Scientific Computing CS660 Fall '11

Angjoo Kanazawa

October 24, 2011

1 September 1st Class 1

1.1 Logistics

Prof. Howard Elman CSI 2120 TR 2pm-3:15pm class url:http://www.cs.umd.edu/ elman/660.11/syl.html

- Scientific Computing puts heavier emphasis on computing, Numerical Analysis is more about proofs/theories.
- 4-6 hw asg: **35%** Penalty on late assignments (-15% after 24 hrs, -30% after 48 hrs.
- in-class midterm: **25%**
- final project: **40%**

1.2 Content

Newton's Method: Root finding. Objective: Fine x s.t. $f(x) = 0$, where f is a scalar, $f : \mathbf{R} \Rightarrow \mathbf{R}$ function. Where does the function cross 0 (x-axis)?

Given x_n, some guess, find where the line through $(x_n, f(x_n))$ tangent to the solution curve intersects the x-axis. Call that pt of intersection x_{n+1}

The equation of the tangent line:

$$\frac{y - f(x_n)}{x - x_n} = f'(x_n)$$

Set $y = 0$: then

$$\begin{aligned}\frac{0 - f(x_n)}{x_{n+1} - x_n} &= f'(x_n)\\ \frac{x_{n+1} - x_n}{-f(x_n)} &= 1/f'(x_n)\\ x_{n+1} &= x_n - f(x_n)/f'(x_n)\end{aligned}$$

Another Derivation Consider the taylor series $f(x_n + (x - x_n)) = f(x_n) + f'(x_n)(x - x_n) + 1/2 f''(x_n)(x - x_n)^2 +$ etc. This is a function of some variable x. Approximate it just by using the first two terms (linear approximation). So above becomes a new function $f(x_n+(x-x_n)) \ = f(x_n)+f'(x_n)(x-x_n) = l(x)$. Find where $l(x) = 0$. That is: $x_{n+1} = x_n - f(x_n)/f'(x_n)$

This is not guaranteed to work, i.e. when the tangent doesn't cross the x-axis.

Problem: given $\alpha \in \mathbf{R} > 0$ Find $1/\alpha$ without doing any division. First thing you need to do if identify (concoct) a $f(x)$ whose root is $1/\alpha$. Naive: try $f(x) = x - 1/\alpha$. But this won't work, because this requires division. Howabout: $f(x) = \alpha x - 1$. $f'(x) = \alpha$, so in the newton iteration..

$$\begin{aligned} x_{n+1} &= x_n - \frac{\alpha x - 1}{\alpha} \\ &= x_n - x_n + 1/\alpha \\ &= 1/\alpha \end{aligned}$$

No! because you need to divide.

Answer: $f(x) = \alpha - 1/x$. Not transparent, because intuitively it looks like there's a divide into it. $f'(x) = - - 1/x^2 = 1/x^2$. Then, $f/f' = \alpha x^2 - x$. So given x_n, the iteration goes $x_{n+1} = x_n - (\alpha x_n^2 - x_n) = 2x_n - \alpha x_n^2 = x_n(2 - \alpha x_n)$.

Numerical example in matlab: let $\alpha = 2$, solve with this method. Use $x_0 = 0.1$. Notice that $err(i)/err(i-1)^2$ is constant and *is* α.

$$\frac{|x - x_{n+1}}{x - x_n}^{2} \approx 1/2\frac{f''(x_n)}{f'(x_n)}$$

Notice the $err(i)$ decreases faster as iteration moves on, this is the super linear convergence property of Newton's method. With $\alpha = 0.25$, same thing.

Analysis of the Newton's Method: $|x_{x_{n+1}}|/|x - x_n|^2 \approx 1/2|f''(x_n)/f'(x_n)| \approx 1/x$ So @ $x = 1/\alpha$, this turns out to be α, just for this example.

The ratio was derived from the idea to find a patter in the errors s.t. $e_{n+1} \ ce_n^2$, for some $c \in \mathbf{R}$

The question is to find trends in data, and this relationship $e_{n+1} \ ce_n^p$ is useful to tell us the rate of convergence as the solution approches the optimal one. (I think p goes down to the golden ratio). Our goal is to find what p is for a specific problem.

2 September 6th class 2

2.1 Process of Scientific Computing

- Start with a mathematical model. (In general we don't have an analytic solution, so we get insight from numerical computation)
- Example with heat conduction in a bar:
  - A 1-D object $\in [0,1]$, $u(x)$ =temperature in the bar, with $u(0) = 0, u(1) = 0$. q=heat flow induced by a heater of intensity f.
  - We want what the temperature will be given q.
  - get some models: *Fourrier's Law*: $q = -ku'$, k=conductivity coefficient, transfer of heat in direction of decreasing temperature (hence the -). *Conservation of energy*: $q' = f$.
  - **Goal**: find u. **Equation of interest**: $-(ku')' = f$, the 1D diffusion equation.
  - Typical strategy: lay down a grid $x_o = 0, x_1, \ldots, x_n, x_n + 1 = 1$. Compute a discrete solution, $\bar{u}$, vector of size n. $\bar{u} = [u_1, \cdot, u_n]^T$, where $u_i \approx u(x_i)$
  - **Claim**: we can find the discrete sol, $\bar{u}$ by solving an algebraic system of equations. For this example, this system is a matrix equation

    $$A\bar{u} = \bar{b}$$

- No matter how hard I try, we're never going to get the exact solution. This process $A\bar{u} = \bar{b}$ leads to errors
- **Sources of error**:
  1. Modeling error: we may not know k exactly.
  2. Discretization error/Truncation error: difference between the discrete and the continuous values (from the approximation on a discrete set of points)
  3. Representation error: we don't have the entire **R**, we only have a finite set, in floating point format. (A and b may have error)
  4. Additional error: from the computation of $\tilde{u}$, will get something else $\hat{\tilde{u}} \neq \tilde{u}$
     we can show:

     $$\frac{||\hat{\tilde{u}} - \tilde{u}||}{||\hat{\tilde{u}}||} \leq K(A)\mu$$

     That is if we solve the problem appropriately (like being careful about pivoting etc).
- We're really trying to solve $\tilde{A}\tilde{u} = \tilde{b}$., where $\tilde{A} \approx A$, $\tilde{u} = \bar{u}$. Typically $\approx$ is machine precision, 10^{-16}
- In the end, we want $u(x)$, and would be happy with $u(x_j)$, $j = 1, \ldots, n$. That will get $\tilde{u_j}$
- The moral is that when we do this stuff, we're not just doing mathematics.

2.2 Floating Point Arithmetic

decimal numbers (base 10). Consider the example 6522 and 10.31

- $6522 = 6(10^3) + 5(10^2) + 2(10^1) + 2(10^0)$
- $10.31 = 1(10^1) + 0 + 3(10^-1) + 1(10^-2)$
- Normalize the numbers! then,,
- $6522 = 6.522 \times 10^3$ so we can express numbers with one digit to the left o fthe decimal point.
- $10.31 = 1.031 \times 10^1$
- For any number but 0, it has a fomr $z \times 10^p$, where $z \in [1, 10)$.

Computers use binary representation. Example: $3_{10} = 1(2^1) + 1(2^0) = 11$ or 1.1000×2^1.

$23_{10} = 16+4+2+1 = 1(2^4)+0(2^3)+1(2^2)+1(2^1) = 10111_2$ or 1.0111×2^4 *normalized.* Here, normalized means $z \times 2^p$, where $z \in [1, 2)$

3 September 8th Class 3

3.1 Floating Point Operations

Example: addition using d=5 binary digits

Add $3 + 23$.

Normalized 5-digit binary expressions:

$3 = 2^1 + 2^0 = 1.1000 \times 2^1$

$23 = 16 + 4 + 2 + 1 = 1.0111 \times 2^4$

To add these, shift the smaller number so that the exponents agree.

$$1.1000 \times 2^1 = 0.11000 \times 2^2 \tag{1}$$
$$= 0.01100 \times 2^3 \tag{2}$$
$$= 0.00110 \times 2^4 \tag{3}$$

May need to round the result: 3+34 32+2 =

Shift smaller number

$$\begin{array}{r l} 3 ==> & 0.000110 \times 2^5 \\ 34 ==> & 1.00010 \times 2^5 \\ ------------------ & \\ 37 ==> & 1.00101 \end{array}$$

The way the arithmetic is done: take 2 floating point numbers. The computer hardware will take the correct result (stored temporarily), but will always round to d digits. In our examples we are using only $d = 5$. in the real world we are usually dealing with $d = 27$ or more. number will always be rounded to closest representation– can be either up or down.

so in this case, the stored result is rounded to 1.0011×2^5 which is $32+4+2 = 38$.

Exercise: what two number would have the property whose sum's **correct** result is 1.001001? Answer: 32+4+0.5. (Note: 2.5 in binary is $1.0100 \times 2^1 = 2 + 1/2$).

In general, given the real x, $fl(x)$ = closest floating point number to x. $|x = fl(x)|$ is called the rounding error. For operations op $= \{+, -, \times, \div\}$, if x and y are floating point numbers, then $fl(x \text{ op } y)$ = floating point number closest to x *op* y.

It is possible that both inputs are perfectly good FP numbers, but that the result of the operation is not able to be represented in the floating point system (eg. d is too small). For example, in our single precision example, $d = 23$, $m = -126 \leq p \leq M = 127$. Then, if $x = 1 \times 2^{100}$ and $y = 1 \times 2^{100}$ then $x * y = 1 \times 2^{200}$ which results in an overflow.

Definition *Machine Precision* or *Machine Epsilon* is the smallest floating point number μ such that $fl(1 + \mu) \neq 1$ (> 1)

In IEEE arithmetic, for any real $x \neq 0$:

$$\frac{|x - fl(x)|}{|x|} \leq \mu$$

Example: $d = 5$. Find μ.

$$\begin{aligned} 1 &= 1.0000 \times 2^0 \\ \text{Add } z &= 0.00001 \times 2^0 \\ &-------------- \\ &= 1.00001 \end{aligned}$$

=¿ which gets rounded to 1.0001.

