11-19-2004

Squared Extrapolation Methods (SQUAREM): A New Class of Simple and Efficient Numerical Schemes for Accelerating the Convergence of the EM Algorithm

Ravi Varadhan
The Center of Aging and Health, Johns Hopkins University, rvaradhan@jhmi.edu

Ch. Roland
Laboratoire Paul Painleve, UFR Mathematiques Pures et Appliquees-M3, Universite des Sciences et Technologies de Lille

Squared Extrapolation Methods (SQUAREM): A New Class of Simple and Efficient Numerical Schemes for Accelerating the Convergence of the EM Algorithm

R. Varadhan * and Ch. Roland †

November 19, 2004

1 Abstract

We derive a new class of iterative schemes for accelerating the convergence of the EM algorithm, by exploiting the connection between fixed point iterations and extrapolation methods. First, we present a general formulation of one-step iterative schemes, which are obtained by cycling with the extrapolation methods. We, then *square* the one-step schemes to obtain the new class of methods, which we call SQUAREM. Squaring a one-step iterative scheme is simply applying it twice within each cycle of the extrapolation method. Here we focus on the first order or rank-one extrapolation methods for two reasons, (1) simplicity, and (2) computational efficiency. In particular, we study two first order extrapolation methods, the reduced rank extrapolation (RRE1) and minimal polynomial extrapolation (MPE1). The convergence of the new schemes, both one-step and squared, is non-monotonic with respect to the residual norm. The first order one-step and SQUAREM schemes

*The Center on Aging and Health, Johns Hopkins University, 2024 E. Monument Street, Suite 2-700, Baltimore, Maryland 21205, USA (rvaradhan@jhmi.edu)

†Laboratoire Paul Painlevé, UFR de Mathématiques Pures et Appliquées-M3, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq cedex, France (christophe.roland@math.univ-lille1.fr).

are linearly convergent, like the EM algorithm but they have a faster rate of convergence. We demonstrate, through five different examples, the effectiveness of the first order SQUAREM schemes, SqRRE1 and SqMPE1, in accelerating the EM algorithm. The SQUAREM schemes are also shown to be vastly superior to their one-step counterparts, RRE1 and MPE1, in terms of computational efficiency. The proposed extrapolation schemes can fail due to the numerical problems of stagnation and near breakdown. We have developed a new hybrid iterative scheme that combines the RRE1 and MPE1 schemes in such a manner that it overcomes both stagnation and near breakdown. The squared first order hybrid scheme, SqHyb1, emerges as the iterative scheme of choice based on our numerical experiments. It combines the fast convergence of the SqMPE1, while avoiding near breakdowns, with the stability of SqRRE1, while avoiding stagnations. The SQUAREM methods can be incorporated very easily into an existing EM algorithm. They only require the basic EM step for their implementation and do not require any other auxiliary quantities such as the complete data log likelihood, and its gradient or hessian. They are an attractive option in problems with a very large number of parameters, and in problems where the statistical model is complex, the EM algorithm is slow and each EM step is computationally demanding.

2 Introduction

Consider the mapping, $F : \Omega \subset \mathbb{R}^p \mapsto \Omega$. We are interested in finding the fixed point, x^*, of this mapping (if it exists), where x^* satisfies the equation

$$x = F(x). \tag{1}$$

A obvious and natural way to find x^* is to start with some x_0, and form a sequence x_n defined by the Picard iteration

$$x_{n+1} = F(x_n) \qquad (n = 0, 1, 2, ...). \tag{2}$$

If x_n converges to some point x^* and $F(x)$ is continuous, then $x^* = F(x^*)$. Thus x^* is a fixed point of the map F. Furthermore, in all the following discussions we assume that the function F is Lipschitz-continuous, with the Lipschitz constant smaller than 1, i.e.,

$$\forall x, y \in \Omega : \|F(x) - F(y)\| \leq L\|x - y\|,$$

where $L < 1$ is the Lipschitz constant. We also assume that the mapping F admits continuous, bounded partial derivatives. Under these assumptions, the Picard iteration scheme, Eq. 2, is linearly convergent, and its rate of convergence is very slow when the dominant eigenvalue of the Jacobian of F is close to 1.

We are interested in accelerating the convergence of the basic Picard iteration scheme. We exploit the strong connection between extrapolation methods and fixed point iterations for solving Eq. 1, using the idea of *cycling*. The most well known instance of this connection is that between Aitken's Δ^2 process and Steffensen's method in the scalar case, i.e. $p = 1$. We first cycle with the extrapolation methods to derive a broad class of one-step iterative schemes. Within each cycle, the fixed point iteration, Eq. 2, is applied a specified number of times to generate a sequence, which is then extrapolated to obtain a new vector that forms the starting value for the next cycle of the one-step iterative scheme. We, then, employ a relatively novel strategy called *squaring* to the first order one-step iterative schemes and obtain a new class of fast, linearly convergent schemes. We call these first order squared extrapolation methods, SQUAREM. In the SQUAREM methods, the update x_{n+1} is obtained in two steps. In the first step, we apply a one-step scheme to determine an intermediate vector z_n, to which we once again apply the same one-step scheme to obtain the desired update, x_{n+1}.

Numerical analysts have used one-step iterative schemes, based on extrapolation methods, for solving linear and nonlinear fixed point problems [42]. However, the strategy of squaring, as used in the SQUAREM schemes, is relatively new. It was first employed by Raydan and Svaiter [33] to accelerate the convergence of the classical Cauchy (or steepest descent) method for solving the quadratic optimization problem, which is equivalent to solving a linear system of equations. Roland and Varadhan [35] extended the squaring technique to solve the general nonlinear fixed point problem. They proposed a squared version of the Lemarechal iterative scheme, which is a multivariate extension of the Steffensen's method. Here we propose a broader class of iterative schemes based on the idea of cycling with the extrapolation methods. We highlight two members of this class, the minimal polynomial extrapolation method (MPE) and the reduced rank extrapolation method (RRE). In this paper, we only focus on the first order MPE and RRE schemes, MPE1 and RRE1. Lemarechal's method and RRE1 are the same. MPE1 and RRE1 are one-step methods in that the parameter update x_{n+1} is obtained from x_n in a single step. We, then, obtain iterative schemes, SqMPE1 and SqRRE1,

by squaring MPE1 and RRE1. SqMPE1 is a new iterative scheme, whereas SqRRE1 is the same as the *adjusted* Δ^1 method developed in [35]. We also develop another new SQUAREM method in this paper, called SqHyb1, which is a hybrid of SqMPE1 and SqRRE1 schemes. The SqHyb1 scheme retains the advantages of both SqMPE1 and SqRRE1, while eliminating their weaknesses, and thus, promises to be a useful technique.

Our main goal is to apply these schemes, one-step and SQUAREM, to accelerate the convergence of the EM algorithm, which is essentially a fixed point iterative scheme of the form of Eq. 2, for solving the likelihood maximization problem. The new iterative schemes can also be employed to accelerate the linear convergence of variants of the basic EM algorithm such as the generalized EM (GEM) [24], and the expectation-conditional maximization (ECM) [28] algorithms. These variants are useful when the M-step is either analytically intractable or can't be solved using readily available computer packages, and they all share the important property, along with the original EM, of stable monotone convergence.

In statistical modeling it is common to have situations where there is "incomplete" information of two kinds: (a) direct missingness involving the lack of information on measured or measurable quantities, and (b) latency, where some of the variables in the statistical models are by construction unobservable or "latent", so information on them will obviously be missing, but a hypothetical scenario can be conceived where the missing information would be available. These situations occur in diverse applications such as evaluation of programs and policies, social behavior, public health, epidemiology, and medicine. The EM algorithm is by far the most popular approach for solving these "incomplete data" problems typically containing a large number of parameters. The key notion in the use of EM is that while the maximization of the likelihood for the observed (or incomplete) data is difficult, augmenting the observed data with the missing information typically yields a complete data log-likelihood that is easily maximized. Even though there are powerful numerical schemes available for the maximization of the observed data log-likelihood, their successful implementation is very difficult in many complex problems. Often, EM is the only practically feasible approach to solving the maximum likelihood estimation problem. In most, if not all, of these applications, the EM is quite slow to converge to the maximum likelihood estimates. Therefore, it is of great practical interest to develop broadly applicable iterative schemes to speed up the convergence of the EM.

Various numerical techniques have been proposed for accelerating the

EM algorithm. They include quasi-Newton methods ([25], [18]) and conjugate gradient methods [17]. Although these schemes generally improve the convergence of EM, it is important to recognize that their use involves significant trade-offs. The EM algorithm (a) is simple to devise for most missing data problems, (b) is globally convergent, (c) exhibits monotone increase of the likelihood, and (d) satisfies parameter constraints naturally. The acceleration schemes, on the other hand, tend to lack one or more of these desirable attributes. For example, a quasi-Newton method such as the Broyden's method is superlinearly convergent, and hence is fast. However, in its simplest form of implementation, the scheme is not globally convergent and therefore superlinearity can only be realized for "good" starting values. It is also non-monotone in the likelihood function value. Therefore, line search techniques are implemented to ensure global convergence. Furthermore, it involves the storage and handling of approximations to the Hessian matrix, which in large problems with thousands of parameters can be prohibitively expensive. Then there is the necessity to monitor the negative definiteness of the Hessian, and to devise a strategy to handle situations when it is not negative definite. Conjugate gradient methods for accelerating the EM do not involve matrix manipulations. However, they involve computations of the complete data log-likelihood function and its gradient. They also require a line-search to ensure global convergence. Furthermore, convergence is only linear.

As noted in [18], there are two types of costs associated with any computational scheme: thinking costs associated with developing and implementing the algorithm, and the costs associated with the use of computer resources and the time to produce the results. We call the first type of costs, analyst costs, and the second, machine costs. In many scientific research problems, it is typically the analyst costs that are limiting. In complex statistical models, it is expensive to evaluate each EM step, and the gradient and hessian of complete data log likelihood. Consequently, the ability to develop good models is severely hampered because of excessive computational times required to run simulations and other tasks which require repeated model evaluations. Some technological applications are based on statistical models, such as image reconstruction in PET scans. In such problems, the models and the EM steps may be quite simple, but they involve the estimation of tens of thousands to millions of parameters. Because the main focus is on the real time production of accurate and reliable results, the machine costs are critical.

The SQUAREM methods achieve a good balance between analyst and

machine costs because of the following important attributes: (a) they are simple and only require the basic EM step, (b) they do not require the computation of auxiliary quantities such as the incomplete or complete data log-likelihood functions or their gradients, (c) because of (a) and (b) they can be implemented easily and without disturbing existing EM routines, (d) they are as broadly applicable as the EM itself, (e) they do not involve any matrix storage and/or handling, (f) in vector computing environments such as R and MATLAB, they require negligible additional effort to that of the basic EM algorithm, and (g) they converge linearly, just like the EM, but at a faster rate, where the gains can be substantial, especially in problems where the EM is very slow due to a large fraction of missing information.

3 Background and Basic Results

3.1 EM Algorithm and Its Convergence

We have observations, y, which are assumed to be generated by a statistical model having the probability density function, $g(y;\theta)$, where $y = (y_1, \ldots, y_n)$ and $\theta \in \Omega \subset \mathbb{R}^p$. The EM algorithm is a flexible procedure that provides an iterative approach for computing the maximum likelihood estimates (MLE), typically in situations where such estimation would be easy, but for the lack of some additional information, z. So, $x = (y, z)$ is the complete data vector. Thus we have y, z, x, denoting the observed, missing, and complete data, respectively. Even when a problem is not overtly a missing-data problem, MLE computations can often be greatly facilitated by reformulating the problem as one such that the maximum likelihood estimates, given the missing information, can either be obtained analytically, or in some other straightforward manner. Let us describe specifically how the EM algorithm works. Let $g_c(x;\theta)$ be the density of the complete data vector. Let us denote by $L_c(\theta; x)$, the logarithm of g_c, with θ as the variable and x as being fixed. Formally, we have two sample spaces, $\mathbb{X}$ and $\mathbb{Y}$, and a many-to-one mapping from $\mathbb{X}$ to $\mathbb{Y}$. The observed data, y, are a realization from $\mathbb{Y}$ and we don't observe x directly, but it is only known to lie in $\mathbb{X}(y)$, which is the subset of $\mathbb{X}$ determined by the relation $y = y(x)$. We then have the following relation between the complete-data model and the observed data model:

$$g(y;\theta) = \int_{\mathbb{X}(y)} g_c(x;\theta)\, dx. \tag{3}$$

Note that given the observed model, $g(y;\theta)$, there are many ways to specify $g_c(x;\theta)$ that would satisfy Eq. 3. The EM algorithm computes the MLE by iteratively proceeding from an initial value (guess) for the parameters, $\theta_0 \in \mathbb{R}^p$. The k-th step of the iteration is given as

$$\theta_{k+1} = \operatorname{argmax} Q(\theta;\theta_k); \quad k = 0, 1, \ldots, \tag{4}$$

where

$$\begin{aligned} Q(\theta;\theta_k) &= E[L_c(\theta;x); y, \theta_k], \\ &= \int L_c(\theta;x) f(z; y, \theta_k) dz, \end{aligned} \tag{5}$$

where $f(z; y, \theta_k)$ denotes the density of the missing data, conditional on having observed y. The E-step of the EM algorithm, Eq. 5, is the computation of the Q-function, which is the conditional expectation of the log-likelihood of the complete data, given the observed data and $\theta_k \in \mathbb{R}^p$. The M-step, Eq. 4, entails the determination of the parameter, θ_{k+1}, that maximizes the Q-function, i.e. θ_{k+1} has the property that

$$Q(\theta_{k+1};\theta_k) \geq Q(\theta;\theta_k), \forall \theta \in \Omega.$$

A central property of the EM algorithm [8] is that its convergence is monotone in the likelihood of the observed data, i.e. $L(\theta_{k+1}) \geq L(\theta_k)$. Therefore, if $L(\theta)$ is bounded from above, the sequence $L(\theta_k)$ must converge monotonically to some L^*. A simple sufficient condition for L^* to be a stationary value of L is that $Q(\theta;\theta')$ be continuous in both θ and θ'. However, convergence of $L(\theta_k)$ to a stationary value L^* doesn't automatically imply the convergence of the EM iterates θ_k to a point θ^*. This usually requires more stringent conditions than the continuity of the Q-function. If there cannot exist two different stationary points with the same L value, then θ_k converges to a stationary point. Furthermore, if $L(\theta)$ is unimodal in Ω and has one and only one stationary point, and if the first derivative of the Q-function (where the derivative is with respect to its first argument θ) is continuous in both θ and θ', then θ_k converges to the unique maximizer θ^* of $L(\theta)$. The interested reader should consult [44] for a rigorous treatment of the convergence theorems for the EM algorithm.

The EM algorithm implicitly defines a mapping F from the parameter space onto itself, i.e. $F : \Omega \subset \mathbb{R}^p \mapsto \Omega$, such that

$$\theta_{k+1} = F(\theta_k), \quad k = 0, 1, \ldots. \tag{6}$$

Assuming that θ_k converges to the MLE θ^* and that F is Fréchet differentiable at θ^*, a Taylor series expansion yields

$$\begin{aligned} \theta_{k+1} - \theta^* &= F(\theta_k) - F(\theta^*) (\text{since } \theta^* \text{ is a fixed point of } F) \quad (7) \\ &= J(\theta^*)(\theta_k - \theta^*) + o(\|\theta_k - \theta^*\|^2), \quad (8) \end{aligned}$$

where

$$J_{ij}(\theta) = \frac{\partial F_i(\theta)}{\partial \theta_j},$$

is the Jacobian matrix of $F(\theta) = (F_1(\theta), \ldots, F_p(\theta)$. Thus, the EM is essentially a linear iteration in the neighborhood of the MLE θ^*, with the iteration matrix $J(\theta^*)$. It was shown in [8] that, for the EM (where $Q(\theta; \theta_k)$ is maximized with respect to θ), the Jacobian of the fixed-point mapping can be written as

$$J(\theta^*) = I_{miss}(\theta^*; y) I_{comp}^{-1}(\theta^*; y) \quad (9)$$

where

$$I_{comp}(\theta; y) = -\int \frac{\partial^2 \log g_c(x; \theta)}{\partial \theta^2} f(z; y, \theta) dz, \quad (10)$$

and

$$I_{miss}(\theta; y) = -\int \frac{\partial^2 \log f(z; y, \theta)}{\partial \theta^2} f(z; y, \theta) dz. \quad (11)$$

The Jacobian matrix $J(\theta^*)$ thus measures the fraction of missing information. Now, using the "missing information principle", which states that

$$I_{obs}(\theta^*; y) = I_{comp}(\theta^*; y) - I_{miss}(\theta^*; y), \quad (12)$$

we can rewrite Eq. 9 as

$$J(\theta^*) = I_p - I_{obs}(\theta^*; y) I_{comp}^{-1}(\theta^*; y), \quad (13)$$

where I_p is a $p \times p$ identity matrix, and $I_{obs}(\theta; y)$ is the observed information matrix given by

$$I_{obs}(\theta; y) = -\frac{\partial^2 \log g(y; \theta)}{\partial \theta^2}. \quad (14)$$

The rate of convergence of the EM algorithm is essentially governed by the spectral radius of the rate matrix, $\rho(J(\theta^*))$, which is the eigenvalue with the largest modulus, i.e.

$$\rho(J(\theta^*)) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } J(\theta^*)\}.$$

In general, since the Jacobian of the EM mapping is not a symmetric matrix, its eigenvalues can be complex. However, when $I_{obs}(\theta^*; y)$ is positive-definite, not only are the eigenvalues of $J(\theta^*)$ real, but the spectral radius $\rho(J(\theta^*))$ is less than 1, which guarantees the convergence of the EM. Actually the positive definiteness of $I_{obs}(\theta^*; y)$ is also a sufficient condition for θ^* to be a local maximum. Thus, it is also a sufficient condition for the EM iterates θ_k to converge to a local maximum θ^*. When $I_{obs}(\theta^*; y)$ is positive semidefinite, the eigenvalues of $J(\theta^*)$ lie in the interval $[0, 1]$, and therefore, convergence is not guaranteed.

Another sufficient condition for the EM iterates to converge (to some θ^∞, not necessarily equal to the MLE, θ^*) is that the EM mapping F be Lipschitz continuous with Lipschitz constant $L < 1$, i.e.

$$\forall \theta_1, \theta_2 \in \Omega : \|F(\theta_1) - F(\theta_2)\| \leq L\|\theta_1 - \theta_2\|, \text{ where } L < 1. \tag{15}$$

In other words, the mapping F should be contractive over entire Ω to ensure convergence. This would be guaranteed if $\rho(J(\theta)) < 1$, $\forall\, \theta \in \Omega$, which is a stronger condition than $\rho(J(\theta^*)) < 1$. Thus, the convergence of the EM can be very slow when $\rho(J(\theta)) = 1 - \epsilon$, for some small $\epsilon > 0$. The convergence is logarithmic (or sublinear) when $\rho(J(\theta)) = 1$, in which case the convergence is excruciatingly slow. Even Newton's method crawls to a linear convergence. An eigenvalue of unity in a neighborhood of θ^* implies a ridge in $L(\theta^*)$ through θ^*. The EM iterations may not converge to a local maximum or may even diverge when $\rho(J(\theta)) > 1$. Interestingly, we will later see in Section 3.5 (page 22) that the proposed extrapolation methods can converge to the "anti-limit" of diverging sequences under less restrictive (than contractivity) conditions on F.

