SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. xx, pp. x

A hybrid sparse-grid approach for nonlinear filtering problems based on adaptive-domain of the Zakai equation approximations

Feng Bao *, Yanzhao Cao †, Clayton Webster ‡, and Guannan Zhang §

Abstract. A hybrid finite difference algorithm for the Zakai equation is constructed to solve nonlinear filtering problems. The algorithm combines the splitting-up finite difference scheme and hierarchical sparse grid method to solve moderately high dimensional nonlinear filtering problems. When applying hierarchical sparse grid method to approximate bell-shaped solutions in most applications of nonlinear filtering problem, we introduce a logarithmic approximation to reduce the approximation errors. Some space adaptive methods are also introduced to make the algorithm more efficient. Numerical experiments are carried out to demonstrate the performance and efficiency of our algorithm.

Key words.

1. Introduction.

In this effort, we consider the following stochastic differential system that combines an equation for the "state" and for the "observation" defined on the probability space $(\Omega, \mathcal{F}, P)$

$$\begin{cases} dX_t = b(X_t)dt + \sigma(X_t)dW_t, & (state) \\ dY_t = h(X_t)dt + dB_t. & (observation) \end{cases} \tag{1.1}$$

Here $\{X_t \in \mathbb{R}^d, t \geq 0\}$ and $\{Y_t \in \mathbb{R}^r, t \geq 0\}$ are two stochastic processes, $\{W_t, t \geq 0\}$ and $\{B_t, t \geq 0\}$ are independent Brownian Motions in $\mathbb{R}^p$ and $\mathbb{R}^r$, with covariance matrices C_W (identity) and C_B, respectively, and the given initial value X_0 is independent of W_t and B_t with probability distribution $u_0(x)dx$. The nonlinear filtering problem is to obtain the best estimate of X_t as the conditional expectation with respect to the observed paths $\{Y_s, 0 \leq s \leq t\}$ ([27, 34]). This can be expressed as finding stochastic process $\tilde{X}_t$ such that

$$\tilde{X}_t = E(X_t \mid \mathcal{Y}_t) = \inf\{E(|X_t - Y|^2);\ Y \in \mathcal{K})\} \tag{1.2}$$

where $\mathcal{Y}_t$ is the σ-algebra generated by the observation process up to t, and $\mathcal{K}$ is the space of all $\mathcal{Y}_t$-measurable and square integrable random variables.

Tremendous efforts have been made to solve linear and nonlinear filtering problems in the last few decades. The first major breakthrough is the landmark work on linear filtering [22], resulting in the Kalman filter (see also [11, 25, 26, 30]). For nonlinear filter problems, three of the most widely used methods are the extended Kalman filter (EKF) [1, 10, 19, 20, 21], the particle filter method (PFM) [3, 6, 8, 13, 28, 32] and the Zakai filter [7, 12, 18, 31, 34, 36]. For

*Department of Mathematics and Statistics, Auburn University, Auburn, Alabama, 36849 (fzb0005@auburn.edu).
†Department of Mathematics and Statistics, Auburn University, Auburn, Alabama, 36849 (yzc0009@auburn.edu).
‡Computer Science and Mathematics Division, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS-6164, Oak Ridge, TN 37831-6164 (webstercg@ornl.gov).
§Computer Science and Mathematics Division, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS-6164, Oak Ridge, TN 37831-6164 (zhangg@ornl.gov).

the EKF the state equation and the observation equation are linearized so that the standard Kalman filter can be applied. The main drawback of the EKF is its inaccuracy when the system is highly nonlinear. The central theme behind the PFM involves the representation of the desired probability density function (PDF) of the system state with a set of adaptively selected random samples. Since PFMs are essentially sequential Monte Carlo methods, with sufficiently large number of samples, PFMs can provide an accurate representation of the PDF for the nonlinear filtering solution.

On the other hand, the Zakai filter represents the PDF of the nonlinear filtering solution $\tilde{X}_t$ through the solution of a parabolic-type stochastic partial differential equation, known as the Zakai equation [34]. Similar to the PFM, the Zakai filter allows one to accurately compute conditional distributions. A number of numerical algorithms have been proposed to solve Zakai equations [2, 4, 9, 14, 15, 16, 17]. One of the most effective methods is the splitting-up approximation scheme [2, 23, 36] where the original Zakai equation is split into a second order deterministic PDE, related with the prediction step, and a degenerated second order stochastic PDE in the update step. In the numerical simulation process, a prior PDF is obtained by solving the deterministic PDE at the prediction; then this prior PDF is updated following a posteriori criterion.

The main drawback of the Zakai filter is that the numerical approximation is grid based, thus it suffers the so-called "curse of dimensionality" since the computing cost increases exponentially as the dimension of the system increases. Another difficulty related to solutions of the Zakai equation is that the domain is the whole space $\mathbb{R}^d$. To address these challenges, in this paper, we propose the construction of an efficient hybrid numerical algorithm which combines the advantages of the splitting-up approximation scheme for the Zakai equation, a hierarchical sparse grid method for moderately high dimensional nonlinear filtering problems [24, 33, 35] to compute the numerical solution of the Zakai filter, and an importance sampling method to adaptively construct a bounded domain at each time step of the temporal discretization. Specifically, this enables us to use the splitting-up finite difference scheme to solve the Zakai equation on the sparse grid of the bounded domain. The hierarchical sparse grid method, which was originally created to approximate multi-variable functions, uses only $O(n(\log n)^{d-1})$ number of grid points instead of $O(n^d)$ number of grid points, required by the standard full-grid approximation. Though sparse grid methods are not new, to the best of our knowledge this is the first time such a higher dimension approximation technique is used to solve the nonlinear filtering problem through the adaptive-domain solution of the Zakai equation. Our overall goal of this hybrid technique is to reduce the computational complexity of the Zakai filter while maintaining its advantages over other methods for nonlinear filtering problems.

The paper is organized as follows. In the next section, we briefly introduce the Zakai equation and its relation to the nonlinear filtering problem. In Section 3, we describe the sparse grid interpolation scheme. Then in Section 4 we construct our hybrid numerical algorithm for the nonlinear filtering problem. Finally in Section 5, we conduct numerical experiments to illustrate the effectiveness and efficiency of our new algorithm.

2. Nonlinear Filtering Problems and Zakai Equations.

In this section, we outline the derivation of the Zakai equation and its relationship to the nonlinear filtering problem (1.1).

Throughout this paper, we assume that the coefficients $b : \mathbb{R}^d \to \mathbb{R}^d$, $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times p}$ and $h : \mathbb{R}^d \to \mathbb{R}^r$ in (1.1) are globally Lipschitz continuous functions. Denote

$$\rho(t) \doteq \exp\left\{\int_0^t h^*(X_s)dY_s - \frac{1}{2}\int_0^t |h(X_s)|^2 ds\right\},$$

then the measure $\tilde{P}$ defined by $\tilde{P} = \rho(t)dP$ is also a probability measure on $(\Omega, \mathcal{F})$ equivalent to P . Furthermore, in the probability space $(\Omega, \mathcal{F}, \tilde{P})$, Y_t is a Browanian motion independent of X_t (for details, see [34]).

Assuming that $u = u(t, x)$ is the conditional density function of the state X_t given an observed path Y_t, then the optimal filtering solution is given by (see [34, 36])

$$E(\Phi(X_t) \mid \mathcal{Y}_t) = \frac{\int \Phi(x)u(t,x)dx}{\int u(t,x)dx}. \tag{2.1}$$

Under regularity assumptions for the coefficients b and h given above, u satisfies the following stochastic partial differential equation, known as Zakai equation

$$du(t,x) = L^*u(t,x)dt + h^*(x)u(t,x)dY_t, \quad x \in R^d, \tag{2.2}$$

with the initial value $u(0, x)$, and L the infinitesimal generator associated with the state process X_t such that

$$Lu = \frac{1}{2}\sum_{i,j}^{d}(\sigma\sigma^*)_{i,j}\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{d} b_i \frac{\partial u}{\partial x_i}, \tag{2.3}$$

and "$*$" is the transpose operator which transforms Lu to be

$$L^*u = \frac{1}{2}\sum_{i,j}^{d}\frac{\partial^2(\sigma\sigma^*)_{i,j}u}{\partial x_i \partial x_j} - \sum_{i=1}^{d}\frac{\partial b_i u}{\partial x_i}. \tag{2.4}$$

The intent of the Zakai filter method is to obtain numerical solutions of the Zakai equation (2.2). However, there are several challenges involved in the construction of an efficient numerical algorithm for the Zakai equation include (i) high-dimensionality of the state equations; (ii) low regularity of the solution; and (iii) unbounded solution domain. In the next two sections we intend to construct a hybrid algorithm combining the ideas of split-up finite difference method, sparse grid interpolation, and the importance sample approach to overcome these obstacles.