$fl(1+z) \neq 1$

Next smallest number to z. Normalized, $z = 1.0000 \times 2^{-5}$. Next smaller number is $z = 1.1111 \times 2^{-6}$.

$$\begin{aligned} 1 &= 1.00000 \\ +z &= 0.0000011111 \\ &----------------- \\ &= 1.0000011111 \end{aligned}$$

which gets rounded to 1.

In general, in d-digit binary arithmetic, the machine precision μ is 2^{-d}.

For IEEE single precision, this is $2^{-23} \approx 1.2x10^{-7}$ For IEEE double precision, this is $2^{-53} \approx 1.1x10^{-16}$

Does IEEE arithmetic support the existence of numbers like 2^m or 2^M?

3.2 Relative Error

We have a number x, and a representation of $x \approx \hat{x}$.

The relative error is given by

$$\frac{||x - \hat{x}||}{||x||}$$

And the absolute error is given by $||x - \hat{x}||$.

The relative error is more important than the absolute error. Consider for example a census, x% absolute error in the number of ppl in class compared to the same x% absolute error in the population of nyc does not mean the same thing.

3.3 Forward vs. Backward Error Anlysis

Generically speaking: we are seeking $y = f(x)$, and get $\hat(y) \neq y$. We want insight into $\frac{||y-\hat{y}||}{||y||}$.

Definition: *Forward error analysis* tries to keep track of errors as the computation proceeds.

Example:

Solve $Ax = b$. We get $\hat{x}$ instead. And we are interested in $\frac{||x-\hat{x}||}{||x||}$. Forward error analysis (egf. for gaussian elimination) is not possible. (Well, it is possible, but the estimates tend to be wildly inaccurate).

Definition:*Backward error analysis* makes the claim that $\hat{x}$ is the solution of a perturbed problem, $\hat{A}\hat{x} = b$ such that (if the solution is done right):

$$\frac{||x-\hat{x}||}{||x||} \lessapprox \mu(p(n)\mu)$$

where $p(n)$ is a slowly growing function of $n =$ problem size.
We use this observation: $Ax = b$ $\hat{A}\hat{x} = b$
and

$$\begin{aligned}
A(x-\hat{x}) &= Ax - A\hat{x} &(4)\\
&= Ax - \hat{A}\hat{x} + \hat{A}\hat{x} - A\hat{x} &(5)\\
&= b - b + (\hat{A} - A)\hat{x} &(6)\\
&= 0 + E &(7)
\end{aligned}$$

where E is our error.

$$\begin{aligned}
&\to x - \hat{x} = A^{-1}E\hat{x} &(8)\\
&||x-\hat{x}|| \leq ||A^{-1}||||E||||\hat{x}|| &(9)\\
&||A^{-1}||||A||||E||/||A||||\hat{x}|| &(10)
\end{aligned}$$

$||A^{-1}||||A||$ is called $\kappa(A)$, and $||E||/||A|| \lessapprox \mu$
this $\to ||x-\hat{x}||/||\hat{x}|| \lessapprox \kappa(A)\mu$
$\mu \approx 10^{-16}$
$\kappa(A)$ depends on A, the statement of the problem..
We claim, that with a bit more work, we could put $||x||$ in the denominator.

4 September 13rd Class 4

4.1 Intro to Probability

Discrete models: Consider a game with a finite or countable number of outcomes, Ω, the *sample space*. **Definition:** *Probability* for each $\omega_j \in \Omega$, we have a number $p_j(\omega_j) = p_j$ s.t. $p_j \in [0,1]$, and $\sum_{j=1}^{\infty} p_j = 1$

Examples:

1. Roll a fair die outcomes: $\Omega = \{1,2,3,4,5,6\}$, $p_j = 1/6$, $j = 1,2,\ldots,6$
2. Roll two fair dice, outcomes: $\Omega = \{(1,1),(1,2),\cdots,(1,6),(2,1),\cdots,(6,6)\}$, $p_j = 1/36$, $j = 1,\ldots,36$.
3. put n distinct balls into N urns $N > n$. 1st ball has N choices, so does the 2nd ball, etc, so the total number of outcomes is N^n.
4. A die is rolled until a six appears. possible outcomes:
   (a) Get a 6 on 1st trial: 6, $p(\omega_1) = 1/6$
   (b) something other than 6 on 1st trial: 16,26,36,46,56 (5 ways), $p(\omega_2) = 5/6(1/6) = 5/36$
   (c) etc $p(\omega_i) = (1 - p(\omega_1))^{i-1})p(\omega_1) = (5/6)^{i-1}(1/6)$

   Here Ω is countably infinite. The sum:

$$\begin{aligned}\sum_{j=1}^{\infty} p_j &= 1/6 + (5/6)(1/6) + (5/6)^2(1/6) + \cdots \\ &= 1/6\sum_{l=0}^{\infty}(5/6)^l \\ &= 1/6(\frac{1}{1-5/6}) = 1/6(\frac{1}{1/6}) = 1\end{aligned}$$

Definition: An *event* is a subset $A \subseteq \Omega$. Notation $P(A) = \sum_{\omega_J \in A} p(\omega_j)$. Properties: $\P(\Omega) = 1, P() = 0, P(A^c) = 1 - P(A)$. If $A_1, A_2, \ldots$ are pairwise disjoint, then $P(\cup_i A_i) = \sum_i P(A_i)$.

Definition: A *random variable* X is a function $X : \Omega \to \mathbf{R}$. $\forall E \in \mathbf{R}$, define an *event* $\{X \in E\} = \{\omega \in \Omega | X(\omega) \in E\}$ Can talk about probability of such an event as $P(X \in E)$

Definition: A *distribution* of a discrete random variable X is a collection of distinct real numbers, or a set of values $\{m_j\} \in [0,1]$, s.t. $\sum m_j = 1$, and $P(X = x_j) = m_j$.

Example: *Binomial distribution.* Consider a random experiment with two possible outcomes. Psobability of success = p, failure $q = 1 - p$. With coins, $p = 1/2$. Perform this experiment n times, $X =\#$ of successes. X has a $Binom(p, n)$. The probability of exactly l sucesses is $P(X = l) = \binom{n}{k} p^k q^{n-k}$

5 September 15th Class 5

5.1 Probability Cont.

Experiment: three tosses of a coin $|\Omega| = 2^3$

Definitino: The distribution of discrete random variables is a collection of real positives $\{x_j\}_{j=1}^{\infty}$, s.t. $P(X = x_j) = m_j$

Definition: A *mass function* of a discrete random variable X is $m : \mathbf{R} \to [0,1]$ s.t. $m(x) = P(X = x) \forall x \in \mathbf{R}$.

Definition: The binomial distriubtion is a random experiment with two outcomes where p is the probability of success, $1 - p$ is the probability of failure. It performs the experiment n times, and X denotes the number of total successes.

Q: what is the probability of exactly k successes? $\binom{n}{k} p^k (1-p)^{n-k}$, let's check $\sum_{j=1}^{\infty} m_j = 1$. This is $\sum_{k=0}^{n} p^k (1-p)^{n-k} = (p + (1-p))^n = 1^n = 1$ so yes.

Definition: Continuous random variables and distributions. ex: Average weight of 100 randomly selected , where Ω={100 tuples of weights }, $X(\omega) = 1/100$ (sum entries of ω). Random number uniformly chosen from $[a, b]$.

Definition: A function f on $\mathbf{R}$ is called a *density function* if it's nonnegative and integrable and $\int_{-\infty}^{\infty} f(x)dx = 1$.

A random variable with a density function f means

$$P(\{\omega \in \Omega | X(\omega) \in A\}) = \int_A f(x)dx$$

for $A \subseteq \mathbf{R}$.

example: we say X is uniformly distributed on $[a, b]$ if it has density function

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } x \in [a,b] \\ 0 & \text{otherwise} \end{cases}$$

Alternatively, X is normally distributed with mean μ and variance σ^2 if

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Exercise: check $\int_{-\infty}^{\infty} f(x)dx = 1$

Defintion: *mean*, $\mathbf{E}(x) = \mu$, of a random variable. Discrete case: $\mu = \sum_{x \in \mathbf{R}} x \times m(x)$, continuous case: $\mu = \int_{-\infty}^{\infty} x \times f(x)dx$

$$\begin{cases} \sum_{x \in \mathbf{R}} x \times m(x) & \text{Discrete} \\ \int_{-\infty}^{\infty} x \times f(x) & \text{Continuous} \end{cases}$$

Definition: *Distribution of a continuous r.v.*: $F(x) = P(X(\omega) \leq x) = \int_{-\infty}^{x} f(\xi)d\xi$. So if $F(x)$ is normal with $\mathcal{N}(\mu, \sigma)$, μ is at the tip of the gaussian mountain, and μ has val 0.5 in the CDF, $F(x)$.

Let $\Phi : \mathbf{R} \to \mathbf{R}$. *Claim:*

$$E(\Phi(x)) = \begin{cases} \sum_{x \in \mathbf{R}} \Phi(x) m(x) & \text{Discrete} \\ \int_{-\infty}^{\infty} \Phi(x) f(x) dx & \text{Continuous} \end{cases}$$

This is not obvious.

