UBC CPSC 303, 2018 Winter Term 2

Problem Set 5b, due Wednesday, April 3, 11:00am

Instructions: Produce *well-written* solutions that answer the problems that follow.

- Introduce notation/terminology as required clearly.
- Use the rubric provided on Piazza to guide your solution development. Remember to reference the prescribed learning objectives as well.
- You may work in pairs. Indicate both your CSIDs on your solution write-up and submit through Gradescope. *Do not put your name on work uploaded to Gradescope!*
- Submit your work as a single PDF file; additional code files can be uploaded to Gradescope as well. Include the code and the output from running your programs in your main write-up (i.e., provide some evidence of the code's function that can be assessed prior to executing the code).
- You are not obligated to use MATLAB for programming work; you can use Python (or even another tool language that can run within a Jupyter notebook). **However, some starter code in Matlab is provided; therefore, using another language would require more work.**

Note: This problem set is a continuation of Problem Set 5a.

Note: You are provided with two starter files, `main_student.m` and `createA.m`.

(f) Create the A matrix using the provided code, and plot the sparsity pattern (`spy` in MATLAB). Briefly describe the pattern and relate it to the mathematical form of A from the earlier part of the Problem Set. (You may need to zoom in on the plot!)

(g) Numerically solve the convection-diffusion equation for wind parameters $W = 1$ and $\theta = \pi/2$ and initial pollution parameters $a_1 = 2$, $a_2 = 1$, $s_1 = 100$, $s_2 = 150$ using a time-step $\Delta t = 0.025$. This will involve solving the linear system at each iteration, which can be achieved with the \ command ("backslash") in MATLAB. Plot the pollution concentration profiles at $t = 0$, $t = 0.125$ and $t = 0.25$.

(h) Was your code unbearably slow? Mine was too. It would also be nice to use a smaller Δt, but you are probably taking several other courses and don't have all day to wait! To improve the efficiency of the code, notice that the system matrix is very sparse. Indeed, the fraction of nonzero elements is a mere $\frac{31525}{81^4} \approx 7 \times 10^{-4}$, or less than 0.1%.

It turns out that calculations with sparse matrices can often be done much more quickly (because you can skip all the multiplications by zero and additions of zero). Modify your code so that the system matrix is represented as a sparse matrix. In MATLAB, this is just a simple one-line change using the command `sparse` to turn your dense matrix into a sparse one.[1] Once you create a sparse matrix, you can use it exactly as you would a regular matrix, and MATLAB will automatically perform the calculations in a different way. With this new code, report your speedup factor (original elapsed time divided by new elapsed time) when using $\Delta t = 0.025$. (In MATLAB, you can use `tic` and `toc` to measure the running times.) Then, repeat part (g) but using a time-step $\Delta t = 0.005$. We will use $\Delta t = 0.005$ from now on.

Optional FYI: We sped up the code by doing sparse matrix computations, but we still need to create the dense matrix, only to throw it away a moment later when we converted it to a sparse matrix. This is a serious problem when the dense matrix does not even fit into memory. To get around this, there are commands to directly create sparse matrices, such as `spdiags` in MATLAB (I'm not asking you to implement it this way, though). You can use the `whos` command in the MATLAB terminal to see a list of variables and the memory usage in each case. Doing so reveals that the sparse representation of A reduces the memory usage by a factor of over 600. (It's not exactly the number of nonzero elements in A, because there's some overhead in the sparse representation.) For a bigger problem, this could easily be the difference between running out of memory or not.

(i) Suppose that the kindergarten is at the position $\mathbf{x}_K = (0.5, 0.5)$, and that the school day starts at $t_0 = 0$ and ends at $t_f = 0.25$. Let

$$k(t) \equiv u(\mathbf{x}_K, t) \tag{1}$$

be the pollution level at the kindergarten as a function of time, and let

$$K \equiv \int_{t_0}^{t_f} k(t)dt \tag{2}$$

be the total pollution experienced by the kindergarteners each school day. From your solution in part (h), plot your approximation to $k(t)$ for $t \in [0, t_f]$, and use a composite trapezoid rule (you can implement it yourself, or use `trapz`) to determine K.