3. Hierarchical Local Sparse Grid Interpolation.

In this section, we introduce sparse grid interpolation constructed from a local hierarchical basis, which will be used in the finite difference approximation of the Zakai equation in the spatial domain.

3.1. Standard Hierarchical Sparse Grid Interpolation. Assume that we have the following one dimensional interpolation formula at our disposal:

$$\mathcal{Q}^i(u) = \sum_{j=1}^{m_i} u(x_j^i) \cdot \phi_j^i(x), \quad x \in R, \tag{3.1}$$

where $i \in \mathbb{N}$ is the *resolution level* of the interpolant $\mathcal{Q}^i$, m_i is the number of grid points on level i, x_j^i and $\phi_j^i(x)$ for $j = 1, \ldots, m_i$ are the interpolation points and the corresponding basis functions, respectively. In the context of linear hierarchical interpolation, m_i, x_j^i and ϕ_j^i in the standard interval $[-1, 1]$ for $i \in \mathbb{N}, j = 1, \ldots, m_i$ are defined by

$$m_i = \begin{cases} 1, & \text{if } i = 1, \\ 2^{i-1} + 1, & \text{if } i > 1, \end{cases} \tag{3.2}$$

$$x_j^i = \begin{cases} 0, & \text{for } j = 1, \quad \text{if } m_i = 1, \\ \dfrac{2(j-1)}{m_i - 1} - 1, & \text{for } j = 1, \ldots, m_i, \quad \text{if } m_i > 1, \end{cases} \tag{3.3}$$

and for $i = 1$, $\phi_1^1 = 1$; for $i > 1$ and $j = 1, \ldots, m_i$,

$$\phi_j^i = \begin{cases} 1 - \dfrac{m_i - 1}{2} \cdot |x - x_j^i|, & \text{if } |x - x_j^i| < \dfrac{2}{m_i - 1}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Note that the nodal basis function ϕ_j^i has local support $[x_j^i - 2^{1-i}, x_j^i + 2^{1-i}]$.

In the multi-dimensional case, i.e. $d > 1$, the tensor-product interpolatant is

$$\left(\mathcal{Q}^{i_1} \otimes \cdots \otimes \mathcal{Q}^{i_d}\right)(u) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_d=1}^{m_{i_d}} u\left(x_{j_1}^{i_1}, \cdots, x_{j_d}^{i_d}\right) \cdot \phi_{\mathbf{j}}^{\mathbf{i}}(\boldsymbol{x}), \tag{3.5}$$

where $\phi_{\mathbf{j}}^{\mathbf{i}} = \prod_{k=1}^{d} \phi_{j_k}^{i_k}$. Clearly, the above product requires $\Pi_{i=1}^{d} m_i$ function values, which is computationally prohibitive when d is large. The sparse gird interpolation [5] is a linear combination of a series of tensor-product interpolants, each of which is defined on a coarse grid with different resolutions in different dimensions, i.e.,

$$\mathcal{I}^{L,d}(u) = \sum_{L-d+1 \leqslant |\mathbf{i}| \leqslant L} (-1)^{L-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} \left(\mathcal{Q}^{i_1} \otimes \cdots \otimes \mathcal{Q}^{i_d}\right)(u), \tag{3.6}$$

where $L \geq d$, the multi-index $\mathbf{i} = (i_1, \ldots, i_d)$ and $|\mathbf{i}| = i_1 + \cdots + i_d$. Here, $i_k (k = 1, \ldots, d)$ is the level of the tensor-product interpolant $\mathcal{Q}^{i_1} \otimes \cdots \otimes \mathcal{Q}^{i_d}$ along the kth direction. The Smolyak algorithm builds the interpolant by adding a combination of all tensor-product interpolants satisfying $L - d + 1 \leq |\mathbf{i}| \leq L$. The structure of the algorithm becomes clearer when one considers the incremental interpolant, Δ^i given in [5]

$$\mathcal{Q}^0(u) = 0, \quad \Delta^i = \mathcal{Q}^i(u) - \mathcal{Q}^{i-1}(u). \tag{3.7}$$

The sparse grid interpolant (3.6) is then equivalent to

$$\mathcal{I}^{L,d}(u) = \sum_{|\mathbf{i}|\leqslant L} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}) = \mathcal{I}^{L-1,d}(u) + \sum_{|\mathbf{i}|=L} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d})(u). \tag{3.8}$$

The corresponding sparse grid associated with $\mathcal{I}^{L,d}(u)$ is represented by

$$\mathcal{H}^{L,d} = \bigcup_{L-d+1\leqslant|\mathbf{i}|\leqslant L} (\chi^{i_1} \times \cdots \times \chi^{i_d}), \tag{3.9}$$

where χ^i denotes the set of interpolation points used by $\mathcal{Q}^i$. According to (3.8), to extend the Smolyak interpolant $\mathcal{I}^{L,d}(u)$ from level $L-1$ to L, one only needs to evaluate the function at the incremental grid $\Delta\mathcal{H}^{L,d}$ defined by

$$\Delta\mathcal{H}^{L,d} = \bigcup_{|\mathbf{i}|=L} (\Delta\chi^{i_1} \times \cdots \times \Delta\chi^{i_d}), \tag{3.10}$$

where $\Delta\chi^{i_j} = \chi^{i_j} \setminus \chi^{i_j-1}$, $j = 1, \ldots, d$. According to the nested structure of the one-dimensional hierarchical grid defined by (3.3), it is easy to see that $\chi^{i-1} \subset \chi^i$ and $\Delta\chi^i = \chi^i \backslash \chi^{i-1}$ has $m^i_\Delta = m^i - m^{i-1}$ points. By consecutively numbering the points in $\Delta\chi^i$, and denoting the jth point of $\Delta\chi^i$ as x^i_j, the incremental interpolant in (3.7) can be represented by (see [5, 24] for details)

$$\Delta^i(u) = \sum_{j=1}^{m^i_\Delta} \phi^i_j \cdot \left[u(x^i_j) - \mathcal{Q}^{i-1}(u)(x^i_j)\right], \tag{3.11}$$

where $\omega^i_j = u(x^i_j) - \mathcal{Q}^{i-1}(u)(x^i_j)$ is defined as the one-dimensional hierarchical surplus on level i. This is just the difference between the values of the interpolating polynomials and the function evaluated at x^i_j. From (3.11), the hierarchical sparse grid interpolant (3.8) can be rewritten as

$$\begin{aligned} \mathcal{I}^{L,d}(u) &= \mathcal{I}^{L-1,d}(u) + \sum_{|\mathbf{i}|=L} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d})(u) \\ &= \mathcal{A}^{L-1,d}(u) + \sum_{\substack{|\mathbf{i}|=L \\ \mathbf{j}\in B_\mathbf{i}}} \omega^\mathbf{i}_\mathbf{j} \cdot \phi^\mathbf{j}_\mathbf{i}(\boldsymbol{x}) \\ &= \sum_{|\mathbf{i}|\leqslant L} \sum_{\mathbf{j}\in B_\mathbf{i}} \omega^\mathbf{i}_\mathbf{j} \cdot \phi^\mathbf{j}_\mathbf{i}(\boldsymbol{x}), \end{aligned} \tag{3.12}$$

where the multi-index set $B_\mathbf{i}$ is

$$B_\mathbf{i} = \left\{\mathbf{j} \in \mathbb{N}^d : x^{i_k}_{j_k} \in \Delta\chi^{i_k} \text{ for } j_k = 1, \ldots, m^{i_k}_\Delta, k = 1, \ldots, d\right\}, \tag{3.13}$$

and the surpluses $\omega^\mathbf{i}_\mathbf{j}$ are

$$\omega^\mathbf{i}_\mathbf{j} = u(x^{i_1}_{j_1}, \ldots, x^{i_d}_{j_d}) - \mathcal{I}^{L-1,d}(u)(x^{i_1}_{j_1}, \ldots, x^{i_d}_{j_d}). \tag{3.14}$$

As proved in [5], for smooth functions, the hierarchical surpluses tend to zero as the interpolation level tends to infinity. On the other hand, the magnitude of the surplus is a good indicator about the smoothness of the interpolated function. In general, the larger the magnitude, the stronger the underlying discontinuity.

