How to compute the eigenvector when we know the eigenvalue?

For Numerical Linear Algebra fans

The algebraic form of a standard eigenvalue problem reads

$$A\boldsymbol{v} = \lambda\boldsymbol{v}, \tag{1}$$

where A is a real and complex matrix of size n and $\{\lambda_i, \boldsymbol{v}^{(i)}\}_{i=1}^{n}$ are the so-called eigenpairs with λ_i being the eigenvalue and $\boldsymbol{v}^{(i)}$ – its corresponding eigenvector.

Applications of eigenvalues and eigenvectors are found everywhere - in communication systems, designing bridges, car stereo systems, mechanical engineering, image compression, page ranking, spectral clustering (in data analysis), to name a few.

There are various methods to compute the eigenvalues. They are different for dense and sparse matrices and are provided in the major linear algebra packages. However, we need to do more computations to compute the eigenvectors, which adds to the overall computational complexity of the methods.

In this project we pose the question how to efficiently compute an eigenvector in the case when we know the eigenvalue exactly or within machine precision. From (1) we see directly that if λ_i is a known eigenvalue then $\boldsymbol{v}^{(i)}$ is the solution of the system (2)

$$(A - \lambda_i I)\boldsymbol{v}^{(i)} = \mathbf{0}, \tag{2}$$

where now the matrix $\widehat{A} = A - \lambda_i I$ is singular. (Here I is the idednity matrix of order n.) Due to the singularity of $\widehat{A}$ the solution of the linear system (2) is more difficult, as we seek a nonzero vector, lying in the nullspace of the matrix. In general, in such a case the standard methods to solve linear systems do not work. We cannot use Gauss elimination, the iterative methods have difficulties to converge. If we have an approximation of the eigenvector we can apply a nonlinear method to solve (2). Meanwhile the need to solve a singular system remains.

To present the idea in this project, let assume that the matrix A is real and has real eigenvalues. Let λ_i is a known simple eigenvalue, thus, λ_i is not a multiple eigenvalue. The idea to explore is to modify the system (2) and solve a nonsingular system instead. Consider

$$(A - \lambda_i I + Z_i)\boldsymbol{x} = \boldsymbol{z_i}, \tag{3}$$

where Z_i is a zero matrix with only one nonzero entry equal to 1 at position (i,i) and $\boldsymbol{z}_i$ is a zero vector with one nonzero element, equal to 1 at ith position. We note that the matrix $\widehat{A} = A - \lambda_i I + Z_i$ is not singular anymore and can be solved with conventional methods.

Several tests done in Matlab on relatively small matrices show that the solution vector $\boldsymbol{x}$ is equal to the eigenvector $\boldsymbol{v}^{(i)}$ up to normalization and sign change. Thus, $\boldsymbol{x}$ is an eigenvector, corresponding to λ_i.

Project tasks:

T1 Get acquainted with the standard eigenvalue problem, how it is solved numerically, and with some notions as eigenvalue (or spectral) decomposition, Schur decomposition. 'Meet' the matrix zoo - diagonalizable matrices, normal/non-normal matrices, positive definite, symmetric, Hermitian matrices.

T2 Do some literature search to see how the problem has been solved by others.

T3 Test the idea in Matlab on 'nice' matrices, which are symmetric, diagonally dominant and all eigenvalues are real and simple.

T4 Prove that the solution of (3) is an eigenvector of (2) for λ_i.

T5 Test the idea on more general types of matrices, only on real simple eigenvalues.

T6 Study if the idea is applicable for simple complex eigenvalues and corresponding complex eigenvectors.

T7 Study the case of multiple real eigenvalues. See if the idea is applicable and what modification could be made to $\widetilde{A}$ and the right-hand side to compute more that one eigenvector to λ_i.

T8 Perform extensive numerical tests and write a report.

Note: The question how do we obtain the exact eigenvalue is very relevant. In some cases it is known, as in the Google page ranking problem. For some classes of matrices, such as some Toeplitz matrices, all eigenvalues can be computed cheaply and very accurately, even for very large systems, not using conventional methods. Projects 9 and 10 in this course use such techniques but rely on other techniques to compute the eigenvectors.

Practical details

- Supervisor: Maya Neytcheva, Division of Scientific Computing, IT department, Uppsala University, `maya.neytcheva@it.uu.se`
- Prerequisites: Basic understanding of linear algebra.
- Preferred programming language: MATLAB.

Some references

[1] Gene H.Golub, Henk A.van der Vorst, Eigenvalue computation in the 20th century, *Computers and Mathematics with Applications*, 123 (2000), 35–65.
[2] J.J.Dongarra, C.B.Moler, J.H.Wilkinson, Improving the accuracy of computed eigenvalues and eigenvectors, *SIAM Journal on Numerical Analysis*, 20 (1983), 23-45.