What we have done so far assumes we know the wind direction, θ, the wind speed, W, and the initial pollution parameters a_1 and a_2 (we will leave $s_1 = 100$ and $s_2 = 150$ to keep things a bit simpler). However, in reality we do not know these values but can only estimate their probability distributions.[2] We now describe the model for each of these parameters:

- **Wind direction**, θ. Let us assume that the wind is equally likely to blow in any direction, meaning $p(\theta) = \frac{1}{2\pi}$ for $\theta \in [0, 2\pi]$.[3] You can generate uniform random numbers in $[0, 1]$ using `rand`.
- **Wind speed**, W. Assume the wind speed follows the *Weibull distribution*.[4] The Weibull distribution has support on $[0, \infty)$ and has two parameters called the shape and scale, which

[1] Internally, this means MATLAB stores the locations and values of all the nonzero elements, rather than just storing all the elements in order.

[2] Note that we are assuming these four parameters are independent of each other, so that we can write their joint probability distribution $p(W, \theta, a_1, a_2)$ as the product $p(W)p(\theta)p(a_1)p(a_2)$. This seems fairly reasonable.

[3] For those unused to seeing probability distributions written like this, please forgive this commonplace abuse of notation: the function $p(\theta)$ is a different $p(\cdot)$ than the function $p(W)$. We should really write $p_\theta(\theta)$ and $p_W(W)$ to indicate that these distributions are not the same, but we tend not to because the subscripts get too cumbersome. Also, you may be used to the symbol f for probability densities, but I'm using p.

[4] Researchers have found fairly good agreement between empirical observations of wind speed and the Weibull distribution. For more information on this distribution, see https://en.wikipedia.org/wiki/Weibull_distribution.

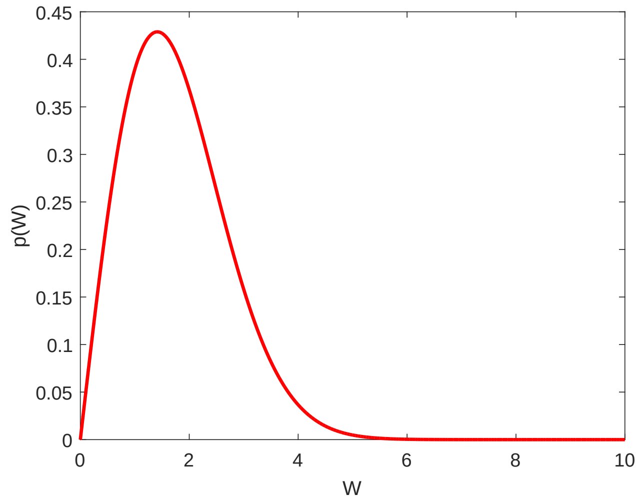

Figure 1: Weibull distribution PDF with shape 2 and scale 2.

are related to the mean and variance of the distribution; here, we will take both parameters to be 2; see Figure 1 for a plot of $p(W)$ with these parameters.[5] You can access the probability density function $p(W)$ using `wblpdf(W,2,2)` and you can generate a sample from the distribution using `wblrnd(2,2)`.[6]

- **Initial pollution strengths**, a_1, a_2. We model the pollution parameters with the exponential distribution, $p(x) = \frac{1}{\lambda}e^{-x/\lambda}$, where λ is the mean of the distribution. We will set the mean of each distribution to the values initially used in part (g), namely $\lambda_{a_1} = 2$ and $\lambda_{a_2} = 1$. To sample from the exponential distribution you can use `exprnd`, which generates a sample from the exponential distribution with the mean supplied as an argument; for example `exprnd(2)` samples from the exponential distribution with $\lambda = 2$.[7]

