# Probabilistic interpretations and accurate algorithms

## for stochastic fluid models

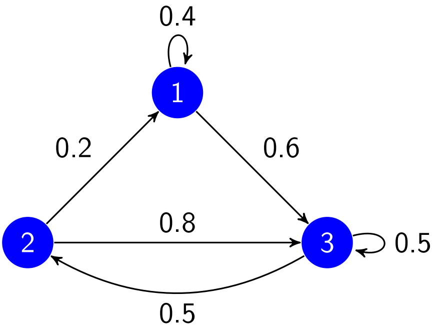

Federico Poloni[1]
Joint work with Giang T. Nguyen[2]

[1]U Pisa, Computer Science Dept.
[2]U Adelaide, School of Math. Sciences


U Adelaide – 23rd January 2014

# Goal of this research

- Markovian Models of queues/buffers — computing stationary measures
- Many algorithms have multiple interpretations in different "languages", e.g. Newton's method [Bean, O'Reilly, Taylor '05]
  - Linear algebra: invert matrices, compute eigenvalues
  - Probability: $M_{ij} = \mathbb{P}\,[\text{something}]$
  - Differential equations (sometimes): discretize $\frac{d}{dt} f(t) = \dots$
- However, the fastest algorithm available, doubling, is 100% abstract linear algebra
- We try to gain more probabilistic insight on what it does + turn this insight into better accuracy



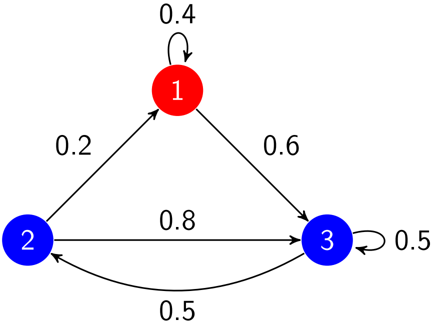

$$P = \begin{bmatrix} 0.4 & 0 & 0.6 \\ 0.2 & 0 & 0.8 \\ 0 & 0.5 & 0.5 \end{bmatrix}$$

$P_{ij} = \mathbb{P}\left[\text{transition } i \to j\right]$

If $\pi_t = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix}$ = probabilities of being in the states at time $t$

Time evolution: $\pi_{t+1} = \pi_t P$

# Probabilistic interpretations: censoring




$$P = \begin{bmatrix} a & b^T \\ c & D \end{bmatrix}$$

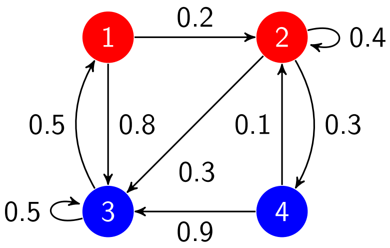
Censoring: ignore time spent in state 1, consider only states $S = \{2, 3\}$

Transitions $2 \leftrightarrow 3$ may happen directly or through state 1.

## Censored Markov chain

$$\widehat{P} = \underbrace{D}_{S\to S} + cb^T + \underbrace{c}_{S\to 1}\underbrace{a}_{1\to 1}\underbrace{b^T}_{1\to S} + ca^2 b^T + \cdots = D + c(1-a)^{-1} b^T$$

# Probabilistic interpretations: censoring II




$$P = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

Can also censor multiple states at the same time

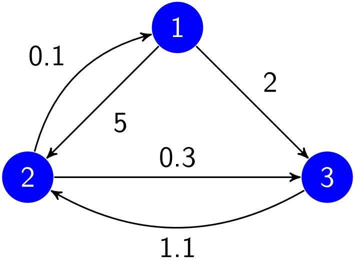
## Censored Markov chain

$$\widehat{P} = D + CB + CAB + CA^2B + \cdots = D + C(I - A)^{-1}B$$

Schur complementation on $I - P$

# Continuous-time Markov chains

Continuous time; transition probability = exponential distribution with parameter $Q_{ij}$

