NuMe Disposition

xzb187

November 2022

1 Topic 1. floating points - floating point representation, truncation and rounding errors

First do the general formula then the one that includes floating points.

$$\pm(d_N \dots d_0)_b = s(\sum_{i=0}^{N} d_i \cdot b^i)_{10}$$

$$\pm(d_n \dots d_0, d_{-1}, d_{-2} \dots d_{-m})_b = s \sum_{i=-m}^{n} d_i \cdot b^i$$

Talk about truncating error

$$|(x)_b - (\hat{x})_b| < b^{-n}$$

And then rounding error, which is twice as "good"

$$|(x) - (\tilde{x})| \leq \frac{1}{2} b^{-n}$$

Normalized scientific notation, and then use that to explain

$$(x)_b = q \cdot b^m$$

where q must be a zero followed by a comma and then a number that is not zero.

machine numbers with 32 bits.
The machine number can be written as 1 sign bit, 8 exponent bits and 23 mantissa bits (fraction bits). The exponent bias is 127 ($e = 127 + m$).
When converting, we can improve precision by 2, if we use a modified scientific notation like $1.01 \cdot 2^3$, because then we just remember the first 1 and just save the fraction.

Absolute error in machine number using rounding is

$$|x - \tilde{x}| \leq \frac{1}{2} 2^{m-q}$$

2 Topic 2: solving nonlinear equations - bisection/dichotomy method, Newton method, Horner algorithm

Introduction: Root finding and evaluation of polynomials. Naive vs. nested multiplication.

Bisection method to find the root of a function. Draw example and then talk about error Size of interval after n iterations

$$(b_n - a_n) = (\frac{1}{2})^n (b_0 - a_0)$$

$$(\frac{1}{2})^{1+n} (b_0 - a_0) \leq \delta$$

And it only converges when given an interval that has a root. It can stop at either at a relative/absolute stopping criteria or at a max iterations. Linear convergence rate.

Newtons method to find the root of a function. Draw example. It uses the derivative to find the next x to evaluate. This can be written as the newton update

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)}$$

We can again stop at either at a relative/absolute stopping criteria or at some max iterations.
Newtons does not always converge, it can oscillate in some cases (draw example), and $f'(x) \neq 0$. Converges quadratic

Secant method is like newton but where we either cant differentiate or it is too consuming.

Illustrate naive evalutation. General example to illustrate nested multiplication to evaluate polynomials. This evaluates a polynmial using only n multiplications and n additions, compared to n^2 multiplications using naive.

This is the principle behind Horners method to evaluate polynomials. This can easily be writtin in table-form. Using the new coefficients we can calculate the derivatives. Going back to Newtons method, we see that we can evaluate the newton iteration just using horners method for both $f(x)$ and for $f'(x)$.

3 Topic 3: Solving linear systems - Matrix norms, LU decomposition, a bit of Jacobi/Gauss-Seidel solvers

Introduction: We want to solve a system of equations

$$Ax = b$$

First we talk about the vector norm. Draw norms when $||x|| = 1$.

Then we look at matrix norms. Here we define what norm 1, 2 and infinity are.

$$||A||_1 = max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|$$

$$||A||_2 = max_{1 \leq i \leq n} |\sigma_i|$$

Where σ is a singular value of A. or as

$$||A||_2 = \sqrt{\rho(A^T A)}$$

Where ρ is a function that gives the spectral radius. Where

$$\rho(A^T A) = max_{1 \leq i \leq n} \lambda_i$$

Where λ is an eigenvalue of $A^T A$. It just finds the largest eigenvalue of $A^T A$. In short: The 2 matrix norm is the square root of largest eigenvalue of $A^T A$

$$||A||_\infty = max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|$$

Then I will look at different LU decompositions. These are best with dense matrix and give exact value.
First we know that an LU decomposition can be made of a square symmetric matrix using Cholesky

$$A = LL^T$$

then we look at Crout (and Doolittle), where we create a lower triangular and an upper. In Crout the diagonal of the upper is 1 and in Doolittle the diagonal of the lower is one. To make the factorization we just make a system of equations using matrix multiplication.

Using LU to solve $Ax = b$, we first define $LUx = b$. Then we make $Ux = z$ and then we solve $Lz = b$ using forward substitution to then solve $Ux = z$ using backwards substitution.

Finnaly I will talk about splitting matrixes how to use them and the examples of Jacobi and gaus seidel.
Splitting matrices: We redefine the equation, such that we introduce a Q matrix:

$$Ax = b \iff Qx = (Q - A)x + b$$

Analytically we can define a splitting matrix iteration as

$$x_{n+1} = (I - Q^{-1}A)x_n + Q^{-1}b$$

Different methods set Q as different matrices.
In the Jacobi method we set $Q = D$ and get

$$Dx = (D - A)x + b = (-L - U)x + b$$

And therefore the iteration

$$x_{k+1} = (I - D^{-1}A)x_n + D^{-1}b$$

Using the Gauss-seidel we set $Q = D + L$.

