Eidgenössische Technische Hochschule Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

A stochastically generated preconditioner for stable matrices

F.M. Buchmann and W.P. Petersen

Research Report No. 2002-24
October 2002

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

A stochastically generated preconditioner for stable matrices

F.M. Buchmann and W.P. Petersen

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Research Report No. 2002-24 October 2002

Abstract

In this paper we simulate the Ornstein-Uhlenbeck process (OUP) to generate an approximate inverse of any real valued stable matrix. Matrix A being stable means that analytically the OUP will converge to a stationary Gaussian process in n dimensions with a covariance $(2A)^{-1}$. If the eigenvalues of A are widely separated in absolute value, however, the stiffness of the simulated linear stochastic differential equations must be considered. Hence, we consider a splitting scheme to permit large step sizes but keep convergence. Methods are described for both symmetric and non-symmetric matrices. Our preconditioner is also tested in the symmetric positive definite case by its effect on the convergence of conjugate gradient iterations.

1 INTRODUCTION

In this paper we use a novel approach to preconditioning a linear system of the usual form $Ax = b$. We treat only the case that $(n \times n)$ matrix A is real valued. The remainder of this introduction considers the symmetric positive definite problem; then we generalize this to the non-symmetric stable case (Section 3.2). In what follows, stability ($A > 0$ here) means that all of the eigenvalues of A are in the right-half plane [5].

The task of preconditioning is to find an approximate inverse of the matrix A. Namely, we are looking for a matrix $M \approx A^{-1}$. Our procedure involves an n-dimensional stochastic process $X(t)$ (a vector) satisfying the system of n stochastic differential equations (SDEs)

$$dX = -AXdt + dW \quad \text{with} \quad X(0) = X_0. \tag{1}$$

Here, $W = \{W\}_{t\geq 0}$ is a standard n-dimensional Brownian motion and the initial condition X_0 is independent of the driving process W. Multiplying both sides of (1) by the matrix valued integrating factor $\exp(At)$ and a little rearrangement, one gets a formal solution to (1),

$$X(t) = e^{-tA}X_0 + \int_0^t e^{-(t-s)A}dW(s), \quad 0 \leq t < \infty. \tag{2}$$

This is an n dimensional Ornstein-Uhlenbeck process (e.g. [10], exercise 3.14, or in the notes by Carmona in [2]). To extract information from this formal solution, we need expectation values of the increments of $dW(s)$ [6]:

$$\mathbb{E}\left[dW^k(u)dW^l(v)\right] = \delta^{kl}\delta(u-v)dudv. \tag{3}$$

On the right-hand side, δ^{kl} is the Kronecker delta (= 1 if $k = l$, zero otherwise), and $\delta(u-v)$ is the Dirac delta function (zero everywhere except $u = v$, but with unit area).

Our purposes are well served by the deterministic initial conditions $X_0^i = 0$, although these are not essential to the argument: Because $A > 0$, from (2) we see that the influence of the initial conditions soon exponentially decays to zero. For these $X(0) = 0$ initial conditions, we get using (3) the expectation $\mathbb{E}\left[X(t)\right] = 0$ and a matrix valued correlation function

$$\mathbb{E}\left[X(s)X^T(t)\right] = \int_0^{s\wedge t} e^{-((t+s)-2u)A}du,$$

where $s \wedge t = \min(s,t)$. In particular, by setting $t = s$ we get the covariance

$$\mathbb{E}\left[X(t)X^T(t)\right] = \int_0^t e^{-2(t-u)A}du. \tag{4}$$

We now show this gives our approximate inverse to A in the limit that t becomes large.

Lemma 1.1 *The integral expression of (4) is*

$$\int_0^t e^{-2(t-u)A}du = \frac{1}{2}A^{-1}\{I - e^{-2tA}\} \tag{5}$$

where I is the unit matrix.

Proof: *The matrix* $\exp(-2tA)$ *may be pulled out of both sides of (5), leaving*

$$L(t) = \int_0^t e^{2Au} du \tag{6}$$

$$R(t) = \frac{1}{2} A^{-1}(e^{2At} - I) \tag{7}$$

It is easy to see that $dL(t) = dR(t)$ *by taking derivatives of* L *and* R *in (6) and (7). Since* $L(0) = R(0) = 0$*, by the uniqueness of the matrix valued solution to this ordinary differential equation, we get* $L(t) = R(t)$*. Multiplying* L, R *by* $\exp(-2tA)$*, we have (5).* □

Now let t become large in ((4), respectively (5)). Because $A > 0$,

$$\lim_{t\to\infty} \mathbb{E}\left[X(t)X^T(t)\right] = \frac{1}{2} A^{-1}. \tag{8}$$

Explicitly writing out the indices, this is

$$2 \lim_{t\to\infty} \mathbb{E}\left[X^i(t)X^j(t)\right] = (A^{-1})^{ij}, \quad i, j = 1, \ldots n.$$

In practice, we select a set of such indices (say p of these) to form a possibly incomplete inverse as preconditioner.

Hence, asymptotically ($t \to \infty$) there exists a unique invariant probability measure, μ, for the process (2) which is Gaussian with mean 0 and covariance $(2A)^{-1}$:

$$\mu(dx) = \frac{1}{((2\pi)^n (\det 2A)^{-1})^{\frac{1}{2}}} e^{-\langle Ax, x\rangle} d^n x.$$

Following Talay [11], we use a numerical method for approximating this invariant law for a stochastic process. In his paper, a system of stochastic differential equations $dX = b(X)dt + \sigma(X)dW$ was discretized using a scheme of weak order 2 to get a chain $X_1, X_2, \ldots, X_N$. He approximated an integral $T^{-1}\int_0^T f(X(t))dt$ by $1/(Nh)\sum_{i=1}^N hf(X_i)$. Our approach is similar, but instead of this discrete sum, we extend the system (1) by equations of the form $dZ^{ij} = X^i(t)X^j(t)dt$, which will be simultaneously integrated along with the system (1). It is important to note that the differential equations for the extensions do not directly contain the Brownian motion (Wiener process) W. We will see that the integration rules for the variables Z^{ij} are somewhat simpler than those for X because the Brownian motion terms do not appear explicitly. The choice of which Z^{ij} to actually compute may well depend on the problem: for example, one might choose an incomplete inverse with only those i, j values of the sparsity pattern of matrix A.

An approximation $M = (M^{ij}) \approx A^{-1} = ((A^{-1})^{ij})$ is generated by limiting the integration to $T < \infty$ as a long time average in a finite number, N, of steps. To accelerate convergence to a stationary state (of the approximation $\overline{\mu}^h \approx \mu$) we will also simulate a small sample of N_P paths of n-vectors $X^{(k)}$ and compute the average over the corresponding $Z^{(k)}$. This yields an N_P-sample average,

$$\frac{1}{T}\int_0^T X(t)X^T(t)dt \approx \frac{1}{N_P}\sum_{k=1}^{N_P} \frac{Z_N^{(k)}}{Nh},$$

where the $Z_N^{(k)}$ are the approximations obtained by discretizing each independent $dZ = X^i X^j dt$ system numbered (k). A nice benefit of this approach is that we can use the sample variance (25) over the N_P parallel realizations to get an heuristic measure of statistical convergence.

We will apply weak approximations for both the system (1) for each realization (k) of X (that is, $X^{(k)}$) and the corresponding integrals in the extended system

$$\frac{T}{2} M^{(k)} = Z^{(k)} = \int_0^T X^{(k)}(t)(X^{(k)})^T(t)dt.$$

In the details, we will examine the simplest numerical scheme, the Euler method, and compare it with two higher order schemes, one of which is of Runge–Kutta type. These latter procedures will be weak order two. For a comprehensive review on numerical methods for SDEs see [6, 8].

