The Sparse Grid Combination Technique for Functionals with Applications

Yuancheng Zhou

Nov 2021

A thesis submitted for the degree of Doctor of Philosophy
of the Australian National University

To my parents

Declaration

The work in this thesis is my own except where otherwise stated.

Yuancheng Zhou

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor Prof. Markus Hegland. During my study at the Australian National University, he gave me many valuable suggestions and tremendous help. His broad knowledge and dedication to research inspire me throughout my graduate career.

I would also like to extend my sincere thanks to the other members of my supervisory panel: Prof. Stephen Roberts, Assoc Prof. Qinian Jin and Assoc Prof. Peter Strazdins for their support and guidance.

I am also grateful to Prof. Dirk Pflüger and many others in the University of Stuttgart. During my short visit in Germany, I learned a lot about the sparse grid through discussions with them.

I would also like to thank my friends and colleagues in MSI: Brendan Harding, Matthias Yiu Lam Wong, Chaitanya Kapiolani Shettigara, Lishan Fang, Xifu Sun, Fanzi Meng, Rommel Real, Arunav Kumar, Shilu Feng, Christopher Williams, Abhishek Bhardwaj, Jordan Pitt and many others. Thank all of you for enriching my life and helping me in many ways.

I am also grateful to the financial support from the China Scholarship Council.

Most importantly, I would like to thank my parents for their unconditional support and encouragement throughout my studies.

Abstract

The sparse grid method is a special discretisation technique used to solve high dimensional problems. There are a wide range of applications of the sparse grid method in calculating high dimensional integrals and the solution of high dimensional PDEs. The sparse grid combination technique is a kind of method used to approximate the numerical result of the sparse grid method. The general idea of the sparse grid combination technique is to compute a linear combination of approximations of the solution of the problem. The approximations are computed on some anisotropic regular grids. The combination technique is based on the inclusion-exclusion principle. Compared with the sparse grid method, there are two advantages of the combination technique. First, only nodal basis functions are required in combination technique rather than the hierarchical basis functions in sparse grid method. Second, the combination technique is easier for parallelisation. Generalised combination techniques, e.g. the truncated combination technique, the dimension-adaptive combination technique etc, are developed to further reduce the cost when solving a high dimensional problem.

For many real world problems, people are interested in some functionals related to the solution of the problem rather than the solution itself. These functionals which capture the important features of the problem are usually key for people to further understand it. When a high dimensional problem is considered, the computational cost of the functionals can be large since the numerical solution of a high dimensional partial differential equation is usually expensive to compute. We apply the generalised combination techniques to reducing the cost of computation of important functionals. Our method is based on the error models of the functionals. We build the error models for some special types of functionals when numerical schemes used to compute the PDEs and the functionals are known. We show the connection between the decay of the surpluses and the error models. By using the connection, we can also apply generalised combination techniques to functionals when we only know their computed surpluses.

Numerical experiments are provided to illustrate error models for the functionals and the performance of our generalised combination techniques.

Stochastic optimisation problems minimise expectations of random cost functions. Thus they require accurate quadrature methods in order to evaluate the objective, gradient and Hessian which appear in the computation. Two categories of methods are studied here. One is the discretise then optimise method, the other is the optimise then discretise method. For the methods in the first category, the application of the sparse grid methods leads to high quadrature accuracy in approximating the objective. However, the sparse grid surrogates have negative quadrature weights which potentially destroy the convexity of the objective and thus may lead to totally wrong results. We prove that the sparse grid surrogates maintain the convexity of the objective for sufficiently fine grids. For the methods in the second category, it is more flexible for us to choose the numerical schemes which used to approximate the objective, gradient and Hessian. Therefore, the application of the dimension adaptive method is possible and reasonable for optimise then discretise approaches. It further reduces the computational costs and has even better performance compared with the classical sparse grid method for many stochastic optimisation problems. Applications are provided to demonstrate the superiority of our approaches over the classical Monte Carlo and product rule based approaches.

Notation and terminology

Notation

X	closed set in $\mathbb{R}^n$
Δ	Laplace operator
$L^2(X)$	space of square-integrable function over X
$L^p(X)$	space of p-integrable function over X
$C^k(X)$	set of functions with continuous derivatives up to order k over X
$C^\infty(X)$	set of smooth functions over X
$C_0^k(X)$	subspace of $C^k(X)$ of functions with compact support
$C_0^\infty(X)$	subspace of $C^\infty(X)$ of functions with compact support
$H^m(X)$	Sobolev space of L^2 functions with square-integrable derivatives up to order m
$H^s_{mix}(X)$	Sobolev space of L^2 functions with square-integrable mixed derivatives up to order $s = [s_1, \ldots, s_d]$
$H_0^m(X)$	Subspace of $H^m(X)$ of functions which are zero on the boundary
$H^s_{0,mix}(X)$	Subspace of $H^s_{mix}(X)$ of functions which are zero on the boundary
$\lVert \cdot \rVert_{L^2(X)}$	L^2 norm
$\lVert \cdot \rVert_{L^2}$	L^2 norm

$\Vert\cdot\Vert_2$	L^2 norm
$\Vert\cdot\Vert_{H^m(X)}$	Sobolev norm of order m
$\Vert\cdot\Vert_{H^m}$	Sobolev norm of order m
$\Vert\cdot\Vert_{L^p(X)}$	L^p norm
$\Vert\cdot\Vert_{L^p}$	L^p norm
$\Vert\cdot\Vert_p$	L^p norm
$\vert\cdot\vert_p$	l^p norm
$\vert\alpha\vert$	l^1 norm if α is a vector
$\underline{1}$	vector $(1,\cdots,1)$
$\underline{0}$	vector $(0,\cdots,0)$
F_1^s	Space $C^s[-1,1]$ where $s\in\mathbb{N}$
F_d^s	Set of functions with continuous mixed derivatives up to order $s=[s_1,\cdots,s_d]$
$\Vert\cdot\Vert_d^s$	The norm defined on set F_d^s. $\Vert f\Vert=\max\left\{\Vert D^i f\Vert_\infty,\ i\in\mathbb{N}^d, i_k\leq s\right\}$
$\Vert\cdot\Vert_d^s$	Induced operator norm
$\Vert\cdot\Vert$	Simplified notation of $\Vert\cdot\Vert_d^s$ in Chapter 1
K_γ	Piecewise linear interpolation operator, $K_\gamma : V\to V_\gamma$, $f\mapsto f_\gamma$
I	Integration operator, $If=\int_X f(x)\,dx$
$L_n(f)$	$n+1$ points (Lagrangian) polynomial interpolation operator
f_I^c	Interpolant generated by the sparse grid combination technique with respect to a downset I
Q_I^c	Quadrature operator generated by the sparse grid combination technique with respect to a downset I
f_n^c	Interpolant generated by the level n classical sparse grid combination technique

Q_n^c	Quadrature operator generated by the level n classical sparse grid combination technique
$C_1(x, h_{\gamma_1})$	coefficient (function) in the 2D error splitting model of $f(x) - f_\gamma(x)$
$C_2(x, h_{\gamma_2})$	coefficient (function) in the 2D error splitting model of $f(x) - f_\gamma(x)$
$C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})$	coefficient (function) in the 2D error splitting model of $f(x) - f_\gamma(x)$
$\mathcal{P}(\mathbb{N}^d)$	power set if the set of all multi-indices
$\mathcal{D}(\mathbb{N}^d)$	subset of $\mathcal{P}(\mathbb{N}^d)$ which only contains finite downsets
$L_n(X, Y)$	the space of all continuous multilinear maps from $X \times \cdots \times X$ to Y
$g(\xi)$	amplification factor
$\lVert \cdot \rVert_N$	a new norm defined in Chapter 4
$\lVert \cdot \rVert_F$	Frobenius norm of a matrix, $\lVert A \rVert_F = \sqrt{\operatorname{trace}(AA^*)}$
$\operatorname{supp} \Phi$	The support of a function Φ
$f\vert_{\operatorname{supp} \Phi}$	f on the support of a function Φ and 0 elsewhere

Contents

Chapter 1

An introduction to Sparse Grid

The underlying ideas of the sparse grid method [86] can be dated back to Smolyak when he was studying the quadrature on tensor product function spaces. In 1990, Zenger first formally introduced the sparse grid and applied it to the solution of partial differential equations [91]. He noticed many high frequencies of the solution of a problem resolved on the full grid contribute little to the solution if the problem is sufficiently smooth. Thus, one can greatly reduce the cost of computation by neglecting these high frequencies while only slightly increasing the error of the solution. Griebel and Bungartz extended Zenger's idea and wrote a survey [16] of sparse grids which is a key reference in this area. A conciser introduction [36] of sparse grids is written by Garcke. In the past 30 years, many other mathematicians, computer scientists and engineers worked in this area. They further extended and generalised the original sparse grid concept. Some of their works will be mentioned in the following Chapters. Sparse grid methods are widely used to accelerate the computation of high dimensional partial differential equations [1, 5, 15, 31, 42, 44] and high dimensional integrals [14, 27, 39]. There are also many applications in data science [75, 4, 35, 37]. In this Chapter, we will introduce the classical sparse grid interpolation and quadrature. We will study numbers of grid points which are required in different types of sparse grids. Finally, we will look into the errors of different sparse grid approximations. Compared with previous works in [45], the results in this Chapter are not restricted to the sparse grids which are built based on regular equally spaced grids.

1.1 Sparse Grid Interpolation

We consider multi-indices $i, j \in \mathbb{N}^d$ with the partial ordering $i \leq j$ iff $i_k \leq j_k$ for all $k = 1, \dots, d$. Suppose $X \subset \mathbb{R}^d$, $L^2(X)$ is the space of square-integrable function over X. $L^2(X)$ is a Hilbert space with the scalar product

$$(f, g)_{L^2(X)} := \int_X f(x)g(x)\,dx$$

and the corresponding norm

$$\|f\|_{L^2(X)} := \int_X f^2(x)\,dx.$$

Definition 1.1. $f \in L^2(X)$ has the weak derivative $g = D^i f$ in $L^2(X)$ provided that $g \in L^2(X)$ and

$$(\Psi, g) = (-1)^{|i|}(\partial^i \Psi, f),\ \forall \Psi \in C_0^\infty(X)$$

where $C_0^\infty(X)$ is the subspace of all infinitely differentiable functions which are nonzero only on a compact subset of X.

Suppose $m \in \mathbb{N}$. $H^m(X)$ is the Sobolev space of $L^2(X)$ functions with square-integrable derivatives up to order m. We then introduce the function space $H^s_{mix}(X)$, $s \in \mathbb{N}^d$.

Definition 1.2. Suppose $X \subset \mathbb{R}^d$. A real valued function $f \in L^2(X)$ and $s \in \mathbb{N}^d$ then $f \in H^s_{mix}(X)$ if for each $0 \leq i \leq s$, $i \in \mathbb{N}^d$ the weak derivative $D^i f$ exists and has finite L^2 norm. The function space is equipped with the norm

$$\|f\|^2_{H^s_{mix}(X)} := \sum_{0 \leq i \leq s} \left\| \frac{\partial^{|i|} f}{\partial x^i} \right\|^2_{L^2(X)} = \sum_{0 \leq i \leq s} \|D^i f\|^2_{L^2(X)}.$$

In particular, if $s = [s_1.s_2, \dots, s_d]$ and $s_i = s_j$, $\forall i, j$, we take $H^{s_1}_{mix}(X)$ as a conciser notation of $H^s_{mix}(X)$. We further denote its subset consisting functions which are zero on the boundary as $H^{s_1}_{0,mix}(X)$. By the definition, we have

$$H^2_{mix}(X) \subset H^2(X) \subset L^2(X).$$

1.1.1 1D Interpolation

We first consider 1D sparse grid interpolation. Then we will generalise it to the multidimensional case.

Suppose we have a 1D real valued function $f : X \to \mathbb{R}$. We further assume f is continuous and bounded. X is a bounded closed interval. In particular, we set our standard interval $X = [0, 1]$. For any other closed interval $X = [a, b]$. We can apply the transformation

$$y = \frac{x - a}{b - a}$$

to $f(x)$, then we get $\tilde{f}(y) = f((b - a)y + a)$ which is a function defined on the standard interval $[0, 1]$.

Next, we consider discretisation of our standard domain $[0, 1]$. In order to build the sparse grid, we need a sequence of grids G_γ, $\gamma \in \mathbb{N}$. On each grid, the standard domain $[0, 1]$ is discretised by grid points

$$0 = x_{\gamma,0} \leq x_{\gamma,1} \leq \cdots \leq x_{\gamma,n_\gamma} = 1$$

where the total number of grid points of G_γ is $n_\gamma + 1$. We also require these grids to be nested, which means

$$G_\alpha \subset G_\beta, \quad \alpha \leq \beta.$$

In particular, it is common to use equally spaced grids. The common choice of the spacing is $2^{-\gamma}$ for grid G_γ. In this case, G_γ, $\gamma \in \mathbb{N}$ is a sequence of nested grids. We will further study the interpolation on unequally spaced grids in the next section.

Notation 1.3. Given domain $X = [0, 1]$. We define a sequence of girds G_γ, $\gamma \in \mathbb{N}$ such that each grid G_γ includes $2^\gamma + 1$ equally spaced points $x_{\gamma,i}$, $0 \leq i \leq 2^\gamma$ where $x_{\gamma,i} = i2^{-\gamma}$. Here γ is called the level of grid.

It is commonplace to use a piecewise polynomial f_γ as an interpolant of the function f. For simplicity, in this section, we will only consider the piecewise linear interpolants. More general cases will be discussed in the next section. The piecewise linear interpolants can be written as a linear combination of linear nodal basis functions.

Definition 1.4. The linear nodal basis function $\Phi_{\gamma,i}$ is defined as

$$\Phi_{\gamma,i}(x) = \begin{cases} 1 - 2^i|x - x_{\gamma,i}|, & x \in [x_{\gamma,i} - 2^{-\gamma}, x_{\gamma,i} + 2^{-\gamma}] \cap [0, 1] \\ 0, & \mathit{otherwise} \end{cases}$$

We have $\Phi_{\gamma,i}(x_{\gamma,i}) = 1$ and $\Phi_{\gamma,i}(x_{\gamma,j}) = 0$ for any $i \neq j$. In addition, the basis function $\Phi_{\gamma,i}$ is linear on the intervals $[x_{\gamma,i-1}, x_{\gamma,i}]$ and $[x_{\gamma,i}, x_{\gamma,i+1}]$. By using these linear nodal basis functions, we have following interpolant.

Definition 1.5. The piecewise linear interpolant f_γ on the grid G_γ is defined as

$$f_\gamma := \mathcal{K}_\gamma f = \sum_{i=0}^{2^\gamma} f(x_{\gamma,i})\Phi_{\gamma,i}(x)$$

where $\mathcal{K}_\gamma$ is denoted as the interpolation operator.

Suppose f is an element of a function space $V \subset H^2_{mix}(X)$. The piecewise linear interpolation f_γ is in a subspace V_γ of V. V_γ is spanned by the linear nodal basis functions, i.e.

$$V_\gamma := \operatorname{span}\left\{\Phi_{\gamma,i},\ i = 0, \dots, 2^\gamma\right\}.$$

Now we start to construct f_γ in an alternative way which can be generalised to construct a sparse grid in the multidimensional case. First, we define the hierarchical basis functions and hierarchical spaces.

Definition 1.6. Given the index sets

$$B_\gamma = \begin{cases} \{1, 3, 5, \dots, 2^\gamma - 3, 2^\gamma - 1\}, & \text{if } \gamma > 0, \\ \{0, 1\}, & \text{if } \gamma = 0. \end{cases}$$

The level γ hierarchical basis functions are the $\Phi_{\gamma,i}$ with $i \in B_\gamma$. The hierarchical space W_γ is defined as the space spanned by all the level γ hierarchical basis functions, which is denoted as

$$W_\gamma := \operatorname{span}\left\{\Phi_{\gamma,i}, i \in B_\gamma\right\}.$$

By using the hierarchical basis, we can write a linear interpolant of f as

$$f'_\gamma = \sum_{\alpha=0}^{\gamma} \sum_{i \in B_\alpha} c_{\alpha,i}\Phi_{\alpha,i}.$$

Here we take the hierarchical coefficients

$$c_{\alpha,i} = \begin{cases} \begin{bmatrix} -\frac{1}{2}, & 1, & \frac{1}{2} \end{bmatrix}_{\alpha,i} f, & \alpha \geq 1 \\ f(x_{0,0}), & \alpha = 0, i = 0 \\ f(x_{0,1}), & \alpha = 0, i = 1 \end{cases}$$

where

$$\begin{bmatrix} -\frac{1}{2}, & 1, & \frac{1}{2} \end{bmatrix}_{\alpha,i} f = f(x_{\alpha,i}) - \frac{f(x_{\alpha-1,\frac{i-1}{2}}) + f(x_{\alpha-1,\frac{i+1}{2}})}{2}.$$

The following Lemma shows that the linear interpolants f'_γ and f_γ are the same.

Lemma 1.7. $f'_\gamma = f_\gamma = \mathcal{K}_\gamma f$

Proof. Since we use linear basis function, the interpolants f'_γ and f_γ are piecewise linear functions. Thus we only need to show on any grid points $x_{\gamma,j}$

$$f'_\gamma(x_{\gamma,j}) = f_\gamma(x_{\gamma,j}).$$

Here we prove this by mathematical induction. First, when $\gamma = 0$, we have

$$f_0(x) = f(x_{0,0})\Phi_{0,0}(x) + f(x_{0,1})\Phi_{0,1}(x)$$

and

$$f'_0(x) = c_{0,0}\Phi_{0,0}(x) + c_{0,1}\Phi_{0,1}(x).$$

By definition, we know $c_{0,0} = f(x_{0,0})$ and $c_{0,1} = f(x_{0,1})$. Thus, $f_0 = f'_0$.

Next we need to show $f'_\gamma = f_\gamma$, when $f'_{\gamma-1} = f_{\gamma-1}$. For any grid point $x_{\gamma,j}$, we have

$$\begin{aligned} f'_\gamma(x_{\gamma,j}) &= \sum_{\alpha=0}^{\gamma-1}\sum_{i\in B_\alpha} c_{\alpha,i}\Phi_{\alpha,i}(x_{\alpha,j}) + c_{\gamma,j}\Phi_{\gamma,j}(x_{\gamma,j}) \\ &= f'_{\gamma-1}(x_{\gamma,j}) + c_{\gamma,j} \\ &= f_{\gamma-1}(x_{\gamma,j}) + c_{\gamma,j} \\ &= \frac{1}{2}(f(x_{\alpha-1,\frac{i-1}{2}}) + f(x_{\alpha-1,\frac{i+1}{2}})) + c_{\gamma,j} \\ &= f(x_{\gamma,j}) = f_\gamma(x_{\gamma,j}). \end{aligned}$$

In the first equality, we use the property that the supports of $\Phi_{\gamma,i}$, $i \in B_\gamma$ are nonoverlapping. The second equality is satisfied because $\Phi_{\gamma,j}(x_{\gamma,j}) = 1$. In the third equality, we use our assumption. The fourth equality is obtained by directly using the definitions of $f_{\gamma-1}$ and $c_{\gamma,j}$. □

The proof of Lemma 1.7 actually implies

$$V_\gamma = V_{\gamma-1} \oplus W_\gamma.$$

Consequently, we can write

$$V_\gamma = \bigoplus_{\alpha=0}^{\gamma} W_\alpha = \mathrm{span}\,\{\Phi_{\alpha,i}, \alpha = 0, \ldots, \gamma \text{ } and \text{ } i \in B_\alpha\}$$

We can also write the hierarchical coefficients into an integral form if the function $f \in H^2(X)$.

Lemma 1.8 ([16],[36]). *Let $f \in H^2(X)$ and $c_{\alpha,i}$ be hierarchical coefficients such that $f_\gamma = \mathcal{K}_\gamma f = \sum_{\alpha=0}^{\gamma} \sum_{i\in B_\alpha} c_{\alpha,i}\Phi_{\alpha,i}$. Then for $\alpha > 0$, one has*

$$c_{\alpha,i} = -\frac{2^{-\alpha}}{2}\int_X \Phi_{\alpha,i}\frac{d^2 f}{dx^2}\, dx$$

Proof. Using integration by parts, one obtains

$$\begin{aligned}\int_X \Phi_{\alpha,i}\frac{d^2 f}{dx^2}\, dx &= \Phi_{\alpha,i}\left.\frac{df}{dx}\right|_{x_{\alpha,i-1}}^{x_{\alpha,i+1}} - \int_{x_{\alpha,i-1}}^{x_{\alpha,i+1}} \frac{d\Phi_{\alpha,i}}{dx}\frac{df}{dx}\, dx\\ &= 0 - \frac{1}{2^{-\alpha}}\int_{x_{\alpha,i-1}}^{x_{\alpha,i}} \frac{df}{dx}\, dx + \frac{1}{2^{-\alpha}}\int_{x_{\alpha,i}}^{x_{\alpha,i+1}} \frac{df}{dx}\, dx\\ &= \frac{1}{2^{-\alpha}}(-(f(x_{\alpha,i}) - f(x_{\alpha,i-1})) + (f(x_{\alpha,i+1}) - f(x_{\alpha,i})))\\ &= -\frac{2}{2^{-\alpha}}c_{\alpha,i}.\end{aligned}$$

□

From Lemma 1.8, as

$$\left|\int_X \Phi_{\alpha,i}\frac{d^2 f}{dx^2}\, dx\right| \leq ||\Phi_\alpha||_2||f||_2,$$

the hierarchical coefficients decrease like $O(2^{-\alpha})$. The level α hierarchical surplus from the hierarchical space W_α is bounded by

$$\left|\left|\sum_{i\in B_\alpha} c_{\alpha,i}\Phi_{\alpha,i}\right|\right|_2 \leq \sum_{i\in B_\alpha} |c_{\alpha,i}|\ ||\Phi_{\alpha,i}||_2$$

where $||\Phi_{\alpha,i}||_2 = \left(\frac{2}{3}\right)^{1/2} 2^{-\alpha/2}$ by direct computation. Combined with the result in Lemma 1.8, the level α hierarchical surplus also decreases exponentially with respect to α. However, the number of the basis functions which span the space W_α increases exponentially as α increases. This means the computational cost increases exponentially while only little improvement is obtained in the accuracy. This insight in the relation between computational cost and accuracy motivates the idea of the sparse grid interpolation in high dimensional case.