It is also interesting to observe that the iteration history of the EM algorithm is unaffected by parameter transformations that are *homeomorphic*. In fact, the convergence of the transformed sequence is identical to that of the original sequence. Although this observation is relatively easy to demonstrate, to our knowledge, it has not been made before in the EM literature. Let $F : \Omega \mapsto \Omega \subset \mathbb{R}^p$ denote the EM mapping. A transformation $T : D \subset \mathbb{R}^p \mapsto \Omega$ is a homeomorphism of D onto Ω if T is one-to-one on D and T and T^{-1} are continuous on D and Ω, respectively. We state and prove the following theorem [31]:

Theorem 3.1. *Given F and a homeomorphism T, such that $T^{-1}FT$ is a contraction on D, then F has precisely the same number of fixed points in Ω*

as does $T^{-1}FT$ in D. For any $x_0 \in \Omega$ the iterates given by Eq. 2 remain in Ω and converge if and only if the sequence $z_{n+1} = T^{-1}FTz_n$, $n = 0, 1, \cdots$, with $z_0 = T^{-1}x_0$, remains in D and converges.

Proof. If $x^* \in \Omega$ is a fixed point of F, then $z^* = T^{-1}x^*$ is a fixed point of $T^{-1}FT$, and conversely. Furthermore, since the sequences x_n and z_n are related by $x_n = T\,z_n$, they must have identical convergence behavior. □

We also remark that in analogy to matrix theory, the mapping F is "similar" to the contraction $T^{-1}FT$. Therefore, they have identical eigen structure. This property has an important consequence that while it is not possible to affect a change in the rate of convergence of the EM via a parameter transformation, we can, as will be seen later in the examples, significantly improve the rate of convergence of the one-step and SQUAREM schemes using appropriate parameter transformations.

3.2 Scalar Sequence Transformations and Extrapolation

Extrapolation is based on interpolation. In fact, it is interpolation at a point outside, rather than inside, the interval containing the interpolation points. Usually, this point either 0 or ∞. Extrapolation is commonly used in numerical analysis to improve the accuracy of a process depending on a parameter or to accelerate the convergence of a sequence. In dealing with sequences, we are typically interested in approximating the limit of a sequence x_n as $n \to \infty$, given $x_n, x_{n+1}, \cdots, x_{n+m}$. This is clearly an extrapolation problem. The most celebrated extrapolation schemes are the Romberg's method for improving the convergence of the trapezoidal rule for numerical quadrature, and the Aitken's Δ^2 process. The new class of schemes proposed here for the acceleration of fixed point iterations, in general, and the EM algorithm, in particular, exploit the strong connection between extrapolation methods, sequence transformations, and fixed point iterations. In this and the following sections, we present the essential background material on sequence transformations and extrapolation methods for gaining a deeper understanding of the new acceleration schemes. The interested reader should consult [2] for a more extensive treatment of this material.

Let us consider a scalar sequence x_n that converges to a limit x, but whose convergence needs to be accelerated. We will construct a sequence transform t_n that has the following properties:

1. t_n converges.

2. t_n also converges to x.

3. t_n converges to x faster than x_n, i.e.

$$\lim_{n\to\infty}(t_n - x)/(x_n - x) = 0.$$

If these three conditions are satisfied then t_n is said to accelerate the convergence of x_n. In general, a sequence transformation can be written as:

$$t_n = F(x_n, \ldots, x_{n+k}). \tag{16}$$

Two instances of such a sequence transformation are $t_n = (x_n + x_{n+1})/2$, and $t_n = (x_n x_{n+2} - x_{n+1}^2)/(x_{n+2} - 2x_{n+1} + x_n)$. The first one is a linear transformation that satisfies properties (1) and (2). In order to find the class of sequences which it accelerates, we write

$$\frac{t_n - x}{x_n - x} = \frac{1}{2}\left(1 + \frac{x_{n+1} - x}{x_n - x}\right).$$

Therefore,

$$\lim_{n\to\infty}\frac{t_n - x}{x_n - x} = 0, \text{ if and only if } \lim_{n\to\infty}\frac{x_{n+1} - x}{x_n - x} = -1,$$

which shows that this linear transformation is only able to accelerate the convergence of a very restricted class of sequences. This is essentially the case for all linear summation processes. Let us now look at the second example, which is easily recognized as the Aitken's Δ^2 process. It is easy to show that it accelerates the convergence of all the sequences for which there exists a $\rho \in [-1, 1)$ such that

$$\lim_{n\to\infty}\frac{x_{n+1} - x}{x_n - x} = \rho,$$

which is clearly a much wider class of sequences. It can be proved that if t_n converges, then its limit is the same as that of x_n, although there are situations where t_n from the Aitken's process has two accumulation points.

A foremost aspect of the study of sequence transformations is the determination of the *kernel* of the transformation, which is the set of sequences

for which $\exists\, x$ such that $\forall n > N, t_n = x$, for some N. The kernel of the linear summation process is the set of sequences of the form

$$x_n = x + c(-1)^n,$$

where c is a scalar. For the Aitken's process the kernel is given by

$$x_n = x + c\lambda^n,$$

where c and λ are scalars with $a \neq 0$ and $\lambda \neq 1$. Note that the kernel of Aitken's process contains that of the linear summation process. In the Aitken's process, x is the limit of the sequence x_n if $|\lambda| < 1$. If $|\lambda| > 1$, x_n diverges and x is called its anti-limit. If $|\lambda| = 1$, x_n has no limit at all, or it takes a finite number of distinct values in which case x is their arithmetic mean. In the above two example, we were able to obtain the kernel in an explicit form. However, the kernel may also be written in an implicit manner by means of a relation which holds among consecutive terms of the sequence. Thus for the linear summation process we write its kernel implicitly as

$$x_{n+1} - x = -(x_n - x),\ \forall n > N.$$

For the Aitken's process we write

$$x_{n+1} - x = \lambda(x_n - x),\ \forall n > N.$$

Solving the difference equation leads to the explicit form of the kernel. Both forms are equivalent and depend on parameters, x and c in the case of linear summation process, and x, c and λ in the case of Aitken's Δ^2 process. The implicit form of the kernel can generally be written as

$$K(x_n, \ldots, x_{n+q}; x, c_1, \ldots, c_p) = 0,$$

which must be satisfied by any sequence x_n, that belongs to the kernel $\mathcal{K}_T$ of the transformation T. A sequence transformation $T : x_n \mapsto t_n$ is said to be an extrapolation method if it is such that $\forall n, t_n = x$ if and only if $x_n \in \mathcal{K}_T$.

Let us now see how an extrapolation method is built from its kernel, that is from the implicit relation K. We are given $x_n, x_{n+1}, \ldots, x_{n+p+q}$, and we would like to develop a sequence $u_n \in \mathcal{K}_T$ satisfying the "interpolation" conditions:

$$u_i = x_i,\ i = n, n+1, \ldots, n+p+q.$$

Since $u_n \in \mathcal{K}_T$, it satisfies the implicit kernel relation

$$K(u_i, \ldots, u_{i+q}; x, c_1, \ldots, c_p) = 0,\ i = n, \ldots, n+p.$$

This is a system of $(p+1)$ equations in $(p+1)$ unknowns, $x, c_1, \ldots, c_p$ whose solution (if it exists) depends on the index n. In order for this system to have a solution, we assume that $\frac{\partial K}{\partial x} \neq 0$. This guarantees by the implicit function theorem, the existence of a function G (depending on the unknown parameters $c_1, \ldots, c_p$) such that

$$x = G(x_i, \ldots, x_{i+q}),\ i = n, \ldots, n+p.$$

The solution $t_n = x$ of this system depends only on the terms of the original sequence, $x_n, \ldots, x_{n+p+q}$. Thus, we obtain the extrapolation method, which since it depends on n, will be denoted as t_n

$$t_n = F(x_n, \ldots, x_{n+k}).$$

Sometimes it is also denoted by $t_n^{(k)}$ to signify that it also depends on $k = p+q$.

Let us illustrate the development of an extrapolation method with an example. Let us assume that the implicit kernel has the following form:

$$K(u_i, u_{i+1}; x, c_1, c_2) = c_1(u_i - x) + c_2(u_{i+1} - x) = 0,$$

where $c_1 + c_2 \neq 0$. We can assume, without any loss of generality, that $c_1 + c_2 = 1$. We now have to solve the system

$$\begin{aligned} c_1(x_i - x) + c_2(x_{i+1} - x) &= 0 \\ c_1(x_{i+1} - x) + c_2(x_{i+2} - x) &= 0. \end{aligned}$$

The system has a unique solution for x, since the derivative $\frac{\partial K}{\partial x} = -(c_1+c_2) = -1$. The function G is given by

$$G = c_1 u_i + c_2 u_{i+1},$$

and the system to be solved now becomes

$$\begin{aligned} t_n &= x = c_1 x_n + (1 - c_1) x_{n+1} \\ t_n &= x = c_1 x_{n+1} + (1 - c_1) x_{n+2}. \end{aligned}$$

Now adding and subtracting x_n in the first equation and x_{n+1} in the second equation, results in the equivalent system

$$\begin{aligned} x_n &= t_n + (c_1 - 1)\Delta x_n \\ x_{n+1} &= t_n + (c_1 - 1)\Delta x_{n+1}, \end{aligned}$$

where Δ is the forward difference operator defined as $\Delta x_i = x_{i+1} - x_i$. The solution for t_n can be written using Cramer's rule as a ratio of two determinants

$$t_n = \frac{\begin{vmatrix} x_n & x_{n+1} \\ \Delta x_n & \Delta x_{n+1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ \Delta x_n & \Delta x_{n+1} \end{vmatrix}}. \tag{17}$$

Performing the above computation yields

$$\begin{aligned} t_n &= \frac{x_n \Delta x_{n+1} - x_{n+1}\Delta x_n}{\Delta x_{n+1} - \Delta x_n} \\ &= \frac{x_n x_{n+2} - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n}, \end{aligned} \tag{18}$$

which is the Aitken's Δ^2 process.

Let us now consider a more complicated problem that will illustrate the problems in our approach which we had just described. We assume that the implicit kernel K has the form

$$K(u_i, \ldots, u_{i+q}; x, c_1, \ldots, c_p) = c_1(u_i - x) + c_2(u_{i+1} - x) + \cdots + c_{p+1}(u_{i+q} - x) = 0,$$

where we assume that $c_1.c_{p+1} \neq 0, \sum_{i=1}^{p+1} c_i = 1$, and also let $p = q = k$. Performing the same procedures describe above for the Aitken's method leads to the following $(k+1) \times (k+1)$ system, where the variables b_i depend on $c_j, j = 1, \ldots, p$.

$$\begin{array}{l} t_n + \ b_1\Delta x_n \ + \cdots + b_k\Delta x_{n+k-1} = \ x_n \\ t_n + b_1\Delta x_{n+1} + \cdots + \ b_k\Delta x_{n+k} \ = x_{n+1} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ t_n + b_1\Delta x_{n+k} + \cdots + b_k\Delta x_{n+2k-1} = x_{n+k}. \end{array} \tag{19}$$

Solving this system once again by the classical Cramer's rule yields

$$t_n^{(k)} = \frac{\begin{vmatrix} x_n & x_{n+1} & \cdots & x_{n+k} \\ \Delta x_n & \Delta x_{n+1} & \cdots & \Delta x_{n+k} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta x_{n+k-1} & \Delta x_{n+k} & \cdots & \Delta x_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta x_n & \Delta x_{n+1} & \cdots & \Delta x_{n+k} \\ \cdots & \cdots & \cdots & \cdots \\ \Delta x_{n+k-1} & \Delta x_{n+k} & \cdots & \Delta x_{n+2k-1} \end{vmatrix}}. \tag{20}$$

This is the classical sequence transformation known as the *Shanks transformation* [37]. It involves the computation of two determinants each of dimension $(k+1)$, so requires $2(k+1)(k+1)!$ multiplications. This is prohibitive even for moderate k, and more importantly the results will be extremely susceptible to roundoff errors due to finite arithmetic of the computers. So clearly this is not the way to compute t_n. Numerical analysts have developed special algorithms to compute such ratios of determinants with special structures, since sequence transformations with linear kernels can be expressed as the ratio of two determinants. These algorithms are called the *extrapolation algorithms.*

3.3 Vector Sequence Transformations and Extrapolation

Let us examine the determinant in the numerator of the Shanks transformation, Eq. 20. If this determinant is expanded with respect to its first row we obtain

$$t_n^{(k)} = \alpha_0 x_n + \cdots + \alpha_k x_{n+k}, \tag{21}$$

where α_i's are the solution to the following system of equations

$$\begin{array}{lllllll} \alpha_0 & + & \alpha_1 & + \cdots + & \alpha_k & = 1 \\ \alpha_0 \Delta x_n & + & \alpha_1 \Delta x_{n+1} & + \cdots + & \alpha_k \Delta x_{n+k} & = 0 \\ \cdots & & & & & \\ \alpha_0 \Delta x_{n+k-1} & + & \alpha_1 \Delta x_{n+k} & + \cdots + & \alpha_k \Delta x_{n+2k-1} & = 0 \end{array}. \tag{22}$$

Expressing the Shanks sequence transformation in this manner facilitates its applicability to vector sequences, where $x_n \in \mathbb{R}^p$. For vector sequences, we observe that in Eq. 22, $\Delta x_n, \ldots, \Delta x_{n+2k-1}$ are all vectors, i.e. $\forall i, \Delta x_{n+i} \in \mathbb{R}^p$.

Hence, each equation (except the first) in Eq. 22 is a vector equation describing system of p equations. Therefore, we have a total of $pk+1$ equations in $k+1$ unknowns. This overdetermined system is inconsistent, in general. Consequently, there are many different approaches to obtain $\alpha_0, \ldots, \alpha_k$. One fairly general way to accomplish this is to form the inner product of each member equation of the system (except the first equation) with arbitrary vectors in $\mathbb{R}^p$, $y_i^{(n)}, i = 1, \ldots, k$. This will results in the following system which is equivalent to Eq. 22:

$$\begin{array}{cccccc}
\alpha_0 & + & \alpha_1 & + \cdots + & \alpha_k & = 1 \\
\alpha_0 \langle y_1^{(n)}, \Delta x_n \rangle & + \alpha_1 \langle y_1^{(n)}, \Delta x_{n+1} \rangle & + \cdots + & \alpha_k \langle y_1^{(n)}, \Delta x_{n+k} \rangle & = 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\alpha_0 \langle y_k^{(n)}, \Delta x_{n+k-1} \rangle & + \alpha_1 \langle y_k^{(n)}, \Delta x_{n+k} \rangle & + \cdots + & \alpha_k \langle y_k^{(n)}, \Delta x_{n+2k-1} \rangle & = 0
\end{array} \tag{23}$$

Now, this is a system of $k+1$ equations in $k+1$ unknowns. If the determinant of this system is non-singular, its solution provides $\alpha_0, \ldots, \alpha_k$, which are then used to obtain the vector sequence transformation via Eq. 21. The determinantal form of the vector Shanks transformation, analogous to Eq. 20, can now be presented as follows:

$$t_n^{(k)} = \frac{\begin{vmatrix} x_n & x_{n+1} & \cdots & x_{n+k} \\ \langle y_1^{(n)}, \Delta x_n \rangle & \langle y_1^{(n)}, \Delta x_{n+1} \rangle & \cdots & \langle y_1^{(n)}, \Delta x_{n+k} \rangle \\ \cdots & \cdots & \cdots & \cdots \\ \langle y_k^{(n)}, \Delta x_{n+k-1} \rangle & \langle y_k^{(n)}, \Delta x_{n+k} \rangle & \cdots & \langle y_k^{(n)}, \Delta x_{n+2k-1} \rangle \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \langle y_1^{(n)}, \Delta x_n \rangle & \langle y_1^{(n)}, \Delta x_{n+1} \rangle & \cdots & \langle y_1^{(n)}, \Delta x_{n+k} \rangle \\ \cdots & \cdots & \cdots & \cdots \\ \langle y_k^{(n)}, \Delta x_{n+k-1} \rangle & \langle y_k^{(n)}, \Delta x_{n+k} \rangle & \cdots & \langle y_k^{(n)}, \Delta x_{n+2k-1} \rangle \end{vmatrix}}. \tag{24}$$

We can obtain many of the popular vector sequence extrapolation methods from Eq.24 by choosing the arbitrary vector $y_i^{(n)}, i = 1, \ldots, k$, as follows:

- Minimal polynomial extrapolation (MPE) of Cabay and Jackson [7]: $y_i^{(n)} = \Delta x_{n+i}$
- Reduced rank extrapolation (RRE) of Eddy [10] and Mesina [30]: $y_i^{(n)} = \Delta^2 x_{n+i}$.
- Topological epsilon algorithm (TEA) of Brezinski: $y_i^{(n)} = y$

- Modified minimal polynomial extrapolation (MMPE) of Sidi [38]: $y_i^{(n)} = y_{i+1}$
- Henrici's method or full rank extrapolation [16]: $k = p$ which is the dimension of the vectors, and $y_i^{(n)} = e_i$, where e_i is a vector whose components are all zero except the i-th which is unity.

In the TEA, $y \in \mathbb{R}^p$ is an arbitrary fixed vector, and in the MMPE algorithm, $\{y_1, \ldots, y_k\}$ is a set of linearly independent vectors in $\mathbb{R}^p$.

Let us denote by Y_k and $\Delta^j X_{k,n} (j = 1, 2)$ the matrices whose columns are, respectively, $y_{1,n}, \ldots, y_{k,n}$ and $\Delta^j x_n, \ldots, \Delta^j x_{n+k-1}$. Now, in the numerator and the denominator of Eq. 24, we replace each column, starting from the second column, by its difference with the immediately previous column, to obtain

$$t_n^{(k)} = \frac{\begin{vmatrix} x_n & \Delta X_{k,n} \\ Y_{k,n}^T \Delta x_n & Y_{k,n}^T \Delta^2 X_{k,n} \end{vmatrix}}{\left| Y_{k,n}^T \Delta^2 X_{k,n} \right|}$$

Using Schur's formula for determinants of block matrices, we can now express the vector extrapolation scheme, $t_n^{(k)}$, in a compact matrix form as

$$t_n^{(k)} = x_n - \Delta X_{k,n} (Y_{k,n}^T \Delta^2 X_{k,n})^{-1} Y_{k,n}^T \Delta x_n, \tag{25}$$

where $Y_{k,n} = \Delta X_{k,n}$ for the MPE, and $Y_{k,n} = \Delta^2 X_{k,n}$ for the RRE. Note that Aitken's Δ^2 method is recovered when $p = k = 1$, regardless of the choice of $Y_{k,n}$. In order for $t_n^{(k)}$ to be defined, the $k \times k$ matrix $(Y_{k,n}^T \Delta^2 X_{k,n})$ must be non-singular. Sidi [40] provides the conditions under which this is satisfied. For $k = 1, 2$, the vector extrapolation methods, $t_n^{(k)}$ can be computed directly from Eq. 25. For larger values of k, however, the computation of $t_n^{(k)}$ can be done using the recursive algorithms proposed in [12] and [20].

