Project list for Integration Workshop

Colin Clark
Graduate Program in Applied Mathematics,
University of Arizona, Tucson

August 2021

Objectives:

1. Make sure each student has a basic working knowledge of Matlab, Python, Julia, or some other high-level programming language for scientific computing.
2. Make sure each student has a basic working knowledge of LaTeX.
3. Make sure students can use computational approaches to solve math problems (turning mathematics into computer code).
4. Writing, reading, and discussing algorithms in a group setting.
5. Having fun!

1 The Fredholm Alternative & Least Squares Solutions

Sometimes we focus too much on solving the matrix equation $A\boldsymbol{x} = \boldsymbol{b}$ for situations where A is a square matrix, and we ignore situations where A is not square. In many practical applications, A is an $m \times n$ matrix with $m \neq n$. In this problem, we're going to explore what we mean by *solutions to the matrix equation for non-square matrices.*

The Fredholm Alternative

The Fredholm alternative states that for the matrix equation $Ax = b$, exactly one of the following statements is true:

- (Either) There exists an x that solves the matrix equation $Ax = b$.
- (Or) There exists a y that solves $A^\top y = 0$ such that $y^\top b \neq 0$.

Let A_m be the $3 \times m$ matrix (below), and let b' and b'' be the 3×1 vectors (below).

$$A_m = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & m \end{bmatrix}, \qquad b' = \begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix}, \quad b'' = \begin{bmatrix} -1 \\ 0 \\ +1 \end{bmatrix}$$

(a) Plot the vectors that make the columns of A_m for $m = 5$, and use your figure to describe the column space of A_m.

(b) Use pencil-and-paper to verify the Fredholm Alternative for b' and for b''.

(c) Plot the vectors b' and b'' on the same figure from part (a), and use your figure to provide an intuitive explanation for the Fredholm Alternative.

Pseudo-inverses & Least Squares Solutions

When there is no solution to the matrix equation $Ax = b$, we may have to look for the 'next best thing'. Your co-worker suggests the following matrix algebra to find the 'next best thing'.

$$Ax = b \quad \Rightarrow \quad A^\top Ax = A^\top b \quad \Rightarrow \quad x = (A^\top A)^{-1} A^\top b$$

(d) What are the dimensions of the matrix $(A^\top A)$? What are requirements on A for the matrix $(A^\top A)$ to be invertable?

(e) For $m = 2$, find the matrix $A^\dagger := (A_m^\top A_m) A_m^\top$ and compute the vectors $x' := A^\dagger b'$ and $x'' := A^\dagger b''$.

(f) Does $Ax' = b'$? What about $Ax'' = b''$? Can you describe what we mean when we say that $A^{\dagger}$ gives the 'next best thing'?

(g) (Bonus) Use some calculus to verify your answer in part (f).

2 Lagrange Interpolation for Function Approximation

Let $\{(x_k, y_k), k = 1, \ldots, N\}$ be N points in $\mathbb{R}^2$ that satisfy $x_k \neq x_j$ whenever $k \neq j$, the *Lagrange interpolating polynomial* is defined as

$$L(x) = \sum_{k=1}^{N} \left(\prod_{j \neq k} \frac{x - x_j}{x_k - x_j} \right) y_k. \tag{2.1}$$

We wish to determine how well a Lagrange interpolating polynomial can approximate the functions

$$f(x) = (x - .9)(x - .4)(x + .1)(x + .7)(x + .8) \tag{2.2}$$

and

$$g(x) = \frac{1}{1 + 10x^2} \qquad \text{for } -1 \leq x \leq 1 \tag{2.3}$$

(a) Observe that the Lagrange interpolating polynomial is linear combination of N terms in the form

$$T_k(x) := \left(\prod_{j \neq k} \frac{x - x_j}{x_k - x_j} \right) \tag{2.4}$$

Let $x_1 = -1$, $x_2 = 0$, and $x_3 = 1$. Plot the functions $T_k(x)$ for $k = 1, \ldots, 3$, and describe what you see.

(b) i. Sample $f(x)$ at the points $x_1 = -1$, $x_2 = 0$, and $x_3 = 1$. Compute and plot the Lagrange Interpolating polynomial given by these 3 equi-spaced points.

ii. Repeat part (b)(i) with 5 equi-spaced points, and again with 9 equi-spaced points. Plot your results and describe whether your approximation improves with more points.

(c) Repeat part (b) for the function $g(x)$.

(d) There is a well-known rule of thumb for improving numerical approximations: Increase the sampling density in the regions where the error is greatest. Does this rule of thumb seem to work with the functions f and g?