We conclude this section by summarizing our notation. We use superscripts for vector and matrix entries, for example $A = (A^{ij})$ or $x = (x^i) = (x^1, \ldots, x^n)^T$. We use superscripts here because subscripts index the steps of our Markov processes ($X_k = X(t_k)$). We also use the summation convention that repeated indices imply summation: $a^k b^k \equiv \sum_k a^k b^k$.

2 NUMERICAL SCHEMES

In the introduction we established that for stable matrices, OUP (2) converges to a stationary process whose covariance gives us our desired preconditioner. From the ergodic theorem, we may compute the covariance by long time averages:

$$\lim_{t\to\infty} \mathbb{E}\, X^i(t)X^j(t) = \overline{Z^{ij}} = \lim_{T\to\infty} \frac{1}{T}\int_0^T X^i(t)X^j(t)dt, \tag{9}$$

where $X(t)$ is one realization ($N_P = 1$) of SDE system (1). In our examples, we choose a subset P of the possible $1 \le i, j \le n$ which index matrix elements of the approximate inverse $M = (M^{ij})$; there will be $p = |P|$ of these. Then we have a system of $n + p$ SDEs

$$\begin{aligned} dX &= -AX dt + dW, \quad X^i(0) = 0, \quad i = 1, \ldots n \\ dZ^{ij} &= X^i X^j dt, \qquad\quad Z^{ij}(0) = 0, \quad (i,j) \in P. \end{aligned} \tag{10}$$

Our approximation to (9) is $(A^{-1})^{ij} \approx \overline{Z^{ij}} = Z^{ij}(T)/T$ for large T. The next section deals with numerical simulations of $X(t)$ and $Z(t)$ appearing in the system (10).

2.1 The basic scheme

The simplest procedure for simulating (10) is an **Euler scheme**. At time t_k, we advance to step $t_{k+1} = t_k + h$ by the formulae

$$\begin{aligned} X_{k+1} &= X_{euler} = (I - hA)X_k + \Delta W \\ Z_{k+1}^{ij} &= Z_k^{ij} + hX_k^i X_k^j \end{aligned} \tag{11}$$

where ΔW is some model for the increments of the Brownian motion. At each time step, n elements of ΔW are generated. To the desired weak order [6, 8], one wants an approximately Gaussian increment. It is not expensive to generate Gaussian increments, so we choose

$$\Delta W^j = \sqrt{h}\,\xi^j, \tag{12}$$

where the ξ^j are univariate, zero mean, and mutually independent normally distributed random variables. On modern computers, it is often faster to generate these by the Box-Muller procedure (see [3], section 4.2.1) than the polar method [7] which uses acceptance/rejection. We have

$$\begin{aligned} \mathbb{E}\,\Delta W^i &= 0 \\ \mathbb{E}\,\Delta W^i \Delta W^j &= h\delta^{ij} \\ \mathbb{E}\,\Delta W^i \Delta W^j \Delta W^l \Delta W^m &= h^2(\delta^{ij}\delta^{lm} + \delta^{il}\delta^{jm} + \delta^{im}\delta^{jl}). \end{aligned}$$

Actually, the last relation follows from the second because ΔW^j are independent variance h Gaussians. All odd order moments are zero. The weak accuracy of the Euler method is 1 [6, 8].

Next, let us explore higher-order procedures. An explicit 2nd order method is a Heun or 2nd order **Runge-Kutta scheme**,

$$\begin{aligned} X_{euler} &= (I - hA)X_k + \Delta W, \\ X_{k+1} &= X_k - \frac{h}{2}(AX_k + AX_{euler}) + \Delta W, \\ Z^{ij}_{k+1} &= Z^{ij}_k + \frac{h}{2}(X^i_k X^j_k + X^i_{euler} X^j_{euler}). \end{aligned} \tag{13}$$

This procedure is an explicit trapezoidal rule. Expanding this RK scheme in a Taylor expansion to $O(h^2)$ accuracy, that is, ignoring terms of $O(h^{5/2})$ and higher in the stochastic Taylor series, we get the **Taylor scheme** (14). In this expansion, $h(\Delta W^i \Delta W^j) \equiv h^2\delta^{ij}$ in statistics to $O(h^2)$. This follows because $\mathbb{E}\,\Delta W^i \Delta W^j = h\delta^{ij}$. We get

$$X_{k+1} = X_k - hAX_k - \frac{h}{2}A\,\Delta W + \frac{h^2}{2}A^2 X_k + \Delta W,$$

$$\begin{aligned} Z^{ii}_{k+1} = Z^{ii}_k + h(X^i_k)^2 + hX^i_k\,\Delta W^i \qquad & (i = j \text{ case}) \\ + h^2(\frac{1}{2} - (AX_k)^i X^i_k), & \end{aligned} \tag{14}$$

$$\begin{aligned} Z^{ij}_{k+1} = Z^{ij}_k + hX^i_k X^j_k + \frac{h}{2}(X^i_k\,\Delta W^j + \Delta W^i X^j_k) \quad & (i \neq j \text{ case}) \\ - \frac{h^2}{2}((AX_k)^i X^j_k + X^i_k (AX_k)^j). & \end{aligned}$$

The X-updates in (13) and (14) are identical since the OUP is linear. The Z^{ij} equation is quadratic, however, so these two methods will not give the same Z's.

2.2 Semi-implicit schemes

Because A only appears on the right hand side of (11) in the updates for X, large variations in the sizes of the eigenvalues of A can force the stepsize down to keep the R term in (15) contracting. This means more steps are necessary to reach convergence to a stationary process.

Quite generally, the update for X can be written

$$X_{k+1} = R(h, A)X_k + S(h)\xi_{k+1} \tag{15}$$

where

$$R(h, A) = \begin{cases} I_n - hA, & \text{for the scheme (11)} \\ I_n - hA + \frac{h^2}{2}A^2, & \text{for the schemes (13, 14),} \end{cases}$$

and

$$S(h) = \begin{cases} \sqrt{h}, & \text{for the scheme (11)} \\ \sqrt{h}I_n - \frac{h^{\frac{3}{2}}}{2}A, & \text{for the schemes (13, 14).} \end{cases}$$

In order that the numerical simulation converges to a stationary process, the matrix in the drift term, $R(h, A)$, must satisfy

$$\sup_{\lambda \in \sigma(A)} |R(h, \lambda)| < 1, \tag{16}$$

where $\sigma(A)$ is the spectrum of A. In the positive definite A case using the explicit Euler method, this criteria is

$$h < \frac{2}{\sup_{\lambda \in \sigma(A)} \lambda}. \tag{17}$$

In the second order explicit weak algorithm (13), the analogous concern is to keep $\sup |1 - h\lambda + \frac{1}{2}h^2\lambda^2| < 1$. This conflicts with our desire to use a large stepsize to speed up convergence to stationarity. In the non-symmetric A case, the spectra may be complex so the analysis is more subtle than (17). However, this theory is well known in the numerical analysis of ordinary differential equations: (16) when using the Euler scheme implies that $h\sigma(A)$ must fall into Gershgorin circles.

