

Extrapolation methods as nonlinear Krylov subspace methods

Conor McCoid^a, Martin J. Gander^b

^a*Université Laval, 2325 rue de l'Université, Québec, QC G1J 3V4, Canada*

^b*Université de Genève, 24 rue de Général-Dufour, 1211 Genève 4, Switzerland*

Abstract

When applied to linear vector sequences, extrapolation methods are equivalent to Krylov subspace methods. Both types of methods can be expressed as particular cases of the multiseccant equations, the secant method generalized to higher dimensions. Through these equations, there is also equivalence with a variety of quasi-Newton methods. This paper presents a framework to connect these various methods.

Keywords: Krylov subspace methods, extrapolation methods, quasi-Newton methods, GMRES, MPE, multiseccant equations, root-finding methods

2020 MSC: 65F25, 65B05, 90C53

1. Introduction

The equivalence between extrapolation methods and Krylov subspace methods is well-studied [1, 2, 3]. These works have largely focused on individual extrapolation methods and the orthogonalization processes involved in each. Equivalence between Krylov subspace methods and quasi-Newton methods, including the multiseccant equations, is less studied but still known [4]. Gragg and Stewart describe using a QR factorization to solve the multiseccant equations [5]; if the function evaluations form a Krylov subspace this would be exactly GMRES [6].

Sidi [7] has developed a framework for extrapolation methods, while Fang and Saad [8] have developed one for quasi-Newton methods, but neither includes consideration of the other type of methods, nor Krylov subspace methods. While the framework presented here does not include every method considered in both of these previous frameworks, it shows all three types of methods connect fundamentally.

To show these connections in the simplest way possible, we begin by considering the multiseccant equations in various forms. From these various forms we build quasi-Newton methods, extrapolation methods, and Krylov subspace methods.

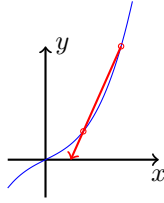


Figure 1: Example of the secant method. Two points $(x_1, f(x_1))$ and $(x_2, f(x_2))$ are used to draw a line. The zero of this line is then used as the next estimate of the root of $f(x)$.

2. Multisecant equations

The multisecant equations are a generalization of the secant method into higher dimensions. Recall that the secant method seeks the root of the function $f(x)$ by computing an approximation of the derivative $f'(x_n)$:

$$\hat{x} = x_{n+1} - (x_{n+1} - x_n) (f(x_{n+1}) - f(x_n))^{-1} f(x_{n+1}).$$

In higher dimensions it is necessary to find an approximation to the Jacobian $J(\mathbf{x}_n)$. To do so, one can expand $\mathbf{f}(\mathbf{x}_{n+i})$, the function evaluated at a given point \mathbf{x}_{n+i} , into a Taylor series about some other point \mathbf{x}_n :

$$\mathbf{f}(\mathbf{x}_{n+i}) = \mathbf{f}(\mathbf{x}_n) + J(\mathbf{x}_n)(\mathbf{x}_{n+i} - \mathbf{x}_n) + \frac{1}{2}(\mathbf{x}_{n+i} - \mathbf{x}_n)^\top H(\mathbf{x}_n)(\mathbf{x}_{n+i} - \mathbf{x}_n) + \dots$$

As a first order approximation we can take the first two terms of this series, resulting in the following approximate equation [9]:

$$\mathbf{f}(\mathbf{x}_{n+i}) - \mathbf{f}(\mathbf{x}_n) \approx J(\mathbf{x}_n)(\mathbf{x}_{n+i} - \mathbf{x}_n).$$

This system can be solved for $J(\mathbf{x}_n)$, though such a system would be underdetermined. However, if one had as many $\mathbf{f}(\mathbf{x}_{n+i})$ as there are dimensions in the space, d , then one could solve

$$[\mathbf{f}(\mathbf{x}_{n+1}) \quad \dots \quad \mathbf{f}(\mathbf{x}_{n+d})] - \mathbf{f}(\mathbf{x}_n)\mathbf{1}^\top = \hat{J}([\mathbf{x}_{n+1} \quad \dots \quad \mathbf{x}_{n+d}] - \mathbf{x}_n\mathbf{1}^\top), \quad (1)$$

where \hat{J} approximates $J(\mathbf{x}_n)$. This system is nonsingular given sufficient conditions on the choice of the \mathbf{x}_{n+i} .

Now that an approximation \hat{J} has been found for the Jacobian, an approximate root may be calculated:

$$\hat{\mathbf{x}} = \mathbf{x}_n - \hat{J}^{-1}\mathbf{f}(\mathbf{x}_n). \quad (2)$$

This system is referred to as the multisecant equations. They can be further generalized by allowing \hat{J} to be the solution of an underdetermined system, such as by using fewer than d function evaluations and multiplying both sides of equation (1) by a matrix B^\top .

The multiseccant equations are but one instance of several equivalent methods. We begin with a specific form and prove the general form. Let

$$\begin{aligned} F_{n,k} &= [\mathbf{f}(\mathbf{x}_n) \quad \dots \quad \mathbf{f}(\mathbf{x}_{n+k})], \\ X_{n,k} &= [\mathbf{x}_n \quad \dots \quad \mathbf{x}_{n+k}], \\ \Delta_n &= \begin{bmatrix} -1 & \dots & -1 \\ 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} \end{aligned}$$

then equation (1) may be rewritten as

$$\hat{J}^{-1} F_{n,k} \Delta_n = X_{n,k} \Delta_n. \quad (3)$$

Recall from equation (2) that we require the vector $\hat{J}^{-1} \mathbf{f}(\mathbf{x}_n)$ to compute our approximation to the root $\hat{\mathbf{x}}$. If there exists a vector $\tilde{\mathbf{u}}$ such that $F_{n,k} \Delta_n \tilde{\mathbf{u}} = \mathbf{f}(\mathbf{x}_n)$ then $\hat{J}^{-1} \mathbf{f}(\mathbf{x}_n) = X_{n,k} \Delta_n \tilde{\mathbf{u}}$. Thus, the multiseccant equations can be represented in the following compact form:

$$F_{n,k} \Delta_n \tilde{\mathbf{u}} = \mathbf{f}(\mathbf{x}_n), \quad \hat{\mathbf{x}} = \mathbf{x}_n - X_{n,k} \Delta_n \tilde{\mathbf{u}}. \quad (4)$$