For a bounded box domain $\mathcal{D} \subset \mathbb{R}^d$:

$$\mathcal{D} = [\boldsymbol{a}, \boldsymbol{b}] := \Pi_{i=1}^{d}[a_i, b_i], \tag{3.15}$$

we first transform it to $[-1, 1]^d$ through a simple linear transform. The corresponding sparse grid interpolation is then defined according to (3.6).

3.2. Hierarchical Sparse Grid Approximation of Bell-Shaped Curves. In many nonlinear filtering problems in practical applications, the conditional target state PDF resembles a "bell-shaped" Gaussian curve or surface, if not exactly Gaussian. In such cases, standard hierarchical sparse grid interpolations may lead to large errors. This is especially the case for two or higher dimensional problems. As a demonstration, we consider a Gaussian-type function

$$U(x) \doteq \prod_{i=1}^{d} \exp\left\{-\frac{1}{2}\left(\frac{x_i - \mu}{\sigma}\right)^2\right\}, \quad x \in \mathbb{R}^d, \tag{3.16}$$

where $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$. From [5], we know that the L^2 error between the function U and the standard sparse grid approximation $\bar{U}$ is

$$\|U - \bar{U}\|_{L^2} \leq \frac{2 \cdot |U|_{\mathbf{2},2}}{12^d} \cdot 2^{-2L} \cdot A(d, L), \tag{3.17}$$

where $L \in \mathbb{N}^+$ represents the level of hierarchical sparse grid with

$$A(d, L) = \frac{L^d}{(d-1)!} + \mathcal{O}(k^{d-2}), \text{ and } |U|_{\mathbf{2},2} = \|D^{\mathbf{2}}U\|_{L^2}.$$

It can be shown that $|U|_{\mathbf{2},2}$ in (3.17) is bounded by (see [29] for detailed derivation)

$$|U|_{\mathbf{2},2} \leq \frac{1}{\sigma^{4d}}(\max\{x_i - \mu\}^{2d})\|U\|_{L^2}. \tag{3.18}$$

We can see from the above estimate that when the constant σ is small, the right hand side becomes exponentially large as d increases.

Here, we consider a special case of (3.16) with $\mu = 0$ and $\sigma = 0.15$, i.e.,

$$U(x) = \prod_{i=1}^{d} \exp\left(-\frac{x_i^2}{0.045}\right). \tag{3.19}$$

Figure 1 shows the interpolation errors using the L_2-norm for regular sparse grid approximations of U. We can see that as the dimension d increases, the interpolation errors barely decrease even, though the number of interpolation points has increased significantly.

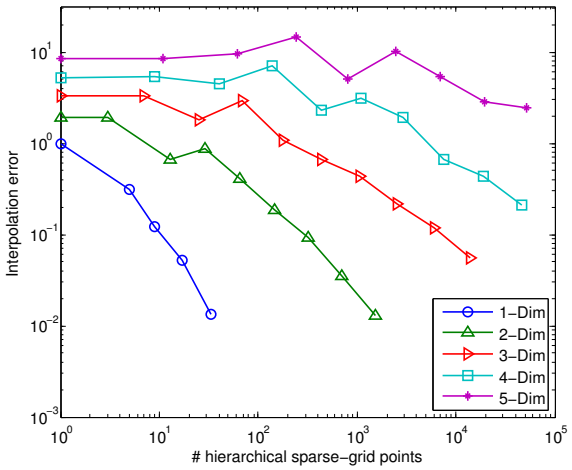


Figure 1: The interpolation error for regular sparse grid approximation up to 5 dimensions.

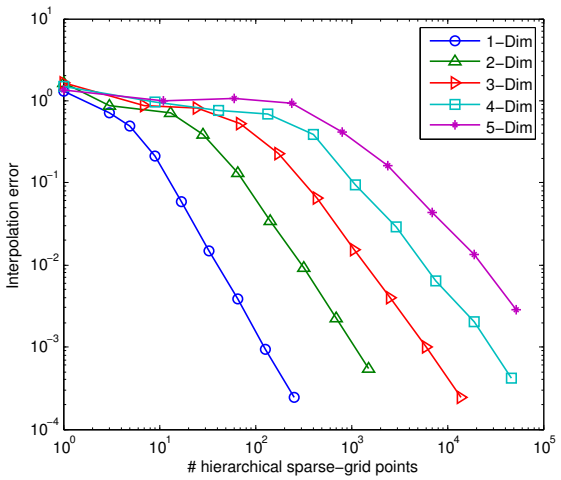


Figure 2: The interpolation error for logarithmic sparse grid approximation up to 5 dimensions.

One way of alleviating this poor performance of the sparse grid interpolant when approximating bell-shaped-functions is to utilize logarithmic interpolation (see [29]), in which we take the logarithm of U, i.e. $V \doteq \log(U)$, and build the approximation $\bar{V}$ using the standard sparse grid approach. Then we obtain the approximation of U by $\bar{U} = e^{\bar{V}}$. Figure 2 shows the absolute interpolation errors measured in L_2-norm for the logarithmic sparse grid approximation of function U defined in (3.19). Compared with Fig. 1, we can see from Figure 2 that the convergence is improved as the dimension increases.

A visual demonstration of the efficiency of the logarithmic sparse grid interpolation is shown in Fig. 3, where we plot the level 6 approximation for a marginal distribution of both U and $\log(U)$ in $d = 4$ dimensions.

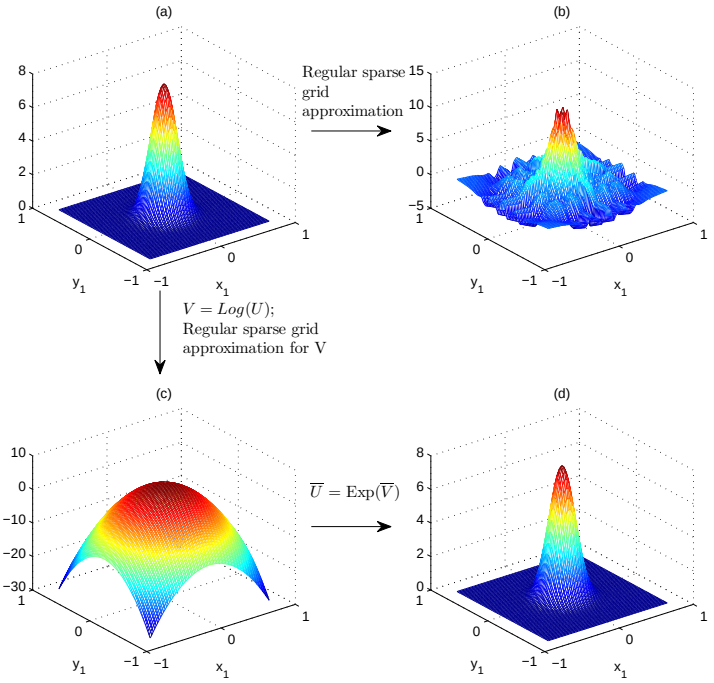


Figure 3: (a) is the x_1x_2-marginal surface of U with $x_3 = x_4 = 0$ while (b) is its interpolate approximation obtained by applying the regular sparse grid approximation to function U. We can see from the Figure that the approximation is quite poor and significant oscillations occur at the bottom of the surface. On the other hand, from (d) one can see that the logarithmic interpolation approximation described above is far more accurate.

4. Hybrid approach for numerical solution of the nonlinear filtering problem..

In this section we describe our hybrid numerical algorithm for the solution of the Zakai equation (2.2).

4.1. Adaptive selection of solution domains.

Since the Zakai equation (2.2) is defined on the whole space R^d, it is essential to choose a proper bounded domain to achieve an accurate numerical approximation. Motivated by importance sampling and particle filter methods, we adaptively select a hypercube at each time step according to an estimation of the density function of the solution at the next step.

