# A componentwise accurate shift technique for *M*-matrix algebraic Riccati equations

E. Addis, B. Iannazzo, *F. Poloni*

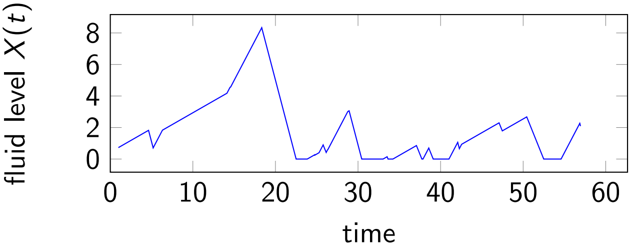
ILAS Conference, Madrid, June 2023

# Plan

- What is a fluid queue?
- What is doubling? (ADDA)
- What is componentwise accuracy?
- What is the shift technique?
- How do we combine them all together?

# Fluid queues

Fluid queue: "infinite-size bucket" in which the fluid level $X(t)$ changes with a rate which depends on the state $\varphi(t)$ of a continuous-time Markov chain with generator matrix $Q$.




[Moran '54, Mitra '88, Kulkarni '97, Ahn-Ramaswami '03, Bean-O'Reilly-Taylor '05, etc.]

In this talk: rates $\pm 1$; states $S = S_+ \cup S_-$.

$$\Psi_{ij} = P[\text{first return to } X(t) = 0 \text{ in state } \varphi(t) = j \in S_- \mid \varphi(0) = i \in S_+].$$

# $M$-matrix algebraic Riccati equation

$M$-matrix algebraic Riccati equation (MARE):

$$Q_{+-} + Q_{++}\Psi + \Psi Q_{--} + \Psi Q_{-+}\Psi = 0.$$

The matrix

$$M = -Q = -\begin{bmatrix} Q_{--} & Q_{-+} \\ Q_{+-} & Q_{++} \end{bmatrix}$$

is an irreducible, singular M-matrix.

$Q\mathbf{1} = 0$, $\boldsymbol{\pi} Q = 0$ for a row vector $\boldsymbol{\pi} > 0$.

# Eigenvalues

The most important thing: eigenvalues.

Solving the MARE $\leftrightarrow$ finding an invariant subspace:

$$\underbrace{\begin{bmatrix} -Q_{--} & -Q_{-+} \\ Q_{+-} & Q_{++} \end{bmatrix}}_{:=\mathcal{H}} \begin{bmatrix} I_{n_-} \\ \Psi \end{bmatrix} = \begin{bmatrix} I_{n_-} \\ \Psi \end{bmatrix} \underbrace{(-Q_{--} - Q_{-+}\Psi)}_{:=-U}.$$

$$\Lambda(\mathcal{H}) = \{ \underbrace{\lambda_1, \lambda_2, \dots, \lambda_{n_+}}_{=\Lambda(V)\subset\text{left half-plane}}, \underbrace{\lambda_{n_++1}, \lambda_{n_++2}, \dots, \lambda_{n_++n_-}}_{=\Lambda(-U)\subset\text{right half-plane}} \}.$$

For simplicity we assume $\lambda_{n_++1} = 0$: recurrent process.

Critical case: $\lambda_{n_+} \approx 0$. Algorithms slow down.

# Spectrum of $\mathcal{H}$ – figure

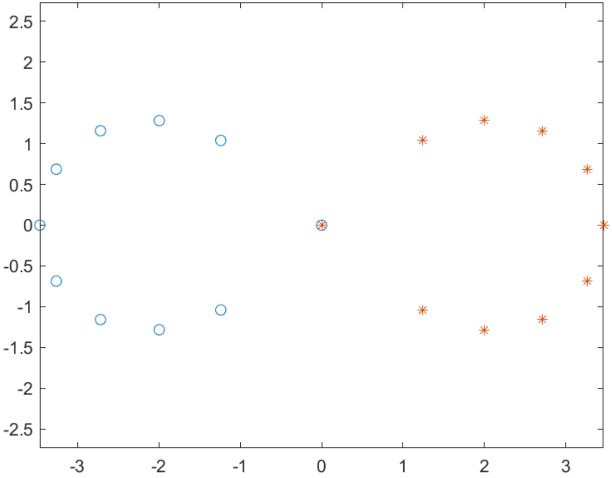



Figure: $\Lambda(\mathcal{H})$ in the critical case where $\lambda_{n_+} = \lambda_{n_+ + 1} = 0$