If there are fewer function evaluations than the dimension of the space, $k < d$, the equation to solve \hat{J} is underdetermined and the first system of equation (4) is overdetermined. There are two options to then solve this system: either to pad out the matrix $F_{n,k}$ with additional vectors, or add constraints. We will focus on the latter, but will discuss the former in Section 3. Adding constraints results in the underdetermined Newtonian form of the multiseccant equations:

$$B^\top F_{n,k} \Delta_n \tilde{\mathbf{u}} = B^\top \mathbf{f}(\mathbf{x}_n), \quad \hat{\mathbf{x}} = \mathbf{x}_n - X_{n,k} \Delta_n \tilde{\mathbf{u}}. \quad (5)$$

One can replace $\Delta_n \tilde{\mathbf{u}}$ with $\hat{\mathbf{u}}$ for the right of equation (5), so that $\hat{\mathbf{x}} = \mathbf{x}_n - X_{n,k} \hat{\mathbf{u}}$. We then need a system for which $\hat{\mathbf{u}}$ is a solution. Note first that $\tilde{\mathbf{u}} \in \mathbb{R}^k$ while $\hat{\mathbf{u}} \in \mathbb{R}^{k+1}$, and so we require an additional constraint on $\hat{\mathbf{u}}$ not found in the system for $\tilde{\mathbf{u}}$. Since the columns of Δ_n sum to zero this constraint is $\mathbf{1}^\top \hat{\mathbf{u}} = 0$. The remaining k constraints come from replacing $\Delta_n \tilde{\mathbf{u}}$ in the left of equation (5), resulting in

$$\begin{bmatrix} \mathbf{1}^\top \\ B^\top F_{n,k} \end{bmatrix} \hat{\mathbf{u}} = \begin{bmatrix} 0 \\ B^\top \mathbf{f}(\mathbf{x}_n) \end{bmatrix}, \quad \hat{\mathbf{x}} = \mathbf{x}_n - X_{n,k} \hat{\mathbf{u}}. \quad (6)$$

We must now prove there is a bijection between $\tilde{\mathbf{u}}$ and $\hat{\mathbf{u}}$.

Proposition 1. *For all $\hat{\mathbf{u}} \in \mathbb{R}^k$ there exists a unique $\tilde{\mathbf{u}} \in \mathbb{R}^{k+1}$ such that $\hat{\mathbf{u}} = \Delta \tilde{\mathbf{u}}$ where $\mathbf{1}^\top \hat{\mathbf{u}} = 0$, the columns of Δ sum to zero, and its first k rows form an invertible matrix.*

Proof. Using the constraint $\mathbf{1}^\top \hat{\mathbf{u}} = 0$ one can write $\hat{u}_{k+1} = -\mathbf{1}^\top R_k \hat{\mathbf{u}}$, where R_k is the restriction operator that takes the first k elements of a vector of length $k+1$. Then $\tilde{\mathbf{u}}$ can be found by solving the system $R_k \Delta \tilde{\mathbf{u}} = R_k \hat{\mathbf{u}}$. The matrix $R_k \Delta$ is square and invertible, meaning $R_k \hat{\mathbf{u}}$ uniquely determines $\tilde{\mathbf{u}}$. Since $R_k \hat{\mathbf{u}}$ also uniquely determines \hat{u}_{k+1} , there exists exactly one $\tilde{\mathbf{u}}$ for any given $\hat{\mathbf{u}}$. \square

Note that Proposition 1 applies for any matrix $\Delta \in \mathbb{R}^{(k+1) \times k}$ satisfying the stated conditions, and not merely Δ_n . This allows us to apply the result broadly.

Since \mathbf{x}_n is the first column of $X_{n,k}$, the equation for $\hat{\mathbf{x}}$ in (6) can be written as

$$\hat{\mathbf{x}} = \mathbf{x}_n - X_{n,k} \hat{\mathbf{u}} = X_{n,k} (\mathbf{e}_1 - \hat{\mathbf{u}}) =: X_{n,k} \mathbf{u}.$$

We can then find a system for the vector $\mathbf{u} = \mathbf{e}_1 - \hat{\mathbf{u}}$ by multiplying \mathbf{u} by the matrix found in equation (6). The products of this matrix with the vectors \mathbf{e}_1 and $\hat{\mathbf{u}}$ are known:

$$\begin{aligned} \begin{bmatrix} \mathbf{1}^\top \\ B^\top F_{n,k} \end{bmatrix} \mathbf{u} &= \begin{bmatrix} \mathbf{1}^\top \\ B^\top F_{n,k} \end{bmatrix} (\mathbf{e}_1 - \hat{\mathbf{u}}) \\ &= \begin{bmatrix} 1 \\ B^\top \mathbf{f}(\mathbf{x}_n) \end{bmatrix} - \begin{bmatrix} 0 \\ B^\top \mathbf{f}(\mathbf{x}_n) \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}. \end{aligned}$$

This leaves us with the base form of the multiseccant equations [10]:

$$\begin{bmatrix} \mathbf{1}^\top \\ B^\top F_{n,k} \end{bmatrix} \mathbf{u} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}, \quad \hat{\mathbf{x}} = X_{n,k} \mathbf{u}. \quad (7)$$