In particular, let $\mathcal{R}_t$ be a partition of $[0, T]$ such that:

$$\mathcal{R}_t = \{t_n | t_n \in [0, T], t_n < t_{n+1}, n = 0, 1, \cdots, N_T - 1, t_0 = 0, t_{N_T} = T\}$$

and denote $\Delta t_n = t_{n+1} - t_n$, $n = 0, 1, \cdots, N_T - 1$. For $n = 0, 1, \cdots N_T$, assume that u_n is the numerical solution of the Zakai equation (2.2) at t_n. We use an importance sampling method to draw M realizations according to the conditional PDF u_n of the state X_n, denoted by $\{p_n^m\}_{m=1,\cdots,M}$, where M is a pre-defined positive integer. We then propagate each of these samples from time step t_n to t_{n+1} using the *state equation* in the nonlinear filtering problem

(1.1) to get M updated sample points, denoted by $\{p_{n+1}^m\}_{m=1,\cdots,M}$. To complete our adaptive domain selection, we denote

$$\mathcal{D}_{n+\frac{1}{2}} = [\boldsymbol{a}_{n+\frac{1}{2}}, \boldsymbol{b}_{n+\frac{1}{2}}] \subset \mathbb{R}^d$$

as the smallest hypercube containing all the samples $\{p_{n+1}^m\}_{m=1,\cdots,M}$ and $\boldsymbol{\Sigma} = (\Sigma^1, \ldots, \Sigma^d)$ as the vector of marginal standard deviations of these samples. Then, for a user defined positive constant λ we let

$$\begin{aligned} \boldsymbol{a}_{n+1} &\doteq \boldsymbol{a}_{n+\frac{1}{2}} - \lambda\boldsymbol{\Sigma} \\ \boldsymbol{b}_{n+1} &\doteq \boldsymbol{b}_{n+\frac{1}{2}} + \lambda\boldsymbol{\Sigma} \end{aligned} \tag{4.1}$$

and finally we choose

$$\mathcal{D}_{n+1} = [\boldsymbol{a}_{n+1}, \boldsymbol{b}_{n+1}] \tag{4.2}$$

as the solution domain for u_{n+1}.

Remark 1. *The idea of the adaptive selection of solution domain $\mathcal{D}_n$ is similar to that in the prediction step of particle filter method. As such the region $\mathcal{D}_{n+\frac{1}{2}}$ which includes all updated samples is similar to the domain where particle filter method builds the prior PDF of the target state. In our approach, we choose a confidence region surrounding $\mathcal{D}_{n+\frac{1}{2}}$ as our solution domain. This way, we can use a much smaller number of samples than the particle filter technique and maintain the accuracy of our approximation. In addition, our numerical experiments also indicate that this adaptive domain selection approach is more accurate at predicting the tail distribution than the particle filter method.*

4.2. Spliting-up finite difference method on sparse grid. Following [2, 23, 36], the splitting-up scheme for (2.2) consists of prediction and update steps, described in Sections 4.2.1 and 4.2.2 respectively. At each time step t_n, we define the sampled observation Z_n from the observation process Y_t by

$$Z_n \doteq \frac{Y_{t_{n+1}} - Y_{t_n}}{\Delta t_n} = \frac{1}{\Delta t_n} \cdot \left(\int_{t_n}^{t_{n+1}} h(X_s)ds + B_{t_{n+1}} - B_{t_n} \right).$$

In what follows, we denote u_n to be the approximate solution for u_t at $t = t_n$, $n = 0, 1, 2, \cdots, N_T$.

4.2.1. Prediction Step. In the prediction step we solve for $u_{n+\frac{1}{2}}$ from

$$u_{n+\frac{1}{2}} = u_n + L^* u_n \Delta t_n, \tag{4.3}$$

which is equivalent to solving the deterministic PDE, known as the Fokker Plank equation [34, 36],

$$\frac{\partial u_t}{\partial t} = L^* u_t, \quad t_n \le t \le t_{n+1} \tag{4.4}$$

with an one-step forward Euler scheme. An efficient spatial discretization technique for the Fokker-Planck equation (4.4) in the prediction step is essential to the success of the splitting-up scheme. In the one-dimensional case, a simple finite difference method to discretize the

operator L is sufficient. In a straight forward fashion, such a discretization can be extended to multi-dimensional case with use of a direct tensor product. However, the computational cost of the numerical solution based on a tensor-product approximation increases exponentially as the dimension d increases, known as the "*curse of dimensionality*". To alleviate this numerical challenge and reduce the overall computational complexity, we use the sparse-grid method described in section 3 to construct a finite difference algorithm for solving (4.4).

To see this, we present the upwind finite difference method for approximating the partial differential operator L defined in (2.3) on a sparse grid. First, let u_0 be the initial value of the solution u of the Zakai equation (2.2). For a positive integer n, let $\mathcal{H}_n^{L,d}$ be the set of sparse grid points defined by (3.10) in the hypercube $\mathcal{D}_n$ defined by (3.9). For $n = 0, \cdots, N_T$ and $x \in \mathcal{H}_{n+1}^{L,d}$, we approximate the first order partial derivative with given coefficient function $\mu \in \mathbb{R}^d$ by

$$\mu_i \frac{\partial u_n}{\partial x_i}(x) \approx \mu_i \tilde{D}_{x_i} u_n(x) \doteq \begin{cases} \mu_i \dfrac{\hat{u}_n(x + e_i h_i) - u_n(x)}{h_i} & \text{if } \mu_i \geq 0 \\ \mu_i \dfrac{u_n(x) - \hat{u}_n(x - e_i h_i)}{h_i} & \text{if } \mu_i < 0 \end{cases},$$

where e_i is the unit vector in the ith coordinate direction and h_i is a properly chosen meshsize in the ith coordinate direction.

To approximate second order partial derivatives at the sparse grid point $x \in \mathcal{H}_{n+1}^{L,d}$ with given coefficient function $\alpha \in \mathbb{R}^{d \times d}$, we use central differences to obtain

$$\alpha_{i,i} \frac{\partial^2 u_n}{\partial x_i \partial x_i}(x) \approx \alpha_{i,i} \tilde{D}^2_{x_i x_i} u_n(x) \doteq \alpha_{i,j} \frac{\hat{u}_n(x + e_i h_i) - 2u_n(x) + \hat{u}_n(x - e_i h_i)}{h_i^2}$$

and

$$\begin{aligned} &\alpha_{i,j} \frac{\partial^2 u_n}{\partial x_i \partial x_j}(x) \approx \alpha_{i,j} \tilde{D}^2_{x_i x_j} u_n(x) \\ &\doteq \begin{cases} \dfrac{\alpha_{i,j}}{2h_i} \left[\dfrac{\hat{u}_n(x + e_i h_i + e_j h_j) - \hat{u}_n(x + e_i h_i)}{h_j} - \dfrac{\hat{u}_n(x + e_j h_j) - u_n(x)}{h_j} \right. \\ \left. + \dfrac{u_n(x) - \hat{u}_n(x - e_j h_j)}{h_j} - \dfrac{\hat{u}_n(x - e_i h_i) - \hat{u}_n(x - e_i i - e_j h_j)}{h_j} \right], \text{if } \alpha_{i,j} \geq 0, \\ \dfrac{\alpha_{i,j}}{2h_i} \left[\dfrac{\hat{u}_n(x + e_i h_i) - \hat{u}_n(x + e_i h_i - e_j h_j)}{h_j} - \dfrac{u_n(x) - \hat{u}_n(x - e_j h_j)}{h_j} \right. \\ \left. + \dfrac{\hat{u}_n(x + e_j h_j) - u_n(x)}{h_j} - \dfrac{\hat{u}_n(x - e_i h_i + e_j h_j) - \hat{u}_n(x - e_i h_i)}{h_j} \right], \text{if } \alpha_{i,j} < 0. \end{cases} \end{aligned}$$

With the above finite difference operators in hand, we define the finite difference approximation of the Fokker-Planck equation (4.4) on sparse grid $\mathcal{H}_{n+1}^{L,d}$ as follows.

$$u_{n+\frac{1}{2}}(x) = u_n(x) + L_n^* u_n(x) \Delta t_n, \quad x \in \mathcal{H}_{n+1}^{L,d}, \tag{4.5}$$

where

$$L_n^* u_n(x) = \frac{1}{2}\sum_{i,j}^{d}\left\{\frac{\partial^2(\sigma\sigma^*)_{i,j}}{\partial x_i \partial x_j}u_n(x) + \frac{\partial(\sigma\sigma^*)_{i,j}}{\partial x_i}\tilde{D}_{x_j}u_n(x) + \frac{\partial(\sigma\sigma^*)_{i,j}}{\partial x_j}\tilde{D}_{x_i}u_n(x) + (\sigma\sigma^*)_{i,j}\tilde{D}^2_{x_i x_j}u_n(x)\right\} - \sum_{i=1}^{d}\left(\frac{\partial b^i}{\partial x_i}u_n(x) + b^i\tilde{D}_{x_i}u_n(x)\right).$$