# Alternating-direction doubling algorithm [Wang et al. 2012]

1. Choose rates

$$\alpha \geq \alpha_{\text{opt}} = \max_i |(Q_{++})_{ii}|, \;\; \beta \geq \beta_{\text{opt}} = \max_i |(Q_{--})_{ii}|.$$

2. Compute initial values

$$\begin{bmatrix} E_0 & G_0 \\ H_0 & F_0 \end{bmatrix} = -\underbrace{\begin{bmatrix} Q_{--} - \alpha I & Q_{-+} \\ Q_{+-} & Q_{++} - \beta I \end{bmatrix}^{-1}}_{\text{M-matrix}} \underbrace{\begin{bmatrix} Q_{--} + \beta I & Q_{-+} \\ Q_{+-} & Q_{++} + \alpha I \end{bmatrix}}_{\text{non-negative}}.$$

3. Iterate

$$P_k = \begin{bmatrix} E_k & G_k \\ H_k & F_k \end{bmatrix} \mapsto P_{k+1} = \begin{bmatrix} E_{k+1} & G_{k+1} \\ H_{k+1} & F_{k+1} \end{bmatrix},$$

$$\begin{aligned} E_{k+1} &= E_k(I - G_k H_k)^{-1} E_k, \\ F_{k+1} &= F(I - H_k G_k)^{-1} F_k, \\ G_{k+1} &= G_k + E_k(I - G_k H_k)^{-1} G_k F_k, \\ H_{k+1} &= H_k + F_k(I - H_k G_k)^{-1} H_k E_k. \end{aligned}$$

# What is ADDA?

The algebra constructing a factorization

$$f(\mathcal{H})^{2^k} = \begin{bmatrix} I & -G_k \\ 0 & F_k \end{bmatrix}^{-1} \begin{bmatrix} E_k & 0 \\ -H_k & I \end{bmatrix}, \quad f(z) = \frac{z-\beta}{z+\alpha}.$$

The interpretation Constructing a "level-crossing" QBD associated to the queue

$$A_{-1} = \begin{bmatrix} 0 & 0 \\ 0 & F_0 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 0 & G_0 \\ H_0 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} E_0 & 0 \\ 0 & 0 \end{bmatrix},$$

and applying Cyclic Reduction to it. [Bean-Nguyen-P. '18]

The practice Fastest iteration in literature for this problem.

Added benefit: componentwise accuracy.

# Componentwise accuracy

An algorithm to compute a matrix/vector quantity $\Psi$ is componentwise accurate if all entries, even tiny ones, are computed with small forward error

$$\frac{|\Psi_{ij}^{computed} - \Psi_{ij}|}{\Psi_{ij}} \approx \mathsf{u}$$

Very strict requirements:

- High condition numbers prevent forward stable computations.
- Often errors are proportional to $\|\Psi\|$ or $\max_{ij} \Psi_{ij}$. E.g., when computing
$$\Psi = \begin{bmatrix} 0.5 & 0.499999 & 0.000001 \end{bmatrix}$$
on the last component we expect $\approx 10$ correct digits, not $\approx 16$.

# Keys to componentwise accuracy

## Triplet representations

Given an M-matrix $M$, we can compute $M^{-1}$ with componentwise accuracy if we know (in addition to the entries of $M$) vectors $\mathbf{v} > 0, \mathbf{w} \geq 0$ such that $M\mathbf{v} = \mathbf{w}$.