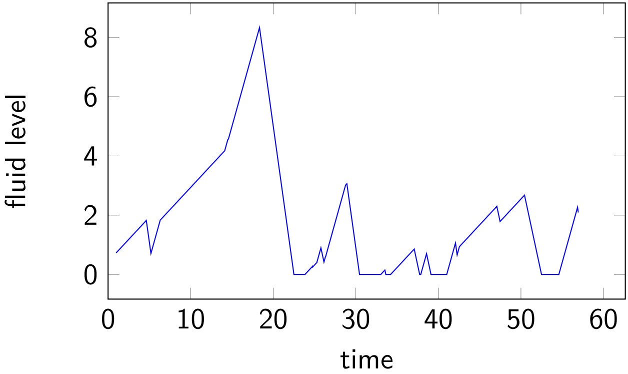



$$Q = \begin{bmatrix} -7 & 5 & 2 \\ 0.1 & -0.4 & 0.3 \\ 1.1 & 0 & -1.1 \end{bmatrix}$$

Evolution follows $\frac{d}{dt}\pi(t) = \pi(t)Q$, or equivalently $\pi(t) = \pi_0 \exp(tQ)$

# Fluid queues

Queue, or buffer: “infinite-size bucket” in which fluid (or data) flows in or out at a rate $c_i$, depending on the state of a continuous-time Markov chain




We want the “long-time behavior” (stationary probabilities) of the fluid level, density vector $f(x)$ of $P[\text{level} = x]$

# Stationary density and ODEs

**Theorem** [Karandikar, Kulkarni '95, Da Silva Soares Thesis]

The stationary density vector satisfies

$$\frac{d}{dx} f(x) C = f(x) Q$$

$C = \mathrm{diag}(c_1, \ldots, c_n)$

Different ways to see it...

Differential equations:

The solutions of this linear ODE are linear combinations of the "elementary solutions"

$$f^{(i)}(x) = u_i \exp(x \lambda_i),$$

with $(u_i, \lambda_i)$ (left) eigenvector-eigenvalue pairs of $QC^{-1}$

Throw in boundary conditions. Stable ones? Keep only $\Re \lambda < 0$.

# Invariant probabilities and linear algebra

**Theorem** [Karandikar, Kulkarni '95, Da Silva Soares Thesis]

The invariant density satisfies

$$\frac{d}{dx} f(x) C = f(x) Q$$

$C = \operatorname{diag}(c_1, \ldots, c_n)$

Different ways to see it...

Numerical linear algebra

Find the stable invariant subspace of $QC^{-1}$, i.e.,

$$\mathcal{U} = \operatorname{span}(u_1, u_2, \ldots, u_h)$$

$u_1, \ldots, u_h$ with eigenvalues in the left complex half-plane

# Invariant probabilities and probability

**Theorem** [Karandikar, Kulkarni '95, Da Silva Soares Thesis]

The invariant density satisfies

$$\frac{d}{dx} f(x) C = f(x) Q$$

$C = \operatorname{diag}(c_1, \dots, c_n)$

Order states so that $C$ has positive elements on top; a basis for $\mathcal{U}$ are the rows of

$$\begin{bmatrix} I & -\Psi \end{bmatrix}$$

for the "first return probabilities" $\Psi$:

$$\Psi_{ij} = P[0 \twoheadrightarrow 0 \text{ after some time (for the first time), and state } i \twoheadrightarrow j]$$

# Structured doubling algorithm

There's a linear algebra algorithm to solve this:

## Structured doubling algorithm

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

$E_0, F_0, G_0, H_0 =$ more unilluminating formulas

# What's going on

What's going on: SDA is related to scaling and squaring

- To look for stable modes, build $\exp(t\mathcal{H})$ for a large $t$, look at what subspace "goes to 0" and what "to $\infty$"
- Choose initial step-length $\gamma$, start from first-order arccurate

$$S = \exp(\gamma\mathcal{H}) \approx (I + \frac{\gamma}{2}\mathcal{H})(I - \frac{\gamma}{2}\mathcal{H})^{-1}$$

- Then keep squaring: $\exp(2^k\gamma\mathcal{H}) = \left(\left(\dots\left(S^2\right)^2\dots\right)^2\right)^2$
- Keep iterates in the form $S^{2^k} = \begin{bmatrix} I & -G_k \\ 0 & F_k \end{bmatrix}^{-1} \begin{bmatrix} E_k & 0 \\ -H_k & I \end{bmatrix}$

  Why?
  - A method to prevent instabilities from large entries
  - Natural in a different problem in control theory
  - It works!