3.4 Some Convergence Results for the Vector Extrapolation Methods

Consider the p-dimensional, linear, fixed point problem

$$x = A\,x + b. \tag{26}$$

Let $\lambda_1, \ldots, \lambda_p$ and $v_1, \ldots, v_p$ be the eigenvalues and corresponding eigenvectors of A. Assume that $\lambda_i \neq 1, \forall i$, so that Eq. 26 has a unique solution x^*.

For a given x_0, we generate the sequence $x_n, n = 1, 2, \ldots$, by the iteration

$$x_{n+1} = A\, x_n + b,\ n = 0, 1, \cdots . \tag{27}$$

If we let $x_0 - x^* = \sum_{i=1}^{p} \rho_i v_i$, for some scalars ρ_i, then

$$x_n = x^* + \sum_{i=1}^{p} \rho_i v_i \lambda_i^n,\ n = 0, 1, \cdots . \tag{28}$$

The limit of the sequence x_n will be x^* provided the modulus of the largest eigenvalue is less than unity, i.e. $|\lambda_1| < 1$; otherwise the sequence diverges and x^* is the antilimit. Sidi [39] proved the following important result to establish that the extrapolation methods given by Eq. 25, in particular the MPE and RRE, are bona fide acceleration procedures of the iterative method, Eq. 27 for solving 26.

Theorem 3.2. *If A is diagonalizable and its distinct nonzero eigenvalues denoted by $\lambda_j, j = 1, 2, \ldots$, are ordered such that*

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \ldots,$$

and provided

$$|\lambda_k| > |\lambda_{k+1}|,$$

we have

$$t_n^{(k)} - x^* = O(|\lambda_{k+1}|^n),\ n \to \infty. \tag{29}$$

The coefficient of $|\lambda_{k+1}|^n$ on the right-hand side of Eq. 29 becomes large when the dominant eigenvalues $\lambda_1, \lambda_2, \ldots$, are close to 1. Also, in view of the fact that $x_n - x^* = O(|\lambda_1|^n),\ n \to \infty$, we have the following Lemma:

Lemma 3.3. *The MPE and RRE are true acceleration methods in the sense that*

$$\frac{\|t_n^{(k)} - x^*\|}{\|x_{n+k+1} - x^*\|} = O\left[\left(\frac{\lambda_{k+1}}{\lambda_1}\right)^n\right],\ n \to \infty. \tag{30}$$

The reason for writing x_{n+k+1} in Eq. 30 is that the extrapolation $t_n^{(k)}$ in both MPE and RRE schemes is computed from $x_n, x_{n+1}, \cdots, x_{n+k+1}$. The lemma implies that if $x_n \to x^*$, i.e. $|\lambda_1| < 1$, then $t_n^{(k)} \to x^*$, and more quickly. Also, if $\lim_{n\to\infty} x_n$ does not exist, i.e. $|\lambda_1| > 1$, then $t_n^{(k)} \to x^*$, provided that

$|\lambda_{k+1}| < 1$. It also implies that when the MPE and RRE methods are applied to a vector sequence generated as in Eq. 27, they will be especially effective when A has a small number of dominant eigenvalues (k-many when $t_n^{(k)}$ is used) that are well separated from the small eigenvalues.

The extrapolation methods MPE and RRE are direct methods for solving the linear system $(I_p - A)x = b$, in the sense that their application with $k = k^*$, where k^* is the degree of the minimal polynomial of A with respect to $x_n - x^*$, yields the solution x^* in a finite number of steps, provided $I_p - A$ is non-singular. In particular, we have $t_n^{(k^*)} = x^*$. If the sequence generator F is not linear (as in Eq. 6), but has a Taylor series expansion in which the linear part dominates in a suitably small neighborhood of a fixed point, then the extrapolation methods can still be applied. The powerful results of Theorem 3.2 and Lemma 3.3 for the linear system of equations, are also asymptotically valid for the nonlinear fixed point problem of the EM algorithm, since for n sufficiently large, $x_n, x_{n+1}, \ldots$ are all very close to x^*, and we have

$$x_{n+1} - x^* = J(x^*)(x_n - x^*) + O(\|x_n - x^*\|^2), \tag{31}$$

where $J(x^*)$ is the Jacobian matrix of the EM mapping F evaluated at the fixed point x^*. This implies that the sequence behaves linearly at infinity in the sense that

$$x_{n+1} \approx J(x^*)x_n + (I_p - J(x^*))x^*,$$

for all n sufficiently large. For nonlinear fixed point problems, such as the EM algorithm, we typically do not know either the linearization matrix $A = J(x^*)$ or the additive vector $b = (I_p - J(x^*))x^*$, but only have the sequence x_n. This is not a problem since the proposed iterative methods do not require explicit knowledge of A and b.

3.5 Cycling With Extrapolation Methods

Now we discuss a strategy called *cycling* that can be utilized to take advantage of the powerful vector extrapolation methods given by Eq.25 to solve the problem of accelerating the convergence of the EM algorithm. Note that the iterative methods proposed here, based on the strategy of cycling vector sequence transformations, are applicable more generally for accelerating the convergence of Picard iterations for fixed-point determination. The general problem is to accelerate the convergence of any iterative scheme that is linearly converging with a slow rate of convergence. We will term as "base

iteration" the basic numerical scheme that needs to be accelerated. The strategy of cycling works by building, within each cycle, a vector sequence with a certain number of base iterations, and then using these iterates in the extrapolation scheme given by Eq. 25 to compute an "extrapolated" vector. This vector will be used to start the next cycle to generate another vector sequence using the base iterations, and so on. It is important to distinguish the terms "cycles" and "iterations." An iteration denotes the single application of the base scheme such as the EM or Picard's method, whereas a cycle refers to the application of a vector extrapolation method using two or more iterates from the base scheme. Specifically, this is how we obtain an iterative scheme by cycling with an extrapolation method such as Eq. 25:

1. Let x_n be the value of parameters at the start of the $(n+1)$-th cycle, and let $u_0^{(n+1)} = x_n$.

2. Apply the base iterations k times to get $u_1^{(n+1)}, \ldots, u_k^{(n+1)}$, where
$$u_{i+1}^{(n+1)} = F(u_i^{(n+1)}),\ i = 0, \ldots, k-1.$$

3. Apply the extrapolation scheme given by Eq.25 to the sequence $u_0^{(n+1)}, \ldots, u_k^{(n+1)}$ to obtain $t_n^{(k)}$.

4. Set $x_{n+1} = t_n^{(k)}$, and check for convergence.

5. If convergence has been attained stop cycling, otherwise go back to step (1) for the next cycle.

It can be argued that cycling is the best and most natural way to implement vector extrapolation schemes. There are several compelling reasons to cycle with the extrapolation schemes:

1. The accuracy of a cycled k-th order MPE or RRE iterative scheme with m cycles is comparable to that of extrapolation $t_{m(k+1)}$, obtained from $x_0, x_1, \ldots, x_{m(k+1)}$.

2. Cycling is m times less expensive computationally than pure extrapolation.

3. Cycling requires m times less storage than pure extrapolation, since it only uses $k+1$ vectors at a time, whereas pure extrapolation needs to store all the $m(k+1)$ vectors.

4. With cycling we have explicit control of the extrapolation error via the number of cycles, whereas in pure extrapolation the number of terms in the sequence is fixed a priori.

A minor drawback with cycling is that the vector iterates have to be computed each time the code is executed, whereas in pure extrapolation the members of the sequence $x_0, x_1, \ldots, x_{m(k+1)}$ can be computed once and stored away to be used again with different extrapolation parameters and/or schemes.

Smith et al. [42] show that under the assumptions that (1) F has continuous Fréchet derivative in Ω, (2) the Jacobian matrix $J(x^*)$ does not have 1 as an eigenvalue, (3) k is chosen on the $(n+1)$-th cycle to be the degree of the minimal polynomial of $J(x^*)$ with respect to $x_n - x^*$, and (4) x_0 is sufficiently close to x^*, the cycled extrapolation method just described is quadratically convergent in the following sense

$$\|x_{n+1} - x^*\| = O\left(\|x_n - x^*\|^2\right).$$

A classical example that illustrates the strategy of cycling is the connection between Aitken's Δ^2 method for extrapolation and the Steffensen's iterative scheme for fixed point problem in the scalar case. As we saw earlier in Section 3.2, Aitken's Δ^2 method, Eq. 18, is a sequence transformation method that takes a sequence $x_n \in \mathbb{R}$ and produces a faster converging sequence t_n. In this approach, all the members of the sequence x_n are produced a priori by the EM or Picard iterations, and then the Aitken's extrapolation method is applied to them, three terms at a time. The terms x_i, x_{i+1}, and x_{i+2}, $i = 0, \ldots$ are used to obtain t_i, $i = 0, \ldots$. The Steffensen's method, on the other hand, is an iterative scheme, which can be stated explicitly as

$$x_{n+1} = x_n - \frac{(F(x_n) - x_n)^2}{F(F(x_n)) - 2\,F(x_n) + x_n}. \tag{32}$$

It can also be obtained from Aitken's Δ^2 process, using cycling, as follows:

1. Let x_n be the value of the parameter at the start of $(n+1)$-th cycle. Also, let $u_0^{(n+1)} = x_n$.
2. Apply the base iterations twice to get $u_1^{(n+1)} = F(u_0^{(n+1)})$, and $u_2^{(n+1)} = F(u_1^{(n+1)})$.
3. Apply the Aitken's Δ^2 formula, Eq. 18, to $u_0^{(n+1)}, u_1^{(n+1)}$ and $u_2^{(n+1)}$, to obtain x_{n+1}.

4. Increment the cycle counter and repeat steps (1) - (3) until convergence.

While the base iterations are only linearly convergent, the Steffensen's method produces a sequence that converges quadratically to a fixed point of F, provided $F'(x) \neq 1$ and the starting value x_0 is sufficiently "close" to the fixed point. This shouldn't be surprising given that the Steffensen's method is very closely related to the Newton-Raphson method for solving the nonlinear equation, $f(x) = F(x) - x = 0$. While the Newton-Raphson scheme uses the derivative of $f(x)$, $f'(x) = F'(x) - 1$, Steffensen's method uses a secant (forward difference) approximation to the derivative of F, i.e.

$$\begin{aligned} F'(x_n) &= \left(F(x_{n+1}) - F(x_n)\right)/(x_{n+1} - x_n) \\ &= \left(F(x_{n+1}) - F(x_n)\right)/(F(x_n) - x_n). \end{aligned}$$

An interesting property is that the mapping F must be contractive in order for the base iterations to converge, but this condition is not necessary for the convergence of Steffensen's method. This is also the case for the multivariate extensions of the Steffensen's method obtained from Eq. 25 via cycling. For the scalar case we saw in Section 3.2 that the kernel for Aitken's method is the sequence of the form

$$x_n = x^* + a\lambda^n.$$

The sequence converges when $\lambda < 1$, and diverges if $\lambda > 1$, in which case x^* is called its *antilimit.* Even when the base iterations diverge, the cycled extrapolation method may converge to the *antilimit*, x^*. This was observed by Henrici [16] for a multivariate scheme that he proposed as an extension of the Steffensen's method for a scalar parameter. In fact, Henrici gave an example (in $\mathbb{R}^2$) which showed the convergence of his multivariate extension of Steffensen scheme with an expanding map F, where not only was $\|J(x^*)\| > 1$, but also $|\lambda_{min}(J(x^*))| > 1$. It was later proved by Nievergelt [32] that the conditions of boundedness of second partial derivatives of F and of the coincidence of the minimal and characteristic polynomial of the Jacobian of F at x^* (which is trivially true for the scalar situation), are sufficient to guarantee the stability and convergence of the multivariate Steffensen's method proposed by Henrici.

Henrici's method can also be recovered, via cycling, from the general sequence transformation in Eq. 25, by setting $k = p$ and $y_i = e_i$, where e_i are the unit coordinate vectors of the p-dimensional Euclidean space, i.e. e_i

is a vector whose components are all zero except the i-th which is unity. This method has been termed "the multivariate Aitken's method" by Louis [26] and it was implemented in [22]. However, as Smith et al. [42] demonstrate, a more natural extension of the Aitken's method to vector sequences is obtained using the RRE method with $k = k^* \leq p$, where k^* is the degree of the minimal polynomial of J^* with respect to $x_n - x^*$, and J^* is the Jacobian matrix of F evaluated at a fixed point x^* (The minimal polynomial of a $p \times p$ matrix A with respect to a vector y is defined as the monic polynomial $P(.)$ of smallest degree $k \leq p$ such that $P(A).y = 0$. Note that the minimal polynomial divides the characteristic polynomial and is unique [23]). To see this, we write RRE (from Eq. 25) as

$$t_n^{(k)} = x_n - \Delta X_{k,n}(\Delta^2 X_{k,n})^+ \Delta x_{k,n} \tag{33}$$

where $A^+ = (A^T A)^{-1} A^T$ denotes the Moore-Penrose generalized inverse of the matrix A. When A is a non-singular square matrix, then $A^+ = A^{-1}$, in which case the RRE method with $k = p$ is the same as Henrici's method.

Typically, the degree of the minimal polynomial is less than thedimension of the vectors, hence Henrici's method is inefficient when compared to RRE. It is also more susceptible to numerical roundoff errors. Furthermore, it is clearly infeasible in problems with a large number of parameters. However, there are some problems with the RRE and MPE methods: (1) we do not know the degree of the minimal polynomial since we don't know J^*, and (2) the degree is dependent on the vector $x_n - x^*$ and hence can vary between cycles. Therefore, Smith et al. [42] suggest that an appropriate way to implement MPE and RRE iterative schemes on the first cycle is to extrapolate $t_0^{(k)}$ from $x_0, x_1, \ldots, x_{k+1}$ with $k = 1, 2, 3, \ldots$, and to stop when the residual norm $\|t_0^{(k)} - F(t_0^{(k)})\|$ is acceptable small. When there is a strong separation between the "dominant" and the "small" eigenvalues of $J(x^*)$, there is often a sharp decrease in the residual norm as soon as k equals the number of dominant eigenvalues (including multiplicities). If there is no such strong separation the decline in the residuals is gradual. However, with cycling, even a k large enough to produce only an order of magnitude difference between $\|t_0^{(k)}\|$ and $\|t_0^{(k)} - F(t_0^{(k)})\|$, can produce a significantly faster converging iterative scheme.

Furthermore, even though larger values of $k \leq k^*$ provide more accurate extrapolations, they are also prone to numerical problems such as stagnation, in the case of RRE, and breakdown (or near breakdown) in the case of

MPE. These problems will be discussed later in Section 5.2. The RRE and MPE methods can often be effectively used with small values of k. Even a value as low as 1 can surprisingly yield accurate extrapolations. First order SQUAREM schemes ($k = 1$) are most efficient from a computational perspective, since they involve little additional effort but yield significantly faster convergence than the corresponding one-step iterative schemes (described in the next section, Section 4). In contrast, higher order SQUAREM schemes involve greater additional computational effort, and thus may be less efficient. Therefore, for the rest of this report our focus will be on schemes with the lowest order, $k = 1$. Even though we don't expect the convergence to be quadratic, it will be demonstrated here that for a variety of problems the convergence of the SQUAREM schemes is still much faster than that of the base iterations of the EM. Evaluation of the performance of other low order schemes (e.g., $k = 2$) for accelerating the EM will be the focus of future studies.

4 One-Step Iterative Schemes for EM Acceleration

Consider schemes of the following form

$$x_{n+1} = x_n - A_n(F(x_n) - x_n), \tag{34}$$

where A_n is a $p \times p$ matrix, for solving the fixed point problem:

$$f(x) = x - F(x) = 0.$$

This is a very general representation and we can obtain a large number of numerical schemes, both old and new, by choosing A_n. Note that the extrapolation schemes given by Eq. 25 yield iterative schemes of this form (via cycling) with

$$A_n = \Delta X_{k,n} \left(Y_{k,n}^T \Delta^2 X_{k,n} \right)^{-1} Y_{k,n}^T.$$

Brezinski [6] provides an interesting and insightful classification of the various quasi-Newton schemes based on different strategies for choosing A_n. Three broad strategies are (1) A_n is a full matrix, (2) A_n is a diagonal matrix, and (3) A_n is a scalar matrix. Choosing A_n as a full matrix that approximates

$M(x^*) = I - F'(x^*)$, the Jacobian of $f(x)$ at x^*, yields the well-known quasi-Newton methods for the acceleration of EM. For example, Louis' [26] multivariate Aitken scheme is obtained when

$$A_n = (M(x_n) - I_p)^{-1} = I_{obs}(x_n; y)^{-1} I_{comp}(x_n; y),$$

where I_p is a $p \times p$ identity matrix, and I_{obs} and I_{comp} are given by Eqs. 14 and 10, respectively. The case where A_n is a diagonal matrix correspond to using a different relaxation parameter, $\alpha_n^j, j = 1, \ldots, p$, for each component of the vector function $f(x_n)$. Brezinski and Chehab [5] developed multiparameter iterative schemes for the solution of systems of linear and nonlinear equations.

When A_n is a scalar matrix of the form $\alpha_n I_p$, we obtain the classical gradient-type algorithms, which include steepest descent method:

$$x_{n+1} = x_n - \alpha_n (F(x_n) - x_n) \tag{35}$$

When $\alpha_n = -1$, $\forall n$, we obtain the EM scheme, and more generally, when $\alpha_n = \alpha$, we obtain relaxation methods, where the EM step $F(x_n) - x_n$ is relaxed by the factor $-\alpha$. Alternatively, they can also be considered as a cycled version of the linear extrapolation method, with the sequence transformation defined as

$$t_n = (1 - \alpha)\, x_n + \alpha\, x_{n+1}.$$

However, these methods with constant α_n are different from the extrapolation methods of the class of schemes given by Eq. 25, which are nonlinear extrapolation schemes. Here we remark that the relaxation (or the linear extrapolation) schemes are invariant under parameter transformations, whereas this is not the case for the nonlinear extrapolation schemes, where α_n depends nonlinearly on current and previous values of parameters. This is an important point because in many problems of MLE estimation, it is possible to accelerate nonlinear iterative schemes by an appropriate parameter transformation. We will demonstrate this later in some examples.

In Eq. 35, the quantity $F(x_n) - x_n$ is the search direction or the gradient direction, and the scalar α_n is the steplength. Solving the problem $f(x) = F(x) - x = 0$ is equivalent to minimizing the squared L_2-norm of $f(x)$, i.e. $\|f(x)\|^2$, for which the gradient is given by $\pm f(x)$. In the context of the EM algorithm, $F(x_n) - x_n$ can be interpreted as a generalized gradient, since we can show that

$$F(x_n) - x_n \approx -\{\partial^2 Q(x; x_n)/\partial x\, \partial x^T\}^{-1}_{x=x_n} \{\partial Q(x; x_n)/\partial x\}_{x=x_n}.$$

In fact, this result forms the basis of the EM gradient algorithm discussed by Lange [24]

$$\begin{aligned} x_{n+1} &= F(x_n) \\ &= x_n + (F(x_n) - x_n) \\ &\approx x_n - \{\partial^2 Q(x; x_n)/\partial x\, \partial x^T\}^{-1}_{x=x_n} \{\partial Q(x; x_n)/\partial x\}_{x=x_n}. \end{aligned}$$

This gradient algorithm is potentially useful in situations where the maximization of the Q function has to be performed iteratively, since it avoids the complete solution of the M step of the EM algorithm by just performing one Newton iteration while preserving the stability and convergence properties of the EM. However, unlike the EM, it is not necessarily an ascent algorithm, since the matrix $\{\partial^2 Q(x; x_n)/\partial x\, \partial x^T\}_{x=x_n}$ is not necessarily negative-definite, except when the complete data comes from a distribution belonging to the linear exponential family.

