \log n)$ usually.

Merge sort We divide the array into small subarrays, and merge them to produce larger subarrays.

It is $O(n^2)$ in the worst case, $O(n \log n)$ usually.

Fri Nov 08 2019

One can time a bash command using

```
time [Command]
```

2 Linear systems

We want to solve a linear system, in the form $A\vec{x} = \vec{b}$, with unknown $\vec{x}$. How do we solve this numerically? We can transform our system to an equivalent one, by

1. exchanging two rows;
2. multiplying an equation by a nonzero constant;
3. adding an equation to another.

These allow us to do Gaussian elimination, and LU decomposition. These are *direct methods*.

Another class is that of *indirect methods*: we start with an *ansatz* and refine it. These are easier to implement, more generally applicable, more efficient if the matrix is sparse. They, however, do not always converge.

An example is the Gauss-Seidel method.

Thu Nov 14 2019

2.1 Gauss-Seidel

We can rewrite $\sum_j A_{ij} x_j = b_i$ as

$$x_i = \frac{1}{A_{ii}} \left(b_i - \sum_{j \neq i} A_{ij} x_j \right) , \tag{1}$$

and the algorithm works by starting with and *ansatz*, updating it with this formula, and iterating. The update can be written more generally as

$$x_i^{n+1} = \frac{\omega}{A_{ii}} \left(b_i - \sum_{j \neq i} A_{ij} x_j^n \right) + (1 - \omega) x_i^n , \tag{2}$$

with the *relaxation parameter* ω. Do note that n is not an exponent but an iteration number.

A good choice for ω after the 5th iteration:

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - (\Delta x^{k+p} / \Delta x^k)^{1/p}}} \,, \tag{3}$$

Fri Nov 15 2019

Implementation of Gauss-Seidel.

Thu Nov 28 2019

We want to generate random numbers in order to draw samples from known distributions. We can only really generate *pseudo*-random numbers: for example, we vary the integer$x$ in the formula

$$x' = (ax + c) \mod m \,, \tag{4}$$

with constant integer a, c and m.

Importantly, the values of the constants and the starting value of x constitute a *seed* which can be used to reproduce our results.

If we want to produce numbers distributed according to an arbitrary pdf, we first produce uniformly distributed numbers, and then use the laws of probability. Say we have a desired pdf, $p(x)$, and numbers uniformly distributed from 0 to 1, then

$$\int_{-\infty}^{y(x)} p(\widetilde{y}) \, \mathrm{d}\widetilde{y} = \int_0^x \mathrm{d}\widetilde{x} = x \,, \tag{5}$$

so if we are able to solve this and make $y(x)$ explicit as a function of x we are done! The final expression is given in terms of the cumulative pdf:

$$y = P^{-1}(x) \,. \tag{6}$$

We want to generate numbers m between 0.1 and 150 distributed according to $p(m) = m^{-\alpha}$.

The cdf is

$$\frac{1}{N} \int_{0.1}^{m} m^{-\alpha} \, \mathrm{d}m = \frac{1}{N} \left(-\frac{0.1^{-\alpha}}{1-\alpha} + \frac{1}{1-\alpha} m^{1-\alpha} \right) , \tag{7}$$

where the normalization is

$$N = \int_{0.1}^{150} p(m) \, \mathrm{d}m = \frac{1}{1-\alpha} \left(150^{1-\alpha} - 0.1^{1-\alpha} \right) , \tag{8}$$

therefore we have

$$Nx = \left(-\frac{0.1^{-\alpha}}{1-\alpha} + \frac{1}{1-\alpha} m^{1-\alpha} \right) , \tag{9}$$

so

$$m = \left(\left(0.1^{1-\alpha} + (1-\alpha) Nx \right) \right)^{1/(1-\alpha)} . \tag{10}$$

This works; a note of caution: when using a linear axis the histogram (plot of the number "dN" of counts in a bin of constant width "dm") looks (up to a proportionality constant) like the probability density function, however if we plot the logarithm of our random variable we get

$$\frac{\mathrm{d}N}{\mathrm{d}\log m} = \frac{\mathrm{d}N}{\mathrm{d}m} \frac{\mathrm{d}m}{\mathrm{d}\log m} = \frac{\mathrm{d}N}{\mathrm{d}m} m , \tag{11}$$

so if we want somthing which is proportional to the pdf we need to normalize the bin counts by multiplying them by $1/m$.