Given these distributions, we wish to determine the expected (average) total pollution that children in the kindergarten are exposed to during a school day:

$$\mathbb{E}[K] = \int_0^{2\pi} \int_0^{\infty} \int_0^{\infty} \int_0^{\infty} K(W, \theta, a_1, a_2)\, p(\theta)p(W)p(a_1)p(a_2)\, da_2\, da_1\, dW\, d\theta\,. \tag{3}$$

Note that here we explicitly show the dependence of K on the parameters W, θ, a_1, a_2, which we omitted in eq. (2). In words, this is a 4-dimensional integral representing the total pollution experienced by the kindergarten over the course of a day, averaged over the unknown wind and factory conditions. The function K itself involves an integral, and in fact within that lives a PDE.

(j) Use Monte Carlo integration to estimate $\mathbb{E}[K]$ in eq. (3). This is achieved by repeatedly *sampling* our four parameters from their respective probability distributions, and computing $K(W, \theta, a_1, a_2)$ given these samples. Then, an approximation to eq. (3) is the average

[5] In case you were curious, the mean of the distribution is $2\Gamma(\frac{3}{2}) \approx 1.8$ and the variance is $4(\Gamma(2) - \Gamma(\frac{3}{2})^2) \approx 0.86$.

[6] If you do not have the Statistics and Machine Learning Toolbox, and thus do not have access to these functions, use the provided `wblpdf303` and `wblrnd303`.

[7] If you do not have access to the Statistics and Machine Learning Toolbox, and thus do not have access to the `exprnd` function, you can define it with `exprnd=@(mu)-mu*log(rand)`.

$K(W, \theta, a_1, a_2)$ over your n trials. In symbols:

$$\mathbb{E}[K] \approx \frac{1}{n} \sum_{i=1}^{n} K\left(W^{(i)}, \theta^{(i)}, a_1^{(i)}, a_2^{(i)}\right) , \tag{4}$$

where $W^{(i)}$, $\theta^{(i)}$, $a_1^{(i)}$, and $a_2^{(i)}$ are the ith samples of W, θ, a_1, and a_2 drawn from $p(W)$, $p(\theta)$, $p(a_1)$, and $p(a_2)$, respectively. Report the expected total pollution that you compute with $n = 100$.

(k) The various parts of your code involve many different approximations. Thinking back to Chapter 1 of the textbook, which talks about different error types, write a few sentences about the different error sources in the entire pipeline you developed for this assignment. How might you assess the severity of these different error sources?

(l) **(Bonus, not for marks)** In part (j) we computed the *expected*, or average, pollution exposure to the children in a single day. However, we may also be interested in the *maximum* pollution exposure in a single day. In answering this question, it is not that interesting to think about a_1 and a_2, because the pollution exposure just increases monotonically as we increase these parameters. But, it is an interesting problem if we fix a_1 and a_2 and ask what wind conditions are most dangerous for the children. This amounts to solving the optimization problem,

$$K_\star = \max_{W,\theta} K(W, \theta) \tag{5}$$

$$\{W_\star, \theta_\star\} = \arg\max_{W,\theta} K(W, \theta) \,. \tag{6}$$

MATLAB provides optimization routines to perform maximization and minimization given arbitrary objective functions. In this case, we will use `fmincon`, which is a routine for constrained optimization. The constraints here are the bounds we will impose on W and θ; in this case we will use $W \in [0, 5]$ and $\theta \in [0, 2\pi]$. Use your existing code along with `fmincon` to find the most dangerous wind parameters, $W_\star$ and $\theta_\star$ in the domain for two separate occasions: (1) $a_1 = 1$ and $a_2 = 2$, and (2) $a_1 = 2$ and $a_2 = 1$. `fmincon` is a local optimization method, which means you will have to provide an initial guess. You may get different solutions depending on your initial guess, so you will need to think about good initial guesses or perhaps perform some trial and error.

Use `fmincon`'s default options, and don't pass gradient or Hessian data to `fmincon`.

What are $W_\star$ and $\theta_\star$ in each of the two cases? Do the results make sense intuitively?