The simplest extrapolation methods are obtained from Eq. 25 by setting $k = 1$, in which case the matrices $\Delta^i X_{n,k} (i = 1, 2)$ have only one column $\Delta^i x_n (i = 1, 2)$, thus resulting in A_n being a scalar matrix. We obtain, via cycling, the following iterative procedures, which we call MPE1 and RRE1:
MPE1:

$$\begin{aligned} x_{n+1} &= x_n - \frac{\|r_n\|^2}{\langle r_n, v_n \rangle} r_n \\ &:= x_n - \alpha_n^{\text{MPE1}} r_n \end{aligned} \tag{36}$$

RRE1:

$$\begin{aligned} x_{n+1} &= x_n - \frac{\langle r_n, v_n \rangle}{\|v_n\|^2} r_n \\ &:= x_n - \alpha_n^{\text{RRE1}} r_n \end{aligned} \tag{37}$$

where $r_n = F(x_n) - x_n$ and $v_n = F(F(x_n)) - 2F(x_n) + x_n$.

The RRE1 scheme, Eq.45, is also known as the Lemaréchal method [3]. It can also be derived as follows. Consider iterative methods of the form 35. We set $u_0 = x_n, u_{i+1} = F(u_i), i = 0, 1, \ldots,$ and $z_i = u_i - \alpha_n(u_{i+1} - u_i)$. Note that $z_0 = x_{n+1}$. We now consider the differences,

$$\Delta z_i = \Delta u_i - \alpha_n \Delta^2 u_i,$$

where $\Delta^k, k = 1, 2$, is the forward difference operator: $\Delta^k u_i = \Delta^{k-1} u_{i+1} - \Delta^{k-1} u_i$. Let $y \in \mathbb{R}^p$ be an arbitrary vector. We choose α_n such that $\langle y, \Delta z_0 \rangle = 0$, and obtain:

$$\alpha_n = \frac{\langle y, \Delta u_0 \rangle}{\langle y, \Delta^2 u_0 \rangle}.$$

It can be readily shown that the choice $y = \Delta^2 u_0$ minimizes $\|\Delta z_0\|^2$, and leads to the Lemarechal's or RRE1 scheme. Similarly, the choice $y = \Delta u_0$, which results in the MPE1 scheme, minimizes $\|\Delta z_0\|^2 / \alpha_n^2$. According to Brezinski [6], the MPE1 scheme of Eq. 44 is new as an iterative method.

It should be remarked that the one stage extrapolation methods entail negligible additional computational effort beyond that of the basic EM algorithm. They require the computation of two vectors (r_n and v_n), two inner products, a scalar-vector multiplication, and a vector addition, all of which can be easily performed.

5 Squared Methods

Here we propose a new class of schemes which are based on the idea of applying the extrapolation schemes described in Section 4 in two stages. In the first stage, the extrapolation scheme (with α_n) is applied, yielding an intermediate update of the parameter vector. Then the extrapolation scheme is applied for the second time with the same α_n, but this time to update the intermediate vector. This idea was originally proposed in [33] for improving the performance of the classical Cauchy (steepest descent) method for minimizing a quadratic function. It was first proposed in Roland and Varadhan [35] for accelerating the Picard iterations for nonlinear fixed point problem. Roland and Varadhan developed the squared version of Lemarechal's and Marder-Weitzner schemes. Here we further develop this idea by incorporating the idea of squaring into a broader class of extrapolation schemes such as minimal polynomial extrapolation (MPE), and reduced rank extrapolation. This yields a rich class of iterative schemes for solving the nonlinear fixed point problem generated by the EM algorithm. In particular, we focus on the first order schemes ($k = 1$). We also propose an interesting hybrid scheme, for the first order extrapolation methods, which combines the nicer properties of the MPE and RRE schemes, while reducing their limitations such as stagnation and near breakdown.

5.1 Cauchy-Barzilai-Borwein Method

$$\min f(x) = \frac{1}{2}x^T Q x - b^T x, \qquad x \in \mathbb{R}^p \tag{38}$$

where $Q \in \mathbb{R}^{p\times p}$. This problem is equivalent to solving the linear system, $Qx = b$. The classical steepest descent method for this problem can be written as

$$x_{n+1} = x_n - \lambda_n g_n,$$

where $g_n = \bigtriangledown f(x_n) = Qx_n - b$, and the optimal choice of steplength is given by

$$\lambda_n = \frac{g_n^T g_n}{g_n^T Q g_n}.$$

For the optimal choice of steplength, the steepest descent method converges linearly as

$$\|x_{n+1} - x^*\|_Q \leq \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} \|x_n - x^*\|_Q$$

where for any $z \in \mathbb{R}^p$, the Q-norm is defined as $\|z\|_Q^2 = z^T Q z$, and λ_{max} and λ_{min} are the largest and the smallest eigenvalues of Q, respectively.

It is clear that the linear rate of the steepest descent method can be woefully inadequate if the smallest eigenvalue is very small relative to the largest eigenvalue, i.e. if the problem is ill-conditioned. Raydan and Svaiter argued that the notoriously slow convergence of the steepest descent method, even in problems only mildly ill-conditioned, was due primarily to the optimal choice of steplength and not to the choice of the gradient direction. Consider a relaxed Cauchy scheme of the form

$$x_{n+1} = x_n - \theta_n \lambda_n g_n,$$

where $0 \leq \theta_n \leq 2$, is a relaxation parameter. Notice that $\theta_n = 1$ (corresponding to non-relaxation of steplength) is the Cauchy method. Raydan and Svaiter demonstrate that relaxing the optimal Cauchy steplength in any manner would improve the performance of the Cauchy method, unless the search direction g_n happens to be an eigenvector of Q, in which case the Cauchy steplength yields the solution in one iteration. However, this optimal situation would seldom occur in practice. By choosing, at each iteration, the relaxation parameter to be a uniform random number in the interval $(0, 2)$, Raydan and Svaiter showed that even the randomly relaxed Cauchy method significantly outperformed the classical Cauchy method.

Barzilai and Borwein [1] proposed a nonmonotone gradient method, i.e. a gradient method that does not guarantee descent in the objective function, which for the quadratic function of Eq. 38 can be written as

$$x_{n+1} = x_n - \lambda_{n-1} g_n$$

λ_{n-1} is the optimal steplength (Cauchy step length) at the previous iteration. Raydan and Svaiter proposed a combination of the Cauchy (steepest descent) and the Barzilai-Borwein method to obtain a new method. They called this new non-monotone method the Cauchy-Barzilai-Borwein (CBB) method, and defined it as follows:
Given $x_0 \in \mathbb{R}^p$, at each iteration n we set:

$$\begin{aligned} h_n &= Q g_n, \\ t_n &= \frac{g_n^T g_n}{g_n^T h_n}, \\ z_n &= x_n - t_n g(x_n), \\ x_{n+1} = z_n - t_n g(z_n) &= z_n - t_n (Q z_n - b). \end{aligned}$$

Since

$$Q z_n - b = Q(x_n - t_n g_n) - b = g_n - t_n h_n,$$

we obtain

$$\begin{aligned} x_{n+1} &= z_n - t_n (g_n - t_n h_n) \\ &= x_n - 2\, t_n g_n + t_n^2 h_n. \end{aligned} \tag{39}$$

5.2 Squared Methods for EM Acceleration

It is easy to obtain the following error equations for the Cauchy and CBB iterative schemes:

$$\begin{aligned} e_{n+1} &= (I - t_n Q)^2 e_n \text{ for CBB} \quad \text{and} \\ e_{n+1} &= (I - t_n Q) e_n \text{ for Cauchy,} \end{aligned} \tag{40}$$

where $e_n = x_n - x$. Therefore, the CBB method could be considered as a "squared" Cauchy method.

We now extend Raydan and Svaiter's idea of *squaring* to the general nonlinear fixed-point problems. This, of course, means that the squaring idea is also applicable to the EM and its extensions such as GEM, ECM, and ECME, since they are all nonlinear fixed-point iterations. We first apply the one stage extrapolation scheme at x_n to obtain an intermediate vector z_n. Then we once again apply the extrapolation method (note: we can actually use a different extrapolation method at the second stage, resulting in a mixed scheme), but this time at z_n, to obtain the parameter update for the next cycle, x_{n+1}. The resulting scheme can be represented as

$$\begin{aligned} z_n &= x_n - \alpha_n \Delta x_n \\ x_{n+1} &= z_n - \alpha_n \Delta z_n \\ &= x_n - 2\alpha_n \Delta x_n + {\alpha_n}^2 \Delta^2 x_n \\ &= x_n - 2\alpha_n r_n + {\alpha_n}^2 v_n \end{aligned} \tag{41}$$

where $r_n = \Delta x_n = F(x_n) - x_n$ and $v_n = \Delta^2 x_n = F(F(x_n)) - 2F(x_n) + x_n$. Thus, in the squared method, the steplength α_n is computed once, but used twice. Two evaluations of the base iteration, F, are performed in each cycle of squared methods, as in the case of MPE1 and RRE1 schemes. The squared methods require only an additional scalar-vector product and a vector addition beyond that of the one stage schemes, MPE1 and RRE1. Hence the additional computational burden is negligible. Futhermore, it does not require any additional vector storage beyond that of the one stage methods. Therefore, for the one-step method given by (35), we have

$$\varepsilon_{n+1} = (I - \alpha_n(\psi - I))\varepsilon_n + o(\varepsilon_n), \tag{42}$$

where ψ is the jacobian of F at the fixed point x^*. By the remark (40), this new scheme has the following equation for the propagation of the error:

$$\varepsilon_{n+1} = [I - \alpha_n(\psi - I)]^2 \varepsilon_n + o(\varepsilon_n). \tag{43}$$

The one-step and squared extrapolation methods are nonlinear since the parameter α_n is a nonlinear function of iterates $x_i, i = 0, 1, \cdots, n$. The α_n for the one stage and squared MPE1 and RRE1 schemes are given in Eqs. 44 and 45 as

$$\alpha_n^{MPE1} = \frac{\langle r_n, r_n \rangle}{\langle v_n, r_n \rangle} \tag{44}$$

$$\alpha_n^{RRE1} = \frac{\langle v_n, r_n \rangle}{\langle v_n, v_n \rangle}, \tag{45}$$

where $r_n = F(x_n) - x_n$, and $v_n = F(F(x_n)) - 2F(x_n) + x_n$. The SqMPE1 and SqRRE1 methods are defined by Eq. 41, with α_n given, respectively, by Eqs. 44 and 45.

If, at some n, $\|r_n\| > \tau$, where τ is the stopping criterion, but r_n and v_n are nearly orthogonal, i.e. $|\langle v_n, r_n \rangle| \leq \varepsilon$, where $\varepsilon > 0$, is very small, then the schemes encounter numerical problems. The SqMPE1 scheme becomes unstable due to the magnification of round-off errors. In particular, due to a large value of α_n^{MPE1}, any small error in computing v_n, which is the second order difference vector, is magnified. This situation is known as *near breakdown* in numerical analysis. The SqRRE1, on the other hand, experiences *stagnation*, since $x_{n+1} = x_n$, although the iterations have not necessarily converged, i.e. $\|x_n - F(x_n)\| > 0$. An obvious way to partially overcome the closely related problems of stagnation and near breakdown is to not take the SQUAREM step, but instead use the two EM steps calculated in that cycle, i.e., when $|\langle v_n, r_n \rangle| \leq \varepsilon$, we set $x_{n+1} = F(F(x_n))$. A value of 0.01 seems to be a good choice for ε. Larger values of ε might slow down the convergence of the SQUAREM methods by frequently taking EM steps, while smaller values might be ineffective in avoiding stagnation and near breakdown.

We have also developed a new hybrid scheme, SqHyb1, that combines the squared MPE1 and RRE1 schemes. Noting that

$$-1 \leq \cos\theta_n = \frac{\langle v_n, r_n \rangle}{\|r_n\| \|v_n\|} \leq 1,$$

where θ_n is the angle between r_n and v_n, we define $w_n = |\cos\theta_n|$. Also note that

$$w_n = \left(\frac{\alpha_n^{RRE1}}{\alpha_n^{MPE1}} \right)^{1/2}, \tag{46}$$

which means that the extrapolation parameter of the RRE1 scheme is never greater, in magnitude, than that of the MPE1 scheme (they always have the same sign). This also explains the increased stability of the RRE1 and SqRRE1 schemes compared to that of MPE1 and SqMPE1. The squared hybrid scheme, SqHyb1, is given by

$$x_{n+1} = x_n - 2\alpha_n^{Hyb1} r_n + \left(\alpha_n^{Hyb1}\right)^2 v_n, \tag{47}$$

where

$$\alpha_n^{Hyb1} = w_n \alpha_n^{MPE1} + (1 - w_n)\alpha_n^{RRE1}.$$

Near breakdown and stagnation occur when u_n is nearly orthogonal to v_n, i.e. when $\langle v_n, r_n \rangle < \varepsilon$, but neither r_n nor v_n are correspondingly small. Under these conditions, we can write the steplength for the hybrid scheme using Eq. 46, as

$$\alpha_n^{Hyb1} \approx sgn(\langle v_n, r_n \rangle) \frac{\|r_n\|}{\|v_n\|} + \alpha_n^{RRE1}, \tag{48}$$

where $sgn(x) = x/|x|, x \neq 0$ is the *signum* function. It is evident that the scheme avoids near breakdown, due to the inner product $\langle v_n, r_n \rangle$ becoming nearly zero. Furthermore, it also overcomes stagnation since the term $\frac{\|r_n\|}{\|v_n\|}$ is added to the steplength of RRE1, thus ensuring a significant change in the iterates. However, there is still the possibility that v_n could be very small, without r_n being correspondingly small. Such an occurrence is not very likely, since it would require the cancellation of all the components in the vector difference $r_{n+1} - r_n$. If this happens, we accept the EM step for the cycle and set $x_{n+1} = F(F(x_n))$.

The MPE1 and RRE1 schemes can also be combined in a composite manner, where we apply the MPE1 (or RRE1) scheme to obtain z_n, followed by an application of the RRE1 (or MPE1) scheme to obtain x_{n+1}. This yields a scheme, which we call COMP1, defined as

$$x_{n+1} = x_n - \left(\alpha_n^{MPE1} + \alpha_n^{RRE1}\right) r_n + \left(\alpha_n^{MPE1} \alpha_n^{RRE1}\right) v_n. \tag{49}$$

The performance of COMP1, in terms of speed, is quite similar to that of the SqHyb1, but it is somewhat more prone to near breakdowns than the SqHyb1. We do not present any results from this composite scheme, since it does not belong to the family of SQUAREM schemes.

5.3 Convergence and Stability of the SQUAREM Methods

We first discuss the convergence and stability of SQUAREM methods when the relaxation parameter is constant, i.e $\alpha_n = \alpha$.

5.4 Stability

Let x^* a fixed point of the map F in a neighborhood of which we choose the initial point x_0. By (43), we have the following equation for the propagation

of the error

$$\varepsilon_{n+1} = [I - \alpha(\psi - I)]^2 \varepsilon_n + o(\varepsilon_n).$$

A sufficient stability condition for the SQUAREM method is:

$$\rho([I - \alpha(\psi - I)]^2) < 1 \text{ or equivalently } , -\frac{2}{a} < \alpha < 0, \tag{50}$$

where $a = sup_{\ t \in sp(\psi)} |1 - t|$. Then, by (50) and standard arguments, we deduce the following result [35]:

Lemma 5.1.
Assume that $I - \psi$ is non singular. There exists a neighborhood V of X such that, for $X_0 \in V$ and for $-\frac{2}{a} < \alpha < 0$, the SQUAREM method is convergent.

Let us now discuss the convergence of the squared methods for $k = 1$, and $\alpha_n = \alpha$, for all n, i.e.

$$x_{n+1} = x_n - 2\alpha(F(x_n) - x_n) + \alpha^2(F^2(x_n) - 2F(x_n) + x_n). \tag{51}$$

For this class of schemes, we can obtain the optimum value for α that would have the fastest linear convergence. Let $e_n = x_n - x^*$ denote the error at the n-th iteration, and $F^i(x_n)$ denote the composition of mapping F applied i times to x_n. We first note the following important relation (obtained by a simple Taylor series expansion of $F^i(x)$ about the fixed point x^*):

$$F^i(x_n) = x^* + [J(x^*)]^i \ e_n \ + \ o(e_n). \tag{52}$$

Using this we can write (ignoring the error term on the RHS of Eq. 52) the SQUAREM method with constant relaxation parameter α as

$$\begin{aligned} e_{n+1} &= e_n - 2\alpha \ [J(x^*) - I_p] \ e_n \ + \ \alpha^2 \, [J(x^*) - I_p]^2 \ e_n \\ &= [I_p \ - \ \alpha \, (J(x^*) - I_p) \,]^2 \, e_n \qquad (53) \\ &= A \, e_n. \qquad (54) \end{aligned}$$

where I_p is the $p \times p$ identity matrix. A sufficient condition for the convergence of this method is that

$$\rho(A) < 1,$$

i.e. the spectral radius (modulus of the largest eigenvalue) must be strictly less than unity. In general, the eigenvalues of the Jacobian are complex

since the Jacobian is not symmetric. In the EM settings, however, when the Hessian of the observed data log likelihood is negative definite, not only are the eigenvalues real, but they also lie in the half-open unit interval $[0, 1)$. So, for convergence it is sufficient to have

$$\sup_i \left[1 - \alpha\left(\lambda_i - 1\right)\right]^2 < 1,$$

where $0 \leq \lambda_i < 1$, $\forall\, i$. This condition can be rewritten as

$$\frac{-2}{1-\lambda_i} < \alpha < 0, \quad \forall i. \tag{55}$$

It can be shown (see [31], page 310) that the optimal value α_{opt}, i.e. the value of α that minimizes the spectral radius, $\rho(A)$, is given as

$$\alpha_{opt} = -\frac{2}{a+b}, \tag{56}$$

with $a = \inf_i |1-\lambda_i|$ and $b = \sup_i |1-\lambda_i|$. However, this convergence analysis is not of much practical value since the optimum relaxation parameter requires the knowledge of the MLE, x^*, which is, of course, unknown. Now we look at convergence for the general case in which the relaxation parameter α_n varies from cycle to cycle.