Certainly a fully implicit method like

$$X_{k+1} = X_k - \frac{h}{2}(AX_k + AX_{k+1}) + \Delta W$$

is useless for our purposes because finding $(I + hA/2)^{-1}$ is at least as difficult as finding A^{-1} – our objective. However, imagine splitting A into two pieces

$$A = B + C,$$

where $1 + hB/2$ is easy to invert. In that situation, the following semi-implicit method becomes useful (see (11) and (13)),

$$\begin{aligned} X_{k+1} &= X_k - \frac{h}{2}(AX_k + BX_{k+1} + CX_{euler}) + \Delta W, \\ &= (I + \frac{h}{2}B)^{-1}\left(X_k - \frac{h}{2}(AX_k + CX_{euler}) + \Delta W\right). \end{aligned} \tag{18}$$

An obvious choice is $B = D = diag(A)$. There is a lot of freedom to choose such splittings: see, for example [9]. Substituting update (18) for X_{euler} in the last equation (for Z^{ij}) in (13), we get a semi-implicit 2nd order weak scheme. A careful choice of B may permit a relatively large stepsize h. If we pick $B = D = diag(A)$, the resulting **splitting scheme** algorithm uses (18) then a trapezoidal rule to get the Z^{ij} extended variables:

- compute X_{k+1} from (18) with $B = D$,
- and compute the extended system variables Z by

$$Z^{ij}_{k+1} = Z^{ij}_k + h(X^i_k X^j_k + X^i_{k+1} X^j_{k+1}). \tag{19}$$

3 NUMERICAL EXPERIMENTS

In this section we show results from numerical experiments. First we will study the behaviour of the explicit schemes (11, 13, 14) for some simple one dimensional problems. Continuing with the explicit schemes, our example will be the approximation of A^{-1} for a finite difference tridiagonal matrix. For this example, we compare the condition numbers of A and MA using our $M \approx A^{-1}$ estimate. We finish for these schemes with the results from solving a linear system $Ax = b$ iteratively using the (preconditioned) conjugate gradients method [4].

In a (shorter) second part, we show results from the splitting scheme (18,19) and compare it with the Euler scheme (11).

3.1 Testing the explicit schemes

To start with, consider the $(n \times n)$-matrix

$$A = T_4 = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}. \tag{20}$$

The eigenvalues of T_4 are

$$\lambda^i = 4 - 2\cos(i\pi/(n+1)) \tag{21}$$

with the largest and smallest behaving as

$$\max \lambda^i = \lambda^n \longrightarrow 6 \quad \text{and} \quad \min \lambda^i = \lambda^1 \longrightarrow 2$$

when $n \to \infty$. Its inverse is dense and easily found using MATLAB or MAPLE, so it provides a useful example for incomplete preconditioning. To test convergence of our schema, we first study one dimensional problems with $A = a = 2$ and $A = a = 6$.

3.1.1 One dimensional simulations

For this range of eigenvalues, we compare first order (11) and second order schemes (13), (14) for the system

$$dX = -aX\,dt + dW,\ X(0) = 0 \quad \text{and} \quad dZ = X^2 dt,\ Z(0) = 0 \tag{22}$$

for scalar $a = 2, 6$. For both values of a we will make $N = 1000$ steps for different stepsizes h. Stability condition (16) requires $h < 1/3$ when $a = 6$. Simulation results are shown for $h = 3/10, 3/20, 3/40$ giving the corresponding final times $T = 300, 150, 75$. To examine the

statistical convergence, we use $N_P = 30$ independent simulations. Denote by $\tilde{Z}_k^{(i)}$ the discrete time average up to time kh of the ith simulation, $\tilde{Z}_k^{(i)} = Z_k^{(i)}/(kh)$, and by $\overline{Z}_k$ its sample mean at step k:

$$\overline{Z}_k = \frac{1}{N_P} \sum_{i=1}^{N_P} \tilde{Z}_k^{(i)}. \tag{23}$$

As a monitor of convergence for $\overline{Z}_k$ we follow the relative error

$$e_k = 2a\overline{Z}_k - 1 \tag{24}$$

and the sample standard deviation

$$\sigma_k(Z) = \sqrt{\frac{D_k}{N_P - 1}} \quad \text{with} \quad D_k = \frac{1}{N_P} \sum_{i=1}^{N_P} \left(\overline{Z}_k - \tilde{Z}_k^{(i)} \right)^2. \tag{25}$$

Our results for $a = 2$ are shown in Figure 1. Those for $a = 6$ are in Figure 2. The better convergence of the higher order schemes (13), (14) compared to the Euler method (11) is evident. The Euler scheme appears also much more sensitive to the stepsize h. Second order methods convergence to a stationary state is faster at $a = 6$. This can be seen by comparing trajectories and relative errors but more convincingly by smaller standard deviations. For $h = 3/10$, which is near the unstable value $h = 1/3$, the Euler scheme (11) converges to a state, which is completely wrong. Instead of converging to the dark horizontal line at $1/(2a)$, for large stepsizes convergence is to a line above – easiest seen in the different scale of abscissa in Figure 2 for the Euler method compared to the scale for second order schema (13), (14) in the first and second columns. We see that the Euler scheme has large fluctuations (high standard deviation) as the stepsize h approaches the maximal $h < 2/a = 1/3$ value. For these 1-D tests, the higher order schemes still give reasonable results which are more than a factor of ten better than the low order Euler method near this critical h. Reducing the stepsize h improves the results for the Euler method and eventually it becomes comparable to the higher order schemes. Another interesting result is that the Taylor scheme (14) approaches the true solution from below for all chosen stepsizes h for $a = 2$: during the first 1000 steps, this scheme is *under*estimating the true value of $1/(2a)$. When $a = 6$, the Taylor scheme (14) still underestimates the value of $1/(2a)$ with small stepsizes $h = 3/20, 3/40$.

