High Performance Computing in Stochastic PDE

Chenyue Wu  Yifei Sun

May 22, 2015

1 Background

In [1], numerical methods for evolutionary equation

\[ du = [-Au + f(u)]dt + G(u)dW(t), u(0) = u_0 \]

are introduced, where \( D \) is a bounded domain, \( A \) is a linear differential operator, \( W(x,t) \) is a Q-Wiener process or Cylindrical Wiener process (also called space-time white noise). Here we suppress the dependence on space (and time) and write \( u(t) \) for \( u(t,x) \), \( f(u) \) for \( f(u,t,x) \), \( G(u) \) for \( G(u,t,x) \) and \( W(t) \) for \( W(t,x) \).

As discussed in [1], a general setting for \( u, A, f, G \) is following:

\( u_0 \in H \), where \( H \) is a Hilbert space (e.g. \( L^2(D) \)). \( W \) is a U-valued Q-Wiener process or Cylindrical Wiener process, where \( U \) is another Hilbert space and \( Q \) is its covariance operator. \( u \) is viewed as a \( H \)-valued stochastic process. \( A : D(A) \subset H \rightarrow H \) satisfies that it has a complete set of orthonormal eigenfunction with all positive eigenvalues.

\( f : H \rightarrow H \). \( G : H \rightarrow L^2_0 \), where \( L^2_0 \) is set of linear operators \( B : U_0 \rightarrow H \) such that:

\[ \|B\|_{L^2_0} := \|BQ^{1/2}\|_{HS(U,H)} < \infty \) (here \( U_0 = \{Q^{1/2}u : u \in U\} \)).

This document discusses about parallel algorithm for the same kind of spdes.

2 Numerical Scheme

2.1 Semi-implicit Euler-Maruyama and Galerkin finite element approximation

[1] talked about Semi-implicit Euler-Maruyama and Galerkin finite element approximation:

Consider a finite dimensional subspace \( V_h \) of \( H \), say, the finite element space, and seek the finite element approximation of the solution \( u_h(t) \in V_h \) to the SPDE:

\[ < du_h, v > = < [-Au + f(u)]dt, v > + < G(u)dW(t), v > \]

\[ = < -Au, v > dt + < f(u)], v > dt + < G(u)dW(t), v > \forall v \in V_h \]
The term \( < -Au, v > \) is rewritten (usually by integral by part) to a bi-linear form, which satisfies: 1. Continuity, 2. Garding’s inequality (To have well-posedness). And we denote by \( A_h \) the matrix for the bilinear form on space \( V_h \). The stochastic field \( W(t) \) has the expansion

\[
W(t) = \sum_{j=1}^{\infty} \sqrt{q_j} \xi_j \beta_j(t),
\]

where \( q_j \) are eigenvalues of covariance operator \( Q \), \( \xi_j \) are eigenfunctions of covariance operator \( Q \). \( \beta_j(t) \) are independent Brownian motions. Notice that for space-time white noise, \( q_j = 1 \) for all \( j \).

We make a truncation for \( W(t) \), only take the first \( J \) terms:

\[
W_J(t) = \sum_{j=1}^{J} \sqrt{q_j} \xi_j \beta_j(t),
\]

and use semi-implicit Euler-Maruyama in time discretization, i.e., take implicit in \( -Au \) part, but explicit in \( f(u) \) and \( G(u) \) part:

\[
< u_{h,n+1}, v > - < u_{h,n}, v > = - < Au_{h,n+1}, v > \Delta t + < f(u_{h,n}), v > \Delta t + < G(u_{h,n}) \Delta W_{J,n}, v > .
\]

After these preparation, we now have the vector form for this numerical scheme:

\[
(I + \Delta t A_h)u_{h,n+1} = u_{h,n} + f(u_{h,n}) + < NOISE TERM >, \tag{1}
\]

where assuming \( V_h \) has basis \( \phi_i, i = 1, 2, \cdots, m \), the i-th entry of the noise term is given by matrix \((< G(u_{h,n}) \sqrt{q_j} \xi_j, \phi_i >)(\Delta \beta_j)\) times the increment of Brownian motion \((\Delta \beta_j)\) during \( \Delta t \).

### 2.2 Parallelization

We observe that two parts of this scheme can be parallelized:

1. Parallely assemble the noise term \((< G(u_{h,n}) \sqrt{q_j} \xi_j, \phi_i >)(\Delta \beta_j)\)
2. Use parallel algorithm to solve linear system (1).

### 3 Examples

#### 3.1 2D: stochastic Allen-Cahn equation

We use the same example in [1]. On domain \([0, 4] \times [0, 4]\), consider stochastic Allen-Cahn equation

\[
du = [\epsilon u \Delta u + u - u^3]dt + \sigma dW(t)
\]

, with periodic boundary condition \( u(x,0) = u(x,4), u(0,y) = u(4,y) \).
In the following examples, we take the parameters to be: $\epsilon = 10^{-3}$, $\sigma = 0.1$. We take the triangular finite element spaces with 800 elements and want to obtain the solution at $T = 10$. The initial value is $u_0 = \sin(\pi x)\cos(\pi y)$ and the noise

$$W(t) = \sum_{j_1,j_2} \sqrt{q_{j_1,j_2}^{(1)}} \cos\left(\frac{j_1}{2}\pi x\right)\cos\left(\frac{j_2}{2}\pi y\right)\beta_{j_1,j_2}^{(1)}(t) + \sqrt{q_{j_1,j_2}^{(2)}} \cos\left(\frac{j_1}{2}\pi x\right)\sin\left(\frac{j_2}{2}\pi y\right)\beta_{j_1,j_2}^{(2)}(t)$$

$$+ \sqrt{q_{j_1,j_2}^{(3)}} \sin\left(\frac{j_1}{2}\pi x\right)\cos\left(\frac{j_2}{2}\pi y\right)\beta_{j_1,j_2}^{(3)}(t) + \sqrt{q_{j_1,j_2}^{(4)}} \sin\left(\frac{j_1}{2}\pi x\right)\sin\left(\frac{j_2}{2}\pi y\right)\beta_{j_1,j_2}^{(4)}(t),$$

where $q_{j_1,j_2}^{(1)} = q_{j_1,j_2}^{(2)} = q_{j_1,j_2}^{(3)} = q_{j_1,j_2}^{(4)} = e^{-\alpha(j_1^2+j_2^2)/16}$.

Here we use MPI to do the summation of noise term parallely, and use MPI reduce to get the final sum. The linear system here is not too large ($800 \times 800$), so we do not use parallel algorithm on solving linear equation. We use conjugate gradient to solve the linear equation.

<table>
<thead>
<tr>
<th>Number of Processor</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assembling Noise Term (s)</td>
<td>30.45</td>
<td>19.32</td>
<td>15.61</td>
</tr>
<tr>
<td>Solving Linear Equation (s)</td>
<td>11.26</td>
<td>13.96</td>
<td>21.13</td>
</tr>
</tbody>
</table>

References