1.1.2 Multidimensional Interpolation

Now we consider to build the sparse grid for the dimension $d \geq 2$. Suppose the domain $X \subset \mathbb{R}^d$, where d is the dimension of domain X and it is a integer. In particular, we consider the domain X with the following tensor product structure

$$X = \chi_1 \times \cdots \times \chi_d$$

where $\chi_k = [a_k, b_k]$, $k = 1, \dots, d$ are closed intervals. Suppose f is a multivariate real valued function defined on the domain X. Similar as we did in 1D case, we apply the transformation

$$y_k = \frac{x_k - a_k}{b_k - a_k}, \quad k = 1, \dots, d,$$

then we get a function defined on standard d dimensional unit cube $[0,1]^d$.

We again assume our function f is continuous and thus bounded. In order to build the sparse grid, we first consider the discretisation of our standard domain $[0,1]^d$. We construct the d dimensional full grid point set with multi-index* $\gamma = (\gamma_1, \dots, \gamma_d)$ as the product set of 1D grid points set G_γ, which is

$$G_\gamma = G_{\gamma_1} \times \cdots \times G_{\gamma_d}.$$

All the multi-indices γ form a set $\{\gamma = (\gamma_1, \dots, \gamma_d) \,|\, \gamma_i \in \mathbb{N}\}$. We can define the natural partial ordering on this set, which for any multi-index $\alpha \leq \beta$ iff $\alpha_k \leq \gamma_k$ for all $k = 1, \dots, d$. The set $\{G_\gamma\}$ which contains all the full grids inherits the partial ordering from the multi-index set, which is

$$G_\alpha \subset G_\beta, \quad \alpha \leq \beta.$$

If we further choose the spacing as $2^{-\gamma_k}$ for the kth dimension, following Notation 1.3, we have

$$G_\gamma = \left\{x = (x_1, \dots, x_d) \subset \mathbb{R}^d \,|\, x_k = i_k 2^{-\gamma_k} \ for\ i_k = 0, \dots, 2^{\gamma_k} \ and\ k = 1 \dots, d\right\}.$$

We denote elements in G_γ as

$$x_{\gamma,i} = (x_{\gamma_1,i_1}, \dots, x_{\gamma_d,i_d}) = (i_1 2^{-\gamma_1}, \dots, i_d 2^{-\gamma_d})$$

for any $i_k \in \{0, \dots, 2^{\gamma_k}\}$ for each $k = 1, \dots, d$. Again in the above notation $i = (i_1, \dots, i_d)$ is a multi-index.

The d dimensional nodal basis function is defined as product of 1D nodal basis functions.

Notation 1.9. The d dimensional nodal basis functions are

$$\Phi_{\gamma,i}(x) = \prod_{k=1}^{d} \Phi_{\gamma_k,i_k}(x_k).$$

*Here we still use the same index notation as the notation for 1D case. In this thesis, we will mostly focus on the multidimensional case unless otherwise notified.

The support of $\Phi_{\gamma,i}(x)$ is

$$[x_{\gamma_1,i_1-1}, x_{\gamma_1,i_1+1}] \times \cdots \times [x_{\gamma_d,i_d-1}, x_{\gamma_d,i_d+1}] \cap [0,1]^d.$$

We have $\Phi_{\gamma,i}(x_{\gamma_k,i_k}) = 1$ and $\Phi_{\gamma,i}(x) = 0$ for any other grid points in G_γ.

By using the d dimensional nodal basis functions, we can write the d dimensional piecewise linear interpolant of function f as

$$f_\gamma := K_\gamma f = \sum_{i=0}^{2^\gamma} f(x_{\gamma,i})\Phi_{\gamma,i}(x)$$

where $2^\gamma := (2^{\gamma_1}, \ldots, 2^{\gamma_d})$ and the summation

$$\sum_{i=0}^{2^\gamma} := \sum_{i_d=0}^{2^{\gamma_d}} \cdots \sum_{i_1=0}^{2^{\gamma_1}}. \tag{1.1}$$

We denote the function space which is spanned by the d dimensional nodal basis functions as

$$V_\gamma := \operatorname{span}\left\{\Phi_{\gamma,i}, 0 \le i \le 2^\gamma\right\}.$$

It can be also written as the tensor product of the 1D function spaces spanned by 1D nodal basis functions, i.e.

$$V_\gamma = V_{\gamma_1} \otimes \cdots \otimes V_{\gamma_d}.$$

Next, we define the d dimensional hierarchical basis functions and hierarchical spaces as we did for 1D case.

Notation 1.10. Let $\gamma \in \mathbb{N}^d$, then we define the following multi-index set

$$\begin{aligned} B_\gamma :&= B_{\gamma_1} \times \cdots \times B_{\gamma_d} \\ &= \begin{cases} \{i \mid i_k = 1,3,5,\ldots,2^{\gamma_k}-1\}, & \text{if } \gamma_k > 0, k = 1,\ldots,d \\ \{i \mid i_k = 0,1\}, & \text{if } \gamma_k = 0, k = 1,\ldots,d \end{cases} \end{aligned} \tag{1.2}$$

and denote

$$W_\gamma = \operatorname{span}\left\{\Phi_{\gamma,i}, i \in B_\gamma\right\}$$

as the hierarchical space with respect to the multi-index γ. The d dimensional nodal basis functions which generates the space W_γ are called the hierarchical basis functions.

Like we did for space V_γ, the d dimensional hierarchical space W_γ can also be written into tensor product of 1D hierarchical spaces, i.e.

$$W_\gamma = W_{\gamma_1} \otimes \cdots \otimes W_{\gamma_d}.$$

Actually, we can write the function space V_γ as a direct sum of hierarchical spaces W_α, $0 \leq \alpha \leq \gamma$. We can check this by

$$V_\gamma = \bigotimes_{k=1}^{d} V_{\gamma_k} = \bigotimes_{k=1}^{d} \bigoplus_{\alpha_k=0}^{\gamma_k} W_{\alpha_k} = \bigoplus_{0\leq\alpha\leq\gamma} \bigotimes_{k=1}^{d} W_{\alpha_k} = \bigoplus_{0\leq\alpha\leq\gamma} W_\alpha. \tag{1.3}$$

From the decomposition, we can write the d dimensional piecewise linear interpolant of function f in the following alternative way

$$f_\gamma = \sum_{\alpha\leq\gamma} \sum_{i\in B_\alpha} c_{\alpha,i}\Phi_{\alpha,i}. \tag{1.4}$$

The coefficients $c_{\alpha,i}$ can also be obtained from the tensor product structure, we have

$$c_{\alpha,i} = \left(\prod_{k=1}^{d} H_{\alpha_k,i_k}\right) f$$

where

$$H_{\alpha_k,i_k} := \begin{cases} [-\frac{1}{2} \quad 1 \quad -\frac{1}{2}]_{\alpha_k,i_k}, & if\ \alpha_k > 0, \\ [0 \quad 1 \quad 0]_{\alpha_k,i_k}, & if\ \alpha_k = 0. \end{cases} \tag{1.5}$$

The following Lemma is an extension of Lemma 1.8 in d dimensional case. Again the lemma shows the exponential decay of the coefficients in the interpolation formula 1.4.

Lemma 1.11 ([16],[36]). *Let $X = [0,1]^d$ and $f \in H^2_{mix}(X)$ and $c_{\gamma,i}$ be hierarchical coefficients such that $f_\gamma = K_\gamma f = \sum_{\alpha\leq\gamma}\sum_{i\in B_\alpha} c_{\alpha,i}\Phi_{\alpha,i}$, then for $\gamma \geq \underline{1}$, one has*

$$c_{\alpha,i} = (-1)^d 2^{-|\alpha|-d} \int_X \Phi_{\alpha,i} D^{\underline{2}} f \, dx$$

where $|\cdot|$ is the l_1 norm. Additionally for $\alpha \geq \underline{1}$(and $\alpha \geq \underline{0}$), let k be the number of non zero components of α and $\{m_1, \ldots, m_k\} \subset \{1, \ldots, d\}$ be such that $\alpha_{m_1}, \ldots, \alpha_{m_k} \neq 0$ and $\{m_{k+1}, \ldots, m_d\}$ are the remaining indices, then one has

$$c_{\alpha,i} = (-1)^k 2^{-|\alpha|-k} \int_0^1 \cdots \int_0^1 \Phi_{\alpha,i} \frac{\partial}{\partial^2 x_{m_1}} \cdots \frac{\partial}{\partial^2 x_{m_k}} f\bigg|_{x_{m_{k+1}}=x_{i_{k+1}},\ldots,x_{m_d}=x_{i_d}} dx_{m_1} \ldots dx_{m_k}.$$

Proof. Similar as the proof in Lemma 1.8, by using the integration by parts and noticing the product structure of basis functions, we can achieve the result. □

From the decomposition (1.3), we can write the approximation as

$$f_\gamma = \sum_{0\le\alpha\le\gamma} f_\alpha^h \tag{1.6}$$

where each $f_\alpha^h \in W_\alpha$ and $f_\alpha^h = \sum_{\gamma\in B_\alpha} c_{\alpha,i}\Phi_{\alpha,i}$. Here f_α^h is called the hierarchical surplus. The following Lemma provides a bound to the hierarchical surplus from W_α.

Lemma 1.12 ([16],[36]). *Let $f \in H^2_{mix}$ and f_γ is the piecewise linear approximation of f and f_α^h are hierarchical surpluses. Then for $\alpha \ge \underline{1}$*

$$\|c_{\alpha,i}\Phi_{\alpha,i}\|_2 \le 3^{-d}2^{-2|\alpha|}\| \, D^{\underline{2}}f\big|_{\operatorname{supp}\Phi_{\alpha,i}}\|_2$$

where the notation supp *is the support. The hierarchical surplus from W_α*

$$\|f_\alpha^h\| \le \left(\frac{1}{3}\right)^d 2^{-2|\alpha|}\|D^{\underline{2}}f\|_2.$$

Proof. For any $\alpha \ge 1$, the L^2 norm of the basis function $\Phi_{\alpha,i}$ is

$$\left(\frac{1}{3}\right)^{\frac{d}{2}} 2^{\frac{d}{2}-\frac{|\alpha|}{2}}.$$

By using the result in Lemma 1.11 and the Hölder inequality, we have

$$\begin{aligned}
\|c_{\alpha,i}\Phi_{\alpha,i}\|_2^2 &= |c_{\alpha,i}|^2\|\Phi_{\alpha,i}\|_2^2\\
&= (-1)^{2d}2^{-2|\alpha|-2d}\left(\int_\Omega \Phi_{\alpha,i}D^{\underline{2}}f\,dx\right)^2\left(\frac{1}{3}\right)^d 2^{d-|\alpha|}\\
&= \left(\frac{1}{6}\right)^d 2^{-3|\alpha|}\left(\int_\Omega \Phi_{\alpha,i}D^{\underline{2}}f\,dx\right)^2\\
&\le \left(\frac{1}{6}\right)^d 2^{-3|\alpha|}\|\Phi_{\alpha,i}\|_2^2\| \, D^{\underline{2}}\big|_{\operatorname{supp}\Phi_{\alpha,i}} f\|_2^2\\
&= \left(\frac{1}{9}\right)^d 2^{-4|\alpha|}\| \, D^{\underline{2}}\big|_{\operatorname{supp}\Phi_{\alpha,i}} f\|_2^2.
\end{aligned}$$

For any $i \ne j \in B_\alpha$, we have

$$\operatorname{supp}\Phi_{\alpha,i} \cap \operatorname{supp}\Phi_{\alpha,j} = \emptyset.$$

and

$$\bigcup_{i\in B_\alpha} \operatorname{supp}\Phi_{\alpha,i} = X.$$

Therefore, we can bound the hierarchical surplus by

$$\begin{aligned}\|f_\alpha^h\|_2 = \|\sum_{i\in B_\alpha} c_{\alpha,i}\Phi_{\alpha,i}\|_2 &\le \sum_{i\in B_\alpha} \|c_{\alpha,i}\Phi_{\alpha,i}\|_2 \\ &\le \left(\frac{1}{3}\right)^d 2^{-2|\alpha|} \sum_{i\in B_\alpha} \| \, D^2|_{\operatorname{supp}\Phi_{\alpha,i}} f\| \\ &= \left(\frac{1}{3}\right)^d 2^{-2|\alpha|}\|D^2 f\|_2.\end{aligned}$$

□

From Lemma 1.12, we see that the L_2 norm of the hierarchical surplus decays exponentially as the L_1 norm of multi index(level) α increases. At the same time, in order to compute the level α hierarchical surplus, we have to use $|B_\alpha| = 2^{|\alpha|-d}$ nodal basis functions. The computational cost grows exponentially as the L_1 norm of level α increases. These observations lead us to consider if we can only leave these hierarchical surplus terms with multi index α which has relative small L_1 norm and throw away the rest terms to get a cheaper but still accurate approximation of the original function f. The following classical sparse grid method is one of the most commonly used way to do that. The level n classical sparse grid interpolation formula is

$$f_n^s = \sum_{|\alpha|\le n} f_\alpha^h. \tag{1.7}$$

Its corresponding full grid interpolation is when we take multi index γ as $(n,\dots,n)$ in formula 1.6. The sparse grid interpolant f_n^s is in the following defined classical sparse grid function space

$$V_n^s := \bigoplus_{|\alpha|\le n} W_\alpha.$$

V_n^s is a subset of V_n from the definition. The computing grid we used when we find the sparse grid interpolant f_n^s in the space V_n^s is called sparse grid.

1.2 Sparse Grid Quadrature

As mentioned in the previous section, the sparse grid method has originally been developed to compute high dimensional integrals by Smolyak [86]. In 1998, Gerstner and Griebel reviewed this idea in their paper [40] and extended it by trying several new 1D quadrature rules. The error analysis of general sparse grid quadrature can be found in Novak and Ritter's paper [68]. They also studied Polynomial exactness of the sparse grid quadrature. Holtz [49] gave a thorough review of the sparse grid quadrature in his PhD thesis. In addition, he also tested and compared the performances of different kinds of sparse grid quadrature by many high dimensional applications in Finance and Insurance.

We consider computing the following integral

$$If = \int_X f(x)\,dx. \tag{1.8}$$

$X \subset \mathbb{R}^d$ is again assumed to be tensor product of closed intervals. In most cases, we will take X as the standard d dimensional unit cube $[0,1]^d$. The integrand $f \in H^2_{mix}(X)$ is a multidimensional function.

The product rule is the most common way to construct a computational method for the multidimensional integral (1.8). However, the computational cost will be very expensive if d is large. As the sparse grid method can mitigate the curse of dimensionality in the interpolation of high dimensional functions, it can also be used here to reduce the huge computational cost when we compute high dimensional integrals. Both product rule and sparse grid quadrature are built upon 1D quadrature rules. Many 1D quadrature rules we consider here are of interpolatory type, e.g. Newton Cotes formulas [25] and the Clenshaw Curtis rule [23, 25]. We say a 1D quadrature is of interpolatory type if the quadrature formula can be obtained by first interpolating $f(x)$ by a polynomial and then integrating the interpolating polynomial [25]. In the previous section, we only consider using piecewise linear basis function when we interpolate a function. Here we will generalise it to piecewise (Lagrangian) polynomial basis function of degree n, $\forall n \in \mathbb{N}$ in order to construct more general 1D quadrature rules of interpolatory type. Another class of 1D quadrature rule is the quadrature rule of Gauss type. However, this type of quadrature rule does not have nested structure. Nested 1D quadrature rules are required to construct a sparse grid. Patterson [72, 70] found a way to add new quadrature points to the original 1D Gauss rules to make it nested while still keep high polynomial degree of exactness. The so called Gauss Patterson rule is among one of the best 1D quadrature rules used to construct sparse grid quadrature.

1.2.1 1D Quadrature Rules of Interpolatory Type

Suppose we have a sequence of grids G_γ,$\gamma \in \mathbb{N}$. On grid G_γ, the domain $X = [a, b]$ is spaced by $n_\gamma + 1$ grid points

$$a = x_{\gamma,0} \leq x_{\gamma,1} \leq \cdots \leq x_{\gamma,n_\gamma} = b.$$

Definition 1.13. If $l_{\gamma,j}(x)$, $j = 0, \ldots, n_\gamma$ are polynomials of degree n_γ and satisfy

$$l_{\gamma,j}(x_{\gamma,k}) = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases} \qquad j, k = 0, \ldots, n_\gamma,$$

then $l_{\gamma,j}(x)$, $j = 0, \ldots, n_\gamma$ are the (Lagrangian) polynomial basis functions of degree n_γ associated to the grid G_γ.

In fact, the Lagrangian basis functions can be expressed as

$$l_{\gamma,j}(x) = \prod_{k=0,k\neq j}^{n_\gamma} \frac{(x - x_{\gamma,k})}{(x_{\gamma,j} - x_{\gamma,k})}, \quad j = 0, \ldots, n_\gamma.$$

The $n_\gamma + 1$ points (Lagrangian) polynomial interpolant is

$$L_{n_\gamma}(x) = \sum_{j=0}^{n_\gamma} f(x_{\gamma,j}) l_{\gamma,j}(x).$$

The grids $G_\gamma, \gamma \in \mathbb{N}$ here are not restricted to be equally spaced grids. However, we still need to assume $G_\gamma, \gamma \in \mathbb{N}$ is a sequence of nested grids in order to apply the sparse grid method. A common choice of the number of grid points is $2^{\gamma-1}+1$ for each γ.

A well known result states that the Lagrangian interpolant is unique.

Theorem 1.14 (Uniqueness of Lagrangian Interpolation)**.** *Given any $n + 1$ distinct numbers $x_0, \ldots, x_n$ and any set of numbers $y_0, \ldots, y_n$, there is exactly one polynomial $p_n(x)$ of degree n or less that satisfies the interpolation conditions*

$$p_n(x_i) = y_i, \quad 0 \leq i \leq n.$$

The associated quadrature of interpolatory type is

$$Q_\gamma f = \int_X L_{n_\gamma}(x)\, dx = \sum_{j=0}^{n_\gamma} w_{\gamma,j} f(x_{\gamma,j})$$

where $w_{\gamma,j}$, $j = 0, \ldots, n_\gamma$ are weights and

$$w_{\gamma,j} = \int_X l_{\gamma,j}(x)\, dx \tag{1.9}$$

according to the linearity of integration.

Next we will introduce two classes of quadrature formulas of interpolatory type. The first class is the Newton-Cotes formulas. Newton-Cotes formulas are frequently used quadrature formulas of interpolatory type computed on equally spaced grids. Here we will construct the 2-point trapezoidal rule and the 3-point Simpson's rule using Lagrangian basis functions as examples. Other higher accuracy Newton-Cotes formulas can be similarly derived.

- 2-point Trapezoidal rule on domain $X = [0, 1]$

  Suppose the grid used is $G_0 = \{0, 1\}$. The two basis functions are

  $$l_{0,0}(x) = x - 1, \quad l_{0,1}(x) = x$$

  according to the definition of Lagrangian basis function. They are actually linear basis functions as we used in the previous section. From (1.9), we can compute the weights of two points are

  $$w_{0,0} = w_{0,1} = \frac{1}{2}.$$

  Thus
  $$Q_0 f = \frac{1}{2}(f(0) + f(1)).$$

- 3-point Simpson's rule on domain $X = [0, 1]$

  Suppose the grid used is $G_0 = \{0, 0.5, 1\}$. The three Lagrangian basis functions are
  $$\begin{aligned} l_{0,0}(x) &= 2(x - \frac{1}{2})(x - 1) \\ l_{0,1}(x) &= -4x(x - 1) \\ l_{0,2}(x) &= 2x(x - \frac{1}{2}). \end{aligned}$$

  They are quadratic basis functions and their associated weights are

  $$w_{0,0} = \frac{1}{6}, \quad w_{0,1} = \frac{2}{3}, \quad w_{\gamma,2} = \frac{1}{6}.$$

  Thus
  $$Q_\gamma f = \frac{1}{6}\left(f(0) + 4f(\frac{1}{2}) + f(1)\right).$$

In practice, Newton Cotes formulas are mostly used in its composite form.

Definition 1.15 ([25])**.** A composite rule arises when the interval of integration is subdivided into a number of equal subintervals and a fixed rule of integration is applied to each of the subintervals.

Here we take the composite trapezoidal rule as our example. Suppose we compute on the equally spaced grid G_γ. The number of grid points is $2^\gamma + 1$. As in the previous section, we approximate the function f using the piecewise linear interpolant

$$f_\gamma = \sum_{i=0}^{2^\gamma} f(x_{\gamma,i})\Phi_{\gamma,i}(x).$$

In fact, $\Phi_{\gamma,i}$ can be viewed as combination of two linear transformed Lagrangian basis functions on the neighbouring subintervals. Integrating the interpolant f_γ on space $X = [0,1]$, we will get the following composite trapezoidal rules

$$\begin{aligned}\mathcal{T}_\gamma f :&= \int_X f_\gamma \, dx = \sum_{i=0}^{2^\gamma} f(x_{\gamma,i}) \int_X \Phi_{\gamma,i}(x)\, dx \\ &= \frac{1}{2^\gamma}(\frac{1}{2}f(x_{\gamma,0}) + \sum_{i=1}^{2^\gamma-1} f(x_{\gamma,i}) + \frac{1}{2}f(x_{\gamma,2^\gamma})) \\ &= \frac{1}{2^\gamma}\sum_{i=0}^{2^\gamma}{}'' f(x_{\gamma,i}).\end{aligned}$$

The double dash of the summation means the first and the last terms are to be halved in the computation. Similar as constructing the composite trapezoidal rule, we can also derive higher accuracy composite rules by integrating corresponding higher order piecewise interpolant on domain X.

