SQUAREM

Acceleration Schemes for Monotone Fixed-Point Iterations Including the EM and MM algorithms in Statistical Modeling

Ravi Varadhan[1]

[1]Johns Hopkins University
Baltimore, MD, USA
Email: rvaradhan@jhmi.edu

SC 2011
Cagliari, Italy
October 13, 2011

Gratitude

- Professor Claude Brezinski
- Christophe Roland
- Marcos Raydan
- *R* Core Development Team

Fixed-Point Iteration

$$x_{k+1} = F(x_k), \quad k = 0, 1, \ldots.$$

$F : \Omega \subset \mathbb{R}^p \mapsto \Omega$, and differentiable

- F is a contraction: $||F(x) - F(y)|| \leq ||x - y||$, $\forall x, y \in \Omega$
- Associated Lyapunov function $L(x)$ such that $L(x_{k+1}) \geq L(x_k)$
- Guaranteed convergence: $\{x_k\} \to x^* \in \Omega$

EM Algorithm

Let y, z, x, be observed, missing, and complete data, respectively.
The k-th step of the iteration:

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

where

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

Ascent property: $L_{obs}(\theta_{k+1}) \geq L_{obs}(\theta_k)$

The goal is to maximize $L_{obs}(\theta; y)$

Why is EM So Popular?

- Seminal work of Dempster, Laird, and Rubin (1977)
- Most popular approach in computational statistics
- Computes MLE in "incomplete" data type problems
- Reduces incomplete-data problem (difficult) to complete-data problem (easier).
- Versatile, stable (ascent property), globally convergent under weak regularity conditions (Wu, 1983)
- Meng's paper: *EM: An old folk song sung to a new tune*

MM Algorithm

A *majorizing* function, $g(\theta|\,\theta^k)$:

$$\begin{aligned} f(\theta_k) &= g(\theta_k|\,\theta_k), \\ f(\theta) &\leq g(\theta|\,\theta_k), \ \forall\,\theta. \end{aligned}$$

- To minimize $f(\theta)$, construct a majorizing function and minimize it (MM)

$$\theta_{k+1} = \text{argmin}\, g(\theta|\theta_k); \quad k = 0, 1, \ldots$$

- Descent property: $f(\theta_{k+1}) \leq f(\theta_k)$
- EM may be viewed as a subclass of MM.

Linear Convergence of EM/MM

The EM/MM as a fixed-point iteration F:

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

Assume $\theta_k \to \theta^*$ and F is differentiable at θ^*,

$$\theta_{k+1} - \theta^* = J(\theta^*)(\theta_k - \theta^*) + o(\|\theta_k - \theta^*\|^2),$$

Jacobian of F can be written as (DLR77):

$$\begin{aligned} J(\theta^*) &= I_{miss}(\theta^*; y) I_{comp}^{-1}(\theta^*; y) \\ &= \mathrm{I}_{p\times p} - I_{obs}(\theta^*; y) I_{comp}^{-1}(\theta^*; y) \end{aligned}$$

Rate of convergence $\propto \rho\left[J(\theta^*)\right]$.

Why Accelerate the EM?

- Slow, linear convergence in practice.
- Acceleration is useful in:
  - high-dimensional and/or large scale problems (e.g., PET imaging, machine learning)
  - complex statistical models (e.g., GLMM, NLME, longitudinal data)
  - repeated model estimation (e.g., simulations, bootstrapping)

What is Desirable in an Accelerator?

- Ken Lange (1995) - "it is likely that **no** acceleration method can match the stability and simplicity of the unadorned EM algorithm."
- Simple and easy to apply (low intellectual and implementation costs)
- Stability (monotonicity and/or global convergence)
- Generally applicable to (most) **all** EM problems (exception, MCEM)
- Automatic - no problem-specific "tweaking".
- Without much additional information (e.g., gradient/hessian of L_{obs})

Iterative Acceleration Schemes

At least 2 ways to motivate these acceleration methods

1. Vector sequence extrapolation with cycling
2. Classical Newton-type root-finders

Steffensen-Type Methods (STEM)

