Sparse-Grid Approximation Methods for Numerical Solution of the Zakai Equation

Guannan Zhang

Computer Science and Mathematics Division
Oak Ridge National Laboratory

Collaborators
Feng Bao (ORNL), Clayton Webster (ORNL), Yanzhao Cao (Auburn)

May 15, 2015
Outline

1. Background of numerical methods for nonlinear filtering
2. Global sparse-grid polynomial approximation
3. Piecewise hierarchical sparse-grid polynomial approximation
4. Application of sparse-grid approximation for solving Zakai equations
5. Numerical examples
Outline

1. Background of numerical methods for nonlinear filtering
2. Global sparse-grid polynomial approximation
3. Piecewise hierarchical sparse-grid polynomial approximation
4. Application of sparse-grid approximation for solving Zakai equations
5. Numerical examples
The discrete formulation:

- The state process
  \[ X(k + 1) = h_k(X(k), w(k)) \quad \text{for} \quad k = 0, 1, 2, \ldots \]

- The observation process
  \[ Z(k) = g(X(k)) + \xi(k) \quad \text{for} \quad k = 0, 1, 2, \ldots \]

where \( w(k), \xi(k) \) are two independent zero mean white noises with variance \( \sqrt{\delta} \) and \( \delta \) is the sampling period.

The continuous formulation:

- The state equation
  \[ dX_t = b(X_t)dt + \sigma(X_t)dW_t \]

- The observation equation
  \[ dY_t = g(X_t)dt + dB_t \]

where \( W_t, B_t \) are two independent Wiener processes.
The goal of nonlinear filtering is to get the optimal filter $\hat{\Psi}(X_t)$ of the process of interest $\Psi(X_t)$ based on the observation data $Y_t$ up until time instant $t$. Specifically, we would like

- the optimal filter $\hat{\Psi}(X_t)$ such that

$$
\mathbb{E} \left[ \hat{\Psi}(X_t) \mid \mathcal{B}(Y_0^t) \right] = \arg \min \left\{ \mathbb{E} \left[ |\Psi(X_t) - \eta_t|^2 \right] \mid \eta_t \text{ is } \mathcal{B}(Y_0^t) \text{ measurable} \right\}
$$

where $\mathcal{B}(Y_0^t)$ is the $\sigma$-field induced by the stochastic process $Y_s, 0 \leq s \leq t$.

- We are interested in the statistical information about $\hat{\Psi}(X_t)$, such as
  - high-order moments of the filter $\hat{\Psi}(X_t)$
  - the confidence interval of the filter $\hat{\Psi}(X_t)$
  - the \textbf{conditional} probability density function (PDF) $p(X_t \mid Y_{0:t})$ of $\hat{\Psi}(X_t)$
An example: bearing-only tracking

The state equation
\[
\begin{cases}
    dX_t = 20dt + 20dW_t \\
    X_0 = 3000
\end{cases}
\]

The observation equation
\[
dY_t = \arctan \left( \frac{5000}{X_t} \right) dt + dB_t
\]

Nonlinear filtering problem:
\[
\mathbb{E} \left[ X_t \bigg| \sigma \left\{ Y_s, 0 < s < t \right\} \right] = \arg \min \left\{ \mathbb{E} \left[ |X_t - \eta_t|^2 \right] \bigg| \eta_t \text{ is } Y_{0:t} \text{ measurable} \right\}
\]
The basic idea of the extended Kalman filter method is to linearize the nonlinear functions $h_k$, $g$ and follow the Kalman filter method. The state transition and observation matrices are the Jacobians of $h_k$ and $g$, i.e.

$$H_k = \frac{\partial h}{\partial x}(X_{k|k}) \quad \text{and} \quad G_{k+1} = \frac{\partial g}{\partial x}(X_{k+1|k})$$

The basic idea of the particle filter method is to approximate the probability distribution on supporting points (particles) with corresponding weights.

- The approximate probability distribution at time step $k$ is given by a discrete probability measure supported on the points $x^k_1, \ldots x^k_L$ and weights $p^k_1, \ldots, p^k_L$.

- The probability measure from time $k$ to $k+1$ on updated supporting points $x^{k+1}_1, \ldots x^{k+1}_L$ is updated by incoorparting new data $Z(k+1)$, i.e.

$$(x^k_1, p^k_1), \ldots, (x^k_L, p^k_L) \rightarrow (x^{k+1}_1, p^{k+1}_1), \ldots, (x^{k+1}_L, p^{k+1}_L)$$
Under some regularity conditions, the optimal filter \( \widehat{\Psi}(X_t) \) based on the observation is given by the formula:

\[
\widehat{\Psi}(X_t) = \frac{\int_{\mathbb{R}^d} \Psi(x)p(t, x)dx}{\int_{\mathbb{R}^d} p(t, x)dx},
\]

where \( p(t, x) \) is \textit{unnormalized filtering density}.

The PDF \( p(t, x) \) satisfies the following SPDE, i.e. the Zakai equation

\[
p(t, x) = \mathcal{L}^* p(t, x)dt + g(x)p(t, x)dY_t
\]

\[
\mathcal{L} = \frac{1}{2} \sum_{i,j=1}^{d} (\sigma \sigma^*)_{ij} \frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{d} b_i \frac{\partial}{\partial x_i}
\]

where \( b, \sigma \) and \( g \) are functions in the continuous nonlinear filtering problem. \( Y_t \) is a Wiener process under proper probability measure.

The goal of Zakai equation is to approximate the unnormalized density \( p(t, x) \).
Comments on current methods and our ideas

**Bayesian filters:**
- Large ensemble size is required to accurately approximate the PDF $p(t, x)$, especially the tail behaviors.

**Zakai filters:**
- Exponential growth of the number of grid points to solve the Zakai equation, i.e. the *curse of dimensionality*.
- Lack of knowledge about the *high-probability region* of the desired PDF $p(t, x)$.

**Our ideas:**
- The sparse-grid method is used to approximate the target PDF, so as to alleviate the *curse of dimensionality*.
- Importance sampling are incorporated to adaptively estimate the *high-probability region* at each time step.
1. Background of numerical methods for nonlinear filtering

2. Global sparse-grid polynomial approximation

3. Piecewise hierarchical sparse-grid polynomial approximation

4. Application of sparse-grid approximation for solving Zakai equations

5. Numerical examples
Global polynomial approximation

Let $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_N)$, $\Psi_\boldsymbol{\nu}$ be \textbf{pre-determined} multivariate polynomials in $\text{span} \left\{ \prod_{i=1}^{N} x_i^{\mu_i}, \text{with } \mu_i \leq \nu_i \ \forall i \right\}$ and $\Lambda \subset \mathbb{N}^N$ denote an \textbf{index set}, approximate the target function $u$ by:

$$u_\Lambda(\mathbf{x}) = \sum_{\nu \in \Lambda} c_\nu \Psi_\nu(\mathbf{x}).$$

1. **Taylor approximation:**
   - $\Psi_\nu(\mathbf{x}) = x^\nu$, $c_\nu = \frac{1}{\nu!} \partial^\nu u(0)$, computed recursively.
   - Not suitable for parallel computing.