Talk about conditioning number if needed

4 Topic 4: Interpolation - theorem of polynomial interpolation (core ideas), Largange interpolation form, Spline interpolation

If we have n distinct coordinate pairs, then we can find a polynomial of degree $n - 1$ that will pass all points. There exists only one unique polynomial that satisfy this for every case. This is what enables secret sharing.

The lagrangian interpolation is defined on the form

$$f(x) = c_1\delta_1(x) + c_2\delta_2(x) + c_3\delta_3(x) + \ldots + c_n\delta_n(x)$$

Which formally an be written as

$$p(x) = \sum c_i\delta_i(x) = \sum_i f(x_i) \cdot \Pi_j \frac{x - x_j}{x_i - x_j} =$$

Where the delta functions are "binary indicator functions", that indicate what the y value at an x is supposed to be, when x is at a data point.

Spline interpolation. Instead of a straight line or just one polynomial, we can make a new polynomia between every to points. We call them splines.
One way it to make 1 degree, those a straight lines. But a better approximation is second degree polynomials. To calculate the two splines, we need to do it simultanous to have the same slope. We can define the two splines as

$$y_{-1} = a_{-1,0} + a_{-1,1}x + a_{-1,2}x^2$$

$$y_1 = a_{1,0} + a_{1,1}x + a_{1,2}x^2$$

Then we need the following to be true

$$\frac{d}{dx}(a_{-1,0} + a_{-1,1}x + a_{-1,2}x^2) = \frac{d}{dx}(a_{1,0} + a_{1,1}x + a_{1,2}x^2)$$

And we know it has to be true at x_i therefore $x = x_i$

$$\iff$$

$$a_{-1,1} + 2a_{-1,2}x_i = a_{1,1} + 2a_{1,2}x_i$$

To be able to have a system with as many unknowns as equations, we have to set one of the parameters to begin with (usually $a_{2,0}$ such that the first spline is linear.)

5 Topic 5: Eigenvalues, Orthogonal factorization and LS problems - Power Method, Gram Schmidt factorisation, QR factorisation

Power method: Used to find the largest eigenvalue. We have a matrix A and a random start vector x. We then multiply Ax, save it in y. Then we calculate the ration (which is the eigenvalue) by using a linear transformation (ϕ) of y and x, and then dividing y by x to get the ration of how much it changes. Then we normalize y and save it in x. Reapeat. Mention inverse to get the smallest eigenvalue.
Mention Gershgorins discs?

Gram-Schmidt: We want to create an ortonormal basis by removing the "direction" of each vector from one another and then normalize them.

QR using householder.
If we have a matrix A with M rows and N columns (MxN) and $M \geq N$ (more or equally as many rows as columns), then A can be factorized as $A = QR$ where Q is a MxM orthonormal matrix and R is a MxN upper triangular matrix. Householders calculates $Q^T A = R$ recursively by calculating

$$(Q_N^T \dots Q_2^T Q_1^T)A = Q^T A = R$$

With every iteration we first calculate the length (norm 2) of the k'th vector in A $B_k = ||a_k||_2$. Then we calculate $y_k = a_k - B_k e_1$ where e_1 is the elementary vector with as many elements as a_k (e.g. $[1, 0, 0]^T$ with 3 elements). Then normalize it $\hat{y_k} \frac{y_k}{||y_k||_2}$, and then Q_k^T is given by

$$I - 2\hat{y}\hat{y}^T$$

By every iteration the dimension of Q^T matrix is reduced by one row and one column, with the identity matrix to fill it out, so they all have the same dimension. Generic example with the first 2 iterations, and then the k'th iteration:

6 Topic 6: Numerical differentiation - Numerical differentiation error and Taylor series, Richardson Extrapolation

6.1 Introduction

1Introduction to different ways to differentiate:
We know the analytic defintion of the derivative

$$\frac{df}{dx} = lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

We can then make a simple method to calculate the derivative using a "small h":

$$\frac{df}{dx} \approx \frac{f(x+h) - f(x)}{h}$$

This is not a good approximation since the error scales linearly. This can be proven by using a taylor series.
A better approximation is using

$$\frac{df}{dx} \approx= \frac{f(x+h) - f(x-h)}{2h}$$

This can be proven by Taylor series

$$f(x+h) = f(x) + \frac{h^1}{1!}f'(x) + \frac{h^2}{2!}f''(x) + \ldots + \frac{h^n}{n!}f^{(n)}(x)$$

(Quick note about the error term of this to the earlier method).
And now we also have $f(x-h)$ which is

$$f(x-h) = f(x) - \frac{h^1}{1!}f'(x) + \frac{h^2}{2!}f''(x) - \ldots + \frac{h^n}{n!}f^{(n)}(x)$$