Now how to define R.V, X, formally. $\Omega \to \mathbf{R}$. Using the example of 3 coin tosses, $\Omega = \{000, 001, \ldots, 111\}$. $X(\omega) =$ decimal version of binary value of ω. Then, $X(\omega) = [1, 7] \in \mathbf{N}$. Now, we can do: $\forall x \in \mathbf{R}$, define $m(x)$ by

$$m(x) = \begin{cases} 1/8 & \text{if } 1 \geq x \leq 7 \\ 0 & \text{otherwise} \end{cases}$$

Let us call 1 or more consecu, tive tosses of same type as *runs*. This is a new R.V. where Ω is the same, and $Y(\omega)$ =number of runs.

i.e. $Y(000) = 1 = \Phi(0), Y(001) = 2 = \Phi(1), Y(010) = 3 = \Phi(2), Y(011) = 2, \ldots$. Note $Y = \Phi(X)$. So $m_y(y) = P(Y = y)$, and $m_y(1) = 1/4, m_y(2) = 1/2, m_y(3) = 1/4, m_y(\text{else}) = 0$. Then,

$$\begin{aligned} \mathbf{E}(Y) &= \sum_{y \in R} y m_y(y) \\ &= 1/4 + 2(1/2) + 3(1/4) = 2 \end{aligned}$$

This is the same as

$$\begin{aligned} \sum_{x \in R} \Phi(x) m(x) &= (1 + 2 + 3 + 2 + 3 + 2 + 1)(1/8) \\ &= 16/8 = 2 \end{aligned}$$

6 September 20th Class 5

6.1 Continue on Probability

Given random variable X and $E(x) = \mu = \int_{-\infty}^{\infty} xf(x)dx$, the variance is $var(x) = \sigma = E[(X-\mu)^2] = E(X^2) - E(X)^2$ If x has a desity function $f(x)$, then $E(\phi(x)) = \int_{-\infty}^{\infty} \phi(x)f(x)dx$

6.2 Monte Carlo Integeration

Let X be a uniformly distributed random variable on $[0,1]$. If $\phi : [0,1] \to \mathbf{R}$, suppose we want an approximation to $\int_0^1 \phi(x)dx = I(\phi)$.

Sample X from $[0,1]$, and get $x_1, x_2, \ldots, x_n$. We'll approximate $I(\phi)$ by the average $\frac{1}{N}\sum_{i=1}^{N} \phi(x_i)$. This is the quadrature rule $Q_{MC,N}(\phi)$.

Notice: $\phi(x)$ is a random variable. The expected value of $\phi(x)$ is $E(\phi(x)) = \int_0^1 \phi(x)dx = I(\phi)$, because X is uniform. (so $f(x) = \begin{cases} 1 & \text{if } x \in [0,1] \\ 0 & \text{otherwise} \end{cases}$)

The approximation $Q_{MC,N}(\phi) = Q_N(\phi)$ is called the sample mean of $\phi(x)$. Approximates the mean of ϕ, which is $I(\phi)$ or $\mu(\phi)$. i.e. a trivial example if $\phi(x) = x$, then $I(\phi) = \int_0^1 xdx = \frac{1}{2}x^2|_0^1 = 1/2$

Contrast this with standard ways to do quadrature. *examples:*

- the trapezoidal rule: area under the curve is approximated by the trapezoid bewteen $f(a)$ and $f(b)$. $\int_a^b \phi(x)dx \approx 1/2(b-a)(\phi(a)+\phi(b))$
- the simpson's rule: $\int_a^b \phi(x)dx \approx 1/6(\phi(a) + 4\phi(\frac{a+b}{2}) + \phi(b))(b-a)$
- Composite trapezoidal rule on [0, 1]: $\approx h/2\phi(x_0) + h\sum_i^{N-1} f(x_i) + h/2\phi(x_N)$

6.2.1 Monte Carlo history

From high energy physics by John von Newmann and Stanislaw Ulam. Monte Carlo, an island in the medditteranian where people gamble.

6.3 Error Analysis

Simpson's rule is better. The error analysis of sympson's rule is: $I(\phi) - Q_s(\phi) \leq ch^4 = c1/N^4 = O(1/N^4)$

With trapezoid it's $O(1/N^2)$. Is this better than monte carlo?
In $1D$, monte carlo does worse.

Claim: error $|I(\phi) - Q_{MN,N}(\phi)|$" $\leq$ "$O(1/\sqrt{N})$. Suppose $N = 10000$, so $1/100$, but in simpson's rule's error, $O(1/10^{16})$. Simpson's does way better.

From probability analysis, using the Law of Large Numbers, if we let $S_N = \phi(x_1) + \cdots + \phi(x_N)$, and the sample mean $S_N/N = \mu_N$, (which is our approximated integral), by LNN, $\lim_{n\to\infty} P(|\frac{S_N}{N} - I(\phi)| < \epsilon) = 1$, where $I(\phi)$ is the real μ. That is $P(|\frac{S_N}{N} - I(\phi)| > \epsilon) \to 0$ as $N \to \infty$. In words, it's "very unlikely" that $Q_{MC,N}(\phi)$ is different from $I(\phi)$ when N is very large.

Chebyshev's inequality: Given $c > o$,

$$P(|\frac{S_N}{N} - \mu| \geq c) \leq \frac{\sigma^2}{Nc^2}$$

Commentary: suppose we want this probability to correspond to 95% confidence, that is the probability on the left to be .05, then we require $.05 \leq \frac{\sigma^2}{Nc^2}$, solve for c and we get $c^2 \leq \frac{\sigma^2}{.05N}$ or $c \leq \frac{\sigma}{\sqrt{.05N}}$. So with 95% confidence, $error \leq \frac{\sigma}{\sqrt{.05}}\frac{1}{\sqrt{N}}$ That's what we can say about the error. If we want 99% confidence, then $\sqrt{.05}$ becomes $\sqrt{.01}$

Two cons: convergence is quite a bit slower, we're not as confident as the other quadrature methods (the others give us a guarantee)

Suppose instead of uniform, say we had two box, N samples of 2-tuples, then with double integral and think about monte carlo. The procedure is the same, sample, evaluate the value at that sample, then take the average. Suppose we have a cube, same thing. Nothing of monte carlo is tied to the underlying domain. The analysis using LLN and chebyshev is always the same.

Consider 2D. Simpsons rule on $[a, b] \times [c, d]$.

$$\int_0^\phi \int_a^b \phi(x,y)dxdy \approx \text{let } \Phi(y) = \frac{b-a}{6}(\phi(a,y) + 4\phi(\frac{a+b}{2}, y) + \phi(b,y))$$
$$\approx \frac{d-c}{6}(\Phi(c) + 4\Phi(\frac{c+d}{2}) + \Phi(d))$$

, where $\Phi(\frac{c+d}{2}) = \phi(\frac{a+b}{2}, \frac{c+d}{2})$ Total number of points $N = n^2$, $h = 1/n = 1/\sqrt{N}$. The error is $O(h^4) = O(1/n^4) = O(1/N^2)$

In 3D, the error on simpsons is still $O(h^4)$, but $h = 1/n, N = n^3$, so this is $O(1/N^{4/3})$, in D dimensions, it's $O(1/N^{4/d})$)

In monte carlo, the error analysis is the same in any dimension. Has nothing to do with the integral. so the big pro is that we can never be certain but we're always working with the same error around $\frac{\sigma}{\sqrt{a}}\frac{1}{\sqrt{N}}$.

Now, which technique is better? If we ignore the uncertainty, we're asking when is $\frac{1}{N^{4/d}} < \frac{1}{N^{1/2}}$?

$$\frac{1}{N^{4/d}} < \frac{1}{N^{1/2}}?$$
$$N^{1/2} < N^{4/d}$$
$$\frac{1}{2} < \frac{4}{d}$$
$$d < \frac{4}{1/2} = 8$$

So when $d < 8$, simpson is better, else, MC is better. (Although we're ignoring the constant and the uncertainty factor so it won't be exact like this but still around there)

In model of particle physics, the number of dimensions is basically is equal to the number of particles, which could be hundreds. It's not unusual to have large d.

Example: of high dimensional integral (in the text/wiki). In particle physics, a *partition function* $Z(\beta)$ describes the statistical properties a system of d particles, thermodynamic equilibrium,
$Z(\beta) = \int exp(-\beta H(p_1, \cdots, p_d, x_1, \cdots, x_d) d^3p_1 \cdots d^3p_d d^3x_1 \cdots d^3x_d$,
where this integral is a $3d$ integral. where $d =$ number of particles, and $\beta = \frac{1}{\alpha\tau}$, a Boltzmann constant, τ is temperature, and H is hte hamiltonian function. And the point is that people care about this. This is about a 3 million dimensional integral.

7 September 22 Class 6

7.1 Continue on Monte Carlo

Monte Carlo Integration: Given a r.v. Y, $P(Y \subseteq A) = E(I_A(Y))$, where $I_A(Y) = \begin{cases} 1 & \text{if } Y \in A \\ 0 & \text{otherwise} \end{cases}$. And $E(I_A(Y)) = \int_A I_a(y)dy$, where the integral can be multi-dimensional. One way to deal with high-dimensional integral is to use Monte Carlo Integration.

Review: Monte Carlo gives us the approximation $\int_{\mathbf{R^D}} \phi(x)dx = I(\phi)$, where $I(\phi) \approx 1/N \sum_{n=1}^{N} Q(x_n) = Q_{MC,N}(\phi)$ where $\{x_n\}$ are samples of randome variable X.
We showed that the error was $|I(\phi) - Q_{MC,N}(\phi)|" \leq " \frac{\sigma}{c\sqrt{N}}$
If the goal is to make this bound $\leq \tau$ tolerance, i.e. $\frac{\sigma}{c\sqrt{N}} \leq \tau$, then we need

$$\begin{aligned} \sqrt{N} &\leq \frac{\sigma}{c\tau} \\ N &\leq \frac{\sigma^2}{c^2\tau^2} \end{aligned}$$

many samples.