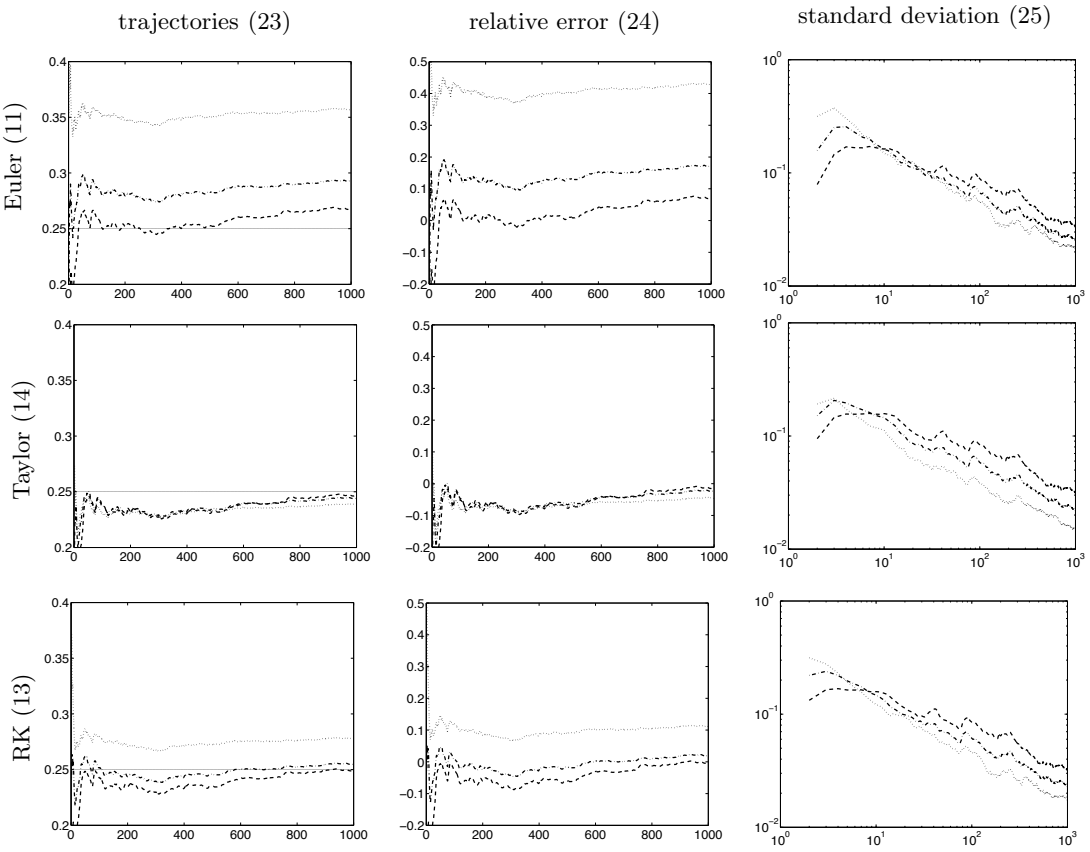

Figure 1: One dimensional simulations ($n = p = 1$) for system (22) with $A = a = 2$. From left to right: mean of trajectories, mean of relative error and standard deviation of $Z_k/(kh)$ for the different schemes (from top to bottom): Euler (11), weak order 2.0 Runge-Kutta (13), and weak order 2.0 Taylor (14). Simulation parameters were: $N = 1000$, $N_P = 30$ and $h = \frac{3}{10}(\cdots)$, $h = \frac{3}{20}(-\cdot-)$ and $h = \frac{3}{40}(---)$. In the left column the exact solution $1/(2a)$ is marked with a solid line (—).

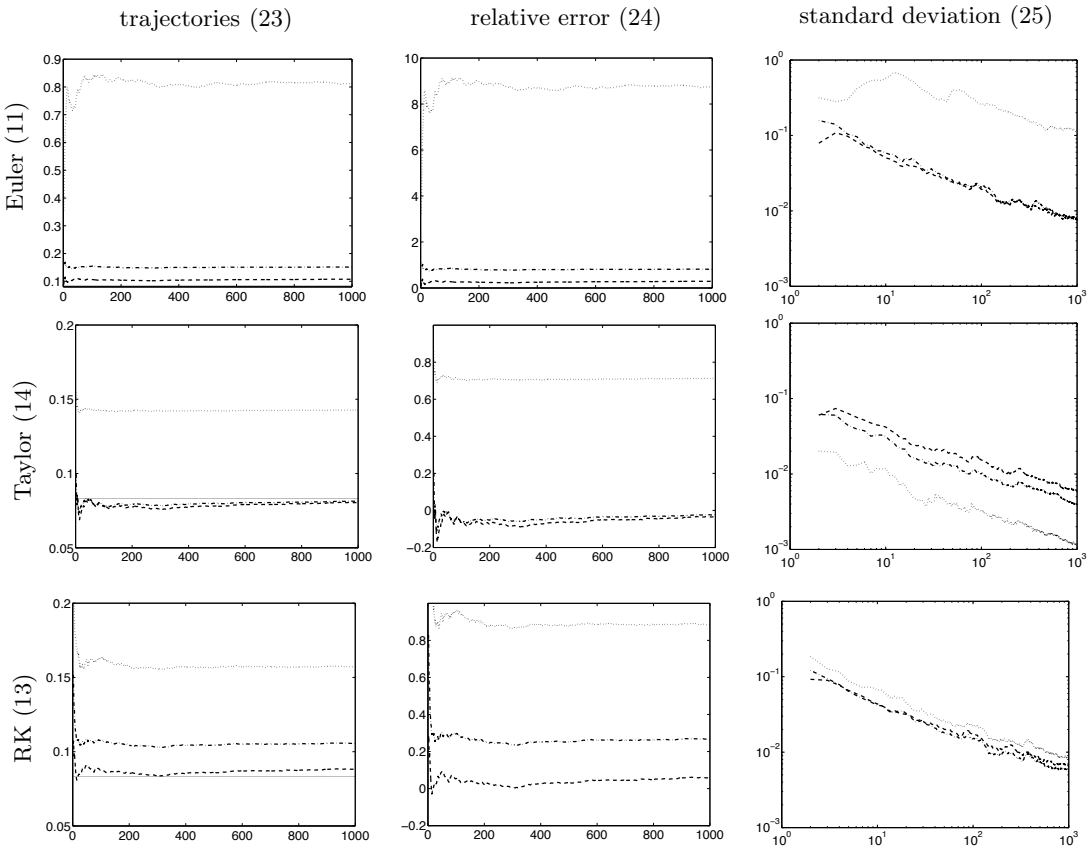

Figure 2: One dimensional simulations with same settings as in Figure 1 with higher $A = a = 6$. Note the different scaling in the first two plots of the first row compared with rows two and three due to the big error arising from the Euler scheme with $h = 0.3$.

3.1.2 Preconditioning T_4

We now turn to the task of preconditioning $A = T_4$ (see (20)) using our incomplete inverse estimates. Simulations in $n = 10, 100$ and 1000 dimensions were run. For these sizes, T_4 has the condition number in the two-norm rounded to four digits (using 21),

$$\kappa_2(T_4) = \frac{\lambda_{max}}{\lambda_{min}} = \frac{4 + 2\cos(\pi/(n+1))}{4 - 2\cos(\pi/(n+1))} \approx \begin{cases} 2.844, & n = 10 \\ 2.998, & n = 10^2 \\ 3.000, & n = 10^3. \end{cases}$$

Diagonal preconditioning We first examine the incomplete inverse consisting of only the diagonal of T_4^{-1}, i.e. $M^{ij} = 0$ when $i \neq j$; that is, $p = n$ from (10). We applied the schemes of section 2.1 using only the Z^{ii} updates of (13) and the $(i = j)$ case (14). Using an *exact* diagonal preconditioning ($M^{ii} = (T_4^{-1})^{ii}$) computed by MATLAB using `inv(.)`), we have

$$\kappa_2(MT_4) \approx \begin{cases} 2.847, & n = 10 \\ 2.998, & n = 10^2 \\ 3.000, & n = 10^3, \end{cases}$$

which shows that diagonal preconditioning does not improve the condition number. Our results are consistent with this analysis of estimates for M using weak approximations. The different simulations yield condition numbers listed in Table 1: We include a column with the improvement ratio $r_2 = \kappa_2(T_4)/\kappa_2(MT_4)$.

Table 1: Condition number of MT_4 with diagonal M and improvement ratio $\kappa_2(T_4)/\kappa_2(MT_4)$.

		$h = 1/6$, $T = 500/3$ $N = 1000$		$h = 1/12$, $T = 500/6$ $N = 1000$		$h = 1/12$, $T = 500/3$ $N = 2000$	
scheme	n	$\kappa_2(MT_4)$	r_2	$\kappa_2(MT_4)$	r_2	$\kappa_2(MT_4)$	r_2
Euler (11)	10	2.852	0.997	2.860	0.995	2.849	0.998
	100	3.017	0.994	3.030	0.989	3.018	0.994
	1000	3.028	0.991	3.061	0.980	3.043	0.986
Taylor (14)	10	2.853	0.997	2.860	0.994	2.850	0.998
	100	3.015	0.994	3.032	0.989	3.025	0.991
	1000	3.041	0.986	3.071	0.977	3.052	0.983
RK (13)	10	2.853	0.997	2.861	0.994	2.850	0.998
	100	3.018	0.993	3.032	0.989	3.020	0.993
	1000	3.031	0.990	3.066	0.978	3.047	0.984

For the ten dimensional problem we will have a closer look at the resulting eigenvalues of MT_4. The left plot in Figure 3 shows the eigenvalues of the diagonally preconditioned matrix T_4 for $n = 10$. It can be seen that the Euler scheme (11) is much more sensitive to the choice of the stepsize h than the higher order schemes. Further, we note that for $h = 1/12$, another 1000 numerical integration steps to $T = 500/3$ does not improve the distribution of the eigenvalues. This shows that the stochastic process has clearly converged to a stationary state. The Taylor scheme (14) gives very good concordance with exact diagonal preconditioning for all chosen parameter sets. Still, this scheme underestimates certain eigenvalues.