All the odd terms are negative, since $(-h)^n$ gives a positive number when n is even.
If we now subtract them from each other

$$f(x+h) - f(x-h) = 2hf'(x) + 2\frac{h^3}{3!}f'''(x) + \ldots + 2\frac{h^n}{n!}f^{(n)}(x)$$

Since all the even terms cancelled out, we are left with all the odd terms. Now isolate $f'(x)$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6}f'''(x) + \ldots + \frac{h^{n-1}}{2n!}f^{(n)}(x)$$

And we now see that the error term is $O(n^2)$ and therefore scales quadraticly and is a better approximation.
To get an even better approximation we can get rid of the error term using

Richardsons extrapolation:
Redefining the error as $-t \cdot h^2$ such that

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + t \cdot h^2 \tag{1}$$

We want to minimize the error term, so we make a new equation with $\frac{h}{2}$

$$f'(x) = \frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} + t \cdot \frac{h^2}{4} \tag{2}$$

Now we subtract (1) with 4 times equation (2) such that $(1) - 4(2)$:

$$-3f'(x) = \frac{f(x+h) - f(x-h)}{2h} - 4\frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} + 0$$

We see that the error term cancels out since $th^2 - 4t\frac{h^2}{4} = 0$.
Again isolating $f'(x)$

$$f'(x) = \frac{4}{3}\frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} - \frac{f(x+h) - f(x-h)}{6h}$$

This now approximates the derivative very precisely, and now only comes down to machine precision.
Quick note about diriving using interpolation: E.g. lagrange interpolation.

7 Topic 7: Numerical integration - Method of undetermined coefficients, Gaussian Quadrature, Simpson's rule

Introduction to numerical integration: Average method and trapozoid rule.
A better approximation is the undetermined coefficients method.

Now we define it to be equal to some linear combination of $f(a)$ and $f(b)$ as

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2} = A_0 \cdot f(a) + A_1 \cdot f(b)$$

And the straight line between a and b can be defined at $c_0 + c_1 x$. And the integral of that line is

$$\int_a^b c_0 + c_1 x dx = \left[c_0 x + \frac{c_1 x^2}{2}\right]_a^b = (c_0 b + \frac{c_1 b^2}{2}) - (c_0 a + \frac{c_1 a^2}{2}) = c_0(b-a) + c_1(\frac{b^2-a^2}{2})$$

We can now define $f(a)$ and $f(b)$ using the straight line just defined

$$A_0 \cdot f(a) + A_1 \cdot f(b) = A_0(c_0 + c_1 \cdot a) + A_1(c_0 + c_1 \cdot b)$$

We rearrange to have it in terms of c_0 and c_1

$$A_0(c_0 + c_1 \cdot a) + A_1(c_0 + c_1 \cdot b) = c_0(A_0 + A_1) + c_1(A_0 \cdot a + A_1 \cdot b)$$

We can now compare that to the true integral we calculated for the straight line, which was $c_0(b-a)+c_1(\frac{b^2-a^2}{2})$, and we can therefore make the 2 equations

$$b - a = A_0 + A_1$$

$$\frac{b^2 - a^2}{2} = A_0 \cdot a + A_1 \cdot b$$

We solve for A_0 and A_1 and get

$$A_0 = A_1 = \frac{b-a}{2}$$

Which are the same coefficients as the trapezoid rule.

Simpsons rule:
Using 3 points a, $\frac{a+b}{2}$ and b, and now we define a polynomial of 2nd degree to go through a and b:

$$f_{ab}(x) = a_0 + a_1 x + a_2 x^2$$

And the integral of that function to be

$$\int_a^b f(x)dx = \left[a_0 x + \frac{a_1 x^2}{2} + \frac{a_2 x^3}{3}\right]_a^b = a_0(b-a) + \frac{a_1}{2}(b^2 - a^2) + \frac{a_2}{3}(b^3 - a^3)$$

And we now define $f(a)$, $f(\frac{a+b}{2})$ and $f(b)$ in terms of that polynomial to make 3 equatoins

$$f(a) = a_0 + a_1 \cdot a + a_2 \cdot a^2$$

$$f(\frac{a+b}{2}) = a_0 + a_1(\frac{a+b}{2}) + a_2(\frac{a+b}{2})^2$$

$$f(b) = a_0 + a_1 \cdot b + a_2 \cdot b^2$$

We now have 4 equations (with the true integral) and 4 unknowns (with the value of the integral). We solve the equations, and the integral is then equal to

$$\int_a^b f(x)dx = \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b))$$

$$\iff \int_a^b f(x)dx = \frac{h}{3}(f(a) + 4f(a+h) + f(a+2h))$$

8 Topic 8: Monte-Carlo simulation - MC integration, an example of using MC (not the examples that were presented in the lectures)