2. **Least squares projection:**
   - $\{\Psi_\nu\}$ is a multivariate \textbf{orthonormal polynomial basis} in $\mathbf{x}$, e.g., Legendre polynomials, Hermite polynomials.
   - Require more samples than the degree of freedom.

3. **Interpolation:**
   - $\{\Psi_\nu\}$ is a \textbf{Lagrange interpolatory polynomial basis}.
   - $u_\Lambda$ is the \textbf{interpolant} of $u$ over the associated grid.
   - May be unstable if the interpolation nodes are poorly chosen.
The efficiency of polynomial approximation methods heavily depends on the selection of the index sets $\Lambda$.

**Standard approach:** imposing index sets $\Lambda$ a priori. The cardinality of the polynomial space $\mathbb{P}_\Lambda(\Gamma)$ can grow quickly with respect to the dimension $N$.

Some most common choices of $\Lambda$

- Tensor product (TP): $\max_n \alpha_n \nu_n \leq p$ (Intractable for large $N$),
- Total degree (TD): $\sum_{n=1}^N \alpha_n \nu_n \leq p$,
- Hyperbolic cross (HC): $\prod_{n=1}^N (\nu_n + 1)^{\alpha_n} \leq p + 1$,
- Smolyak method (SM): $\sum_{n=1}^N \alpha_n f(\nu_n) \leq f(p)$ with $f(p) = \begin{cases} 0, & p = 0 \\ 1, & p = 1 \\ \lceil \log_2(p) \rceil, & p \geq 2 \end{cases}$
The efficiency of polynomial approximation methods heavily depends on the selection of the index sets $\Lambda$.

**Standard approach:** imposing index sets $\Lambda$ a priori. The cardinality of the polynomial space $\mathbb{P}_\Lambda(\Gamma)$ can **grow** quickly with respect to the dimension $N$.

Some most common choices of $\Lambda$

- **Tensor product (TP):** $\max_n \alpha_n \nu_n \leq p$ (Intractable for large $N$),
- **Total degree (TD):** $\sum_{n=1}^N \alpha_n \nu_n \leq p$,
- **Hyperbolic cross (HC):** $\prod_{n=1}^N (\nu_n + 1)^{\alpha_n} \leq p + 1$,
- **Smolyak method (SM):** $\sum_{n=1}^N \alpha_n f(\nu_n) \leq f(p)$ with $f(p) = \begin{cases} 0, & p = 0 \\ 1, & p = 1 \\ \lfloor \log_2(p) \rfloor, & p \geq 2 \end{cases}$
Example: $N = 2$ with monomial basis
TD space vs. TP space

### 4th order accurate TD space compared with the TP space:

<table>
<thead>
<tr>
<th>$p_1 + p_2 \leq 0$</th>
<th>$p_1 + p_2 \leq 1$</th>
<th>$p_1 + p_2 \leq 2$</th>
<th>$p_1 + p_2 \leq 3$</th>
<th>$p_1 + p_2 \leq 4$</th>
<th>$\max(p_1, p_2) \leq 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\begin{aligned} &amp; y_1^4 \ &amp; y_1^4 y_2 \ &amp; y_1^4 y_2 \ &amp; y_1^4 y_2 \end{aligned}$</td>
<td>$\begin{aligned} &amp; y_1^3 \ &amp; y_1^3 y_2 \ &amp; y_1^3 y_2 \ &amp; y_1^3 y_2 \end{aligned}$</td>
<td>$\begin{aligned} &amp; y_1^2 \ &amp; y_1^2 y_2 \ &amp; y_1^2 y_2 \ &amp; y_1^2 y_2 \end{aligned}$</td>
<td>$\begin{aligned} &amp; y_1 \ &amp; y_1 y_2 \ &amp; y_1 y_2 \ &amp; y_1 y_2 \end{aligned}$</td>
<td>$\begin{aligned} &amp; 1 \ &amp; y_2 \ &amp; y_2 \ &amp; y_2 \end{aligned}$</td>
<td>$\begin{aligned} &amp; y_4 \ &amp; y_4 y_2 \ &amp; y_4 y_2 \ &amp; y_4 y_2 \end{aligned}$</td>
</tr>
</tbody>
</table>

Monomials up to 4th degree. Those below the line are the **useless monomials** we capture (using tensor products) and **are not needed** (and not possible) in higher dimensions - they don’t add the asymptotic accuracy and the cost increases exponential as the dimensions increase.

Recall $M = \dim [P_{\mathcal{J}(p)}(\Gamma)] \implies M_{TD} = \frac{(N+p)!}{N!p!} < M_{TP} = (p+1)^N$
Example: \( N = 2 \) with monomial basis

4th order accurate TD space compared with the TP space:

<table>
<thead>
<tr>
<th>( p_1 + p_2 )</th>
<th>1</th>
<th>( y_1 )</th>
<th>( y_2 )</th>
<th>( y_1^2 )</th>
<th>( y_1 y_2 )</th>
<th>( y_2^2 )</th>
<th>( y_1 y_2^2 )</th>
<th>( y_2^3 )</th>
<th>( y_1 y_2 y_2^2 )</th>
<th>( y_2^4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 + p_2 \leq 0 )</td>
<td>( y_1^4 )</td>
<td>( y_1^3 y_2 )</td>
<td>( y_1^2 y_2^2 )</td>
<td>( y_1 y_2^3 )</td>
<td>( y_2^4 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_1 + p_2 \leq 1 )</td>
<td>( y_1^3 y_2 )</td>
<td>( y_1^2 y_2^2 )</td>
<td>( y_1 y_2^3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_1 + p_2 \leq 2 )</td>
<td>( y_1^2 y_2^2 )</td>
<td>( y_1 y_2^3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_1 + p_2 \leq 3 )</td>
<td>( y_1 y_2^4 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_1 + p_2 \leq 4 )</td>
<td>( y_1^4 y_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \max(p_1, p_2) \leq 4 \]

Monomials up to 4th degree. Those below the line are the useless monomials we capture (using tensor products) and are not needed (and not possible) in higher dimensions - they don’t add the asymptotic accuracy and the cost increases exponential as the dimensions increase.

