

## 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. Since the ability to avoid creating the exact Jacobian comes directly from using a Krylov method, a brief overview of those methods will be given before discussing the particular one leveraged in this work's JFNK solver: Generalized Minimal Residual (GMRES).

## 1.1 Newton's Method

Newton's method is an iterative technique that seeks to find a vector  $x$  such that

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

where  $x$  is an unknown  $N \times 1$  vector and the function  $F(x)$  is a known nonlinear  $N \times 1$  vector-valued function of  $x$ . Because the function  $F(x)$  is taken to be nonlinear, a closed-form solution to the equation may be impractical, if not impossible, to find even for a modest value of  $N$ .

In an effort to make the solution of equation [1.1](#) tractable, Newton's method aims to form an

analytical 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 shift  $\Delta x$  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

$$F(x_0 + \Delta x) = F(x_0) + \mathbb{J}_F(x_0)\Delta x + C(x_0, \Delta x) \quad (1.2)$$

where  $\mathbb{J}_F(x_0)$  is the  $N \times N$  Jacobian matrix and  $C(x_0, \Delta x)$  is a higher-order correction that enforces the equality. In order to solve for the shift, Newton's method makes two assumptions:  $x_0 + \Delta x$  is the exact solution to the problem and all higher-order corrections are negligible. These assumptions allow both  $F(x_0 + \Delta x)$  and  $C(x_0, \Delta x)$  to be set to zero, and the resulting linear system can be solved for the unknown shift

$$\Delta x = -J_F^{-1}(x_0)F(x_0). \quad (1.3)$$

Of course, since the higher-order corrections were ignored in attaining the linear system, this  $\Delta x$  will unlikely lead to the exact solution. However, since  $\Delta x$  is in the opposite direction of the Jacobian-function product, it is in a direction toward lower values (called a descent direction) and could lead to a better solution. From the new point  $x_1 = x_0 + \Delta x$ , another linear system can be formed and the process repeated ad infinitum until, hopefully, the solution is reached. 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:

$$x_k = x_{k-1} - J_F^{-1}(x_{k-1})F(x_{k-1}) \quad (1.4)$$

Complications set aside for the moment, update equation is the core of Newton's method.

## 1.2 Krylov Methods

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

$$Ax = b \tag{1.5}$$

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 1.5 is simply

$$x = A^{-1}b. \tag{1.6}$$

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 1.6 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  $\Delta x$  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

$$\tilde{r} = r_0 - A\Delta x. \quad (1.7)$$

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:

$$\tilde{x} = x_0 + A^{-1}r_0. \quad (1.8)$$

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 1.8 yields

$$\tilde{x} = x_0 + \sum_{i=0}^{N-1} c_i A^i r_0. \quad (1.9)$$

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 1.9 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 1.9: 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

$$x_m = x_0 + Z_m w_m. \quad (1.10)$$

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**