

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 $r(x)$, determine an $N \times 1$ vector x^* such that

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

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 $r(x)$, determine an $N \times 1$ vector x^* such that

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

where $\|\cdot\|$ is any suitable norm for $r(x)$ and ArgMin_x is a function that returns the argument which minimizes $\|r(x)\|$ over 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 $r(x)$ 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 a solution from a recursive series of linear approximations and solves. The method begins with some initial value of x denoted as x_0 . If $r(x_0)$ 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 $r(x_0 + \Delta x)$. The Taylor expansion of $r(x_0 + \Delta x)$ about x_0 is

$$r(x_0 + \Delta x) = r(x_0) + J_r(x_0)\Delta x + c(x_0, \Delta x) \quad (1.3)$$

where $J_r(x_0)$ is an $N \times N$ Jacobian matrix and $c(x_0, \Delta x)$ is an $N \times 1$ vector of higher-order corrections that ensure the equality. To solve for Δx using Newton's method, the stepped residual $r(x_0 + \Delta x)$ and higher-order correction 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 $r(x)$ at the guess value x_0 . This procedure can be thought of as assuming the direction and magnitude of Δx leads to the exact solution of the problem. The resulting linear system used to solve for the direction is

$$J_r^{-1}(x_0)\Delta x = -r(x_0). \quad (1.4)$$

Any search direction calculated from 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 a norm of $r(x_0)$, it is arguably a better point and can likely be used to return an even better approximation. From the new point $x_1 = x_0 + \Delta x$, another linear system can be formed and a new search direction calculated

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:

$$x_k = x_{k-1} - J_r^{-1}(x_{k-1})r(x_{k-1}) \quad (1.5)$$

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

1.2 Krylov Methods

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

$$Ax = b \quad (1.6)$$

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 a required condition for all methods is that the coefficient matrix A be invertible. The exact solution to [equation 1.6](#) is simply

$$x = A^{-1}b. \quad (1.7)$$

Although trivially written, 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.

Krylov methods aim to provide an approximate solution to [equation 1.6](#) without ever forming the inverse of A explicitly. To explain the procedure used, first, the solution will be written

in an inverse-free form. Then, a general explanation of Krylov subspaces and methods will be given. The section will conclude with the particular Krylov method used for this work: GMRES.

1.2.1 Inverse-free Form

Let x_N represent the exact solution to [equation 1.6](#) and x_0 represent an initial guess to the solution. Assuming the problem is solvable, there exists some Δx such that $x_N = x_0 + \Delta x$. Therefore, the residual vector for the initial guess from the solution would be

$$r_0 = b - Ax_0 \tag{1.8}$$

Isolating b on the left-hand side and multiplying through by A^{-1} then gives

$$A^{-1}b = x_0 + A^{-1}r_0 \tag{1.9}$$

Substituting this into [equation 1.7](#) gives

$$x_N = x_0 + A^{-1}r_0. \tag{1.10}$$

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 1.10](#) yields

$$x_N = x_0 + \sum_{i=0}^{N-1} c_i A^i r_0. \tag{1.11}$$

The true solution to the problem is now in a form that incorporates some initial guess and is free of matrix inversions. However, the weights c_i , which are required for the equality to hold, are based on A 's characteristic polynomial. Calculating these weights exactly are, as in the explicit calculation of the inverse, difficult to efficiently compute for large N and an approximation must be made for tractability.

1.2.2 General Method

The set of vectors $\{A^i r_0\}$ present in [equation 1.11](#) is called the Krylov subspace of A , denoted $\mathcal{K}_N(A, r_0)$, and is the basis for all Krylov methods. The set 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 $\mathcal{K}_m(A, r_0)$.

The approximate solution formed by Krylov methods is based on one simplification to [equation 1.11](#): only form as much of the full subspace as needed. At every iteration m of a Krylov method, a vector is chosen from the Krylov subspace and added to the current vector set $\mathcal{K}_m(A, r_0)$. Since the subspace is grown through, essentially, a power iteration scheme, the newly generated vector is typically orthogonalized against all previously added vectors through any appropriate process (e.g., Gram-Schmidt) to ensure good conditioning. This vector set is then used as the basis for the approximate solution to the linear problem. The set is grown and approximate solution updated until the norm of the residual is sufficiently low (typically, a user-defined criterion). The maximum number of basis vectors needed varies by problem, but it is guaranteed that the approximation will reach the true solution once the full subspace is formed (assuming exact arithmetic).