Recall \( M = \dim \left[ P_{\mathcal{J}(p)}(\Gamma) \right] \implies M_{TD} = \frac{(N+p)!}{N!p!} \ll M_{TP} = (p+1)^N \).
General basis in $N$ dimensions

Total degree vs. Tensor products

<table>
<thead>
<tr>
<th>$N =$ # RVs, dim(Γ)</th>
<th>$p =$ maximal degree of polynomials</th>
<th>$K =$ total # of probabilistic degrees of freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>using total degree basis</td>
</tr>
<tr>
<td>3</td>
<td>3, 5</td>
<td>20, 56</td>
</tr>
<tr>
<td>5</td>
<td>3, 5</td>
<td>56, 252</td>
</tr>
<tr>
<td>10</td>
<td>3, 5</td>
<td>286, 3,003</td>
</tr>
<tr>
<td>20</td>
<td>3, 5</td>
<td>1,771, 53,130</td>
</tr>
<tr>
<td>100</td>
<td>3, 5</td>
<td>176,851, 96,560,646</td>
</tr>
</tbody>
</table>

- tensor products become computational infeasible in higher dimensions
Recall that $U_n^{m(i_n)}$ be the $i$th level interpolant in the direction $x_n$ using $m(i_n)$ points

$$U_n^{m(i_n)} : C^0(\Gamma_n) \rightarrow \mathcal{P}_{m(i_n)-1}(\Gamma_n), \quad U_n^0[u] = 0 \ \forall u \in C^0(\Gamma_n)$$

The tensor-product interpolation is defined by

$$u_p^{TP}(x) = \bigotimes_{n=1}^{N} U_n^{m(i_n)}[u](x), \quad \max_n \alpha_n p_n \leq p.$$

The $n$th difference operator:

$$\Delta_n^{m(i_n)}[u] = U_n^{m(i_n)}[u] - U_n^{m(i_n-1)}[u].$$

The hierarchical surplus:

$$\Delta_n^{m(i)}[u](x) = \bigotimes_{n=1}^{N} \Delta_n^{m(i_n)}[u](x),$$

where $i = (i_1, \ldots, i_N) \in \mathbb{N}_+^N$ is a multi-index.
○ Recall that $U_{n}^{m(i_n)}$ be the $i$th level interpolant in the direction $x_n$ using $m(i_n)$ points

$$U_{n}^{m(i_n)} : C^0(\Gamma_n) \rightarrow P_{m(i_n)-1}(\Gamma_n), \quad U_{n}^0[u] = 0 \forall u \in C^0(\Gamma_n)$$

○ The tensor-product interpolation is defined by

$$u^{TP}_p(x) = \bigotimes_{n=1}^{N} U_{n}^{m(i_n)}[u](x), \quad \max_{n} \alpha_n p_n \leq p.$$ 

○ The $n$th difference operator:

$$\Delta_{n}^{m(i_n)}[u] = U_{n}^{m(i_n)}[u] - U_{n}^{m(i_n-1)}[u].$$

○ The hierarchical surplus:

$$\Delta_{n}^{m(i_n)}[u](x) = \bigotimes_{n=1}^{N} \Delta_{n}^{m(i_n)}[u](x),$$

where $i = (i_1, \ldots, i_N) \in \mathbb{N}_{+}^{N}$ is a multi-index
Recall that \( U_n^{m(i_n)} \) be the \( i \)th level interpolant in the direction \( x_n \) using \( m(i_n) \) points

\[
U_n^{m(i_n)} : C^0(\Gamma_n) \rightarrow P_{m(i_n)-1}(\Gamma_n), \quad U_n^0[u] = 0 \ \forall u \in C^0(\Gamma_n)
\]

The tensor-product interpolation is defined by

\[
u^\text{TP}_p(\mathbf{x}) = \bigotimes_{n=1}^N U_n^{m(i_n)}[u](\mathbf{x}), \quad \max_n \alpha_n p_n \leq p.
\]

The \( n \)th difference operator:

\[
\Delta_n^{m(i_n)}[u] = U_n^{m(i_n)}[u] - U_n^{m(i_n-1)}[u].
\]

The hierarchical surplus:

\[
\Delta_n^{m(i)}[u](\mathbf{x}) = \bigotimes_{n=1}^N \Delta_n^{m(i_n)}[u](\mathbf{x}),
\]

where \( i = (i_1, \ldots, i_N) \in \mathbb{N}_+^N \) is a multi-index.
Recall that $U_n^{m(i_n)}$ be the $i$th level interpolant in the direction $x_n$ using $m(i_n)$ points

$$U_n^{m(i_n)} : C^0(\Gamma_n) \rightarrow P_{m(i_n)-1}(\Gamma_n), \quad U_n^0[u] = 0 \ \forall u \in C^0(\Gamma_n)$$

The tensor-product interpolation is defined by

$$u_{\alpha p}^{TP}(x) = \bigotimes_{n=1}^N U_n^{m(i_n)}[u](x), \quad \max_n \alpha_n p_n \leq p.$$ 

The $n$th difference operator:

$$\Delta_n^{m(i_n)}[u] = U_n^{m(i_n)}[u] - U_n^{m(i_n-1)}[u].$$

The hierarchical surplus:

$$\Delta_n^{m(i)}[u](x) = \bigotimes_{n=1}^N \Delta_n^{m(i_n)}[u](x),$$

where $i = (i_1, \ldots, i_N) \in \mathbb{N}_+^N$ is a multi-index
Basic idea: linear combination of tensor-product interpolants, with a relatively low number of points (but maintain the asymptotic accuracy)

- The sparse-grid interpolation is defined as

\[
 u_{SG}^p(x) = \sum_{g(i) \leq p} \bigotimes_{n=1}^{N} \Delta_n^{m(i_n)}[u](x) = \sum_{g(i) \leq p} c(i) \bigotimes_{n=1}^{N} \mathcal{U}_n^{m(i_n)}[u](x)
\]

with \( c(i) = \sum_{j \in \{0,1\}^N} (-1)^{|j|_1} \) and \( g : \mathbb{N}^N \to \mathbb{N} \) a strictly increasing function.

- SM is the most widely used approach and corresponds to the original Smolyak construction [Smolyak '63]

\[
 c(i) = (-1)^{p+N-|i|_1} \left( \begin{array}{c} N - 1 \\ p + N - |i|_1 \end{array} \right), \quad \text{with } p - N + 1 \leq g(i) \leq p
\]

\[
 u_{SG}^p(x) = \sum_{p-N+1 \leq g(i) \leq p} c(i) \bigotimes_{n=1}^{N} \mathcal{U}_n^{m(i_n)}[u](x)
\]
**Basic idea:** linear combination of tensor-product interpolants, with a relatively low number of points (but maintain the asymptotic accuracy)

- The sparse-grid interpolation is defined as

\[
    u_p^{SG}(\vec{x}) = \sum_{g(\vec{i}) \leq p} N \prod_{n=1}^{N} \Delta_n^{m(i_n)}[u](\vec{x}) = \sum_{g(\vec{i}) \leq p} c(\vec{i}) \prod_{n=1}^{N} \mathcal{U}_n^{m(i_n)}[u](\vec{x})
\]

with \( c(\vec{i}) = \sum_{\vec{j} \in \{0,1\}^N} (-1)^{|\vec{j}|_1} \) and \( g : \mathbb{N}^N \rightarrow \mathbb{N} \) a strictly increasing function.

