Componentwise accurate numerical methods for Markov-modulated Brownian motion

Giang T. Nguyen[1]   Federico Poloni[2]

[1]U of Adelaide, School of Mathematical Sciences
[2]U Pisa, Italy, Dept of Computer Science

9th Matrix Analytic Methods Conference
Budapest, June 2016

Subtraction-free algorithms

Error analysis in an (imprecise) slogan

TL;DR: when you subtract two close numbers, you lose accuracy.

More precise: instead of a and b, a computer may store $a + \delta$ and $b + \varepsilon$; the number $(a + \delta) - (b + \varepsilon)$ may be at a large relative distance from $a - b$ (if a and b have the same sign).

So, let's stop doing subtractions.

Luckily, for many probabilities computations this is possible.
E.g., computing AB, for $A \geq 0, B \geq 0$.

Most subtractions come from M-matrices, but we can avoid them!

Regular M-matrices

A matrix A with sign pattern (possibly including zeros)

$$A = \begin{bmatrix} + & - & - & - \\ - & + & - & - \\ - & - & + & - \\ - & - & - & + \end{bmatrix}$$

is called regular M-matrix if there are $\mathbf{v} > \mathbf{0}$, $\mathbf{w} \geq \mathbf{0}$ such that $A\mathbf{v} = \mathbf{w}$.
E.g., $(-Q)\mathbf{1} = \mathbf{0}$ for the rate matrix of a CTMC

Attention! [Guo CH, 2013]

Not all M-matrices are regular! E.g., $\begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}$

GTH algorithm

For a regular M-matrix A, one can store its off-diagonal entries, $\mathbf{v}$ and $\mathbf{w}$ (triplet representation).

$$A = \begin{bmatrix} ? & - & - & - \\ - & ? & - & - \\ - & - & ? & - \\ - & - & - & ? \end{bmatrix} \qquad A\mathbf{v} = \mathbf{w}$$

GTH-like algorithm [Grassmann et al '85, O'Cinneide '93, Alfa et al '02...]

Given a triplet representation for $A \in \mathbb{R}^{n\times n}$, one can compute $B = A^{-1}$ subtraction-free, obtaining perfect componentwise accuracy:

$$|\tilde{b}_{ij} - b_{ij}| \leq O(n^3) \cdot |b_{ij}| \cdot \texttt{eps}.$$

Variants: LU factorization, left and right kernel, Perron vector.
Variant: $\boldsymbol{v}^\top A = \boldsymbol{w}^\top$

GTH algorithm

An example

$A = \begin{bmatrix} 1 & -1 \\ -1 & 1+\varepsilon \end{bmatrix}$: can only compute inverse up to accuracy $\kappa(A) \approx \varepsilon^{-1}$.

$A = \begin{bmatrix} ? & -1 \\ -1 & ? \end{bmatrix}$ such that $A \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon \end{bmatrix}$: full accuracy possible.

Works especially well when dealing with different orders of magnitude.

Plan: triplet representation are a great idea — let's rewrite our matrix iterations to use them!

Obtaining triplets

Theorem [Nguyen P. '16 — or earlier?]

Given a regular M-matrix and its triplet representation partitioned as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix},$$

one can obtain explicitly without subtractions triplet representations for its submatrices and Schur complements (censorings):

$$Dv_2 = w_2 - Cv_1,$$
$$(A - BD^{-1}C)v_1 = w_1 - BD^{-1}w_2.$$

Cyclic reduction

CR solves matrix equations of the form $R^2A - RB + C = 0$, with $A, C \geq 0$, and $B - A - C$ a regular M-matrix [Bini et al '01, BLM book]

Cyclic Reduction

$$\begin{aligned} A_0 = A, B_0 = \hat{B}_0 = B, C_0 = C \\ A_{k+1} &= A_k B_k^{-1} A_k, \\ B_{k+1} &= B_k - A_k B_k^{-1} C_k - C_k B_k^{-1} A_k, \\ C_{k+1} &= C_k B_k^{-1} C_k, \\ \hat{B}_{k+1} &= \hat{B}_k - C_k B_k^{-1} A_k. \end{aligned}$$

At the end of the iteration, $R = C_0 \hat{B}_\infty^{-1}$, where $\hat{B}_\infty = \lim \hat{B}_k$.