4.2.2. Update Step. In the update step, we use the new observation Z_n and the Bayes formula [36] to update the prior $u_{n+\frac{1}{2}}$ to the posterior u_{n+1} as follows.

$$u_{n+1}(x) = C_n\Psi^n(x, Z_n)u_{n+\frac{1}{2}}, \quad x \in \mathcal{H}^{L,d}_{n+1}, \tag{4.6}$$

where C_n is a normalization factor and function Ψ^n is defined by

$$\Psi^n(x, Z_n) = \exp\left\{-\frac{\Delta t_n}{2}\cdot|Z_n - h(x)|^2_R\right\}.$$

The norm $|\cdot|_R$ is defined by $|\alpha|^2_R = \alpha R^{-1}\alpha$, where R is the covariance matrix of $\{B_t, t \geq 0\}$ in (1.1) Please see Page 2, [36] for more details. Finally following the procedure described in Section 3, we derive the logarithmic sparse grid interpolation $u_{n+1} = u_{n+1}(x),\ x \in \mathbb{R}^d$ using its values on the sparse grid $\mathcal{H}^{L,d}_{n+1}$.

We summarize our hybrid sparse grid adaptive-domain splitting-up finite difference algorithm as follows:

Step 1: Input u_0 as the initial value of the solution u of the Zakai equation (2.2).

Step 2: For $n = 0\cdots, N_T - 1$,

1 Compute dynamic domain $\mathcal{D}_{n+1}$ for the solution u_{n+1} using the importance sampling method.

2 Generate sparse grid $\mathcal{H}^{L,d}_{n+1}$ on the solution domain $\mathcal{D}_{n+1}$.

3 Evaluate u_{n+1} on the sparse grid $\mathcal{H}^{L,d}_{n+1}$ by using finite difference scheme (4.5).

4 Extend the solution u_{n+1} to the whole space R^d through the logarithmic sparse grid interpolation described in Section 3.

Step 3: Normalization.

Remark 2. *Since we use an explicit finite difference scheme to solve equation (4.4), time step Δt_n must satisfy the following stability condition*

$$\max_{0\leq n\leq N_T-1}\Delta t_n \leq \frac{1}{\sum_{i=1}^{d}\frac{|(\sigma\sigma^T)_{i,i}| + |b^i|h_i}{h_i^2}}.$$

5. Numerical Experiments. In this section, we present three numerical experiments to demonstrate the effectiveness of our new numerical algorithm, for solving nonlinear filtering problems.

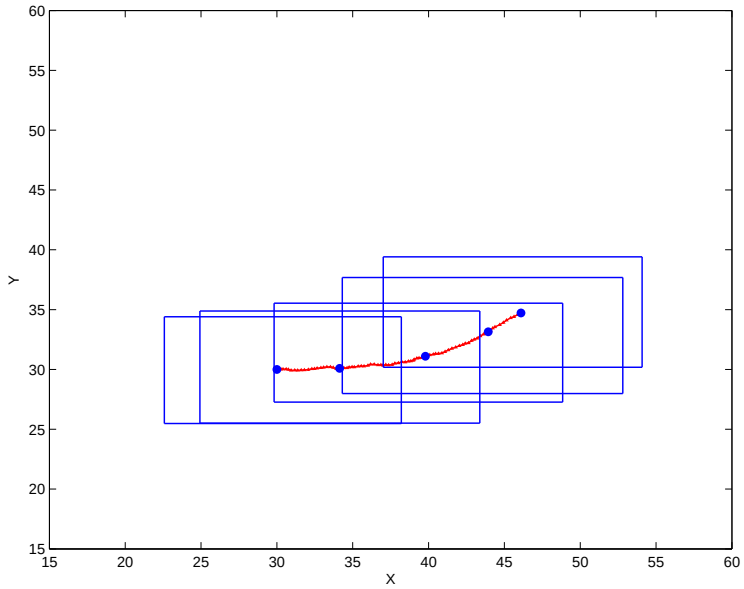


Figure 4: Target Trajectory and Adaptive Solution Domain. The red curve shows the real target state. The blue points are actual states of the target and the blue boxes are the corresponding solution domains.

Example 1 . In the first example, we use a two dimensional nonlinear filtering problem to illustrate the accuracy of the selection process of the dynamic solution domain $\mathcal{D}_n$. To see this, we consider the following dynamical system

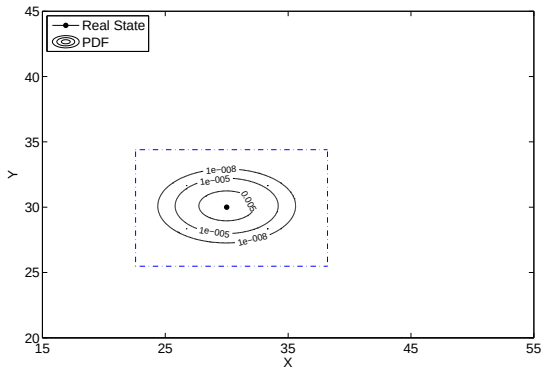
$$dX_t = (40, 2 \cdot (10t)^2)^T \, dt + 0.5 \, dW_t, \tag{5.1}$$

where W_t is a two-dimensional Brownian Motion and the initial state is given by $X_0 = (30, 30)^T$. The observation process is given by

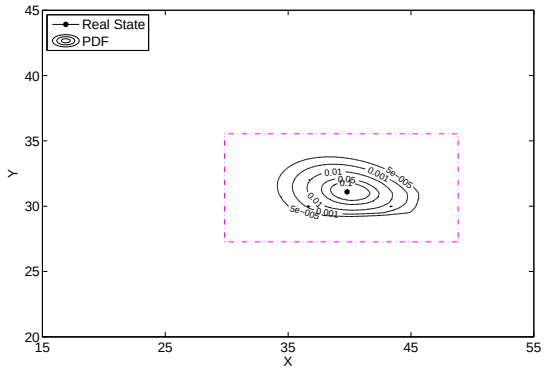
$$dY_t = \sqrt{(X_t^1 - 20)^2 + (X_t^2)^2} \cdot dt + dB_t, \tag{5.2}$$

which measures the perturbed distance between the target state and a reference point $P = (20, 0)$, and B_t is a one-dimensional Brownian motion independent of W_t.

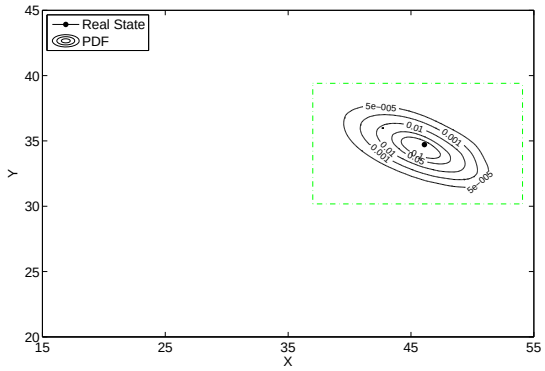
In this numerical simulation, we take $T = 0.4$ and use an uniform partition in time with stepsize $\Delta t_n = 0.005$. The initial value is given by $u_0 \sim N(X_0, \Sigma)$; a normal distribution with mean X_0 and standard deviation $\Sigma = (1, 0.5)^T$. In the adaptive solution domain selection process, we choose the sample size $M = 500$ and the parameter λ in (4.1) as $\lambda = 4$. Figure 4 shows the trajectory of the target state and solution domain $\mathcal{D}_n$ for $n = 1, 20, 50, 70, 80$. In Figure 5 we show solution domain $\mathcal{D}_n$ with the corresponding contour plot of target state PDF for $n = 1, 50, 80$. From this figure one can see the that solution domains are extremely accurate in approximating the high density area of u_n.