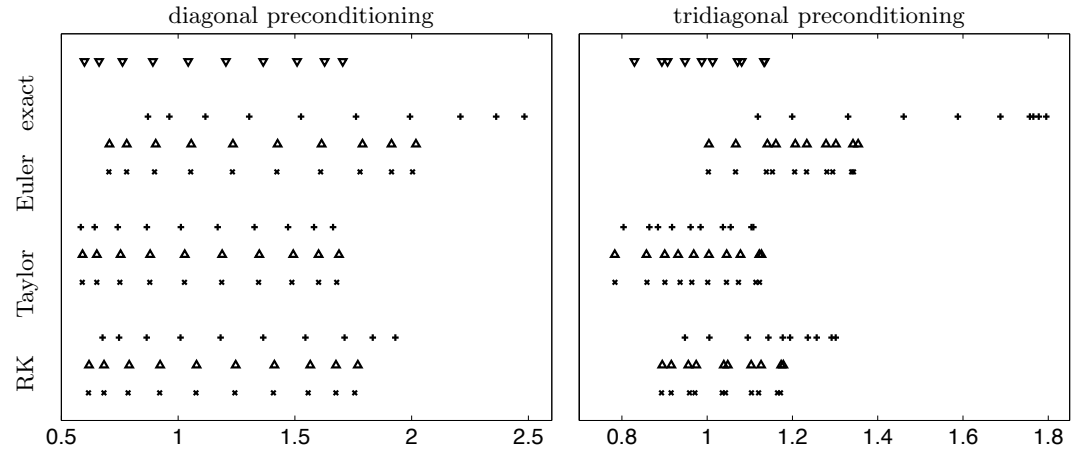

Figure 3: Eigenvalues of preconditioned (10×10) matrix MT_4 for (from top to bottom) exact preconditioning ($\bigtriangledown$) and for the three approximations Euler (11), RK (13), and Taylor (14) for the three parameter sets $h = 1/6$, $T = 500/3$ (+), $h = 1/12$, $T = 500/6$ ($\bigtriangleup$) and $h = 1/12$, $T = 500/3$ ($\times$). On the left (right), results from diagonal (tridiagonal) preconditioning are shown.

Tridiagonal preconditioning In this section, results from tridiagonal preconditioning of T_4 are shown, i.e. $M^{ij} = 0$ for $|i-j| > 1$. For exact (i.e. MATLAB) tridiagonal preconditioning we have

$$\kappa_2(MT_4) \approx \begin{cases} 1.369, & n = 10 \\ 1.437, & n = 10^2 \\ 1.438, & n = 10^3 \end{cases} \quad \text{and} \quad r = \frac{\kappa_2(T_4)}{\kappa_2(MT_4)} \approx \begin{cases} 2.078, & n = 10 \\ 2.086, & n = 10^2 \\ 2.086, & n = 10^3. \end{cases}$$

Again we computed approximations to M using weak approximations yielding the condition numbers and improvement ratios listed in Table 2. As for diagonal preconditioning, we show a plot of the eigenvalues for the ten dimensional problem, see the right plot in Figure 3. The points noted in the diagonally preconditioned example are even more obvious here: the Euler method (11) is very sensitive to the choice of the stepsize h, whereas the higher order schemes (13,14) give much better results also for higher stepsizes. Especially the Taylor scheme (14) gives very satisfactory results for all choices of the parameters shown, but underestimation of certain eigenvalues is still evident.

3.1.3 Preconditioned Conjugate Gradients with T_4

In this section we show results with the method of conjugate gradients (CG, [4]) to solve the linear system $Ax = b$ with symmetric positive definite A. We computed approximate solution to $T_4 x = b$ with $b = (1, \ldots, 1)^T$ using preconditioned CG. As seen in the first paragraph of Section 3.1.2 (Table 1), diagonal preconditioning yields no improvement. Therefore, only results for tridiagonal preconditioning in $n = 1000$ and $n = 2000$ dimensions are shown. Starting with the initial guess $x_0 = (0, \ldots, 0)^T$, we performed CG iterations until the two

Table 2: Condition number of MT_4 with tridiagonal M and improvement ratio $\kappa_2(T_4)/\kappa_2(MT_4)$: N is the number of timesteps.

		$h = 1/6$, $T = 500/3$ $N = 1000$		$h = 1/12$, $T = 500/6$ $N = 1000$		$h = 1/12$, $T = 500/3$ $N = 2000$	
scheme	n	$\kappa_2(MT_4)$	r_2	$\kappa_2(MT_4)$	r_2	$\kappa_2(MT_4)$	r_2
Euler (11)	10	1.604	1.772	1.349	2.108	1.340	2.123
	100	1.708	1.755	1.429	2.103	1.418	2.115
	1000	1.729	1.735	1.456	2.061	1.432	2.095
Taylor (14)	10	1.379	2.062	1.440	1.975	1.433	1.985
	100	1.450	2.068	1.544	1.942	1.518	1.942
	1000	1.476	2.032	1.563	1.919	1.539	1.950
RK (13)	10	1.373	2.071	1.318	2.158	1.313	2.167
	100	1.445	2.074	1.407	2.131	1.377	2.177
	1000	1.464	2.049	1.410	2.128	1.401	2.142

norm of the relative (w.r.t. the initial) residual

$$\frac{\|Ax_k - b\|_2}{\|b\|_2}$$

was smaller than 10^{-40}. The three algorithms (11,13, 14) were tested to compute the approximate inverse M which was subsequently used to precondition the system $Ax = b$. That is, $Ax = b \to MAx = Mb$ was solved using CG. Without preconditioning, the CG algorithm needed 68 iterations for both $n = 1000$ and $n = 2000$. Table 3 shows results for means over $N_P = 30$ samples. Looking at Table 3, several points can be observed:

1. The Euler method (columns marked with "E") is very sensitive to the chosen stepsize h. Recall that the critical h for T_4 is $\approx 1/3$. For the large $h = 0.3$, the preconditioned system needs many iterations more than the pure CG. For smaller h, however, the results also improve for first order Euler method.

2. Choosing the stepsize around half the critical stepsize (results for $h = 0.15$), the Euler method gives comparable results with the higher order algorithm and is hence favorable (as it is much faster to compute), especially if the number of steps N increases.

3. As the number of simulated paths (sample size N_P) is decreased, the algorithms becomes more sensitive to the final time $T = Nh$. Results for small number of steps become worse. This holds for all three schemes discussed: (11), (13), and (14).

4. For a moderate sample size of $N_P = 30$, good results can be obtained with as few as $N = 16$ steps with the higher order schemes using a stepsize $h = 0.15$ (around half the critical stepsize).

5. It is evident that the approximation of M becomes better, the longer system (10) is integrated numerically and the larger the sample is.