Define $g(\theta) = F(\theta) - \theta$; $M_n = J(\theta_n) - I$; $u_0 = \theta_n$; $u_1 = F(\theta_n)$; $r_n = u_1 - u_0$; $v_n = g(u_1) - g(u_0)$
Newton's method is obtained by finding the zero of the linear approximation of $g(\theta)$:

$$g(\theta) = g(u_0) + M_n \,.\, (\theta - u_0).$$

We approximate M_n with the scalar matrix $\frac{1}{\alpha_n} I$, and write two different approximations for the fixed point $\theta^* : \; g(\theta^*) = 0$:

$$\begin{aligned} t^0_{n+1} &= u_0 - \alpha_n \, g(u_0) \\ t^1_{n+1} &= u_1 - \alpha_n \, g(u_1). \end{aligned}$$

Steffensen-Type Methods(STEM)

We now choose α_n to minimize discrepancy between t^0_{n+1} and t^1_{n+1}.
An obvious measure of discrepancy is $\|t^1_{n+1} - t^0_{n+1}\|^2$, yielding steplength

$$\alpha_n = \frac{r_n^T v_n}{v_n^T v_n}, \tag{1}$$

Another measure of discrepancy: $\|t^1_{n+1} - t^0_{n+1}\|^2/\alpha_n^2$, yielding the steplength

$$\alpha_n = \frac{r_n^T r_n}{r_n^T v_n}. \tag{2}$$

Another minimizes the discrepancy: $-\|t^1_{n+1} - t^0_{n+1}\|^2/\alpha_n$, where $\alpha_n < 0$:

$$\alpha_n = -\frac{\|r_n\|}{\|v_n\|}. \tag{3}$$

STEM

STEM:

$$\theta_{n+1} = \theta_n - \alpha_n r_n,$$

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

α_n can be one of 3 steplengths as defined in previous slide.

Mediocre performance. How can we improve it?

Cauchy-Barzilai-Borwein

Motivation: Cauchy-Barzilai-Borwein for quadratic minimization (Raydan and Svaiter, 2002)

$$\min f(x) = \frac{1}{2} x^T A x + b^T x$$

where A is symmetric and positive-definite.

- Cauchy (steepest-descent) ill-conditioned when $\rho(A) \approx 1$
- Barzilai-Borwein gradient method uses previous steplength
- RS2002 combined Cauchy and BB to obtain:

$$x_{n+1} = x_n - 2\alpha_n g_n + \alpha_n^2 h_n$$

where $g_n = Ax_n - b_n$, $h_n = Ag_n$, $\alpha_n = \frac{g_n^T g_n}{g_n^T h_n}$

SQUAREM

SQUAREM:

$$\theta_{n+1} = \theta_n - 2\alpha_n r_n + \alpha_n^2 v_n.$$

- SqS1: $\alpha_n = \frac{r_n^T v_n}{v_n^T v_n}$
- SqS2: $\alpha_n = \frac{r_n^T r_n}{r_n^T v_n}$
- SqS3: $\alpha_n = -\frac{\|r_n\|}{\|v_n\|}$,

Pseudocode of SQUAREM

While not converged

1. $\theta_1 = F(\theta_0)$
2. $\theta_2 = F(\theta_1)$
3. $r = \theta_1 - \theta_0$
4. $v = (\theta_2 - \theta_1) - r$
5. Compute α with r and v.
6. $\theta' = \theta_0 - 2\,\alpha\, r + \alpha^2\, v$
7. $\theta_0 = F(\theta')$ (stabilization)
8. Check for convergence.

SQUAREM

- An R package implementing a family of algorithms for speeding-up any slowly convergent multivariate sequence from a monotone fixed-point mapping
- Also contains higher-order cycled, squared, extrapolation schemes
- Very easy to use
- Ideal for high-dimensional problems
- Input: *fixptfn* = fixed-point mapping F
- Optional Input: *objfn* = merit function (if any)
- Two main control parameter choices: order of extrapolation and monotonicity
- Available on *CRAN*.