The composite trapezoidal rule is exact for linear functions. The error of the composite trapezoidal rule is $O(2^{-2\gamma})$ if the integrand has a continuous second order derivative.

Lemma 1.16. *Let $f \in C^2[0,1]$. Then*

$$If - \mathcal{T}_\gamma f = -\frac{1}{12}\left.\frac{d^2 f}{dx^2}\right|_{x=\theta} 2^{-2\gamma}, \quad \textit{for some } \theta \in [0,1].$$

Proof. See in [25]. □

The second class is the Clenshaw-Curtis quadrature which is a quadrature formula of interpolatory type computed on an unequally spaced grid. As we did before, we first construct the interpolating polynomial for the integrand f, then we integrate it over domain X to get the weights in the quadrature formula. The corresponding interpolating polynomial is called the Chebyshev interpolant of f. We take domain X as $[-1, 1]$ for simplicity when we discuss the Clenshaw-Curtis quadrature and the Chebyshev interpolant.

We begin with introducing the following Chebyshev Polynomials

$$T_k(x) = \cos(k \arccos(x)), \quad k = 0, 1, 2 \ldots.$$

They are orthogonal polynomials with respect to the weight function

$$w(x) = (1 - x^2)^{-\frac{1}{2}}.$$

For each k, $T_k(x)$ is a polynomial of degree k.

If we take all these orthogonal polynomials as the basis functions of L_2 space, then any function in L_2 space can be expressed by this basis. We have the following Chebyshev expansion for the integrand f

$$f(x) = \sum_{k=0}^{\infty}{}' c_k T_k.$$

where

$$c_k = \frac{2}{\pi} \int_{-1}^{1} (1 - x^2)^{-\frac{1}{2}} f(x) T_k(x)\, dx. \tag{1.10}$$

As we used the double dash in the trapezoidal rule formula, the single dash here indicates that the first term of the summation is to be halved. We denote the $N + 1$ terms truncated Chebyshev expansion as

$$S_N(x) = \sum_{k=0}^{N}{}' c_k T_k(x). \tag{1.11}$$

If f is continuous in $[-1, 1]$, then we have

$$S_N(x) \to f(x), \quad N \to \infty$$

pointwise.

In the truncated Chebyshev expansion (1.11), the coefficients c_k, $k = 0, 1, \ldots$ are still in integral forms. They can not be computed exactly in most cases. The discrete Chebyshev expansion is an approximation of the truncated Chebyshev expansion. It is obtained by approximating the integrals in (1.10) using trapezoidal rule after applying polar coordinates transformation $x = -\cos\theta$,

$$\int_{-1}^{1} (1 - x^2)^{-\frac{1}{2}} f(x) T_k(x)\, dx = \int_{0}^{\pi} f(-\cos\theta) T_k(-\cos\theta)\, d\theta.$$

Suppose we discretise the interval $[0, \pi]$ by a sequence of equally spaced grid. We denote the γth grid as Θ_γ. It contains the following grid points.

$$0 = \theta_{\gamma,0} \leq \theta_{\gamma,1} \leq \cdots \leq \theta_{\gamma,n_\gamma} = \pi$$

Transforming these grid points back to onto the original domain X, we get the following unequally spaced grid points

$$-1 = x_{\gamma,0} \leq x_{\gamma,1} \leq \cdots \leq x_{\gamma,n_\gamma} = 1.$$

We denote the set

$$G_\gamma = \{x_{\gamma,i},\ i = 0, , \ldots, n_\gamma\}$$

as γth grid. In order to make sure $G_\gamma, \gamma \in \mathbb{N}$ are nested, we require Θ_γ, $\gamma \in \mathbb{N}$ are nested. The common choice of the spacing is $2^{-\gamma}/\pi$ for Θ_γ. The grid points $x_{\gamma,i}$, $i = 0, \ldots, n_\gamma$ are the extreme points of the Chebyshev polynomial. They are called the Chebyshev-Gauss-Lobatto (CGL) points in the literature [60]. Approximating the integral by trapezoidal rule, we get

$$\begin{aligned} b_k =& \frac{2}{\pi} \int_0^\pi f(\cos\theta) T_k(\cos\theta)\, d\theta \\ \approx& \frac{2}{n_\gamma} \sum_{i=0}^{n_\gamma}{}'' f(\cos\theta_{\gamma,i}) T_k(\cos\theta_{\gamma,i}) \\ =& \frac{2}{n_\gamma} \sum_{i=0}^{n_\gamma}{}'' f(x_{\gamma,i}) T_k(x_{\gamma,i}) \end{aligned} \tag{1.12}$$

We take $N = n_r$ in the truncated Chebyshev expansion which means we approximate f by a polynomial of degree less or equal to n_γ. The corresponding discrete Chebyshev expansion is

$$J_\gamma f(x) = \sum_{k=0}^{n_\gamma}{}' b_k T_k(x)$$

Next we will show that the discrete Chebyshev expansion $J_\gamma f$ is the same polynomial as the Lagrangian interpolant $L_{n_\gamma} f$ defined on the unequally spaced grid G_γ.

Lemma 1.17 (discrete orthogonality,[60]). *Let*

$$d_{\gamma,pq} = \sum_{i=0}^{n_\gamma}{}'' T_p(x_{\gamma,i}) T_q(x_{\gamma,i})$$

for any γ. We have

$$\begin{aligned} d_{\gamma,pq} &= 0, && p \neq q, p, q \leq n_\gamma \\ d_{\gamma,pp} &= \frac{1}{2} n_\gamma, && 0 < p < n_\gamma \\ d_{\gamma,00} &= d_{\gamma,n_\gamma n_\gamma} = n_\gamma. \end{aligned}$$

Proof. Direct computation by using the definition of the Chebyshev polynomials and trigonometric identities. See details in [60]. □

Lemma 1.18. *For any grid points $x_{\gamma,i} \in G_\gamma$, we have*

$$f(x_{\gamma,i}) = J_\gamma f(x_{\gamma,i}).$$

Proof.

$$\begin{aligned} J_\gamma f(x_{\gamma,i}) &= \sum_{k=0}^{n_\gamma} \frac{2}{n_\gamma} \left[\sum_{j=0}^{n_\gamma}{}'' f(x_{\gamma,j}) T_k(x_{\gamma,j}) \right] T_k(x_{\gamma,i}) \\ &= \sum_{j=0}^{n_\gamma} \frac{2}{n_\gamma} f(x_{\gamma,j}) \sum_{k=0}^{n_\gamma}{}'' T_k(x_{\gamma,j}) T_k(x_{\gamma,i}) \\ &= \sum_{j=0}^{n_\gamma} \frac{2}{n_\gamma} f(x_{\gamma,j}) \sum_{k=0}^{n_\gamma}{}'' T_j(x_{\gamma,k}) T_i(x_{\gamma,k}) \\ &= f(x_{\gamma,i}). \end{aligned}$$

In the second equality we change the order of summation. The third equality is obtained by using the definition of the Chebyshev polynomial. The fourth equality is achieved by applying the discrete orthogonality. □

Combining the result in Lemma 1.18 and the uniqueness of the Lagrangian interpolant, we have

$$J_\gamma f(x) = L_{n_\gamma} f(x) = \sum_{i=0}^{n_\gamma} f(x_{\gamma,i}) l_{\gamma,i}(x).$$

We can obtain weights of the Clenshaw-Curtis quadrature rule by integrating the Lagrangian basis functions $l_{\gamma,i}(x)$ over the domain $X = [-1, 1]$. However, these integrals are not easy to compute in this case. The following method gives a conciser way to find out the weights of the Clenshaw-Curtis quadrature rule. It was first given in the literature [83] by Sloan and Smith. Instead of integrating the Lagrangian basis functions, they derived the expression of weights from the discrete Chebyshev expansion $J_\gamma f$.

Let

$$a_k = \int_{-1}^{1} T_k(x)\,dx, \quad k = 0, \ldots n_\gamma$$

Computing it by using the polar coordinate, we have

$$a_k = \int_{0}^{\pi} \cos k\theta \sin\theta\, d\theta.$$

Then a_k is straightforward to calculate. Integrating $J_\gamma f$ over the domain $[-1, 1]$, we get

$$\begin{aligned} Q_{n_\gamma} f &= \int_{-1}^{1} J_\gamma f(x)\,dx = \int_{-1}^{1} \sideset{}{'}\sum_{k=0}^{n_\gamma} b_k T_k(x)\,dx \\ &= \sideset{}{'}\sum_{k=0}^{n_\gamma} b_k \int_{-1}^{1} T_k(x)\,dx = \sideset{}{'}\sum_{k=0}^{n_\gamma} b_k a_k. \end{aligned} \tag{1.13}$$

Plug (1.12) into (1.13), we can obtain

$$\begin{aligned} Q_{n_\gamma} f &= \sum_{k=0}^{n_\gamma} a_k \frac{2}{n_\gamma} \sideset{}{''}\sum_{i=0}^{n_\gamma} f(x_{\gamma,i}) T_k(x_{\gamma,i}) \\ &= \sum_{i=0}^{n_\gamma} f(x_{\gamma,i}) \sideset{}{''}\sum_{k=0}^{n_\gamma} \frac{2a_k}{n_\gamma} T_k(x_{\gamma,i}). \end{aligned}$$

Comparing with the general quadrature formula

$$Q_{n_\gamma} f = \sum_{i=0}^{n_\gamma} w_{\gamma,i} f(x_{\gamma,i}),$$

we have

$$w_{\gamma,i} = \sideset{}{''}\sum_{k=0}^{n_\gamma} \frac{2a_k}{n_\gamma} T_k(x_{\gamma,i}).$$

1.2.2 Sparse Grid Quadrature Based on Interpolation

Now, we start to construct the sparse grid quadrature based on these 1D quadrature rules of interpolatory type. First the multidimensional grid G_γ is the Cartesian product of 1D grids. Suppose the dimension is d. Then

$$G_\gamma = G_{\gamma_1} \times \cdots \times G_{\gamma_d}$$

where $\gamma = (\gamma_1, \ldots, \gamma_d)$. The sequences of 1D grid G_{γ_k}, $\gamma_k = 0, 1, \ldots,, k = 1, \ldots, d$ are required to be nested but can be unequally spaced. The d dimensional Lagrangian basis functions are defined as product of 1D Lagrangian basis functions, namely,

$$l_{\gamma,i}(x) = \prod_{k=1}^{d} l_{\gamma_k,i_k}(x_k).$$

The d dimensional Lagrangian interpolant[†] of integrand f defined on grid G_γ is

$$f_\gamma := L_{n_\gamma} f = \sum_{i=0}^{n_\gamma} f(x_{\gamma,i}) l_{\gamma,i}(x). \tag{1.14}$$

Here we follow the notation in (1.1). The product rule can be derived by integrating the interpolant f_γ over space X. If we denote the product rule as the operator Q_γ where $\gamma = (\gamma_1, \dots, \gamma_d)$, then we have

$$Q_\gamma f = \sum_{i=0}^{n_\gamma} w_{\gamma,i} f(x_{\gamma,i})$$

where the weights are

$$\begin{aligned} w_{\gamma,i} &= \prod_{k=1}^{d} w_{\gamma_k,i_k}, \\ w_{\gamma_k,i_k} &= \int_{X_k} l_{\gamma_k,i_k}(x_k)\, dx_k, \quad k = 1, \dots, d. \end{aligned}$$

We still denote the function space which is spanned by the d dimensional Lagrangian basis functions as

$$V_\gamma = \operatorname{span}\left\{ l_{\gamma,i} \,|\, \underline{0} \le i \le n_\gamma \right\}.$$

If we denote V_{γ_k} as the space spanned by 1D Lagrangian basis functions, i.e.

$$V_{\gamma_k} = \operatorname{span}\left\{ l_{\gamma_k,i_k} \,|\, 0 \le i_k \le n_{\gamma_k} \right\}, \quad k = 1, \dots, d$$

and P_{n_k}, $k = 1, \dots, d$ as the set contains all polynomials of degree up to n_{γ_k}, then $V_{\gamma_k} = P_{\gamma_k}$. We can rewrite the space V_γ as

$$V_\gamma = V_{\gamma_1} \otimes \cdots \otimes V_{\gamma_d} = P_{\gamma_1} \otimes \cdots \otimes P_{\gamma_d}.$$

The hierarchical space with respect to the multi-index γ is

$$W_\gamma = \operatorname{span}\left\{ l_{\gamma,i},\ i \in B_\gamma \right\}$$

[†] Piecewise polynomial interpolation is not discussed here. For this case, in order to construct a sparse grid, we require the domain of function to be subdivided into several equal subintervals and a fixed interpolation method to be applied to each of these subintervals. A composite quadrature rule will be derived if we integrate such interpolant over the domain.

where B_γ is the multi-index set defined in Notation 1.2. It can also be written as the tensor product of 1D hierarchical spaces

$$W_\gamma = W_{\gamma_1} \otimes \cdots \otimes W_{\gamma_d}$$

where

$$W_{\gamma_k} = \operatorname{span}\left\{l_{\gamma_k,i_k},\ i_k \in B_{\gamma_k}\right\}.$$

By using the definition of W_{γ_k}, we can decompose the the 1D space V_{γ_k} as the direct sum of W_{α_k}, $0 \le \alpha_k \le \gamma_k$, namely,

$$V_{\gamma_k} = \bigoplus_{\alpha_k=0}^{\gamma_k} W_{\alpha_k}.$$

For the multidimensional case, we have

$$V_\gamma = \bigotimes_{k=1}^{d} V_{\gamma_k} = \bigotimes_{k=1}^{d} \bigoplus_{\alpha_k=0}^{\gamma_k} W_{\alpha_k} = \bigoplus_{0\le\alpha\le\gamma} \bigotimes_{k=1}^{d} W_{\alpha_k} = \bigoplus_{0\le\alpha\le\gamma} W_\alpha.$$

From the decomposition of V_γ, the d dimensional Lagrangian interpolants can also be written into the following expression

$$f_\gamma = \sum_{\alpha\le\gamma} \sum_{i\in B_\alpha} c_{\alpha,i} l_{\alpha,i} \tag{1.15}$$

where the coefficients are

$$c_{\alpha,i} = \left(\prod_{k=1}^{d} H_{\alpha_k,i_k}\right) f.$$

Here H_{α_k,i_k} is the same as we defined in (1.5). This is because by using the uniqueness of the Lagrangian interpolant, we only need to check if the two expressions (1.14) and (1.15) achieve the same value on every grid points for 1D case. This has already been shown in the proof of the Lemma 1.7.

Dropping the terms in (1.15) for which the multi-index α which has large l_1 norm, we get the d dimensional classical sparse grid Lagrangian interpolants

$$f_n^s = \sum_{|\alpha|\le n} \sum_{i\in B_\alpha} c_{\alpha,i} l_{\alpha,i}.$$

The classical sparse grid function space is

$$V_n^s = \bigoplus_{|\alpha|\le n} W_\alpha.$$

We again integrate the sparse grid Lagrangian interpolant over domain X. Then we get the following computing formula of sparse grid quadrature

$$\begin{aligned} Q_n^s f = \int_X f_n^s(x)\, dx &= \int_X \sum_{|\alpha|\le n} \sum_{i\in B_\alpha} c_{\alpha,i} l_{\alpha,i}(x)\, dx. \\ &= \sum_{|\alpha|\le n} \sum_{i\in B_\alpha} c_{\alpha,i} \int_X l_{\alpha,i}(x) dx \\ &= \sum_{|\alpha|\le n} \sum_{i\in B_\alpha} c_{\alpha,i} w_{\alpha,i}. \end{aligned}$$

1.2.3 Integration Rules of Gauss Type

In general, we hope the quadrature formula

$$Qf = \sum_{j=0}^{n} w_j f(x_j)$$

can exactly integrate as many functions as possible. From the Weierstrass Approximation Theorem [78], we know that any real valued continuous function on a closed interval can be approximated by a polynomial. Thus, it is common to use polynomials as the test function class. The following definition of the polynomial degree of exactness is frequently used in the discussion of the performance of a quadrature formula.

Definition 1.19. The polynomial degree of exactness is the largest value of n so that all the polynomials of degree n and below are integrated exactly.

Suppose $L_n(x)$ is the $n+1$ points Lagrangian interpolant. If we further denote the remainder as

$$R_n(x) = f(x) - L_n(x),$$

then we have

Theorem 1.20 ([25]). *If $f^{(n)}(x)$ is continuous on $[a,b]$ and $f^{(n+1)}(x)$ exists in (a,b), then for any $x \in [a,b]$, the remainder*

$$R_n(x) = f(x) - L_n(x) = \frac{f^{(n+1)}(\theta)}{n+1} w_{n+1}(x).$$

Here $\theta \in (a,b)$ and it depends on the choice of x. $w_{n+1}(x)$ is defined as

$$w_{n+1}(x) = (x - x_0)\dots(x - x_n).$$

Proof. See in [25]. □

From this remainder theorem, we can prove the following result

Theorem 1.21. *An $n+1$ points quadrature rule is an integration rule of interpolatory type if and only if its polynomial degree of exactness is at least n.*

Proof. Suppose the $n+1$ points interpolating polynomial of integrand is $L_n(x)$ and the remainder is $R_n(x)$. By using the result in the remainder theorem, we have

$$\int_a^b f(x)\,dx - \int_a^b L_n(x)\,dx = \int_a^b R_n(x)\,dx = 0$$

provided that f is a polynomial of degree less or equal than n. This is because $f^{(n+1)}(x) = 0$ for such polynomial f.

If the polynomial degree of exactness is at least n, then for each Lagrangian basis function $l_k(x)$, $k = 0, \ldots, n$, the following integral

$$\int_a^b l_k(x)\,dx = \sum_{j=0}^{n} w_j l_k(x_j)$$

is computed exactly. Using the definition of the Lagrangian basis function, the right-hand side of the equation is w_k. Thus the quadrature rule is of interpolatory type. □

From the above Theorem, we have the following remarks

Remark 1.22. The 2-point trapezoidal rule is exact for all polynomials of degree less or equal than 1.

Remark 1.23. The 3-point Simpson's rule is exact for all polynomials of degree less or equal than 2.

Remark 1.24. The $2^{\gamma-1}+1$ points Clenshaw-Curtis rule is exact for all polynomials of degree less or equal than $2^{\gamma-1}$.

Next we further look into the polynomial degree of exactness of these quadrature rules. The degree of exactness of the 2-point trapezoidal rule is 1. We can check this by applying it to compute the integral $\int_0^1 x^2\,dx$. The trapezoidal rule gives the value $1/2$ while the exact value is $1/3$. Similarly, we can also check the degree of exactness of a $2^{\gamma-1}+1$ points Clenshaw-Curtis rule is $2^{\gamma-1}$. However, for 3-point Simpson's rule, the exact value of the integral is

$$\int_a^b x^3\,dx = \frac{b^4 - a^4}{4}$$

which is equal to the result obtained from Simpson's rule

$$\frac{b-a}{6}\left[a^3+4\left(\frac{a+b}{2}\right)^3+b^3\right].$$

In addition, it is not exact when we apply it to integrating $f(x)=x^4$. Thus, the polynomial degree of exactness of 3-point Simpson's rule is 3.

We now consider the quadrature rules for the integral with a weight function $\rho(x)$,

$$\int_a^b f(x)\rho(x)\,dx \approx Qf = \sum_{j=0}^{n} w_j f(x_j). \tag{1.16}$$

It is natural to consider when the 1D quadrature rule (1.16) achieves the highest polynomial degree of exactness. How to choose the weights w_j, $j = 0,\dots,n$ and the quadrature points x_j, $j=0,\dots,n$ for such a case?

There are $2n+2$ unknowns in total in the quadrature rule (1.16). Thus it is possible for us to derive a quadrature rule of degree of exactness $2n+1$. In particular, if the quadrature points are given, then there are $n+1$ unknowns in total. If we further assume the quadrature rule is interpolatory, then according to the Theorem 1.21, we know the degree of exactness of such a quadrature rule is at least n.

Definition 1.25. If the degree of exactness of the quadrature rule (1.16) is $2n+1$, then the quadrature points are defined as Gauss points and the quadrature rule is defined as Gauss quadrature.

In order to obtain the Gauss points and its corresponding weights, we take $f(x)=x^m$, $m=0,\dots,2n+1$ and then we get the following system of polynomial equations

$$\sum_{j=0}^{n} w_j x_j^m = \int_a^b x^m \rho(x)\,dx, \quad m=0,\dots,2n+1.$$

Since this system of equations is nonlinear, it is not easy to solve it directly. The commonly used way to tackle this system of equations is to first find the Gauss points x_j, $j=0,\dots,n$ and then solve the linear system with unknowns w_j, $j=0,\dots,n$.

The following theorem provides a way to find the Gauss points. Actually, the Gauss points are the zeros of a polynomial which is orthogonal to any polynomial with degree less than n with respect to the weight function $\rho(x)$.

Theorem 1.26 ([25]). *The quadrature points of an integration rule of interpolatory type is the Gauss points iff the following defined polynomial*

$$w_{n+1}(x) = (x - x_0) \ldots (x - x_n)$$

is orthogonal to any polynomial $p(x)$ *with degree less than* n *with respect to weight function* $\rho(x)$*,i.e.*

$$\int_a^b p(x) w_{n+1}(x) \rho(x)\, dx = 0.$$

Proof. See in [25]. □

It is natural to consider constructing a sparse grid quadrature based on the 1D Gauss rules. Since 1D Gauss rules achieve maximal polynomial degree of exactness, one can expect the new sparse grid quadrature built upon these rules also has high polynomial degree of exactness. However, unfortunately, when we take a sequence of Gauss grids which consist of Gauss points, these grids are not nested. Since nested 1D grids are required to construct a sparse grid, it is impossible to use the sequence of Gauss rules directly as our 1D quadrature rules.

Suppose we have some preassigned quadrature points. Can we build a new quadrature rule such that it has as high as possible polynomial degree of exactness after adding some new quadrature points? If this is possible, we can iterate this procedure to build a sequence of nested 1D grids which can be used in constructing a sparse grid with high polynomial degree of exactness.