7.2 Variance Reduction Methods

The aim is to reduce σ 2 examples of idea to do this

7.2.1 Antithetical variables

Assume $\phi \in [0,1]$, and $I(\phi) = \int_0^1 \phi(x)dx = E(\phi(x))$ (remember !!) where X is a r.v. from a uniform distribution on $[0,1]$. ($E(x) = \int_0^1 xdx = 1/2x^2|_0^1 = 1/2$, and we approximate $I(\phi)$ with $1/N \sum_n = 1^N Q(x_n)$).
Antithetical variables are x, and x^c s.t. $x + x^c = 1$ or $x^c = 1 - x$, $x = \mu + d$, $x^c = \mu - d$

Assume σ small. Write $d = \sigma\hat{d}$ (d will not grow too much because they're from a bounded distribution..) Let $D = X - \mu$, another r.v., and let $\hat{D} = D/\sigma$ also a r.v. Then,

$$\begin{aligned} E(\hat{D}) &= \frac{1}{\sigma}E(D) \\ &= \frac{1}{\sigma}E(x - \mu) \\ &= \frac{1}{\sigma}[E(x) - E(\mu)] \\ &= \frac{1}{\sigma}[\mu - \mu] = 0 \end{aligned}$$

Now, consider $\phi(x)$. (he used $1/2$ as μ following the example on board)

$$\begin{aligned}\phi(x) &= \phi(\mu + d) = \phi(\mu + \sigma\hat{d}) \\ &= \phi(\mu) + \phi'(\mu)\sigma\hat{d} + \phi''(\xi)\sigma^2\hat{d}^2 \text{ by taylor series} \\ &\approx \phi(\mu) + \phi'(\mu)\sigma\hat{d} + O(\sigma^2)\end{aligned}$$

And $\phi(x^c)$

$$\begin{aligned}\phi(x^c) &= \phi(\mu - d) = \phi(\mu - \sigma\hat{}) \\ &\approx \phi(\mu) - \phi'(\mu)\sigma\hat{d} + O(\sigma^2)\end{aligned}$$

SO,

$$\frac{\phi(x) + \phi(x^c)}{2} = \phi(\mu) + O(\sigma^2)$$

Now look at $I(\phi) = E(\phi(x)) = \int_0^1 \phi(x)dx$. This is

$$\begin{aligned}&= E[\phi(\mu) + \phi'(\mu)\sigma\hat{D} + O(\sigma^2)] \\ &= \phi(\mu) + \phi'(\mu)\sigma E[\hat{D}] + O(\sigma^2) \\ &= \phi(\mu) + 0 + O(\sigma^2)\end{aligned}$$

Now a new integration method. Smaple X N times and get $x_1, \dots, x_N$, also let $x_1^c, \dots, x_N^c$. Let the complementary quantities:

$$Q^c_{MC,N}(\phi) = \frac{1}{2N}(\phi(x_1) + \phi(x_1^c) + \cdots + \phi(x_n) + \phi(x_n^c))$$

For each n,

$$\begin{aligned}\phi(x_n) &= \phi(\mu + \sigma\hat{d}_n) = \phi(\mu) + \phi'(\mu)\sigma\hat{d}_n + O(\sigma^2) \\ \phi(x_n^c) &= \phi(\mu - \sigma\hat{d}_n) = \phi(\mu) - \phi'(\mu)\sigma\hat{d}_n + O(\sigma^2) \\ 1/2(\phi(x_n) + \phi(x_n^c)) &= \phi(\mu) + O(\sigma^2) \\ \Rightarrow Q^c_{MC,N}(\phi) &= \frac{1}{N}(N\phi(\mu) + N(\phi(\sigma^2)) = \phi(\mu) + O(\sigma^2)\end{aligned}$$

Now the difference: $I(\phi) - Q^c_{M,N}(\phi) = \phi(\mu) - \phi(\mu) + O(\sigma^2)$ So **the error is 1/4th the size** or σ^2 instead of σ. Smmmary:

Method 1 Take N samples, error $\frac{\sigma}{\sqrt{N}}$

Method 1 Take $2N$ samples, error $\frac{\sigma}{\sqrt{2N}}$

Method 2 Take $N + N_{\text{antithetical}}$ samples, error $\frac{\sigma^2}{\sqrt{N}}$

7.2.2 Importance Sampling

$I(\phi) = \int_0^1 \phi(x)dx$, written in a different way is

$$\int_0^1 \frac{\phi(x)}{p(x)} p(x)dx$$

Where p is a positive function on $[0,1]$ s.t. $\int_0^1 p(y)dy = 1$. Notice $p(x)$ is a density function.

Suppose we can randomly sample variable x from the distribution defined by p. This means $P(X \in A) = \int_A p(x)dx$.

Rule:

1. Sample $x_1, x_2, \ldots, x_n$
2. Form the MC cost estimate, $I_N(\phi) = \frac{1}{N}\sum_{n=1}^{N} \frac{\phi(x_n)}{p(x_n)}$

The variance of this process is

$$\hat{\sigma}^2 = \int_0^1 (\frac{\phi(x)}{p(x)} - I(f))^2 p(x)dx$$

Error is proportional to $\frac{\hat{\sigma}}{\sqrt{N}}$ We want p s.t. $\hat{\sigma} < \sigma_\phi$ $I(f)$ is just a number. We would like p close to $\frac{\phi}{I(\phi)}$. But we don't know what $I(\phi)$ is.

8 September 27th Class 7

8.1 Continue on Importance Sampling

For variance reduction, we're computing $\int_0^1 \phi(x)dx$, MC says that this is approximated by $\frac{1}{N}\sum_i^N \phi(x_i)$. We'll write the integral as $I(\phi) = \int_0^1 \frac{\phi(x)}{p(x)}p(x)dx$, where p is positive and $\int_0^1 p(x) = 1$, i.e. $p(x)$ is a density function.

We know that the error for MC is $\frac{\sigma_\phi^2}{\sqrt{N}}$. Supposition is that we can sample $\{x_i\}$ from a distributionwith density $p(x)$, and sampling will approximate $I(\phi)$ by $\frac{1}{N}\sum_i^N \frac{phi(x_i)}{p(x_i)}$. The variance of this process: $\hat{\sigma}^2 = \int_0^1 (\frac{\phi(x)}{p(x)} - I(\phi))^2 p(x)dx$, and the error $\frac{\hat{\sigma}^2}{\sqrt{N}}$, and $\hat{\sigma} < \sigma_p$. We need p and be able to sample from it to do this.

We need a way to define $p(x)$ and a technique to sample from it.

8.1.1 Technique: Accept/Reject sampling method

Supposed we have $q(x) \geq p(x)$ (q could be a constant too). Let $\hat{q}(x) = \frac{q(x)}{\int_0^1 q(x)dx} = \frac{q(x)}{I(q)}$, and so $I(\hat{q}) = 1$ (scaled version of q). (in the hw $\hat{q}$ is just a constant)

Algorithm to draw random numbers with density $p(x)$.

1. Pick two random numbers x' with density $\hat{q}$ and y from uniform distribution [0,1].

2. Then make a decision: Accept x' as one of my samples if $y \leq \frac{p(x')}{q(x')}$, reject otherwise.

3. *claim* the number of accepted samples is approximately $\frac{1}{I(q)}$

Why does this work?

Consider a function $C(\xi) = C_\alpha(\xi) = \mathbf{X}(\xi \leq \alpha)\begin{cases} 1 & \text{is } \xi \leq \alpha \\ 0 & \text{otherwise} \end{cases}$. For $\alpha \in [0,1]$, $\int_0^1 C_\alpha(\xi)d\xi =$ area under line from 0 to α, which is α. Now..use

$$C(\xi) = C_{\frac{p(x)}{q(x)}}(\xi) \tag{11}$$

where x is given.

We want to know if

$$\mathbf{E}(X) \approx \int_0^1 xp(x)dx = \frac{1}{N}\sum_j^N x_j \tag{12}$$

(we can take out $p(x)$ because of the way we sampled x_j). Notice $p(x) = \frac{p(x)}{q(x)}q(x) = \frac{p(x)}{q(x)}\hat{q}(x)I(q)$, now using 11, this is the same as $\int_0^1 C_{\frac{p(x)}{q(x)}}(\xi)d\xi\hat{q}(x)I(q)$.

Then,

$$\begin{aligned}\int_0^1 xp(x)dx &= \int_0^1 xdx \int_0^1 C_{\frac{p(x)}{q(x)}}(\xi)d\xi\hat{q}(x)I(q) \\ &= \int_0^1 \int_0^1 xC_{\frac{p(x)}{q(x)}}(\xi)\hat{q}(x)d\xi dxI(q) \\ &\approx \frac{1}{N'}\sum x_iC_{\frac{p(x)}{q(x)}}(\xi_i)I(q) \text{ this step was doing MC to estimate}\end{aligned}$$

Using N' samples from the distribution for x taken from $\hat{q}$, and ξ, taken from uniform dist. $C(\xi_i) = 1$ if $\xi_i \leq \frac{p(x)}{q(x)}$, 0 otherwise. So the entire thing is equal to

$$\frac{1}{N}\sum_{\text{indicies of accepted samples}} x_i$$

Using the claim that $\frac{N=\#\text{ accepted}}{N'=\#\text{ sampled}} = \frac{1}{I(q)}$ $(\frac{I(q)}{N'} = \frac{1}{N})$ So this is what we wanted 12.

Proof of the claim: Rule is accept if $y \leq \frac{p(x')}{q(x')}$, which is $yq(x') \leq p(x')$. For a given y, P(resulting experiment leads to accept x') is deteremined by integrating $y\int_0^1 q(x')dx' \leq \int_0^1 p(x')dx'$. So

$$y \leq \frac{I(p)}{I(q)} = \frac{1}{I(q)}$$

because p is a density function.

Then P(experiment leads to acceptance) $= P$(sample y from uniform $\leq \frac{1}{I(q)}$) $= \frac{1}{I(q)}$ (since $\int_0^{\frac{1}{I(q)}} dy = \frac{1}{I(q)}$

This probability is approximately just $\frac{N}{N'} \approx \frac{1}{I(q)}$, which is our claim! (Reference R. Caflisch Acta Numerica 1:49, 1998 Monte Carlo and quasi-Monte Carlo methods)

8.2 Matrix Factorization Methods

Four basic algorithms:

1. Gaussian Elimination (solving $Ax = b$ where A is nonsingular)

2. QR Factorization (same, or solving least squares $\min||Ax - b||$ where A is a long thin $m$by$n$ matrix, $m >> n$)

3. Singular value decomposition (also for least squares)

4. Eigenvalue decomposition (SVD and this are used to solve eigen value problems, $Av = \lambda v$, find v, λ

9 October 4th Class 8

Corrections for hw 2:

- Problem 1(a): $f(X) \in [0,1]$, not $[-1,1]$
- Problem 3: should read $\rho(x) = \frac{4}{10}\hat{\rho}(x)$ because we need $\int_0^1 \rho(x)dx = 1$. Also $q(x) = 1.6$ constant. When sampling, $q(x)$ induces a density function $\hat{q}(x)$ by $\hat{q} = \frac{q}{\int q(x)dx} = \frac{1.6}{1.6}$, which means that we still sample from $U(0,1)$, but when you compare the ratio p/q, we use 1.6.
- For histogram & density estimation: If we have a pdf $p(x)$, the $\int_{D\ X} p(x)dx = 1$. For r.v. X with density p, then $P(\alpha \leq X \leq \beta) = \int_\alpha^\beta p(x)dx$.