- SM is the most widely used approach and corresponds to the original Smolyak construction [Smolyak '63]

\[
    c(\vec{i}) = (-1)^{p+N-|\vec{i}|_1} \left( \frac{N-1}{p+N-|\vec{i}|_1} \right), \quad \text{with } p - N + 1 \leq g(\vec{i}) \leq p
\]

\[
    u_p^{SG}(\vec{x}) = \sum_{p-N+1 \leq g(\vec{i}) \leq p} c(\vec{i}) \prod_{n=1}^{N} \mathcal{U}_n^{m(i_n)}[u](\vec{x})
\]
Basic idea: linear combination of tensor-product interpolants, with a relatively low number of points (but maintain the asymptotic accuracy)

- The sparse-grid interpolation is defined as

\[
    u_p^{SG}(\mathbf{x}) = \sum_{g(\mathbf{i}) \leq p} \bigotimes_{n=1}^{N} \Delta_n^{m(i_n)}[u](\mathbf{x}) = \sum_{g(\mathbf{i}) \leq p} c(\mathbf{i}) \bigotimes_{n=1}^{N} U_n^{m(i_n)}[u](\mathbf{x})
\]

with \( c(\mathbf{i}) = \sum_{j \in \{0,1\}^N} (-1)^{|j|_1} \) and \( g : \mathbb{N}^N \to \mathbb{N} \) a strictly increasing function.

- SM is the most widely used approach and corresponds to the original Smolyak construction [Smolyak '63]

\[
    c(\mathbf{i}) = (-1)^{p+N-|\mathbf{i}|_1} \binom{N-1}{p+N-|\mathbf{i}|_1}, \quad \text{with } p - N + 1 \leq g(\mathbf{i}) \leq p
\]

\[
    u_p^{SG}(\mathbf{x}) = \sum_{p-N+1 \leq g(\mathbf{i}) \leq p} c(\mathbf{i}) \bigotimes_{n=1}^{N} U_n^{m(i_n)}[u](\mathbf{x})
\]
Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $p = 5$

$\Rightarrow |i|_1 \leq p + N = 7$

\[
\mathcal{A}(5, 2)
\]
Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $p = 5$

$\Rightarrow |i|_1 \leq p + N = 7$

$i = (6, 1) \Rightarrow (33 \times 1)$
Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $p = 5$

$\Rightarrow |i|_1 \leq p + N = 7$

$i = (5, 2) \Rightarrow (17 \times 3)$
Generating Smolyak sparse grids: $N = 2$
Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $p = 5$

$\Rightarrow |i_1| \leq p + N = 7$

$i = (4, 3) \Rightarrow (9 \times 5)$
Generating Smolyak sparse grids: $N = 2$
Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $p = 5$

$\Rightarrow |i|_1 \leq p + N = 7$

$i = (3, 4) \Rightarrow (5 \times 9)$
Generating Smolyak sparse grids: $N = 2$
Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $p = 5$

$\Rightarrow |i_1| \leq p + N = 7$

$i = (2, 5) \Rightarrow (3 \times 17)$
Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $p = 5$

$\Rightarrow |i|_1 \leq p + N = 7$

$i = (1, 6) \Rightarrow (1 \times 33)$

Guannan Zhang  http://www.csm.ornl.gov/~gz3
Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $p = 5$

$\Rightarrow |\mathbf{i}|_1 \leq p + N = 7$

$\mathcal{A}(5, 2)$
Recall: convergence of the isotropic FTP: $\varepsilon_{TP}(M) \approx \mathcal{O}\left(M^{-g_{\text{min}}/N}\right)$

Theorem [Nobile-Tempone-Webster, 08a]:

For functions $u \in C^0(\Gamma^N)$, the isotropic sparse grid method satisfies

$$
\varepsilon(M) = \|u - u_p\|_{L^2} \leq C(N) M^{-g_{\text{min}}/\log(2N)},
$$

- An analogous result holds for the Gaussian abscissas
- $C \leq 1$ for highly isotropic problems (sub-exponential convergence)
- Exploits the smoothness of the function (as opposed to MC, QMC, LHS, etc.) while reducing the curse of dimensionality
- $g_{\text{min}} \sim r$ when $u \in \mathcal{W}^{r(N)}$ (bdd mixed derivatives of order $r$)
- $\varepsilon(M) \approx \mathcal{O}\left(M^{-r} (\log M)^{N-1}(r+1)\right)$ is simply an upper upper bound
Recall: convergence of the *isotropic* FTP: $\varepsilon_{TP}(M) \approx O\left(\frac{M^{-g_{\min}}}{N}\right)$

**Theorem [Nobile-Tempone-Webster, 08a]:**

For functions $u \in C^0(\Gamma^N)$, the *isotropic sparse grid* method satisfies

$$\varepsilon(M) = \|u - u_p\|_{L^2} \leq C(N)M^{-\frac{g_{\min}}{\log(2N)}},$$

- An analogous result holds for the Gaussian abscissas
- $C \leq 1$ for highly isotropic problems (sub-exponential convergence)
- Exploits the smoothness of the function (as opposed to MC, QMC, LHS, etc.) while reducing the *curse of dimensionality*
- $g_{\min} \sim r$ when $u \in \mathcal{W}_r(N)$ (bdd mixed derivatives of order $r$)
- $\varepsilon(M) \approx O\left(M^{-r}\left(\log M\right)^{(N-1)(r+1)}\right)$ is simply an upper upper bound

Guannan Zhang  http://www.csm.ornl.gov/~gz3
Let $g(i) = \sum_{n=1}^{N} \alpha_n(i_n - 1)$, $\alpha_n = g_n$ and $m(i) = \begin{cases} 1, & i = 1 \\ 2^{i+1} - 1, & i > 1 \end{cases}$

$N = 2$ anisotropic sparse grid: $g(i) \leq p = 0$
Let \( g(i) = \sum_{n=1}^{N} \alpha_n(i_n - 1) \), \( \alpha_n = g_n \) and \( m(i) = \begin{cases} 1, & i = 1 \\ 2^{i+1} - 1, & i > 1 \end{cases} \)

\[ N = 2 \text{ anisotropic sparse grid: } g(i) \leq p = 1 \]

\[ \alpha_2/\alpha_1 = 1 \quad \alpha_2/\alpha_1 = 1.5 \quad \alpha_2/\alpha_1 = 2 \]
Let $g(i) = \sum_{n=1}^{N} \alpha_n (i_n - 1)$, $\alpha_n = g_n$ and $m(i) = \begin{cases} 1, & i = 1 \\ 2^{i+1} - 1, & i > 1 \end{cases}$

$N = 2$ anisotropic sparse grid: $g(i) \leq p = 2$
Let $g(i) = \sum_{n=1}^{N} \alpha_n (i_n - 1)$, $\alpha_n = g_n$ and $m(i) = \begin{cases} 1, & i = 1 \\ 2^{i+1} - 1, & i > 1 \end{cases}$

$N = 2$ anisotropic sparse grid: $g(i) \leq p = 3$
Let \( g(i) = \sum_{n=1}^{N} \alpha_n(i_n - 1) \), where \( \alpha_n = g_n \) and \( m(i) = \begin{cases} 1, & i = 1 \\ 2^{i+1} - 1, & i > 1 \end{cases} \).