Suppose

$$a \le y_1 \le \cdots \le y_{m_1} \le b$$

are the preassigned m_1 quadrature points. We add m_2 new quadrature points

$$a \le x_1 \le \cdots \le x_{m_2} \le b.$$

Then the integral with weight function $\rho(x)$ is approximated by

$$\int_a^b \rho(x) f(x)\, dx \approx \sum_{j=1}^{m_1} w_j^y f(y_j) + \sum_{j=1}^{m_2} w_j^x f(x_j). \tag{1.17}$$

The weights

$$\begin{aligned} &w_j^y, \quad j = 1, \ldots, m_1 \\ &w_j^x, \quad j = 1, \ldots, m_2 \end{aligned}$$

and the newly added quadrature points

$$x_j, \quad j = 1, \ldots, m_2$$

are $2m_2+m_1$ unknowns. Thus it is possible to design the extended quadrature rule such that it is exact for polynomials of the highest possible degree $2m_2+m_1-1$ by choosing these weights and new quadrature points.

The following Theorem is a generalisation of the Theorem 1.26.

Theorem 1.27 ([25]). *The extended quadrature rule* (1.17) *is of degree of exactness* $2m_2+m_1-1$ *if and only if*

(a)it is exact for all polynomials of degree less than $2m_2+m_1-1$

(b)Let the polynomials

$$\begin{aligned} r(x) &= (x-y_1)\dots(x-y_{m_1}),\\ s(x) &= (x-x_1)\dots(x-x_{m_2}). \end{aligned}$$

Their product $r(x)s(x)$ *is orthogonal to any polynomials of degree less than* m_2 *with respect to the weight function* $\rho(x)$*, i,e.*

$$\int_a^b \rho(x)r(x)s(x)p(x)\,dx = 0.$$

Proof. See in [25]. □

Kronrod [55] first extended an n-points Gauss quadrature rule by adding $n+1$ new quadrature points. The polynomial degree of exactness of the new quadrature rule is $3n+1$ if n is even and is $3n+2$ if n is odd. In particular, if we further take the weight function $\rho(x)=1$, $a=-1$ and $b=1$ in Theorem 1.17, then it turns out that the newly added quadrature points are zeros of the Stieltjes polynomial [25] E_{n+1} satisfying

$$\int_{-1}^1 p_n(x)E_{n+1}(x)x^k\,dx = 0, \quad k=0,\dots,n$$

where $p_n(x)$ is the nth Legendre polynomial. The polynomial E_{n+1} can be computed by expanding it in terms of Legendre polynomials and solving the resulting linear system. The roots of the Stieltjes polynomials [25] can be solved by applying Newton's method. The weights can then be computed by solving the corresponding linear system of equations.

Patterson [72] generalised Kronrod's idea by iterating the Kronrod scheme. Finally a sequence of nested grid with maximal degree of exactness is derived. If we start from the 3-point Gauss rule $\mathcal{G}_3$, then the sequence of quadrature rule we obtained is

$$\mathcal{G}_3 = \mathcal{P}_3^3, \mathcal{K}_7 = \mathcal{P}_7^3, \mathcal{P}_{15}^3, \mathcal{P}_{31}^3, \dots$$

where we denote $\mathcal{P}$ as Patterson scheme and $\mathcal{K}$ as Kronrod scheme. The superscript in the Patterson scheme means the starting rule is a 3-point Gauss rule. The γth rule is of $n_\gamma = 2^{\gamma+1} - 1$, $\gamma = 1, 2, \dots$ number of grid points and is of polynomial degree of exactness $3 \times 2^\gamma - 1$ when $\gamma = 1, 2, \dots$.

Remark 1.28. Not all Gauss formulas can be extended. For example, only four possible extensions are possible for the 2-point Gauss rule [71].

Next, we will give a more general way to construct a sparse grid quadrature rule. The construction not only works for the simple interpolatory rules which have already been discussed in the previous section, but also for composite rules and the quadrature rules of non-interpolatory type. In order to do that, we only require function values on a sequence of nested grids. The grid points of each grid in the sequence can be the quadrature points of any simple/composite 1D rule of interpolatory/non-interpolatory type.

We again denote the sequence of grids as G_γ, $\gamma \geq 1$. For each grid, the grid points are

$$a \leq x_{\gamma,0} \leq x_{\gamma,1} \leq \cdots \leq x_{\gamma,n_\gamma} \leq b.$$

Notice here we also allow integration rules of open type of where the end points are omitted from the evaluation. Suppose the weight which is associated with quadrature point $x_{\gamma,i}$, $i = 0, \dots, n_\gamma$ is $w_{\gamma,i}$. The 1D quadrature rule on grid G_γ can be computed as

$$Q_\gamma f = \sum_{i=0}^{n_\gamma} w_{\gamma,i} f(x_{\gamma,i}).$$

Then we can define the following difference operator

$$\Delta_\gamma f := Q_\gamma f - Q_{\gamma-1} f = (Q_\gamma - Q_{\gamma-1}) f,$$

and set

$$Q_0 f := 2f(\frac{a+b}{2}).$$

Then we can rewrite the nth 1D quadrature as the following telescoping sum

$$Q_n f = \sum_{\alpha \leq n} \Delta_\alpha f.$$

Now we consider computing the multi-dimensional integral. Let γ, α to be d dimensional vectors in this case. By applying the product rule to 1D quadrature rules Q_{γ_i}, $i = 1, \dots, d$, the d dimensional integral can be approximated by the following quadrature formula

$$Q_\gamma f = (Q_{\gamma_1} \otimes \cdots \otimes Q_{\gamma_d})f = \sum_{i_1=1}^{n_{\gamma_1}} \cdots \sum_{i_d=1}^{n_{\gamma_d}} w_{\gamma_1,i_1} \dots w_{\gamma_d,i_d} f(x_{\gamma_1,i_1}, \dots, x_{\gamma_d,i_d}).$$

The d-dimensional difference operator Δ_γ is defined as the tensor product of 1D difference operators

$$\Delta_\gamma := (\Delta_{\gamma_1} \otimes \cdots \otimes \Delta_{\gamma_d})f.$$

By using the d-dimensional difference operator, the product rule Q_γ where $\gamma = (n, \dots, n)$ can be written as

$$Q_\gamma f = (Q_n \otimes \cdots \otimes Q_n)f = \sum_{\alpha \leq (n,\dots,n)} \Delta_\alpha f$$

where the summation is defined in (1.1). Then the classical sparse grid quadrature is defined as

$$Q_n f = \sum_{|\alpha| \leq n} \Delta_\alpha f. \tag{1.18}$$

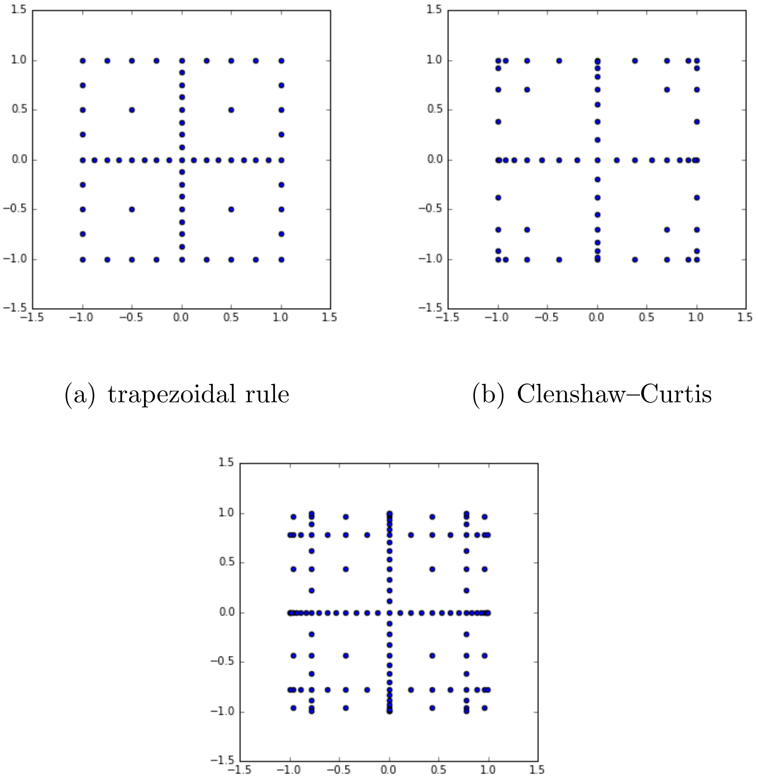
1.3 Number of Grid Points Used in a Sparse Grid

From the previous sections, we know a d dimensional sparse grid is constructed from sequences of 1D grids. In order to find out the number of grid points used in the d dimensional sparse grid, we first review and sum up the number of grid points in different 1D grids we have discussed before.

Case 1. In piecewise linear interpolation and the composite trapezoidal rule, we use the equally spaced 1D grid. It is common to take a grid with $2^\gamma + 1$, $\gamma = 1, 2, \dots$ points as the γth grid in the sequence. If the function f is zero on the boundary , the number of grid points is reduced to $2^\gamma - 1$, $\gamma = 1, 2, \dots$.

Case 2. In the simple Chebyshev interpolation and Clenshaw Curtis rule, a set which contains Chebyshev-Gauss-Lobatto(CGL) points forms a grid. Although the grid points are not equally spaced, we can map these CGL points one to one onto an equally spaced grid. Therefore, as the common choice of grid in Case 1, the number of grid points is $2^\gamma + 1$, $\gamma = 1, 2, \dots$ with non-zero boundary while $2^\gamma - 1$, $\gamma = 1, 2, \dots$ with zero boundary.

Case 3. In the simple Gauss-Patterson rule, we also use a sequence of unequally discretised grids. However, we can't find a one to one map from its grid



(a) trapezoidal rule

(b) Clenshaw–Curtis

(c) Gauss–Patterson

Figure 1.1: 2D sparse grids with respect to trapezoidal rule, Clenshaw–Curtis and Gauss–Patterson rules for level $l = 5$. More quadrature points are close to the boundary in the Clenshaw–Curtis and Gauss–Patterson rules.

to an equally discretised grid for all grids generated by a certain Gauss-Patterson extension. For the commonly used Patterson extensions G_3, K_7, P^3_{15}, P^3_{31}, $\ldots$, the number of grid points are $2^{\gamma+1}-1$, $\gamma = 1, 2, \ldots$. Notice that we do not require any points in the boundary when we use Gauss-Patterson rule.

In the first two cases, the number of grid points of the product rule

$$Q_n \otimes \cdots \otimes Q_n \tag{1.19}$$

is $(2^n+1)^d$ with non-zero boundary while $(2^n-1)^d$ with zero boundary. For Patterson's extensions G_3, K_7, P^3_{15}, P^3_{31}, $\ldots$, the product rule (1.19) requires $(2^{n+1}-1)^d$ grid points.

Proposition 1.29 ([36]). *For Case 1 and Case 2, the number of grid points in a d dimensional level n sparse grid without boundaries is*

$$(-1)^d + 2^{n-d+1} \sum_{k=0}^{d-1} \binom{n}{d-1-k} (-2)^k. \tag{1.20}$$

Proof. We only need to discuss the piecewise linear interpolation since the number of grid points of other sparse grids discussed in Case 1 and Case 2 can be computed in the same way. Since we do not count the points on the boundary, the number of elements in the index set B_γ where γ is a d dimensional vector is

$$|B_\gamma| = 2^{\gamma_1 - 1} \times \cdots \times 2^{\gamma_d - 1} = 2^{|\gamma|-d}.$$

Thus the total number of grid points in the sparse grid is

$$\sum_{\gamma \geq \underline{1}, |\gamma| \leq n} |B_\gamma| = \sum_{k=d}^{n} \left(\sum_{\gamma \geq \underline{1}, |\gamma| = k} 2^{k-d} \right) = \sum_{k=d}^{n} \binom{k-1}{d-1} 2^{k-d} = \sum_{k=0}^{n-d} \binom{k+d-1}{d-1} 2^k.$$

The right-hand side can be computed as following

$$\begin{aligned} N(n,d) :=& \sum_{k=0}^{d} \binom{k+d-1}{d-1} 2^k \\ =& \frac{1}{(d-1)!} \sum_{k=0}^{n-d} (x^{k+d-1})^{(d-1)} \Big|_{x=2} \\ =& \frac{1}{(d-1)!} \left(\frac{x^{d-1} - x^n}{1-x} \right)^{(d-1)} \Bigg|_{x=2} . \end{aligned}$$

Applying the product rule to the last expression, we have

$$\begin{aligned}
N(n,d) =& \frac{1}{(d-1)!}\sum_{k=0}^{d-1}\left(\binom{d-1}{k}(x^{d-1}-x^n)^{(k)}\left(\frac{1}{1-x}\right)^{(d-1-k)}\right)\Bigg|_{x=2}\\
=& \frac{1}{(d-1)!}\sum_{k=0}^{d-1}\binom{d-1}{k}\left(\frac{x^{d-1-k}(d-1)!}{(d-1-k)!}-\frac{x^{n-k}n!}{(n-k)!}\right)\left(\frac{(d-1-k)!}{(1-x)^{d-k}}\right)\Bigg|_{x=2}\\
=& \sum_{k=0}^{d-1}\left(\binom{d-1}{k}x^{d-1-k}-\binom{n}{k}x^{n-k}\right)(1-x)^{-(d-k)}\Big|_{x=2}
\end{aligned}$$

Taking $x=2$, then

$$\begin{aligned}
N(n,d) =& \sum_{k=0}^{d-1}\left(\binom{d-1}{k}2^{d-1-k}-\binom{n}{k}2^{n-k}\right)(-1)^{-(d-k)}\\
=& -\sum_{k=0}^{d-1}\binom{d-1}{k}(-2)^{d-1-k}-2^n\sum_{k=0}^{d-1}\binom{n}{k}(-1)^{d-k}2^{-k}\\
=& (-1)^d-2^n\sum_{k=0}^{d-1}\binom{n}{d-1-k}(-1)^{k+1}2^{-(d-1-k)}\\
=& (-1)^d+2^{n-d+1}\sum_{k=0}^{d-1}\binom{n}{d-1-k}(-2)^k.
\end{aligned}$$

In the third equation, we apply the binomial theorem

$$(x+y)^{d-1}=\sum_{k=0}^{d-1}\binom{d-1}{k}x^ky^{d-1-k}$$

to the case $x=1$ and $y=-2$. □

Proposition 1.30. *For Case 3, the number of grid points in a d dimensional level n sparse grid without boundaries is*

$$(-1)^d+2^n\sum_{k=0}^{d-1}\binom{n+d-1}{k}(-2)^{d-1-k}. \tag{1.21}$$

Proof. Since the numbers of grid points of the nested girds generated by Patterson's extension are $2^{\gamma+1}-1$, $\gamma=1,2,\ldots$. Thus the index set B^g_γ is

$$B^g_\gamma=\begin{cases}\left\{1,3,5,\ldots,2^{\gamma+1}-3,2^{\gamma+1}-1\right\}, & \text{if } \gamma>0,\\ \{0\} & \text{if } \gamma=0.\end{cases}$$

in 1D case. Here we follow the construction in (1.18). In d dimensional case, let γ be a vector, the number of elements in d dimensional index set

$$B^g_\gamma = B^g_{\gamma_1} \times \cdots \times B^g_{\gamma_d}$$

is

$$|B^g_\gamma| = 2^{\gamma_1} \times \cdots \times 2^{\gamma_d} = 2^{|\gamma|}$$

where $\gamma \geq 0$. Then we have

$$\begin{aligned}\sum_{\gamma\geq\underline{0},|\gamma|\leq n} |B^g_\gamma| = \sum_{\gamma\geq\underline{1},|\gamma|\leq n+d-1} |B_\gamma| = \sum_{k=d}^{n+d-1} \sum_{\gamma\geq 1,|\gamma|=k} 2^{k-d} \\ = \sum_{k=d}^{n+d-1} \binom{k-1}{d-1} 2^{k-d} = \sum_{k=0}^{n-1} \binom{k+d-1}{d-1} 2^k.\end{aligned}$$

In the first equation, we shift the origin in the summation from $\gamma = \underline{0}$ to $\gamma = \underline{1}$. By using the same method in Proposition 1.29, we can get the desired result. □

The number of grid points in a full grid without boundary points is approximately $O(2^{dn})$. From the result of Proposition 1.29, the largest term in the sum of the (1.20) is when we take $k = 0$, i.e.

$$\begin{aligned}2^{n-d+1}\binom{n}{d-1} =& 2^{n-d+1}\frac{n!}{(n-d+1)!(d-1)!} \\ =& 2^{n-d+1}\left(\frac{n^{d-1}}{(d-1)!} + O(n^{d-2})\right).\end{aligned}$$

From the result of Proposition 1.30, the largest term in the sum of (1.21) is when we take $k = d - 1$, i.e.

$$\begin{aligned}2^n\binom{n+d-1}{d-1} &= 2^n\frac{(n+d-1)!}{n!(d-1)!} \\ &= 2^n\left(\frac{n^{d-1}}{(d-1)!} + O(n^{d-2})\right).\end{aligned}$$

Thus, the numbers of grid points in both cases grow asymptotically as $O(2^n n^{d-1})$. This much slower than $O(2^{dn})$ for full grid.

The computation of the number of grid points in a sparse grid with boundary points is more complicated. The following proposition gives an upper bound of the number of grid points used.

Proposition 1.31 ([45]). *For Case 1 and Case 2, for a level $n \geq 2(d-1)$ sparse grid with boundary points, the number of grid points is bounded above by*

$$1 + 2^{n-d+1}(6^d - 5^d)\binom{n}{d-1}.$$

Proof. See in [45]. □

From the proposition, we see the number of grid points in a d dimensional level n sparse grid also grows asymptotically as $O(2^n n^{d-1})$.

1.4 Error of Sparse Grid Approximations

In this section, we will consider the error of sparse grid interpolations and sparse grid quadratures. First, we will focus on the interpolation and quadrature using sparse grid which is build upon a sequence of equally spaced grids. In particular, we will review the result of the piecewise linear interpolation and the composite trapezoidal rule. These results are well known and can be found in the literature [16, 36]. Then we will consider more general case when the sparse grid is allowed to be constructed by a sequence of unequally spaced grids. In this case, we only discuss the simple polynomial interpolations and the simple quadrature rules of interpolatory type. The results for the composite interpolations and composite rules can be derived by using the results of the simple rules. The error of the sparse grid quadrature rules of high order polynomial interpolatory type was first discussed by Novak and Ritter in [68]. Later, the result on the error of high dimensional polynomial interpolation on sparse grids was given by Barthelmann, Novak and Ritter in [6]. We give new proofs for the main results in [68, 6].

First, we consider the piecewise linear interpolation with zero boundary condition. We require the function $f \in H^2_{0,mix}(X)$. The result in Lemma 1.12 ,i.e.

$$\|f^h_\alpha\|_2 \leq 3^{-d} 2^{-2|\alpha|} \|D^{\underline{2}} f\|_2$$

for any multi-index $\alpha \geq 1$ provides an upper bound for the contribution from the hierarchical space W_α. Using this bound together with the full grid approximation formula

$$f_\gamma = \sum_{\underline{0} \leq \alpha \leq \gamma} f^h_\alpha = \sum_{\underline{1} \leq \alpha \leq \gamma} f^h_\alpha. \tag{1.22}$$

We can bound the error of the full grid approximation

$$\epsilon_\gamma = f - f_\gamma = \sum_{\alpha \not\leq \gamma, \alpha \geq 1} f^h_\alpha.$$

Similarly, for the level n sparse grid approximation

$$f_n^s = \sum_{|\alpha|\leq n} f_\alpha^h,$$

the (1.22) can be used in bounding the error

$$\epsilon_n^s = f - f_n^s = \sum_{|\alpha|>n} f_\alpha^h.$$

For the multidimensional integration, if the integrand $f \in H^2_{0.mix}(X)$ and X is a bounded domain, by using the Cauchy-Schwarz inequality, we have

$$\left|\int_X f_\alpha^h \, dx\right| \leq \int_X |f_\alpha^h| \, dx \leq |X|^{\frac{1}{2}} ||f_\alpha^h||_2. \tag{1.23}$$

If we further assume the composite trapezoidal rule is used in building product rule and sparse grid quadrature and let f_γ and f_n^s be the underlying interpolating polynomials, then the error of the product rule is

$$e_\gamma = If - T_\gamma f = \int_X f \, dx - \int_X f_\gamma \, dx = \sum_{\alpha \nleq \gamma, \alpha \geq 1} \int_X f_\alpha^h \, dx \tag{1.24}$$

and the error of the corresponding sparse grid quadrature is

$$e_n^s = If - T_n^s f = \int_X f \, dx - \int_X f_n^s \, dx = \sum_{|\alpha|>n} \int_X f_\alpha^h \, dx. \tag{1.25}$$

The following two Propositions(adapted from [45]) give detailed computation of the bound of the error ϵ_γ, ϵ_n^s and e_γ, e_n^s.

