Chapter 1

Jacobian-Free Newton-Krylov Solver

A Jacobian-Free Newton-Krylov (JFNK) solver is a general name for an algorithm that solves a system of nonlinear equations. As implied by the name, these solvers are characterized by two main features: lack of a need to form the exact Jacobian and Newton-like updates obtained from a Krylov method. The need for a Jacobian and an explanation of Newton updates will be discussed first. An overview of Krylov methods and the more specific Generalized Minimal Residual (GMRES) method will follow. The section will conclude with how the pieces fit together to form the JFNK solver used in this work.

1.1 Newton's Method

Systems of coupled, nonlinear equations are pervasive across all disciplines of science and engineering. They are also one of the most important and difficult class of problems to analyze and solve mathematically. While there are numerous techniques for solving such problems, Newton's method is the singular option that will be explored in this section.

1.1.1 Problem Statement

The nonlinear problem under consideration is defined as follows: given an $N \times 1$ vector of algebraic, nonlinear equations $F(x)$, determine an $N \times 1$ vector x^* such that

$$F(x^*) = 0 \tag{1.1}$$

This problem is sometimes referred to as a multi-dimensional root-finding problem. A different but mostly equivalent definition for the above problem is: given an $N \times 1$ vector of algebraic, nonlinear equations $F(x)$, determine an $N \times 1$ vector x^* such that

$$x^* = \underset{x}{\text{ArgMin}} \, \|F(x)\| \tag{1.2}$$

where $\|\cdot\|$ is any suitable norm for $F(x)$ and ArgMin_x is a function that returns the minimum possible value of its argument within the domain of x. Equation 1.1 is preferred for the linearizations to be discussed while equation 1.2 is preferred for the stopping criteria of Newton's method also to be discussed. Because the function $F(x)$ is taken to be nonlinear, a closed-form solution to either equation may be impractical, if not impossible, to find even for a modest value of N. Therefore, numerical techniques, like Newton's method, are frequently used for solving the problem approximately.

1.1.2 Newton Update

In an effort to make the solution of equation 1.1 tractable, Newton's method aims to form solution from a recursive series of linear approximations and solves. The method begins with some initial value of x denoted as x_0. If $F(x_0)$ is not sufficiently close to zero (see below), Newton's method seeks a better solution by computing a search direction Δx from x_0 via a linearized (first-order Taylor expanded) version of $F(x_0 + \Delta x)$. The Taylor expansion of $F(x_0 + \Delta x)$ about x_0 is

$$F(x_0 + \Delta x) = F(x_0) + \mathbb{J}_F(x_0)\Delta x + C(x_0, \Delta x) \tag{1.3}$$

where $\mathbb{J}_F(x_0)$ is an $N \times N$ Jacobian matrix and $C(x_0, \Delta x)$ is an $N \times 1$ vector of higher-order corrections that enforce the equality. To solve for Δx using Newton's method, the higher-order corrections are taken to be identically zero; therefore, each row of the approximate Taylor expansion is an equation for the tangent plane of the associated element of $F(x)$ at the guess value x_0. The search direction is then calculated from the intersections of all the tangent planes when they each equal zero; this can also be thought of as assuming the direction and magnitude of Δx leads to the exact solution of the problem. Using either interpretation, the resulting linear system used to solve for the direction is

$$\Delta x = -J_F^{-1}(x_0)F(x_0). \tag{1.4}$$

Any search direction calculated using the above formula, regardless of the point of evaluation, is called a Newton update or a Newton direction.

Of course, since the higher-order corrections were ignored in attaining the linear system, this Δx will unlikely lead to the exact solution. However, assuming Δx leads to a new point that reduces any suitable norm of $F(x_0)$, it is arguably a better point and can hopefully return an even better approximation. From the new point $x_1 = x_0 + \Delta x$, another linear system can be formed and the above linearize-then-solve process repeated.

Repeatedly updating the Taylor expansion point by solving a repeatedly updated linear system is the essence of Newton's method. Starting from x_0, a sequence of linearly updated vectors x_k is formed:

$$x_k = x_{k-1} - \mathbb{J}_F^{-1}(x_{k-1})F(x_{k-1}) \tag{1.5}$$

Setting aside algorithmic flow, stopping criteria for the recursion, and other items to be delineated, this update equation is Newton's method.