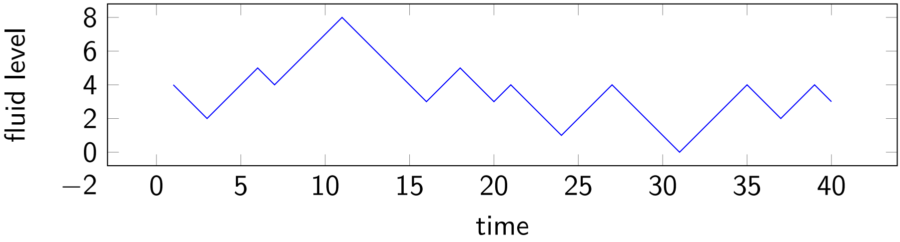
# Probabilistic interpretation for SDA — the grand scheme

We construct a discrete-time process with the same behavior

1. Rescaling
2. Discretization
3. Doubling

Rescaling: (state-dependent) change of time scale to get $\pm 1$ slopes

Well understood probabilistically; linear algebra: diagonal similarity




Discrete time and $\pm 1$ rates $\implies$ discrete space "level"

# Discretization

Probabilists often use $P = I + \gamma Q$, $\gamma > 0$, as a discretization of the continuous-time Markov chain $Q$ (uniformization)

Differential equations : explicit Euler's method!

$$\text{discretize } \frac{d}{dt} f(t) = f(t)Q \text{ to } f_{t+1} = f_t(I + \gamma Q)$$

It turns out that something slightly different happens in SDA:

**Theorem (similar to [P., Reis, preprint], [P., thesis])**

$$\begin{bmatrix} E_0 & G_0 \\ H_0 & F_0 \end{bmatrix} = (I + \gamma Q)(I - \gamma Q)^{-1}$$

Differential equations Midpoint method with stepsize $\frac{\gamma}{2}$

Probability on/off switch; observe the queue only if it is on

We encountered before $(I + \gamma \mathcal{H})(I - \gamma \mathcal{H})^{-1}$, but on $\mathcal{H} = QC^{-1}$ instead

# Doubling step

So, $\begin{bmatrix} E_0 & G_0 \\ H_0 & F_0 \end{bmatrix}$ is a discrete-time Markov chain.

**Observation**

After one doubling step

$$\begin{bmatrix} E_1 & G_1 \\ H_1 & F_1 \end{bmatrix}$$

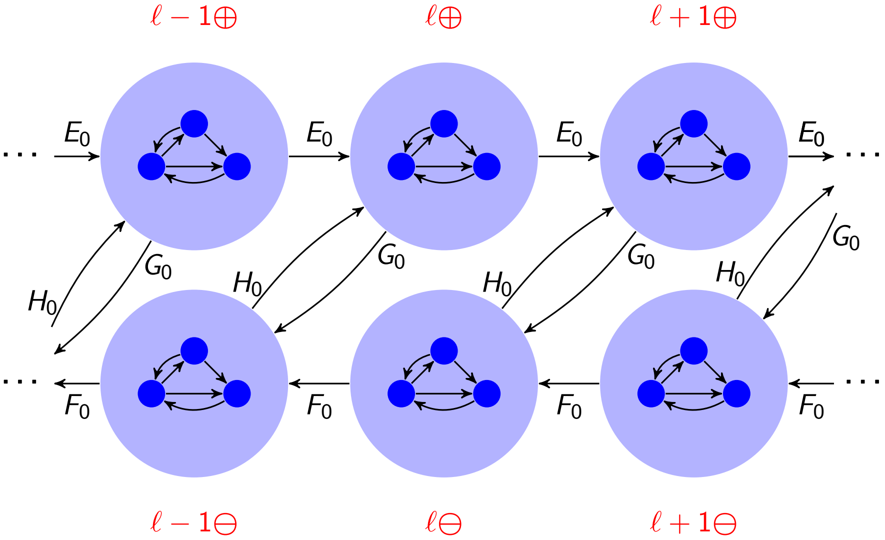
is still the transition matrix of a DTMC

What do its states represent?