\( N = 2 \) anisotropic sparse grid: \( g(i) \leq p = 4 \)

\( \alpha_2/\alpha_1 = 1 \)

\( \alpha_2/\alpha_1 = 1.5 \)

\( \alpha_2/\alpha_1 = 2 \)
Let $g(i) = \sum_{n=1}^{N} \alpha_n(i_n - 1)$, $\alpha_n = g_n$ and $m(i) = \begin{cases} 1, & i = 1 \\ 2^{i+1} - 1, & i > 1 \end{cases}$

$N = 2$ anisotropic sparse grid: $g(i) \leq p = 5$
Let \( g(i) = \sum_{n=1}^{N} \alpha_n(i_n - 1) \), \( \alpha_n = g_n \) and \( m(i) = \begin{cases} 1, & i = 1 \\ 2^{i+1} - 1, & i > 1 \end{cases} \).

\( N = 2 \) anisotropic sparse grid: \( g(i) \leq p = 6 \)

\( \alpha_2 / \alpha_1 = 1 \)  \hspace{1cm} \alpha_2 / \alpha_1 = 1.5 \)  \hspace{1cm} \alpha_2 / \alpha_1 = 2
Let \( g(i) = \sum_{n=1}^{N} \alpha_n (i_n - 1) \), \( \alpha_n = g_n \) and \( m(i) = \begin{cases} 1, & i = 1 \\ 2^{i+1} - 1, & i > 1 \end{cases} \)

\( N = 2 \) anisotropic sparse grid: \( g(i) \leq p = 7 \)

\[ \frac{\alpha_2}{\alpha_1} = 1 \quad \frac{\alpha_2}{\alpha_1} = 1.5 \quad \frac{\alpha_2}{\alpha_1} = 2 \]
Convergence Comparisons:

\( N = 21, 31, 41, 81, \ldots, 121 \) random variables (A posteriori approach)

\[ N=21, 31, 41, 81, 121 \& \ L=1/16 \]

\[
\begin{align*}
\log_{10}(L^2 \text{ error}) & \quad \log_{10}(\# \text{ points}) \\
\text{slope} = -1 & \quad \text{slope} = -1/2 \\
N = 21 \quad & \\
N = 31 \quad & \\
N = 41 \quad & \\
N = 81 \quad & \\
N = 121 \quad &
\end{align*}
\]
For each dimension $n = 1, \ldots, N$, a sequence of subspaces $V_{i_n}$ is defined by

$$V_{i_n} = \text{span}\{\phi_{j_n}^{i_n}(x_n) \mid 0 \leq j_n \leq 2^{i_n}\},$$

where $\phi_{j_n}^{i_n}(x_n)$ is a nodal piecewise polynomial of order $p$ with support $O(2^{-i_n})$.

In the multi-dimensional case, a sequence of subspaces $\mathcal{V}_{l}^{N}$ is defined by

$$\mathcal{V}_{l}^{N} = \bigcup_{g(i) \leq l} \bigotimes_{n=1}^{N} V_{i_n} = \bigcup_{g(i) \leq l} \text{span}\{\prod_{n=1}^{N} \phi_{j_n}^{i_n}(x_n) \mid 0 \leq j_n \leq 2^{i_n}\},$$

Subspace splitting: due to the nested property of $\{\mathcal{V}_{l}^{N}\}_{l=0}^{\infty}$, we decompose the finer subspaces as a direct sum, i.e.

$$\mathcal{V}_{l}^{N} = \mathcal{V}_{l-1}^{N} \bigoplus \mathcal{W}_{l}^{N}, \quad \text{with} \quad \mathcal{W}_{l}^{N} = \mathcal{V}_{l}^{N} \bigcap_{m=0}^{l-1} \mathcal{V}_{m}^{N}$$

Thus, we have a decomposition of $\mathcal{V}_{L}^{N}$ ($L$ denotes the hierarchical level) as

$$\mathcal{V}_{L}^{N} = \mathcal{V}_{0}^{N} \oplus \mathcal{W}_{1}^{N} \oplus \cdots \oplus \mathcal{W}_{L}^{N} \quad \text{and} \quad \mathcal{V}^{N} = \bigcup_{l=0}^{\infty} \mathcal{V}_{l}^{N}$$
Hierarchical basis for $V_{i_n}$

For $p = 1$, $p = 2$ and $p = 3$, [Bungartz-Griebel 04]

linear hierarchical basis (left); quadratic hierarchical basis (center); cubic hierarchical basis (right); defined on equally space Newton-Cotes nodes
The hierarchical sparse-grid interpolant $F_L(x) \in \mathcal{V}_L^N$ is defined by

$$F_L(x) = \mathcal{I}_L^N(F)(x) = \sum_{l=0}^{L} \sum_{g(i) = l} \sum_{j \in B_i} c_j^i \psi_j^i(x).$$

- $\mathcal{I}_L^N : \mathcal{V}_L^N \rightarrow \mathcal{V}_L^N$
- $\psi_j^i(x) = \prod_{n=1}^{N} \phi_{j_n}^i(x_n)$
- $c_j^i$ is the surplus of $\psi_j^i(x)$
- $B_i$ is the index set of the “block” $i = (i_1, \ldots, i_N)$
- Full tensor-product grid: $g(i) = \max_{n=1,\ldots,N} i_n$
- Isotropic sparse grid: $g(i) = i_1 + \cdots + i_N$
Properties of $\mathcal{V}_L^N$ and the approximation $\mathcal{I}_L^N(\cdot)$

\begin{itemize}
  \item[$P_1)$] **Nested hierarchical subspaces:** $\mathcal{V}_0^N \subset \mathcal{V}_1^N \subset \cdots \subset \mathcal{V}_\infty^N$.
  \item[$P_2)$] **Basis having compact support:** $\text{supp} \left( \prod_{n=1}^{N} \phi_{j_n}^{i_n} \right) = O \left( 2^{-\sum_{n=1}^{N} i_n} \right)$.
  \item[$P_3)$] **Interpolatory basis:** $\{ \phi_{j}^{i} \}$ is an interpolating basis for $V_i$, e.g., the “hat” functions, so that the approximation operator $\mathcal{I}_L^N$ is a multi-dimensional interpolation operator.
  \item[$P_4)$] **Decay of the coefficients for smooth functions in $L^2(\Gamma)$:** there exists a constant $C$, independent of the level $L$, such that for every $F(x) \in L^2(\Gamma)$ the following holds:

\[
\sum_{l=0}^{L} \sum_{|i|=l} \sum_{j \in B_i} \left| c_{i,j}^{i} \right|^2 2^{2l} \leq CL \|F(x)\|_{L^2(\Gamma)}^2.
\]
  \item[$P_5)$] **Convergence rate** [Bungartz, Griebel 2004]:

\[
\mathcal{O} \left( M^{-(p+1)} (\log_2 M)^{3(N-1)} \right)
\]

where $p$ the is order of the used polynomial basis functions.
Compute the hierarchical surplus $c_j^i$

The local hierarchical sparse-grid approximation can be written as the following recursive form:

$$\mathcal{I}_L^N(F)(x) = \mathcal{I}_{L-1}^N(F)(x) + \Delta \mathcal{I}_L^N(F)(x)$$

- $\mathcal{I}_{L-1}^N(F)$ is the sparse-grid approximation in $\mathcal{V}_{L-1}^N$ and $\Delta \mathcal{I}_L^N(F)$ is the hierarchical difference interpolant corresponding to $\mathcal{W}_L^N$.