Proposition 1.32. *Let $f \in H^2_{0,mix}(X)$, where X is a bounded domain. Then the error of the piecewise multi-linear interpolant f_γ satisfies*

$$\|\epsilon_\gamma\|_2 \leq 9^{-d} \|D^2 f\|_2 \sum_{k=1}^{d} 4^{-\gamma_k},$$

the error e_γ of the composite trapezoidal rule T_γ satisfies

$$|e_\gamma| \leq 9^{-d} |X|^{\frac{1}{2}} \|D^2 f\|_2 \sum_{k=1}^{d} 4^{-\gamma_k}.$$

Proof.

$$\begin{aligned}
\|\epsilon_\gamma\|_2 &\le \sum_{\alpha \not\le \gamma, \alpha \ge \underline{1}} \|f_\alpha^h\|_2 \\
&\le 3^{-d} \|D^{\underline{2}} f\|_2 \sum_{\alpha \not\le \gamma, \alpha \ge \underline{1}} 2^{-2|\underline{\alpha}|} \\
&= 3^{-d} \|D^{\underline{2}} f\|_2 \left(\sum_{\alpha \ge \underline{1}} 4^{-|\alpha|} - \sum_{\underline{1} \le \alpha \le \underline{\gamma}} 4^{-|\alpha|} \right) \\
&= 3^{-d} \|D^{\underline{2}} f\|_2 \left(\prod_{k=1}^{d} \left(\sum_{\alpha_k=1}^{\infty} 4^{-\alpha_k} \right) - \prod_{k=1}^{d} \left(\sum_{\alpha_k=1}^{\gamma_k} 4^{-\alpha_k} \right) \right) \\
&= 9^{-d} \|D^{\underline{2}} f\|_2 \left(1 - \prod_{k=1}^{d} (1 - 4^{-\gamma_k}) \right) \\
&\le 9^{-d} \|D^{\underline{2}} f\|_2 \sum_{k=1}^{d} 4^{-\gamma_k}.
\end{aligned}$$

The first inequality is due to the triangle inequality. The second inequality is derived by applying the Lemma 1.12. The following three equalities are definitions and direct computations. The last inequality satisfies because of the following result. Given $y_1, \dots, y_d \in (0, 1)$, we have

$$\sum_{k=1}^{d} y_k \ge 1 - \prod_{k=1}^{d} (1 - y_k). \tag{1.26}$$

This inequality can be proved by using the methods of mathematical induction. For the error of the composite trapezoidal rule, we have

$$\begin{aligned}
|e_\gamma| &\le \sum_{\alpha \not\le \gamma, \alpha \ge 1} \left| \int_X f_\alpha^h \, dx \right| \\
&\le |X|^{\frac{1}{2}} \sum_{\alpha \not\le \gamma, \alpha \ge \underline{1}} \|f_\alpha^h\|_2 \\
&\le 9^{-d} |X|^{\frac{1}{2}} \|D^{\underline{2}} f\|_2 \sum_{k=1}^{d} 4^{-\gamma_k}.
\end{aligned}$$

The first inequality is an application of the triangle inequality to (1.24). In the second inequality, we use the result in (1.23). The third inequality is due to the estimation in the piecewise multi-linear interpolant proof. □

Proposition 1.33. *Let $f \in H^2_{0,mix}(X)$, where X is a bounded domain. Then the error of the sparse grid interpolant f^s_n satisfies*

$$\|\epsilon^s_n\|_2 \leq 2^{-2n}\frac{1}{3}\left(\frac{1}{3}\right)^d \|D^{\underline{2}}f\|_2 \sum_{k=0}^{d-1}\binom{n}{k}\left(\frac{1}{3}\right)^{d-1-k}.$$

The error of the sparse grid quadrature T^s_n satisfies

$$\|e^s_n\|_2 \leq 2^{-2n}\frac{1}{3}\left(\frac{1}{3}\right)^d |X|^{\frac{1}{2}}\|D^{\underline{2}}f\|_2 \sum_{k=0}^{d-1}\binom{n}{k}\left(\frac{1}{3}\right)^{d-1-k}.$$

Proof.

$$\begin{aligned}
\|\epsilon^s_n\|_2 &\leq \sum_{|\alpha|>n,\gamma\geq\underline{1}} \|f^h_\alpha\|_2 \\
&\leq \sum_{|\alpha|>n,\gamma\geq\underline{1}} \left(\frac{1}{3}\right)^d 2^{-2|\alpha|}\|D^{\underline{2}}f\|_2 \\
&= \left(\frac{1}{3}\right)^d \|D^{\underline{2}}f\|_2 \sum_{k=n+1}^{\infty} \sum_{|\alpha|=k,\underline{\alpha}\geq\underline{1}} 2^{-2k} \\
&= \left(\frac{1}{3}\right)^d \|D^{\underline{2}}f\|_2 \sum_{k=n+1}^{\infty} \binom{k-1}{d-1} 2^{-2k} \\
&= 2^{-2(n+1)}\left(\frac{1}{3}\right)^d \|D^{\underline{2}}f\|_2 \sum_{k=0}^{\infty} \binom{k+n}{d-1} 2^{-2k} \\
&= 2^{-2(n+1)}\left(\frac{1}{3}\right)^d \|D^{\underline{2}}f\|_2 \left[\frac{4}{3}\sum_{k=0}^{d-1}\binom{n}{k}\left(\frac{1}{3}\right)^{d-1-k}\right] \\
&= 2^{-2n}\frac{1}{3}\left(\frac{1}{3}\right)^d \|D^{\underline{2}}f\|_2 \sum_{k=0}^{d-1}\binom{n}{k}\left(\frac{1}{3}\right)^{d-1-k}.
\end{aligned}$$

In the first inequality, we apply the triangle inequality. We use the Lemma 1.12 to get the second inequality. The first equality is the direct computation. The second equality is due to the fact

$$\binom{k-1}{d-1} = \sum_{|\alpha|=k,\alpha\geq\underline{1}} 1.$$

In the third equality, we change the starting index in the summation. For the following equality, we apply a similar approach as in the proofs of the Propositions

1.29 and 1.30. Suppose $x \in (0,1)$, then

$$\begin{aligned}
\sum_{k=0}^{\infty} x^k \binom{k+n}{d-1} &= \frac{x^{-n}}{(d-1)!} \left(\sum_{k=0}^{\infty} x^{k+n} \right)^{(d-1)} \\
&= \frac{x^{-n}}{(d-1)!} \left(\frac{x^n}{1-x} \right)^{(d-1)} \\
&= \frac{x^{-n}}{(d-1)!} \sum_{k=0}^{d-1} \binom{d-1}{k} (x^n)^{(k)} \left(\frac{1}{1-x} \right)^{(d-1-k)} \\
&= \frac{x^{-n}}{(d-1)!} \sum_{k=0}^{d-1} \binom{d-1}{k} \frac{x^{n-k} n!}{(n-k)!} \frac{(d-1-k)!}{(1-x)^{(d-k)}} \\
&= \sum_{k=0}^{d-1} \binom{n}{k} \left(\frac{x}{1-x} \right)^{(d-1-k)} \frac{1}{1-x}.
\end{aligned}$$

If we take $x = 2^{-2}$, then

$$\sum_{k=0}^{\infty} \binom{k+n}{d-1} 2^{-2k} = \frac{4}{3} \sum_{k=0}^{d-1} \binom{n}{k} \left(\frac{1}{3} \right)^{d-1-k}. \tag{1.27}$$

For the error of the sparse grid quadrature T_n^s, as we did in the proof of the previous proposition, we have

$$\begin{aligned}
|e_\gamma| &\leq \sum_{|\alpha|>n, \alpha \geq 1} \left| \int_X f_\alpha^h \, dx \right| \\
&\leq |X|^{\frac{1}{2}} \sum_{|\alpha|>n, \alpha \geq \underline{1}} \|f_\alpha^h\|_2 \\
&\leq 2^{-2n} \frac{1}{3} \left(\frac{1}{3} \right)^d |X|^{\frac{1}{2}} \|D^{\underline{2}} f\|_2 \sum_{k=0}^{d-1} \binom{n}{k} \left(\frac{1}{3} \right)^{d-1-k}.
\end{aligned}$$

The first inequality is an application of the triangle inequality to (1.4). In the second inequality, we again use the result in (1.23). □

In the Proposition 1.33, if we further assume that $n > 2(d-1)$, then we have

$$\|\epsilon_n^s\|_2 \leq 2^{-2n} \frac{1}{2} \left(\frac{1}{3} \right)^d \|D^{\underline{2}} u\|_2 \binom{n}{d-1}.$$

Therefore, the convergence rate of the sparse grid methods f_n^s and T_n^s is approximately $O(2^{-2n} n^{d-1})$.

Now we start to further study the errors of full grid approximation and sparse grid approximation in more general cases, i.e. the multi-dimensional polynomial interpolation and quadrature of polynomial interpolatory type. Faster convergence will be expected if the function or the integrand f has higher order smoothness. Therefore, new function spaces are required to be used for further discussion since functions in the function space $H^2_{0,mix}$ are not 'smooth' enough. Here we use the spaces introduced in [68, 6]. For $d = 1$, we denote

$$F_1^s = C^s([-1,1]), \quad s \in \mathbb{N}$$

with the norm

$$\|f\| = \max\left\{\|D^i f\|_\infty,\ i = 0,\ldots,s\right\}. \tag{1.28}$$

For $d > 1$, we consider

$$F_d^s = \left\{f : X \to \mathbb{R} \,|\, D^i f \text{ continuous if } i_k \leq s \text{ for all } k\right\} \tag{1.29}$$

with the norm[‡]

$$\|f\| = \max\left\{\|D^i f\|_\infty \,|\, i \in \mathbb{N}^d, i_k \leq s\right\}. \tag{1.30}$$

For any $f \in F_d^s$, it can be written as a finite linear combination of functions

$$(f_1 \otimes \cdots \otimes f_d)(x_1,\ldots,x_d) = f_1(x_1)\cdots f_d(x_d)$$

where $f_i \in F_1^s$ are dense in F_d^s and

$$\|f_1 \otimes \cdots \otimes f_d\| = \|f_1\|\cdots\|f_d\|.$$

Suppose $\mathcal{U}$ is a bounded linear operator defined as

$$\begin{aligned} \mathcal{U} : F_1^s &\to V_1 \\ f &\mapsto \mathcal{U}f. \end{aligned}$$

The V_1 here is a subspace of F_1^s. Then we have

$$\|\mathcal{U}f\| \leq \|\mathcal{U}\|\|f\|$$

where the norm of the operator $\mathcal{U}$ is induced by the norm $\|\cdot\|$. If $\mathcal{U}_i$, $i = 1,\ldots,d$ are above defined operators, then the operator $\mathcal{U}_1 \otimes \cdots \otimes \mathcal{U}_d$ is defined as

$$\begin{aligned} \mathcal{U}_1 \otimes \cdots \otimes \mathcal{U}_d : F_d^s &\to V_d \\ f &\mapsto (\mathcal{U}_1 \otimes \cdots \otimes \mathcal{U}_d)f \end{aligned}$$

[‡] The formal way to write this norm is $\|f\|_{F_d^s}$. Since this norm is frequently used in our proof, we simplify the notation here when d and s are known.

where V_d is a subspace of F_d^s. The norm of the operator $\mathcal{U}_1 \otimes \cdots \otimes \mathcal{U}_d$ is

$$\|\mathcal{U}_1 \otimes \cdots \otimes \mathcal{U}_d\| = \|\mathcal{U}_1\| \dots \|\mathcal{U}_d\|. \tag{1.31}$$

We first study the error bound of the polynomial interpolation and integration rule of the polynomial interpolatory type for 1D case. The d dimensional error bound can be derived by using the result from 1D case.

According to the Lebesgue's Lemma [29], the $n_\gamma + 1$ points Lagrangian interpolant operator L_{n_γ} satisfies

$$\|f - L_{n_\gamma} f\|_p \leq (1 + \|L_{n_\gamma}\|) E^p_{n_\gamma}(f), \quad 1 \leq p \leq \infty \tag{1.32}$$

where $\|L_{n_\gamma}\|$ is the Lebesgue constant [22] and $E^p_{n_\gamma} f$ is the error of the best approximation by polynomial of degree up to n_γ in L_p norm.

Let C be a constant. By using the Jackson's theorem [21], we have for $f \in F_1^s$

$$E^p_{n_\gamma}(f) \leq C n_\gamma^{-s} \|f^{(s)}\|_p. \tag{1.33}$$

The constant C here depends on the smoothness s.

The Lebesgue constant depends on the choice of the grid points used in an interpolation/quadrature formula and also the norm. Here we only focus on the two kinds of grid points mentioned in the previous sections, i.e. the Chebyshev-Gauss-Lobatto points and quadrature points used in the Kronrod's scheme. For both cases, the Lebesgue constants are of $O(\log n_\gamma)$(CGL points [29, 32], Kronrod's scheme [34, 33]) when we use the L^p, $p = 1, \infty$ norm. The operator L_{n_γ} are bounded [29, 34] when we use the L^p, $1 \leq p \leq \infty$ norm.

Combining the results on the Lebesgue constant with (1.32) and (1.33), we have

$$\|f - L_{n_\gamma} f\|_p = \begin{cases} C \log n_\gamma n_\gamma^{-s} \|f^{(s)}\|_p, & p = 1, \infty, \\ C n_\gamma^{-s} \|f^{(s)}\|_p, & 1 < p < \infty \end{cases} \tag{1.34}$$

Since we are only interested in the convergence rate, all the constant terms are denoted as C here and in the following proofs.

Lemma 1.34. *Suppose $f \in F_1^s$. If we again denote the level α 1D hierarchical surplus as f_α^h, i.e.*

$$f_\alpha^h = L_{n_\alpha} f - L_{n_{\alpha-1}} f = (L_{n_\alpha} - L_{n_{\alpha-1}}) f,$$

then we have

$$\|L_{n_\alpha} - L_{n_{\alpha-1}}\| \leq C \alpha 2^{-\alpha s}.$$

where C is a constant. For the quadrature rule of interpolatory type, the 1D hierarchical surplus is

$$\int_X f_\alpha^h \, dx = Q_\alpha f - Q_{\alpha-1} f = (Q_\alpha - Q_{\alpha-1}) f$$

and the operator $Q_\alpha - Q_{\alpha-1}$ satisfies

$$\|Q_\alpha - Q_{\alpha-1}\| \leq C 2^{-\alpha s}.$$

C is again a constant.

Proof. We have

$$\begin{aligned}
\|f_\alpha^h\| &= \|L_{n_\alpha} f - L_{n_{\alpha-1}} f\| \\
&\leq \|f - L_{n_\alpha} f\| + \|f - L_{n_{\alpha-1}} f\| \\
&\leq C_1 \log n_\alpha n_\alpha^{-s} \|f\| + C_2 \log n_{\alpha-1} n_{\alpha-1}^{-s} \|f\| \\
&\leq C\alpha 2^{-\alpha s} \|f\| + C(\alpha-1) 2^{-(\alpha-1)s} \|f\| \\
&= C\alpha 2^{-\alpha s}(1 + 2^s) \|f\| \\
&= C\alpha 2^{-\alpha s} \|f\|.
\end{aligned}$$

The first inequality is due to the triangle inequality. In the second inequality we use the result (1.34). We take $C = \max\{C_1, C_2\}$ and use the fact

$$n_\alpha = 2^{\alpha-1} + 1 \leq 2^\alpha, \quad \alpha \geq 1$$

for the third inequality. The following two equations are direct computation and the term $1 + 2^s$ is absorbed into the constant term C. For the hierarchical surplus of integral, we have

$$\begin{aligned}
\left| \int_X f_\alpha^h \, dx \right| &\leq |X|^{\frac{1}{2}} \|f_\alpha^h\|_2 \\
&\leq |X|^{\frac{1}{2}} \|L_{n_\alpha} f - L_{n_{\alpha-1}} f\|_2 \\
&\leq |X|^{\frac{1}{2}} (\|f - L_{n_\alpha} f\|_2 + \|f - L_{n_{\alpha-1}} f\|_2) \\
&\leq C|X|^{\frac{1}{2}} (n_\alpha^{-s} + n_{\alpha-1}^{-s}) \|f^{(s)}\|_2 \\
&\leq C|X| (n_\alpha^{-s} + n_{\alpha-1}^{-s}) \|f\| \\
&\leq C|X| (2^{-\alpha s} + 2^{-(\alpha-1)s}) \|f\| \\
&\leq C 2^{-\alpha s} \|f\|.
\end{aligned}$$

We apply the Cauchy Schwarz inequality in the first inequality. The fourth inequality is the result of the Jackson's Theorem. In the fifth inequality, we use fact that the L_2 norm is bounded by the infinite norm on a bounded domain, i.e.

$$\|f^{(s)}\|_2 = \left(\int_X (f^{(s)})^2\,dx\right)^{\frac{1}{2}} \leq \left(\int_X \|f^{(s)}\|_\infty^2\,dx\right)^{\frac{1}{2}} \leq |X|^{\frac{1}{2}}\|f^{(s)}\|_\infty.$$

□

Lemma 1.35. *Suppose $f \in F_d^s$. Let $\alpha = (\alpha_1, \ldots, \alpha_d)$ be a vector. The d dimensional level α hierarchical surplus is f_α^h for the polynomial interpolation. It is bounded by*

$$\|f_\alpha^h\| \leq \|L_{n_\alpha} - L_{n_{\alpha-1}}\|\|f\| \leq C\left(\prod_{i=1}^d \alpha_i\right)2^{-|\alpha|s}\|f\|.$$

The d dimensional level α hierarchical surplus is $\int_X f_\alpha^h\,dx$ for the quadrature rule of interpolatory type. It is bounded by

$$\left|\int_X f_\alpha^h\,dx\right| \leq \|Q_\alpha - Q_{\alpha-1}\|\|f\| \leq C2^{-|\alpha|s}\|f\|.$$

Proof. According to (1.31), we have

$$\|L_{n_\alpha} - L_{n_{\alpha-1}}\| = \prod_{i=1}^d \|L_{n_{\alpha_i}} - L_{n_{\alpha_i-1}}\| \leq C\left(\prod_{i=1}^d \alpha_i\right)2^{-|\alpha|s}.$$

and

$$\|Q_\alpha - Q_{\alpha-1}\| \leq \prod_{i=1}^d \|Q_{\alpha_i} - Q_{\alpha_i-1}\| \leq C2^{-|\alpha|s}.$$

□

Lemma 1.36. *Suppose we have an arithmetic sequence $\{a_\alpha\}$, $\alpha = 1, 2, \ldots$ and a geometric sequence $\{b_\alpha\}$, $\alpha = 1, 2, \ldots$. The αth term of the arithmetic sequence is given by*

$$a_\alpha = a_1 + (\alpha - 1)d, \quad \alpha \geq 2$$

where the initial value a_1 and the difference d are given. The αth term of the geometric sequence is given by

$$b_\alpha = b_1 q^{\alpha-1}, \quad \alpha \geq 2.$$

where the initial value b_1 and the ratio q are given. In addition $|q| < 1$. Then the sum of the first n term of the sequence $\{a_\alpha b_\alpha\}$, $\alpha = 1, 2, \ldots$ is

$$S_n = \frac{a_1 b_1}{1-q} + \frac{d b_1 (q - q^{n-1})}{(1-q)^2} - \frac{[a_1 + (n-1)d]b_1 q^n}{1-q}. \tag{1.35}$$

and its limit is

$$\lim_{n\to\infty} S_n = \frac{a_1 b_1}{1-q} + \frac{d b_1 q}{(1-q)^2}. \tag{1.36}$$

Proof. By direct computation, we have

$$\begin{aligned}(1-q)S_n &= \sum_{\alpha=1}^{n}(a_1+(\alpha-1)d)b_1q^{\alpha-1} - \sum_{\alpha=1}^{n}(a_1+(\alpha-1)d)b_1q^{\alpha} \\ &= a_1b_1 + \sum_{\alpha=1}^{n-1}(a_1+\alpha d)b_1q^{\alpha} - \sum_{\alpha=1}^{n}(a_1+(\alpha-1)d)b_1q^{\alpha} \\ &= a_1b_1 + \sum_{\alpha=1}^{n-1} db_1q^{\alpha} - [a_1+(n-1)d]b_1q^{n} \\ &= a_1b_1 + \frac{db_1(q-q^{n-1})}{(1-q)} - [a_1+(n-1)d]b_1q^{n}.\end{aligned}$$

Since $|q|<1$, we can divide both sides by $1-q$ and get the (1.35) and (1.36). □

Proposition 1.37. *Let $f \in F_d^s$. X is a bounded domain. Then the error of the polynomial interpolant $L_{n_\gamma} f$ satisfies*

$$\|f - L_{n_\gamma} f\| \leq C\sum_{k=1}^{d} 2^{-s(\gamma_k-1)}\|f\|.$$

The error of the quadrature rule Q_γ of the interpolatory type satisfies

$$|If - Q_\gamma f| \leq C\sum_{k=1}^{d} 2^{-s\gamma_k}\|f\|.$$

Proof. For the polynomial interpolation error, we have

$$\begin{aligned}\|f - L_{n_\gamma f}\| &\leq \sum_{\alpha\nleq\gamma,\alpha\geq\underline{1}} \|f_\alpha^h\| \\ &\leq \sum_{\alpha\nleq\gamma,\alpha\geq\underline{1}} C\left(\prod_{i=1}^{d}\alpha_i\right)2^{-|\alpha|s}\|f\| \\ &= C\left(\sum_{\alpha\geq\underline{1}}\left(\prod_{i=1}^{d}\alpha_i\right)2^{-|\alpha|s} - \sum_{\underline{1}\leq\alpha\leq\gamma}\left(\prod_{i=1}^{d}\alpha_i\right)2^{-|\alpha|s}\right)\|f\| \\ &\leq C\left(\prod_{i=1}^{d}\left(\sum_{\alpha_i\geq1}\alpha_i2^{-\alpha_i s}\right) - \prod_{i=1}^{d}\left(\sum_{1\leq\alpha_i\leq\gamma_i}\alpha_i2^{-\alpha_i s}\right)\right)\|f\|\end{aligned}$$

By using the Lemma 1.36, we have

$$\begin{aligned}\sum_{1\leq\alpha_i\leq\gamma_i}\alpha_i2^{-\alpha_i s} &= 2^{-s}\left[\frac{1}{1-2^{-s}} + \frac{2^{-s}-2^{-s(\gamma_i-1)}}{(1-2^{-s})^2} - \frac{\gamma_k 2^{-s\gamma_i}}{1-2^{-s}}\right] \\ &\leq \frac{2^{-s}}{(1-2^{-s})^2}(1-2^{-s(\gamma_i-1)}).\end{aligned}$$

and

$$\sum_{\alpha_i \geq \alpha_i} \alpha_i 2^{-\alpha_i s} = \frac{2^{-s}}{(1-2^{-s})^2}.$$

By applying the inequality (1.26) in Proposition 1.32. We have

$$\begin{aligned}\|f - L_{n_\gamma} f\| \leq & C\frac{2^{-sd}}{(1-2^{-s})^{2d}} \prod_{i=1}^{d} [1-(1-2^{-s(\gamma_i-1)})]\|f\| \\ \leq & C\frac{2^{-sd}}{(1-2^{-s})^{2d}} \sum_{i=1}^{d} 2^{-s(\gamma_i-1)}\|f\|.\end{aligned}$$