"States" of the queuing model $= (\ell, s) =$ (level, state of the DTMC)

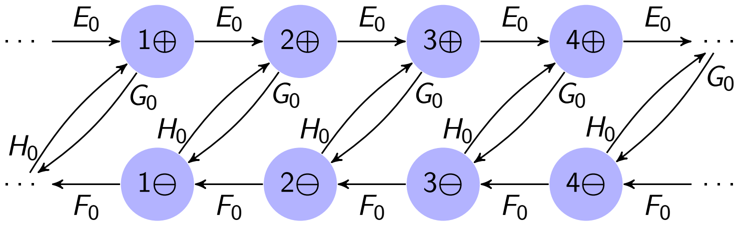
- some states are associated to a $+1$ rate, we call them $\oplus$
- resp. $-1$ rate, $\ominus$

# Levels and states



# More states

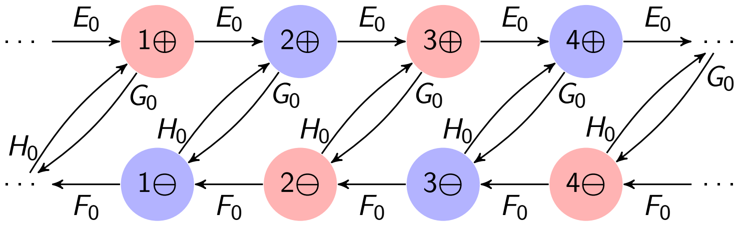
- in a state with $\oplus$ rate, $E_0$ or $G_0$ is applied
- in a state with $\ominus$ rate, $F_0$ or $H_0$




$$E_{k+1} = E_k(I - G_k H_k)^{-1} E_k$$
$$F_{k+1} = F_k(I - H_k G_k)^{-1} F_k$$
$$G_{k+1} = G_k + E_k(I - G_k H_k)^{-1} G_k F_k$$
$$H_{k+1} = H_k + F_k(I - H_k G_k)^{-1} H_k E_k$$

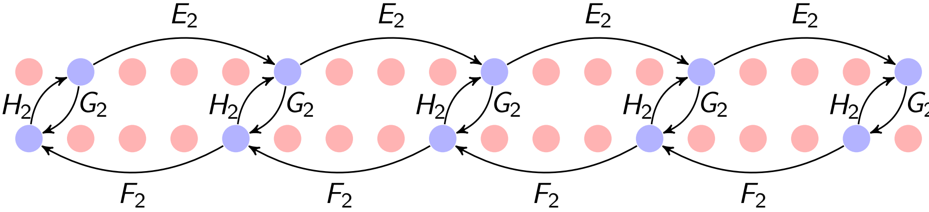
# The solution

Censor in this way:




$$E_{k+1} = E_k(I - G_k H_k)^{-1} E_k$$
$$F_{k+1} = F_k(I - H_k G_k)^{-1} F_k$$
$$G_{k+1} = G_k + E_k(I - G_k H_k)^{-1} G_k F_k$$
$$H_{k+1} = H_k + F_k(I - H_k G_k)^{-1} H_k E_k$$

# Structured doubling algorithm: probabilistic interpretation




## Result

$$E_k = P[0\oplus \twoheadrightarrow 2^k \text{ before } \twoheadrightarrow -1]$$

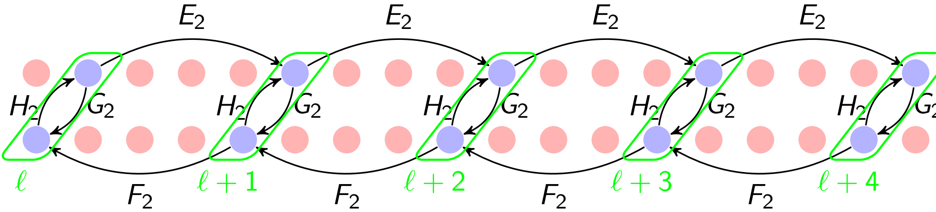
$$G_k = P[0\oplus \twoheadrightarrow -1 \text{ before } \twoheadrightarrow 2^k]$$

$$F_k = P[0\ominus \twoheadrightarrow -2^k \text{ before } \twoheadrightarrow 1]$$

$$E_k = P[0\ominus \twoheadrightarrow 1 \text{ before } \twoheadrightarrow -2^k]$$

$$\lim_{k\to\infty} G_k = P[0\oplus \twoheadrightarrow -1 \text{ before "escaping to infinity"}] = \Psi$$

# Tilt your head diagonally

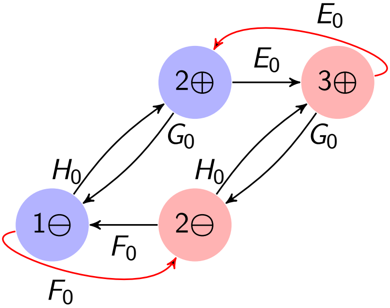



$$\text{SDA} \iff \text{Cyclic reduction on QBD} \left( \begin{bmatrix} E_k & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & H_k \\ G_k & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & F_k \end{bmatrix} \right)$$

