

1.1)

The problem is to use numerical integration to find the perimeter of an ellipse. To perform numerical integration, I use the trapezoid method. The merits of the method are that for a given function, it's very easy to calculate as it involves just evaluating the function and taking a sum. Additionally, the method is scalable to any function over any domain. The trapezoid method does converge to the true result as the number of points you use goes to infinity. Furthermore, as the formula for the trapezoid method is just taking a sum, numerical integration should be backwards stable. One restriction on numerical integration is that the function you are integrating will have to be continuous on the domain you are integrating it over. Additionally, as you are taking a sum of functions, the function will need to be well conditioned to guarantee good results.

1.2)

The problem is to perform numerical integration using Gaussian quadrature, which uses a system of orthogonal polynomials to pick points to calculate a weighted sum that approximates the integral. There are issues with simplicity, as to perform Gaussian quadrature you have to know the weights, which requires you to differentiate and evaluate a polynomial, which gets harder for higher degrees. However, as it can be seen in the problem, at least for this equation, the integration converges quickly. Additionally, though the form of quadrature we discussed was over $[-1,1]$, you can scale it to any interval. Like 1.1, Gaussian quadrature is also backwards stable as you are taking a sum. Quadrature has the same issue with the function being continuous, and as you are using orthogonal polynomials, you could still run into the issue with ill conditioning.

3)

The problem is to find the dominant eigenvector and eigenvalue of a column stochastic matrix. To calculate this, I use power iteration. The merit of the method is that it will globally converge from any starting point to the dominant eigenvector. Additionally, it is more efficient at finding the eigenvector than other methods. Power iteration is scalable to any matrix of any size, and it's not necessarily backwards stable, as you are performing matrix vector multiplication. In the case of this problem, where the matrix is column stochastic, it will converge to the dominant eigenvector. However, generally, convergence cannot be guaranteed. A practical limitation of power iteration is that though it can find the dominant eigenvector, it becomes increasingly difficult to find all other eigenvectors and eigenvalues.