5.5 Convergence

We assume that F is monotone decreasing and satisfies a Lipschitz condition, that is

$$\forall y, z, \langle F(y) - F(z), y - z\rangle \leq 0 \tag{57}$$

$$\exists L > 0 \text{ such that } \forall y, z, || F(y) - F(z) || \leq L|| y - z || \tag{58}$$

where $\forall x,\ || x ||^2 = \langle x, x\rangle$. Let us remark that the condition (57) implies the uniqueness of the solution x^*. An important aspect of the proof of convergence is to show that the projection/extrapolation parameter, α_n, lies in $[0, 1]$. We first establish this for the three squared methods, RRE, MPE, and Hybrid. For SqRRE1 scheme, we have

$$\alpha_n^{RRE} = \frac{\langle r_n, v_n\rangle}{\langle v_n, v_n\rangle}. \tag{59}$$

Letting D_n be the denominator of α_n, we have the following relations

$$\alpha_n D_n = -|| F(x_n) - x_n ||^2 + \langle F(x_n) - x_n, F^2(x_n) - F(x_n)\rangle,$$

which by condition 57 shows that $\alpha_n < 0$, since $D_n > 0$. We also have, by virtue of (57), that

$$(1+\alpha_n)D_n = || F^2(x_n) - F(x_n) ||^2 - \langle F(x_n) - x_n, F^2(x_n) - F(x_n)\rangle > 0.$$

Therefore, $\alpha_n \in (-1, 0)$.

Similarly for SqMPE1, we have

$$\alpha_n^{MPE} = \frac{\langle r_n, r_n\rangle}{\langle r_n, v_n\rangle}. \tag{60}$$

But,

$$\begin{aligned}
\langle r_n, v_n\rangle &= \langle F(F(x_n)) - 2F(x_n) + x_n, F(x_n) - x_n\rangle \\
&= \langle F(F(x_n)) - F(x_n), F(x_n) - x_n\rangle - \|r_n\|^2 \\
&< 0 \quad \text{(because of monotonicity, Eq. 57)}
\end{aligned}$$

Therefore, $\alpha_n^{MPE} < 0$. Now consider $1 + \alpha_n$:

$$\begin{aligned}
1 + \alpha_n^{MPE} &= \frac{\langle v_n + r_n, r_n\rangle}{\langle v_n, r_n\rangle} \\
\langle v_n + r_n, r_n\rangle &= \langle F(F(x_n)) - F(x_n), F(x_n) - x_n\rangle \\
&< 0 \quad \text{(because of monotonicity, Eq. 57)}
\end{aligned}$$

Moreover, we just showed that $\langle v_n, r_n\rangle < 0$, and therefore, $1 + \alpha_n > 0$. Hence we have $-1 < \alpha_n^{MPE} < 0$. Since both α_n^{MPE} and α_n^{RRE} lie in $(-1, 0)$, any convex combination of them also lies in that interval. Thus, $\alpha_n^{Hyb} \in (-1, 0)$.

Now, by definition of the terms in the SQUAREM methods, we have

$$x_{n+1} - x^* = a_n(x_n - x^*) + b_n(F(x_n) - x^*) + c_n(F^2(x_n) - x^*)$$

with the positive variables a_n, b_n and c_n given by

$$\begin{aligned}
a_n &= (1+\alpha_n)^2 \\
b_n &= -2\alpha_n(1+\alpha_n) \\
c_n &= \alpha_n^2.
\end{aligned}$$

Thus, we have

$$\begin{aligned} || x_{n+1} - x^* ||^2 &= a_n^2 || x_n - x^* ||^2 + b_n^2 || F(x_n) - x^* ||^2 + c_n^2 || F^2(x_n) - x^* ||^2 \\ &+ 2a_n b_n (x_n - x^*, F(x_n) - x^*) + 2a_n c_n (x_n - x^*, F^2(x_n) - x^*) \\ &+ 2b_n c_n (F(x_n) - x^*, F^2(x_n) - x^*). \end{aligned}$$

Finally, we deduce by the assumptions (57) and (58) that

$$|| x_{n+1} - x^* ||^2 \leq M_n^2 \, || x_n - x^* ||^2,$$

with $M_n^2 = (a_n + L^2 c_n)^2 + b_n^2 L^2 > 0$.
But, M_n^2 is a polynomial of degree 4 in the variable α_n

$$M_n^2 = P(\alpha_n) = 1 + 4\alpha_n + (6 + 6L^2)\alpha_n^2 + (4 + 12L^2)\alpha_n^3 + (1 + L^4 + 6L^2)\alpha_n^4.$$

With standard arguments and by the fact that $\alpha_n \in (-1, 0)$, this polynomial decreases in $(-1, \alpha^*)$ and increases in $(\alpha^*, 0)$, where α^* is the zero of the derivative of $P(\alpha_n)$, in the interval $(-1, 0)$. On the other hand, $P(0) = 1$ and $P(-1) = L^4$. So, we deduce the following result

- If $L < 1$, then $M_n < M < 1$ and so the SQUAREM method converges.
- If $L > 1$, α_n must be such that $\mu < \alpha_n < 0$, where μ is such that $P(\mu) = 1$ in order to have $M_n < M < 1$.

Figure 1 plots the convergence polynomial, $P(\alpha_n)$, for various values of the Lipschitz constant, L. When $L < 1$, we have convergence for all $-1 \leq \alpha_n < 0$. As the Lipschitz constant increases, the maximum absolute value, μ, of α_n, for which convergence is guaranteed, becomes smaller. This value of μ can be obtained from the plots by picking out the abcissa of the intersection of the $P(\alpha_n)$ curve with the dashed line drawn at $y = 1$.

6 Description of Test Problems

We tested the performance of the SQUAREM methods on five different statistical problems where EM is the algorithm of choice for the computation of the maximum likelihood estimates. In all these problems, the SQUAREM methods are evaluated at three levels. At the first level, they are evaluated on the basis of their acceleration of the EM algorithm. At the second level, each SQUAREM scheme is compared to its one stage counterpart. Finally, their performance, relative to each other, is evaluated.

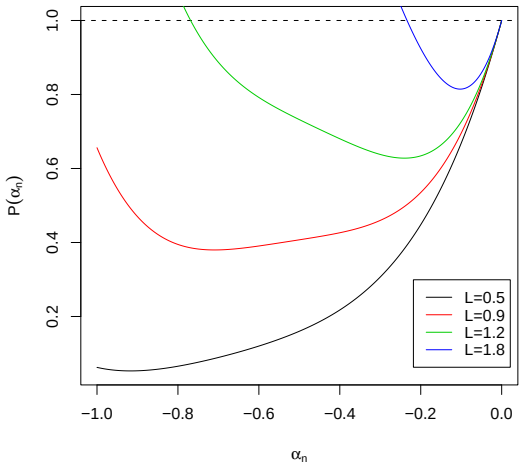

Figure 1: Convergence polynomials of the SQUAREM schemes for different Lipschitz constants

6.1 Poisson Mixtures

A finite mixture distribution with C components can be written as,

$$g(y; \theta_1, \cdots, \theta_C) = \sum_{j=1}^{C} p_j \, g_j(y; \theta_j), \tag{61}$$

where the probability density functions, g_j, are typically of the same form but with different parameter vectors $\theta_j \in \mathbb{R}^M$. If we have independent and identically distributed data $(y_1, \cdots, y_n)$ from this mixture, we can easily compute the MLE using the EM algorithm. The missing data are z_{ij} which are the indicators of membership of each data point, y_i, in the j-th component, i.e.

$$z_{ij} = \begin{cases} 1 & , \quad y_i \in j \\ 0 & , \quad y_i \notin j \end{cases}$$

The complete data log-likelihood is given as

$$\text{CDLL} = \sum_{i=1}^{n} \sum_{j=1}^{C} z_{ij} \log g_j(y_i; \theta_j),$$

from which the Q function, Eq. 5, can be easily computed as

$$Q(\theta;\theta^{(m)}) = \sum_{i=1}^{n}\sum_{j=1}^{C} \hat{\pi}_{ij}^{(m)} \log g_j(y_i;\theta_j), \tag{62}$$

where

$$\hat{\pi}_{ij}^{(m)} = \frac{p_j^{(m)}\, g_j(y_i;\theta_j^{(m)})}{\sum_{j=1}^{C} p_j^{(m)}\, g_j(y_i;\theta_j^{(m)})}. \tag{63}$$

Now this allows us to update the parameters of each component, $\theta_j^{(m)}$, by separately maximizing the log-likelihoods of the components by solving the maximum likelihood equations

$$\sum_{i=1}^{n} \hat{\pi}_{ij}^{(m)} \frac{\partial \log g_j(y_i;\theta_j)}{\partial \theta_j^{(m)}} = 0\,,\ j = 1,\cdots,C, \quad m = 1,\ldots,M \tag{64}$$

i.e. the maximum likelihood equations for estimating the parameters are the weighted average of the individual maximum likelihood equations arising from each component considered separately, where the weights are the probabilities of membership of y_i in each component. The updates for the mixing proportions are obtained as the mean of $\hat{\pi}_{ij}$ over all i, i.e.,

$$p_j^{(m+1)} = (1/n)\sum_{i=1}^{n} \hat{\pi}_{ij}^{(m)}. \tag{65}$$

Thus, Eq. 65 and the system of equations (64) constitute the EM algorithm for finite mixtures.

For the test problem, we chose the famous data from The London Times during the years 1910-1912 reporting the number of deaths of women 80 years and older [**?**]. The tabulation was in terms of the number of days, n_i, in which y_i deaths occurred (see Table 1). A two-component mixture of Poisson distribution provides a good fit to the data, whereas a single Poisson distribution had a poor fit. This is possibly due to different patterns of death during the winter and summer.

In this problem, $y_i = i$, so that we can write the likelihood as

$$\prod_{i=0}^{9} \left[pe^{-\mu_1}\mu_1^i/i! + (1-p)e^{-\mu_2}\mu_2^i/i!\right]^{n_i}.$$

Table 1: Initial Guess 1: MOM estimate from Lange

Deaths, y_i	Frequency, n_i	Deaths, y_i	Frequency, n_i
0	162	5	61
1	267	6	27
2	271	7	8
3	185	8	3
4	111	9	1

Here, we have to estimate three parameters, $\theta = (p, \mu_1, \mu_2)$. The EM algorithm is as follows:

$$p^{(m+1)} = \frac{\sum_i n_i \, \hat{\pi}_{i1}^{(m)}}{\sum_i y_i},$$

$$\mu_1^{(m+1)} = \frac{\sum_i i \, n_i \, \hat{\pi}_{i1}^{(m)}}{\sum_i y_i \, \hat{\pi}_{i1}^{(m)}},$$

$$\mu_2^{(m+1)} = \frac{\sum_i i \, n_i \, \left(1 - \hat{\pi}_{i1}^{(m)}\right)}{\sum_i y_i \left(1 - \hat{\pi}_{i1}^{(m)}\right)},$$

where

$$\hat{\pi}_{i1}^{(m)} = \frac{p^{(m)} \left(\mu_1^{(m)}\right)^i e^{-\mu_1^{(m)}}}{p^{(m)} \left(\mu_1^{(m)}\right)^i e^{-\mu_1^{(m)}} + (1 - p^{(m)}) \left(\mu_2^{(m)}\right)^i e^{-\mu_2^{(m)}}}.$$

The MLEs for the parameters are: $(p, \mu_1, \mu_2) = (0.3599, 1.256, 2.663)$. The EM is very slow to converge for this problem, regardless of the starting value, because the data does not contain adequate information to clearly separate the two components. This can be seen by examining the eigenvalues of the Jacobian matrix of the EM mapping at the MLE, i.e. the eigenvalues of $J(\theta^*)$, which are computed as 0.9957, 0.7204 and 0. The extremely slow convergence of the EM is well explained by the fact that the largest eigenvalue is very close to 1. Tables 2 and 3 report the performance of various acceleration schemes in relation to the EM algorithm, for two different starting values. The first one was the method of moments estimate, $x_0 = (0.2870, 1.101, 2.582)$, reported by Lange (1995), and the second set of starting values was, $x_0 =$

(0.3,1.0,2.5). We can observe that the Squared MPE1 method had the best performance. It accelerated EM by a factor of around 6 to 7. The other two SQUAREM methods also provided acceleration by a factor of about 4. Among the one-step schemes, the RRE1 scheme did not accelerate the EM whereas the MPE1 showed a modest gain. It also worth noting that the one stage RRE1 scheme exhibited major numerical problems due to stagnation and had to be restarted numerous times.

Table 2: Initial Guess 1: MOM estimate from Lange

	EM	MPE1	RRE1	SqMPE1	SqRRE1	SqHyb1
fevals	2045	1986	1242	308	584	462
restarts	0	0	308	0	1	0
log-lik	−1989.95	−1989.95	−1994.05	−1989.95	−1989.95	−1989.95

Table 3: Initial Guess 2: $p_0 = (0.3, 1.0, 2.5)$

	EM	MPE1	RRE1	SqMPE1	SqRRE1	SqHyb1
fevals	2056	1800	2342	244	572	268
restarts	0	0	363	0	0	0
log-lik	−1989.95	−1989.95	−1994.05	−1989.95	−1989.95	−1989.95

The EM algorithm naturally satisfies constraints on mixing proportions. Generally, this is not the case for the extrapolation schemes, and consequently the updated parameters can lie outside of the allowable intervals. Transforming parameters such that they range over the entire real line is a useful device which not only handles the issue of parameter constraints, but, often, it also improves the convergence of the numerical algorithms, as seen in our example. Parameter transformation is particularly useful and effective for parameters that have a narrowly defined range of allowable values, such as in the case of mixing proportions which can only lie in the unit interval. Tables 3 and 4 report interesting results on the performance of the schemes after a simple parameter transformation, where the mixing proportion p was

transformed such that $p' = \log\frac{p}{1-p}$. We observe several interesting results concerning parameter transformation. It dramatically improved rate of convergence of the squared extrapolation methods, by factors ranging from 4 to 10. The largest improvement was seen for the reduced rank extrapolation methods. Interestingly, the one stage reduced rank extrapolation schemes, RRE1, exhibited the most dramatic improvement in performance, both in terms of faster convergence and in eliminating numerical problems due to stagnation. Parameter tranformation did not significantly alter the convergence of the one stage MPE1 scheme. There is no impact on the convergence of EM. It was already demonstrated in Section 3.1 that the EM is invariant to homemorphic transformations. The small differences in the number of fevals, when we compare Table 2 with Table 4 and Table 3 with Table 5, are actually due to the difference in stopping criteria. We used the same stopping criterion, $\epsilon = 10^{-7}$, for both original and the transformed EM iterations. Let $z_{n+1} = G(z_n)$, be the EM mapping in transformed parameter space, where G is defined by $T^{-1}FT$, and let z^*, be the fixed point of G. Since $x_n = T(z_n)$ and $\|x_n - F(x_n)\| = \|T(z_n) - T(G(z_n))\|$, the stopping criterion for the transformed EM iterations, z_n, is approximately equal to the stopping criterion on x_n multiplied by a factor of $\det\left[\partial T(z^*)\right]$, where $\partial T(z^*)$ is the Jacobian of the transformation T, evaluated at z^*, given by $T(z^*) = x^*$. We can obtain $z^* = (-0.575, 1.256, 2.663)$. The parameter transformation is $T : z \mapsto x$ is given by

$$x^{(1)} = \left(1 + \exp(-z^{(1)})\right)^{-1}, x^{(2)} = z^{(2)}, x^{(3)} = z^{(3)}.$$

Hence,

$$\det\left[\partial T(z^*)\right] = \left(\exp(z^{*(1)}/2) + \exp(-z^{*(1)}/2)\right)^{-2} = 0.230.$$

Thus, the stopping criterion on z_n is more stringent than that on x_n, explaining why the number of fevals in Tables 4 and 5 are larger than the corresponding ones in Tables 2 and 3.

6.2 von Mises Mixtures

The von Mises distribution is often used to describe unimodal data on the circumference of a circle. Circular data arise in many applications such as (i) diurnal pattern of adverse events (e.g, myocardial infarctions, death), (ii) seasonal variations of adverse events (e.g., certain types of cancers, suicides), and (iii) the study of hydrologic processes, e.g. amounts of rainfall per month

Table 4: Initial guess 1, with parameter transformation

	EM	MPE1	RRE1	SqMPE1	SqRRE1	SqHyb1
fevals	2211	1482	212	46	72	94
restarts	0	0	0	0	0	0
log-lik	−1989.95	−1989.95	−1989.95	−1989.95	−1989.95	−1989.95

Table 5: Initial guess 2, with parameter transformation

	EM	MPE1	RRE1	SqMPE1	SqRRE1	SqHyb1
fevals	2223	1736	212	40	46	86
restarts	0	0	0	0	0	0
log-lik	−1989.95	−1989.95	−1989.95	−1989.95	−1989.95	−1989.95

or monthly evaporation from a reservoir. von Mises distribution is often used an exploratory device to test simple hypotheses concerning the presence of a dominant peak in such data, where the location/timing of the peak may indicate a plausible link between the events and a putative cause. The von Mises distribution is given by

$$g(y;\mu,\kappa) = \frac{1}{2\pi I_0(\kappa)}\exp(\kappa(y-\mu)), \tag{66}$$

where y is a variable distributed on the circle, i.e. $y \in (0, 2\pi)$, and the parameters indexing the distribution, $\mu \in (0, 2\pi)$ and $\kappa > 0$, are the location and concentration parameters, and $I_0(.)$ is the modified Bessel function of the zeroth order. Parameter κ is called the concentration parameter since the peakedness of the von Mises density is directly related to its value, and the extreme value of $\kappa = 0$ corresponds to the uniform density $1/2\pi$.

It is not so uncommon for the circular data to be bimodal, indicating either the presence of two different processes or that the same process is operating at two intensities at different times. For example, the monthly runoff in a watershed might have two dominant sources, one from rainfall in the Fall and the other from the snowmelt in the Spring. In such cases,

a two-component mixture of the von Mises distribution, given below, may provide a better and more flexible description of the data:

$$g(y; p, \mu_1, \kappa_1, \mu_2, \kappa_2) = \frac{p}{2\pi I_0(\kappa_1)} \exp(\kappa_1(y-\mu_1)) + \frac{1-p}{2\pi I_0(\kappa_2)} \exp(\kappa_2(y-\mu_2)). \tag{67}$$

Let $(y_1, \cdots, y_n)$ be the observed directions in radians. The MLEs of the mixture parameters can be obtained using Eqs. 64 and 65, using the MLEs for the von Mises distribution, Eq. 66, as

$$\hat{\mu_j} = \arctan(\bar{S}_j/\bar{C}_j),\ j = 1, 2, \tag{68}$$

and $\hat{\kappa}_j$ are the solutions to

$$I_1(\kappa) - A_j\, I_0(\kappa) = 0, j = 1, 2, \tag{69}$$

where

$$\bar{C}_j = \frac{1}{n}\sum_{i=1}^{n} \hat{\pi}_{ij} \cos y_i, \quad \bar{S}_j = \frac{1}{n}\sum_{i=1}^{n} \hat{\pi}_{ij} \sin y_i,$$

and

$$A_j = \sqrt{\bar{C}_j^2 + \bar{S}_j^2}$$

and $\hat{\pi}_{ij}$ is given by Eq. 63. Thus, in order to obtain the MLE of the mixture, we need to twice solve the transcendental equation 69, which can be easily done using a uniroot finder such as the Newton-Raphson scheme. Alternatively, accurate approximations based on aymptotic expansions are also available for $\hat{\kappa}$. Dobson [9] provides a number of them, each accurate over a specific interval. In particular, the following approximation provides the uniformly most accurate (i.e. smallest maximum relative error) solution over the entire range of A values:

$$\hat{\kappa} = (1.28 - 0.53A^2) \tan(\pi A/2).$$

It is interesting to note that for the von Mises distribution the MLE for the location parameter, $\hat{\mu}$, is independent of the MLE for the concentration parameter $\hat{\kappa}$. This implies that the vector length, A, of a sample of size n from a von Mises distribution is sufficient for the estimation of the concentration parameter κ, when the mean direction μ is unknown (see [36] for a proof of this). This is analogous to the independence between the MLEs for mean and

variance for the normal distribution, where the marginal distribution of sample variance s^2 is sufficient for the variance σ^2, in the absence of information about the mean.

