Paul Buttles

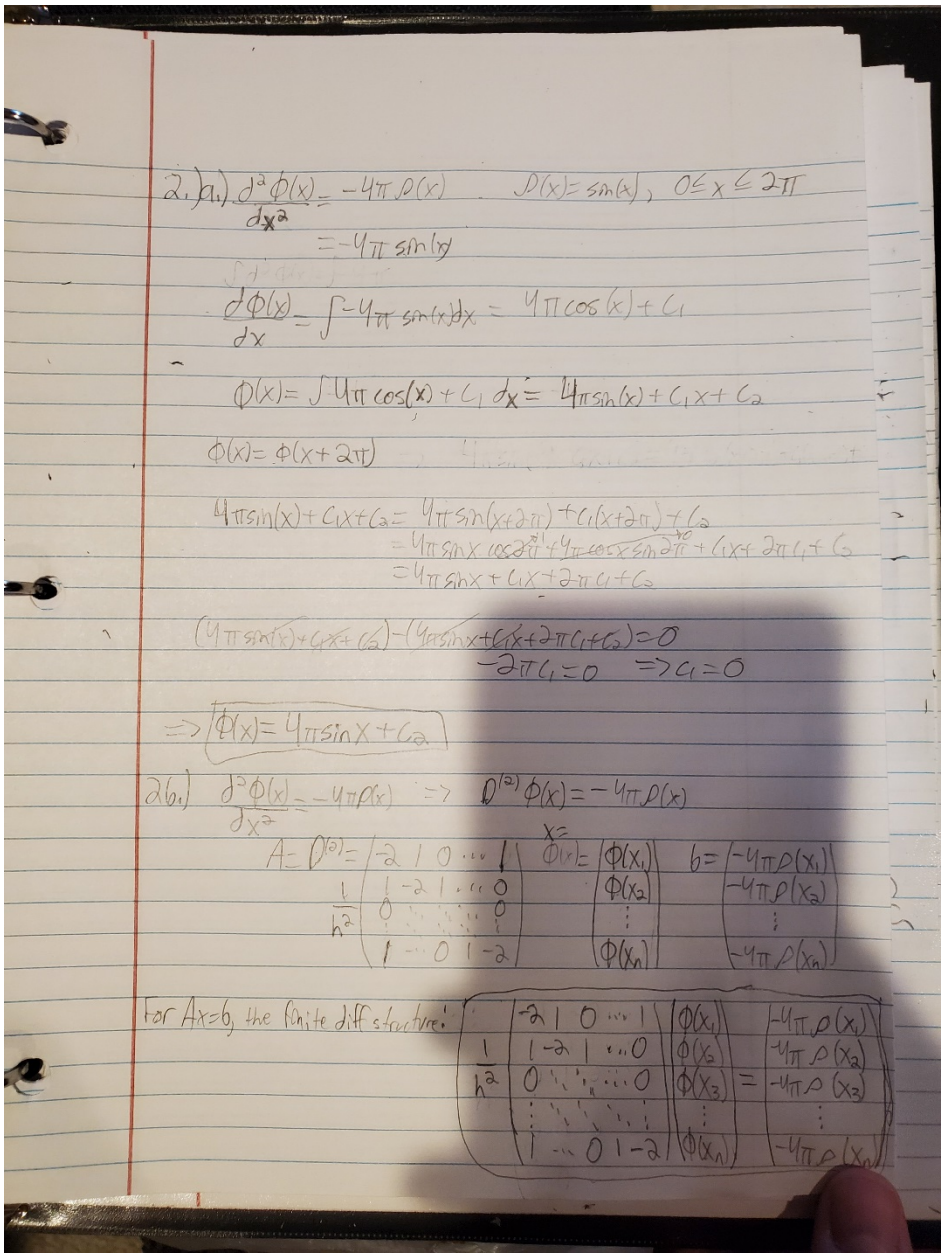
11/06/2020

PHYS 331, HW10

1a.) For A1 and b1, the exact solution is [1,1], and the Jacobi method solution is [1.00003499 1.00003499], which took 512 iterations to compute. These values agree to 0.0001 precision, and as such can be considered to be a valid solution. For A2 and b2, the exact solution is [1,1], and the Jacobi method solution is [0.99998306 0.99998306], which took only 9 iterations. These values agree to 0.0001 precision, and as such can be considered to be a valid solution. Solving for A2 and b2 took considerably fewer iterations than A1 and b1, taking only 9 instead of 512. This is because A1 is fairly ill-conditioned with C=100, and A2 is comparatively well conditioned with C=2. As a result, A2 converged on a solution considerably faster than A1. This implies that the efficiency of the Jacobi method, without relaxation, is still dependent on the conditioning of the matrix. For fun, I tried setting w=0.5 for A1, and this underrelaxation allowed it to converge in only 4 iterations. Thus relaxation is a solution to problems endemic to ill-conditioned matrices.