  If we're sampling $\{x_i\}$ from such an X(using reject/accept), then the number of num of $x_i \in [\alpha, \beta]$/num of samples $\approx \int_\alpha^\beta p(x)dx$. This is called the kernel density estimate. The goal here is to approximate $p(x)$ by some constant $p_c(x)$ on $[\alpha, \beta]$. If we had p_c, then $\int_\alpha^\beta p(x) \approx p_c[\beta - \alpha]$. Turn this around to define p_c as

  $$p_c = \frac{\#\text{ of } x_i \in [\alpha, \beta]}{\#\text{ of samples}} \frac{1}{\beta - \alpha}$$

  Given x, let $[\alpha_j, \beta_j]$ be the interval containing x, define $p_c(x) =$ number of $x_i \in [\alpha_j, \beta_j]$/number of samples$\frac{1}{\beta_j - \alpha_j}$, and this is the histogram he's looking for. Does it come out the way it should?

  In matlab use *hist*, and if you really want to compare/plot p_c, specify bin set `hist(X, [1/8,3/8/,5/8,7/8])`, it will give you exactly what you want (same centers). Better yet, let that hist be z and do a bar graph `bar(z, centers)`.
- For prob 1.b, do log(error)vs$log(N)$ scale. Second graph is to ask whether things really look clean (no, not really bc of the probabilistic nature). Because if we were to use trapezoid, we'll get a very clean ratio between the error.
- For prob 1.c, 3 sets, each of 4 pictures.. sample E_N n times, try different sets of ns. (I did 1000, use that for $\frac{\sigma}{n}$..?, but try different samples).

9.1 Matrix Factorization

9.1.1 Gaussian Elimination

Solve $Ax = b$, where A $NbyM$ is non-singular. Procedure also called LU factorization.

1. Add multiple of first row of A and first entry of b to the second to the last rows to produce zeroes on the first column of A except the first entry. We get $\hat{A} = L_1 A$
2. repeat this, then $L_{n-1} \cdots L_1 = U$ the upper triangle.

Cost: number of floating point multiplication and divisions:

- $n-1$ for column of L_1.
- $(n-1)(n-1)$ to produce $\hat{A}$
- Just the operation on A is $n(n-1)$ for step 1.
- plus $(n-1)(n-2)$ for step 2
- $\cdots$ and we get $(n-(n-2))(n-(n-1)) = 2$ at step $n-1$

So total is

$$\begin{aligned}
\sum_{j=1}^{n-1} j(j+1) &= \sum_{j=1}^{n-1} j^2 + \sum_{j=1}^{n-1} j \\
&= \sum_{j=1}^{n-1} j^2 + \frac{n(n-1)}{2} \\
&= O(\frac{1}{3}n^3) + n^2 = O(n^3)
\end{aligned}$$

Exercise: Identify how to get x, *back substitution* and what's the cost? ($O(n^2)$)

What happens if $a_{00} == 0$? Since A nonsingular, we know that there must be at least one $a_{j1} \neq 0$. Take the j with largest $|a_{j1}|$ and switch it with a_{11}. Formally, the switch is done by multiplying the system by P_1, a matrix with all diagonal values 1 but a_{11} and a_{jj} and all 0 but a_{0j} and a_{j1}., the permutation matrix.

So result is $L_1P_1Ax = L_1P_1b$, where L_1 is taken from P_1A. This is called *pivotting*.

10 October 6th Class 9

10.1 More Linear Algebra

10.1.1 Cont on Gaussian Elimination

Solving $Ax = b$, A nonsingular. Without pivoting, $L_{n-1}\cdots L_1 A = U$, where $L_{n-1}\cdots L_1 = L^{-1}$ Claim: L^{-1} and L are lower triangulare

- product of lower triangular matrices is a lower triangular.
- Inverse of lower triangular matrices is lower triangular.

So we have

$$A = LU$$

Formally, solving $Ax = b$ consists of forming $U = L^{-1}A$ (step 1a) and $\hat{b} = L^{-1}b$ (step 1b) and solving $Ux = \hat{b}$ for x (step 2).

We showed last time that (1a) costs $O(n^3/3)$. Claim: cost of (1b) is $O(n^2)$ and cost of (2) is $O(n^2)$.

Step 1 is $x_n = b_n/U_{n,n}$ Idea of solving $Ux = \hat{b}$ to do row j

```
for j=n:-1:1
   x_j = (\hat b_j - \sum_{l=j+1}^n U_{j,l}x_l)U_{j,j}
end
```

For each step j, cost is: $n - j + 1$ plus 1 for divide So the total cost is $\sum_{j=1}^{n}(n - j + 1)$, $n + 1 - 1 + n + 1, -2 + \cdots + n + 1 - n = n + n - 1 \cdots + 1$ so total $\frac{n(n+1)}{2}$

One thing we missed is what's the entries of matrix L look like. We know U, it's what's been left behind. *Claim*: the entries of L are - cost of L_i's that are constructed. i.e. first column is $\begin{matrix} 1 \\ a_{21}/a_{11} \\ a_{31}/a_{11} \\ \vdots \\ a_{nn}/a_{11} \end{matrix}$

To do this correctly, we may need to *pivot*. When we pivot in floating point, we compute a solution $\hat{x}$ that satisfies $(A + \delta A)\hat{x} = b$, where $\frac{||\delta A||}{||A||}$ is small. What we looked at in week 1, $\frac{||x-\hat{x}||}{||x||} = ||A||||A^{-1}||\frac{||\delta A||}{||A||} = \kappa(A)$

Now, look at the relative error

$$\frac{||x - \hat{x}||}{||x||} = \frac{||x - \hat{x}||}{||\hat{x}||}\frac{||\hat{x}||}{||x||}$$

$$\begin{aligned}
Ax &= b \\
(A + \delta A)\hat{x} &= b \\
A\hat{x} &= b - (\delta A)\hat{x} \\
\hat{x} &= A^{-1}b - A^{-1}(\delta A)\hat{x} \\
&= x - A^{-1}(\delta A)\hat{x} \\
||\hat{x}|| &\leq ||x|| + ||A^{-1}(\delta A)\hat{x}|| \text{ take the norm} \\
&\leq ||x|| + ||A^{-1}||||(\delta A)||||\hat{x}|| \\
||\hat{x}|| - ||A^{-1}||||(\delta A)\hat{x}|| &\leq ||x|| 1 - \kappa(A)\frac{||\delta A||}{||A||}||\hat{x}|| \qquad \leq ||x||
\end{aligned}$$

$\kappa(A)$ is a property of our matrix A. In numerical computing, we can only work with A with reasonable conditioning number.
Put things together we have

$$\frac{||x-\hat{x}||}{||x||} \le \frac{\kappa(A)\frac{||\delta A||}{||A||}}{1-\kappa(A)\frac{||\delta A||}{||A||}}$$

The way to think bout hte denominator: $\frac{||\delta A||}{||A||}$ small means aroung 10^{-12} 10^{-14} 10^{-16}. Suppose $\kappa(A) = 500$, then the denominator is $1 - 500(10^{-12}) = 1 - 5(10^{-10})$ not that bad at all. In fact, we can deal up till like 5000 can handle it until 10^{10} (for this proble)

10.2 Least Squares Problem

Given data $\{(x_i, y_i)\}$, find a simple function $p(x)$ s.t. $p(x_i) \approx y_i$. Example of such a function is $p(x) = \alpha_0 + \alpha_1 x$ a linear polynomial.
A least squares fit is to find α_0 and α_1 s.t. sum of squares of $y_i - (\alpha_0 + \alpha_1 x_i)$ is minimized. $min \sum_{i=1}^{n}[y_i - (\alpha_0 + \alpha_1 x_i)]^2$ Note: if we tried to make $p(x_i) = y_i$ for all i, we have

$$\begin{aligned}\alpha_0 + \alpha_1 x_1 &= y_1\\ \alpha_0 + \alpha_1 x_2 &= y_2\\ &\vdots\\ \alpha_0 + \alpha_1 x_m &= y_m\end{aligned}$$

m equations and 2 unknowns. Write in matrix form:

$$\begin{pmatrix}1 & x_1\\ 1 & x_2\\ \vdots & \\ 1 & x_m\end{pmatrix}\begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix} = \begin{pmatrix}y_1\\ \vdots\\ y_m\end{pmatrix}$$

Write this $Aa = y$. In general, there is no solution. Gvien any $\alpha = {\alpha_0 \atop \alpha_1}$, let $r = y - Aa$, a vector of length m.

Notice $r_i = y_i - [Aa]_i = y_i - (\alpha_0 + \alpha_i x_i)$. We can restate the goal as to minimize $\sum_i^n r_i^2 = ||r||_2^2$

11 October 11th Class 10

11.1 Continue on Least Squares

Given $A \in \mathbf{R}^{m\times n}$, $m > n$, full-rank, $y \in \mathbf{R}^m$, we want to **find** $a \in \mathbf{R}^n$ **s.t.** $||y - Aa||_2$ **is minimal.**

Example: Data $(x_1, y_1), \dots, (x_m, y_m)$, we want to find a polynomial $p_1(x) = a_0 + a_1 x$ s.t. $p(x_i) \approx p(y_i)$.

$$\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

Geometric Interpretation (m=3): $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. see notebook. Projection of $\vec{y}$ on the plane spanned by $\vec{1}$ and $\vec{x}$ is the solution $a_0 \vec{1} + a_1 \vec{x} = Aa$ that minimizes $||y - Aa||_2$.