(a)



(b)

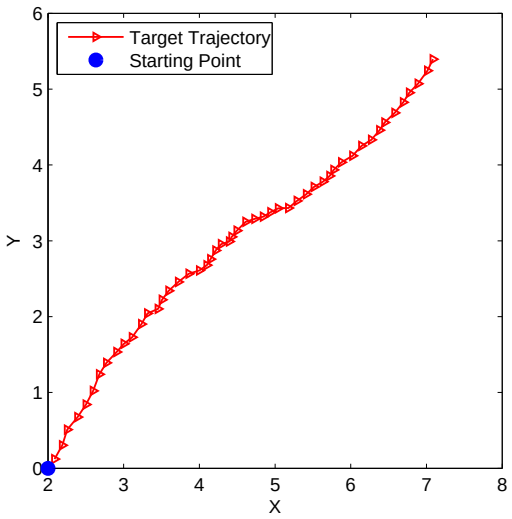


(c)

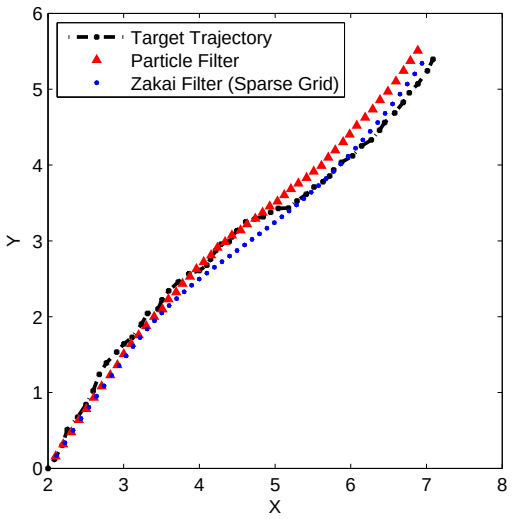
Figure 5: Target state PDF at time step: (a) n=1; (b) n=50; (c)=80; in the corresponding solution domain

Example 2. In this example, we consider the following nonlinear filtering problem:

$$\begin{cases} dX_t = \left(10, 6 \cdot (\sin X_t^1 + 2)\right)^T dt + dW_t, & (state) \\ dY_t = \left(X_t^1, X_t^2\right)^T dt + dB_t. & (observation) \end{cases} \tag{5.3}$$



(a) Signal trajectory: Example 2



(b) Comparison of estimated mean of the target state between particle filter and sparse grid Zakai filter

Figure 6

Table 1: Comparison of computing costs

Particle filter	CPU time (seconds)
2,000 particles	1.49
8,000 particles	32.67
40,000 particles	516.91
160,000 particles	7622.29
Sparse grid Zakai filter	20.45

In this numerical experiment, we let $T = 0.5$ and use a uniform partition in time with stepsize $\Delta t_n = 0.01$. An example of the signal trajectory is shown in figure 6a. The initial value is given by $u_0 \sim N(X_0, \Sigma)$; a normal distribution with mean $X_0 = (2, 0)^T$ and standard deviation $\Sigma = (0.5, 0.5)^T$. We also choose the hierarchical sparse grid level as $L = 6$. In the adaptive solution domain selection process, the sample size and parameter λ are given by $M = 500$ and $\lambda = 4$, respectively.

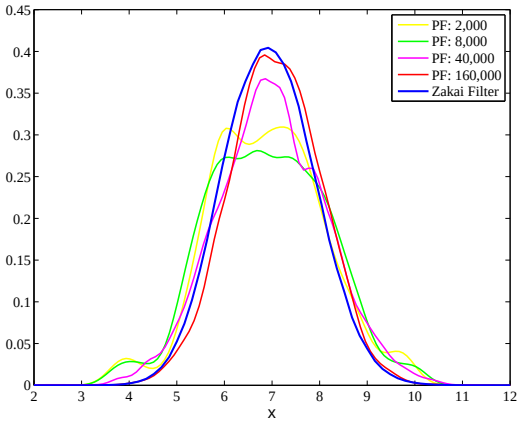
Figure 6b shows the comparison of the estimated values of the target state between the particle filter and sparse grid Zakai filter. In the particle filter simulation, we use 8000 particles to represent the PDF of the target state. The black dashed line shows the trajectory of the real target state, the red triangles and blue dots show the estimate target state (the mean of the estimate posterior PDF) obtained by using the particle filter and the Zakai filter respectively. As we can see from Figure 6b, our hybrid approach is more accurate than the PFM for a longer period of time.

To further examine the performance of the hybrid sparse grid Zakai filter method, in Figure 7a and Figure 7b we plot the marginal probability density functions at time $T = 0.5$, obtained by using the particle filter and our hybrid with respect to xy-coordinates. From the plots we observe that the convergence of the particle filter as the particle size increases. Moreover, the PDE obtained by our hybrid sparse grid Zakai filter is very close to the one obtained by the particle filter with $160,000$ particles. However, in table 1 we can see that with $160,000$ particles the particle filter is far more costly in terms of CPU time than our method. Finally, we also show the confidence bands in Figure 8. The blue dashed curves show the real target trajectory with respect to xy-coordinates, respectively, the red curves represent the estimate of posterior means, and the green dashed curves represent the estimated 95% confidence bands.

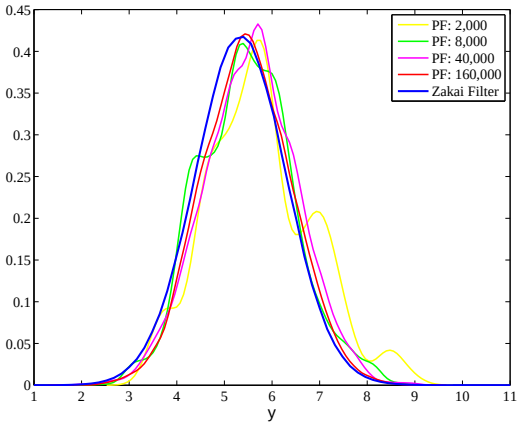
Example 3. In this example, we consider the following " bearing-only " tracking problem given by:

$$dX_t = bdt + \sigma dW_t, \tag{5.4}$$

where $X_t = (x, y, u, v)_t^T$ is a four-dimensional vector which models the movement of a target ship sailing on the sea plane ($x - y$ plane). Here (x, y) and (u, v) are the position and velocity



(a) Marginal distribution with respect to X-coordinate



(b) Marginal distribution with respect to Y-coordinate

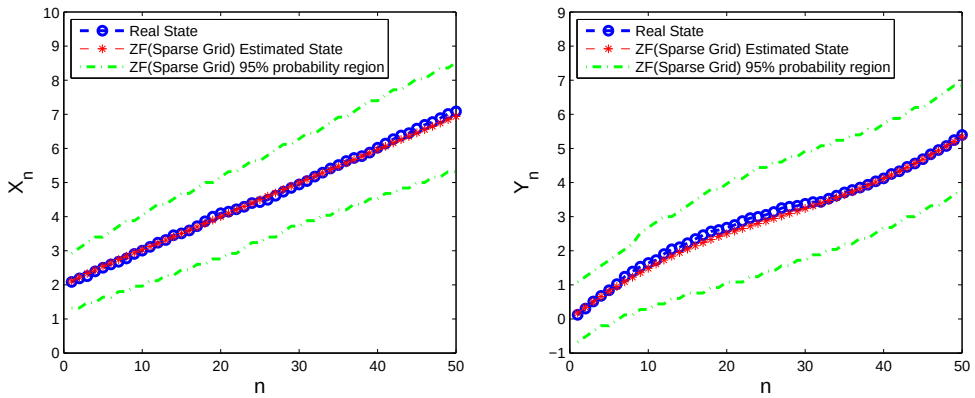
Figure 7: Marginal distributions

components respectively, the vector $b = (u, v, 0, 0)_t^T$, the covariance matrix σ^2 is defined by

$$\sigma^2 = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix}, \tag{5.5}$$

and W_t is a four-dimensional Brownian motion. To estimate the target state, a passive sonar is located on an observation ship, denoted "ownship". The observation process is given by:

$$dY_t = h(X_t)dt + dB_t,$$



(a) (b)

Figure 8: Sparse grid Zakai filter estimate of posterior mean with 95% probability region .
(a) Estimate of posterior mean and 95% probability region: X-coordinate
(b) Estimate of posterior mean and 95% probability region: Y-coordinate

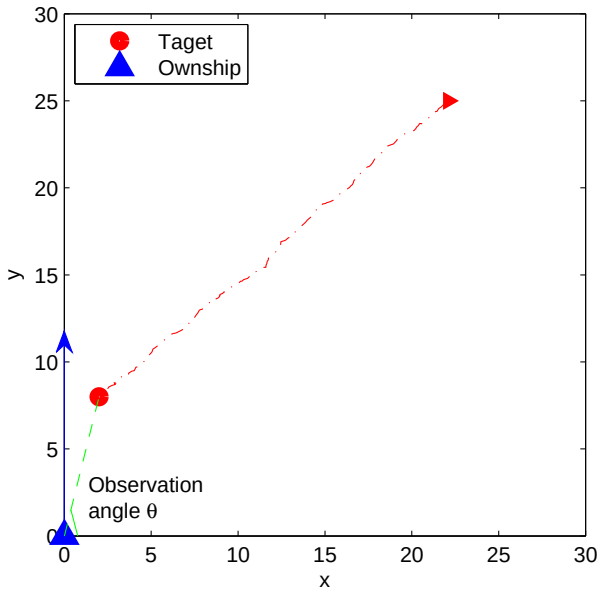


Figure 9: Bearing-only tracking

where the observation function h is the angle

$$h(X_t) = \arctan\left(\frac{y_t - y_t^{obs}}{x_t - x_t^{obs}}\right),$$

and B_t is a one-dimensional Brownian motion independent from W_t. Here, $X_t^{obs} = \left(x^{obs}, y^{obs}\right)_t^T$ describes the movement of the ownship given by $dX_t^{obs} = b^{obs}dt$.