Relation appeared (only algebraically) in [Bini, Meini, P., 2010]

# Work on a torus

Let's "wrap the chain on itself" after two steps




Transitions probabilities in this queue are the same as in the big one

$$\begin{bmatrix} E_1 & G_1 \\ H_1 & F_1 \end{bmatrix} = \text{Schur compl of first two blocks in } I - \begin{bmatrix} 0 & G_0 & E_0 & 0 \\ H_0 & 0 & 0 & F_0 \\ E_0 & 0 & 0 & G_0 \\ 0 & F_0 & H_0 & 0 \end{bmatrix}$$

# Part II

# Componentwise accurate algorithms

# Componentwise accurate linear algebra

Traditional algorithms are normwise accurate: $\tilde{v} = v + \varepsilon \|v\|$

Suppose $v = \begin{bmatrix} 1 & 10^{-8} \end{bmatrix}$ and $\varepsilon = 10^{-8}$

$$\tilde{v} = \Big[\underbrace{1+\varepsilon}_{\text{ok}}, \quad \underbrace{10^{-8}+\varepsilon}_{\text{junk}}\Big]$$

Here we want componentwise accurate algorithms

$$\tilde{v} = \left[1+\varepsilon, \quad 10^{-8} + 10^{-8}\varepsilon\right]$$

$$|v - \tilde{v}| \leq \varepsilon v \quad \text{(with } \leq, |\cdot| \text{ on components)}$$

Recent componentwise error analysis for doubling [Xue et al., '12]

Algorithms almost ready, but a detail is missing

# Subtraction-free computations

Error amplification in floating point op's (think "loss of significant digits")

- bounded by 1 for $\oplus$ (of nonnegative numbers), $\odot, \oslash$
- can be arbitrarily high for $\ominus$, e.g., 1.000000000 − 0.9999999999

**Solution**

Avoid all the minuses!

Most come from Z-matrices, i.e., matrices with sign pattern

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

# Triplet representations

Gaussian elimination & inversion of Z-matrices: cancellation only on diagonal entries

**Algorithm (GTH trick** [Grassmann et al, '85?]**)**

Let $Z$ be a Z-matrix. If we know its off-diagonal entries and $v > 0, w \geq 0$ such that $Zv = w$, then we can run subtraction-free Gaussian elimination

$(\mathrm{offdiag}(Z), v, w)$ is called triplet representation

GE knowing a triplet representation always componentwise perfectly stable!

**Theorem** [Alfa, Xue, Ye '02]

The GTH algorithms to solve a linear system $Zx = b$, given $(P, v, w)$ and $b$ exact to machine precision $\mathbf{u}$, returns $\tilde{x}$ such that

$$|x - \tilde{x}| \leq \frac{4}{3} n^3 \mathbf{u}\, x + \text{lower order terms}$$

# No condition number?

No condition number! How is this even possible? Example:

$$\begin{bmatrix} 1 & -1 \\ -1 & 1+\varepsilon \end{bmatrix}^{-1} = \varepsilon^{-1} \begin{bmatrix} 1+\varepsilon & 1 \\ 1 & 1 \end{bmatrix}$$

No way to get around (unstable) subtraction $(1+\varepsilon) - 1$
A triplet representation (blue entries):

$$\begin{bmatrix} 1 & -1 \\ -1 & 1+\varepsilon \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon \end{bmatrix}$$

It already contains $\varepsilon$, no need to compute it

The catch: a triplet representation is ill-conditioned to compute from the matrix entries

But what if we had it for free?