For the quadrature error, we have

$$\begin{aligned}|Qf - Q_\gamma f| &\leq \sum_{\alpha \nleq \gamma, \alpha \geq \underline{1}} \left| \int_X f_\alpha^h \, dx \right| \\ &\leq \sum_{\alpha \nleq \gamma, \alpha \geq \underline{1}} C 2^{-|\alpha| s} \|f\| \\ &= C \left(\sum_{\alpha \geq \underline{1}} 2^{-|\alpha| s} - \sum_{\underline{1} \leq \alpha \leq \gamma} 2^{-|\alpha| s} \right) \|f\| \\ &\leq C \left(\prod_{i=1}^{d} \left(\sum_{\alpha_i \geq 1} 2^{-\alpha_i s} \right) - \prod_{i=1}^{d} \left(\sum_{1 \leq \alpha_i \leq \gamma_i} 2^{-\alpha_i s} \right) \right) \|f\|\end{aligned}$$

By applying the inequality (1.26) in Proposition 1.32. We have

$$\begin{aligned}|Qf - Q_\gamma f| &\leq C \left(1 - \prod_{i=1}^{d} (1-2^{-\gamma_i s}) \right) \|f\| \\ &\leq C \sum_{i=1}^{d} 2^{-\gamma_i s} \|f\|.\end{aligned}$$

□

Proposition 1.38. *Let $f \in F_d^s$. X is a bounded domain. Then the error of the sparse grid polynomial interpolant f_n^s satisfies*

$$\|f_n^s - f\| \leq C 2^{-sn} \sum_{k=0}^{2d-1} \binom{n+d}{k} \left(\frac{2^{-s}}{1-2^{-s}} \right)^{2d-k}.$$

The error of the sparse grid quadrature $Q_n^s f$ which is built on the quadrature rules of the interpolatory type satisfies

$$\|Q_n^s f - f\| \leq C 2^{-sn} \sum_{k=0}^{d-1} \binom{n}{k} \left(\frac{2^{-s}}{1-2^{-s}} \right)^{d-1-k}.$$

Proof. From the Lemma 1.35 and using the inequality

$$\alpha_1 \dots \alpha_d \le \left(\frac{\alpha_1 + \cdots + \alpha_d}{d}\right)^d,$$

we have

$$\begin{aligned} \|f_\alpha^h\| &\le C\alpha_1 \dots \alpha_d 2^{-s|\alpha|} \\ &\le C\frac{1}{d^d}|\alpha|^d 2^{-s|\alpha|} \\ &= C|\alpha|^d 2^{-s|\alpha|}. \end{aligned}$$

Similar as the estimation in Proposition 1.33, the interpolation error bound can be computed by

$$\begin{aligned} \|f_n^s - f\| &\le \sum_{|\alpha|\ge n, \alpha \ge \underline{1}} \|f_\alpha^h\| \\ &\le \sum_{|\alpha|\ge n, \alpha \ge \underline{1}} C|\alpha|^d 2^{-s|\alpha|} \\ &= C\sum_{k=n+1}^{\infty} \sum_{|\alpha|=k, \alpha\ge\underline{1}} k^d 2^{-sk} \\ &= C\sum_{k=n+1}^{\infty} \binom{k-1}{d-1} k^d 2^{-sk} \\ &= C\sum_{k=0}^{\infty} \binom{k+n}{d-1} (k+n+1)^d 2^{-s(k+n+1)} \\ &= C2^{-s(n+1)} \sum_{k=0}^{\infty} \binom{k+n}{d-1} (k+n+1)^d 2^{-sk}. \end{aligned} \tag{1.37}$$

Here we estimate the following power series when $x \in (0, 1)$.

$$\begin{aligned} &\sum_{k=0}^{\infty} \binom{k+n}{d-1} (k+n+1)^d x^k \\ =&\sum_{k=0}^{\infty} \frac{1}{(d-1)!}(k+n)\dots(k+n-d+2)(k+n+1)^d x^k \\ \le&\frac{1}{(d-1)!}\sum_{k=0}^{\infty}(k+n+d)\dots(k+n-d+2)x^k \\ =&\frac{x^{-n+d-1}}{(d-1)!}\sum_{k=0}^{\infty}(x^{k+n+d})^{(2d-1)} \end{aligned} \tag{1.38}$$

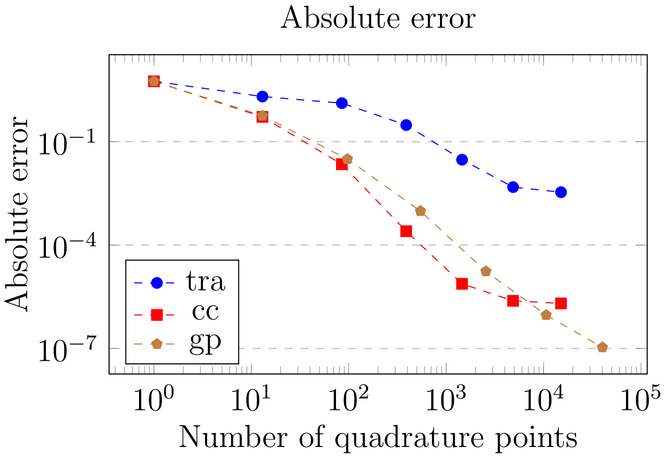


Figure 1.2: We compute the integral $\int_{[0,1]^d} \prod_{i=1}^d \exp(x_i)\ dx$ when $d = 6$ using sparse grids generated by the three different univariate quadrature rules. The Gauss–Patterson(gp) and Clenshaw–Curtis(cc) rules perform much better than the trapezoidal rule.

$$\begin{aligned}
&=\frac{x^{-n+d-1}}{(d-1)!}\left(\frac{x^{n+d}}{1-x}\right)^{(2d-1)} \\
&=\frac{x^{-n+d-1}}{(d-1)!}\left(\sum_{k=0}^{2d-1}\binom{2d-1}{k}(x^{n+d})^{(k)}\left(\frac{1}{1-x}\right)^{2d-1-k}\right) \\
&=\frac{x^{-n+d-1}}{(d-1)!}\left(\sum_{k=0}^{2d-1}\binom{2d-1}{k}\frac{x^{n+d-k}(n+d)!}{(n+d-k)!}\frac{(2d-1-k)!}{(1-x)^{(2d-k)}}\right) \\
&=\frac{(2d-1)!}{(d-1)!}\sum_{k=0}^{2d-1}\binom{n+d}{k}\left(\frac{x}{1-x}\right)^{(2d-k-1)}\frac{1}{1-x}.
\end{aligned} \tag{1.39}$$

If we take $x = 2^{-s}$ in the (1.38), (1.39) and combine the result in (1.39), we get

$$\|f_n^s - f\| \leq C\frac{(2d-1)!}{(d-1)!}2^{-sn}\sum_{k=0}^{2d-1}\binom{n+d}{k}\left(\frac{2^{-s}}{1-2^{-s}}\right)^{2d-k}.$$

For the error of the sparse grid quadrature built on the quadrature rules of the interpolatory type, the proof is the same as the proof in the Proposition 1.33 except we replace the L_2 norm with the norm $\|\cdot\|$ defined in (1.28) and take $x = 2^{-s}$ instead of $x = 2^{-2}$ in (1.27) . □

In the Proposition 1.38, if we further assume that $n > 3d - 2$, then we have

$$\|f_n^s - f\| \leq C2^{-sn}\binom{n+d}{2d-1}.$$

Therefore, the convergence rate of the sparse grid polynomial interpolant f_n^s is approximately $O(2^{-sn}n^{2d-1})$. The error of the sparse grid quadrature $Q_n^s f$ is approximately $O(2^{-sn}n^{d-1})$ according to the Proposition 1.38.

In the Figure 1.4, we show the convergence of three sparse grid quadratures. As shown in the error bounds, The Gauss Patterson and Clenshaw Curtis rules perform much better than the trapezoidal rule.

1.5 Conclusions

In this Chapter, we review the constructions of the sparse grid interpolation and the sparse grid quadrature. Since all the 1D quadrature rules we have discussed are rules of interpolatory type, sparse grid interpolations and sparse grid quadratures are closely related in this case. We also look into the error of the sparse grid interpolation and quadrature. From the error analysis, we can see the superiority of the sparse grid method in solving high dimensional problems.

Chapter 2

The Sparse Grid Combination Technique

The sparse grid combination technique [43, 36, 16] is used to approximate the numerical result of the sparse grid method. The general idea of the sparse grid combination technique is to first compute approximations of the function or the integral on several anisotropic regular grids. Then we compute a linear combination of these approximations to get a new approximation. The coefficients in the linear combination can be obtained by the inclusion-exclusion principle. The advantage of the sparse grid combination technique is that we can avoid using hierarchical basis functions. The usage of the hierarchical basis functions leads to a dense stiffness matrix which makes it hard to implement a fast matrix-vector product. Another advantage of the sparse grid combination technique is that the underlying algorithm is suitable for parallel computation which makes it possible to solve high dimensional problems and large scale problems.

In Chapter 2, we will first introduce the classical sparse grid combination technique. Then we will discuss the number of unknowns and the error splitting models [59, 19, 18]. Next, we will study the how to derive an error splitting model for linear and polynomial interpolation. Finally, we will give an introduction to generalised sparse grid combination techniques [46, 38, 45, 48]. Different from previous works in [43, 76], we establish new error splitting models when the underlying grids are not equally spaced. We also prove a new convergence result of the generalised combination technique based on the works in [46, 45].

2.1 The Classical Sparse Grid Combination Technique

Suppose we have a sequence of 1D nested grids. Here we do not require these 1D grids to be equally spaced. The multidimensional grid G_γ where γ is now a vector is the Cartesian product of these 1D grids. Using the same notation as in the Chapter 1, the anisotropic regular grid is defined as

$$G_\gamma = G_{\gamma_1} \times \cdots \times G_{\gamma_d}.$$

The interpolant of function f on grid G_γ is f_γ. The quadrature rule computed on grid G_γ is $Q_\gamma f$ with the integrand f. Suppose we have a multi-indices set I. The sparse grid combination technique with respect to I gives the interpolant

$$f_I^c = \sum_{\gamma \in I} c_\gamma f_\gamma$$

and the quadrature rule

$$Q_I^c f = \sum_{\gamma \in I} c_\gamma Q_\gamma f.$$

Since all commonly used quadrature rules discussed in Chapter 1, i.e. the trapezoidal rule, the Clenshaw Curtis rule and the Gauss Patterson rule are interpolatory, we only need to study the sparse grid combination technique for function interpolation. The sparse grid combination technique for integration can then be obtained by

$$Q_I^c f = \int_X f_I^c \, dx = \int_X \sum_{\gamma \in I} c_\gamma f_\gamma \, dx = \sum_{\gamma \in I} c_\gamma \int_X f_\gamma \, dx = \sum_{\gamma \in I} c_\gamma Q_\gamma f.$$

Therefore, we will only focus on the sparse grid combination technique for interpolation in the following discussion.

In particular, if we take the multi-indices set I as

$$I = \{\gamma \,|\, \gamma_1 + \cdots + \gamma_d = n - k, \, k = 0, \ldots, d-1, \, \gamma \geq 0\}$$

We get a combination technique which approximates the classical sparse grid interpolant f_n^s. It is defined as

$$f_n^c := f_I^c := \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} \sum_{|\gamma| = n-k} f_\gamma. \tag{2.1}$$

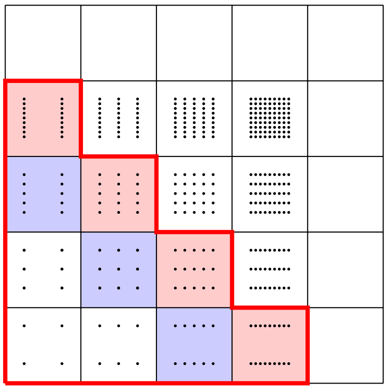


Figure 2.1: 2D classical sparse grid combination technique with $n = 3$. All the component grids are arranged according to the frequency of data points in each dimension. The colored component grids are used in the computation. Red component grids have plus signs while blue component grids have minus signs in the combination technique formula.

When $d = 2$, we have

$$f_n^c = \sum_{|\gamma|=n} f_\gamma - \sum_{|\gamma|=n-1} f_\gamma.$$

The Figure 2.1 shows the 2D combination technique when $n = 3$.

In fact, we have $f_n^c = f_n^s \in V_n^s$. In the following Lemma, we prove this result in 2D case*. The proof is adapted from [36]. The difference is the original proof is considered on equally discretised grid. Here we consider computing on a more generalised grid.

Lemma 2.1. *For a given 2D function f, the interpolant f_n^c given by the 2D combination technique* (2.1) *is equal to the classical sparse grid interpolant f_n^s.*

Proof. Here we replace the vector index γ with two indices γ_1 and γ_2 for simplicity. Then the 2D combination technique is

$$f_n^c = \sum_{\gamma_1+\gamma_2=n} f_{\gamma_1,\gamma_2} - \sum_{\gamma_1+\gamma_2=n-1} f_{\gamma_1,\gamma_2}. \tag{2.2}$$

For the interpolant f_{γ_1,γ_2} computed on the regular anisotropic grids, we can decompose it as a sum of all the hierarchical surpluses $f^h_{\alpha_1,\alpha_2}$, $\alpha_1 \leq \gamma_1$, $\alpha_2 \leq \gamma_2$ as we did in (1.6), namely,

$$f_{\gamma_1,\gamma_2} = \sum_{\alpha_1 \leq \gamma_1} \sum_{\alpha_2 \leq \gamma_2} f^h_{\alpha_1,\alpha_2}. \tag{2.3}$$

[*] The d dimensional case can be similarly derived. This will be further discussed in the Section 2.4

Insert (2.3) into (2.2), we have

$$\begin{aligned}
f_n^c =& \sum_{\gamma_1+\gamma_2=n} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2\le\gamma_2} f^h_{\alpha_1,\alpha_2} - \sum_{\gamma_1+\gamma_2=n-1} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2\le\gamma_2} f^h_{\alpha_1,\alpha_2} \\
=& \sum_{\gamma_1\le n} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2\le n-\gamma_1} f^h_{\alpha_1,\alpha_2} - \sum_{\gamma_1\le n-1} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2\le n-\gamma_1-1} f^h_{\alpha_1,\alpha_2} \\
=& \sum_{\gamma_1=n} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2=n-\gamma_1} f^h_{\alpha_1,\alpha_2} \\
&+ \left(\sum_{\gamma_1\le n-1} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2\le n-\gamma_1} f^h_{\alpha_1,\alpha_2} - \sum_{\gamma_1\le n-1} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2\le n-\gamma_1-1} f^h_{\alpha_1,\alpha_2} \right) \\
=& \sum_{\gamma_1=n} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2=n-\gamma_1} f^h_{\alpha_1,\alpha_2} + \sum_{\gamma_1\le n-1} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2=n-\gamma_1} f^h_{\alpha_1,\alpha_2} \\
=& \sum_{\gamma\le n} \sum_{\alpha_1\le\gamma_1} \sum_{\alpha_2=n-\gamma_1} f^h_{\alpha_1,\alpha_2} \\
=& \sum_{\alpha_1+\alpha_2\le n} f^h_{\alpha_1,\alpha_2} = f_n^s.
\end{aligned}$$

The underlying idea of the above computation is based on the inclusion-exclusion principle. □

2.2 Number of Grid Points in the Sparse Grid Combination Technique

Suppose the number of grid points in an anisotropic regular grid G_γ is $|G_\gamma|$. According to the classical sparse grid combination technique formula (2.1), the number of points used in computing it is

$$\sum_{k=0}^{d-1} \sum_{|\gamma|=n-k} |G_\gamma|. \tag{2.4}$$

Since the grid points are considered only once on a classical sparse grid while they are considered multiple times in (2.4), the number of grid points used in a classical sparse grid combination technique is more than those used in a classical sparse grid.

In Chapter 1, we discuss the number of grid points used in a sparse grid in details. It grows asymptotically as $O(2^n n^{d-1})$ for all the cases we discussed. These cases include the equally discretised grid/CGL points(which is defined in the previous Chapter) with or without boundary and the grid generated in the

Gauss Patterson method. We need to show that the number of grid points used in a classical sparse grid combination technique still grows approximately $O(2^n n^{d-1})$, otherwise the cost of combination technique can be far more expensive than that of computing the sparse grid directly.

For the number of grid points in an anisotropic grid G_γ, we have

Case 1: Equally discretised grid/CGL points without boundary

$$|G_\gamma| = \prod_{k=1}^{d} (2^{\gamma_k} - 1).$$

Case 2: Equally discretised grid/CGL points with boundary

$$|G_\gamma| = \prod_{k=1}^{d} (2^{\gamma_k} + 1).$$

Case 3: Grid used in Gauss Patterson method

$$|G_\gamma| = \prod_{k=1}^{d} (2^{\gamma_k+1} - 1).$$

We first consider the case when $d = 2$. We have the following Lemma

Lemma 2.2. *Suppose $d = 2$. The total number of grid points used in computation of f_n^c is*

Case 1: Equally discretised grid/CGL points without boundary

$$2n + 5 + \left(\frac{3}{2}n - 5\right) 2^n$$

Case 2: Equally discretised grid/CGL points with boundary

$$2n - 3 + \left(\frac{3}{2}n + 7\right) 2^n$$

Case 3: Grid used in the Gauss Patterson method

$$2n + 9 + (6n - 8)2^n$$

Thus, the total number of grid points grows approximately as $O(2^n\, n)$.

Proof. For Case 1, let

$$a_{n,2} = \sum_{\gamma_1+\gamma_2=n} (2^{\gamma_1} - 1)(2^{\gamma_2} - 1)$$

where the subscript 2 denotes dimension. Then

$$\begin{aligned} a_{n,2} &= \sum_{\gamma_1+\gamma_2=n} (2^{\gamma_1+\gamma_2} - 2^{\gamma_1} - 2^{\gamma_2} + 1) \\ &= \sum_{\gamma_1=0}^{n} (2^n - 2^{\gamma_1} - 2^{n-\gamma_1} + 1) \\ &= (n+3) + (n-3)2^n. \end{aligned}$$

The number of grid points used in this case is

$$a_{n,2} + a_{n-1,2} = 2n + 5 + \left(\frac{3}{2}n - 5\right) 2^n.$$

For Case 2, let

$$b_{n,2} = \sum_{\gamma_1+\gamma_2=n} (2^{\gamma_1} + 1)(2^{\gamma_2} + 1)$$

where the subscript 2 denotes dimension. Then

$$\begin{aligned} b_{n,2} &= \sum_{\gamma_1+\gamma_2=n} (2^{\gamma_1+\gamma_2} + 2^{\gamma_1} + 2^{\gamma_2} + 1) \\ &= \sum_{\gamma_1=0}^{n} (2^n + 2^{\gamma_1} + 2^{n-\gamma_1} + 1) \\ &= (n-1) + (n+5)2^n. \end{aligned}$$

The number of grid points used in this case is

$$b_{n,2} + b_{n-1,2} = 2n - 3 + \left(\frac{3}{2}n + 7\right) 2^n.$$

For Case 3, let

$$c_{n,2} = \sum_{\gamma_1+\gamma_2=n} (2^{\gamma_1+1} - 1)(2^{\gamma_2+1} - 1)$$

where the subscript 2 denotes dimension. Then

$$\begin{aligned} c_{n,2} &= \sum_{\gamma_1+\gamma_2=n} (2^{\gamma_1+\gamma_2+2} - 2^{\gamma_1+1} - 2^{\gamma_2+1} + 1) \\ &= \sum_{\gamma_1=0}^{n} (2^{n+2} - 2^{\gamma_1+1} - 2^{n-\gamma_1+1} + 1) \\ &= (n+5) + (4n-4)2^n. \end{aligned}$$

The number of grid points used in this case is

$$c_{n,2} + c_{n-1,2} = 2n + 9 + (6n - 8)2^n.$$

□

The $a_{n,d}$, $b_{n,d}$ and $c_{n,d}$ can be computed recursively when $d \geq 2$ by

$$\begin{aligned} a_{n,d} &= \sum_{\gamma_d=0}^{n} (2^{\gamma_d} - 1) a_{n-\gamma_d, d-1}, \\ b_{n,d} &= \sum_{\gamma_d=0}^{n} (2^{\gamma_d} + 1) b_{n-\gamma_d, d-1}, \\ c_{n,d} &= \sum_{\gamma_d=0}^{n} (2^{\gamma_d+1} - 1) c_{n-\gamma_d, d-1}. \end{aligned}$$

Therefore, for a specific d, we can exactly compute the total number of grid points used in combination technique. One can find a detailed computation for the case 2 when $d = 3$ in [45] and the method can be generalised to compute all three cases for any d. In fact, we do not really need a very complicated expression for exact number of grid points. We only require an asymptotically growth rate to compare with growth rate in classical sparse grid method. The following lemma motivated by the idea in [76] provides us the asymptotically growth rate.

Lemma 2.3. *The number of grid points used in a d dimensional classical sparse grid combination technique grows as $O(n^{d-1}2^n)$ for all three cases discussed in Lemma* 2.2.