If the initial starting point x_0 is "close" to the true solution of the problem

1.2 Krylov Methods

Krylov methods are a class of techniques used to solve linear systems of the form

$$Ax = b \tag{1.6}$$

where x and b are $N \times 1$ vectors and A is an $N \times N$ nonsingular matrix. Different methods place other limitations on A, but the most general condition (and the only condition required by GMRES) is that the coefficient matrix A be invertible. The exact solution to equation 1.6 is simply

$$x = A^{-1}b. \tag{1.7}$$

While trivially written symbolically, if N is large, the exact inversion of A can be computationally and memory intensive. The computational cost of inverting A is only exacerbated when solving nonlinear problems with a Newton-like scheme since several, if not many, linear solves are required to make one nonlinear advancement.

The solution in equation 1.7 will now be rewritten in a form that expresses the solution as a correction from some arbitrary initial guess and is inverse-free. The new form is introduced to reflect how these methods are often presented in the literature, which often lack such a discussion. Additionally, the form does lend itself well toward expressing the solution and error which accompany any Krylov method.

Given an initial guess vector x_0 with an associated residual vector $r_0 = b - Ax_0$, a new vector $\tilde{x}$ can be created by shifting in a direction Δx off of the guess vector: $\tilde{x} = x_0 + \Delta x$. If the new vector is not the solution to the problem, the shift induces a new residual $\tilde{r}$ of the form

$$\tilde{r} = r_0 - A\Delta x. \tag{1.8}$$

Assuming $\tilde{x}$ is the true solution x, the induced residual $\tilde{r}$ is precisely 0, and the shift is therefore $\Delta x = A^{-1}r_0$. The solution can then be written in a guess-correction form:

$$\tilde{x} = x_0 + A^{-1}r_0. \tag{1.9}$$

The burden of calculating A's inverse is still present. To avoid this problem, we use a result of the Cayley-Hamilton theorem: the inverse of a square $N \times N$ matrix can be expressed as a linear combination of the matrix's powers from 0 to $N-1$. To wit, given the proper weights c_i, the following equality is satisfied: $A^{-1} = \sum c_i A^i$. Substituting this expansion into equation 1.9 yields

$$\tilde{x} = x_0 + \sum_{i=0}^{N-1} c_i A^i r_0. \tag{1.10}$$

The true solution to the problem is now in a form that incorporates some initial guess and is free of matrix inversions. The weights c_i, which are required for the equality to hold, are based on A's characteristic polynomial and are, therefore, difficult to efficiently compute for large N. In truth, any method that aims to find an exact solution will be hampered by computational complexity for large systems.

Krylov methods aim to mitigate computational deficiencies by finding an approximate solution. The search for an approximate solution does not render all of the above work worthless but forms its heart. In fact, the set of vectors $\{A^i r_0\}$ present in equation 1.10 is called the Krylov subspace of A, denoted as $\mathcal{K}_N(A, r_0)$, and is the basis for all Krylov methods. That set of vectors is of dimension N and will hereafter be referred to as the full subspace. Typically, the methods to be discussed use a variable number m of lower dimensional spaces that will likewise be denoted as $\mathcal{K}_m(A, r_0)$. The approximate solution formed by Krylov methods is based on one simplification to equation 1.10: only form as much of the full subspace as needed. That is, the approximation is built from an iteratively grown set of chosen basis vectors that span the lower dimensions of the full subspace. The maximum number of basis vectors needed varies by problem, but it is guaranteed that the approximation becomes better as the maximum increases.

Thus, letting Z_m be an $N \times m$ matrix whose columns form a basis for $\mathcal{K}_m(A, r_0)$ and w_m be an $m \times 1$ weight vector (analogous to the aforementioned c_i), the m-th approximate solution is

$$x_m = x_0 + Z_m w_m. \tag{1.11}$$

A side effect of forming only a subset of the full Krylov subspace is that the weights are no longer based on the matrix's characteristic polynomial. To remedy this problem, Krylov methods define the weights through some optimality condition placed upon the system at the current iteration. Therefore, Krylov methods are defined by their choice of basis vectors and optimality condition. The discussion of GMRES concretely covers both of these choices.

1.3 GMRES

GMRES was first proposed by Saad and Schultz in 1986. The original formulation used the starting basis vector

1.3.1 Algorithm

1.4 Jacobian-Free Augmentation