We test the extrapolation schemes using data simulated from a two component von Mises mixture with parameters

$$(p, \mu_1, \kappa_1, \mu_2, \kappa_2) = (0.75, \pi/2, 0.8, 3\pi/2, 1.6).$$

We simulated 200 data sets and Tables 6 and 7 provide a summary of the performance of various numerical schemes, without and with parameter transformation, respectively. In addition to transforming the mixing proportion using a logit transformation, we also transformed the concentration parameter using a log transform. Parameter transformations not only improved the rate of convergence of the squared extrapolation schemes significantly (by a factor of 2 or more), as in the case of Poisson mixtures, but they also helped in keeping the parameters within constraints. Once again, the parameter transformation did not influence the EM at all, since the Jacobian matrix is invariant to one-to-one parameter transformations. Also, the MPE1 scheme behaved much like the EM and showed only a slightly faster rate of convergence. The RRE1 shows a faster rate of convergence than the EM, if we look the 1st quartile and median, but in a number of simulations, it also exhibited problems due to poor convergence and stagnation. The SqMPE1 scheme was clearly the fastest. When it worked well (looking at the median), it accelerated the EM by a factor of 6 to 18. However, it also had near breakdown problems in several simulations. The SqHyb1 is clearly the best numerical scheme overall, for this problem, considering both the speed of convergence and the ability to overcome stagnation and avoid near breakdown.

We also note the large number of failures for many of the extrapolation methods. A failure can occur for three reasons, (1) exceeding the maximum limit for the number of evaluations of F , which is set at 10000, (2) exceeding the maximum number of restarts, which is set at 100, and (3) the scheme blowing up due to inadmissible or unacceptably large values of some of the parameters. The first type of failure seldom occurs in the SQUAREM schemes. Only the EM and the first order one-stage extrapolation (although this is rare) methods exhibit near-sublinear convergence. The RRE1 schemes, both the one-stage and squared methods are susceptible to the second type of failure, due to stagnation. When the scheme stagnates, it is restarted using the base EM iteration. However, failure is said to occur, if after a pre-specified

number of restarts there is no progress made in terms of reduction of the residual norm. The third type of failure occurs in the MPE1 schemes, due to numerical instability called near breakdown, in which errors are magnified with each cycle, until the scheme blows up.

Table 6: Results for VM mixture - without transformation. The numbers in the first 4 columns denote the number of evaluations of the fixed point mapping F.

	1st quartile	median	mean	3rd quartile	# failures
EM	656	1030	1343	1403	1
MPE1	508	840	1393	1330	11
RRE1	138	340	3822	10000	43
SqMPE1	132	171	184	235	20
SqRRE1	206	388	889	878	2
SqHyb1	148	209	247	289	2

Table 7: Results for VM mixture - with transformation. The numbers in the first 4 columns denote the number of evaluations of the fixed point mapping F.

	1st quartile	median	mean	3rd quartile	# failures
EM	697	958	1500	1442	0
MPE1	577	800	1235	1148	0
RRE1	138	212	2721	5656	23
SqMPE1	44	56	75	83	35
SqRRE1	52	86	1894	1994	15
SqHyb1	56	80	284	184	2

6.3 Latent Class Analysis

The essential idea in latent class modeling is that the observed associations between a set of D dichotomous variables are generated by the presence of different *latent* classes within which the variables are independent. A

latent class model may be formulated as a mixture by supposing that a random vector $\mathbf{y} = (y_1, \cdots, y_D)$ of dichotomous variables, arising from a latent structure, has a probability density function given by

$$g(\mathbf{y}; p, \Theta) = \sum_{j=1}^{C} p_j \, g_j(\mathbf{y}; \theta_j), \tag{70}$$

where $\Theta = (\theta_1, \cdots, \theta_j)$ represents all the parameters and $\theta_j = (\theta_{j1}, \cdots, \theta_{jD}$ represents those of the c-th component, and

$$g_j(\mathbf{y}; \theta_j) = \prod_{k=1}^{D} \theta_{jk}^{y_k} \, (1 - \theta_{jk})^{1-y_k}. \tag{71}$$

The parameters $\theta_{jk}, j = 1, \cdots, C, k = 1, \cdots, D$, give the probability that the k-th variable is present (has the value of 1) in the j-th class. On each unit i, we observe the vector $\mathbf{y}_i$. The MLEs of the parameters p_j and θ_j are obtained by solving the maximum likelihood equations for a finite mixture as before:

$$p_j^{(m+1)} = (1/n) \sum_{i=1}^{n} \hat{\pi}_{ij}^{(m)} \tag{72}$$

$$\theta_j^{(m+1)} = \frac{1}{n \, p_j^{(m)}} \sum_{i=1}^{n} \mathbf{y}_i \hat{\pi}_{ij}^{(m)}, \tag{73}$$

where

$$\hat{\pi}_{ij}^{(m)} = \frac{p_j^{(m)} \, g_j(\mathbf{y}_i; \theta_j^{(m)})}{\sum_{j=1}^{C} p_j^{(m)} \, g_j(\mathbf{y}_i; \theta_j^{(m)})}. \tag{74}$$

We test our extrapolation schemes on the simulation example presented in Everitt (1975) [11]. We simulated 200 observations from the model given by, Eq. 70, with $C = 3, D = 5$, and the parameter values as:

$$\begin{aligned} p &= (1/3, 1/3, 1/3) \\ \theta_1 &= (0.5, 0.5, 0.2, 0.3, 0.1) \\ \theta_2 &= (0.3, 0.2, 0.7, 0.6, 0.4) \\ \theta_3 &= (0.9, 0.7, 0.5, 0.1, 0.7). \end{aligned}$$

Thus, we need to estimate 17 parameters in this problem.

Table 8: Summary of LCA analysis simulation results - 200 simulations

	1-st quartile	median	mean	3rd quartile	# failures
EM	586	1147	1652	2504	6
MPE1	613	1106	1749	2150	6
RRE1	203	415	652	966	40
SqMPE1	30	42	959	134	9
SqRRE1	32	41	121	72	0
SqHyb1	30	40	155	59	2

The interquartile range (IQR) for the number of *fevals* for the EM was (586,2504), with the mean of 1652. Among the one stage methods, RRE1 was clearly superior. Its IQR was (203,966) with a mean of 652. The MPE1 scheme was no better than the EM. The performance of all the SQUAREM methods was spectacular for this problem. They typically accelerated the EM by a factor of about 20 to 25 times. The SqMPE1 scheme failed 9 times and all of them were due to exceeding the maximum limit of 5000 on *feval*. The SqRRE1 did not encounter any numerical difficulties due to stagnation, and the SqHyb1 twice exceeded the maximum limit of 5000 on the number of *fevals*.

6.4 Principal Stratification Models in Causal Inference

Causal inference can be broadly viewed as the evaluation of the impact of programs, policies, and treatments. When the information that forms the basis of an evaluation is obtained in a carefully controlled setting, inferring and estimating the *effect* of the treatment is straightforward, and one only needs to account for random variations. This is the situation in randomized clinical trials, where each participant is randomized to receive one of the treatments. However, when information on the factors determining the receipt of the treatment, that may also have an impact on the outcome, is unavailable, the evaluation is difficult, if not impossible. For example, it may be the case that the units who received a particular treatment are healthier, on average, than those who did not receive that treatment, in a manner that can not be readily characterized. Consequently, comparisons of outcomes between those who did and did not receive the treatment could be misleading.

This is typically the situation in observational studies in fields as diverse as Economics and Epidemiology, where usually the study investigators have no direct control of the receipt of the treatment by units in the study. There are many studies, lying between these two extremes, where the study investigators control, not the receipt of the treatment itself, but some factors that may affect the receipt of treatment. This situation has been termed *partial control* by Frangakis, Brookmeyer, Varadhan et al. [14]. As in any problem of causal inference, inference in problems with partial control of treatment require assumptions, particularly on latent variables, which do not permit direct observation, but play an important role in determining the treatment received. Since the concept of *latency* is central to causal inference, it is natural that the EM algorithm plays an important role in these problems.

An example of a partially controlled situation arose in the evaluation of the impact of Baltimore's needle exchange program (NEP) in reducing the HIV transmission among injection drug users [14]. In this study, the investigators determined the location of the NEP sites, to which the injection drug users could go and exchange on a one-to-one basis used needles for new, sterile ones. Generally, proximity to the NEP site affects both exchange behavior and the provision of serum samples for monitoring HIV status. Therefore, the location of the NEP sites can affect the exchange behavior and the ability to ascertain HIV status. However, the study did not directly and fully control the exchange at NEP (e.g., randomizing participants to exchange needles or not). Frangakis et al. [14] developed an approach based on a novel inference strategy called *principal stratification* [13] to evaluate the causal effect of the controlled factor, which in our example is the placement of the NEP sites, on the HIV transmission, that is attributable to the uncontrolled treatment, which in our example is the exchange of needles.

The main idea in principal stratification is the construction of a latent variable called principal stratum, which is an essential attribute of each study unit. A principal stratum is defined by the joint potential values of the partially controlled factor over the range of values of the directly controlled factor. In the NEP study, the principal stratum for a study unit is a vector of binary indicator variables denoting the exchange behavior of that unit for all possible values of NEP distance, the directly controlled variable. If we dichotomize distance as "near" and "far", then the principal stratum for each study unit is a vector with two values, and there are four such possible vectors, $s_1 = (0,0), s_2 = (1,0), s_3 = (0,1)$, and $s_4 = (1,1)$. A unit with attribute s_1, will not exchange at NEP regardless of where the site is located,

and a unit with attribute s_2 will exchange if the site is near, but not otherwise. The principal stratum is a latent variable, since in the actuality each unit can experience only one of the two possible values of the directly controlled factor, and therefore, we can only observe the exchange behavior corresponding to that value of the controlled factor. Principal stratum is an attribute of a study unit that is not affected by the controlled factor, and hence can be used like any other pre-treatment covariate. Furthermore, contrasts of "potential" outcomes at different values of the controlled variable, within a principal stratum, are causal effects, which can be used to evaluate the effect of the controlled treatment (e.g., distance to NEP site) on the outcome that is attributable to the uncontrolled factor (e.g., needle exchange).

The principal stratification model for the NEP study has four components:

1. A model for the principal stratum, specified in terms of baseline covariates, time, current exchange status, and past history (this may include past values of time-varying covariates, NEP distances, and exchange behavior).

2. A model for the controlled factor, e.g., NEP distance. However, since the factor is controlled, its distribution across principal strata is the same, and hence does not contain any parameters.

3. A model for the censoring indicator, i.e. whether or not a unit provides serum samples for outcome ascertainment, specified in terms of baseline covariates, time, past history, current exchange status, and principal stratum.

4. A model for the outcome variable, i.e. the HIV status of the unit, specified in terms of baseline covariates, time, current exchange status, past history, principal stratum and distance, contingent upon the unit being at risk.

The observed data for a unit i (a person j at time t) at risk are past history H_i, current distance D_i and exchange status E_i, censoring indicator C_i, and outcome Y_i (if $C_i \neq 1$). The unobserved data is the principal stratum, S_i, to which the unit belongs. We can write the models as follows:

$$Pr(S_i|H_i) = f^{(s)}(S_i|H_i;\beta^{(s)}) \tag{75}$$

$$Pr(D_i|H_i, S_i) = f^{(d)}(D_i|H_i), \quad \text{free of parameters} \tag{76}$$

$$Pr(C_i|H_i, S_i) = f^{(c)}(C_i|D_i, H_i, S_i; \beta^{(c)}) \tag{77}$$

$$Pr(Y_i|D_i, H_i, C_i, S_i) = f^{(y)}(Y_i|D_i, H_i, S_i; \beta^{(y)}), \tag{78}$$

where the last equality assumes ignorability of censoring.

From these four probability models, we can write the likelihood for the observed data as follows:

$$\begin{aligned}
&\prod_i \Pr(D_i, E_i, C_i, \{Y_i, \text{if } C_i \neq 1\}|H_i, \boldsymbol{\beta}) \\
&= \prod_i \Pr(D_i|H_i)\Pr(E_i, C_i, , \{Y_i, \text{if } C_i \neq 1\}|H_i, D_i, \boldsymbol{\beta}) \\
&= \prod_i \sum_s \Pr(S_i = s, E_i, C_i, \{Y_i, \text{if } C_i \neq 1\}|H_i, D_i, \boldsymbol{\beta}) \\
&= \prod_i \sum_{s_i} \Pr(S_i = s|H_i, D_i, \beta^{(s)})\Pr(C_i|S_i = s, H_i, D_i, \beta^{(c)})\Pr(Y_i|S_i = s_i, H_i, D_i, \beta^{(y)})^{C_i}),
\end{aligned}$$

where s_i is the set of principal strata for which the exchange status is E_i when the distance is D_i, and $\boldsymbol{\beta} = (\beta^{(s)}, \beta^{(c)}, \beta^{(y)})$ is the vector of all the parameters. Thus we see that the principal stratification model is a mixture model, where the weights are determined by the distribution of the latent principal strata. We can write the likelihood for the complete data (which is observed data + S_i) as

$$\begin{aligned}
L_c &= \prod_{s_i} \left\{ f^{(s)}(s|H_i, \beta^{(s)})\, f^{(c)}(C_i|D_i, S_i = s, H_i, \beta^{(c)})\, \left[f^{(y)}(Y_i|H_i, D_i, S_i = s, \beta^{(y)})\right]^{C_i} \right\}^{\mathbf{1}\{s_i = s\}} \\
&= \prod_{s_i} g_i(s; \boldsymbol{\beta})^{\mathbf{1}\{s_i = s\}},
\end{aligned}$$

from which we obtain the distribution of the principal stratum conditional on observed data as

$$\begin{aligned}
\pi_i(s, \boldsymbol{\beta}) = \Pr(S_i = s|\text{observed data}) &= \frac{g_i(s; \boldsymbol{\beta})}{\sum_{s_i} g_i(s; \boldsymbol{\beta})}, \quad \text{if } s \in s_i \\
&= 0 \quad \text{, otherwise.}
\end{aligned}$$

Now we can compute the E step, at the $(m+1)$-th iteration, as the expectation of L_c conditioned on the observed data, as

$$E\left[\log L_c|\text{observed data}\right] = \sum_s \pi_i(s, \boldsymbol{\beta}_m) \left[\log f^{(s)}(s|H_i, \beta^{(s)}) + \log f^{(c)}(C_i|D_i, s, H_i, \beta^{(c)}) + C_i \log f^{(y)}(Y_i|D_i, s, H_i, \beta^{(y)})\right], \quad (79)$$

where $\boldsymbol{\beta}_m$ is the vector of parameter values from the m-th iteration. This function needs to be maximized with respect to $\boldsymbol{\beta} = (\beta^{(s)}, \beta^{(c)}, \beta^{(y)})$ to obtain the parameters for the next iteration. This is the M step, which is easily performed by noting that the maximization simplifies to three separate maximizations corresponding to the models for S,C, and Y, using as weights, $\pi_i(s, \boldsymbol{\beta})$, for each principal stratum. In particular, in the NEP study, S is an ordinal variable, and therefore, a proportional odds logistic regression model was used (the R function *polr* from the MASS library was employed to fit the model); C and Y are both binary variables and therefore logistic regression models were used for them (using the R function *glm* with logit link). Starting values for the parameters were obtained by assigning equal weights to all the principal strata, and using that in the calls to the *polr* and *glm* functions to compute starting values for β. The R functions for computing the maximum likelihood estimates for longitudinal studies, in which the treatment is partially controlled, using the principal stratification models, are provided in a software package called "PSpack" [15].

The details of the study design and data characteristics are available in [14]. The results of the EM algorithm are presented in Table 9. The convergence history, in terms of residual norm, is shown in Figure 2 for various numerical schemes. Once again we see the clear superiority of the SQUAREM schemes over the EM and the one-stage schemes. The SQUAREM methods accelerate the EM by a factor of 5. The SqHyb1 scheme is slightly faster than the other two SQUAREM methods. Both the E and the M steps are time consuming for this problem. The EM took nearly 40 minutes to reach convergence, whereas the SqHyb1 scheme only took about 7 minutes. The computations were performed on a Windows platform using a 3.2 GHz Pentium 4 processor with 2 GB RAM. We used a convergence criterion of 10^{-7} on the residual norm.

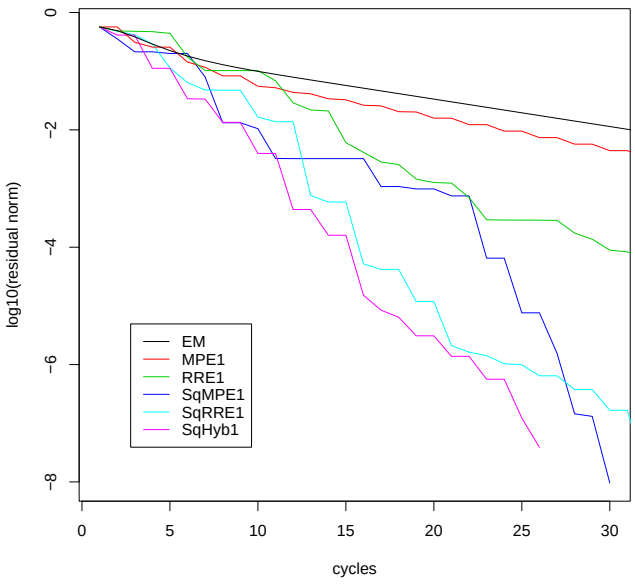

Figure 2: History of residual error norm for different numerical schemes, plotted as a function of cycles, where each cycle consists of two fevals.

6.5 Image Reconstruction in PET

Positron emission tomography (PET) is a very useful medical technology for imaging the activity of an organ, e.g. brain, with the goal of detecting regions of abnormal activity, e.g. tumors. A person is administered a dose of a positron emiting substance (e.g., glucose, tagged with a radioactive isotope), which is deposited in different regions of brain in amounts proportional to the glucose uptake mechanism of that region. Each emitted positron from the susbtance annihilates with a nearby electron, producing two photons that travel in (nearly) opposite directions. An array of scintillation detectors placed around the head of the person record the cylindical volume in which each emission occurs. The pair of photons emitted at the same instant are detected in coincidence by a pair of detector elements, defining a cylindrical volume called a *detector tube*. Typically, several hundreds of thousands of coincident emissions can be detected in a few minutes. We observe which pair of detectors recorded the coincident photon emissions, but we do not know

Table 9: Results of the principal stratification modeling for the NEP evaluation

	EM	MPE1	RRE1	SqMPE1	SqRRE1	SqHyb1
fevals	273	218	126	58	62	50
restarts	0	1	0	1	0	0
log-lik	-5461.851	-5461.851	-5461.851	-5461.851	-5461.851	-5461.851

the exact location in the brain from which the positron was emitted. Vardi, Shepp, and Kaufman [43] developed a relatively simple statistical model for describing the PET data, based on the physics of positron emission and on the geometry of the PET apparatus. We provide a simple presentation of the VSK model, and refer readers to that paper for a more detailed description.

