APPM 4600 — HOMEWORK # 11

For all homeworks, you should use Python. **Do not use** symbolic software such as Maple or Mathematica.

1. Use the transformation $t = x^{-1}$ and Composite Simpson's rule with 5 nodes to approximate

$$\int_1^\infty \frac{\cos(x)}{x^3} dx.$$

2. The gamma function is defined by the formula

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt, \quad x > 0.$$

   Write a program to compute the value of this function from the definition using each of the following approaches:

   (a) Truncate the infinite interval of integration and write a composite trapezoidal rule code to perform the numerical integration. You will need to do some experimentation or analysis to determine where to truncate the interval based upon the usual trade-offs between accuracy and efficiency. Please describe your reasoning for your choice of interval and step size for the Trapezoidal Rule. Compare the relative accuracy of this solution with the value given by the Python gamma function (`scipy.special.gamma`) at $x = 2, 4, 6, 8, 10$ (Recall $\Gamma(k) = (k-1)!$ for positive integer k) .

   (b) Use the Matlab adaptive quadrature routine `quad` to solve the above integral on the same interval you used for part (a). Compare the accuracy of this solution with the one you obtained in part (a) at the same values of x. Also, compare the number of function evaluations required by the two methods.

   (c) Gauss-Laguerre quadrature is designed for the interval $[0, \infty)$ and the weight function $w(t) = e^{-t}$. It is therefore ideal to use for this problem. Call the Numpy subroutine `numpy.polynomial.laguerre.laggauss` to to obtain the n weights $\mathbf{w}$ and n abscissae $\mathbf{x}$ for Gauss-Laguerre quadrature and use this to approximate $\Gamma(x)$.

3. Consider the linear system

$$\begin{aligned} 6x + 2y + 2z &= -2 \\ 2x + (2/3)y + (1/3)z &= 1 \\ x + 2y - z &= 0 \end{aligned}$$

   (a) Verify that $(x, y, z) = (2.6, -3.8, -5)$ is the exact solution.

   (b) Using 4 digit floating point arithmetic with rounding, solve the system via Gaussian elimination without pivoting.

   (c) Repeat part (a) with partial pivoting.

   (d) Which method is more accurate? i.e. stable.

(Remember to do the rounding to 4 significant digits as the machine would.)

4. Let H be a Householder matrix of size n; i.e. $\mathsf{H} = \mathsf{I} - 2\boldsymbol{w}\boldsymbol{w}^*$ where $\boldsymbol{w}^*\boldsymbol{w} = 1$.

   (a) Noting that H is both unitary and Hermitian, show that it is only possible that the eigenvalues are ± 1.

   (b) Show that $\mathsf{H}\boldsymbol{w} = -\boldsymbol{w}$ and that $\mathsf{H}\boldsymbol{v} = \boldsymbol{v}$ if $\boldsymbol{v}^*\boldsymbol{w} = 0$.

   (c) Determine again all the eigenvalues of H (part (c) provided all the eigenvectors).

   (d) Use the knowledge of the eigenvalues of H to determine $det(H)$.

5. We attempt to compute the QR decomposition of a 5×5 matrix A (e.g. using Householder). We reach the point at which we must orthogonalize the 3rd column vector of A. However, what we have found so far gives us the matrix factorization:

$$\mathsf{Q}^T\mathsf{A} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & r_{1,4} & r_{1,5} \\ 0 & r_{2,2} & r_{2,3} & r_{2,4} & r_{2,5} \\ 0 & 0 & 0 & r_{3,4} & r_{3,5} \\ 0 & 0 & 0 & r_{4,4} & r_{4,5} \\ 0 & 0 & 0 & r_{5,4} & r_{5,5} \end{bmatrix}$$

   (a) Explain how this implies that the columns of A are linearly dependent, and in particular, what it says about column $\boldsymbol{a}_3$.

   (b) I clearly can't apply the standard Gram-Schmidt or Householder QR algorithm to the third column of this matrix to find another orthogonal vector. However, I could if I swapped column 3 with either column 4 or 5 (Assume $r_{3,4} \neq 0$ and $r_{3,5} \neq 0$). Explain what criteria you would choose to pick and why.