GTH-like algorithm Modified Gaussian elimination, using the relation $M\mathbf{v} = \mathbf{w}$ to evaluate pivots more accurately. [Alfa, Xue, Ye '01]

Example In the ADDA initial values, we need $M_{\alpha,\beta}^{-1}$, where

$$M_{\alpha,\beta} = -\begin{bmatrix} Q_{--} - \alpha I & Q_{-+} \\ Q_{+-} & Q_{++} - \beta I \end{bmatrix}.$$

Solution Since $Q\mathbf{1} = \mathbf{0}$, we know that the M-matrix satisfies exactly $M_{\alpha,\beta}\mathbf{1} = \left[\begin{smallmatrix} \alpha\mathbf{1} \\ \beta\mathbf{1} \end{smallmatrix}\right]$.

# Keys to componentwise accuracy

More generally:

**No Inaccurate Cancellation (NIC) principle** [Demmel-Dumitriu-Holtz-Koev '08]

To achieve componentwise accuracy, we must avoid subtractions $a - b$ with $a \approx b$.

Example In the ADDA initial values, we need

$$Q_{++} + \alpha I, \quad \text{where } \alpha \geq \alpha_{\text{opt}} = \max_i |(Q_{++})_{ii}|.$$

Choosing $\alpha = \alpha_{\text{opt}}$ may lead to trouble if e.g. $\operatorname{diag}(Q_{++}) = \begin{bmatrix} -1 \\ -0.99999 \end{bmatrix}$.

Solution choose $\alpha$ a bit larger, for instance $\alpha = 1.25\alpha_{\text{opt}}$. Convergence speed is slightly degraded, but we gain in accuracy.

Using these techniques, one can make ADDA componentwise accurate. [Xue-Xu-Li '12, Nguyen-P '15, Xue-Li '17]

# Spectrum of $f(\mathcal{H})$

The most important thing: eigenvalues.

Recall that $f(z) = \frac{z-\beta}{z+\alpha}$. In ADDA we work with $f(\mathcal{H})$, with eigenvalues

- $\Lambda(f(-U)) = \{f(\lambda_{n_++1}), f(\lambda_{n_++2}), \dots, f(\lambda_{n_++n_-})\} \subset \{|z| \leq \frac{\beta}{\alpha}\}$ and
- $\Lambda(f(V)) = \{f(\lambda_1), f(\lambda_2), \dots, f(\lambda_{n_+})\} \subset \{|z| \geq \frac{\beta}{\alpha}\}$.

ADDA convergence depends on the parameter

$$\xi = \rho(f(-U))\,\rho(f(V)^{-1}) = \left|\frac{f(\lambda_{n_++1})}{f(\lambda_{n_+})}\right| \leq 1 :$$

- If $\xi < 1$, $\|H_k - \Psi\| \sim \xi^{2^k}$.
- If $\xi = 1$ (null recurrent queue), $\|H_k - \Psi\| \sim 2^{-k}$.

**Problem**

Slow convergence when $\xi = 1$ or $\xi \approx 1$: how to speed it up?

# Spectrum of $f(\mathcal{H})$ – figure

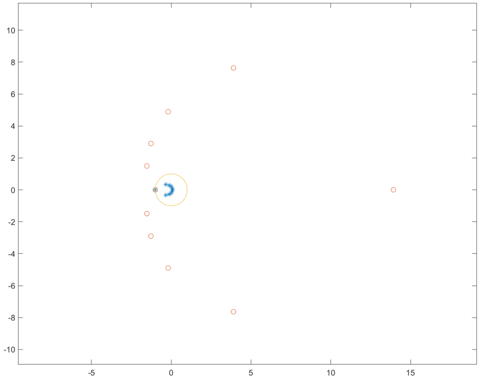



Figure: $\Lambda(f(\mathcal{H}))$ in the critical case where $\lambda_{n_+} = \lambda_{n_++1} = 0$

# Shift: from $\mathcal{H}$ to $\hat{\mathcal{H}}$ [Guo-Iannazzo-Meini '07]

Shift technique: rank-1 modification to accelerate convergence in (near-)critical cases.
We choose $\eta > 0$ and $\mathbf{p} \geq 0$ such that $\mathbf{p}^T \mathbf{1} = 1$.