# Using triplet representations

## Structured doubling algorithm

$$E_{k+1} = E_k(I - G_k H_k)^{-1} E_k$$
$$F_{k+1} = F_k(I - H_k G_k)^{-1} F_k$$
$$G_{k+1} = G_k + E_k(I - G_k H_k)^{-1} G_k F_k$$
$$H_{k+1} = H_k + F_k(I - H_k G_k)^{-1} H_k E_k$$
$$\begin{bmatrix} E_0 & G_0 \\ H_0 & F_0 \end{bmatrix} = (I + \gamma Q)(I - \gamma Q)^{-1}$$

Missing ingredient from [Xue et al, '12]:

deriving triplet representations using stochasticity of $\begin{bmatrix} E_k & G_k \\ H_k & F_k \end{bmatrix}$

## Theorem

$$(I - G_k H_k)\underline{\mathbf{1}} = (H_k E_k + F_k)\underline{\mathbf{1}} \qquad (I - H_k G_k)\underline{\mathbf{1}} = (G_k F_k + E_k)\underline{\mathbf{1}}$$

# After $\Psi$: matrix exponentials

After computing $\Psi$, invariant measure given by

$$f(x) = v \exp(-Kx)$$

Z-matrix $K$ and row vector $v \geq 0$ computed explicitly from $\Psi$

Now, only matrix exponential needed — lots of literature on it

We use a subtraction-free algorithm [Xue et al., '08; Xue et al., preprint; Shao et al., preprint]

Idea:

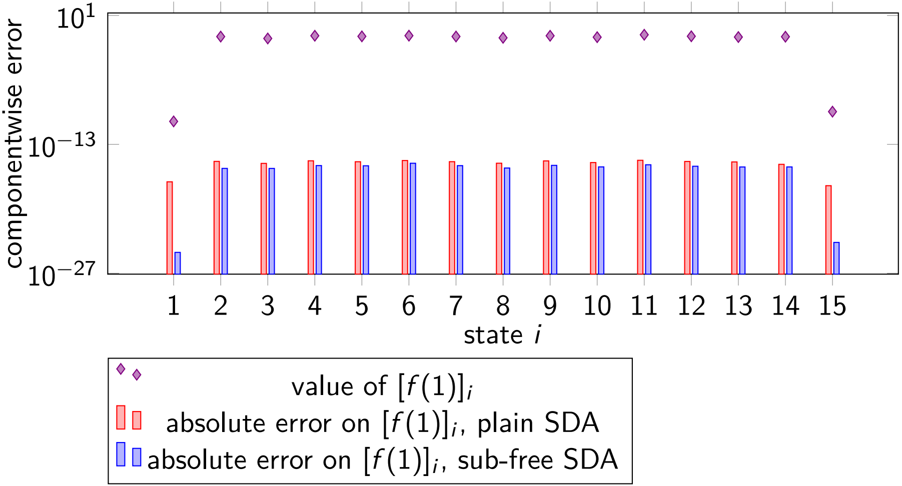
1. shift to reduce to a positive matrix: $\exp(A + zI) = e^z \exp(A)$
2. truncated Taylor series + scaling and squaring:

$$\exp(2^k A) = \left( \left( \dots \left( I + A + \frac{A^2}{2!} \right)^2 \dots \right)^2 \right)^2$$

(Thanks N Higham, MW Shao for useful discussions)

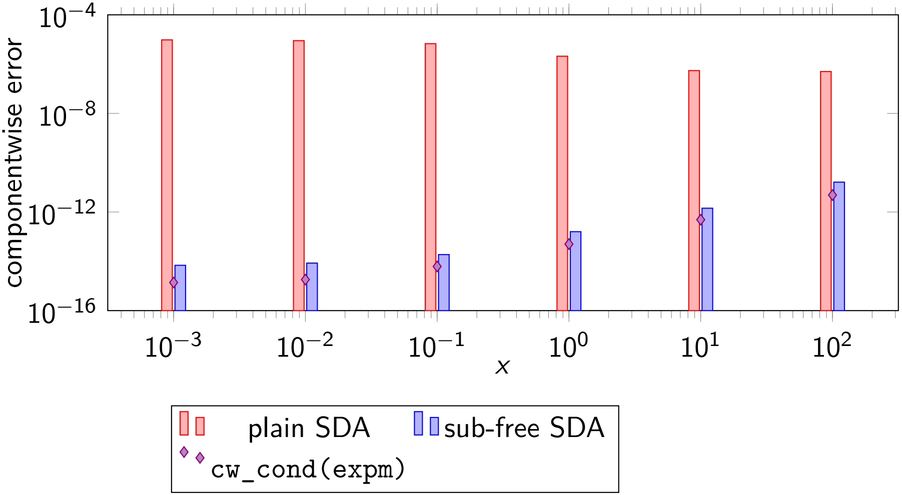
# Numerical results

Figure : Error on the single components. $15 \times 15$ model with two "hard-to-reach" states