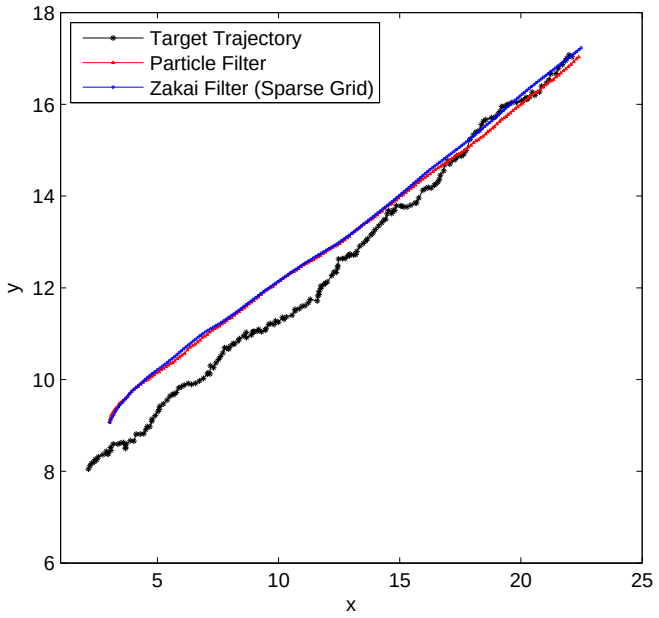


Figure 10: Comparison of estimated mean of the target state between particle filter and sparse grid Zakai filter

In the numerical simulations, we choose $\sigma_1 = \sigma_2 = 0.75$, $\sigma_3 = \sigma_4 = 0.05$ in (5.5) and $b^{obs} = (8, 0)^T$; thus the movement of the ownship is along the Y-axis with a constant speed. In addition, we set $T = 1$, the time partition $\Delta t_n = 0.005$, initial target state $X_0 = (2, 8, 20, 8)^T$, and the initial PDF of the target sate $N(\bar{X}, \Sigma)$, where $\bar{X} = (3, 9, 19.8, 7.9)^T$ and $\Sigma = (0.75, 0.75, 0.5, 0.25)^T$.

The target-observer plane ($x - y$ plane) is illustrated in Figure 9. The red dot shows the initial position of the target ship while the blue triangle shows the initial position of our ownship. The dashed red curve gives a possible trajectory of the target ship and the blue arrow describes the movement of our ownship.

Figure 10 shows the comparison of the estimated mean values of the relative target position with respect to the ownship, in the target-observer plane. For the sparse grid Zakai filter, we let the hierarchical level $L = 6$ and for the adaptive solution domain selection process, we set $M = 1,000$ and $\lambda = 4$. The estimate for the particle filter is obtained by using $160,000$ particles. The actual trajectory of the target position relative to our ownship is given by the black curve. The red curve and blue curve show the estimate target state (the mean of the estimate posterior PDF) obtained by using the particle filter and the Zakai filter respectively. We can see from Figure 10 that our hybrid sparse grid Zakai filter yields similar estimate results compared with the particle filter with $160,000$ particles.

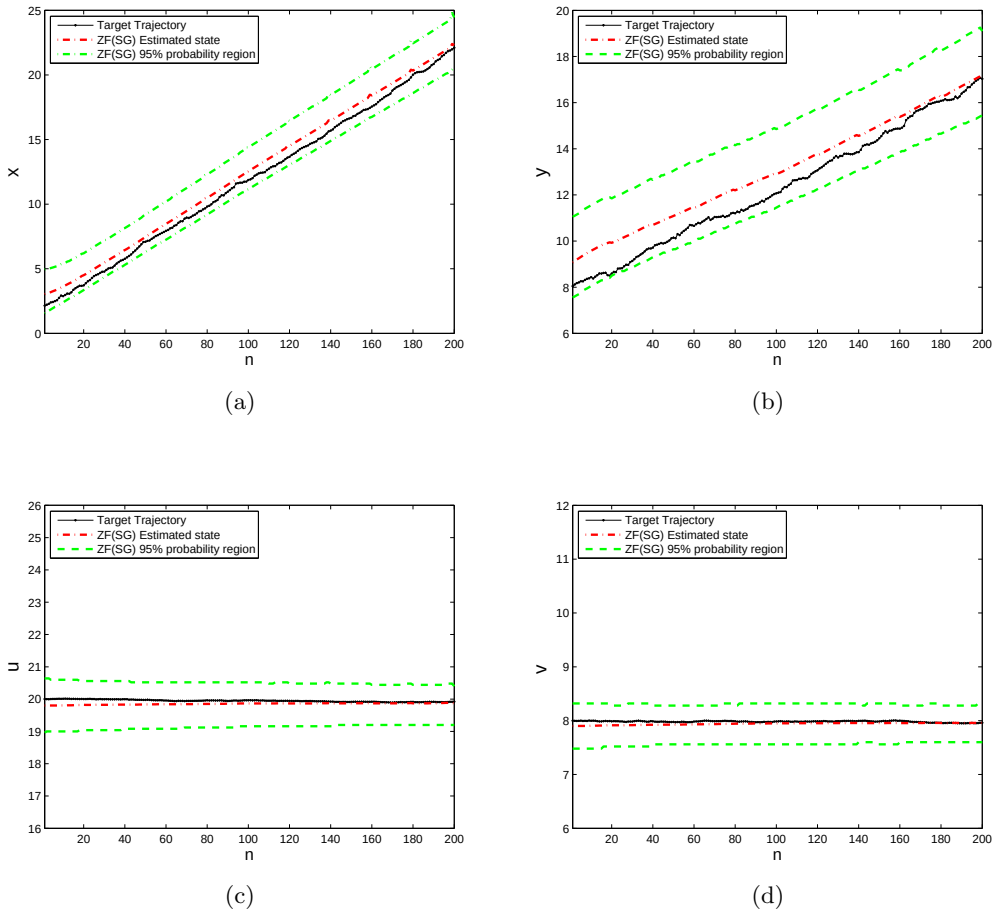
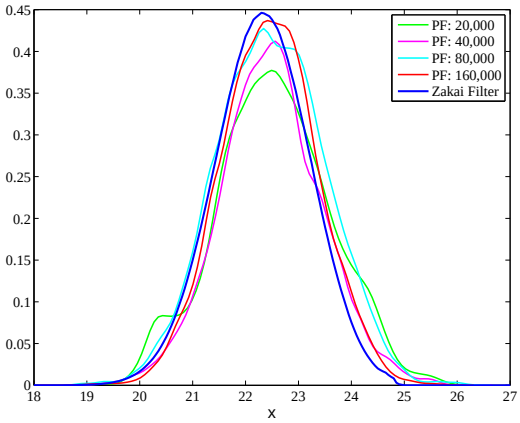
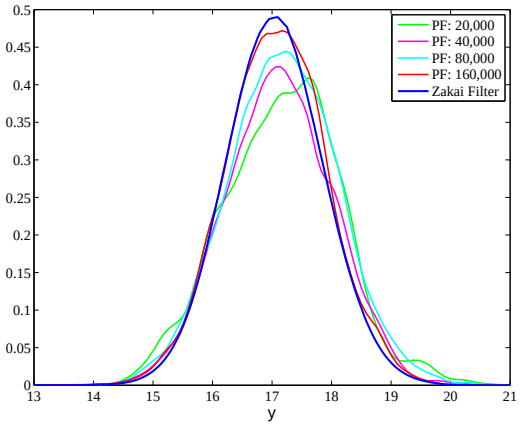