| | |
|---|---|
| $\mathcal{H}$ | $\hat{\mathcal{H}} = \mathcal{H} + \eta \mathbf{1} \mathbf{p}^T$ |
| $Q$ | $\hat{Q} = Q - \eta \begin{bmatrix} \mathbf{1}_{n_-} \\ -\mathbf{1}_{n_+} \end{bmatrix} \mathbf{p}^T$ |
| $Q_{+-} + Q_{++}\Psi + \Psi Q_{--} + \Psi Q_{-+}\Psi = 0$ | $\hat{Q}_{+-} + \hat{Q}_{++}\Psi + \Psi \hat{Q}_{--} + \Psi \hat{Q}_{-+}\Psi = 0$ |
| $\Psi$ | same $\Psi$ |
| $\lambda_{n_+ + 1} = 0$ | $\hat{\lambda}_{n_+ + 1} = \eta$ |
| $\Lambda(-U) = \{0, \lambda_{n_+ + 1}, \dots, \lambda_{n_+ + n_-}\}$ | $\Lambda(-\hat{U}) = \{\eta, \lambda_{n + + 1}, \dots, \lambda_{n_+ + n_-}\}$ |
| $\xi$ | $\hat{\xi} < \xi$ (at least when $\eta < \beta$). |

The technique may decrease the number of steps dramatically.

# Componentwise accurate construction of $\hat{P}_0$

In this talk: Can we combine the two improvements? Shift technique and componentwise accurate computations?

$$\hat{Q} = Q - \eta \begin{bmatrix} \mathbf{1}_{n_-} \\ -\mathbf{1}_{n_+} \end{bmatrix} \mathbf{p}^T$$

Problem: The sign properties may be lost, even for tiny values of $\eta$!

$$\begin{bmatrix} \hat{E}_0 & \hat{G}_0 \\ \hat{H}_0 & \hat{F}_0 \end{bmatrix} = \underbrace{-\begin{bmatrix} \hat{Q}_{--} - \hat{\alpha} I & \hat{Q}_{-+} \\ \hat{Q}_{+-} & \hat{Q}_{++} - \hat{\beta} I \end{bmatrix}^{-1}}_{\text{M-matrix???}} \underbrace{\begin{bmatrix} \hat{Q}_{--} + \hat{\beta} I & \hat{Q}_{-+} \\ \hat{Q}_{+-} & \hat{Q}_{--} + \hat{\alpha} I \end{bmatrix}}_{\text{non-negative???}}.$$

# Delayed shift

Idea: When $\alpha, \beta$ are fixed, $\hat{P}_0$ is a rank-1 modification of $P$:

$$\hat{P}_0 = P_0 - \underbrace{\eta\,(\alpha+\beta)\frac{\mathbf{u}\mathbf{p}^T {M_{\alpha,\beta}}^{-1}}{1+\eta\mathbf{p}^T\mathbf{u}}}_{=:\Sigma_\eta}, \quad \mathbf{u} = {M_{\alpha,\beta}}^{-1}\begin{bmatrix} \mathbf{1}_{n_-} \\ -\mathbf{1}_{n_+} \end{bmatrix}. \qquad (*)$$

We can first compute $P_0 > 0$, then construct $\hat{P}_0$ by subtraction (delayed shift).

$P_0 - \Sigma_\eta$ contains subtractions, but we can compute all the quantities in (*) and then choose $\eta$ afterwards to satisfy two entrywise conditions:

- $\hat{P}_0 > 0$
- No Inaccurate Cancellation in $P_0 - \Sigma_\eta$.

The range of allowed values for $\eta$ is often much larger than when applying the regular "non-delayed" shift.

# The missing steps

- Triplet representations: can be computed from the relation

$$\begin{bmatrix} \tau^{2^k}\hat{E}_k & \hat{G}_k \\ \hat{H}_k & \tau^{-2^k}\hat{F}_k \end{bmatrix}\mathbf{1} = \mathbf{1}, \quad \text{with } \tau = \frac{\alpha+\eta}{\beta-\eta}. \qquad \text{(T)}$$