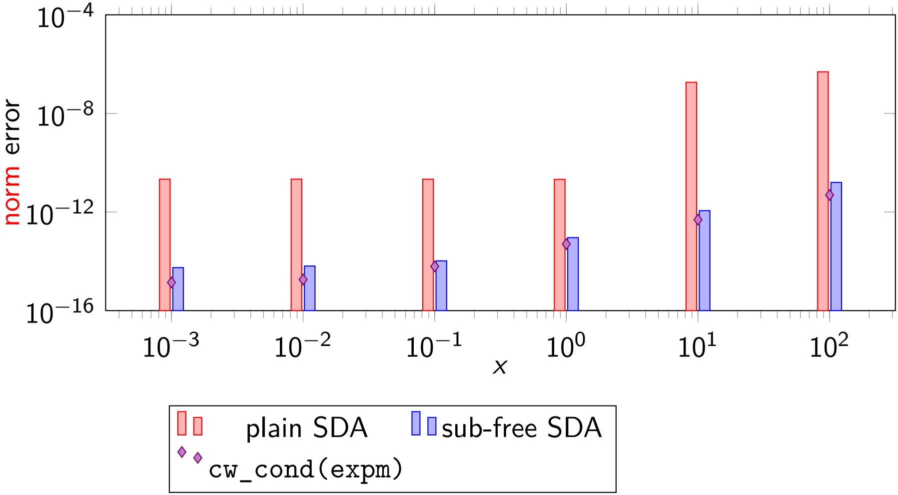
# Numerical experiments

Figure : pdf $f(x)$ in several points



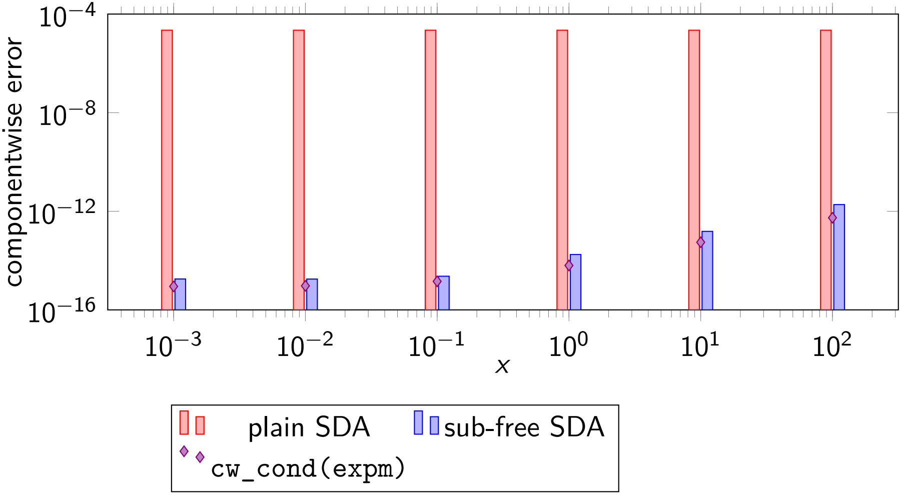
# Numerical experiments

Figure : pdf $f(x)$ in several points



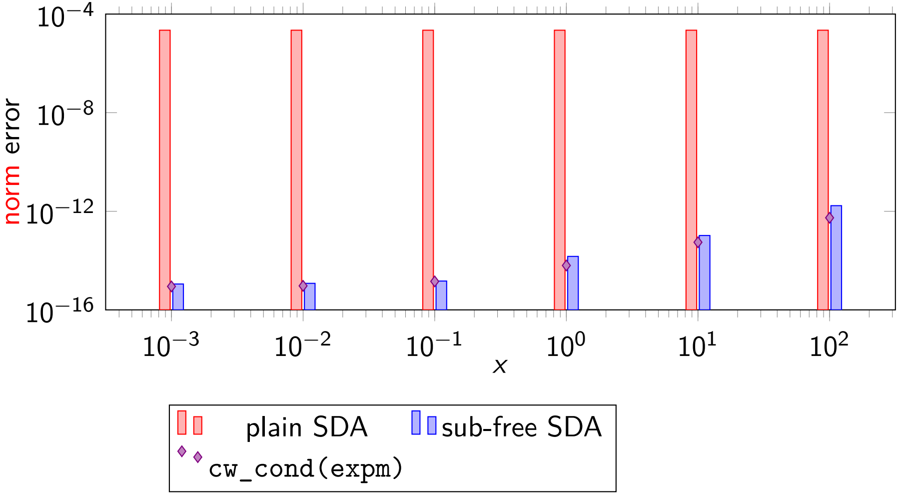
# Numerical experiments

Figure : $10 \times 10$ model with states "each slightly harder to reach"