- Due to the nested property $P_1$, the set of grid points used by $\Delta \mathcal{I}_L^N(F)$ can be denoted by $\Delta \mathcal{H}_L^N = \mathcal{H}_L^N \setminus \mathcal{H}_{L-1}^N$.

Due to the interpolatory property $P_3$ we have

$$c_j^i = \mathcal{I}_L^N(F)(x_j^i) - \mathcal{I}_{L-1}^N(F)(x_j^i) = F(x_j^i) - \mathcal{I}_{L-1}^N(F)(x_j^i)$$

- Using the recursive formula, we can compute all the coefficients $c_j^i$ by calculating the coefficients of $\Delta \mathcal{I}_L^N(F')$ for $l = 1, \ldots, L$. 
Redefine the hierarchical SG approximation $F_L^\alpha(y) \in \mathcal{V}_L^N$ as:

$$F_L^\alpha(x) \equiv \mathcal{I}_L^{N,\alpha}(F)(x) = \sum_{l=0}^{L} \sum_{|i|=l} \sum_{j \in B_i^\alpha} c_j^i \psi_j^i(x)$$

with $B_i^\alpha = \{ j \in B_i \mid |c_j^i| > \alpha \}$

The level $L$ adaptive SG interpolant $\mathcal{I}_L^{N,\alpha}(F)$ retains only the terms of the isotropic SG interpolant $\mathcal{I}_L^{N}(F)$ for which the magnitudes of the corresponding surpluses are larger than $\alpha$

The corresponding adaptive sparse grid can be represented by

$$\mathcal{H}_L^{N,\alpha} = \left\{ x_j^i \mid |i| \leq L \text{ and } j \in B_i^\alpha \right\},$$

which is a sub-grid of the level $L$ isotropic sparse grid $\mathcal{H}_L^N$
1-dimensional example
Level 6 adaptive linear sparse grid with $\alpha = 0.01$

The resulting adaptive grid has 21 points (black points) whereas the full grid has 65 points (black and gray points)
With adaptivity, each point that corresponds to a large surplus, e.g., the points in red, blue, or green, lead to 2 children points added in each direction resulting in the adaptive sparse grid.
An example of adaptive sparse-grid approximation with linear, quadratic and cubic bases and $\alpha = 10^{-3}$.
**Disadvantage of hierarchical polynomials:** There is no lower error estimates of the approximation with constants independent of the scaling levels, so typical hierarchical bases cannot achieve an optimal approximation, i.e., achieve a prescribed error tolerance with minimal number of grid points.

**Optimal approximations require an additional property of the basis:**

\[ P_5 \] **Riesz basis:** there exists a constant \( C_R > 0 \) independent of the level \( L \), such that for all \( I^N_L(u) \) the following holds:

\[
C_R^{-1} \sum_{l=0}^{L} \sum_{|\mathbf{i}|=l} \sum_{j \in B_i} |c_{ij}|^2 \leq \left\| I^N_L(u) \right\|^2_{V^N_L} \leq C_R \sum_{l=0}^{L} \sum_{|\mathbf{i}|=l} \sum_{j \in B_i} |c_{ij}|^2,
\]

where the set of multi-index \( B_i \) is defined as above.
Optimal approximations require more...
Additional requirements for the basis of $\mathcal{V}_L^N$

**Disadvantage of hierarchical polynomials:** There is no **lower** error estimates of the approximation with constants independent of the scaling levels, so typical hierarchical bases cannot achieve an **optimal** approximation, i.e., achieve a prescribed error tolerance with minimal number of grid points.

Optimal approximations require an additional property of the basis:

**($P_5$) Riesz basis:** there exists a constant $C_R > 0$ independent of the level $L$, such that for all $\mathcal{I}_L^N(u)$ the following holds:

$$C_R^{-1} \sum_{l=0}^{L} \sum_{|i|=l} \sum_{j \in B_i} |c^i_j|^2 \leq \|\mathcal{I}_L^N(u)\|_{\mathcal{V}_L^N}^2 \leq C_R \sum_{l=0}^{L} \sum_{|i|=l} \sum_{j \in B_i} |c^i_j|^2,$$

where the set of multi-index $B_i$ is defined as above.
- Standard Riesz basis, e.g. Fourier and orthogonal polynomials, consist of functions that are **globally** supported
- However, certain classes of hierarchical wavelet and pre-wavelet bases are not only Riesz bases but consist of compactly supported basis functions.
- We use one particular class of second-generation wavelets [Sweldens 1996, Sweldens 1998, Sweldens, et al. 2000], namely lifted interpolating wavelets on the bounded interval, that satisfies all desired properties.

\[
\psi_l^n = \phi_l^n - \frac{1}{4}\phi_{l-1}^{n-1} + \frac{1}{2}\phi_{l-1}^{n+1}
\]
\[(3.13)\]

where the three equations are used to define central wavelet, left-boundary wavelet and right-boundary wavelet, respectively. For illustration, we plot these three lifting wavelets in Figure 3.1: Left-boundary wavelet (left), central wavelet (middle), right-boundary wavelet (right)

Left-boundary wavelet, central wavelet and right-boundary wavelet
Consider, on $[-1, 1]^2$, the function $f_1(y_1, y_2)$

$$f_1(y_1, y_2) = \begin{cases} 
\exp(-2(y_1^2 + y_2^2)) & \text{if } y_1^2 + y_2^2 \geq 0.25 \\
2\exp(-\frac{1}{2}) - \exp(-2(y_1^2 + y_2^2)) & \text{if } y_1^2 + y_2^2 < 0.25 
\end{cases}$$

that has jumps in its first derivatives across the circle $y_1^2 + y_2^2 = 0.25$.

(a) $f(y_1, y_2)$ (b) the points used by the sg-AWSCM for a tolerance 0.001 (c) error decay vs. number of points (d) error decay vs. the tolerance

The linear increase in the condition number of the interpolation vs. the number of grid points for AHSC with wavelets
1. Background of numerical methods for nonlinear filtering
2. Global sparse-grid polynomial approximation
3. Piecewise hierarchical sparse-grid polynomial approximation
4. Application of sparse-grid approximation for solving Zakai equations
5. Numerical examples
In practice, the desired conditional PDF is closed to bell-shaped curves or surfaces, if it is not Gaussian. In this case, the sparse-grid interpolation may lead to large errors. This is especially true for high-dimensional problems.