From this geometric argument, we have $y - Aa \perp$ plane, which is the set of linear combination of columns of A. That is:

$$y - Aa \perp range(A)$$

where $range(A) = \{Aw | w \in \mathbf{R}^n\}$ So

$$< Aw, y - Aa >= 0$$
$$< w, A^T(y - Aa) >= 0 \ \forall w \in \mathbf{R}^n$$

For any $\vec{w}$ we get range of A Take for example of $\vec{w} = A^T(y - Aa) \in \mathbf{R}^n$ That is,

$$\begin{aligned} < A^T(y - Aa), A^T(y - Aa)) > &= 0 \\ |A^T(y - Aa)||_2^2 &= 0 \\ &\Rightarrow A^T(y - Aa) = 0 \text{ or} \\ A^T Aa &= A^T y \end{aligned}$$

$A^T Aa = A^T y$ is the normal equations. $A^T A$ (squre, $n \times n$) is non-singular (because A is full-rank). We can solve this by gaussian elimination. We don't want to because of conditioning issues imposed by gaussian elimination.

11.2 QR Decomposition

Definition: A matrix $Q \in \mathbf{R}^{m \times m}$ is *orthogonal* if $Q^T Q = I$

Theorem: For any $A \in \mathbf{R}^{m \times n}$, $m > n$, $\exists$, an orthogonal matrix, $Q \in \mathbf{R}^{m \times m}$, and an upper triangular matrix $R \in \mathbf{R}^{m \times n}$ s.t. $A = QR$

To minimize $||y - Aa||_2^2$, we want

$$\begin{aligned} < y - Aa, y - Aa > &=< y - QRa, y - QRa > \\ &=< QQ^T y - QRa, QQ^T y - QRa > \text{ since } QQ^T = I \\ &=< Q(Q^T y - Ra), Q(Q^T y - Ra) > \\ &=< Q^T y - Ra), (Q^T y - Ra) > \text{ left mult} Q^T \\ &= ||\hat{y} - Ra||_2^2 \end{aligned}$$

Where $\hat{y} = Q^T y$. To minimize $||\hat{y} - Ra||_2$,

$$|| \begin{pmatrix} \hat{y}_{1:N} \\ \hat{y}_{n+1:m} \end{pmatrix} - \begin{pmatrix} R_1 a \\ 0 \end{pmatrix} ||$$

Where R_1 is the upper $n \times n$ subblock of R. Solve $R_1 a = \hat{y}_{1:n}$. We can't do anything about $\hat{y}_{n+1:m}$. R_1 is an upper triangular, it's non-singular because A is full-rank (this is a *claim:* if A if full-rank then R_1 is non-singular). We can do back-substitution!

Pf of claim 1:

$$\begin{aligned} Q &= QQ^TQ = QI \\ &= (QQ^T)Q \text{ mult right by } Q^{-1} \\ QQ^{-1} &= (QQ^T)QQ^{-1} \\ I &= QQ^T \end{aligned}$$

So to solve least squares, do the QR decomposition, then solve $R_a a = \hat{y}_{1:n}$. So to get a, we need R_1 and the first n entries of $\hat{y}$.

So lets write Q as $[Q_1|Q_2]$, where Q_1 is $m \times n$, Q_2 is $m \times (m-(n+1))$. Then, $Q^Ty = \begin{pmatrix} Q_1^T y \\ Q_2^T y \end{pmatrix}$, where $Q_1^T y$ is $\hat{y}_{1:n}$. So all we need is $Q_1 R_1$. But if we want to know the minimum values of least squares, you need all of $\hat{y}$, this will only get us $\hat{y}_{1:n}$.

11.3 Why QR and not LU?: Roundoff effects

Given Q (Q_1), R (R_1), **Effects of Roundoff**: With LU: *Recall from last week*: for solving $Ax = b$ for A $n \times n$ nonsingular, we showed that $\frac{||x-\hat{x}||}{||x||} \leq \kappa(A)\frac{||\delta A||}{||A||}/1-\xi$, ξ very small and irrelevant ($\frac{||\delta A||}{||A||}$ is also small)and $\hat{x}$ solves $(A+\delta A)\hat{x}) = b$.

Consider the normal equation: $A^TAa = A^Ty$. Last week's analysis gives us

$$\frac{||a-\hat{a}||}{||a||} \leq \kappa(A^TA)\frac{||\delta(A^TA)||}{||A^TA||}$$

Claim: if A were square, then $\kappa(A^TA)$ is $\kappa(A)^2$.

If we use QR decomposition to solve the least squares problem, then the analogous bound will be

$$\frac{||a-\hat{a}||}{||a||} \leq [\kappa(A^TA)\frac{||y-Aa||}{||Aa||} + \sqrt{\kappa(A^TA)}]\frac{||\delta A||}{||A||}$$

So only if $||y - Aa||$ is small, effect of $\kappa(A^TA)$ is negligible, only bad term is $\sqrt{(}\kappa(A^TA)$, but this is just $\kappa(A)$. $\hat{a}$ is the least squares solution to minimize $||(A+\delta A)\hat{a} - y||_2$.

12 Class 11 October 13th

Midterm: October 27th

1. some kind of analysis, a program provided to do something or us program something..
2. no proving, but operation counts or reasoning

Projects: use some of the ideas we explored. Testing it against an application, design, apply, write it up. A month work.

12.1 QR Factorization

$A \in \mathbf{R}, m \leq n$, full rank. Claim: $\exists Q$, orthogonal, and R upper triangular s.t. $A = QR$ where Q n by n, R m by n, the upper n by n part upper triangular= R_1, rest all 0.

How to make Q?

12.1.1 Method 1: Gram-Schmid Orthogonalization

For Q_1 and R_1 only here. Steps:

1. set Column 1 of A, $a_1 = q_1 r_{11}$. We need $q_1^T q_1 = 1$ (col of Q is has norm 1). We can get this by taking $\hat{q}_1 = a_1$, then $q_1 = \frac{\hat{q}_1}{||\hat{q}_1||} = \frac{a_1}{||a_1||}$ Let $r_{11} = ||a_1||$, then $a_1 = r_{11} q_1$
2. Require $a_2 = q_1 r_{12} + q_2 r_{22}$ at this point we know q_1 and a_2. we have two conditions: 1. $q_1^T q_2 = 0$ and 2. $q_2^T q_2 = 1$

   For 1: take $\hat{q}_2 = a_2 - q_1 r_{12}$, then impose $\langle \hat{q}_2, q_1 \rangle = 0$ so $\langle \hat{q}_2, q_1 \rangle = \langle a_2, q_1 \rangle - r_{12} \langle q_1, q_1 \rangle$ but $\langle q_1, q_1 \rangle = 1$ and we know $\langle a_2, q_1 \rangle$. So $r_{12} = \langle a_2, q_1 \rangle$ so now $\hat{q}_2$ is defined.

   For 2: take $q_2 = \frac{\hat{q}_2}{||\hat{q}_2||}$ so $r_{22} = ||\hat{q}_2||$, this gives $a_2 = q_1 r_{12} + q_2 r_{22}$.
3. (Part of step 3): $a_3 = q_1 r_{13} + q_2 r_{23} + q_3 r_{33}$, and force $\langle q_1, q_3 \rangle = 0$, $\langle q_2, q_3 \rangle = 0$, $\langle q_3, q_3 \rangle = ||q_3|| = 1$. Define $\hat{q}_3 = a_3 - q_1 r_{13} - q_2 r_{23}$, impose the condition with $\hat{q}_3$, then solve for q_3 using the definition of $\hat{q}_3$.

Pseudocode:

```
for $k = 1 \to n$ do
    $\hat{q}_k \leftarrow a_k$
    for $i = 1 \to k - 1$ do
        $r_{i,k} = \langle q_i, a_k \rangle$
        $\hat{q}_k \leftarrow \hat{q}_k - q_i r_{ik}$
    end for
    $r_{kk} = ||\hat{q}_k||$
    $q_k \leftarrow \hat{q}_k / r_{kk}$
end for
```

To solve min $||y - Aa||$, we know solve $R_1 a = Q_1^T y$, for a. Can we get all the residual just from this? We said no because otherwise we can't calculate the residual but this isn't true. Once we have a, it's an exact solution or it isn't. if it's not, $||y - Aa||$ is how big the residual is. We don't need Q_2! But there must be, so think about it..

If we want all of $Q = [Q_1; Q_2]$ A different way of doing it:

12.1.2 Householder Matrix

Given $\vec{v} \in \mathbf{R}^m$, Define a special orthogonal matrix

$$P = I - 2\frac{\vec{v}\vec{v}^T}{\vec{v}^t\vec{v}}$$

(a function of $\vec{v}$) Claim: P is symmetric, $P = P^T$, ("obvious") and orghotonal

$$\begin{aligned}
P^TP &= (I - 2\frac{\vec{v}\vec{v}^T}{\vec{v}^t\vec{v}})(I - 2\frac{\vec{v}\vec{v}^T}{\vec{v}^t\vec{v}}) \\
&= I - 4\frac{\vec{v}\vec{v}^T}{\vec{v}^t\vec{v}} + 4\frac{(\vec{v}\vec{v}^T)(\vec{v}\vec{v}^T)}{(\vec{v}^t\vec{v})^2} \\
&= I - 4\frac{\vec{v}\vec{v}^T}{\vec{v}^t\vec{v}} + 4\frac{(\vec{v}\vec{v}^T)}{(\vec{v}^t\vec{v})} \text{ insides cancel} \\
&= I
\end{aligned}$$

Given a vector $\vec{x}$, we want to find another vector $\vec{v}$ s.t. P generated by v satisfies $P_v\vec{x} = (\overset{e_1}{\begin{smallmatrix}0\\ \vdots\\ 0\end{smallmatrix}})$. Zero below the first entry. Supposed I had this. Then (looking for $\vec{v}$),

$$\begin{aligned}
P_vx &= (I - 2\frac{\vec{v}\vec{v}^T}{\vec{v}^t\vec{v}})\vec{x} \\
&= \vec{x} - 2\frac{\vec{v}^T\vec{x}}{\vec{v}^t\vec{v}})\vec{v} \\
&= \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} - c\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix}
\end{aligned}$$

We need $x_i = cv_i$, $c \in \mathbf{R} \forall i = 2 \ldots m$ This tells us that $v = c(x + \alpha e_1)$, take $c = 1$ and want $\vec{v} = \vec{x} + \alpha e_1$.