Cyclic Reduction and triplets

Cyclic Reduction = Schur complements on $\begin{bmatrix} \ddots & \ddots & & & \\ \ddots & B & -C & & \\ & -A & B & -C & \\ & & -A & B & \ddots \\ & & & \ddots & \ddots \end{bmatrix}$

Two triplet representations follow:

$(A_k - B_k + C_k)\mathbf{1} = \mathbf{0}$: gives triplet for B_k, already known and used. [e.g. Bini et al, SMCTools software]

$(A_0 - \hat{B}_k + C_k)\mathbf{1} = \mathbf{0}$: gives triplet for $\hat{B}_\infty$, new for the final step.

Theorem [Nguyen P. '16]

CR with triplet representations gives $|\tilde{f} - f| \leq O(2^k n^4)|f|\text{eps}$ for the computed value $\tilde{f}$ of each entry f of $A_k, B_k, C_k, \hat{B}_k, R_k$.

Doubling algorithm

An unusual matrix iteration that can be seen as repeated censoring / Schur complementation:

$$\begin{bmatrix} E_{new} & G_{new} \\ H_{new} & F_{new} \end{bmatrix} = \begin{bmatrix} 0 & G \\ H & 0 \end{bmatrix} + \begin{bmatrix} E & 0 \\ 0 & F \end{bmatrix} \left(I - \begin{bmatrix} 0 & G \\ H & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} E & 0 \\ 0 & F \end{bmatrix}$$

Keep track of triplet representations at each step $\implies$ componentwise-accurate algorithms for fluid queues / Riccati equations.
[Xue et al '12, Nguyen P. '15, Xue Li '16]

Markov-modulated Brownian motion [Asmussen '95, Karandikar Kulkarni '95, Rogers '94]

- $\phi(t)$ continuous-time Markov chain with transition matrix $Q \in \mathbb{R}^{n\times n}$;
- $y(t)$ evolves as Brownian motion with drift $d_{\phi(t)}$ and variance $v_{\phi(t)}$.

The invariant density follows

$$\boldsymbol{p}''(x)V - \boldsymbol{p}'(x)D + \boldsymbol{p}(x)Q = \mathbf{0}.$$

$V, D \in \mathbb{R}^{n\times n}$ diagonal matrices containing v_i, d_i.
Invariant density and many properties can be computed using an invariant pair, i.e., ($X \in \mathbb{R}^{\ell\times\ell}$, $U \in \mathbb{R}^{\ell\times n}$) such that

$$X^2UV - XUD + UQ = 0.$$

[Rogers '94, Ivanovs, '10, Betcke Kressner '11, Gohberg et al '82]
Often, $U = \begin{bmatrix} I & \Psi \end{bmatrix}$.

Invariant pairs and Cyclic Reduction

How do we compute invariant pairs? Cyclic reduction gives a special one:

$$R^2 I A - R I B + I C = 0.$$

Plan: tinker with the problem to turn it into this form.

Discretizing transformation from eigenvalue properties: [Bini et al '10]

- we need a "continuous-time stable" pair: $\mathtt{eig}(X) \subseteq$ left half-plane.
- CR produces a "discrete-time stable" one: $\mathtt{eig}(R) \subseteq$ unit circle

So we make a change of variables $R = f(X)$.

$R = (I + X)(I - X)^{-1}$ won't work: cannot find $X = f^{-1}(R)$ subtraction-free.

Instead, we use $R = I + hX$, with h sufficiently small.