The data collected in a PET scan is the vector of tube counts $(y_1, \cdots, y_D)$, where y_i is the number of coincident photon emissions detected in detector tube i, and D is the total number of tubes. Only a small fraction of the emitted photons are detected by the tubes. The rest either do not intersect the detector tubes or are attenuated by the body. The image reconstruction problem is to obtain the spatial distribution of the photon emission intensities, $\lambda(\mathbf{x})$, where $\mathbf{x}$ is a point in the region $B \subset \mathbb{R}^3$, that defines the brain. The region B is usually discretized into small square grids called *pixels* and the emission intensities are estimated at the center of each pixel.

The statistical model for discretized PET problem is

$$y_i \sim \text{Poisson}\left(\sum_{j=1}^{S} c_{ij}\lambda_j\right), \quad i = \cdots, D, \tag{80}$$

where j denotes a pixel in the brain, S is the total number of pixels, c_{ij} is the probability of an emission from pixel j being detected by tube i, and λ_j is the unknown emission intensity of pixel i. Although, λ_j are intensities, i.e. counts per unit area per unit time, which must be multiplied by the area of the pixel and the time of observation to obtain the counts, we will use the same symbol to actually refer to the counts rather than the intensities. This does not present a problem since each pixel will be of the same size and the time of observation will be the same for all the pixels. The transition matrix, C,

whose elements are $c_{ij}, i = 1, \cdots, D$, $j = 1, \cdots, S$, are assumed to be known exactly, and can be determined, under certain simplifying assumptions, based on the geometry of the brain and the detector configuration.

We can think of this problem as one of estimating the sums, z_{+j}, $j = 1, \cdots, S$, of each column of a $D \times S$ matrix, Z, given data on its row sums, $y_i = z_{i+}$, $i = 1, \cdots, D$. Thus, it is natural to view y_i as the observed (incomplete) data, and z_{ij}, which are the elements of Z, as the complete data defining the random number of emissions from pixel j that are detected by tube i. The likelihood for the complete data can be written as

$$L_c \propto \prod_j \left(\frac{e^{-\lambda_j} \lambda_j^{z_{+j}}}{z_{+j}!} \prod_i c_{ij}^{z_{ij}} \right), \tag{81}$$

which expresses the fact that the likelihood of detection by a tube i of a photon emitted at pixel j is simply a product of the probability of emission at pixel j times the probability of detection by tube i, conditional upon emission at pixel j. This can be rewritten as

$$L_c \propto \prod_j \prod_i (\lambda_j \, c_{ij})^{z_{ij}} \, e^{-\lambda_j \, c_{ij}}, \tag{82}$$

since $z_{+j} = \sum_i z_{ij}$, and $\sum_i c_{ij} = 1$, $\forall \, j$.

The log likelihood of complete data is, then, given as

$$\log L_c = \text{constant} + \sum_j \sum_i z_{ij} \log(\lambda_j \, c_{ij}) \; - \; \lambda_j \, c_{ij}. \tag{83}$$

To obtain the EM algorithm, we, first, compute the expectation of $\log L_c$, conditional upon observed data, y_i, and given $\lambda_j^{(m)}$. Given observed data, y_i, and $\lambda_j^{(m)}$, we can obtain the expectation of complete data as follows

$$z_{ij}^{(m)} := E\left[z_{ij} | y_1, \cdots, y_D, \lambda_j^{(m)} \right] = \frac{c_{ij} \lambda_j^{(m)}}{\sum_k c_{ik} \lambda_k^{(m)}} y_i \tag{84}$$

The conditional expectation of the unobserved z_{ij}, given in Eq. 84 can be understood as follows. The probability of a photon being detected in the i-th detector tube given that it was emitted by pixel j, is given by the fraction on the RHS of Eq. 84. Therefore, the expected value of number of photons

that were emitted by pixel j and detected in tube i is simply the product of this fraction with the number of detections in tube i.

Since the log-likelihood of the complete data, Eq. 83, is linear in the complete data, the EM algorithm simply consists of iterating between estimating the missing complete data, z_{ij}, in the E-step (Eq. 84), and the unknown parameters, λ_j, in the M-step. The M-step consists of plugging $z_{ij}^{(m)}$ into the expression for the expectation of the complete-data log-likelihood, setting the derivatives with respect to λ_j to be zero, and solving for λ_j. This yields the following EM algorithm

$$\begin{aligned} \lambda_j^{(m+1)} &= \sum_i z_{ij}^{(m)} \\ &= \lambda_j^{(m)} \sum_{i=1}^{D} \left(c_{ij} y_i \Big/ \sum_{k=1}^{S} c_{ik} \lambda_k^{(m)} \right), \; j = 1, \cdots, S. \end{aligned} \tag{85}$$

Here we simulate a PET-like image reconstruction problem with a simple geometry for the region B defining the brain. We create a brain phantom that is a square rather than the more realistic oval-shaped phantom. This difference in the geometry is inconsequential as far as the evaluation of the numerical schemes, for image reconstruction, is concerned. We are only concerned with the rate of convergence of numerical schemes for computing the fixed point, λ_j^* of the mapping defined by the iteration in Eq. 85. We take B to be a square region of size 32×32. We consider a detector ring with 32 segments, so that there are $16 \times 31 = 496$ detector tubes. Therefore, our transition matrix C has dimensions 496×1024. The elements of this matrix can be computed using geometrical arguments, however, we computed them using simulations, where we generate a randomly oriented ray from the jth pixel and determine the tube which detects the ray, say, the ith tube. We increment C_{ij} by 1. This is repeated a large number of times, say, 10000, for the same pixel. Then each element of the j-th column is divided by 10000 to obtain C_{ij}. This procedure is repeated for all the pixels. Once computed, this matrix is stored and reused whenever needed, since it is expensive to compute it on the fly. However, even the storage of this matrix is expensive and memory requirements preclude the use of very large transition matrices. For our problem, the C matrix has a little more than half a million elements, and this would require 4 MB of memory, assuming that each element require 8 bytes of memory.

In principle, the schemes should be evaluated based on how well they can reconstruct the underlying intensity field. This is not so straightforward in practice, notwithstanding the obvious limitation that we do not know the true intensity field. The main difficulty stems from the use of the maximum likelihood criterion for image reconstruction. The EM algorithm captures the main signal in the data, corresponding to the dominant features of the image, rather quickly (i.e. within 10 to 20 iterations). If the algorithm is continued further, and as the parameter estimates start "crawling up" the likelihood terrain, the image quality begins to deteriorate in the sense that it becomes more discontinuous and checkerboard-like. This is because the algorithm is now attempting to fit the noise in the data. Therefore, it is common practice in the image reconstruction literature to prematurely terminate the EM algorithm to obtain a smooth image. Such a stopping criterion is arbitrary. Hypothesis testing based approaches have been developed to establish more objective stopping criterion, but they still do not address the central issue behind the checkerboarding problem. It is not a deficiency in the EM algorithm. The algorithm is indeed doing what it is supposed to do, which is to maximize the likelihood. Checkerboarding is an unavoidable consequence of correctly solving an ill-posed problem. A principled way to avoid this and to obtain smoother images is to solve a different, hopefully, not so ill-posed (e.g., regularized), problem. In other words, we should maximize a different objective function. The most natural way to do this is to impose smoothness requirements and have a penalty for violating these requirements. This, however, is not the focus of this paper. We would like to compare and evaluate the numerical schemes in terms of their efficiency in solving the maximum likelihood estimation problem or the equivalent fixed point problem for image reconstruction, putting aside the issue of image roughness.

We compare the schemes based on simulated images. A total of 10 million photon emissions was used. We show both numerical and visual comparisons. The numerical comparisons are based on three different criteria: (1) discrepancy, $\mathcal{D}$, between observed and fitted detector tube counts, (2) residual norm, R (3) observed data log likelihood value, $\mathcal{L}$. These three criteria are defined as follows:

$$\mathcal{D} = \sum_{i=1}^{D} (y_i - \hat{y}_i^{(n)})^2, \tag{86}$$

where $\hat{y}_i^{(n)} = \sum_{j=1}^{S} C_{ij}\lambda_j^{(n)}$ is the model fitted value for the i-th detector tube

and $\lambda_j^{(n)}$ is the parameter estimates at n-th iteration,

$$R = \sum_{j=1}^{S} (\lambda_j^{(n)} - F(\lambda_j^{(n)}))^2, \tag{87}$$

where F is the implicit mapping defined by the EM iteration, Eq. 85, and

$$\mathcal{L} = \sum_{i=1}^{D} \left(y_i \log \hat{y}_i^{(n)} - \hat{y}_i^{(n)} \right) + \text{constant}. \tag{88}$$

Figures 3, 4, and 5 show the comparison among the EM and various extrapolation schemes in terms of the three criteria described above. The interesting feature to observe is the superiority of the reduced rank extrapolation schemes, both one stage, RRE1, and squared, SqRRE1, schemes. The SqRRE1 was slightly better performing than RRE1. The minimal polynomial extrapolation schemes did not fare as well. The squared hybrid scheme was slightly better than the EM and the MPE1 schemes, but not as good as the RRE1 schemes. None of the extrapolation schemes suffered from stagnation or near breakdown. Sometimes, however, they needed minor adjustments to satisfy the non-negativity constraints of the parameters, λ_j. When a parameter became negative, it was simply set to a tiny positive number. This works well for all the SQUAREM methods, since the EM, which is the base iteration scheme for the extrapolation methods, possesses the nice property of self-normalization, i.e. the λ_j always sum to the total number of photon emissions.

The evaluations presented in Figures 3, 4, and 5 are quantitative. They evaluate the schemes based on their ability to efficiently achieve some numerical criteria. It is not at all clear as to how relevant such numerical criteria are to the human visual perception and to the quality of images produced by the schemes. Hence we directly evaluate the numerical schemes based on visual comparison of images in Figures 6 through 11 . The intensity field used in the simulations is depicted in the first frame (the top left quadrant) of each figure. The other three frames in each figure correspond to the sequence of images produced by the schemes at increasing number of fevals. The "heat" colors scheme used to plot the images ranges from white to red, with white denoting areas of lowest activity (around 2000 photon emissions in our problem) and red denoting areas of highest activity (around 50000). The first frame of Figure 6 shows three patches of higher photon emission

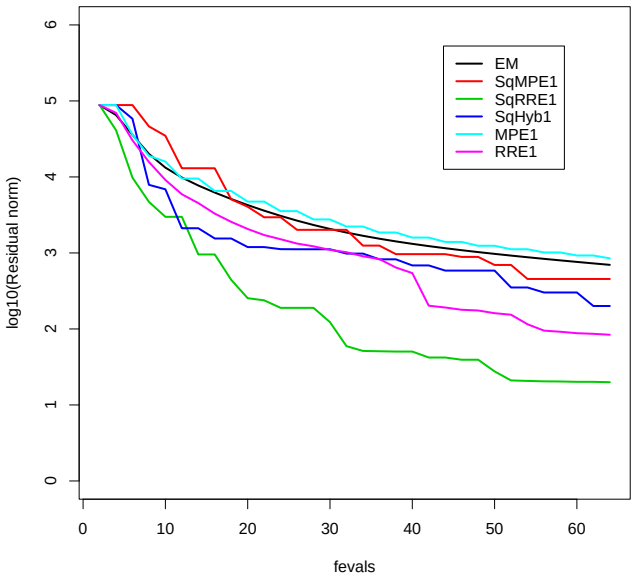

Figure 3: History of residual error norm, R, for different numerical schemes

activity. We reiterate the point made earlier about the difference between solving the mathematical problem of likelihood maximization and the practically important problem of reconstructing smoother looking images that are amenable to clearer interpretation by experts in medical diagnostics. The extrapolation methods, since they are quicker in solving the fixed point problem and the equivalent likelihood maximization probelm, tend to produce images that exhibit the checkerboarding effect sooner than the slower EM algorithm. This can be seen, for example, from a comparison of the EM in Figure 6 and the SqRRE1 in 10. We see that the checkerboarding effect does not appear in the EM images until the number of fevals is 32, whereas it appears earlier, at feval = 8, for the SqRRE1.

7 Discussion

We have proposed a new class of iterative methods for the solution of maximum likelihood estimation problem, via the EM algorithm, by treating the

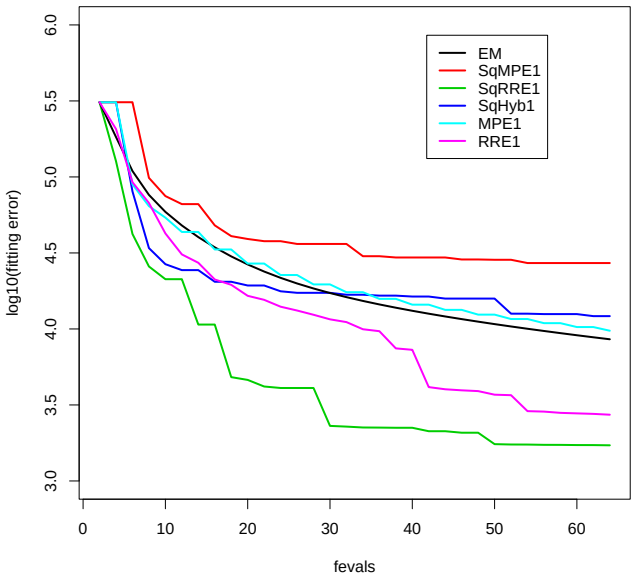

Figure 4: History of Fitting error norm,$\mathcal{D}$, for different numerical schemes

EM algorithm as a fixed point iterative scheme, $x_{n+1} = F(x_n)$. The essential idea behind these schemes is that of *squaring*, which is the two-fold application of the one-step iterative schemes of the form, $x_{n+1} = x_n + \alpha_n(F(x_n) - x_n)$. Various one step schemes are generated by choosing steplength α_n. Schemes with constant steplength α are called the *stationary* methods, and others are *nonstationary* methods. The convergence properties of stationary methods are easy to characterize, but their performance can, in general, only be enhanced by a modest amount beyond that of the fixed point iterations. The same is true for the squared version of stationary schemes, even though they are faster than the corresponding one step schemes. Therefore, we only consider nonstationary one-step iterative schemes, and their squared counterparts, for accelerating the EM algorithm.

Nonstationary one-step iterative schemes can be developed based on extrapolation methods, Eq. 25. By cycling with them, we can obtain effective, yet simple, iterative schemes for nonlinear fixed point problems. This gives rise to schemes such as minimal polynomial extrapolation (MPE) and reduced rank extrapolation (RRE), the simplest of which are the first order

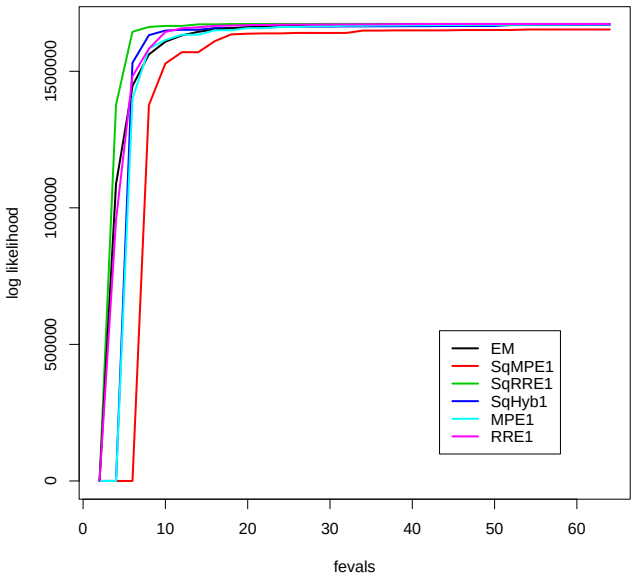

Figure 5: History of log likelihood, $\mathcal{L}$, for various numerical schemes

schemes ($k = 1$) proposed here. Such schemes are already being used in the iterative solutions of linear systems that are large and sparse ([4], [20], [40]), where they are strong competitors to the established methods such as conjugate gradient methods. Squaring the first order one-step schemes yields the first order SQUAREM schemes, that are faster with almost no extra cost. They only require one additional scalar-vector product and a vector addition compared to RRE1 and MPE1. There are very few results on the rates of convergence of extrapolation methods. It has been shown by Jbilou and Sadok [20] that the one-step extrapolation methods given by Eq. 25 are quadratically convergent when the order k of the one-step iterative scheme is equal to the degree of the minimal polynomial of the Jacobian matrix, $J(x^*)$, with respect to the vector $x_n - x_0$, provided x_0 is sufficiently close to the fixed point x^*. Sidi [39] derived useful asymptotic results (i.e., when the number of cycles n gets large) for the rate of convergence of one-step MPE and RRE schemes. Even though these results were derived for the linear fixed point problem, they may also be valid for nonlinear fixed problems. Analogous results for SQUAREM schemes do not exist, and this is a topic

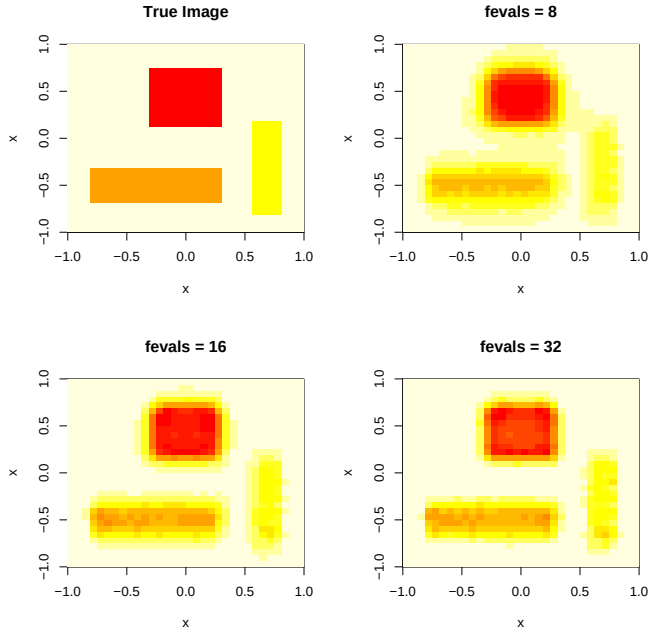

Figure 6: Comparing the true image with the sequence of images produced by the EM at feval = 8, 16, and 32

of future research. However, our empirical evidence clearly demonstrates that the SQUAREM methods significantly outperform their one-step counterparts, more than the factor of two that might be expected based on the theory.