Table: Data from The London Times on deaths during 1910-1912

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

Binary Poisson Mixture

The incomplete-data likelihood:

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

The EM algorithm is as follows:

$$p^{(k+1)} = \sum_i n_i \, \hat{\pi}_{i1}^{(k)} \Big/ \sum_i n_i,$$

$$\mu_j^{(k+1)} = \sum_i i \, n_i \, \hat{\pi}_{ij}^{(k)} \Big/ \sum_i n_i \, \hat{\pi}_{ij}^{(k)}, \; j = 1, 2,$$

$$\hat{\pi}_{ij}^{(k)} = p^{(k)} \left(\mu_j^{(k)} \right)^i e^{-\mu_j^{(k)}} \Big/ \sum_{l=1}^{2} p_l^{(k)} \left(\mu_l^{(k)} \right)^i e^{-\mu_l^{(k)}}, \; j = 1, 2.$$

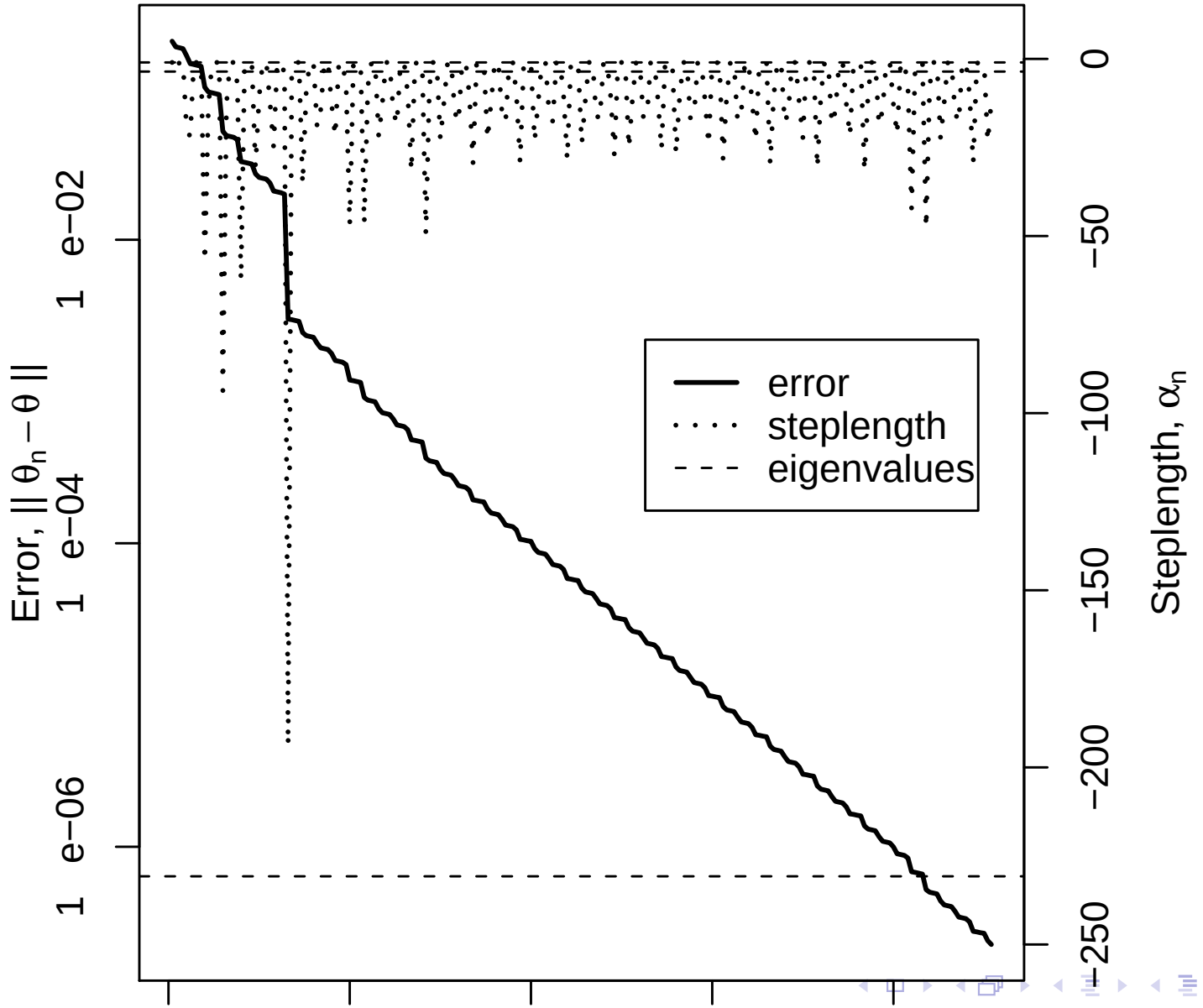
Binary Poisson Mixture (cont...)