- Positivity, applicability, convergence: can be proved by mimicking the original proofs in ADDA: the only assumptions they need are $\hat{P}_0 \geq 0$ and (T).
- Forward error bound: can be obtained, though worse than in the non-shifted case:

$$|\text{computed}(\Sigma) - \Sigma| \leq \underbrace{\mathcal{O}(n^3 \mathrm{u})}_{\text{machine prec.}} \underbrace{\frac{1+\mathbf{p}^T M_{\alpha\beta}^{-1}\mathbf{1}}{(1+\mathbf{p}^T\mathbf{u})^2}(\alpha+\beta) M_{\alpha\beta}^{-1}\mathbf{1}\mathbf{p}^T M_{\alpha\beta}^{-1}}_{\text{not } \Sigma\text{, possibly larger}}.$$

# Example 1 [Nguyen-P '15, Example 5.1]

An example with $Q$ with imbalanced entries.

$$\Lambda(\mathcal{H}) = \{-20.0000, -1.5625, -0.0100, 0.0000, 2.5575, 19.9800\}$$

$$\Lambda(f(\mathcal{H})) = \{-6.9970, -1.2314, -1.0003, -0.9990, -0.7078, 0.1428\}$$

- without shift: convergence rate $\xi = 0.9990$.
- non-accurate shift: $\hat{\xi} = 0.7078$.
  Optimal $\eta = \beta = 14.9850$, $\mathbf{p} = \frac{1}{n}\mathbf{1}$.
- accurate shift, imposing $\hat{P}_0 \geq 0$: $\hat{\xi} = 0.8575$.
  $\eta = 1.1429$, $f(\eta) = -0.8575$, $\mathbf{p} = \mathbf{e}_5$.
- accurate shift, imposing NIC in $\hat{P}_0 = P_0 - \Sigma_\eta$: $\hat{\xi} = 0.9777$.
  $\eta = 0.1614$, $f(\eta) = -0.9777$, $\mathbf{p} = \mathbf{e}_5$.

# Example 1

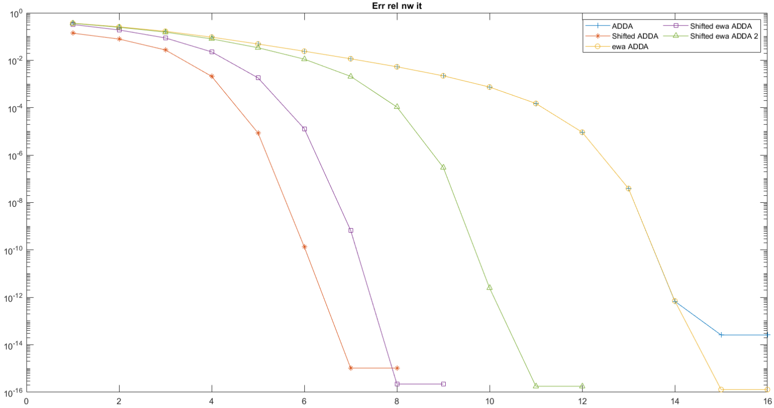



Figure: Normwise relative error $\|H_k - \Psi\|/\|\Psi\|$ vs. iteration $k$.

# Example 1

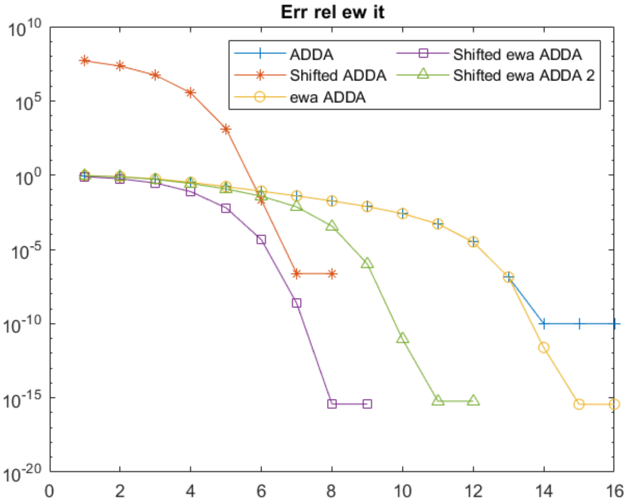
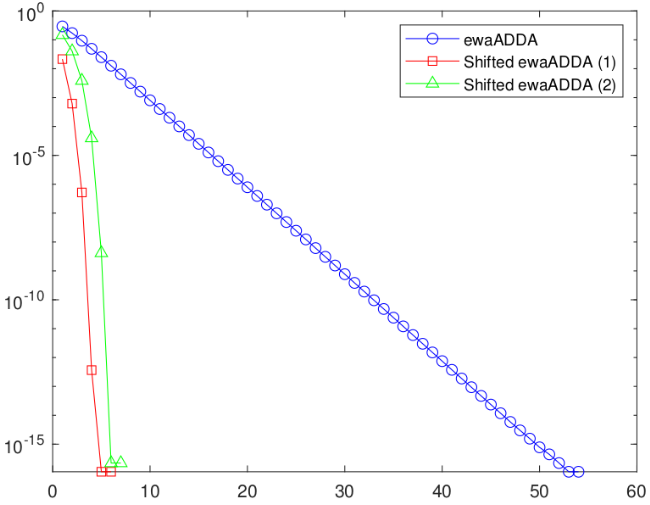