The PDF $p(t, x)$ is a standard $d$-dimensional Gaussian density with mean $(0, \ldots, 0)$ and covariance matrix $\sigma^2 I$ with $\sigma = 0.15$
Interpolation error decay

(Left) error decay using linear hierarchical basis; (Right) error decay using Chebyshev basis

Guannan Zhang  http://www.csm.ornl.gov/~gz3
Interpolants of the PDF using linear hierarchical basis

Hierarchical interpolant with 13 points

Hierarchical interpolant with 29 points

Hierarchical interpolant with 65 points

Hierarchical interpolant with 145 points

Guannan Zhang  http://www.csm.ornl.gov/~gz3
Interpolants of the PDF using Chebyshev basis

CC interpolant with 13 points

CC interpolant with 29 points

CC interpolant with 65 points

CC interpolant with 145 points
The interpolation error between the function $F(x)$ and the regular sparse grid approximation $F_L(x)$ is

$$\|F - F_L\|_{L^2} \leq C \cdot 2^{-2L} \cdot A(d, L), \text{ with } A(d, L) = \frac{L^{d-1}}{(d-1)!} + O(L^{d-1})$$

In the case that $F(x)$ is a multi-variate Gaussian distribution with mean $\mu$ and covariance matrix $\sigma^2 I$, the constant $C$ is (see [Pflueger-Peherstorfer-Bungartz, 10] for detailed derivation)

$$C = \frac{2}{\sigma^{4d} \cdot 12^d} \left( \max\{|x_i - \mu|\}^{2d} \right) \|F\|_{L^2}.$$

We introduce the following logarithmic interpolation:

**Step 1:** take the logarithm of $F$: $V = \log(F)$

**Step 2:** approximate $V$ by the sparse-grid interpolation $V_L$

**Step 3:** obtain the approximation of $F$ by $\hat{F}_L = e^{V_L}$. 
(Left) error decay of the interpolant for $F(x)$; (Right) error decay of the exponential of the interpolant for $\log(F(x))$
Interpolants of the log-PDF using linear hierarchical basis

Hierarchical interpolant with 13 points

Hierarchical interpolant with 29 points

Hierarchical interpolant with 65 points

Hierarchical interpolant with 145 points

Guannan Zhang  http://www.csm.ornl.gov/~gz3
Interpolants of the log-PDF using Chebyshev basis

CC interpolant with 13 points

CC interpolant with 29 points

CC interpolant with 65 points

CC interpolant with 145 points
Algorithm for adaptive domain detection

- At time step $t_n$, draw $M$ samples from $p_n(x)$, i.e. $\{p_n^m\}_{m=1}^M \sim p_n$, using importance sampling
- Propagate these samples from $t_n$ to $t_{n+1}$ using a Bayesian filtering approach:

$$\{p_n^m\}_{m=1}^M \rightarrow \{p_{n+1}^m\}_{m=1}^M$$

- Define $\mathcal{D}_{n+\frac{1}{2}} = [a_{n+\frac{1}{2}}, b_{n+\frac{1}{2}}]$ to be the smallest box containing $\{p_{n+1}^m\}_{m=1}^M$
- Define $a_{n+1} = a_{n+\frac{1}{2}} - \lambda \Sigma$ and $b_{n+1} = b_{n+\frac{1}{2}} + \lambda \Sigma$ where $\Sigma = (\Sigma^1, \ldots, \Sigma^d)$: the vector of marginal standard deviations of $\{p_{n+1}^m\}_{m=1}^M$
- Choose $\mathcal{D}_{n+1} = [a_{n+1}, b_{n+1}]

Remarks:

- It is similar to the prediction step of the particle filter method, but with a much smaller ensemble size.
- The constant $\lambda$ is defined based on the required accuracy of the tail behavior.
We use the upwind finite difference method to approximate differential operator $\mathcal{L}$. For $x \in \mathcal{H}_{n+1}^{L,d}$, we approximate the first order partial derivative with $\mu \in \mathbb{R}^d$ by

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

where $e_i$ is the unit vector in the $i$th coordinate direction and $h_i$ is a properly chosen meshsize in the $i$th coordinate direction.
Approximate second order partial derivatives with \( \alpha \in \mathbb{R}^{d \times d} \) by

\[
\alpha_{i,i} \frac{\partial^2 p_n}{\partial x_i \partial x_i}(x) \approx \alpha_{i,i} \tilde{D}^2_{x_i x_i} p_n(x) \equiv \alpha_{i,j} \frac{\hat{p}_n(x + e_i h_i) - 2p_n(x) + \hat{p}_n(x - e_i h_i)}{h_i^2},
\]

\[
\alpha_{i,j} \frac{\partial^2 p_n}{\partial x_i \partial x_j}(x) \approx \alpha_{i,j} \tilde{D}^2_{x_i x_j} p_n(x)
\]

\[
\begin{aligned}
\frac{\alpha_{i,j}}{2h_i} & \left[ \frac{\hat{p}_n(x + e_i h_i + e_j h_j) - \hat{p}_n(x + e_i h_i)}{h_j} \right. \\
& + \frac{p_n(x) - \hat{p}_n(x - e_j h_j)}{h_j} \\
& \left. \right] - \frac{\hat{p}_n(x + e_j h_j) - p_n(x)}{h_j}, \quad \alpha_{i,j} \geq 0,
\end{aligned}
\]

\[
\begin{aligned}
\frac{\alpha_{i,j}}{2h_i} & \left[ \frac{\hat{p}_n(x + e_i h_i) - \hat{p}_n(x + e_i h_i - e_j h_j)}{h_j} \right. \\
& + \frac{p_n(x) - \hat{p}_n(x - e_j h_j)}{h_j} \\
& \left. \right] - \frac{\hat{p}_n(x - e_i h_i + e_j h_j) - \hat{p}_n(x - e_i h_i)}{h_j}, \quad \alpha_{i,j} < 0.
\end{aligned}
\]
Prediction step:

\[ p_{n+\frac{1}{2}}(x) = p_n(x) + \mathcal{L}_n^* p_n(x) \Delta t_n, \quad x \in \mathcal{H}_{n+1}^{L,d} \]

where

\[
\mathcal{L}_n^* p_n(x) = \frac{1}{2} \sum_{i,j}^d \left\{ \frac{\partial^2 (\sigma \sigma^*)_{i,j}}{\partial x_i \partial x_j} p_n(x) + \frac{\partial (\sigma \sigma^*)_{i,j}}{\partial x_i} \tilde{D}_{x_j} p_n(x) \right. \\
+ \frac{\partial (\sigma \sigma^*)_{i,j}}{\partial x_j} \tilde{D}_{x_i} p_n(x) + (\sigma \sigma^*)_{i,j} \tilde{D}^2_{x_i x_j} p_n(x) \bigg\} \\
- \sum_{i=1}^d \left( \frac{\partial b^i}{\partial x_i} p_n(x) + b^i \tilde{D}_{x_i} p_n(x) \right).
\]