1b.) Without relaxation, this method oscillates around a point and does not actually converge on a solution. Because it oscillates around a solution, it needs to be underrelaxed to coax it inwards towards the solution. A relaxation factor, w, of 0 would obviously zero out the entire equation and thus is not a valid value. With a step size dw of 0.01, the next smallest candidate is 0.01, which is seen to work, thus this is the lower limit. A relaxation factor of 0.34 works, and 0.35 does not, thus 0.34 is the largest w for a dw of 0.01. Thus the range of valid relaxation factors is [0.01,0.34]. For the plot of iterations vs. w, there is an uptick at both ends, with a fairly stable basin in the middle. Particularly on the right side, at 0.34, the iteration count spikes very high. The best (lowest) iteration counts are in the middle, away from the edges. For the true error vs. w plot, the error is high at the left boundary, and decays as w increases, yielding more accurate values for higher values of w. The plot seems to approach the tolerance of 0.0001 as w increases, but does not reach this. This is to be expected, as the values that myJacobi yields are only accurate within that set tolerance, so the difference between the returned values and the expected values will be, at lowest, on that same order of magnitude, and never lower. The optimal choice of relaxation factor by iterations is anywhere from 0.05 to 0.25, as the difference in iterations is negligible along this range. The optimal choice of relaxation factor by true error is the higher the better, as true error is inversely related to w. Combining these two results, the optimal choice of w is approximately 0.25 as it minimizes both true error and iteration count.

2a&b.)



2c.) All solutions are essentially the same in their general shape and trend. They are positive sine waves, and hence periodic, over the interval 0 to 2pi with an amplitude of 4pi. The iterative and analytical solutions align exactly, with an discrepancies being too small to be seen from the graph (and likely within machine precision). They both oscillate around y=0, which is expected as the analytical solution assumes any vertical shifting to be negligible (constant of integration is set to 0). This is also expected from the finite difference solution as this model represents each rho(xi) as a sum of various phi(xi) terms, all of which are functions of x, and hence there is no constant term added to any solution. As a result, the sine wave this is expected to generate is not shifted up or down. They are both very smooth curves. This is expected from the analytical solution as it is a sine wave graphed over a fine mesh size. The iterative solution similarly adheres to the analytical solution's smoothness. The direct inverse solution has a similar general sinusoidal shape and a similar amplitude, but it varies in two ways. First,

while the two prior solutions have their vertical shifting held negligible, the direct inverse does not, and instead oscillates around approximately y=20. Additionally, the direct inverse only approximately follows the trend of a sine wave, and is instead quite jagged, fluctuating up and down rapidly in a roughly sinusoidal manner. The reason for both of these is the periodic nature of the differential operator. Removing the periodic terms A(1n) = 1 and and A(n1) = 1 both smooths the curve into precisely a sine wave and shifts it down to oscillate around y=0. This shows a distinct advantage of iterative methods for solving differential equations. While simply using linear algebra is computationally efficient, it offers a trade-off between smoothness and boundary conditions. The only way to have a smooth, useful curve from this method is to neglect boundary points and hence sacrifice a periodic solution. Making it periodic sacrifices the smoothness, and hence accuracy, of the solution. Iterative methods solve this, at the slight cost of computational efficiency, by offering valid periodic and non-periodic solutions with minimal error. Additionally, by incorporating relaxation into the iterative method, we can improve the computational efficiency greatly, so there are effectively no downsides to iterative methods. They are the clear choice for numerically solving differential equations.

3a.) The governing equation at this mesh point can be rewritten as: $T_2 + T_4 - 4T_5 + T_6 + T_8 = 0$. This corresponds to the 5th row of the matrix operator given.