From this base form one can transform into several equivalent forms. One interesting example is to use the transform $\mathbf{u} = \mathbf{e}_{i+1} - \hat{\mathbf{u}}$, giving

$$\begin{bmatrix} \mathbf{1}^\top \\ B^\top F_{n,k} \end{bmatrix} \hat{\mathbf{u}} = \begin{bmatrix} 0 \\ B^\top \mathbf{f}(\mathbf{x}_{n+i}) \end{bmatrix}, \quad \hat{\mathbf{x}} = \mathbf{x}_{n+i} - X_{n,k} \hat{\mathbf{u}}.$$

This indicates that our solution $\hat{\mathbf{x}}$ is also the solution found from equation (2) replacing n with $n+i$ and keeping \hat{J} fixed:

$$\hat{\mathbf{x}} = \mathbf{x}_n - \hat{J}^{-1} \mathbf{f}(\mathbf{x}_n) = \mathbf{x}_{n+i} - \hat{J}^{-1} \mathbf{f}(\mathbf{x}_{n+i}).$$

One can then replace $\hat{\mathbf{u}}$ with $\Delta \tilde{\mathbf{u}}$ for any $\Delta \in \mathbb{R}^{(k+1) \times k}$ such that its columns sum to zero:

$$F_{n,k} \Delta \tilde{\mathbf{u}} = \mathbf{f}(\mathbf{x}_{n+i}), \quad \hat{\mathbf{x}} = \mathbf{x}_{n+i} - X_{n,k} \Delta \tilde{\mathbf{u}}.$$

Our original Δ_n gives a centered finite difference stencil, but one can also use a path-based stencil:

$$\Delta_s = \begin{bmatrix} -1 & & & \\ 1 & \ddots & & \\ & \ddots & -1 & \\ & & & 1 \end{bmatrix}.$$

This means the system from equation (1) produces an identical approximation to the Jacobian as the system

$$\begin{bmatrix} \mathbf{f}(\mathbf{x}_{n+1}) - \mathbf{f}(\mathbf{x}_n) & \mathbf{f}(\mathbf{x}_{n+2}) - \mathbf{f}(\mathbf{x}_{n+1}) & \dots & \mathbf{f}(\mathbf{x}_{n+k}) - \mathbf{f}(\mathbf{x}_{n+k-1}) \end{bmatrix} = \hat{J} \left(\begin{bmatrix} \mathbf{x}_{n+1} - \mathbf{x}_n & \mathbf{x}_{n+2} - \mathbf{x}_{n+1} & \dots & \mathbf{x}_{n+k} - \mathbf{x}_{n+k-1} \end{bmatrix} \right).$$

Ultimately, all finite difference formulas that use the same function evaluations give the same \hat{J} . Thus, \hat{J} approximates $J(\mathbf{x})$ for all $\mathbf{x} \in \text{conv}(\mathbf{x}_n, \dots, \mathbf{x}_{n+k})$.

Since all three forms are equivalent for all choices of i and valid Δ and the multisection equations represent a specific choice of i and Δ , all three forms provide the same approximation to the root of $\mathbf{f}(\mathbf{x})$ as the multisection equations. Wherever the multisection equations are used one may replace them with any of the forms presented here.

The multisection equations are an example of a quasi-Newton method. A quasi-Newton method is any method of the form

$$\hat{\mathbf{x}}_{n+1} = \mathbf{x}_n - \mathbf{u}_n, \quad (8)$$

where \mathbf{u}_n is an approximate solution to the equation

$$J(\mathbf{x}_n)\mathbf{u} = \mathbf{f}(\mathbf{x}_n), \quad (9)$$

where $J(\mathbf{x})$ is the Jacobian of $\mathbf{f}(\mathbf{x})$ evaluated at \mathbf{x}_n . In particular, one can use the multisection equations in any of their forms to provide such an approximation, namely equations (4), (5), (6), and (7).

Consider, for example, equation (4) with $k = d$. The solution $\tilde{\mathbf{u}}_i$ may be found elementwise by Cramer's rule:

$$\begin{aligned} \tilde{\mathbf{u}}_i &= \frac{\begin{vmatrix} \dots & \mathbf{f}(\mathbf{x}_{n+i-1}) - \mathbf{f}(\mathbf{x}_n) & \mathbf{f}(\mathbf{x}_n) & \mathbf{f}(\mathbf{x}_{n+i+1}) - \mathbf{f}(\mathbf{x}_n) & \dots \end{vmatrix}}{\begin{vmatrix} \mathbf{f}(\mathbf{x}_{n+1}) - \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{f}(\mathbf{x}_{n+d}) - \mathbf{f}(\mathbf{x}_n) \end{vmatrix}} \\ &= (-1)^i \frac{\begin{vmatrix} \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{f}(\mathbf{x}_{n+i-1}) & \mathbf{f}(\mathbf{x}_{n+i+1}) & \mathbf{f}(\mathbf{x}_{n+d}) \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{f}(\mathbf{x}_{n+d}) \end{vmatrix}}. \end{aligned}$$

The quasi-Newton method defined above may then be expressed as

$$\begin{aligned} \hat{\mathbf{x}}_{n+1} &= \mathbf{x}_n - \frac{\begin{vmatrix} 0 & \mathbf{x}_{n+1} - \mathbf{x}_n & \dots & \mathbf{x}_{n+d} - \mathbf{x}_n \\ \mathbf{f}(\mathbf{x}_n) & \mathbf{f}(\mathbf{x}_{n+1}) & \dots & \mathbf{f}(\mathbf{x}_{n+d}) \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{f}(\mathbf{x}_{n+d}) \end{vmatrix}} \\ &= \frac{\begin{vmatrix} \mathbf{x}_n & \dots & \mathbf{x}_{n+d} \\ \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{f}(\mathbf{x}_{n+d}) \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{f}(\mathbf{x}_{n+d}) \end{vmatrix}}, \end{aligned}$$

where one must expand the determinant along the top row, so that

$$\hat{\mathbf{x}}_{n+1} \cdot \mathbf{e}_i = \frac{\begin{vmatrix} \mathbf{x}_n \cdot \mathbf{e}_i & \dots & \mathbf{x}_{n+d} \cdot \mathbf{e}_i \\ \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{f}(\mathbf{x}_{n+d}) \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{f}(\mathbf{x}_{n+d}) \end{vmatrix}}. \quad (10)$$

This is a generalized notion of the determinant used in the theory of extrapolation methods [11].

