

Examination project 14: Inverse iteration method

Practical Programming and Numerical Methods 2020

Marco Majland

June 20, 2020

The objective of the project is to iteratively calculate the eigenvalue of a given real symmetric matrix, \mathbf{A} , closest to an input eigenvalue and its associated eigenvector. This is done using the inverse iteration method which is an improvement of the power iteration method. By shifting the matrix equation of the power method, one obtains the inverse iteration method.

Let \mathbf{A} denote a real symmetric matrix of dimension $n \times n$ and the initial guess of the eigenvalue and eigenvector be s and \mathbf{u} (normalized), respectively. The inverse iteration method is then given by

$$(\mathbf{A} - s\mathbf{1})\mathbf{v} = \mathbf{u} \quad (1)$$

where \mathbf{v} is the refined eigenvector of \mathbf{A} . In the implementation of the inverse iteration algorithm, the above linear matrix equation is solved using QR decomposition. During the iterations, one may update the matrix in 1 used in solving for the iterated vector, \mathbf{v} . The update, however, requires further QR decomposition at the cost of $O(n^3)$ operations, although this may yield faster convergence.

Test of the algorithm

To perform a test of the algorithm, a random real symmetric matrix \mathbf{A} of dimension 5×5 is generated. To benchmark the result, \mathbf{A} is diagonalized using the Jacobi diagonalization procedure implemented in problems 4 (PATH). A random eigenvalue, e_i , is chosen from the sequence of eigenvalues produced in the Jacobi procedure. The initial guess of the sought eigenvalue is parametrized as $s_0 = \delta e_i$ where δ is the deviation. Apparently, the inverse iteration method is sensitive to the deviation from the sought eigenvalue which is demonstrated later in the project. In this test of the algorithm, no Rayleigh updates are made for simplicity. Thus, Eq. 1 is solved iteratively until convergence. Since the resulting scalar s and vector \mathbf{v} should be an eigenvalue and eigenvector of \mathbf{A} , respectively, the convergence criteria for the absolute and relative accuracies are

$$\text{norm}(\mathbf{A}\mathbf{v}_i - s\mathbf{v}_i) \leq \tau \quad (\text{absolute}) \quad (2)$$

and

$$\frac{\text{norm}(\mathbf{A}\mathbf{v}_i - s\mathbf{v}_i)}{\text{norm}(\mathbf{A}\mathbf{v}_i) + \text{norm}(s\mathbf{v}_i)} \leq \frac{\epsilon}{2} \quad (\text{relative}). \quad (3)$$

Such an example is presented in the output file `/exam/test_out.txt`. The file contains the following information. The random matrix \mathbf{A} is generated and a random eigenvalue index is chosen after the Jacobi diagonalization. To track convergence to the correct eigenvalue, the eigenvalues closest to e_i in question are presented. The deviation δ is chosen, in the example, $\delta = 1.05$ which implies the initial eigenvalue guess deviates 5% from the Jacobi eigenvalue which is assumed to be the correct eigenvalue. Absolute and relative accuracies are chosen to be $\tau = \epsilon = 10^{-6}$. Two different approaches are made to the initial guess of the eigenvector. First, a random normalized vector is chosen. Second, the corresponding eigenvector of the Jacobi diagonalization for the sought eigenvalue is chosen, but the entries are deviated with δ . Then, the iteration result is presented. To ensure the correct eigenvector has been found, the scalar quantity

$$s = \mathbf{v}^T \mathbf{A} \mathbf{v} \quad (4)$$

since \mathbf{v} is an eigenvector of \mathbf{A} if the iteration procedure is succesful.

To investigate whether the eigenvalue may have converged to a neighbouring eigenvalue in the spectrum of \mathbf{A} , the deviation from these eigenvalues is also calculated.

NOTE : Only random eigenvalues up to dimension-1 are picked since the file compares the calculated eigenvalue to the neighbouring eigenvalues. Thus, if the program picks the last eigenvalue in the spectrum, this will return an error.

Test of convergence

The inverse iteration method converges slower and potentially to a wrong eigenvalue if the initial deviation is too large. Thus, by picking an eigenvalue closer to the sought eigenvalue, one achieves faster convergence. Furthermore, using the Rayleigh update, one may achieve even faster convergence. To demonstrate this, the inverse iteration method was applied on a random real symmetric matrix \mathbf{A} for different initial deviations, with and without the Rayleigh update. The result is depicted in figure 1. In this figure, the predictions are exhibited clearly. In the calculations, the matrix dimension is 30, $\tau = \epsilon = 10^{-6}$ and the Rayleigh update is performed at each iteration.

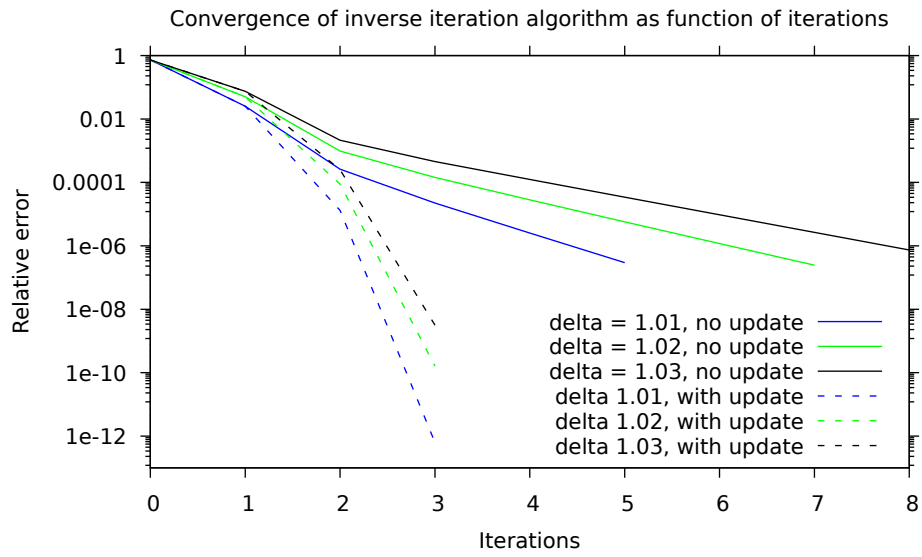


Figure 1: Relative error as function of iterations for the inverse iteration method applied on a random real symmetric matrix \mathbf{A} with and without the Rayleigh update.