Update step:

\[ p_{n+1}(x) = C_n \Psi^n(x, Z_n) p_{n+\frac{1}{2}}(x), \quad x \in \mathcal{H}_{n+1}^{L,d} \]

where \( C_n \) is a normalization factor and function \( \Psi^n \) is defined by

\[
\Psi^n(x, Z_n) = \exp \left\{ -\frac{1}{2\Delta t_n} \cdot |Z_n - h(x)|^2_R \right\}.
\]
Algorithm for the sparse-grid Zakai filter

- Input $p_0$ as the initial value of the solution $p$ of the Zakai equation;
- For $n = 0 \cdots, N_T - 1$,
  - Compute dynamic domain $\mathcal{D}_{n+1}$ for the solution $u_{n+1}$ using importance sampling methods.
  - Generate sparse grid $\mathcal{H}_{n+1}^{L,d}$ on the solution domain $\mathcal{D}_{n+1}$.
  - Evaluate $p_{n+1}$ on the sparse grid $\mathcal{H}_{n+1}^{L,d}$ by using finite difference scheme.
  - Extend the solution $p_{n+1}$ to the whole space $\mathbb{R}^d$ through the logarithmic sparse grid interpolation.
  - Normalization of $p_{n+1}$ such that $\int_{\mathbb{R}^d} p_{n+1}(x) dx = 1$

**Remark:** Time step $\Delta t_n$ has to satisfy the following stability condition

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

1. Background of numerical methods for nonlinear filtering
2. Global sparse-grid polynomial approximation
3. Piecewise hierarchical sparse-grid polynomial approximation
4. Application of sparse-grid approximation for solving Zakai equations
5. Numerical examples
Example 1: a 2D nonlinear filtering problem

The state equation:

\[
\begin{align*}
\begin{pmatrix}
    dX_1^1 \\
    dX_2^1 \\
    dX_1^2 \\
    dX_2^2 \\
X_0^1 \\
X_0^2
\end{pmatrix} &= \begin{pmatrix}
    40 \\
    200 \cdot t^2 \\
    40 \\
    200 \cdot t^2 \\
    0 \\
    0
\end{pmatrix} dt + \frac{1}{2} \begin{pmatrix}
    dW_1^1 \\
    dW_2^2
\end{pmatrix}
\end{align*}
\]

where \( W_t = (W_t^1, W_t^2) \) is a 2 dimensional standard Brownian motion.

The observation equation:

\[
dY_t = \sqrt{(X_t^1 - 20)^2 + (X_t^2)^2} \cdot dt + dB_t,
\]

where \( B_t \) is a 1-dimensional Brownian motion independent of \( W_t \).
Adaptive domain detection

Target trajectory and adaptive solution domain at $n = 1, 20, 50, 70, 80$. 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.
Dynamic solution domains and contours of the PDF

Target state PDF at time step $n = 50, 80$ in the corresponding solution domain
Example 2: 4D bearing-only tracking

- $X_t = (x, y, u, v)$
- $(x, y)$: location
- $(u, v)$: velocity
- $Y_t$: observation angle.
- Input data

\[
X_t = (x, y, u, v)
\]

\[
(x, y): \text{location}
\]

\[
(u, v): \text{velocity}
\]

\[
Y_t: \text{observation angle.}
\]

\[
\begin{align*}
\sigma(X_t) &= \begin{pmatrix}
\sigma_1^2 & 0 & 0 & 0 \\
0 & \sigma_2^2 & 0 & 0 \\
0 & 0 & \sigma_3^2 & 0 \\
0 & 0 & 0 & \sigma_4^2 \\
\end{pmatrix} \\
\end{align*}
\]

\[
g(X_t) = \arctan \left( \frac{y - y_o}{x - x_o} \right).
\]

- Input data

\[
b(X_t) = (u, v, 0, 0);
\]

\[
\begin{align*}
\sigma(X_t) &= \begin{pmatrix}
\sigma_1^2 & 0 & 0 & 0 \\
0 & \sigma_2^2 & 0 & 0 \\
0 & 0 & \sigma_3^2 & 0 \\
0 & 0 & 0 & \sigma_4^2 \\
\end{pmatrix} \\
g(X_t) &= \arctan \left( \frac{y - y_o}{x - x_o} \right).
\end{align*}
\]
Estimate of posterior mean with 95% probability region.
Example 2: 4D bearing-only tracking

approximate PDFs of the location $x$

![Graphs showing marginal densities for different particle numbers and mesh types.](image-url)
Example 2: 4D bearing-only tracking
approximate PDFs of the location $y$

**PF: 4,000 particles**

**PF: 20,000 particles**

**PF: 160,000 particles**

**ZF(Sparse Grid)**

**ZF(Standard Fine Mesh)**

**ZF(Standard Coarse Mesh)**
Comparison of computing time

<table>
<thead>
<tr>
<th>Particle filter</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20,000 particles</td>
<td>554.24</td>
</tr>
<tr>
<td>40,000 particles</td>
<td>1976.39</td>
</tr>
<tr>
<td>80,000 particles</td>
<td>7702.57</td>
</tr>
<tr>
<td>160,000 particles</td>
<td>32380.38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zakai filter method</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse grid</td>
<td>587.14</td>
</tr>
<tr>
<td>Standard</td>
<td>270954.27</td>
</tr>
</tbody>
</table>
Concluding remarks

- For moderately high-dimensional nonlinear filtering problems, sparse-grid approach is an effective technique to reduce the computational complexity of solving the Zakai equation.

- The complexity of our approach is determined by the cardinality of the index set $\Lambda$. Can we extend the classic sparse-grid interpolation to a more general setting by exploiting the regularity of the solution of the Zakai equation?

- **Best $M$-term approximations** [Chkifa, Cohen, DeVore, Schwab ’10,’11,’13,’14]
  Construct the optimal set $\Lambda_M^{\text{opt}}$ of $M$ most effective multi-indices, such that

  $\sup_{y \in \Gamma} \left\| u - \sum_{\nu \in \Lambda_M^{\text{opt}}} c_{\nu} \Psi_{\nu} \right\| \leq \sum_{\nu \notin \Lambda_M^{\text{opt}}} \| c_{\nu} \|.$

- **Quasi-optimal approximations** [Beck-Nobile-Tamellini-Tempone ’12,’14], [Tran-Webster-Z, 15]
  - Construct the quasi-optimal index set $\Lambda_M^{\text{opt}}$ with respect to sharp estimates $B(\nu)$ of $\| c_{\nu} \|_{V}$, rather than $\| c_{\nu} \|_{V}$ themselves.
  - **Advantage**: We do not impose a polynomial space a priori but construct the best index sets and corresponding polynomial subspaces directly.
  - **Challenge**: Sharp bounds can be difficult and somewhat problem dependent.
References


References


- Hoang Tran, Clayton G Webster, and Guannan Zhang. Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients, *submitted*.