6. The second order schemes (13, 14) give comparable results when the parameters are properly chosen.

Table 3: Number of iterations needed with preconditioned CG to solve $T_4 x = b$ in two different dimensions n using a tridiagonal preconditioner M. The entries of M where computed using the Euler scheme (E,11), the Runge-Kutta scheme (RK,13), the Taylor scheme (T,14). Results are for different parameter sets h (stepsize), N (number of steps) and N_P (sample size, number of paths).

		$N_P = 15$						$N_P = 30$					
		$n = 1000$			$n = 2000$			$n = 1000$			$n = 2000$		
h	N	E	T	RK	E	T	RK	E	T	RK	E	T	RK
0.3	8	158	65	77	178	70	83	141	60	69	146	64	72
	16	157	61	69	160	61	67	145	57	62	139	57	64
	32	147	57	60	151	57	62	133	54	57	134	55	57
	62	136	53	56	134	54	56	130	52	54	125	52	52
	125	126	52	52	127	52	52	124	50	50	123	50	50
	250	124	50	50	124	50	51	121	49	49	122	49	49
	500	121	49	48	121	49	49	119	48	47	119	48	47
0.15	8	83	73	79	90	81	83	74	62	67	79	64	71
	16	73	62	68	71	70	67	63	53	58	66	57	61
	32	61	56	55	63	61	59	57	48	50	56	52	50
	62	55	50	49	55	54	50	52	46	46	50	48	46
	125	51	47	45	50	48	45	47	44	43	48	44	43
	250	48	43	42	47	45	42	45	42	40	45	42	41
	500	45	42	40	45	42	40	44	41	39	44	41	39
0.075	8	108	95	100	108	97	102	84	79	81	86	76	80
	16	79	80	77	83	95	91	67	63	65	72	70	71
	32	65	67	64	69	77	72	57	56	56	57	62	58
	62	55	59	55	57	66	61	48	50	48	51	55	52
	125	49	55	50	49	54	50	45	49	46	46	48	45
	250	44	47	43	45	49	45	41	45	41	42	45	42
	500	41	45	41	42	45	42	40	43	39	40	43	40

7. Concluding this test we note that, for T_4, the best performance can be probably be obtained *without* preconditioning. The matrix T_4 is almost an ideal example for the CG method – making preconditioning unnecessary. Nevertheless, the results shown may give some good indications about other linear systems.

3.2 Results with the splitting scheme

In order to examine improved stability when using large stepsizes, we used the $(n \times n)$-matrix

$$T_2 = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

T_2 has Gershgorin bounds of 0 and 4. For the Ornstein Uhlenbeck process (2) to converge to a stationary state, we need $\min \lambda^i > 0$. But as n increases, the smallest eigenvalue

Table 4: Comparison of splitting scheme (18,19) and Euler scheme (11) for $A = T_2 + 0.01I$. Shown are the maximal error ($e = \max_{i,j} |(A^{-1})^{ij} - M^{ij}|$), the condition number of the preconditioned matrices ($\kappa_{MA} = \kappa_2(MA)$ and $\kappa_{MT_2} = \kappa_2(MT_2)$) and the improvement factors ($r_A = \kappa_2(A)/\kappa_2(MA)$ and $r_{T_2} = \kappa_2(T_2)/\kappa_2(MT_2)$). Results are for a fixed sample size $N_P = 30$, various stepsizes h and number of steps N.

			Euler scheme					Splitting scheme				
n	h	N	e	κ_{MA}	r_A	κ_{MT_2}	r_{T_2}	e	κ_{MA}	r_A	κ_{MT_2}	r_{T_2}
10	0.2	4	2.16	30.5	1.42	34.1	1.42	2.13	18.2	2.38	20.3	2.38
		8	1.78	13.9	3.11	15.5	3.11	1.79	9.84	4.39	11.0	4.40
		16	1.42	6.26	6.90	6.98	6.93	1.44	5.33	8.10	5.90	8.20
100	0.2	16	4.27	624	0.59	6716	0.62	4.34	645	0.57	6962	0.59
		32	4.02	227	1.61	2406	1.72	4.00	242	1.51	2597	1.59
		64	3.72	120	3.04	1250	3.30	3.76	133	2.75	1395	2.96
		128	3.02	40.7	8.98	417	9.91	3.01	44.6	8.19	463	8.92
	0.4	16	4.14	382	0.96	4177	0.99	4.06	282	1.30	3011	1.37
		32	3.84	148	2.46	1613	2.56	3.65	100	3.65	1092	3.78
		64	3.42	84.0	4.35	871	4.74	3.30	64.8	5.64	673	6.14
		128	2.48	32.5	11.3	325	12.7	2.33	22.5	16.2	225	18.4
	0.6	16	not stable					3.89	226	1.61	2393	1.73
		32						3.38	77.1	4.74	819	5.05
		64						2.97	51.2	7.14	518	7.99
		128						1.91	18.4	19.9	174	23.7

($\lambda_1 = 2 - 2\cos(1/(n+1)) \sim (n+1)^{-2}$) quickly decreases toward zero. We therefore shift T_2 by ε, $A = T_2 + \varepsilon I$ with $\varepsilon = 0.01$. The extremal eigenvalues for A are then,

$$\lambda_{min} \approx \begin{cases} 0.0910, & n = 10 \\ 0.0110, & n = 100 \end{cases} \quad \text{and} \quad \lambda_{max} \approx \begin{cases} 3.9290, & n = 10 \\ 4.0090, & n = 100 \end{cases} .$$

The resulting condition numbers are $\kappa_2(A) \approx 43.1690$ ($n = 10$) and $\kappa_2(A) \approx 365.54$ ($n = 100$). We compare the splitting scheme with X-updates (18) with the Euler updates (11).

T_2 is no longer (strictly) diagonally dominant like T_4. In consequence, we compare *full* approximations of A^{-1}: that is, $p = n^2$ in (10) computing all the indices (i, j). In this situation, the approximate inverse (preconditioner) M is complete. The results in Table 4 show that in $n = 10$ dimensions (where $\lambda_{min} > 0.09$) the Euler scheme (11) still does well. Larger n decreases the smallest eigenvalue to around 0.01 and thus increases the condition number by a factor of nine. As n increases, the better performance of the splitting scheme (18,19) becomes evident. As it allows larger stepsizes, convergence to stationarity improves significantly.

4 NON-SYMMETRIC CASE

We now extend this procedure for non-symmetric but stable matrices. The modification of system (1) involves two processes instead of one. We require that $A > 0$, but it does not have to be normal ($[A, A^T] \neq 0$).

4.1 Extended system of SDEs

Extending the ideas of §1 to non-symmetric matrices A is not difficult. In this situation, we use highly correlated processes X and Y – a system of $2n$ SDEs:

$$\begin{aligned} dX &= -AXdt \ + dW, \qquad (26)\\ dY &= -A^T Y dt + dW, \qquad (27)\end{aligned}$$

again with deterministic initial conditions $X^i(0) = Y^i(0) = 0$. It is important that in (26,27) $W = \{W\}_{t\geq 0}$ is the *same* n dimensional Brownian motion for both processes $X = \{X\}_{t\geq 0}$ and $Y = \{Y\}_{t\geq 0}$. The solutions with these initial conditions are

$$X(t) = \int_0^t (e^{-(t-u)A})\, dW(u) \quad \text{and} \quad Y(t) = \int_0^t (e^{-(t-v)A^T})\, dW(v).$$

From these formal solutions, we can extract the X, Y covariance.