3 Contour Integration of a Complex-valued Function

(This problem is probably out of reach for students who not yet taken a course in complex analysis. No worries, we will cover this topic during the first few weeks of Math 583A.)

Consider the functions $f : \mathbb{C} \to \mathbb{C}$ and $g : \mathbb{C} \to \mathbb{C}$ defined as

$$f(z) = z^2 \qquad \text{and} \qquad g(z) = z^{-1}$$

(a) Plotting:

  i. Make a surface plot showing $Re(f(z))$ and a surface plot showing $Im(f(z))$

  ii. Make a surface plot showing $Re(g(z))$ and a surface plot showing $Im(g(z))$

  Since g is undefined at $z = 0$ and since it is radially symmetric, the surface plots for g can look scary when done in cartesian co-ordinates. You may try switching to polar coordinates to make prettier plots

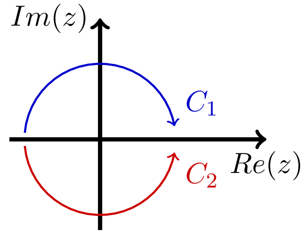
(b) We can approximate the line integral (or contour integral) of a function along a curve by discretizing the curve into $N + 1$ points $z(s) \to \{z_0, z_1, z_2, \ldots, z_N\}$ and then computing the sum:

$$\int_{z_o}^{z_t} f(z)dz \approx \sum_{k=1}^{N} f(z_k)\ (z_k - z_{k-1})$$

  Define the two curves:

  C_1 : The upper half circle of radius 1 centered at $z = 0$.

  C_2 : The lower half circle of radius 1 centered at $z = 0$.



  Approximate the line integrals of f and of g along the curves C_1 and C_2.

(c) (Bonus) Repeat parts (a) and (b) for the function $h(z) = z^{1/2}$.
  Note: You may need to make some 'executive decisions' to make sure your problem is well-posed.

4 Eigenvectors of Circulant Matrices

Define the 16×16 matrix A by

$$A = \begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{bmatrix} \tag{4.1}$$

In this project, we would like to explore (and explain) the behavior of the iterative scheme

$$\boldsymbol{x}_{k+1} = (I + \eta A)\, \boldsymbol{x}_k \tag{4.2}$$

for various initial vectors $\boldsymbol{x}_0$, and for $\eta = 0.1$.

(a) Create a variable for the matrix A by using the function `A=diag(...)` (Matlab), or `A=np.diag(...)` (Python), or `diagm(...)` (Julia).

(b) Try different initial vectors $\boldsymbol{x}_0$ and plot $\boldsymbol{x}_1$, $\boldsymbol{x}_2, \dots$. Can you qualitatively describe the behavior of the matrix $(I + \eta A)$?

(c) Compute the eigenvectors of $(I+\eta A)$ by using the function `[X,V]=eig(...)` (Matlab), `V,X = np.linalg.eig(...)` (Python), or `V,X=eigen(...)` (Julia).

(d) Plot enough of the eigenvectors so you get a sense for what they look like.
Hint: If your software returned complex-valued eigenvectors, it may be helpful to plot real and imaginary components separately (e.g. `plot(real(X[k,:]))` and `plot(imag(X[k,:]))`).

(e) Propose a closed form representation for each of the eigenvectors.

(∗) *Bonus:* Can you prove that every symmetric circulant matrix has eigenvectors in the form your found in part (e).

(f) Without any further computation, and only by considering eigenvectors and corresponding eigenvalues, describe the behavior of (4.2) for an initial vector in the form

$$\boldsymbol{x}_0 = \big(\cos(2\pi k/16) + 5\cos(4 \cdot 2\pi k/16),\ \text{for } k = 0, ..., 15 \big).$$

Repeat these steps for the 16×16 matrix B, defined by

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 \end{bmatrix} \tag{4.3}$$

5 First-order ODE Integrators

Consider the differential equation

$$\begin{cases} \dot{x}(t) = -y, & x(0) = 1 \\ \dot{y}(t) = x, & y(0) = 0 \end{cases} \tag{5.1}$$

(Warm-up) Use pencil-and-paper to solve the ODE. Make a quiver plot. Sketch solutions.
(Project) We wish to build a numerical solver for differential equations like this one. We will test our solver on this ODE, and then use it to solve a more interesting ODE.

For this problems, we will discretize the time interval $0 \le t \le T$ into $n+1$ points. That is, set $\Delta t = T/n$ and $t_k = k\,\Delta t$, so $t = \{t_0, t_1, \dots, t_n\}$.