Suppose that we do not have enough values of $\mathbf{f}(\mathbf{x}_{n+i})$ to fully determine \hat{J} , ie. $k < d$. We apply the solution found above to the underdetermined Newtonian form of the equations, equation (5). The quasi-Newton method that results from this may be written as

$$\hat{\mathbf{x}}_{n+1} = \frac{\begin{vmatrix} \mathbf{x}_n & \dots & \mathbf{x}_{n+k} \\ \mathbf{v}_1^\top \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{v}_1^\top \mathbf{f}(\mathbf{x}_{n+k}) \\ \vdots & & \vdots \\ \mathbf{v}_k^\top \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{v}_k^\top \mathbf{f}(\mathbf{x}_{n+k}) \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \mathbf{v}_1^\top \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{v}_1^\top \mathbf{f}(\mathbf{x}_{n+k}) \\ \vdots & & \vdots \\ \mathbf{v}_k^\top \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{v}_k^\top \mathbf{f}(\mathbf{x}_{n+k}) \end{vmatrix}}, \quad (11)$$

where \mathbf{v}_i is the i -th column of B .

Cramer's rule is not a practical method for finding these solutions. It is used here solely for the purposes of extracting these forms of the solutions.

3. Connection to root-finding methods

As mentioned in the previous section, as an alternative to adding constraints to equation (1) one can add vectors to the matrix $F_{n,k}$. For example, the update for Broyden's method [12] is chosen such that

$$\hat{J}_{n+1} [X_{n,1} \Delta \quad Q] = [F_{n,1} \Delta \quad \hat{J}_n Q],$$

where $Q^\top X_{n,1} \Delta = 0$. The columns of Q are additional search directions, and the product $\hat{J}_n Q$ an approximation to function evaluations in these search directions. In the generalized Broyden's method [13, 14] $X_{n,1}$ and $F_{n,1}$ are replaced by $X_{n,k}$ and $F_{n,k}$ and Q reduced in size by k columns. Since the search directions in Q are arbitrary, removing them will not affect the convergence.

Broyden's family of methods [14, 8] may be written as

$$\hat{J}_{n+1} = \hat{J}_n + \mathbf{f}(\mathbf{x}_{n+1}) \mathbf{v}_n^\top,$$

where $\mathbf{v}_n^\top (\mathbf{x}_{n+1} - \mathbf{x}_n) = 1$. If \mathbf{v}_n is chosen such that

$$\mathbf{v}_n^\top X_{n-k,k} \Delta = [0 \quad \dots \quad 0 \quad 1],$$

with possibly other constraints, then $\hat{J}_{n+1}X_{n-k,k}\Delta = F_{n-k,k}\Delta$, which is functionally equivalent to equation (3).

Anderson mixing [15, 14, 4] solves $F_{n,k}\Delta\mathbf{u} = \mathbf{f}(\mathbf{x}_{n+k})$, see equation (4), in a least-squares sense, then uses the step

$$\hat{\mathbf{x}}_{n+k+1} = \mathbf{x}_{n+k} - X_{n,k}\Delta\mathbf{u} + \beta(\mathbf{f}(\mathbf{x}_{n+k}) - F_{n,k}\Delta\mathbf{u}).$$

For $\beta = 0$ this is the multisection equations, changing Δ_n and the index in \mathbf{x}_n . As has been shown in Section 2, we are free to choose these features. Anderson mixing is then the multisection equations with relaxation.

4. Connection to extrapolation methods

Extrapolation methods seek to accelerate the convergence of sequences. In general, these sequences are nonlinear and can lie within a vector space. If one has $k+1$ iterates of the sequence, $\{\mathbf{x}_n, \dots, \mathbf{x}_{n+k}\}$, then the next element in the accelerated sequence is

$$\hat{\mathbf{x}}_{n+1} = \sum_{i=0}^k u_i \mathbf{x}_{n+i} = X_{n,k}\mathbf{u},$$

where $\mathbf{1}^\top \mathbf{u} = 1$. Let \mathbf{x} be the limit of the sequence \mathbf{x}_n . If \mathbf{x} lies within the span of the iterates then we seek \mathbf{u} that satisfies

$$\hat{\mathbf{x}}_{n+1} = X_{n,k}\mathbf{u} = \mathbf{x} \iff (X_{n,k} - \mathbf{x}\mathbf{1}^\top)\mathbf{u} = \sum_{i=0}^k u_i (\mathbf{x}_{n+i} - \mathbf{x}) = 0.$$

If $X_{n+1,k}\mathbf{u} = \mathbf{x}$ as well, such as in the limit as n approaches infinity or when the columns of $X_{n+1,k}$ and $X_{n,k}$ span the same space, then

$$(X_{n+1,k} - X_{n,k})\mathbf{u} = \sum_{i=0}^k u_i (\mathbf{x}_{n+i+1} - \mathbf{x}_{n+i}) = 0.$$