Proof. For Case 1, the following inequalities hold.

$$2^{|\gamma|-d} = \prod_{i=1}^{d} 2^{\gamma_i - 1} \leq \prod_{i=1}^{d} (2^{\gamma_i} - 1) \leq \prod_{i=1}^{d} 2^{\gamma_i} = 2^{|\gamma|}.$$

The inequalities provides us an upper bound and a lower bound for $a_{n,d}$, namely,

$$\binom{n-1}{d-1} 2^{n-d} \leq a_{n,d} \leq \binom{n-1}{d-1} 2^n.$$

Thus, the number of grid points used in this case is approximately

$$\sum_{k=0}^{d-1} \binom{n-k-1}{d-1} 2^{n-k} = O(n^{d-1}\, 2^n).$$

Similar as what we did for Case 1, we have the following inequalities for Case 2 and Case 3

$$\begin{aligned} 2^{|\gamma|} &= \prod_{i=1}^{d} 2^{\gamma_i} \leq \prod_{i=1}^{d} (2^{\gamma_i} + 1) \leq \prod_{i=1}^{d} 2^{\gamma_i+1} = 2^{|\gamma|+d} \\ 2^{|\gamma|} &= \prod_{i=1}^{d} 2^{\gamma_i} \leq \prod_{i=1}^{d} (2^{\gamma_i+1} - 1) \leq \prod_{i=1}^{d} 2^{\gamma_i+1} = 2^{|\gamma|+d}. \end{aligned}$$

The upper and lower bounds for both $b_{n,d}$ and $c_{n,d}$ are

$$\binom{n+d-1}{d-1} 2^n \le b_{n,d},\ c_{n,d} \le \binom{n+d-1}{d-1} 2^{n+d}.$$

Therefore, the number of grid points used in these two cases are approximately

$$\sum_{k=0}^{d-1} \binom{n-k+d-1}{d-1} 2^{n-k} = O(n^{d-1}\, 2^n).$$

□

From Lemma 2.3, we know the total number of grid points used in a classical sparse grid combination technique is similar to that of a classical sparse grid for all the cases discussed above. Therefore, the classical sparse grid combination technique can be viewed as an alternative way to compute sparse grid interpolation/integration.

2.3 Error of the Combination Technique

The classical error analysis [43] of the sparse grid combination technique is based on an error splitting model [43]. The analysis can be applied to both interpolation and integration. Suppose we have the following simple 2D error splitting model on the anisotropic regular grids $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$

$$f - f_\gamma = C_1(x, h_{\gamma_1}) h_{\gamma_1}^p + C_2(x, h_{\gamma_2}) h_{\gamma_2}^p + C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^p h_{\gamma_2}^p \tag{2.5}$$

where h_{γ_k}, $k = 1, 2$ is the spacing for each dimension. We can figure out the error bound of the sparse grid combination technique f_n^c by using the model when the coefficients in the model are bounded. The method can also be generalised to d-dimensional case [76, 43] if we have a d dimensional error splitting model. In this section, we take the (piecewise) Lagrangian interpolation as an example and study the error of the combination technique for the Lagrangian interpolation. We start with deriving different 2D error splitting models based on different smoothness conditions and choices of the grid points. Then we show how to get the error bounds of the combination technique solution for 2D case in details and generalise the analysis to d dimensional case. We make a general assumption that $f \in \bigotimes_{i=1}^d C^p(X_i)$ for some integer $p > 0$ where $X = \prod_{i=1}^d X_i$ is a bounded set in the analysis. In fact, we have

$$C^p_{mix}(X) = \bigotimes_{i=1}^{d} C^p(X_i)$$

where $C^p_{mix}(X)$ is the same as the space $H^p_{mix}(X)$ except we use derivative but not the weak derivative[†].

2.3.1 Error Models For 2D Piecewise Linear Interpolation

The 2D error splitting model is based on the tensor product of 1D Taylor expansion. If we expand the 1D Taylor series at 0 for both coordinates, we get the following 2D expansion.

Lemma 2.4. *Suppose $f \in C^2(X) \otimes C^2(X)$. We have the following expansion*

$$\begin{aligned} f(x_1, x_2) =& f(0,0) + x_1\partial_1 f(0, x_2) + x_2\partial_2 f(x_1, 0) - x_1x_2\partial_1\partial_2 f(0,0) \\ &+ \int_0^{x_1} (x_1 - s_1)\partial_1^2 f(s_1, 0)\, ds_1 + \int_0^{x_2} (x_2 - s_2)\partial_2^2 f(0, s_2)\, ds_2 \\ &+ \int_0^{x_1}\int_0^{x_2} (x_1 - s_1)(x_2 - s_2)\partial_1^2\partial_2^2 f(s_1, s_2)\, ds_1 ds_2. \end{aligned}$$

Proof. We first define the following operators

$$\begin{aligned} I : C^2(X) &\to C^2(X) \\ g &\mapsto g. \end{aligned}$$

$$\begin{aligned} L_1 : C^2(X) &\to C^2(X) \\ g &\mapsto g(0). \end{aligned}$$

$$\begin{aligned} L_2 : C^2(X) &\to C^2(X) \\ g &\mapsto xg'(0). \end{aligned}$$

$$\begin{aligned} E_3 : C^2(X) &\to C^2(X) \\ g &\mapsto \int_0^x (x - s)g^{(2)}(s)\, ds. \end{aligned}$$

By using the notation above, the 1D Taylor expansion with the integral remainder can be written as

$$g = Ig = L_1 g + L_2 g + E_3 g = (L_1 + L_2 + E_3)g.$$

[†] The choice of the space depends on the problem we have.

For 2D case, when $f \in C^2(X^2)$, we have

$$\begin{aligned}
f &= (I \otimes I)f = [(L_1 + L_2 + E_3) \otimes (L_1 + L_2 + E_3)]f \\
&= (L_1 \otimes L_1)f + (L_2 \otimes L_2)f + [L_2 \otimes (L_1 + E_3)]f \\
&+ [(L_1 + E_3) \otimes L_2]f + (L_1 \otimes E_3)f + (E_3 \otimes L_1)f + (E_3 \otimes E_3)f \\
&= (L_1 \otimes L_1)f + (L_2 \otimes L_2)f + [L_2 \otimes (I - L_2)]f \\
&+ [(I - L_2) \otimes L_2]f + (L_1 \otimes E_3)f + (E_3 \otimes L_1)f + (E_3 \otimes E_3)f \\
&= (L_1 \otimes L_1)f + (L_2 \otimes I)f + (I \otimes L_2)f - (L_2 \otimes L_2)f \\
&+ (L_1 \otimes E_3)f + (E_3 \otimes L_1)f + (E_3 \otimes E_3)f.
\end{aligned}$$

Applying the tensor product operators to function f based on the definitions of I, L_1, L_2 and E_3, we get the expansion formula. □

If the function f has higher order smoothness, then we can generalise the Lemma 2.4 to the following Lemma.

Lemma 2.5. *Suppose $f \in C^p(X) \otimes C^p(X)$. We have the following expansion*

$$\begin{aligned}
f(x_1, x_2) &= f(0,0) + \sum_{i=2}^{p} \frac{1}{(i-1)!} x_1^{i-1} \partial_1^{i-1} f(0, x_2) + \sum_{i=2}^{p} \frac{1}{(i-1)!} x_2^{i-1} \partial_2^{i-1} f(x_1, 0) \\
&- \sum_{i=2}^{p} \sum_{j=2}^{p} \frac{1}{(i-1)!(j-1)!} x_1^{i-1} x_2^{j-1} \partial_1^{i-1} \partial_2^{j-1} f(0,0) \\
&+ \frac{1}{p!} \int_0^{x_1} (x_1 - s_1)^{p-1} \partial_1^p f(s_1, 0)\, ds_1 + \frac{1}{p!} \int_0^{x_2} (x_2 - s_2)^{p-1} \partial_2^p f(0, s_2)\, ds_2 \\
&+ \frac{1}{(p!)^2} \int_0^{x_1} \int_0^{x_2} (x_1 - s_1)^{p-1} (x_2 - s_2)^{p-1} \partial_1^p \partial_2^p f(s_1, s_2)\, ds_1 ds_2.
\end{aligned}$$

Proof. We again define the following operators for 1D case.

$$\begin{aligned}
I : C^p(X) &\to C^p(X) \\
g &\mapsto g.
\end{aligned}$$

For $i = 1, \dots, p$, we define

$$\begin{aligned}
L_i : C^p(X) &\to C^p(X) \\
g &\mapsto \frac{1}{(i-1)!} x^{i-1} g^{(i-1)}(0).
\end{aligned}$$

For the integral remainder operator

$$\begin{aligned}
E_{p+1} : C^p(X) &\to C^p(X) \\
g &\mapsto \frac{1}{p!} \int_0^x (x-s)^{p-1} g^{(p)}(s)\, ds.
\end{aligned}$$

Then the 1D Taylor expansion with the integral remainder can be expressed as

$$g = Ig = (\sum_{i=1}^{p} L_i + E_{p+1})g.$$

For the 2D case, we have

$$\begin{aligned} f = (I \otimes I)f &= [(\sum_{i=1}^{p} L_i + E_{p+1}) \otimes (\sum_{i=1}^{p} L_i + E_{p+1})]f \\ &= (L_1 \otimes L_1)f + \sum_{i=2}^{p}(L_i \otimes I)f + \sum_{i=2}^{p}(I \otimes L_i)f \\ &- \sum_{i=2}^{p}\sum_{j=2}^{p}(L_i \otimes L_j)f + (E_{p+1} \otimes L_1)f + (L_1 \otimes E_{p+1})f \\ &+ (E_{p+1} \otimes E_{p+1})f. \end{aligned} \tag{2.6}$$

The inclusion-exclusion principle is used in the computation of (2.6). Finally, we use the defined operators to (2.6) and obtain the desired result. □

Next, we derive the 2D error splitting model for the piecewise linear interpolant on equally discretised grid based on the expansions in the Lemma 2.4. We first focus on the simplest case when $f \in C^2(X) \otimes C^2(X)$, $X = [0,1]$. The proof here is adapted from the proof in [76].

Theorem 2.6. *Suppose $f \in C^2(X) \otimes C^2(X)$ and $X = [0,1]$. $\mathcal{K}_\gamma f$ is the 2D piecewise linear interpolant of f on equally discretised grid G_γ. Then we have the following error splitting model*

$$f(x) - \mathcal{K}_\gamma f(x) = C_1(x, h_{\gamma_1})h_{\gamma_1}^2 + C_2(x, h_{\gamma_2})h_{\gamma_2}^2 + C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^2 h_{\gamma_2}^2$$

where

$$\begin{aligned} \|C_1(x, h_{\gamma_1})\|_\infty &\le \frac{1}{4}\|\partial_1^2 f\|_\infty, \\ \|C_2(x, h_{\gamma_2})\|_\infty &\le \frac{1}{4}\|\partial_2^2 f\|_\infty, \\ \|C_{1,1}(x, h_{\gamma_1}, h_{\gamma_2})\|_\infty &\le \left(\frac{1}{4}\right)^2 \|\partial_1^2\partial_2^2 f\|_\infty. \end{aligned}$$

Proof. Without loss of generality, we consider a point $x = (x_1, x_2)$ located in the block $[0, h_{\gamma_1}] \times [0, h_{\gamma_2}]$. The four corner points are denoted as

$$\begin{aligned} (p_1^0, p_2^0) &= (0,0) \\ (p_1^1, p_2^0) &= (h_{\gamma_1}, 0) \\ (p_1^0, p_2^1) &= (0, h_{\gamma_2}) \\ (p_1^1, p_2^1) &= (h_{\gamma_1}, h_{\gamma_2}). \end{aligned}$$

The two 1D linear basis functions on interval $[0,h]$ are

$$\begin{aligned} b_0(x) &= \frac{1}{h}x \\ b_1(x) &= 1 - \frac{1}{h}x. \end{aligned}$$

The 2D linear basis function is the tensor product of 1D linear basis functions. Therefore, the four 2D linear basis functions in the block $[0,h_{\gamma_1}] \times [0,h_{\gamma_2}]$ are

$$\begin{aligned} b_{0,0}(x_1,x_2) &= \frac{1}{h_{\gamma_1}}x_1\frac{1}{h_{\gamma_2}}x_2 \\ b_{0,1}(x_1,x_2) &= \frac{1}{h_{\gamma_1}}x_1(1-\frac{1}{h_{\gamma_2}})x_2 \\ b_{1,0}(x_1,x_2) &= (1-\frac{1}{h_{\gamma_1}})x_1\frac{1}{h_{\gamma_2}}x_2 \\ b_{1,1}(x_1,x_2) &= (1-\frac{1}{h_{\gamma_1}})x_1(1-\frac{1}{h_{\gamma_2}})x_2. \end{aligned}$$

Using these basis functions, we can get the following four points linear interpolant on the block

$$L_{(2,2)}f(x_1,x_2) = \sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i - (h_{\gamma_1} - x_1)|}{h_{\gamma_1}}\frac{|p_2^j - (h_{\gamma_2} - x_2)|}{h_{\gamma_2}}f(p_1^i,p_2^j).$$

We apply the Lemma 2.4 to $f(p_1^i,p_2^j)$(expand at point (x_1,x_2)). We can get

$$\begin{aligned} L_{2,2}f(x_1.x_2) = &\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i - (h_{\gamma_1} - x_1)|}{h_{\gamma_1}}\frac{|p_2^j - (h_{\gamma_2} - x_2)|}{h_{\gamma_2}}\Bigg[\\ &f(x_1,x_2) + (p_1^i - x_1)\partial_1 f(x_1,p_2^j) + (p_2^j - x_2)\partial_2 f(p_1^i,x_2) \\ &- (p_1^i - x_1)(p_2^j - x_2)\partial_1\partial_2 f(x_1,x_2) \\ &+ \int_0^{p_1^i}(p_1^i - s_1)\partial_1^2 f(s_1,x_2)\,ds_1 + \int_0^{p_2^j}(p_2^j - s_2)\partial_2^2 f(x_1,s_2)\,ds_2 \\ &+ \int_0^{p_1^i}\int_0^{p_2^j}(p_1^i - s_1)(p_2^j - s_2)\partial_1^2\partial_2^2 f(s_1,s_2)\,ds_1 ds_2\Bigg]. \end{aligned}$$

If we notice that

$$\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i-(h_{\gamma_1}-x_1)|}{h_{\gamma_1}}\frac{|p_2^j-(h_{\gamma_2}-x_2)|}{h_{\gamma_2}}=1$$
$$\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i-(h_{\gamma_1}-x_1)|}{h_{\gamma_1}}\frac{|p_2^j-(h_{\gamma_2}-x_2)|}{h_{\gamma_2}}(p_1^i-x_1)=0$$
$$\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i-(h_{\gamma_1}-x_1)|}{h_{\gamma_1}}\frac{|p_2^j-(h_{\gamma_2}-x_2)|}{h_{\gamma_2}}(p_2^i-x_2)=0$$
$$\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i-(h_{\gamma_1}-x_1)|}{h_{\gamma_1}}\frac{|p_2^j-(h_{\gamma_2}-x_2)|}{h_{\gamma_2}}(p_1^i-x_1)(p_2^j-x_2)=0,$$

We can obtain the following error formula for the linear interpolant

$$\begin{aligned}&L_{2,2}f(x_1,x_2)-f(x_1,x_2)\\=&\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i-(h_{\gamma_1}-x_1)|}{h_{\gamma_1}}\frac{|p_2^j-(h_{\gamma_2}-x_2)|}{h_{\gamma_2}}\Bigg[\\&\int_0^{p_1^i}(p_1^i-s_1)\partial_1^2f(s_1,x_2)\,ds_1+\int_0^{p_2^j}(p_2^j-s_2)\partial_2^2f(x_1,s_2)\,ds_2\\&+\int_0^{p_1^i}\int_0^{p_2^j}(p_1^i-s_1)(p_2^j-s_2)\partial_1^2\partial_2^2f(s_1,s_2)\,ds_1ds_2\Bigg].\end{aligned}$$

We further denote

$$\begin{aligned}C_1(x,h_{\gamma_1})=&\frac{1}{h_{\gamma_1}^2}\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i-(h_{\gamma_1}-x_1)|}{h_{\gamma_1}}\frac{|p_2^j-(h_{\gamma_2}-x_2)|}{h_{\gamma_2}}\\&\int_0^{p_1^i}(p_1^i-s_1)\partial_1^2f(s_1,x_2)\,ds_1\end{aligned}$$

$$\begin{aligned}C_2(x,h_{\gamma_2})=&\frac{1}{h_{\gamma_2}^2}\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i-(h_{\gamma_1}-x_1)|}{h_{\gamma_1}}\frac{|p_2^j-(h_{\gamma_2}-x_2)|}{h_{\gamma_2}}\\&\int_0^{p_2^j}(p_2^j-s_1)\partial_1^2f(x_1,s_2)\,ds_2\end{aligned}$$

$$\begin{aligned}C_{1,2}(x,h_{\gamma_1},h_{\gamma_2})=&\frac{1}{h_{\gamma_1}^2h_{\gamma_2}^2}\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i-(h_{\gamma_1}-x_1)|}{h_{\gamma_1}}\frac{|p_2^j-(h_{\gamma_2}-x_2)|}{h_{\gamma_2}}\\&\int_0^{p_1^i}\int_0^{p_2^j}(p_1^i-s_1)(p_2^j-s_2)\partial_1^2\partial_2^2f(s_1,s_2)\,ds_1ds_2.\end{aligned}$$

Let $M_1 = \|\partial_1^2 f(x_1, x_2)\|_\infty$, we can obtain a bound for the coefficient $C_1(x, h_{\gamma_1})$.

$$\begin{aligned}
&\|C_1(x, h_{\gamma_1})\|_\infty \\
\leq&\left|\frac{M_1}{h_{\gamma_1}^2}\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i - (h_{\gamma_1} - x_1)|}{h_{\gamma_1}}\frac{|p_2^j - (h_{\gamma_2} - x_2)|}{h_{\gamma_2}}\int_0^{p_1^i}(p_1^i - s_1)\,ds_1\right| \\
=&\left|\frac{M_1}{2h_{\gamma_1}^2}\sum_{i=0}^{1}\sum_{j=0}^{1}\frac{|p_1^i - (h_{\gamma_1} - x_1)|}{h_{\gamma_1}}\frac{|p_2^j - (h_{\gamma_2} - x_2)|}{h_{\gamma_2}}(p_1^i - s_1)^2\right| \\
=&\frac{M_1}{2}\left|\left(\frac{x_1}{h_{\gamma_1}}\right)^2\left(1 - \frac{x_1}{h_{\gamma_1}}\right) + \left(\frac{x_1}{h_{\gamma_1}}\right)\left(1 - \frac{x_1}{h_{\gamma_1}}\right)^2\right| \\
\leq&\frac{M_1}{2}\left|\frac{x_1}{h_{\gamma_1}}\left(1 - \frac{x_1}{h_{\gamma_1}}\right)\right| \\
\leq&\frac{1}{4}M_1.
\end{aligned}$$

The coefficients $C_2(x, h_{\gamma_2})$ and $C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})$ can be similarly bounded by the infinite norm of the corresponding derivatives. □

If we consider more general domain $X_1 \times X_2$, then we get the following revised Theorem.

Theorem 2.7. *Suppose $f \in C^2(X_1) \otimes C^2(X_2)$ and $X_1 = [a_1, b_1]$, $X_2 = [a_2, b_2]$. $\mathcal{K}_\gamma f$ is the 2D piecewise linear interpolant of f on equally discretised grid G_γ. The grid $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ where G_{γ_k} consists of the following grid points*

$$a_k \leq x_k^0 \leq x_k^1 \leq \cdots \leq x_k^{n_{\gamma_k}} \leq b_k$$

where $k = 1, 2$. Let

$$\begin{aligned}
\tilde{h}_{\gamma_1} &= \frac{b_1 - a_1}{n_{\gamma_1}} \\
\tilde{h}_{\gamma_2} &= \frac{b_2 - a_2}{n_{\gamma_2}}.
\end{aligned}$$

Then we have the following error splitting model

$$f(x) - \mathcal{K}_\gamma f(x) = C_1(x, G_{\gamma_1})\tilde{h}_{\gamma_1}^2 + C_2(x, G_{\gamma_2})\tilde{h}_{\gamma_2}^2 + C_{1,2}(x, G_{\gamma_1}, G_{\gamma_2})\tilde{h}_{\gamma_1}^2\tilde{h}_{\gamma_2}^2$$

where

$$\begin{aligned}
\|C_1(x, G_{\gamma_1})\|_\infty &\leq \frac{1}{4}\|\partial_1^2 f\|_\infty, \\
\|C_2(x, G_{\gamma_2})\|_\infty &\leq \frac{1}{4}\|\partial_2^2 f\|_\infty, \\
\|C_{1,1}(x, G_{\gamma_1}, G_{\gamma_2})\|_\infty &\leq \left(\frac{1}{4}\right)^2\|\partial_1^2\partial_2^2 f\|_\infty.
\end{aligned}$$

Further, if the grid G_γ is not equally spaced, we have the following result.

Theorem 2.8. *Suppose $f \in C^2(X_1) \otimes C^2(X_2)$ and $X_1 = [a_1, b_1]$, $X_2 = [a_2, b_2]$. $\mathcal{K}_\gamma f$ is the 2D piecewise linear interpolant of f on grid G_γ. G_γ is not necessarily to be equally spaced. The grid $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ where G_{γ_k} consists of the following grid points*

$$a_k \le x_k^0 \le x_k^1 \le \cdots \le x_k^{n_{\gamma_k}} \le b_k$$

where $k = 1, 2$. Let

$$\begin{aligned}\tilde{h}_{\gamma_1} &= \max_{i=1,\dots,n_{\gamma_k}} (x_1^i - x_1^{i-1})\\ \tilde{h}_{\gamma_2} &= \max_{i=1,\dots,n_{\gamma_k}} (x_2^i - x_2^{i-1}).\end{aligned}$$

Then we have the following error splitting model

$$f(x) - \mathcal{K}_\gamma f(x) = C_1(x, G_{\gamma_1})\tilde{h}_{\gamma_1}^2 + C_2(x, G_{\gamma_2})\tilde{h}_{\gamma_2}^2 + C_{1,2}(x, G_{\gamma_1}, G_{\gamma_2})\tilde{h}_{\gamma_1}^2\tilde{h}_{\gamma_2}^2$$

where

$$\begin{aligned}\|C_1(x, G_{\gamma_1})\|_\infty &\le \frac{1}{4}\|\partial_1^2 f\|_\infty,\\ \|C_2(x, G_{\gamma_2})\|_\infty &\le \frac{1}{4}\|\partial_2^2 f\|_\infty,\\ \|C_{1,1}(x, G_{\gamma_1}, G_{\gamma_2})\|_\infty &\le \left(\frac{1}{4}\right)^2 \|\partial_1^2\partial_2^2 f\|_\infty.\end{aligned}$$

2.3.2 Error Models for Polynomial Interpolation

Next, we consider the case when f has higher order smoothness and use the polynomial interpolant to approximate f. We first study the case when f is discretised on the equally spaced grid G_γ. In order to obtain the error splitting model, we need the following results on 1D Lagrangian interpolation and Lagrangian basis functions.