Fri Nov 29 2019

The Plummer sphere is defined by

$$\begin{aligned} \rho(r) \, \mathrm{d}^3 r &= \frac{3M}{4\pi a^3} \left(1 + \frac{r^2}{a^2} \right)^{-5/2} \mathrm{d}^3 r & (12) \\ &= \frac{3M}{4\pi a^3} \left(1 + \frac{r^2}{a^2} \right)^{-5/2} r^2 \, \mathrm{d}r \, \mathrm{d}\Omega , & (13) \end{aligned}$$

and it models the mass density in a star cluster. We can marginalize over the angles to find the pdf of the radius alone: this just amounts to multiplying by 4π by isotropy,

$$\rho(r) \, \mathrm{d}r = \frac{3M}{a^3} \left(1 + \frac{r^2}{a^2} \right)^{-5/2} r^2 \, \mathrm{d}r , \tag{14}$$

and if we rescale the radius as $R = r/a$ we find the PDF

$$\rho(R) \, \mathrm{d}R = 3M \left(1 + R^2 \right)^{-5/2} R^2 \, \mathrm{d}R , \tag{15}$$

which can be integrated analytically: we get

$$\int_0^{R_{\max}} \rho(R)\,\mathrm{d}R = \frac{MR_{\max}^3}{\left(R_{\max}^2+1\right)^{3/2}}\,. \tag{16}$$

The velocities are isotropically distributed, with the probability of their moduli being described by a Maxwellian density:

$$p(v)\,\mathrm{d}v = \sqrt{\frac{2}{\pi}}\frac{v^2}{\sigma^3}\exp\left(-\frac{v^2}{2\sigma^2}\right)\mathrm{d}v\ . \tag{17}$$

Do note that this is *not* normalized on $\mathbb{R}$ as given: its integral is 2, it is normalized on $\mathbb{R}^+$.

The parameters are given by: $M = 10^4 M_\odot$, $a = 5\,\mathrm{pc}$, $\sigma = 5\,\mathrm{km/s}$.

Here is my approach to the problem without reading the suggestions, it might not be the most efficient way to do it.

We need to be able to draw samples from three distributions: the Plummer sphere, the Maxwell distribution and the angular distribution; since both the star's positions and their velocities are isotropically distributed on the 2-sphere. The volume element on the sphere S^2 is given by $\mathrm{d}A = \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\varphi$; so we can draw our φ from a uniform distribution on $[0, 2\pi]$, while our θ will need to be distributed according to $p(\theta)\,\mathrm{d}\theta = \sin\theta\,\mathrm{d}\theta$.

Since we are computing angles spanning the whole 2-sphere, for both the radius r and the velocity v we do not need to simulate negative values.

This was my first approach, but then I noticed that it is really hard to sample from a distribution $f(x) \sim x^2\exp\left(-x^2\right)$.

For the Plummer distribution it makes sense to sample from the radial and angular distribution separately, while for the Maxwell distribution it is much easier to sample the three components of the velocity vector in cartesian coordinates. Let us see this: first, we rescale $V = v/\sigma$. We get:

$$p(V)\,\mathrm{d}V = \sqrt{\frac{2}{\pi}}\exp\left(-V^2/2\right)V^2\,\mathrm{d}V\ , \tag{18}$$

and we can add in the uniform distribution $1/4\pi$ of the angular part: we find

$$\begin{aligned} p(V)\,\mathrm{d}V\,\mathrm{d}\Omega &= \frac{1}{4\pi}\sqrt{\frac{2}{\pi}}\exp\left(-V^2/2\right)V^2\,\mathrm{d}V\,\mathrm{d}\Omega && (19)\\ &= \frac{1}{(2\pi)^{3/2}}\exp\left(-V^2/2\right)\mathrm{d}^3V && (20)\\ &= \prod_{i=1}^{3}\left(\frac{1}{\sqrt{2\pi}}\exp\left(-V_i^2/2\right)\mathrm{d}V_i\right), && (21) \end{aligned}$$

so we can see that in cartesian coordinates each component is just distributed according to a Gaussian.