The iterate $\hat{\mathbf{x}}_{n+1}$ may be expressed as

$$\hat{\mathbf{x}}_{n+1} = \frac{\begin{vmatrix} \mathbf{x}_n & \dots & \mathbf{x}_{n+k} \\ \mathbf{v}_1^\top \mathbf{r}(\mathbf{x}_n) & \dots & \mathbf{v}_1^\top \mathbf{r}(\mathbf{x}_{n+k}) \\ \vdots & & \vdots \\ \mathbf{v}_k^\top \mathbf{r}(\mathbf{x}_n) & \dots & \mathbf{v}_k^\top \mathbf{r}(\mathbf{x}_{n+k}) \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \mathbf{v}_1^\top \mathbf{r}(\mathbf{x}_n) & \dots & \mathbf{v}_1^\top \mathbf{r}(\mathbf{x}_{n+k}) \\ \vdots & & \vdots \\ \mathbf{v}_k^\top \mathbf{r}(\mathbf{x}_n) & \dots & \mathbf{v}_k^\top \mathbf{r}(\mathbf{x}_{n+k}) \end{vmatrix}},$$

using the same generalized notion of the determinant from equation (10), where

$$\mathbf{r}(\mathbf{x}_{n+i}) = \mathbf{x}_{n+i+1} - \mathbf{x}_{n+i} \tag{12}$$

and $\{\mathbf{v}_i\}$ is a linearly independent set of vectors. See [11] for a complete introduction of extrapolation methods. Note this is exactly equation (11) replacing $\mathbf{f}(\mathbf{x})$ with $\mathbf{r}(\mathbf{x})$. Thus, extrapolation methods of this form are identical to the multiseant equations acting on $\mathbf{f}(\mathbf{x}) = \mathbf{r}(\mathbf{x})$ using $\{\mathbf{x}_{n+i}\}$ as search directions. They can therefore be expressed in the form of equation (7).

Methods of this form are called polynomial extrapolation methods [7, 3]. Several extrapolation methods fall in this category:

- minimum polynomial extrapolation (MPE) [16], with $\mathbf{v}_i = \mathbf{r}(\mathbf{x}_{n+i-1})$;
- modified minimum polynomial extrapolation (MMPE) [7], with \mathbf{v}_i some fixed vector;
- reduced rank extrapolation (RRE) [17, 18, 19], with $\mathbf{v}_i = \mathbf{r}(\mathbf{x}_{n+i}) - \mathbf{r}(\mathbf{x}_{n+i-1})$.

There is another category of methods known as ϵ -algorithms [20, 21, 22, 3] that may be connected to the multiseant equations as well, though not in the same manner. Recall equation (1). One can take several such equations and arrive at several approximations of the Jacobian, see equation (3):

$$F_{n+j,k}\Delta_n = \hat{J}_{n+j}X_{n+j,k}\Delta_n \quad \forall j.$$

As before in equation (4), we seek to solve

$$F_{n+j,k}\Delta\tilde{\mathbf{u}}_j = \mathbf{f}(\mathbf{x}_{n+j}), \quad \hat{\mathbf{x}}_{n+1} = \mathbf{x}_{n+j} - X_{n+j,k}\Delta\tilde{\mathbf{u}}_j \quad \forall j.$$

This gives several estimates of $\hat{\mathbf{x}}_{n+1}$. Each of these systems can be reduced to a single equation by taking the inner product with a given vector \mathbf{v} . There are then as many equations as there are values of j . If one assumes all $\tilde{\mathbf{u}}_j$ are equal, essentially that each individual system captures the solution, then one can summarize these equations in the system

$$\begin{bmatrix} \mathbf{v}^\top F_{n,k} \\ \vdots \\ \mathbf{v}^\top F_{n+k-1,k} \end{bmatrix} \Delta\tilde{\mathbf{u}} = \begin{bmatrix} \mathbf{v}^\top \mathbf{f}(\mathbf{x}_n) \\ \vdots \\ \mathbf{v}^\top \mathbf{f}(\mathbf{x}_{n+k-1}) \end{bmatrix}, \quad \hat{\mathbf{x}}_{n+1} = \mathbf{x}_n - X_{n,k}\Delta\tilde{\mathbf{u}}.$$

Following the same work as in Section 2 one arrives at the solution

$$\hat{\mathbf{x}}_{n+1} = \frac{\begin{vmatrix} \mathbf{x}_n & \dots & \mathbf{x}_{n+k} \\ \mathbf{v}^\top \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{v}^\top \mathbf{f}(\mathbf{x}_{n+k}) \\ \vdots & & \vdots \\ \mathbf{v}^\top \mathbf{f}(\mathbf{x}_{n+k-1}) & \dots & \mathbf{v}^\top \mathbf{f}(\mathbf{x}_{n+2k-1}) \end{vmatrix}}{\begin{vmatrix} 1 & \dots & 1 \\ \mathbf{v}^\top \mathbf{f}(\mathbf{x}_n) & \dots & \mathbf{v}^\top \mathbf{f}(\mathbf{x}_{n+k}) \\ \vdots & & \vdots \\ \mathbf{v}^\top \mathbf{f}(\mathbf{x}_{n+k-1}) & \dots & \mathbf{v}^\top \mathbf{f}(\mathbf{x}_{n+2k-1}) \end{vmatrix}}.$$

Replacing $\mathbf{f}(\mathbf{x})$ with $\mathbf{r}(\mathbf{x})$ from equation (12) gives the topological ϵ -algorithm (TEA) [23].

5. Connection to Krylov methods

Suppose the extrapolation methods of the previous section are applied to the linear vector sequence $\mathbf{x}_{n+1} = (A + I)\mathbf{x}_n - \mathbf{b}$. The limit \mathbf{x} of this sequence is then the solution to $A\mathbf{x} = \mathbf{b}$. The functions $\mathbf{r}(\mathbf{x}_{n+i})$ are then

$$\mathbf{r}(\mathbf{x}_{n+i}) = (A + I)\mathbf{x}_{n+i} - \mathbf{b} - \mathbf{x}_{n+i} = A\mathbf{x}_{n+i} - \mathbf{b},$$

and they satisfy

$$\mathbf{r}(\mathbf{x}_{n+i}) = (A + I)\mathbf{x}_{n+i} - \mathbf{b} - (A + I)\mathbf{x}_{n+i-1} + \mathbf{b} = (A + I)\mathbf{r}(\mathbf{x}_{n+i-1}).$$

The vectors $\mathbf{r}(\mathbf{x}_{n+i})$ then form a Krylov subspace, such that $\mathbf{r}(\mathbf{x}_{n+i}) \in \mathcal{K}_i(A + I, \mathbf{r}(\mathbf{x}_n))$.