For this $\vec{v}$:, $v^Tx = x^Tx + \alpha x_1$ since $e_1^Tx = x_1$ Then $vv^T = \langle x + \alpha e_1, x + \alpha e_1 \rangle = x^Tx + 2\alpha x_1 + \alpha^2$ So

$$\begin{aligned}
2\frac{v^Tx}{v^Tv} &= 2\frac{x^tx + \alpha x_1}{x^Tx + 2\alpha x_1 + \alpha^2} \\
&\text{take } \alpha^2 = x^Tx = ||x|| \\
&= 2\frac{x^tx + ||x||x_1}{2(x^Tx + ||x||x_1)} \\
&= 1
\end{aligned}$$

Then $x - v = x - (x + \alpha e_1) = -\alpha e_1$.

We've shown that given $\vec{x}$ we can find a $\vec{v}$ that does this. Big picture, do this to every column of A, then we get bunch of P which will be our Q! then the remainder gives us upper triangular matrix R.

13 Class 11 October 18th 2011

13.1 Eigenvalue Problems

$A \in \mathbf{R}^{n \times n}$, find $\vec{v} \in \mathbf{R}^n$ (or $\mathbf{C}^n$) and a scalar λ (possibly complex) s.t.

$$A\vec{v} = \lambda v$$

Example of source: consider systems of ordinary differntial equations of order n, $\frac{dx}{dt} = Ax$, $x = [x_1(t), x_2(t), \cdots, x_n(t)]^T$, where A is n by n matrix. $\frac{dx}{dt}$ is a vector whos first entry is the first row of A times $x_1(t)$. $\frac{dx_1}{dt} = a_{11}x_1 + \cdots + a_{1n}x_n$, $\frac{dx_2}{dt} = a_{21}x_2 + \cdots + a_{2n}x_n$ and so forth.

If A has n linearly independent eigenvectors, $v_1, v_2, \ldots, v_n$, with associated eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n.(Av_i = \lambda_i v_i$. Then $x(t)$ can be written as linear combination of these eigenvectors (because they form the basis of $\mathbf{R}^n$. i.e. $x(t) = g_1(t)v_1 + g_2(t)v_2 + \cdots + g_n(t)v_n$.

The left hand side of $\frac{dx}{dt} = Ax$ is now:

$$\frac{dx}{dt} = \frac{dg_1}{dt}v_1 + \frac{dg_2}{dt}v_2 + \cdots + \frac{dg_n}{dt}v_n$$

The right hand side is:

$$\begin{aligned} Ax &= g_1(t)Av_1 + g_2(t)Av_2 + \cdots + g_n(t)Av_n \\ &= g_1(t)\lambda_1 v_1 + g_2(t)\lambda_2 v_2 + \cdots + g_n(t)\lambda_n v_n \end{aligned}$$

Now setting them equal to each other, we have $\frac{dg_i}{dt} = g_i(t)\lambda_i$. so $g_i = c_1 e^{\lambda_i t}$, where c_i some constant. (because the der of $\frac{d(c_1 e^{\lambda_i t})}{dt} = c_1 e^{\lambda_i t}\lambda_1 = g_1\lambda_1$) So the solution is

$$x(t) = c_1 e^{\lambda_1 t}v_1 + c_2 e^{\lambda_1 t}v_2 + \cdots + c_n e^{\lambda_1 t}v_n$$

13.2 How to compute eigenvalues?

Solve the characteristic polynomial: if $Av = \lambda v$, then $(A - \lambda I)v = 0$. Which means $(A - \lambda I)v = 0v$, i.e. $(A - \lambda I)$ has a 0 eigen value i.e is singular. Therefore, $det(A - \lambda I)$ is 0. $det(A - \lambda I)$ is a polynomial of degree n, the *characteristic polynomial*.

There is no finite form algorithm to compute the eigenvalues if $n \geq 5$, from Galois theory. (can't find roots of polynomial order bigger than 5).

13.2.1 Power method

assume A has n linearly independent eigenvectors $(v_1, v_2, \cdots, v_n,\ \lambda_1, \cdots, \lambda_n)$. Also assume $|\lambda_1| > |lam_2| \geq \cdots \geq |\lambda_n|$, and we call λ_1 the *dominant* eigenvalue.

Any $w \in \mathbf{R}^n$ can be written as a linear combination of these eigen vectors. $w = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$. So

$$\begin{aligned} Aw &= \alpha_1 Av_1 + \cdots + \alpha_n Av_n \\ &= \alpha_1\lambda_1 v_1 + \alpha_2\lambda_2 v_2 + \cdots + \alpha_n\lambda_n v_n \\ &= \lambda_1(\alpha_1 v_1 + \alpha_2\frac{\lambda_2}{\lambda_1}v_2 + \cdots + \alpha_n\frac{\lambda_n}{\lambda_1}v_n) \end{aligned}$$

We know $(|\frac{\lambda_i}{\lambda_1}|)^2 < |\frac{\lambda_i}{\lambda_1}| < 1 \; \forall i > 1$ and Then

$$A^2w = A(Aw) = \lambda_1^2(\alpha_1 v_1 + \alpha_2(\frac{\lambda_2}{\lambda_1})^2 v_2 + \cdots + \alpha_n(\frac{\lambda_n}{\lambda_1})^2 v_n)$$
$$A^k w = \lambda_1^k(\alpha_1 v_1 + \alpha_2(\frac{\lambda_2}{\lambda_1})^k v_2 + \cdots + \alpha_n(\frac{\lambda_n}{\lambda_1})^k v_n)$$

Eventually, $(|\frac{\lambda_i}{\lambda_1}|)^k$ will go to 0. So for k large enough, $A^k w \approx \lambda_1^k \alpha_1 v_1$.

Pseudocode for this:

```
$w_0 =\rightarrow w$, $||w|| = 1$
for $i = 1 \rightarrow \dots$ do
  $\hat{w}_{i+1} = Aw_i$
  $p_{n+1} = \sqrt{\langle \hat{w}_{i+1}, \hat{w}_{i+1} \rangle}$ just normalization
  $w_{i+1} = \frac{\hat{w}_{i+1}}{p_{i+1}}$
  $\mu_{i+1} = \langle Aw_{i+1}, w_{i+1} \rangle$
end for
```

Because

$$\begin{aligned} Av &= \lambda v \\ \langle Av, v \rangle &= \langle \lambda v, v \rangle \\ \langle Av, v \rangle &= \lambda \langle v, v \rangle \\ \lambda &= \frac{\langle Av, v \rangle}{\langle v, v \rangle} \end{aligned}$$

and μ is the estimate of λ, and w is the estimate of eigenvector.

13.2.2 Direct Method

Using QR algorithm:

```
$A_0 =\rightarrow A$
for $i = 1 \rightarrow \dots$ do
  factor $A_k = Q_k R_k$
  $A_{k+1} \rightarrow R_k Q_k$
end for
```

First assume all the eigenvalues of A are real. *Claim:* as $k \rightarrow \infty$, A_k converges to an upper triangular matrix. The eigenvalues of this upper triangular matrix is the diagonals.

We stop iteratin when everything below the diagonals of A_k is ϵ, machine precision.

Claim: A and A_k have the same eigen values. Pf: $A_0 = A$ trivial. At first iteration, $A_0 = Q_0 R_0$, $A_1 = R_0 Q_0$. R_0 is $Q_0^{-1} A_0$, since Q_0 is an orthogonal matrix, $R_0 = Q_0^T A_0$ So $A_1 = (Q_0^T A_0) Q_0 = Q_0^T A Q_0$. So $A_1 = Q_0^{-1} A Q_0$, this is the *similarity transformation.* A_1 and A are similar, and therfore they have the same eigenvalues. (if $B = P^{-1}AP$¡ then A, B, are similar and they have the same eigenvalues.)

Second iteration: $A_1 = Q_1R_1$, $A_2 = R_1Q_1$. Again $R_1 = Q_1^T A_1$, so

$$\begin{aligned} A_2 &= Q_1^T A_1 Q_1 \\ &= Q_1^T (Q_0^T A Q_0) Q_1 \\ &= (Q_0Q_1)^T A (Q_0 Q_1) \end{aligned}$$

Call $Q = (Q_0Q_1)$. So $A_2 = Q^T A Q = Q^{-1}AQ$, so A and A_2 are similar as well.

To do $A_k = Q_kR_k$ is $O(n^3)$, and we need to iterate at least n times, so the total cost is $O(n^4)$

How to make this into $O(n^3)$.

1. First reduce A to upper hesenberg form:
   Definition:
   $$\begin{pmatrix} y & x & x & x & x & x \\ x & y & x & x & x & x \\ 0 & x & y & x & x & x \\ 0 & 0 & x & y & x & x \\ 0 & 0 & 0 & x & y & x \\ 0 & 0 & 0 & 0 & x & y \end{pmatrix}$$
   A matrix where entries below the subdiagonal (x) is all 0. (y is the diagonal).

   Any real matrix can be transformed into an upper Hessenberg form using Householder transformations $(I - vv^T)$. With cost $O(n^3)$. This is an exercise.

   Let $H = P^TAP$ be the upper Hessenberg matrix, where P is the product of householder transformations. Since P orthogonal, H and A have the same eigenvalues.

2. apply QR algorithms to H. The advantage is that to do QR on hessenberg matrix takes $O(n^2)$ instead of $O(n^3)$.