Let 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). At the m -th iteration of the Krylov method, a new approximate solution Δx_m is calculated from the linear combination $Z_m w_m$. Therefore, the m -th approximate solution is

$$x_m = x_0 + \Delta x_m = x_0 + Z_m w_m. \quad (1.12)$$

A side effect of forming only a subset of the full subspace is that the weights w_m are no longer based on A 's characteristic polynomial. To remedy this problem, Krylov methods define an optimality condition that results in weights that attempt to reduce the residual every iteration.

Since Krylov methods are a family, a specific method arises from consideration of A 's structure, choice of the basis vectors for Z_m , and a choice of optimality condition. GMRES will give concreteness to these three choices.

1.2.3 GMRES

First presented by Saad and Schultz in 1986 ([NEED REFERENCE](#)), GMRES is a Krylov method designed to solve any invertible, square matrix A . For every iteration m of the method, a new approximate solution is calculated from the minimization problem

$$x_m = \text{ArgMin} \|b - Ax_m\|_2^2 \quad (1.13)$$

Using [equations 1.8](#) and [1.12](#), the minimization problem can be rewritten in terms of the unknown weights:

$$w_m = \text{ArgMin } \|r_0 - AZ_m w_m\|_2^2. \quad (1.14)$$

However, subject to this condition, the methodology of Krylov methods presents a new problem since only a subset of the full subspace is formed: solving for the unknown weights is an over-constrained problem. The original method from 1986 proposed an iteratively grown Hessenberg decomposition, but this work implements the procedure of [NEED REFERENCE](#) that maintains a persistent QR factorization of the problem.

The coefficient matrix for [equation 1.14](#) is the $N \times m$ matrix AZ_m with $m < N$. The QR factorization of that matrix has the form

$$AZ_m = \tilde{Q}_m \tilde{R}_m = \begin{bmatrix} Q_m & \times \end{bmatrix} \begin{bmatrix} R_m \\ 0 \end{bmatrix} = Q_m R_m \quad (1.15)$$

where \tilde{Q}_m is an $N \times N$ orthogonal matrix, \tilde{R}_m is a full rank rectangular matrix, Q_m is an $N \times m$ rectangle, and R_m is an $m \times m$ upper-triangular matrix. The matrices \tilde{Q}_m and \tilde{R}_m comprise what is called the full factorization, and Q_m and R_m comprise the reduced factorization.

The QR factorization makes solving the minimization problem straightforward. There are three key properties of the orthogonal matrix \tilde{Q}_m that aid in this solution: the transpose \tilde{Q}_m^T is also the inverse, the transpose is also orthogonal, and orthogonal matrices do not change the 2-norm of a vector ($\|Qx\|_2 = \|x\|_2$).

Using the QR factorization of the coefficient matrix and the properties of \tilde{Q}_m , [equation 1.14](#)

has the equivalent form

$$\begin{aligned}
w_m &= \text{ArgMin} \|r_0 - \tilde{Q}_m \tilde{R}_m w_m\|_2^2 \\
&= \text{ArgMin} \|\tilde{Q}_m^T(r_0 - \tilde{Q}_m \tilde{R}_m w_m)\|_2^2 \\
&= \text{ArgMin} \|\tilde{Q}_m^T r_0 - \tilde{R}_m w_m\|_2^2
\end{aligned}$$

The problem can be further simplified using the reduced factorization to yield

$$w_m = \text{ArgMin} \|Q_m^T r_0 - R_m w_m\|_2^2 \quad (1.16)$$

Since the bottom $N - m$ rows of \tilde{R}_m are 0, the reduced factorization omits them and presents a square, invertible system. The weights can now be calculated from the linear system

$$R_m w_m = Q_m^T r_0 \quad (1.17)$$

1.2.4 Algorithm

1.3 Jacobian-Free Augmentation