Under these conditions the extrapolation methods become Krylov subspace methods. Most notably, since MPE uses the Arnoldi iteration to produce orthogonal search directions, it is identical to GMRES when applied to this linear sequence [4, 1, 3]. These algorithms are presented as Algorithms 1 and 2. The connections to the linear case are noted in the latter. Since $\mathcal{K}_k(A + I, \mathbf{v}) = \mathcal{K}_k(A, \mathbf{v})$ the search directions for both algorithms are identical.

Algorithm 1 GMRES

```

 $\mathbf{q}_1 = \mathbf{b} / \|\mathbf{b}\|$ 
for  $k = 1$  to  $n$  do
   $\mathbf{y} = A\mathbf{q}_k$  ( $\in \mathcal{K}_k(A, \mathbf{b})$ )
  orthogonalize  $\mathbf{y}$  with respect to  $\mathcal{K}_{k-1}(A, \mathbf{b})$ 
   $\mathbf{q}_{k+1} = \mathbf{y} / \|\mathbf{y}\|$ 
end for
minimize  $\|H_n \mathbf{u} - \|\mathbf{b}\| \mathbf{e}_1\|$ 
 $\hat{\mathbf{x}}_{n+1} = Q_n \mathbf{u} + \mathbf{x}_0$ 

```

Algorithm 2 MPE

```

 $\mathbf{q}_1 = \mathbf{r}(\mathbf{x}_{n+1}) / \|\mathbf{r}(\mathbf{x}_{n+1})\|$  ( $\mathbf{b} / \|\mathbf{b}\|$  in the linear case)
for  $k = 1$  to  $n$  do
   $\mathbf{y} = \mathbf{r}(\mathbf{x}_{n+k})$  ( $\in \mathcal{K}_k(A + I, \mathbf{b})$  in the linear case)
  orthogonalize  $\mathbf{y}$  with respect to  $\{\mathbf{r}(\mathbf{x}_n), \dots, \mathbf{r}(\mathbf{x}_{n+k-1})\}$  ( $\mathcal{K}_{k-1}(A + I, \mathbf{b})$ 
in the linear case)
   $\mathbf{q}_{k+1} = \mathbf{y} / \|\mathbf{y}\|$ 
end for
minimize  $\|H_n \mathbf{u}\|$  such that  $\mathbf{1}^\top \mathbf{u} = 1$ 
 $\hat{\mathbf{x}}_{n+1} = [\mathbf{x}_0 \ \dots \ \mathbf{x}_n] \mathbf{u}$ 

```

The minimization steps of the two algorithms can be shown to be equivalent as well. Consider the solution found using GMRES: $\hat{\mathbf{x}} = Q_k \mathbf{y}_k$ where Q_k is

Extrapolation	$\mathbf{q}_{k+1} \perp$	Krylov
MPE	$\mathcal{K}_k(A + I, \mathbf{f}(\mathbf{x}_n))$	GMRES
RRE	$\mathcal{K}_k(A, \mathbf{f}(\mathbf{x}_n))$	GMRES
MMPE	$\mathcal{K}_k(G, \mathbf{q}_0)$	n/a
TEA	$\mathcal{K}_k(A^\top, \mathbf{q})$	BiCG

Table 1: Connections between extrapolation methods and Krylov methods. The extrapolation methods are applied to the sequence $\mathbf{x}_{n+1} = (A + I)\mathbf{x}_n - \mathbf{b}$.

derived from the Arnoldi iteration on $\mathcal{K}_{k-1}(A + I, \mathbf{b})$. Then we seek

$$\begin{aligned} \min \|A\hat{\mathbf{x}} - \mathbf{b}\| &= \min \|AQ_k \mathbf{y}_k - \mathbf{b}\| \\ &= \min \|F_{n,k} \Delta \tilde{\mathbf{u}}_k - \mathbf{b}\|, \end{aligned}$$

since the column space of $F_{n,k} \Delta$ is equal to the Krylov subspace $\mathcal{K}_{k-1}(A, \mathbf{b})$. Recall in the linear case that $\mathbf{f}(\mathbf{x}_n) = \mathbf{b}$, and so this minimization is equivalent to solving equation (5) with $B = F_{n,k}$. We've shown in Section 2 that this is equivalent to minimizing $\|F_{n,k} \mathbf{u}_k\|$ under the constraint $\mathbf{1}^\top \mathbf{u}_k = 1$. This is exactly the minimization step in MPE.

For the linear case of TEA, the term $\mathbf{v}^\top \mathbf{f}(\mathbf{x}_{n+i})$ may now be written as $\mathbf{v}^\top (A + I)^{i-j} \mathbf{f}(\mathbf{x}_{n+j})$. This means TEA can now be expressed in the form of equation (7) with $\mathbf{v}_i = (A^\top + I)^i \mathbf{v}$. TEA then uses the Lanczos biorthogonalization process, making it equivalent in the linear case to the biconjugate gradient method (BiCG).

Table 1 gives the orthogonalization conditions and corresponding Krylov subspace methods for these four extrapolation methods when they are applied to linear vector sequences.

6. Connections to other methods

The multisection equations form the basis of many other methods besides those mentioned here. In particular, Jacobian-free Newton-Krylov methods resolve a Newton step, equation (2) or its exact version, using a Krylov subspace method. See [24] for an introduction and survey of such methods. As these methods eschew the construction of the Jacobian by using finite difference approximations they fall precisely into this framework, suggesting they solve the multisection equations.