Forward Differences

(a) Use what's called a *first-order forward difference* approximation to the derivatives $\dot{x}$ and $\dot{y}$ at each t as follows:

$$\dot{x}(t) \approx \frac{x(t+\Delta t) - x(t)}{\Delta t} \qquad \text{and} \qquad \dot{y}(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}. \tag{5.2}$$

Substitute equation (5.2) into equation (5.1) and show that we can approximate the differential equation by the discrete time-stepping process:

$$\begin{cases} x_{k+1} = x_k - (\Delta t) y_k, & x_0 = 1 \\ y_{k+1} = y_k + (\Delta t) x_k, & y_0 = 0 \end{cases} \tag{5.3}$$

(b) Implement equation 5.3 for $0 \le t \le 1$ and plot the trajectory given by the solution.

(c) Is your approximation qualitatively correct? Will it remain qualitatively correct on $0 \le t \le T$ as T gets very large? Explain.
Hint: It may be useful to rewrite the system of difference equations in matrix form,

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & -\Delta t \\ \Delta t & 1 \end{bmatrix} \boldsymbol{x}_k, \quad \boldsymbol{x}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and to analyze the eigenvalues of the matrix.

Backward Differences

(d) We could have solved this problem by a different approach. We could also have used what's called a *first-order backward difference*:

$$\dot{x}(t) \approx \frac{x(t) - x(t - \Delta t)}{\Delta t} \qquad \text{and} \qquad \dot{y}(t) \approx \frac{y(t) - y(t\Delta t)}{\Delta t}. \tag{5.4}$$

Show that the backward difference gives the approximation

$$\begin{cases} x_k = x_{k-1} - (\Delta t) y_k, & x_0 = 1 \\ y_k = y_{k-1} + (\Delta t) x_k, & y_0 = 0 \end{cases} \tag{5.5}$$

which can be rewritten as

$$\boldsymbol{x}_k = \left(\begin{bmatrix} 1 & \Delta t \\ -\Delta t & 1 \end{bmatrix} \right)^{-1} \boldsymbol{x}_{k-1}, \quad \boldsymbol{x}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

(e) Implement equation (5.5) for $0 \leq t \leq 1$ and plot the trajectory given by the solution.

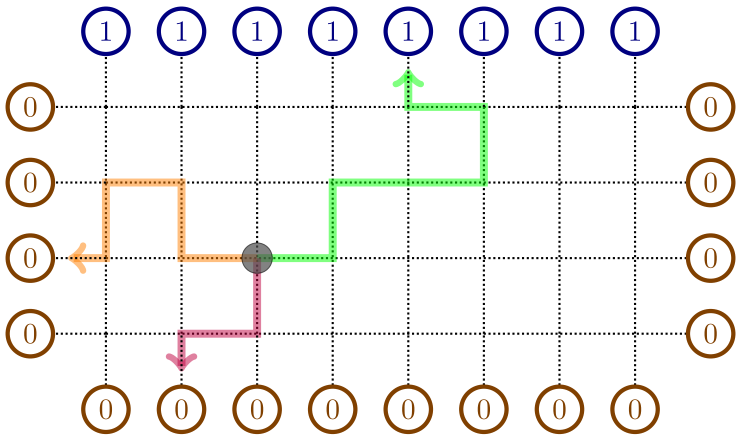
(f) Is your approximation qualitatively correct? Will it remain qualitatively correct on $0 \leq t \leq T$ as T gets very large? Explain.

(Bonus)

(g) (The really fun stuff) Perhaps solve $\dot{x} = -y$, $\dot{y} = \sin x$. Maybe this is too hard.

6 Random Walk on a Grid

A 'particle' takes a random walk on a 4×8 grid. At each step of the walk, the particle may move up, down, left, or right with equal probability. The particle continues this process until it exits the grid. If the particle exits the top of the grid, the particle scores 1 point for the walk. If the particle exits the left, bottom or right sides of the grid, the particle scores 0 points for the walk.



The figure shows three sample paths of the random walk that all begin at the initial point (3,2). For the red path, the particle (randomly) takes the steps $(\downarrow, \leftarrow, \downarrow)$, and exits the grid at the bottom scoring zero points for the walk. The other paths are the results of the steps listed below

Steps: $(\downarrow \leftarrow \downarrow)$	(See red path)	Exit: Bottom	Score: 0
Steps: $(\leftarrow \uparrow \leftarrow \downarrow \leftarrow)$	(See orange path)	Exit: Left	Score: 0
Steps: $(\rightarrow \uparrow \rightarrow \rightarrow \uparrow \leftarrow \uparrow)$	(See green path)	Exit: Top	Score: 1

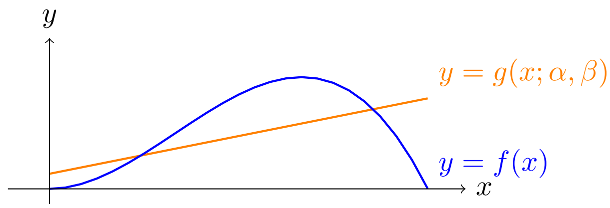
Compute the expected score of the particle as a function of the starting coordinates.