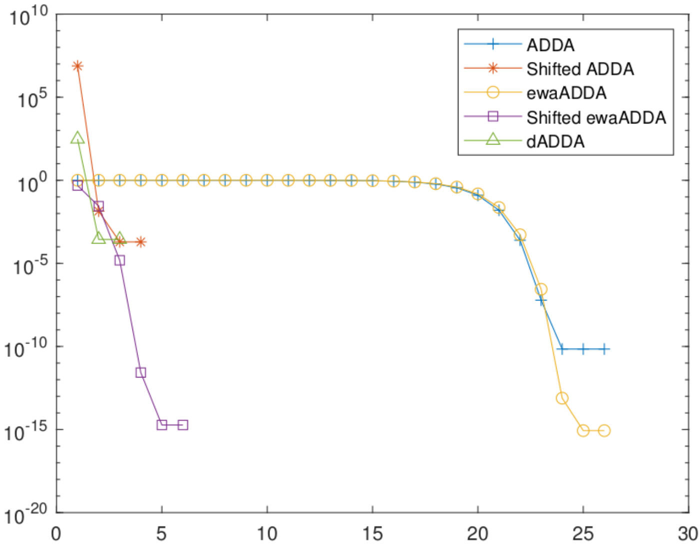

Figure: Componentwise relative error $\max_{i,j} |(H_k)_{ij} - \Psi_{ij}|/\Psi_{ij}$ vs. iteration $k$.

# Example 2



ewaADDA
Shifted ewaADDA (1)
Shifted ewaADDA (2)
$10^{0}$
$10^{-5}$
$10^{-10}$
$10^{-15}$
0
10
20
30
40
50
60

# Example 3



ADDA
Shifted ADDA
ewaADDA
Shifted ewaADDA
dADDA
$10^{10}$
$10^{5}$
$10^{0}$
$10^{-5}$
$10^{-10}$
$10^{-15}$
$10^{-20}$
0
5
10
15
20
25
30

# Conclusions

- You can have your cake (shift) and eat it, too (componentwise accuracy). *Se puede estar a la vez en la procesión y repicando las campanas*
- In many examples, we can lower the number of iterations to match that of shift, while keeping the original high accuracy.
- Are there benefits in delaying the shift even further?

Reference Elena Addis's thesis at UNIFI, *Elementwise accurate algorithms for nonsymmetric algebraic Riccati equations associated with M-matrices*, `https://hdl.handle.net/2158/1275470`. Article version in preparation.

# Conclusions

- You can have your cake (shift) and eat it, too (componentwise accuracy). *Se puede estar a la vez en la procesión y repicando las campanas*
- In many examples, we can lower the number of iterations to match that of shift, while keeping the original high accuracy.
- Are there benefits in delaying the shift even further?

Reference Elena Addis's thesis at UNIFI, *Elementwise accurate algorithms for nonsymmetric algebraic Riccati equations associated with M-matrices*, `https://hdl.handle.net/2158/1275470`. Article version in preparation.

Thanks for your attention!