Theorem 4.1 *The covariance* $\mathbb{E}\, X(t)Y^T(t) \to \frac{1}{2}A^{-1}$ *as* $t \to \infty$.
Proof: *Using (3) and explicitly writing out the indices* $i, j = 1, \ldots, n$, *and using the summation convention over* k, l, *we have*

$$\begin{aligned} \mathbb{E}\,[X^i(t)Y^j(t)] &= \int_0^t \int_0^t (e^{-(t-u)A})^{ik} (e^{-(t-v)A^T})^{jl} \mathbb{E}\,[dW^k(u) dW^l(v)] \\ &= \int_0^t (e^{-(t-u)A})^{ik} (e^{-(t-u)A^T})^{jk} du \\ &= \int_0^t (e^{-2(t-u)A})^{ij} du. \end{aligned}$$

The last matrix valued integral is the same as in our Lemma 1.1 (5) of §1. Since $A > 0$, *as* $t \to \infty$ *we get* $\mathbb{E}\,[X^i(t)Y^j(t)] \to \frac{1}{2}(A^{-1})^{ij}$ *as stated.* □

Again, as in the symmetric case, long time averages (9) are needed to compute $M \approx A^{-1}$.

4.2 Numerical tests

For our non-symmetric cases we do relatively simple tests for convergence using small (10×10 and 4×4) matrices. For these cases, a *full* approximation (or *complete* preconditioner) $M \approx A^{-1}$ is computed. That is, in (10) the set $P = \{(i, j)\}$ has the full set of n^2 indices. Condition numbers $\kappa(A)$ and $\kappa(MA)$ are compared. For the full approximation we consider the $(2n + n^2)$ system of SDEs

$$\begin{aligned} dX &= -AXdt \ + dW \\ dY &= -A^T Y dt + dW \qquad (28)\\ dM &= 2XY^T dt \end{aligned}$$

where X and Y are n dimensional vector processes and M is a $(n \times n)$ dimensional matrix process. The initial conditions are $X^i(0) = Y^i(0) = M^{ij}(0) = 0$ for $i, j = 1, \ldots, n$.

From 4.1, our estimate for A^{-1} is

$$\lim_{T\to\infty} \frac{M(T)}{T} = A^{-1},$$

and we only consider the Euler method for system (28). This is a modification of (11):

$$\begin{aligned} X_{k+1} &= (I - hA)X_k + \sqrt{h}\xi_{k+1} \\ Y_{k+1} &= (I - hA^T)Y_k + \sqrt{h}\xi_{k+1} \\ Z^{ij}_{k+1} &= Z^{ij}_k + hX^i_k Y^j_k. \end{aligned} \tag{29}$$

It is important to note that the *same* random vector ξ_{k+1} is used for both the X- and the Y-updates in (29). As in our analysis of the symmetric case, we compute a small sample of realizations to both accelerate convergence to a stationary state and monitor this convergence as k increases.

4.2.1 The test matrices

Since A is real, its eigenvalues may be real or appear in complex conjugate pairs. Our simulations take account of the different situations that might arise. We tested an upper bidiagonal matrix A_1, and 4×4 block diagonal matrices $A_2, \ldots, A_4$,

$$A_1 = \begin{pmatrix} 1 & -1 & & & \\ & 2 & -1 & & \\ & & \ddots & \ddots & \\ & & & 9 & -1 \\ & & & & 10 \end{pmatrix}, \tag{30}$$

$$A_2 = \begin{pmatrix} B_1 & 0_2 \\ 0_2 & B_2 \end{pmatrix},\ A_3 = \begin{pmatrix} B_1 & 0_2 \\ 0_2 & B_3 \end{pmatrix},\ \text{and } A_4 = \begin{pmatrix} B_3 & 0_2 \\ 0_2 & B_4 \end{pmatrix}, \tag{31}$$

where the diagonal blocks are

$$B_1 = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix},\ B_2 = \begin{pmatrix} 2 & -1 \\ 0 & 2 \end{pmatrix},\ B_3 = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix},\ B_4 = \begin{pmatrix} 2 & -4 \\ 1 & 2 \end{pmatrix}.$$

Corresponding eigenvalues are 1 (for B_1), 2 (B_2), $1 \pm i$ (B_3) and $2(1 \pm i)$ (B_4). These imply the following maximal stepsizes h (see (16) and the reasoning thereafter): 2 (B_1), 1 (B_2 and B_3) and 0.5 (B_4). We also performed experiments with $\tilde{A}_i = U^T A_i U$ with a random orthogonal matrix U, chosen such that the block diagonal structure of A_i was lost and any symmetries were broken for $i = 2, 3, 4$. The results did not differ significantly from the results shown for simulations with the block diagonal A_i's (see Table 6).

4.2.2 Results

We will show the maximal error ($e = \max_{i,j} |(A^{-1})^{ij} - M^{ij}|$), the condition number of the preconditioned matrix ($\kappa_2(MA)$) and the improvement factor ($r = \kappa_2(A)/\kappa_2(MA)$) for the matrices $A = A_1$ (30) and A_i $i = 2, \ldots, 4$ (31).

We start with the matrix A_1, with $\kappa_2(A_1) \approx 11.926$. The critical stepsize is $h < 0.2$, so the simulations were run with $h = 0.1$. The inverse of A_1 is an upper triangular matrix ($M^{ij} = 0$ for $i > j$). In Table 5 we show results where we took this fact into account *a priori*, and additionally results for the *full* approximation M.

Table 5: Maximal error e, condition number $\kappa_2(MA_1)$ and improvement ratio r for the matrix A_1. Results are for a fixed stepsize $h = 0.1$ and different number of steps N and sample sizes N_P. In the upper part, only the elements M^{ij} with $j \geq i$ where computed (with the others set to zero), whereas the lower part shows results from computing the full M.

		$N_P = 15$			$N_P = 30$		
	N	e	$\kappa_2(MA_1)$	r	e	$\kappa_2(MA_1)$	r
upper triangular M	8	0.695	6.550	1.821	0.575	4.600	2.593
	16	0.464	4.328	2.755	0.250	2.707	4.405
	32	0.166	2.804	4.253	0.166	1.938	6.153
	64	0.118	1.898	6.284	0.220	1.816	6.567
full M	8	0.695	9.908	1.204	0.575	5.214	2.287
	16	0.464	5.159	2.312	0.250	3.089	3.861
	32	0.166	3.337	3.574	0.166	2.177	5.479
	64	0.118	2.217	5.379	0.220	2.124	5.615

In Table 6 we show results for the matrices A_2, A_3 and A_4. They have condition numbers $\kappa_2(A_2) \approx 4.145$, $\kappa_2(A_3) \approx 2.618$ and $\kappa_2(A_4) \approx 3.325$. We show results for sample sizes $N_p = 5$ and $N_p = 25$ and a fixed stepsize $h = 0.1$.

Table 6: Maximal error e, condition number $\kappa_2(MA_i)$ and improvement ratio r for the matrices A_i, $i = 2, 3, 4$. Results are for a fixed stepsize $h = 0.1$ and different number of steps N and sample sizes N_P.

		$N_P = 5$			$N_P = 25$		
	N	e	$\kappa_2(MA_i)$	r	e	$\kappa_2(MA_i)$	r
A_2	4	1.053	8.085	0.513	0.937	2.661	1.557
	8	0.865	8.020	0.517	0.695	1.933	2.144
	16	0.481	3.924	1.056	0.486	1.861	2.227
	32	0.500	2.263	1.832	0.276	1.426	2.907
A_3	4	1.053	7.151	0.366	0.937	2.292	1.143
	8	0.865	9.503	0.276	0.695	1.563	1.676
	16	0.481	4.400	0.595	0.486	2.087	1.255
	32	0.500	3.176	0.824	0.276	1.440	1.818
A_4	4	0.553	9.209	0.361	0.513	3.360	0.989
	8	0.554	20.159	0.165	0.519	2.681	1.240
	16	0.356	3.844	0.865	0.361	2.567	1.295
	32	0.331	2.651	1.254	0.220	2.055	1.618

5 FINAL REMARKS

From the experimental results above, some of our important conclusions are as follows.

