Finding Rightmost Eigenvalues of Large Sparse Non-symmetric Parameterized Eigenvalue Problems

Minghao Wu
Applied Mathematics and Scientific Computation Program
Department of Mathematics
University of Maryland, College Park, MD
mwu@math.umd.edu

Advisor: Professor Howard Elman
Department of Computer Sciences
University of Maryland, College Park, MD
elman@cs.umd.edu

Introduction

Consider the eigenvalue problem

$$A_S x = \lambda B_S x \tag{1}$$

where A_S and B_S are large sparse non-symmetric real $N \times N$ matrices and S is a set of parameters given by the underlying Partial Differential Equation (PDE). For simplicity, I will drop the subscript S in the following discussion. People are interested in computing its rightmost eigenvalues (namely, eigenvalues with the largest real parts). The motivation lies in the determination of the stability of steady state solutions of non-linear systems of the form

$$B\frac{du}{dt} = f(u) \quad f : R^N \to R^N, u \in R^N \tag{2}$$

with large N and where u represents a state variable (velocity, pressure, temperature, etc). B is often called the mass matrix. Define the Jacabian matrix for the steady state u^* by $A_S = \partial f_S / \partial u\left(u^*\right)$, then u^* is stable if all the eigenvalues of (1) have negative real parts. Typically, f arises from the spatial discretization of a PDE. Interesting applications of this kind occur in stability analyses in fluid mechanics, structural engineering and chemical reactions. The problem of finding rightmost eigenvalues also frequently occurs in Markov chain models, economic modeling, simulation of power systems and magnetohydrodynamics. When finite differences are used to discretize a PDE, then often $B = I$ and (1) is called a standard eigenproblem. If the equations are discretized by finite elements, then the mass matrix $B \neq I$ and (1) is called a generalized eigenvalue problem. For problems arising from fluid mechanics, B is often singular.

Major computational difficulties of this kind of problems are: (1) both A and B are large and sparse, so the algorithm we use must be efficient in dealing with large systems; (2) in many applications, the rightmost eigenvalues are complex, so we must consider complex arithmetic; (3) B is often singular, so it will give rise to spurious eigenvalues.

Beside the numerical algorithm in computing the rightmost eigenvalues of (1), how the parameter set S gives rise to the bifurcation phenomena, i.e, the steady state solution exchanges in stability, is also of people's interest. Examples are the Rayleigh number in nonlinear diffusion equation (Olmstead model) and the Damköhler number in the tubular reactor model. As the parameters vary, the rightmost eigenvalues might cross the imaginary axis, thus the steady state solution becomes unstable.

Methodology

Eigenvalue Solvers

Since both A and B are large and sparse, the QZ-algorithm for the generalized problem and the QR-algorithm for the standard problem are not feasible. A more efficient approach is the solution of the standard eigenvalue problem $Tx = \theta x$, which is a transformation of $Ax = \lambda Bx$, by iterative methods like Arnoldi's method, subspace iteration and Lanczos' method. In this project, I will use Arnoldi's algorithm and its variants, such as the Implicitly Restarted Arnoldi algorithm and B - orthogonal Arnoldi algorithm. Arnoldi algorithm is a type of eigensolver based on Krylov spaces.

Matrix Transformation

Matrix transformation is crucial in solving problems like (1). There are two important reasons for this approach. First, a practical reason is that iterative methods like Arnoldi's method and subspace iteration cannot solve generalized eigenvalue problems, which makes a transformation necessary. A second reason is of a numerical nature. It is well known that iterative eigenvalue solvers applied to A quickly converge to the well-separated extreme eigenvalues of A. When A arises from the spatial discretization of a PDE, then the rightmost eigenvalues of A are in general not well separated. This implies slow convergence. The iterative method may converge to a wrong eigenvalue. Instead, one applies eigenvalue solvers to a transformation T with the aim of transforming the rightmost eigenvalues of A to well-separated extremal eigenvalues of T, which are easily found by the eigenvalue solvers we consider. I will explore three kinds of matix transformation: Shift-invert transformation, Cayley transformation and Chebyshev polynomial transformation.

Discretization of PDEs

In this project, finite difference and finite element method will be used to discretize the PDEs. They are the most commonly used methods in the discretization of PDEs.

Complex Arithmetic

Since the rightmost eigenvalues are complex in many real-world problems, complex arithmetic is considered in this project.

Implementation

All algorithms of this project will be coded in Matlab. Two software packages that are available are: IFISS ("Incompressible Fluid Iterative Solution Software") and the Implicitly Restarted Arnoldi Algorithm package written by Fei Xue. Computation can be done on laptops.

Testing and Validation

I will apply the algorithm to several test problems that have been solved and try to produce the same results as in the literature. There are three stages of testing:

Stage 1: Test the codes for standard Arnoldi algorithm, Shift-invert transformation and Cayley transformation

I plan to solve two test problems in this stage:

1. Olmstead model (nonlinear diffusion equation)

$$\frac{\partial u}{\partial t} = \frac{\partial^2 v}{\partial X^2} + C\frac{\partial u}{\partial X^2} + Ru - u^3$$
$$B\frac{\partial v}{\partial t} = (1-C)u - v$$

with certain boundary conditions.

2. Tubular reactor model

$$\frac{\partial y}{\partial t} = \frac{1}{Pem}\frac{\partial^2 y}{\partial X^2} - \frac{\partial y}{\partial X} - Dy\exp\left(\gamma - \gamma T^{-1}\right)$$
$$\frac{\partial T}{\partial t} = \frac{1}{Peh}\frac{\partial^2 T}{\partial X^2} - \frac{\partial T}{\partial X} - \beta\left(T - T_0\right) + BDy\exp\left(\gamma - \gamma T^{-1}\right)$$

with certain boundary conditions.

Stage 2: Test the codes for Implicitly Restarted Arnoldi algorithm and $\boldsymbol{B}$-orthogonal Arnoldi algorithm

In this stage, I will first use Matlab random number generator to generate matrices of the following structure:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

This kind of eigenvalue problem appears in the stability analysis of steady state solutions of Stokes and Navier-Stokes equations for incompressible flow. Since $\boldsymbol{B}$ is singular, standard Arnoldi algorithm will produce spurious eigenvalues. But Implicitly Restarted Arnoldi should be able to solve this problem.

Stage 3: Test the code on the eigenvalue problem arises from the mixed finite element discretization of real Navier-Stokes equations in literature.

The finite element discretization of this problem has the matrix structure in stage 2.

Project Schedule

Before November:

1. solve the first test problem (already finished)
2. explore the effect of Rayleigh number in the problem

November:

1. solve the second test problem
2. explore the effect of Damköhler number in the problem

December:

1. modify and test the Implicitly Restarted Arnoldi Algorithm
2. code and test $\boldsymbol{B}$ - orthogonal Arnoldi algorithm
3. finish midterm report
4. give midterm presentation

January and February:

1. discretize the two-dimentional double diffusive convection equation using Mixed Finite Element method
2. modify the code for the Mixed Finite Element method in IFISS

March:

Solve the eigenvalue problem arises from the Mixed Finite Element discretization of the two-dimentional double diffusive convection equation

April:

1. explore the effect of the parameters in the third model
2. write the final report

May:

1. write the final report
2. give final presentataion