Related to this are nonlinear Krylov methods, see for example [25, 26]. These are similar to the above framework and Newton-Krylov methods but may differ by the use of preconditioning on the nonlinear function.

This framework can be considered to cover all methods that share the following aspects:

- linearization of a nonlinear function;
- approximation of a Newton step, generally by finite difference;

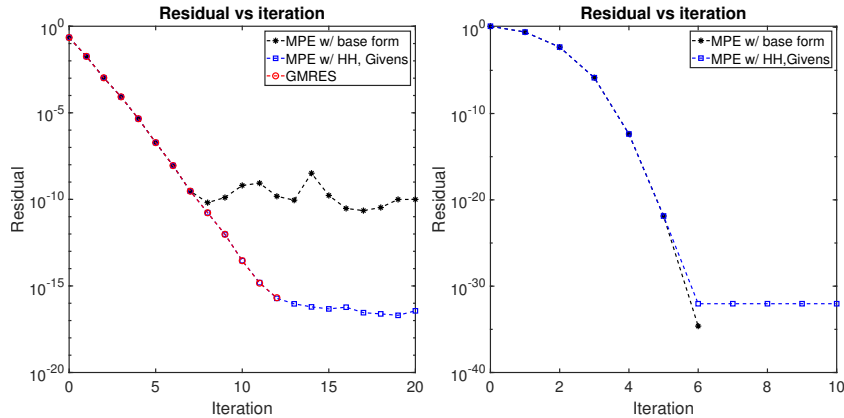


Figure 2: Numerical examples comparing the multisecant equations (MPE in its base form, see equation (7)), MPE and GMRES. (Left) A linear example, see equation (13). (Right) A nonlinear example, see equation (14).

- resolution of the inexact Newton step with a Krylov subspace method, and;
- restart with an updated set of function evaluations, at least as often as the Krylov subspace method finds a complete vector space.

7. Numerical examples

We apply MPE, GMRES and the multisecant equations to numerical vector sequences to compare their behaviour. We first consider a linear example:

$$\mathbf{x}_{n+1} = A\mathbf{x}_n + \mathbf{b}, \quad A \in \mathbb{R}^{20 \times 20}, \quad (13)$$

where A and \mathbf{b} are determined randomly. The norms of A and \mathbf{b} are set to 0.5 so that the sequence \mathbf{x}_n has a limit as opposed to an anti-limit.

The left of Figure 2 shows how each method tackles the residual of the sequence. The GMRES algorithm used is the built-in MATLAB version, while the MPE algorithm makes use of Householder reflections and Givens rotations and the multisecant equations use the backslash operator in MATLAB to solve equation (7). We see all three methods produce identical residuals for the first 7 iterations. After this, round-off error in the calculations prevents further improvements in accuracy for the multisecant equations. GMRES stops iterating once the residual is below machine epsilon.

We next consider a nonlinear example, also in \mathbb{R}^{20} :

$$\mathbf{x}_{n+1} = \sin(A\mathbf{x}_n^2), \quad (14)$$

where the square operation is performed elementwise and A is determined randomly. Convergence of the sequence is fast. The right of Figure 2 shows the

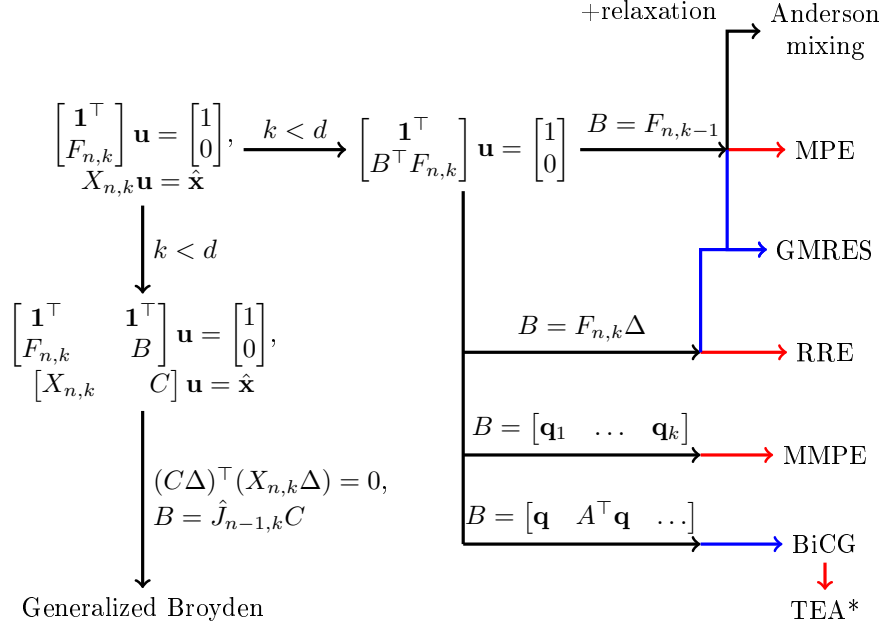


Figure 3: Interconnectivity of extrapolation, acceleration and quasi-Newton methods. Red arrows indicate $\mathbf{f}(\mathbf{x}_n) = \mathbf{x}_{n+1} - \mathbf{x}_n$ while blue arrows indicate $\mathbf{f}(\mathbf{x}_n) = A\mathbf{x}_n - \mathbf{b}$ and $\mathbf{f}(\mathbf{x}_{n+1}) = (A+I)\mathbf{f}(\mathbf{x}_n)$. Note that TEA* is the linear version of the method; general TEA derives directly from the multiseant equations.

residual under the multiseant equations and MPE. GMRES cannot be used here as the problem is nonlinear. Again, MPE and the multiseant equations agree, up to numerical error.

8. Conclusions

Figure 3 presents a summary of the interconnectivity discussed in this paper. The multiseant equations form the basis of numerical methods for root-finding in higher dimensions, see left of Figure 3. They themselves are extensions of the secant method in 1D.