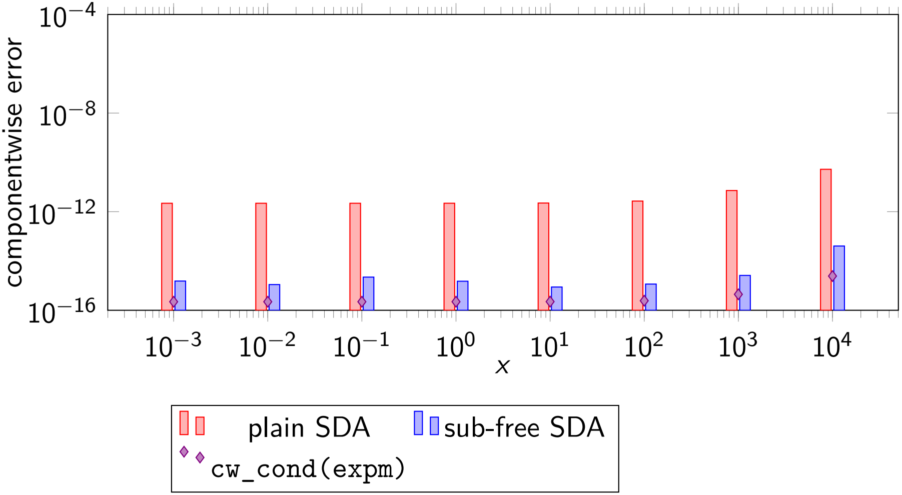
# Numerical experiments

Figure : $10 \times 10$ model with states "each slightly harder to reach"



# Numerical experiments

Figure : Very simple test queue [Bean, O'Reilly, Taylor '05, Example 3]



# Numerical experiments

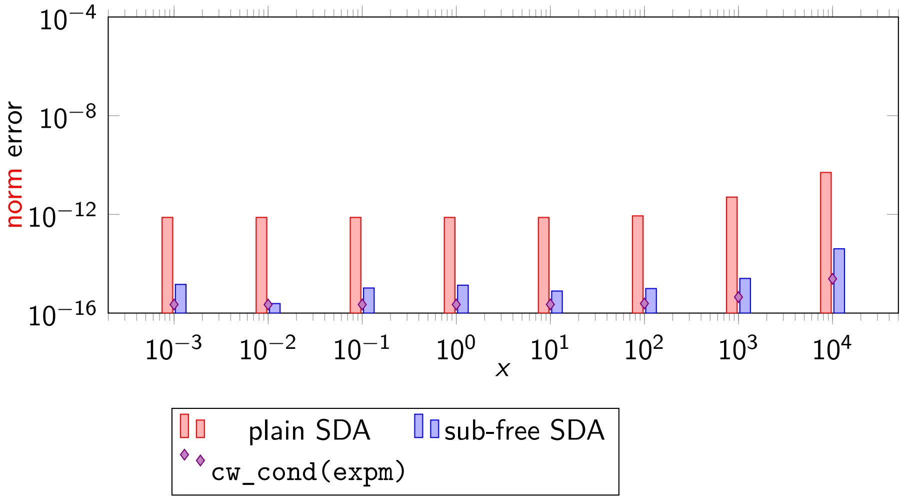



Figure : Very simple test queue [Bean, O'Reilly, Taylor '05, Example 3]

# Conclusions

- Algorithms: now with triplets!
- Improved understanding of doubling on the probabilistic, differential-eq and linear algebra levels
- Step 1 on the way to get new algorithms
- Probabilists prefer to use something that they “see”
- Next targets: second-order models (Brownian motion), finite-horizon

# Conclusions

- Algorithms: now with triplets!
- Improved understanding of doubling on the probabilistic, differential-eq and linear algebra levels
- Step 1 on the way to get new algorithms
- Probabilists prefer to use something that they "see"
- Next targets: second-order models (Brownian motion), finite-horizon

Thanks for your attention!