Figure 11: Sparse grid Zakai filter estimate of posterior mean with 95% probability region. (a) Estimate of posterior mean of the location x and 95% probability region. (b) Estimate of posterior mean of the location y and 95% probability region. (c) Estimate of posterior mean of the velocity u and 95% probability region. (d) Estimate of posterior mean of the velocity v and 95% probability region

In Figure 12a and 12b we compare the marginal probability distribution functions at time $T = 1$ with respect to xy coordinates using our hybrid sparse grid Zakai filter with the particle filter. Similar to Example 2, we see that the convergence of the particle filter with the particle size $160,000$ to the approximate PDE is very close to the approximate PDE obtained by the hybrid sparse grid Zakai filter. In table 2, we show the comparison of computational from which one can see that the particle filter with $160,000$ particles takes 60 times more computing time to achieve a similar PDF. We also plot the confidence curves in Figure 11 for further examination of the performance of sparse grid Zakai filter. The blue dashed curves show the real target trajectory with respect to xy coordinates. The red curves represent the estimate



(a) Marginal distribution of the target state on x-coordinate



(b) Marginal distribution of the target state on y-coordinate

Figure 12: Marginal distributions

of posterior means, and the green dashed curves give the estimate 95% confidence curves (± 2 times of the estimated standard deviation).

REFERENCES

[1] Yaakov Bar-Shalom and Thomas E. Fortmann, *Tracking and data association*, vol. 179 of Mathematics in Science and Engineering, Academic Press Inc., San Diego, CA, 1988.

[2] A. Bensoussan, R. Glowinski, and A. Răşcanu, *Approximation of some stochastic differential equations by the splitting up method*, Appl. Math. Optim., 25 (1992), pp. 81–106.

[3] Miodrag Bolić, Petar M. Djurić, and Sangjin Hong, *Resampling algorithms and architectures for distributed particle filters*, IEEE Trans. Signal Process., 53 (2005), pp. 2442–2450.

[4] A. Budhiraja and G. Kallianpur, *Approximations to the solution of the Zakai equation using multiple Wiener and Stratonovich integral expansions*, Stochastics Stochastics Rep., 56 (1996), pp. 271–315.

Table 2: Comparison of computing costs

Particle filter	CPU time (seconds)
20,000 particles	554.24
40,000 particles	1976.39
80,000 particles	7702.57
160,000 particles	32380.38
Sparse grid Zakai filter	587.14

[5] Hans-Joachim Bungartz and Michael Griebel, *Sparse grids*, Acta Numer., 13 (2004), pp. 147–269.

[6] Zhe Chen, *Bayesian filtering: from kalman filters to particle filters, and beyond*, Unpublished manuscript, (2011).

[7] Dan Crisan, *Exact rates of convergence for a branching particle approximation to the solution of the Zakai equation*, Ann. Probab., 31 (2003), pp. 693–718.

[8] D. Crisan and O. Obanubi, *Particle filters with random resampling times*, Stochastic Process. Appl., 122 (2012), pp. 1332–1368.

[9] A. Davie and J. Gaines, *Convergence of numerical schemes for the solution of the parabolic stochastic partial differential equations*, Math. Comp., 70 (2001), pp. 121–134.

[10] Jindřich Duník, Miroslav Šimandl, and Ondřej Straka, *Unscented Kalman filter: aspects and adaptive setting of scaling parameter*, IEEE Trans. Automat. Control, 57 (2012), pp. 2411–2416.

[11] Garry A. Einicke, *Asymptotic optimality of the minimum-variance fixed-interval smoother*, IEEE Trans. Signal Process., 55 (2007), pp. 1543–1547.

[12] Emmanuel Gobet, Gilles Pagès, Huyên Pham, and Jacques Printems, *Discretization and simulation of the Zakai equation*, SIAM J. Numer. Anal., 44 (2006), pp. 2505–2538 (electronic).

[13] N.J Gordon, D.J Salmond, and A.F.M. Smith, *Novel approach to nonlinear/non-gaussian bayesian state estimation*, IEE PROCEEDING-F, 140 (1993), pp. 107–113.

[14] W. Grecksch and P. E. Kloeden, *Time-discretised Galerkin approximations of parabolic stochastic PDEs*, Bull. Austral. Math. Soc., 54 (1996), pp. 79–85.

[15] István Gyöngy, *Lattice approximations for stochastic quasi-linear parabolic partial differential equations driven by space-time white noise. II*, Potential Anal., 11 (1999), pp. 1–37.

[16] István Gyöngy and David Nualart, *Implicit scheme for quasi-linear parabolic partial differential equations perturbed by space-time white noise*, Stochastic Process. Appl., 58 (1995), pp. 57–72.

[17] Kallianpur G. Hu, Y. and J. Xiong, *An approximation for zakai equation*, Appl. Math. Optim., 45 (2002), pp. 23–44.

[18] Y. Hu, G. Kallianpur, and J. Xiong, *An approximation for the Zakai equation*, Appl. Math. Optim., 45 (2002), pp. 23–44.

[19] Adrew H. Jazwinski, *Stochastic Processing and Filtering Theory*, vol. 64, Academic Press, New York, 1973.

[20] Simon J. Julier and Joseph J. LaViola, Jr., *On Kalman filtering with nonlinear equality constraints*, IEEE Trans. Signal Process., 55 (2007), pp. 2774–2784.

[21] Xiong Kai, Liu Liangdong, and Liu Yiwu, *Robust extended Kalman filtering for nonlinear systems with multiplicative noises*, Optimal Control Appl. Methods, 32 (2011), pp. 47–63.

[22] R. E. Kalman and R. S. Bucy, *New results in linear filtering and prediction theory*, Trans. ASME Ser. D. J. Basic Engrg., 83 (1961), pp. 95–108.

[23] François Le Gland, *Time discretization of nonlinear filtering equations*, in Proceedings of the 28th IEEE Conference on Decision and Control, Vol. 1–3 (Tampa, FL, 1989), New York, 1989, IEEE, pp. 2601–2606.

[24] Xiang Ma and Nicholas Zabaras, *An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations*, J. Comput. Phys., 228 (2009), pp. 3084–3113.

[25] C. Johan Masreliez and R. Douglas Martin, *Robust Bayesian estimation for the linear model and*

robustifying the Kalman filter, IEEE Trans. Automatic Control, AC-22 (1977), pp. 361–371.

[26] Motoyasu Nagata and Yoshikazu Sawaragi, *Error analysis of the Schmidt-Kalman filter*, Internat. J. Systems Sci., 7 (1976), pp. 769–778.

[27] Bernt Øksendal, *Stochastic differential equations*, Universitext, Springer-Verlag, Berlin, sixth ed., 2003. An introduction with applications.

[28] Umut Orguner and Fredrik Gustafsson, *Target tracking with particle filters under signal propagation delays*, IEEE Trans. Signal Process., 59 (2011), pp. 2485–2495.

[29] Dirk Pflüger, (2010), p. 194. Thesis (Ph.D.)– Technische Universität München.

[30] Charles F. Price, *An analysis of the divergence problem in the Kalman filter*, IEEE Trans. Automatic Control, AC-13 (1968), pp. 699–702.

[31] Jacques Printems, *On the discretization in time of parabolic stochastic partial differential equations*, M2AN Math. Model. Numer. Anal., 35 (2001), pp. 1055–1078.

[32] Yener Ulker and Bilge Gunsel, *Multiple model target tracking with variable rate particle filters*, Digit. Signal Process., 22 (2012), pp. 417–429.

[33] Dongbin Xiu and Jan S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139 (electronic).

[34] Moshe Zakai, *On the optimal filtering of diffusion processes*, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete, 11 (1969), pp. 230–243.

[35] Guannan Zhang and Max Gunzburger, *Error analysis of a stochastic collocation method for parabolic partial differential equations with random input data*, SIAM J. Numer. Anal., 50 (2012), pp. 1922–1940.

[36] H. Zhang and D. Laneuville, *Grid based solution of zakai equation with adaptive local refinement for bearing-only tracking*, IEEE Aerospace Conference, (2008).