3b.) The equation corresponding to the 6th row of the matrix operator, simply multiplied out, is: $T_3 + T_5 - 4T_6 + T_9 = 100$, although this should actually be $T_3 + T_5 - 4T_6 + T_9 + 100 = 0$, or $T_3 + T_5 - 4T_6 + T_9 = -100$. Following the general form given, this implies that $\frac{\partial^2 T_6}{\partial x^2} + \frac{\partial^2 T_6}{\partial y^2} = -100$. Looking at the figure, the mesh point 6 is bordering mesh point 5 to the left and the 100° boundary on the right, as well as mesh point 3 above and 9 below. Consider the x and y boundaries independently. Considering the second order differential operator in the y direction, then observing the sixth row, corresponds to an equation of the form $T_3 - 2T_6 + T_9 = 0$. For the x-axis, mesh point six is bordering a 100° source of heat. Considering the second order differential operator in the x direction, then observing the sixth row and adding in the bordering 100° temperature, corresponds to an equation of the form $T_3 - 2T_6 + 100 = 0$. Summing these two equations is equivalent to summing the x direction differential operator and the y direction differential operator. Hence, the equation $\frac{\partial^2 T_6}{\partial x^2} + \frac{\partial^2 T_6}{\partial y^2} = -100$ can be written as $(T_3 - 2T_6 + T_9) + (T_3 - 2T_6 + 100) = 0$, or $T_3 + T_5 - 4T_6 + T_9 = -100$.

3c.) At first, the textbook values for b were positive, which yielded negative temperatures for the mesh segments. Intuitively, this does not make sense, and as stated above mathematically, this is obviously not the case. Intuitively, equilibrium temperatures should be positive, and mathematically, the b vector should contain values less than or equal to zero. This can be corrected by flipping the sign of all the elements in the b vector. The temperature now makes sense, and can be viewed using the heat map in python.

EXTRA CREDIT:

4.) a.) $y_1'(x) = y_1(x) + 3y_2(x)$
$y_2'(x) = 4y_1(x) + 2y_2(x)$

$\begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} = \begin{pmatrix} y_1'(x) \\ y_2'(x) \end{pmatrix}$

$\boxed{\begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} y(x) = y'(x)}$

b.) $\begin{vmatrix} 1-\lambda & 3 \\ 4 & 2-\lambda \end{vmatrix} = (1-\lambda)(2-\lambda) - 12 = \lambda^2 - 3\lambda + 2 - 12 = \lambda^2 - 3\lambda - 10$

$\lambda^2 - 3\lambda - 10 = (\lambda - 5)(\lambda + 2) = 0 \quad \lambda = -2, 5$

for $\lambda = -2$ $\begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -2x_1 \\ -2x_2 \end{pmatrix}$ $x_1 + 3x_2 = -2x_1$ $3x_1 + 3x_2 = 0$

$v_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ $x_1 = -x_2$

for $\lambda = 5$ $\begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5x_1 \\ 5x_2 \end{pmatrix}$ $x_1 + 3x_2 = 5x_1$ $-4x_1 + 3x_2 = 0$

$v_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$

$\boxed{\lambda_1 = -2,\ \lambda_2 = 5,\ v_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix},\ v_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}}$

4c.) $y(x) = C_1 w_1 e^{\lambda_1 x} + C_2 w_2 e^{\lambda_2 x}$ $\quad y_1(0)=2.5 \quad y_2(0)=1$

$y_1(x) = C_1 e^{\lambda_1 x} + 3C_2 e^{\lambda_2 x}$

$y_2(x) = -C_1 e^{\lambda_1 x} + 4C_2 e^{\lambda_2 x}$

$y_1(0) = C_1 e^{\lambda_1 0} + 3C_2 e^{\lambda_2 0} = C_1 + 3C_2 = 2.5$

$y_2(0) = -C_1 e^{\lambda_1 0} + 4C_2 e^{\lambda_2 0} = -C_1 + 4C_2 = 1$

$7C_2 = 3.5 \quad C_2 = 0.5$

$-C_1 + 4C_2 = 1 \quad -C_1 + 2 = 1 \quad C_1 = 1$

$y_1(x) = e^{\lambda_1 x} + 1.5e^{\lambda_2 x}$

$y_2(x) = -e^{\lambda_1 x} + 2e^{\lambda_2 x}$

$y_1(0) = e^{\lambda_1 0} + 1.5e^{\lambda_2 0} = 1 + 1.5 = 2.5 = 2.5$ ✓

$y_2(0) = -e^{\lambda_1 0} + 2e^{\lambda_2 0} = -1 + 2 = 1 = 1$ ✓

5.) $\lambda = -2, 5$

$[D-\lambda_1][D-\lambda_2]y=0$

$[D+2][D-5]y=0$

$(D^2-3D-10)y=0$

$D^2y-3Dy-10y=0$

$y''(x)-3y'(x)-10y(x)=0$

$y_1(x)$ & $y_2(x)$ $\quad y_2(x)=y_1'(x) \quad y_1''(x)=y_2'(x)$

$\Rightarrow y_2(x)=y_1'(x)+3y_2(x)$

$y_1''(x)=y_1'(x)+3y_2'(x)=y_1'(x)+10y_1(x)+6y_2(x)$

$=13y_1(x)+9y_2(x)$