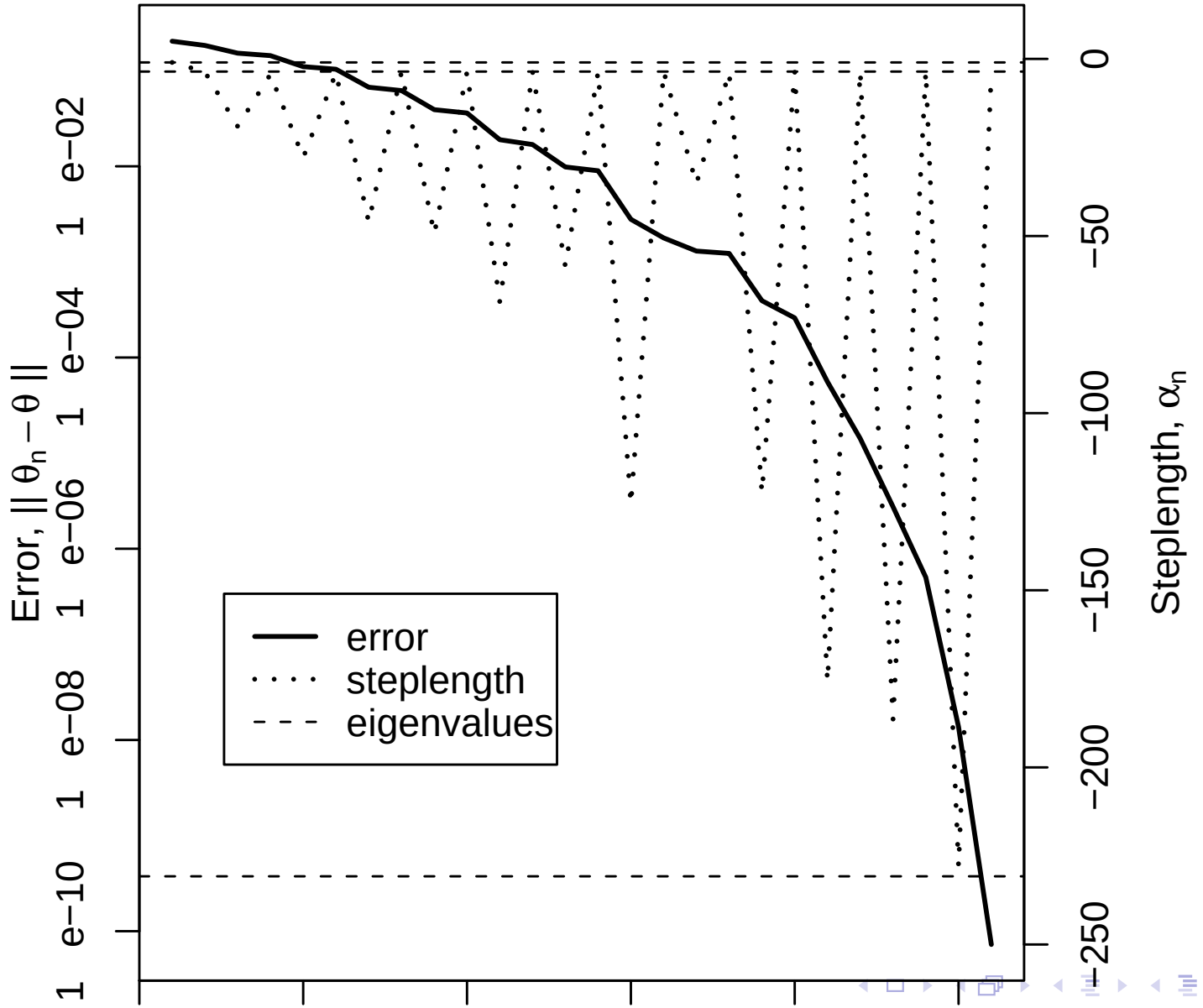
- MLE: $(p, \mu_1, \mu_2) = (0.3599, 1.256, 2.663)$.
- Eigenvalues of Jacobian at MLE: (0.9957, 0.7204 and 0)
- Eigenvalues of $(J - I)^{-1}$: (-1, -3.58, and -230.7)
- Major separation of the largest eigenvalue.
- Steplength α_n must approximate all eigenvalues.
- EM always takes $\alpha_n = -1$.