We can derive higher order SQUAREM methods by squaring the corresponding higher order one-step extrapolation methods, with $k > 1$ in Eq. 25. For example, the second order ($k = 2$) one-step iterative schemes can be written as

$$x_{n+1} = x_n + \alpha_n(F(x_n) - x_n) + \beta_n(F^2(x_n) - 2F(x_n) + x_n), \qquad (89)$$

where α_n and β_n depend on $F^j(x_n), j = 0, 1, 2, 3$. However, squaring the second order schemes will also involve $F^4(x_n)$, and thus entails an additional function evaluation. The k-th order MPE and RRE schemes require $k + 1$ function evaluations, and their squared counterparts require $2k$ function evaluations. Thus, squaring involves an additional $k-1$ function evaluations, compared to one-step schemes. Therefore, the relative number of additional

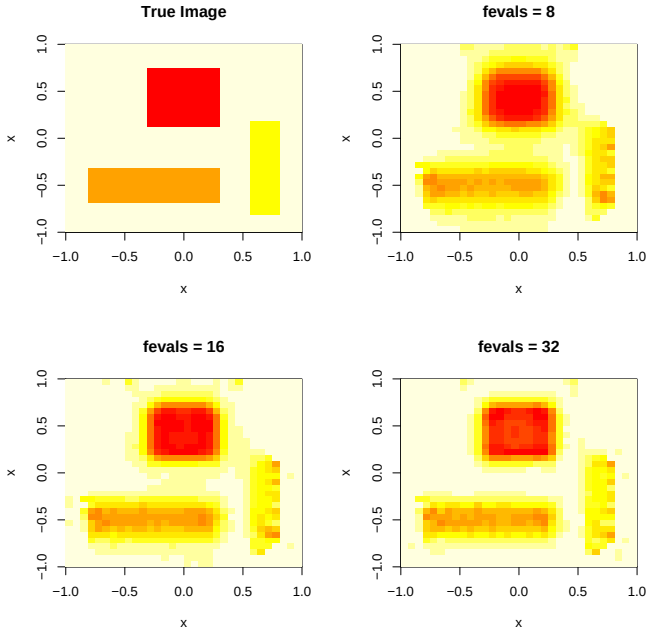

Figure 7: Comparing the true image with the sequence of images produced by MPE1 at feval = 8, 16, and 32

function evaluations is $(k-1)/(k+1)$, which is zero for first order schemes, and approaches 1 as the order increases. However, squaring may not be necessary for higher order schemes, since one-step schemes will themselves be quite fast. Evaluation of higher order extrapolation schemes is another topic of future research.

Convergence of the extrapolation schemes is non-monotone with respect to the fixed point residual norm and the log-likelihood. It is not uncommon for convergence to be erratic, where the residuals can fluctuate by several orders of magnitude. Such erratic behavior is even more pronounced for the SQUAREM methods, due to the term $\alpha_n^2 v_n$, where α_n^2, which can be large, multiplies the second order difference v_n, which is susceptible to cancellation errors as the residuals decrease. In some problems, this can limit the ultimately attainable accuracy of the scheme to be much larger than the machine epsilon. However, in scientific applications it seldom makes practical sense to demand a high level of accuracy approaching that of the machine epsilon, when the maximum attainable precision of data and other impor-

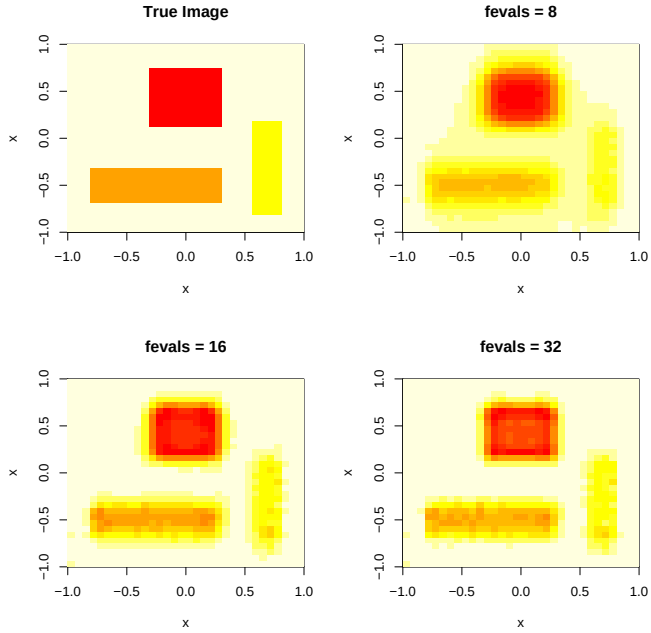

Figure 8: Comparing the true image with the sequence of images produced by RRE1 at feval = 8, 16, and 32

tant parameters is much smaller than that. The residuals can be smoothed to obtain monotonically converging sequences, but this involves additional function evaluations without improving the rate of convergence.

Extrapolation schemes can also fail due to either stagnation or near breakdown. In the case of stagnation the iterative scheme fails to make any progress and needs to be restarted. In the case of near breakdown, the numerical errors get magnified and the scheme fails when a disproportionately large step is taken and some of the updated parameters become infeasible, e.g. a negative value for the mixing proportion in a finite mixture distribution. The RRE schemes, RRE1 and SqRRE1 are susceptible to stagnation. To overcome stagnation, we have devised a simple restart strategy using the base EM iteration. It appears to be effective in the problems that we have tested so far. The SqMPE1 scheme is susceptible to near breakdown. To reduce the occurrence of near breakdowns, we have developed a new hybrid scheme which is more successful at avoiding breakdown. The hybrid scheme is also less susceptible to stagnation than the SqRRE1 scheme. Develop-

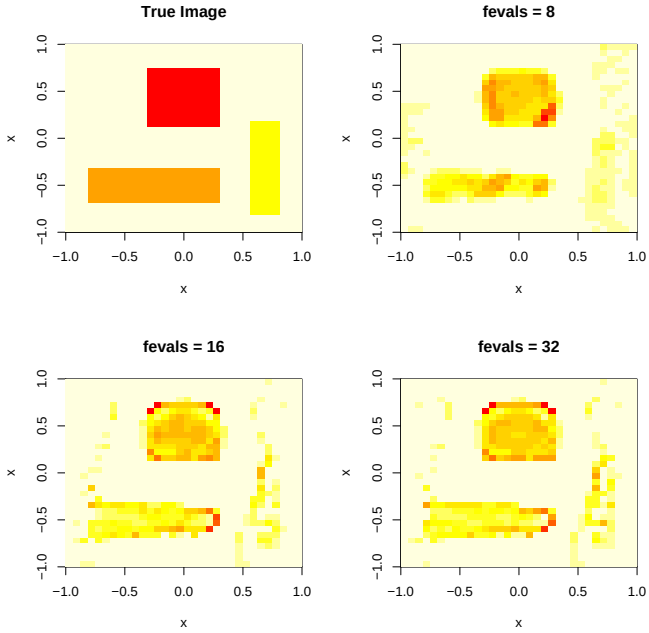

Figure 9: Comparing the true image with the sequence of images produced by SqMPE1 at feval = 8, 16, and 32

ment of strategies to overcome stagnation and to avoid near breakdown is an important area of future inquiry.

8 Summary and Conclusions

We have proposed a broad class of iterative schemes derived by cycling with the extrapolation methods for solving the fixed point problem generated by the EM approach to maximum likelihood estimation. In particular, we have focused on the simplest members of the class, the first order schemes. The first order schemes can be used as either one-step methods or SQUAREM methods. We have demonstrated using theoretical arguments and five different examples, the effectiveness of the SQUAREM methods in solving the fixed point problem of the EM approach, and the overwhelming superiority of the SQUAREM methods over the corresponding one-step schemes. We have also developed a new squared hybrid scheme, which exhibited the best overall

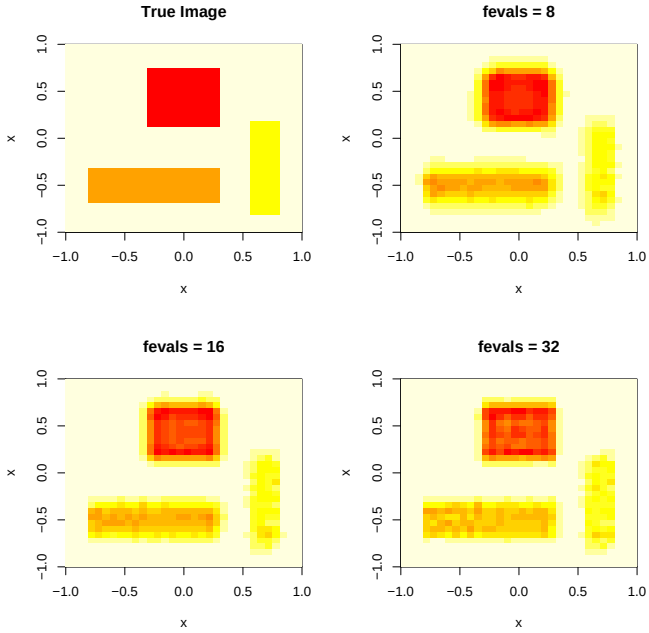

Figure 10: Comparing the true image with the sequence of images produced by SqRRE1 at feval = 8, 16, and 32

performance of all the one-step and SQUAREM schemes, in terms of speed of convergence and stability of computations. Another scheme called COMP1, has also been proposed. The results for the COMP1 scheme were not presented here, because it does not belong to the family of SQUAREM schemes, but extensive numerical experiments have shown that it is quite promising in terms of speed, and in avoiding stagnation and near-breakdown. Thus, SqHyb1 and COMP1 appear to be the best overall choices for EM acceleration.

The EM algorithm is often the only feasible method for estimating the maximum likelihood parameters in complex problems such as latent class regression models, mixed effects models, causal inference in longitudinal studies, and in large scale technological problems such as image reconstruction in PET scans, where a large number of parameters needs to be estimated. However, slow convergence of the EM severely limits the analytical capabilities, because of excessive computational times required to run simulations and other tasks which require repeated model evaluations. For such problems,

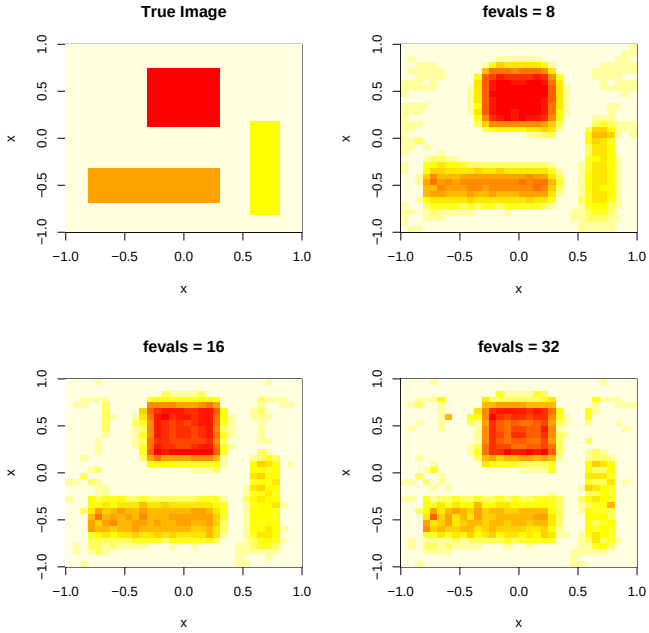

Figure 11: Comparing the true image with the sequence of images produced by SqHyb1 at feval = 8, 16, and 32

the SQUAREM methods are attractive options because of the following important attributes: (a) they are simple and only require the basic EM step, (b) they do not require the computation of auxiliary quantities such as the incomplete or complete data log-likelihood functions or their gradients, (c) because of (a) and (b) they can be implemented easily and without disturbing existing EM routines, (d) they are as broadly applicable as the EM itself, (e) they do not involve any matrix storage and/or handling, (f) in vector computing environments such as R and MATLAB, they require negligible additional effort to that of the basic EM algorithm, and (g) they converge linearly, just like the EM, but at a faster rate, where the gains can be substantial, especially in problems where the EM is very slow due to a large fraction of missing information.

Acknowledgments : We would like to thank Professor Claude Brezinski, who made our intercontinental collaboration possible. The first author would also like to thank Dr. Constantine Frangakis for the many helpful

discussions.

References

[1] Barzilai, J., Borwein, J.M. (1988), Two-Point Step Size Gradient Methods, *IMA Journal of Numerical Analysis*, 8, 141-148.

[2] Brezinski, C., and Redivo Zaglia, M. (1991), *Extrapolation Methods Theory and Practice*, North-Holland, Amsterdam.

[3] Brezinski, C., and Chehab, J.P. (1998), Nonlinear hybrid procedures and fixed point iterations, *Numerical and Function Analysis and Optimization*, 19, 465- 488.

[4] Brezinski, C. (1998), Vector sequence transformations: methodology and applications to linear systems, *Journal of Computational and Applied Mathematics*, 98, 149-175.

[5] Brezinski, C., and Chehab, J.P. (1999), Multiparameter iterative schemes for the solution of systems of linear and nonlinear equations, *SIAM Journal of Scientific Computation*, 20, 2140-2159.

[6] Brezinski, C. (2003), A classification of quasi-Newton methods, *Numerical Algorithms*, 33, 123-135.

[7] Cabay, S., and Jackson, L.W. (1976), A polynomial extrapolation method for finding limits and antilimits of vector sequences, *SIAM Journal of Numerical Analysis*, 13, 734-751.

[8] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), Maximum likelihood from incomplete data via the EM algorithm (with discussion), *Journal of the Royal Statistical Society B*, 39, 1-38.

[9] Dobson, A.J. (1978), Simple approximations for the von Mises concentration statistic, *Applied Statistics*, 27, 345-347.

[10] Eddy, R.P. (1979), Extrapolation to the limit of a vector sequence, In *Information Linkage Between Applied Mathematics and Industry*, P.C.C. Wang (ed.), 387-396, Academic Press, New York.

[11] Everitt, B.S., and Hand, D.J. (1981), *Finite Mixture Distributions*, Chapman and Hall, New York.

[12] Ford, W.F., and Sidi, A. (1988), Recursive algorithms for vector extrapolation methods, *Applied Numerical Mathematics*, 4, 477-489.

[13] Frangakis, C.E., and Rubin, D.B. (2002), Principal stratification in causal inference, *Biometrics*, 58, 21-29.

[14] Frangakis, C.E., Brookmeyer, R.S., Varadhan, R., et al. (2004), Methodology for evaluating a partially controlled longitudinal treatment using principal stratification, with application to a needle exchange program, *Journal of American Statistical Association*, 99, 239-249.

[15] Frangakis, C.E., and Varadhan, R. (2004), Systematizing the evaluation of partially controlled studies using principal stratification: from theory to practice, *Statistica Sinica*, 14, 945-947.

[16] Henrici, P. (1964), *Elements of Numerical Analysis*, John Wiley, New York.

[17] Jamshidian, M., and Jennrich, R.I. (1993), Conjugate gradient acceleration of the EM algorithm, *Journal of the American Statistical Association*, 88, 221-228.

[18] Jamshidian, M., and Jennrich, R.I. (1997), Acceleration of the EM algorithm by using quasi-Newton methods, *Journal of the Royal Statistical Society B*, 59, 569-587.

[19] Jbilou, K., and Sadok, H. (1991), Some results about vector extrapolation methods and related fixed-point iterations, *Journal of Computational and Applied Mathematics*, 36, 385-398.

[20] Jbilou, K., and Sadok, H. (1995), Analysis of some vector extrapolation methods for solving systems of linear equations, *Numerische Mathematik* (Electronic Edition), 70, 73-89.

[21] Jbilou, K., and Sadok, H. (1999), LU implementation of the modified minimal polynomial extrapolation method for solving linear and nonlinear systems, *IMA Journal of Numerical Analysis*, 19, 549-561.

[22] Laird, N., Lange, N., and Stram, D. (1987), Maximum likelihood computation with repeated measures: application of the EM algorithm, *Journal of the American Statistical Association*, 82, 97-105.

[23] Lancaster, P. (1969), *Theory of Matrices*, Academic Press, New York.

[24] Lange, K. (1995a), A gradient algorithm locally equivalent to the EM algorithm, *Journal of the Royal Statistical Society B*, 57, 425-437.

[25] Lange, K. (1995b), A quasi-Newton acceleration of the EM algorithm, *Statisticia Sinica*, 5, 1-18.

[26] Louis, T.A. (1982), Finding the observed information matrix when using the EM algorithm, *Journal of the Royal Statistical Society B*, 44, 226-233.

[27] Meilijson, I. (1989), A fast improvement to the EM algorithm on its own terms, *Journal of the Royal Statistical Society B*, 51, 127-138.

[28] Meng, X.L., and Rubin, D.B. (1993), A maximum likelihood estimation via the EM algorithm: a general framework, *Biometrika*, 80, 267-278.

[29] Meng, X.L., and Rubin, D.B. (1994), On the global and componentwise rates of convergence of the EM algorithm, *Linear Algebra and Its Applications*, 199, 413-425.

[30] Mesina, M. (1977), Convergence acceleration for the iterative solution of $x = Ax + f$, *Computational Methods in Applied Mechanics and Engineering*, 10, 165-173.

[31] Ortega, J.M., and Rheinboldt, W.C. (1970), *Iterative Solutions of Nonnlinear Equations in Several Variables*, Academic Press, New York.

[32] Nievergelt, Y. (1991), Aitken's and Steffensen's accelerations in several variables, *Numerische Mathematik*, 59, 295-310.

[33] Raydan, M., and Svaiter, B.F. (2002), Relaxed steepest descent and Cauchy-Barzilai-Borwein method, *Computational Optimization and Applications*, 21, 155-167.

[34] Redner, R.A., and Walker, H.F. (1984), Mixture densities, maximum likelihood and the EM algorithm, *SIAM Review*, 26, 195-239.

[35] Roland, Ch., and Varadhan, R. (2004), *Applied Numerical Mathematics* (in press).

[36] Schou, G. (1978), Estimation of the concentration parameter in von Mises-Fisher distributions, *Biometrika*, 65, 369-377.

[37] Shanks, D. (1955) Non linear transformations of divergent and slowly convergent sequences, *Journal of Mathematical Physics*, 34, 1-42.

[38] Sidi, A. (1986a), Acceleration of convergence of vector sequencces, *SIAM Journal of Numerical Analysis*, 23, 178-196.

[39] Sidi, A. (1986b), Convergence and stability properties of minimal polynomial and reduce rank extrapolation algorithms, *SIAM Journal of Numerical Analysis*, 23, 197-209.

[40] Sidi, A. (1988), Extrapolation vs. projection methods for linear systems of equations, *Journal of Computational and Applied Mathematics*, 22, 71-88.

[41] Sidi, A. (1991), Efficient implementation of minimal polynomial and reduced rank extrapolation methods, *Journal of Computational and Applied Mathematics*, 36, 305-337.

[42] Smith, D.A., Ford, W.F., and Sidi, A. (1987), Extrapolation methods for vector sequences, *SIAM Review*, 29, 199-233.

[43] Vardi, Y., Shepp, L.A., and Kaufman, L. (1985), A statistical model for positron emission tommography, *Journal of the American Statistical Association*, 80, 8-37.

[44] Wu, C.F.J. (1983), On the convergence properties of the EM algorithm, *The Annals of Statistics*, 11, 95-103.