The algorithm

1. Choose h small enough so that

$$v_i + d_i h + q_{ii} h^2 > 0 \qquad (*)$$

   (and the subtraction is 'tame').
2. Set $A := \frac{1}{h^2} V \geq 0, \quad B := 2\frac{1}{h^2} V + \frac{1}{h} D, \quad C := \frac{1}{h^2} V + \frac{1}{h} D + Q \geq 0.$
3. Use subtraction-free CR on (A, B, C) to compute R.
4. Compute the off-diagonal of $X = h^{-1}(R - I)$.
5. Compute the left Perron vector $\boldsymbol{\mu}$ of Q using the triplet $Q\mathbf{1} = \mathbf{0}$.
6. Compute the triplet $\boldsymbol{\mu}(-X) = \frac{1}{h} \boldsymbol{\mu} A_\infty \hat{B}_\infty^{-1}$, and obtain $\operatorname{diag}(X)$.

Works whenever $v_i > 0$ for all i (positive variances).
Otherwise, we can't enforce (*)

Zero variances

- In $E_2 = \{i : v_i = 0, d_i \leq 0\}$, we can't obtain $v_i + d_i h + q_{ii} h^2 > 0$.
- We won't be able to choose $U = I$, because we only have enough stable eigenvalues to form an invariant pair of size $n - |E_2|$.

Solution to both problems: shift infinite eigenvalues [He et al '01].

From:

$$A = \begin{bmatrix} + & 0 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} + & 0 \\ 0 & - \end{bmatrix}, \quad C = \begin{bmatrix} + & + \\ + & ?? \end{bmatrix}$$

move 2nd column "one matrix to the left" and change its sign:

$$\hat{A} = \begin{bmatrix} + & 0 \\ 0 & + \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} + & - \\ 0 & M \end{bmatrix}, \quad \hat{C} = \begin{bmatrix} + & 0 \\ + & 0 \end{bmatrix}.$$

Shift $\iff$ differentiating some of the equations.

See "index reduction" in ODE literature. [Kunkel Mehrmann '06]

Recovering the solution

Still, R is not the final solution.

The shift trick adds spurious zero eigenvalues: $R = \begin{bmatrix} + & 0 \\ + & 0 \end{bmatrix} \hat{B}_{\infty}^{-1}$.

Solution to remove them: switch to a different invariant pair:

$$(KRK^{-1})^2 KA + (KRK^{-1}) KB + KC = 0.$$

$$\begin{bmatrix} R_{11} & 0 \\ R_{12} & 0 \end{bmatrix}^2 \begin{bmatrix} I & \Psi \\ 0 & I \end{bmatrix} A + \begin{bmatrix} R_{11} & 0 \\ R_{12} & 0 \end{bmatrix} \begin{bmatrix} I & \Psi \\ 0 & I \end{bmatrix} B + \begin{bmatrix} I & \Psi \\ 0 & I \end{bmatrix} C = 0.$$

New R is block-triangular $\implies$ "reduced invariant pair" $(R_{11}, [I \quad \Psi])$.

True but not obvious: all this can be done subtraction-free.

The (general) algorithm

1. Choose h small so that $h^{-2}v_i + h^{-1}d_i + q_{ii} > 0$ for all $i \notin E_2$.
2. Set $A := \frac{1}{h^2}V \geq 0, \quad B := 2\frac{1}{h^2}V + \frac{1}{h}D, \quad C := \frac{1}{h^2}V + \frac{1}{h}D + Q \geq 0$.
3. Shift technique to produce equivalent $\hat{A}, \hat{B}, \hat{C}$.
4. Use componentwise accurate CR on $(\hat{A}, \hat{B}, \hat{C})$ to compute $\hat{B}_\infty$.
5. Use similarity as in the previous slide to compute $(R_{11}, [I \quad \Psi])$.
6. Compute the off-diagonal of $X = h^{-1}(R_{11} - I)$.
7. Compute the left Perron vector $\boldsymbol{\mu}$ of Q using the triplet $Q\mathbf{1} = \mathbf{0}$.
8. Compute a triplet $\boldsymbol{v}^\top(-X) = \boldsymbol{w}^\top$ (long formula omitted).