When underdetermined they can be connected to extrapolation methods, see blue arrows on right of Figure 3. Which particular extrapolation method depends on the additional constraints imposed on the underdetermined systems.

If these methods are applied to linear problems with vector sequences that lie within Krylov subspaces, then they become Krylov subspace methods, see red arrows on right of Figure 3. This is due to the orthogonalization steps in these methods.

The three methods are then intrinsically linked. As a direct consequence, extrapolation methods can be thought of as nonlinear Krylov subspace meth-

ods, as they use the same orthogonalization processes but applied to nonlinear problems. Moreover, this indicates the advanced techniques employed in Krylov subspace methods can be used to improve extrapolation methods. It also suggests that any Krylov subspace method has an extrapolation method counterpart, and vice versa.

Acknowledgements

This research was funded by the Swiss National Science Foundation (SNSF).

References

- [1] A. Sidi, Extrapolation vs. projection methods for linear systems of equations, *Journal of Computational and Applied Mathematics* 22 (1988) 71–88. doi:10.1016/0377-0427(88)90289-0.
- [2] A. Sidi, Efficient implementation of minimal polynomial and reduced rank extrapolation methods, *Journal of Computational and Applied Mathematics* 36 (1991) 305.
- [3] K. Jbilou, H. Sadok, Vector extrapolation methods. Applications and numerical comparison, *Journal of Computational and Applied Mathematics* 122 (2000) 149–165. doi:10.1016/S0377-0427(00)00357-5.
- [4] H. F. Walker, P. Ni, Anderson acceleration for fixed-point iterations, *SIAM Journal on Numerical Analysis* 49 (2011) 1715–1735. doi:10.1137/10078356X.
URL <http://www.siam.org/journals/sinum/49-4/78356.html>
- [5] W. B. Gragg, G. W. Stewart, A stable variant of the secant method for solving nonlinear equations, *SIAM Journal on Numerical Analysis* 13 (1976) 889–903.
- [6] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific and Statistical Computing* 7 (7 1986).
- [7] A. Sidi, W. F. Ford, D. A. Smith, Acceleration of convergence of vector sequences, *SIAM Journal on Numerical Analysis* 23 (1986) 178–196. doi:10.1137/0723013.
- [8] H. R. Fang, Y. Saad, Two classes of multisection methods for nonlinear acceleration, *Numerical Linear Algebra with Applications* 16 (2009) 197–221. doi:10.1002/nla.617.
- [9] J. G. P. Barnes, An algorithm for solving non-linear equations based on the secant method, *The Computer Journal* 8 (1965) 66–72.
URL <https://academic.oup.com/comjnl/article/8/1/66/489886>

- [10] P. Wolfe, The secant method for simultaneous nonlinear equations, *Communications of the ACM* 2 (1959) 12–13.
- [11] C. Brezinski, M. Zaglia, *Extrapolation Methods: Theory and Practice*, ISSN, Elsevier Science, 2013.
- [12] C. G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation* 19 (1965) 577–593.
- [13] D. Vanderbilt, S. G. Louie, Total energies of diamond (111) surface reconstructions by a linear combination of atomic orbitals method, *Physical Review B* 30 (1984) 6118–6130.
- [14] V. Eyert, A comparative study on methods for convergence acceleration of iterative vector sequences, *Journal of Computational Physics* 124 (1996) 271–285.
- [15] D. G. Anderson, Iterative procedures for nonlinear integral equations, *Journal of the Association for Computing Machinery* 12 (1965) 547–560.
- [16] S. Cabay, L. W. Jackson, A polynomial extrapolation method for finding limits and antilimits of vector sequences, *SIAM Journal on Numerical Analysis* 13 (1976) 734–752.
- [17] R. P. Eddy, Extrapolating to the limit of a vector sequence, *Information Linkage Between Applied Mathematics and Industry* (1979) 387–396 [doi:10.1016/B978-0-12-734250-4.50028-X](https://doi.org/10.1016/B978-0-12-734250-4.50028-X).
- [18] M. Mešina, Convergence acceleration for the iterative solution of the equations $x = ax + f$, *Computer Methods in Applied Mechanics and Engineering* 10 (1977) 165–173. [doi:10.1016/0045-7825\(77\)90004-4](https://doi.org/10.1016/0045-7825(77)90004-4).
- [19] D. A. Smith, W. F. Ford, A. Sidi, Extrapolation methods for vector sequences, *SIAM Review* 29 (1987) 199–233.
- [20] P. Wynn, On a device for computing the $e_m(S_n)$ transformation, *Mathematical Tables and Other Aids to Computation* 10 (1956) 91–96.
- [21] E. Gekeler, On the solution of systems of equations by the epsilon algorithm of Wynn, *Mathematics of Computation* 26 (1972).
- [22] C. Brezinski, Numerical stability of a quadratic method for solving systems of non linear equations, *Computing* 14 (1975) 205–211.
- [23] C. Brezinski, Accélération de la convergence en analyse numérique (1977). [doi:10.1007/BFB0089363](https://doi.org/10.1007/BFB0089363).
URL <http://link.springer.com/10.1007/BFB0089363>
- [24] D. A. Knoll, D. E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, *Journal of Computational Physics* 193 (2) (2004) 357–397.

- [25] T. Washio, C. W. Oosterlee, Krylov subspace acceleration for nonlinear multigrid schemes, *Electron. Trans. Numer. Anal* 6 (271-290) (1997) 3.
- [26] H. D. Sterck, A nonlinear GMRES optimization algorithm for canonical tensor decomposition, *SIAM Journal on Scientific Computing* 34 (3) (2012) A1351–A1379. [arXiv:https://doi.org/10.1137/110835530](https://arxiv.org/abs/https://doi.org/10.1137/110835530), [doi:10.1137/110835530](https://doi.org/10.1137/110835530).
URL <https://doi.org/10.1137/110835530>