NTNU – Trondheim
Norwegian University of
Science and Technology

A quick and dirty look at

The conjugate gradient method

1. Column vectors and directions in A
2. Quadratic form and gradients
3. Practical demonstration and considerations

More linear algebra

- We're solving a system like this again:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}$$

An alternative view

- Last time, we read it as one linear equation per row
- Another way to look at it is to think that
  - The matrix is made of some vectors that point in some directions
  - Our x-s let us decide how far to stretch each of them

$$x_1 \begin{bmatrix} 1 \\ 4 \\ 7 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ 5 \\ 8 \end{bmatrix} + x_3 \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}$$

My example is a bit stupid

- The matrix is singular
  - I just chose it to obviously contain indices of the elements
- To begin with today, I just wanted to point out that a matrix contains its own twisted coordinate system
- Let’s take a look at this one instead:

$$\begin{bmatrix} 0 & 3 & 2 \\ 1 & 4 & 2 \\ 3 & 4 & 1 \end{bmatrix}$$

NTNU – Trondheim
Norwegian University of Science and Technology

The coordinates are warped

- When we look at vectors through the lens of A, most of them rotate and stretch:

$$\begin{bmatrix} 0 & 3 & 2 \\ 1 & 4 & 2 \\ 3 & 4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 8 \end{bmatrix}$$

- You can look at this A as a (linear) transformation of the 3D space with real coordinates R^3

Not *all* vectors rotate

- These don't (well, approximately – they're rounded off to 4 figures)

$$\begin{bmatrix}0 & 3 & 2\\1 & 4 & 2\\3 & 4 & 1\end{bmatrix}\begin{bmatrix}0.4552\\0.6103\\0.6484\end{bmatrix}=\begin{bmatrix}3.1275\\7.1930\\4.4549\end{bmatrix}$$

$$\begin{bmatrix}0 & 3 & 2\\1 & 4 & 2\\3 & 4 & 1\end{bmatrix}\begin{bmatrix}0.5574\\0.1790\\-0.8107\end{bmatrix}=\begin{bmatrix}-1.0846\\-0.3482\\1.5774\end{bmatrix}$$

$$\begin{bmatrix}0 & 3 & 2\\1 & 4 & 2\\3 & 4 & 1\end{bmatrix}\begin{bmatrix}0.4257\\-0.4946\\0.7578\end{bmatrix}=\begin{bmatrix}0.031845\\-0.036994\\0.056682\end{bmatrix}$$

...and also...

$$6.870865\begin{bmatrix}0.4552\\0.6103\\0.6484\end{bmatrix}=\begin{bmatrix}3.1275\\7.1930\\4.4549\end{bmatrix}$$

$$-1.945668\begin{bmatrix}0.5574\\0.1790\\-0.8107\end{bmatrix}=\begin{bmatrix}-1.0846\\-0.3482\\1.5774\end{bmatrix}$$

$$0.074803\begin{bmatrix}0.4257\\-0.4946\\0.7578\end{bmatrix}=\begin{bmatrix}0.031845\\-0.036994\\0.056682\end{bmatrix}$$

- These vectors are the *characteristic vectors* (or *eigenvectors*) of A
  - They lie along the axes of A's 'inherent coordinate system' (when it has one)
  - The matching scalars on the right are the *eigenvalues*

Returning to Ax = b

- We can take our A and our b and make this scalar function out of them:

$$f(x) = \frac{1}{2}x^T A x - b^T x + c$$

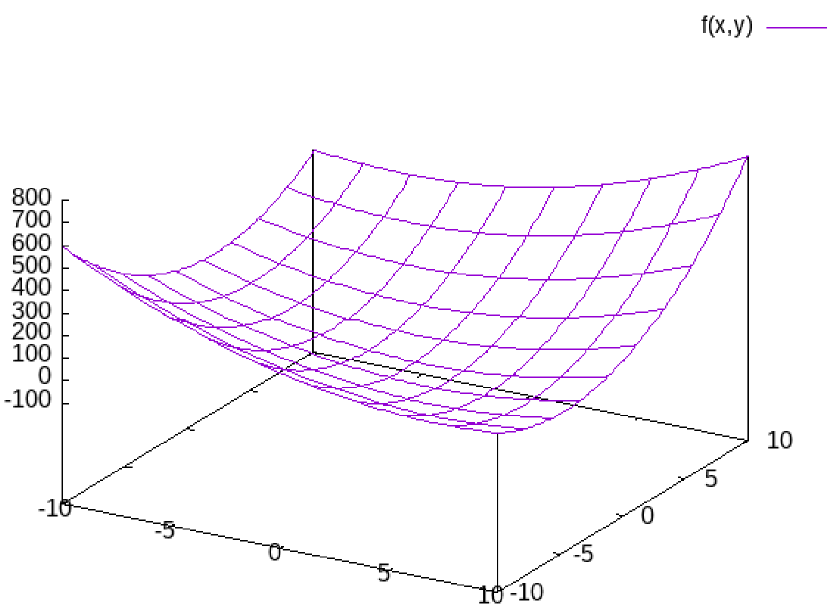
(just let c=0 for our purposes)

- If x is two-dimensional *(i.e.* [x_1,x_2]), we get a number from each pair of coordinates
  - Great, we can draw a picture!