Works even with zero variances.

Experiments: the competitors

- **KK** [Karandikar Kulkarni '95]: compute eigenvalues explicitly.
- **AS** [Agapie Sohraby '01]: iterative algorithm for span(stable eigenvalues), then orthogonal transformations.
- **LN** [Latouche Nguyen '15]: (non-subtraction-free) Cyclic Reduction.
- **QZ** QZ algorithm: orthogonal transformations — well-known linear algebra workhorse.
- **NP** our new algorithm.

Problem	KK	AS	LN	QZ	NP
NP15	2.7e-12	2.5e-07	2.9e-13	1.8e-12	**1.7e-16**
NP15s	1.3e-12	2.2e-07	-	6.2e-13	**1.8e-16**
rand8	2.8e-15	1.5e-15	1.6e-15	2.4e-15	**2.7e-16**
rand8s	2.9e-15	1.8e-13	-	2.3e-15	**3.1e-16**
rand20	4.4e-15	9.6e-14	5.6e-15	4.8e-15	**3.0e-16**
rand20s	3.2e-15	3.0e-12	-	4.1e-14	**1.1e-15**
rand50	5.9e-15	4.0e-14	4.0e-14	5.6e-14	**6.9e-16**
rand50s	5.6e-14	1.2e-10	-	3.5e-14	**5.2e-16**
imb8	9.7e-12	1.9e-09	1.1e+00	**7.1e-13**	9.0e-13
imb8s	2.6e-14	1.3e-08	-	1.3e-12	**1.1e-15**
imb20	4.6e-11	2.1e-07	3.2e-04	1.1e-09	**9.1e-12**
imb20s	4.4e-12	6.9e-06	-	5.9e-12	**4.0e-13**
imb50	**2.0e-10**	9.8e-06	7.2e-01	1.0e-08	8.3e-10
imb50s	2.0e-10	3.3e-05	-	1.0e+00	**2.6e-13**

Table: Forward error $\frac{\|\tilde{X}-X\|}{\|X\|}$

Error on $U = \begin{bmatrix} I & \Psi \end{bmatrix}$

Problem	KK	AS	LN	QZ	NP
NP15s	2.3e-15	1.8e-11	-	2.8e-15	**1.3e-16**
rand8s	1.2e-14	3.7e-13	-	**2.4e-15**	2.5e-15
rand20s	7.1e-15	7.7e-11	-	6.7e-14	**2.1e-15**
rand50s	3.4e-14	3.5e-09	-	5.3e-14	**4.7e-16**
imb8s	8.3e-15	5.2e-09	-	1.1e-11	**5.2e-15**
imb20s	1.4e-10	1.9e-08	-	**2.8e-11**	4.0e-11
imb50s	**6.9e-11**	9.0e-09	-	1.0e-04	6.1e-08

Table: Forward error $\frac{\|\tilde{\Psi}-\Psi\|}{\|\Psi\|}$

Conclusions and open points

- Subtraction-free, componentwise accurate algorithm for MMBM. [Nguyen P. arXiv:1605.01482]
- There's also [Nguyen P. '15] for fluid queues.
- Similar to [Ramaswami '99] QBD construction but for MMBM.
- Future plan: remove the 2^k factor in the error for CR.
- Ideas from ODEs (index reduction, stability conditions) and linear algebra (shift technique, invariant pairs).

Conclusions and open points

- Subtraction-free, componentwise accurate algorithm for MMBM. [Nguyen P. arXiv:1605.01482]
- There's also [Nguyen P. '15] for fluid queues.
- Similar to [Ramaswami '99] QBD construction but for MMBM.
- Future plan: remove the 2^k factor in the error for CR.
- Ideas from ODEs (index reduction, stability conditions) and linear algebra (shift technique, invariant pairs).

Thanks for your attention!
Questions?