Hint: For each starting point, you may estimate the expected score of the particle by simulating many, many random paths and taking their average score.

Bonus: In this project, we have computed the solution to a problem by simulating many random events. This 'indirect' approach is always computationally inefficient, so it is often worth looking for a 'direct' approach. For this problem, there *is* a method of computing the solution directly. Can you find it?

7 Bias and Variance for learning in 1-D

Let $f(x) = x^2(1-x)$ where $0 \leq x \leq 1$. We wish to find the 'best' linear model (or approximation) for f. That is, for functions in the form $g(x;\alpha,\beta) = \alpha + \beta x$, we wish to find the parameters α and β that minimize the error we would expect to incur if we used the function g to model (or approximate) the function f.



For this problem, we will define the error as

$$\text{Error}(f,g) = \int_0^1 \Big(f(x) - g(x;\alpha,\beta)\Big)^2 dx.$$

Use pencil-and-paper (or Mathematica, WolframAlpha, etc) to find a simplified expression for the error between f and g as a function of the parameters α and β. Find the values of α and β that minimize this error and the corresponding error.

Terminology: The error we expect to incur when we use our best possible candidate is called the *bias* of the model.

In many practical applications, we only have imperfect knowledge of f, and therefore it is impossible to know the parameters of the *best possible* candidate $g(x,\alpha,\beta)$. Often, we must commit to a choice of sub-optimal parameters that are chosen after only a small glimpse of f. In this project, we will estimate how much extra error we expect to incur by using an imperfect choice of parameters.

Repeat the following recipe to estimate how well a linear function parameterized by two random data points can be used to approximate f.

`Loop until you are confident you have reasonably accurate answers.`

1. `Generate two random data points on` $x_1, x_2 \in [0,1]$.
2. `Using only` x_1 `and` x_2, `compute a 'best guess' for parameters` α `and` β.
3. `Test the performance of your (sub-optimal) parameters by generating new data points and computing the mean square error on the new points. Record the mean square error.`

`Analyze the mean square error by taking averages and plotting histograms.`

Terminology: When we use 'best guess' parameters instead of 'best possible' parameters, we can expect to incur error beyond the bias of the model. This 'extra error' is called the *variance* of the model.

For this project, you will need to write the following functions:

1. a function that randomly generates two values in $[0, 1]$
2. a function called `train_g`.
   - Arguments (Input): Two values x_1, and x_2.
   - Functionality: Evaluate $y_1 := f(x_1)$ and $y_2 := f(x_2)$. Compute a 'best guess' for the parameters α and β given the points (x_1, y_1) and (x_2, y_2).
   - Return (Output): α and β.
3. a function called `test_g`.
   - Arguments (Input): Parameters α and β.
   - Functionality: Randomly pick N points in $[0, 1]$. For each point, x_k, evaluate $f(x_k)$ and $g(x_k; \alpha, \beta)$. Compute the average error from the N points as defined by
   $$E_N = \frac{1}{N} \sum_{k=1}^{N} \Big(f(x_k) - g(x_k, \alpha, \beta) \Big)^2$$
   - Return (Output): E_N

Someone proposes that the variance is so big that it makes the linear model unusable. They suggest that we should approximate f by a constant function $h(x; \alpha)$. Repeat the steps above to compute the bias and to the estimate the variance of the constant model.

8 Curse of dimensionality

In Progress.

Show figures of three triangles. Students must use the vertices to compute side lengths, interior angles, and areas. TODO: Find a simple measure of the 'quality' of a triangle, (i.e. how close it is to equilateral) The project is to determine whether the expected quality of random triangles in R2 is different than random triangles in R10.

1. Use a normal random number generator to create three random points in R2.
2. Let these points be the vertices of a triangle. Compute the lengths and interior angles of the triangle.
3. Repeat, and histogram triangle quality scores

Repeat process for triangls in R10, (i.e. use a multivariate normal to generate 3 points in R10 and analze the triange. Can you find an heuristic, geometric explanation for why the triangles in high dimensions are much closer to equilateral?