NTNU – Trondheim
Norwegian University of Science and Technology

Jonathan’s example system

- If you work out the quadratic form with the system in the paper, you get something similar to
  - $f(x,y) = 3x^2/2 + 3y^2 + 2yx – 2x + 8y$
    (if I did my arithmetic correctly)
- and it looks like this:
  - Key point:
    it has a bottom
  - The x that minimizes this f(x) solves Ax=b



Why does the minimum of the quadratic form solve Ax=b?

- The quadratic form is chosen so that its (multi-dimensional) derivative is Ax – b
- Consider how we wrote it: $f(x) = \frac{1}{2}x^T A x - b^T x + c$
- If we try it out using only 1 dimension, we get

$$f(x) = \frac{1}{2}xax - bx + c = \frac{1}{2}ax^2 - bx + c$$

and its derivative is

$$f'(x) = ax - b$$

which is 0 at the bottom of the parabola f describes

How do you find the bottom of a bowl?

- Drop a marble in, and it will roll there
  - by mostly going downhill
- That's the philosophy of the *gradient descent* method
  1. Pick a point, any point
  2. Find the gradient vector (differentiate f(x,y) *wrt.* x and y, separately)
  3. That vector points uphill, so downhill is the other way
  4. Take a step to the point where the landscape ascends again
  5. Repeat from step 2
- This method bounces a little bit back and forth, depending on where you start
- If you happen to start descending in the direction of an eigenvector of A, it hits the bottom right away

Slightly more systematically

- The only reason we might miss the bottom, is that our gradients are from viewing the landscape stretched out across the regular x,y,z… axes
- If we translate our search into the space that A suggests, we should never miss because we'd only need to take 1 calculated step along each of its axes
- With an N-dimensional matrix, that's N steps
  - 1 dimension at a time
  - No overshooting or undershooting

That is the conjugate gradient method

- Explicitly finding all the eigenvectors first takes a horrendous amount of time for huge A

  (I know, it's a recurring motif)

- If we take N iterations and transform the search direction by A, we can work out A-orthogonal directions on the fly, though
  - That's what the Gram-Schmidt process in the paper does

Hooray, we have it!

- Equations on p.32
- C code in today's archive
- We can apply it to the ex3 matrix from last time
  - In a minute, I just have to mention the disclaimers first

Positive definite matrices

- If every x vector gives $x^T A x > 0$, we say that it's positive definite
- This gives the shape of the quadratic form its bottom
  - similarly to how a positive coefficient for x^2 gives parabolas with a bottom for all quadratic functions $f(x) = ax^2 + bx + c$
- Without this, we get quadratic points with a maximum instead
  - or even a saddle shape that leads line-seaches for the bottom off into negative infinity

Symmetry

- Symmetric, positive definite NxN matrices have N distinct eigenvectors that create a search space of orthogonal axes
  - It's good to know that we have enough search directions to finish when we're trying to cover one at a time

When conjugate gradients work (or not)

- Plain CG works for symmetric, positive definite matrices
  - Luckily for us, ex3 is both symmetric and positive definite
- It can still be a bit wobbly
  - There's a term

    $$\beta_{(i+1)} = \frac{r_{i+1} \cdot r_{i+1}}{r_i \cdot r_i}$$

    in there which measures how far we are from a solution (as per the size of the residual)
  - When we miss by a tiny amount, this number can go completely bananas

Time to try it

- Most parts are just like last week
  - download.sh gets ex3 from web and pulls out the numbers
  - convert_full_matrix.c translates it into a simple binary file
  - row_sums.c gives us a b.dat file to aim for, and a correct answer to expect
- conjugate_gradient.c is mostly a direct translation of the summary on page 32 in the paper

It doesn’t hit so well in N steps

- Sadly, floating point numbers are not exact
  - We do, however, get something in the right ballpark
- This program also doesn’t have a very formally defined halting criterion
  - It doesn’t run until convergence-within-a-threshold
  - We could make it so, but I want to make a point about it

Gauss-Seidel took 11 steps for a better answer last week

- This takes 1821 steps, and it doesn't even hit the target...?
- True, *but*
  - We didn't have to multiply the diagonal by 10 to make A diagonally dominant this time
  - Solving systems with ex3-size matrices isn't really what it's used for
  - You can apply CG (or even better, its stable friends and relatives) to matrices that are too big for exact solutions
  - It produces approximate ones in reasonable time

The world according to Alex

- When asked about practical convergence criteria, my distinguished ol’ professor of numerical physics said (and I quote)

  “We usually just run it ten times over to make sure.”

- That’s good enough for me
  - Remember that you can always put the answer back into Ax=b to check if it’s good enough for you

The truly practical solution

- Use a library
- As before, I only aspire to expose enough of the inner workings to evaluate whether or not we've chosen the right tool
  - Meticulous treatment, proofs, *etc.* are in the paper

NTNU – Trondheim
Norwegian University of Science and Technology