So the total cost is $O(n^3) + O(n^3) = O(n^3)$

How to make a hessemberg matrix: **Givens Rotation**: $P = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$, where $c^2 + s^2 = 1$, so $\begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} ca + sb \\ -sa + cb \end{pmatrix}$. Our goal is to make the second component 0. so c and s satisfy $-sa + cb = 0$ and $c^2 + s^2 = 1$, which gives us $s = \frac{b}{\sqrt{a^2+b^2}}$ and $c = \frac{a}{\sqrt{a^2+b^2}}$ The rotation matrix P is orthogonal.

$$P^TP = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}\begin{pmatrix} c & s \\ -s & c \end{pmatrix} = \begin{pmatrix} c^2 + s^2 & cs - sc \\ sc - cs & s^2 + c^2 \end{pmatrix} = I$$

Assume we have the following upper hessenberg matrix:

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \end{pmatrix}$$

We want to make this into an upper triangular, so we need to get rid of h_{21}, h_{32}.

Let $G_1 = \begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}$, then

$$G_1 H = \begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h21 & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \end{pmatrix} = \begin{pmatrix} \xi & \xi & \xi \\ -sh_{11} + ch_{21} & \xi & \xi \\ 0 & h_{32} & h_{33} \end{pmatrix}$$

Solve $-sh_{11} + ch_{21} = 0$ constrained to $s^2 + c^2 = 1$, then $s = \frac{h_{21}}{\sqrt{h_{21}^2 + h_{11}^2}}$, and $c = \frac{h_{11}}{\sqrt{h_{11}^2 + h_{21}^2}}$

This cost us $O(n)$.

Now $G_1 H = \begin{pmatrix} \xi & \xi & \xi \\ 0 & \xi & \xi \\ 0 & h_{32} & h_{33} \end{pmatrix}$.

Using $G_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{pmatrix}$, do $G_2 G_1 H$ to get R $G_2 G_1 H = R$, where $H = (G_2 G_1)^T R_0$, and $G_2 G_1$ is Q_0. SO $H_1 = R_1 Q_1 = G_2 G_1 H G_1^T G_2^T$, multiplication is order n.

14 Class 12 October 20th 2011

Midterm 27th (next thursday). Maybe some operation counts, analysis, semi-rigorous on round-off effects (first 3-4 lectures). HW will be out before the exam, no expectation to do it. Cover materials all the way up to this class.

14.1 Why we want Q_2

$A = QR = Q_1R_1$ Only need Q_1,R_1 to solve least squares problem. $Q = [Q_1|Q_2]$, $R = [R_1; 0]$. $A^T = [R_1^T 0][Q_1^T; Q_2^T]$, vector $v = Q_2\alpha \in range(Q_2)$.

Then

$$\begin{aligned} A^Tv &= [R_1^T 0][Q_1^T Q_2\alpha; Q_2^T Q_2\alpha] \\ &= [R_1^T 0][0; \alpha] = 0 \end{aligned}$$

Because

$$I = Q^TQ = [Q_1^T; Q_2^T][Q_1Q_2] = \begin{pmatrix} Q_1^TQ_1 & Q_1^TQ_2 \\ Q_2^TQ_1 & Q_2^TQ_2 \\ \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$$

Now the *claim*: if $A^Tv = 0$, then $v \in range(Q_2)$.

This means the colums of Q_2 span the null space of A^T -¿ this is the reason why we want to know the entire Q and not Q_1 (not on the exam)

Proof of claim: Given v s.t. $A^Tv = 0$, v is an m-vector, (the larger dimension). So $v = Qa$ (Q is a matrix of m linearly independent columns). then $v = Qa = [Q_1Q_2][a_1; a_2] = Q_1a_1 + Q_2a_2$. Then

$$\begin{aligned} A^Tv &= [R_1^T 0][Q_1^T; Q_2^T][Q_1Q_2][a_1; a_2] \\ &= [R_1^T 0]I[a_1; a_2] = 0 \end{aligned}$$

If A is of full-rank, then R_1 is also full-rank, which means $a_1 = 0$ (because R_1 is non-singular, the only way we can get $R_1^Ta_1 = 0$ is if $a_1 = 0$), and $v = Q_2a_2 \in range(Q_2)$.

14.2 Eigenvalue problem

Given A, an n by n matrix (in $\mathbf{R}$ or $\mathbf{C}$). We want to find λ, v s.t. $Av - \lambda v = 0$. Maybe we want n of these (if they exist) linearly independent.

14.2.1 Power method

Start with arbitrary $x^{(0)}$. If A has n eigenvectors, then $x^{(0)} = \alpha_1v_1 + \alpha_2v_2 + \cdots + \alpha_nv_n$, a linear combination of v.

$$
\begin{aligned}
Ax^{(0)} &= \alpha_1\lambda_1 v_1 + \alpha_2\lambda_2 v_2 + \cdots + \alpha_n\lambda_n v_n \\
A^2x^{(0)} &= \alpha_1\lambda_1^2 v_1 + \alpha_2\lambda_2^2 v_2 + \cdots + \alpha_n\lambda_n^2 v_n \\
&\vdots \\
A^kx^{(0)} &= \alpha_1\lambda_1^k v_1 + \alpha_2\lambda_2^k v_2 + \cdots + \alpha_n\lambda_n^k v_n \\
&= \lambda_1^k(\alpha_1 v_1 + \alpha_2(\lambda_2/\lambda_1)^k v_2 + \cdots + \alpha_n(\lambda_n/\lambda_1)^k v_n
\end{aligned}
$$

Suppose $|\lambda_1| > |\lambda_i|$, for all $i \neq 1$. Then, all of $\frac{\lambda_i}{\lambda_1}^k \to 0$ as $k \to \infty$.

Now, we only need to consider $A^k x^{(0)} \sim$ direction of v_1. Even if $\alpha_1 = 0$, unless all of the data is a power of 2, I'll get the answer, because there will be a rounding error. so even if α_1 starts out to be 0, it won't. (Try it with a 3 by 3 example).

Say $A^k x^{(0)}$ normalized. We're only getting v_1, how do I get λ_1? Notice: $Av_1 = \lambda_1 v_1$, look at $v_1^T A v_1 = v_1^T \lambda_1 v_1 = \lambda_1 v_1^T v_1$ so $\lambda_1 = \frac{v_1^T A v_1}{v_1^T v_1}$.

Now look at $x^{(k)}$ k-th iterate obtained by the power method. Now look at

$$\frac{x^{(k)'} T A x^{(k)}}{x^{(k)'} x^{(k)}}$$

and this is the estimate for $\lambda_1^{(k)}$.

To check for convergence, look at

$$\frac{||Ax^{(k)} - \lambda_1^{(k)}||}{||x^{(k)}||} \to_? 0$$

Stop when this is less than some tolerance ϵ.

To look for the smallest λ, look for the largest $\frac{1}{\lambda_i}$ because

$$
\begin{aligned}
Av &= \lambda v \\
v &= \lambda A^{-1} v \frac{1}{\lambda} v = A^{-1} v
\end{aligned}
$$

To do this, factor $A = LU$, then in the power method when multiplying A (to perform the operation $x^{(k+1)} = A^{-1}x^{(k)}$), instead solve the system: $Ax^{(k+1} = x^{(k)}$ using $A = LU$. because you never want to compute inverse of A directly!!!

Two reasons:

- More stable numerically to use gaussian elimination
- This is just cheaper.

14.2.2 Compute all the eigenvalues

of A, Step 0 is to find an orthogonal Q s.t. $Q^T AQ$ =Hessenberg form. Nonzero above the subdiagonal. How to do this? The givens rotation, or householder matrix!

With A, apply householder transformation on the left: $Q_1 A = \begin{pmatrix} \xi & \xi & \cdots & \xi \\ \xi & \xi & \cdots & \xi \\ 0 & \xi & \cdots & \xi \\ \vdots & \xi & \cdots & \xi \\ 0 & \xi & \cdots & \xi \end{pmatrix}$
Then, apply Q_1 on the right. this will give me $Q_1 A Q_1^T$ (can write it like that because householder transformations are symmetric). *Claim*: the result is still 0 in the first column below the second entry. (WHY?? follows from the definition of the householder matrix (how v is made).

Next step is apply it again to the second column, do it $n-1$ times and get the hessenberg-form.

(that's the *preliminary* step –__–)

Now step1: start the QR iteration:

```
$H_0 = H$
for $k = 0, 1, 2, \ldots$ do
  $H_k = Q_k R_k$
  $H_{k+1} \to R_k Q_k$
end for
```

Notice: $H_{k+1} = Q^T H_k Q$ similarity matrix

Shifted version:

```
$H_0 = H$
for $k = 0, 1, 2, \ldots$ do
  $\hat{H}_k = H_k - \rho I$ subtract $\rho$ from the diagonal
  $\hat{H}_k = Q_k R_k$ then do the factorization
  $H_{k+1} \to R_k Q_k + \rho I$
end for
```

Now

$$\begin{aligned} H_{k+1} &= Q_k^T \hat{H}_k Q_k + \rho I \\ &= Q_k^T (H_k - \rho I) Q_k + \rho I \\ &= Q_k^T (H_k Q_k) - Q_k^T \rho I Q_k + \rho I \\ &= Q_k^T (H_k Q_k) - \rho I + \rho I = Q_k^T H_k Q_k \end{aligned}$$

Similarity again! When the subdiagonals go to zero, they go down to ϵ and consider them as 0. THen we're left with an upper triangular matrix. But the eigen values of the uppter triangular matrix is the values in on the diagonal! So we have all of them. (From Galois, we can't do this, but we "can" with machine precision).

Why do tehy go to zero? Claim: given $\left(\begin{smallmatrix} \alpha & \gamma \\ \beta & \delta \end{smallmatrix}\right)$, if β is small, $\beta \approx \epsilon$, then after 1 QR step, $\beta \to O(\epsilon^2)$

The last 2 by 2 matrix in $Q^T A Q$, the (2,1) of the bottom 2 by 2 matrix goes to ϵ^2, so once we do that, move up to the next 2 by 2 matrix on the diagonal.