

## Chapter 1

# Jacobian-Free Newton-Krylov Solver

A Jacobian-Free Newton-Krylov (JFNK) solver is a generic name for an algorithm that solves a system of nonlinear equations. As implied by the name, these solvers are defined by two main characteristics: 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 method leveraged in this work's JFNK solver: Generalized Minimal Residual (GMRES).

## 1.1 Krylov Methods

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

$$Ax = b \tag{1.1}$$

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 be invertible. The exact solution to equation [1.1](#) is trivially

$$x = A^{-1}b. \tag{1.2}$$

However, 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 1.2 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 Krylov 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

$$\tilde{r} = r_0 - A\Delta x. \quad (1.3)$$

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:

$$\tilde{x} = x_0 + A^{-1}r_0. \quad (1.4)$$

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 inverse of matrix  $A$  can be expressed as  $A^{-1} = \sum_i c_i A^i$  where  $i \in \{0, \dots, N - 1\}$ . Substituting this expansion into equation 1.4 yields

$$\tilde{x} = x_0 + \sum_{i=0}^{N-1} c_i A^i r_0. \quad (1.5)$$

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, however, does not render all of the above work worthless. In fact, Krylov methods are based entirely on the set of vectors  $\{A^i r_0\}_{i \in \{0, \dots, N-1\}}$  present in equation 1.5. This set of vectors is called the Krylov subspace of  $A$  and is, as the name may indicate, the basis for all Krylov methods. A Krylov subspace of dimension  $m$  is denoted as

$$\mathcal{K}_m(A, r_0) = \{r_0, Ar_0, A^2 r_0, \dots, A^{m-1} r_0\}. \quad (1.6)$$

Starting with an arbitrary initial guess  $x_0$ , we define the  $k$ -th approximate solution  $x_k$  as

$$x_k = x_0 + Z_k \omega_k, \quad (1.7)$$

where  $Z_k$  is an  $N \times k$  matrix whose columns form a basis for  $\mathcal{K}_k(A, r_0)$  and  $\omega_k$  is an  $N \times 1$  weight vector.

## 1.2 GMRES

### 1.2.1 Algorithm

## 1.3 Jacobian-Free Augmentation