Performance of Schemes

Table: Poisson mixture estimation: initial guess θ_0 = (0.3, 1.0, 2.5)

	EM	S1	S2	S3	SqS1	SqS2	SqS3
CPU (sec)	0.26	0.11	0.13	0.16	0.01	0.03	0
fevals	2055	396	477	576	66	84	66
log-lik	−1989.9	−1989.9	−1989.9	−1989.9	−1989.9	−1989.9	−1989.9


Error, || θn − θ ||
1 e−02
1 e−04
1 e−06
Steplength, αn
0
−50
−100
−150
−200
−250
error
steplength
eigenvalues


Error, $\| \theta_n - \theta \|$
1 e−10
1 e−08
1 e−06
1 e−04
1 e−02
Steplength, α_n
0
−50
−100
−150
−200
−250
error
steplength
eigenvalues

Concluding Thoughts

- SQUAREM tested on a number of applications.
- Significant acceleration (depends on the linear rate of F)
- Globally convergent (with EM/MM safeguard)
- Works efficiently and reliably on high-dimensional and complex nonlinear statistical models
- R package called *SQUAREM available on* CRAN
- Our work has stimulated a lot of research interest in devising acceleration schemes for EM/MM algorithms
- We have compiled state-of-art accelerators and are performing extensive benchmarking studies
- R package called *turboEM*

What Needs to be Done?

- Theoretical characterization of the local convergence of SQUAREM (cf. Barzilai-Borwein)
- Theoretical characterization of the local convergence of k-SQUAREM
- Improved constraints handling
- Multi-parameter schemes of Brezinski and Chehab (1999)

For Further Reading I


R. Varadhan, and C. Roland
Scandinavian Journal of Statistics.
2008.

Thank You!

Scalar Extrapolation

- A sequence that depends on a parameter
- Extrapolation usually at 0 or ∞
- Romberg integration; Aitken's Δ^2 process
- Given a scalar sequence: $\theta_1, \ldots, \theta_n$
- Asymptotic behavior (explicit kernel): $\theta_n = x^* + c\,\lambda^n$.
- Asymptotic behavior (implicit kernel): $\theta_{n+1} - x^* = \lambda(\theta_n - x^*)$
- Aitken's extrapolation: $t_n = \theta_n - \frac{(\Delta\theta_n)^2}{\Delta^2\theta_n}$

Extension to Vector Extrapolation

- Kernel of Aitken's extrapolation: $\theta_n = x^* + c\,\lambda^n$.
- For vector x, assume kernel: $\theta_n = x^* + \sum_{i=1}^{k} c_i\,\lambda_i^n$.
- Implicit kernel: $a_0(\theta_n - x^*) + \cdots + a_k(\theta_{n+k} - x^*) = 0$.
- By subtraction: $a_0\Delta\theta_n + \cdots + a_k\Delta\theta_{n+k} = 0$.
- Inner product : $a_0\langle y, \Delta\theta_n\rangle + \cdots + a_k\langle y, \Delta\theta_{n+k}\rangle = 0$.
- Write out k more equations and solve for $a_0, \ldots, a_k$
- Many ways to obtain $a_0, \ldots, a_k$ (open problem!)

General Vector Extrapolation Methods

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

Special Classes of Schemes

- Minimal polynomial extrapolation (MPE) : $y_i^{(n)} = \Delta\theta_{n+i}$
- Reduced rank extrapolation (RRE) : $y_i^{(n)} = \Delta^2\theta_{n+i}$.
- Topological epsilon algorithm (TEA): $y_i^{(n)} = y$
- Henrici's method : $k = p$ and $y_i^{(n)} = e_i$
- Louis (1982), Laird (1987): Henrici's (multivariate Aitken)
- Compact matrix representation

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

Cycling with Extrapolation Schemes

Most common and optimal way to implement extrapolation.

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 $F()$ $k+1$ times to get $u_1^{(n+1)}, \ldots, u_{k+1}^{(n+1)}$, where
$$u_{i+1}^{(n+1)} = F(u_i^{(n+1)}),\ i = 0, \ldots, k.$$
3. Apply the extrapolation scheme to the sequence $u_0^{(n+1)}, \ldots, u_{k+1}^{(n+1)}$ to obtain $t_n^{(k)}$.
4. Set $x_{n+1} = t_n^{(k)}$, and check for convergence.
5. If no convergence, back to step (1) for next cycle.