- From our tables and figures, particularly Figure 3, it is clear that our basic procedure works reasonably well. In $N = O(n)$ steps, the orbits of an OUP seem to sample the distribution well enough to get a decent approximate covariance. It is unreasonable, however, to expect that $N \ll n$ will give a good result. These are Monte-Carlo simulations with an accuracy of roughly $O((N_P N)^{-1/2})$!
- It is important that the basic numerical operations for the integration are **very simple**: matrix-vector multiply (likely to be $O(n)$ operations for sparse systems), some Gaussian random numbers, and vector to vector additions. Also note that these operations are easily parallelizable: e.g., there are no inner products.
- When large stepsizes are used, 2nd order methods are helpful to get a good result. Decreasing the stepsize well below the critical value when using an Euler method,
$$h < \frac{2}{\max \lambda},$$
seems to work quite well, however. Since the Euler method is so inexpensive, using a smaller stepsize and more steps becomes attractive. Talay [11] comes to more or less the same conclusion in his analysis of the invariant measure.
- Stability of the SDE discretization seems more important than order for these long time integrations. Many choices of splitting methods are possible (18): we chose only the diagonal $B = D = diag(A)$ here. A tridiagonal or banded splitting might be useful in some situations. There is a lot of freedom to modify our procedures.
- We were very concerned about the rank of the preconditioner. Since the condition number $\kappa(MA)$ always moved toward unity from above for stable stepsizes and increasing the number of steps (N), our worry about rank deficiency was unfounded. For our diagonal, tridiagonal, and full preconditioners, **the preconditioner always seems to be of full rank.**

At this juncture, we take the opportunity to speculate. Although it is certainly true our test examples were relatively small and uncomplicated, the results seem to us quite encouraging. Because the basic integration procedure is so cheap, taking $O(n)$ steps is not prohibitive. Several methods to monitor the convergence to stationarity seem evident. Our approach here was to use multiple realizations ($N_P > 1$, see §1). Another possibility is to divide the number of steps taken (N) into "epochs" and monitor the convergence by comparing the Z-variance in two or three such epochs. This has the advantage of saving memory: only $N_P = 1$ copy of the $n + p$ variables is required.

Finally, by choosing a relative small stepsize, Euler methods seem appealing as long as stability can be maintained. In fact, in future work a **split Euler** method should be studied. Such a procedure would again split $A = B + C$, as in §2.2, where $I + hB$ is easily inverted but B chosen to maintain stability in the X update:

$$\begin{aligned} X_{k+1} &= X_k - hBX_{k+1} - hCX_k + \Delta W \\ &= (I + hB)^{-1} (I - hCX_k + \Delta W) . \end{aligned} \tag{32}$$

This is of the same low order (weak order 1) as the explicit Euler method, but by a clever choice of B in (32) could be made more stable than the explicit rule (11) and would permit larger stepsizes – thus reducing the number of time steps needed to sample the stationary stochastic orbits.

ACKNOWLEDGEMENTS

The authors would like to thank G. Golub, M. Gutknecht, and D. Talay for helpful discussions and encouragement.

REFERENCES

[1] Benzi M, Golub GH. Bounds for the entries of matrix functions with applications to preconditioning. *BIT* 1999; **39**(3): 417-438.

[2] Carmona RA. Transport properties of Gaussian velocity fields. In *Real and Stochastic Analysis: Recent Advances*, M.M. Rao (editor). CRC Press: Boca Raton, Florida, 1997; 9–63

[3] Dagpunar J. *Principles of Random Variate Generation*. Oxford Science Publications, Clarendon Press: Oxford, 1988.

[4] Hestenes M, Stiefel E. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards* 1952; **49**: 409–436.

[5] Horn RA, Johnson CR. *Topics in Matrix Analysis*. Cambridge Univ. Press, 1991.

[6] Kloeden PE, Platen E. *Numerical Solution of Stochastic Differential Equations* (1st edition). Springer–Verlag: Berlin, 1992.

[7] Knuth DE. *The Art of Computer Programming, volume 2: Seminumerical Algorithms* (3rd edition). Addison-Wesley publ.: Reading, Mass., 1998.

[8] Milstein GN. *Numerical Integration of Stochastic Differential Equations* (1st edition, revised and updated translation of Russian work). Kluwer Academic Publishers: Dordrecht, 1995.

[9] Petersen WP. A general implicit splitting for stabilizing numerical simulations of Itô stochastic differential equations. *SIAM Journal on Numerical Analysis* 1998; **35**(4): 1439-1451

[10] Revuz D, Yor M. *Continuous Martingales and Brownian Motion*. Springer-Verlag: Berlin, 1998.

[11] Talay D. Second-order discretization schemes for stochastic differential systems for the computation of the invariant law. *Stochastics and Stochastic Reports* 1990; **29**:13-36.

Research Reports

No.	Authors	Title
02-24	F.M. Buchmann, W.P. Petersen	A stochastically generated preconditioner for stable matrices
02-23	A.W. Rüegg, A. Schneebeli, R. Lauper	Generalized *hp*-FEM for Lattice Structures
02-22	L. Filippini, A. Toselli	*hp* Finite Element Approximations on Non-Matching Grids for the Stokes Problem
02-21	D. Schötzau, C. Schwab, A. Toselli	Mixed *hp*-DGFEM for incompressible flows II: Geometric edge meshes
02-20	A. Toselli, X. Vasseur	A numerical study on Neumann-Neumann and FETI methods for *hp*-approximations on geometrically refined boundary layer meshes in two dimensions
02-19	D. Schötzau, Th.P. Wihler	Exponential convergence of mixed *hp*-DGFEM for Stokes flow in polygons
02-18	P.-A. Nitsche	Sparse approximation of singularity functions
02-17	S.H. Christiansen	Uniformly stable preconditioned mixed boundary element method for low-frequency electromagnetic scattering
02-16	S.H. Christiansen	Mixed boundary element method for eddy current problems
02-15	A. Toselli, X. Vasseur	Neumann-Neumann and FETI preconditioners for *hp*-approximations on geometrically refined boundary layer meshes in two dimensions
02-14	Th.P. Wihler	Locking-Free DGFEM for Elasticity Problems in Polygons
02-13	S. Beuchler, R. Schneider, C. Schwab	Multiresolution weighted norm equivalences and applications
02-12	M. Kruzik, A. Prohl	Macroscopic modeling of magnetic hysteresis
02-11	A.-M. Matache, C. Schwab, T. von Petersdorff	Fast deterministic pricing of options on Lévy driven assets
02-10	D. Schötzau, C. Schwab, A. Toselli	Mixed *hp*-DGFEM for incompressible flows
02-09	Ph. Frauenfelder, Ch. Lage	Concepts - An object-oriented software package for partial differential equations
02-08	A.-M. Matache, J.M. Melenk	Two-Scale Regularity for Homogenization Problems with Non-Smooth Fine Scale Geometry
02-07	G. Schmidlin, C. Lage, C. Schwab	Rapid solution of first kind boundary integral equations in $\mathbb{R}^3$