Theorem 2.9 ([25])**.** *If $f \in C^{n+1}(X)$, $X \subset \mathbb{R}$ is bounded, for any $x \in X$, the remainder*

$$R_n(x) = f(x) - L_n f(x) = \frac{f^{(n+1)}(\theta)}{(n+1)!} w_{n+1}(x).$$

Here $\theta \in X$ and it depends on the choice of x. $w_{n+1}(x)$ is defined as

$$w_{n+1}(x) = (x - x_0)\dots(x - x_n).$$

Corollary 2.10. *Suppose $l_i(x)$, $i = 0, \dots, n$ are the Lagrangian basis function and x_i, $i = 0, \dots, n$ are grid points. We have*

$$\sum_{i=0}^{n} x_i^k l_i(x) = x^k, \ k = 0, \dots, n.$$

Proof. Taking $f(x) = x^k$, $k = 0, \ldots, n$ in the above theorem, we have

$$R_n(x) = x^k - \sum_{i=0}^{n} x_i^k l_i(x) = 0.$$

□

Lemma 2.11. *The following equation holds for any $n \in \mathbb{N}$*

$$\sum_{i=0}^{n} \binom{n}{i} (-1)^i = 0. \tag{2.7}$$

Proof. Using the binomial theorem, we have

$$(-1 + x)^n = \sum_{i=0}^{n} \binom{n}{i} (-1)^i x^{n-i}.$$

We get the equation (2.7) when we take $x = 1$. □

Lemma 2.12. *Suppose $l_i(x)$, $i = 0, \ldots, n$ are the Lagrangian basis function and x_i, $i = 0, \ldots, n$ are grid points. We have*

$$\sum_{i=0}^{n} l_i(x)(x_i - x)^k = 0, \ k = 1, \ldots, n.$$

Proof. For any $k = 1, \ldots, n.$, we have

$$\begin{aligned}
\sum_{i=0}^{n} l_i(x)(x_i - x)^k &= \sum_{i=0}^{n} l_i(x) \sum_{j=0}^{n} \binom{n}{j} x_i^j (-x)^{n-j} \\
&= \sum_{j=0}^{n} \binom{n}{j} (-x)^{n-j} \sum_{i=0}^{n} x_i^j l_i(x) \\
&= \sum_{j=0}^{n} \binom{n}{j} (-1)^{n-j} x^n \\
&= (-1)^n \sum_{j=0}^{n} \binom{n}{j} (-1)^j x^n \\
&= 0.
\end{aligned}$$

In the first equation, we use the binomial theorem. We exchange the order of summation in the second equation. For the third equation, we apply the corollary 2.10. Finally, we simplify the equation, use the Lemma 2.11 and obtain the desired result. □

Theorem 2.13. *If $f \in C^{n+1}(X)$, $X \subset \mathbb{R}$ is bounded, $L_n f$ is the $n+1$ points Lagrangian interpolant of f, then*

$$R_n(x) = f(x) - L_n f(x) = -\frac{1}{n!}\sum_{i=0}^{n} l_i(x) \int_x^{x_i} (x_i - s)^n f^{(n+1)}(s)\, ds.$$

where x_i, $i = 0, \dots, n$ are grid points and $l_i(x)$, $i = 0, \dots, n$ are the corresponding Lagrangian basis functions.

Proof.

$$\begin{aligned} R_n(x) &= f(x) - L_n f(x) \\ &= \sum_{i=0}^{n} f(x) l_i(x) - \sum_{i=0}^{n} f(x_i) l_i(x) \\ &= -\sum_{i=0}^{n} (f(x_i) - f(x)) l_i(x) \\ &= -\sum_{i=0}^{n} l_i(x) \Bigg[f^{(1)}(x)(x_i - x) + \frac{1}{2!} f^{(2)}(x_i - x)^2 + \\ &\cdots + \frac{1}{n!} f^{(n)}(x)(x_i - x)^n + \frac{1}{n!} \int_x^{x_i} (x_i - s)^n f^{(n+1)}(s)\, ds \Bigg] \end{aligned}$$

Using Lemma 2.12, we have

$$\sum_{i=0}^{n} \sum_{j=1}^{n} \frac{1}{j!} f^{(j)}(x) l_i(x) (x_i - x)^j = 0$$

and we get

$$R_n(x) = -\frac{1}{n!}\sum_{i=0}^{n} l_i(x) \int_x^{x_i} (x_i - s)^n f^{(n+1)}(s)\, ds.$$

□

Corollary 2.14. *If $f \in C^{n+1}(X)$, $X \subset \mathbb{R}$ is bounded, then for any $x \in X$, there exists a $\theta \in X$ such that*

$$\frac{f^{(n+1)}(\theta)}{(n+1)!} w_{n+1}(x) = -\frac{1}{n!}\sum_{i=0}^{n} l_i(x) \int_x^{x_i} (x_i - s)^n f^{(n+1)}(s)\, ds.$$

The choice of θ depends on the choice of x.

Proof. Combined the results in Theorem 2.9 and Theorem 2.13, we get the desired result. □

Theorem 2.15. *Suppose $f \in C^{n_{\gamma_1}+1}(X) \otimes C^{n_{\gamma_2}+1}(X)$ and $X = [0,1] \subset \mathbb{R}$. $L_{n_\gamma} f$ is the 2D polynomial interpolant of f on the equally spaced grid G_γ. Let h_{γ_1} and h_{γ_2} be the spacings with respect to different dimensions.Then we have the following result*

$$\begin{aligned} f(x) - L_{n_\gamma} f = C_1(x, h_{\gamma_1}) h_{\gamma_1}^{n_{\gamma_1}+1} &+ C_2(x, h_{\gamma_2}) h_{\gamma_2}^{n_{\gamma_2}+1} \\ &+ C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^{n_{\gamma_1}+1} h_{\gamma_2}^{n_{\gamma_2}+1} \end{aligned}$$

and

$$\begin{aligned} \|C_1(x, h_{\gamma_1})\|_\infty &\leq \|\partial_1^{n_{\gamma_1}+1} f\|_\infty, \\ \|C_2(x, h_{\gamma_2})\|_\infty &\leq \|\partial_2^{n_{\gamma_2}+1} f\|_\infty, \\ \|C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})\|_\infty &\leq \|\partial_1^{n_{\gamma_1}+1} \partial_2^{n_{\gamma_2}+1} f\|_\infty. \end{aligned}$$

Proof. Without loss of generality, we consider a point $x = (x_1, x_2)$ located in the block $[0,1] \times [0,1]$. For each dimension, the interval $[0,1]$ is equally discretised by

$$0 = x_k^0 \leq x_k^1 \leq \cdots \leq x_k^{n_\gamma} = 1, \; k = 1, 2.$$

The spacing is $h_{n_{\gamma_i}} = n_{\gamma_i}^{-1}$, $k = 1, 2$. Then the 2D lagrangian basis functions are

$$l_{i,j}(x_1, x_2) = l_i^1(x_1) l_j^2(x_2), \; i = 0, \ldots, n_{\gamma_1}, \; j = 0, \ldots, n_{\gamma_2}$$

where

$$\begin{aligned} l_i^1(x_1) &= \frac{\prod_{s \neq i}(x_1^s - x_1)}{\prod_{s \neq i}(p_1^s - p_1^i)} \\ l_i^2(x_2) &= \frac{\prod_{s \neq i}(x_2^s - x_2)}{\prod_{s \neq i}(p_2^s - p_2^i)}. \end{aligned}$$

The 2D interpolant is

$$L_{n_{\gamma_1}, n_{\gamma_2}} f(x_1, x_2) = \sum_{i=0}^{n_{\gamma_1}} \sum_{j=0}^{n_{\gamma_2}} l_{i,j}(x_1, x_2) f(x_1^i, x_2^j).$$

We consider the error

$$L_{n_{\gamma_1}, n_{\gamma_2}} f(x_1, x_2) - f(x_1, x_2) = \sum_{i=0}^{n_{\gamma_1}} \sum_{j=0}^{n_{\gamma_2}} l_{i,j}(x_1, x_2)(f(x_1^i, x_2^j) - f(x_1, x_2)).$$

Using Lemma 2.5, we have the following expansion

$$
\begin{aligned}
f(x_1^i, x_2^j) = {} & f(x_1, x_2) + \sum_{l=2}^{n_{\gamma_1}+1} \frac{1}{(l-1)!}(x_1^i - x_1)^{l-1}\partial_1^{l-1} f(x_1, x_2^j) \\
& + \sum_{m=2}^{n_{\gamma_2}+1} \frac{1}{(m-1)!}(x_2^j - x_2)^{m-1}\partial_2^{m-1} f(x_2^i, x_2) \\
& - \sum_{l=2}^{n_{\gamma_1}+1} \sum_{m=2}^{n_{\gamma_2}+1} \frac{1}{(l-1)!}\frac{1}{(m-1)!}(x_1^i - x_1)^{l-1}(x_2^j - x_2)^{m-1}\partial_1^{l-1}\partial_2^{m-1} f(0,0) \\
& + \frac{1}{(n_{\gamma_1})!}\int_{x_1}^{x_1^i} (x_1^i - s_1)^{n_{\gamma_1}} \partial_1^{n_{\gamma_1}+1} f(s_1, x_2)\, ds_1 \\
& + \frac{1}{(n_{\gamma_2})!}\int_{x_2}^{x_2^j} (x_2^j - s_2)^{n_{\gamma_2}} \partial_2^{n_{\gamma_2}+1} f(x_1, s_2)\, ds_2 \\
& + \frac{1}{(n_{\gamma_1})!}\frac{1}{(n_{\gamma_2})!}\int_{x_1}^{x_1^i}\int_{x_2}^{x_2^j} (x_1^i - s_1)^{n_{\gamma_1}}(x_2^j - s_2)^{n_{\gamma_2}} \partial_1^{n_{\gamma_1}+1}\partial_2^{n_{\gamma_2}+1} f(s_1, s_2)\, ds_1 ds_2.
\end{aligned}
$$

We next show that

$$
\begin{aligned}
& \sum_{i=0}^{n_{\gamma_1}} \sum_{j=0}^{n_{\gamma_2}} l_{i,j}(x_1, x_2) \sum_{l=2}^{n_{\gamma_1}+1} \frac{1}{(l-1)!}(x_1^i - x_1)^{l-1}\partial_1^{l-1} f(x_1, x_2^j) = 0 \\
& \sum_{i=0}^{n_{\gamma_1}} \sum_{j=0}^{n_{\gamma_2}} l_{i,j}(x_1, x_2) \sum_{m=2}^{n_{\gamma_2}+1} \frac{1}{(m-1)!}(x_2^j - x_2)^{m-1}\partial_2^{m-1} f(x_2^i, x_2) = 0 \\
& \sum_{i=0}^{n_{\gamma_1}} \sum_{j=0}^{n_{\gamma_2}} l_{i,j}(x_1, x_2) \sum_{l=2}^{n_{\gamma_1}+1} \sum_{m=2}^{n_{\gamma_2}+1} \frac{1}{(l-1)!}\frac{1}{(m-1)!}(x_1^i - x_1)^{l-1}(x_2^j - x_2)^{m-1}\partial_1^{l-1}\partial_2^{m-1} f(0,0) = 0.
\end{aligned}
$$

Without loss of generality, we only prove the first equation. By using the Lemma 2.12, we have

$$
\begin{aligned}
& \sum_{i=0}^{n_{\gamma_1}} \sum_{j=0}^{n_{\gamma_2}} l_{i,j}(x_1, x_2) \sum_{l=2}^{n_{\gamma_1}+1} \frac{1}{(l-1)!}(x_1^i - x_1)^{l-1}\partial_1^{l-1} f(x_1, x_2^j) \\
= & \sum_{l=2}^{n_{\gamma_1}+1} \sum_{j=0}^{n_{\gamma_2}} \frac{1}{(l-1)!} l_2^j(x_2)\partial_1^{l-1} f(x_1, x_2^j) \sum_{i=0}^{n_{\gamma_1}} l_1^i(x_1)(x_1^i - x_1)^{l-1} \\
= & 0.
\end{aligned}
$$

Finally, we define

$$\begin{aligned}
C_1(x, h_{\gamma_1}) &= \frac{1}{n_{\gamma_1}! h_{\gamma_1}^{n_{\gamma_1}+1}} \int_{x_1}^{x_1^i} (x_1^i - s_1)^{n_{\gamma_1}} \partial_1^{n_{\gamma_1}+1} f(s_1, x_2)\, ds_1 \\
C_2(x, h_{\gamma_2}) &= \frac{1}{n_{\gamma_2}! h_{\gamma_2}^{n_{\gamma_2}+1}} \int_{x_2}^{x_2^j} (x_2^j - s_2)^{n_{\gamma_2}} \partial_2^{n_{\gamma_2}+1} f(x_1, s_2)\, ds_2 \\
C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) &= \frac{1}{n_{\gamma_1}! h_{\gamma_1}^{n_{\gamma_1}+1}} \frac{1}{n_{\gamma_2}! h_{\gamma_2}^{n_{\gamma_2}+1}} \\
&\quad \int_{x_1}^{x_1^i} \int_{x_2}^{x_2^j} (x_1^i - s_1)^{n_{\gamma_1}} (x_2^j - s_2)^{n_{\gamma_2}} \partial_1^{n_{\gamma_1}+1} \partial_2^{n_{\gamma_2}+1} f(s_1, s_2)\, ds_1 ds_2
\end{aligned}$$

and prove they are bounded. Without loss of generality, we only prove for $C_1(x, h_{\gamma_1})$. Applying the Corollary 2.14 to $f(\cdot, x_2)$, we have

$$\begin{aligned}
C_1(x, h_{\gamma_1}) &= \frac{1}{(n_{\gamma_1}+1)! h_{\gamma_1}^{n_{\gamma_1}+1}} \partial_1^{n_{\gamma_1}+1} f(\theta_1, x_2) w_{n_{\gamma_1}+1}(x_1) \\
&= \frac{1}{(n_{\gamma_1}+1)!} \partial_1^{n_{\gamma_1}+1} f(\theta_1, x_2) (\frac{x_1}{h_{\gamma_1}})(\frac{x_1}{h_{\gamma_1}} - 1) \ldots (\frac{x_1}{h_{\gamma_1}} - n_{\gamma_1})
\end{aligned}$$

where $\theta_1 \in X$. Since $f \in C^{n_{\gamma_1}+1}(X) \otimes C^{n_{\gamma_1}+1}(X)$ and X is bounded, we have for any $x_2 \in X$

$$|\partial_1^{n_{\gamma_1}+1} f(\theta_1, x_2)| \leq \|\partial_1^{n_{\gamma_1}+1} f\|_\infty.$$

We now only need to show for $t \in (0, n_{\gamma_1})$

$$\left| \frac{1}{(n_{\gamma_1}+1)!} t(t-1)\ldots(t-n_{\gamma_1}) \right| \leq 1.$$

We use mathematical induction to prove this result. For $n_{\gamma_1} = 1$, we have

$$\left| \frac{1}{2!} t(t-1) \right| \leq \frac{1}{8} \leq 1.$$

Suppose we have

$$\left| \frac{1}{(k+1)!} t(t-1)\ldots(t-k) \right| \leq 1.$$

Then

$$\left| \frac{1}{(k+2)!} t(t-1)\ldots(t-k)(t-(k+1)) \right| \leq \frac{|t-k|}{k+2} \leq 1.$$

□

Using a similar method, we can also achieve the following result in the space $H^{n_{\gamma_1}+1}(X) \otimes H^{n_{\gamma_2}+1}(X)$.

Theorem 2.16. *Suppose $f \in H^{n_{\gamma_1}+1}(X) \otimes H^{n_{\gamma_2}+1}(X)$ and $X = [0,1] \subset \mathbb{R}$. $L_{n_\gamma} f$ is the 2D polynomial interpolant of f on the equally spaced grid G_γ. Then we have the following result*

$$f(x) - L_{n_\gamma} f = C_1(x, h_{\gamma_1}) h_{\gamma_1}^{n_\gamma+1} + C_2(x, h_{\gamma_2}) h_{\gamma_2}^{n_{\gamma_2}+1} + C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^{n_{\gamma_1}+1} h_{\gamma_2}^{n_{\gamma_2}+1}$$

and

$$\begin{aligned} \|C_1(x, h_{\gamma_1})\|_2 &\le \|\partial_1^{n_{\gamma_1}+1} f\|_2, \\ \|C_2(x, h_{\gamma_2})\|_2 &\le \|\partial_2^{n_{\gamma_2}+1} f\|_2, \\ \|C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})\|_2 &\le \|\partial_1^{n_{\gamma_1}+1} \partial_2^{n_{\gamma_2}+1} f\|_2. \end{aligned}$$

As shown in the Chapter 1, some interpolants are built on the unequally spaced grid. For this case, we have the following Theorem.

Theorem 2.17. *Suppose $f \in C^{n_{\gamma_1}+1}(X_1) \otimes C^{n_{\gamma_2}+1}(X_2)$ and $X_1 = [a_1, b_1]$, $X_2 = [a_2, b_2]$. The function f is approximated on the grid $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ where G_{γ_k} consists of the following grid points*

$$a_k \le x_k^0 \le x_k^1 \le \cdots \le x_k^{n_{\gamma_k}} \le b_k$$

where $k = 1, 2$.

$L_{n_\gamma} f$ is the 2D polynomial interpolant of f on the grid G_γ. G_{γ_k}, $k = 1, 2$ are not necessary to be equally spaced grid. Let

$$\begin{aligned} \tilde{h}_{\gamma_1} &= \frac{b_1 - a_1}{n_{\gamma_1}} \\ \tilde{h}_{\gamma_2} &= \frac{b_2 - a_2}{n_{\gamma_2}}. \end{aligned}$$

We further define

$$\tilde{w}_{n_{\gamma_k}+1}(x_k) = \frac{1}{(n_{\gamma_k}+1)!\tilde{h}_{\gamma_k}^{n_{\gamma_k}+1}} w_{n_{\gamma_k}+1}(x_k)$$

where

$$w_{n_{\gamma_k}+1}(x_k) = (x_k - x_k^0) \ldots (x_k - x_k^{n_{\gamma_k}}).$$

$k = 1, 2$. If there exists $M_1 > 0$, $M_2 > 0$ such that

$$|\tilde{w}_{n+1}^k(x_k)| \le M_k, \; k = 1, 2,$$

then we have the following result

$$f(x) - L_{n_\gamma} f = C_1(x, G_{\gamma_1}) \tilde{h}_{\gamma_1}^{n_{\gamma_1}+1} + C_2(x, G_{\gamma_2}) \tilde{h}_{\gamma_2}^{n_{\gamma_2}+1} + C_{1,2}(x, G_{\gamma_1}, G_{\gamma_2}) \tilde{h}_{\gamma_1}^{n_{\gamma_1}+1} \tilde{h}_{\gamma_2}^{n_{\gamma_2}+1}$$

and

$$\begin{aligned}
&\|C_1(x, G_{\gamma_1})\|_\infty \le M_1 \|\partial_1^{n_{\gamma_1}+1} f\|_\infty,\\
&\|C_2(x, G_{\gamma_2})\|_\infty \le M_2 \|\partial_2^{n_{\gamma_2}+1} f\|_\infty,\\
&\|C_{1,2}(x, G_{\gamma_1}, G_{\gamma_2})\|_\infty \le M_1 M_2 \|\partial_1^{n_{\gamma_1}+1}\partial_2^{n_{\gamma_2}+1} f\|_\infty.
\end{aligned}$$

Proof. Comparing with the result in Theorem 2.15, we only need to give the definitions of the coefficients $C_1(x, G_{\gamma_1})$, $C_2(x, G_{\gamma_2})$, $C_{1,2}(x, G_{\gamma_1}, G_{\gamma_2})$ and show they are bounded. We define

$$\begin{aligned}
C_1(x, G_{\gamma_1}) &= \sum_{i=0}^{n_{\gamma_1}} l_i(x) \frac{1}{n_{\gamma_1}! \tilde{h}_{\gamma_1}^{n_{\gamma_1}+1}} \int_{x_1}^{x_1^i} (x_1^i - s_1)^{n_{\gamma_1}} \partial_1^{n_{\gamma_1}+1} f(s_1, x_2)\, ds_1\\
C_2(x, G_{\gamma_2}) &= \sum_{j=0}^{n_{\gamma_2}} l_j(x) \frac{1}{n_{\gamma_2}! \tilde{h}_{\gamma_2}^{n_{\gamma_2}+1}} \int_{x_2}^{x_2^j} (x_2^j - s_2)^{n_{\gamma_2}} \partial_2^{n_{\gamma_2}+1} f(x_1, s_2)\, ds_2\\
C_{1,2}(x, G_{\gamma_1}, G_{\gamma_2}) &= \sum_{i=0}^{n_{\gamma_1}} \sum_{j=0}^{n_{\gamma_2}} l_i(x) l_j(x) \frac{1}{n_{\gamma_1}! \tilde{h}_{\gamma_1}^{n_{\gamma_1}+1}} \frac{1}{n_{\gamma_2}! \tilde{h}_{\gamma_2}^{n_{\gamma_2}+1}}\\
&\int_{x_1}^{x_1^i} \int_{x_2}^{x_2^j} (x_1^i - s_1)^{n_{\gamma_1}} (x_2^j - s_2)^{n_{\gamma_2}} \partial_1^{n_{\gamma_1}+1} \partial_2^{n_{\gamma_2}+1} f(s_1, s_2)\, ds_1 ds_2
\end{aligned}$$

Without loss of generality, we again only prove $C_1(x, G_{\gamma_1})$ is bounded. Applying the Corollary 2.14 to $f(\cdot, x_2)$, we have

$$\begin{aligned}
C_1(x, G_{\gamma_1}) &= \frac{1}{(n_{\gamma_1}+1)! \tilde{h}_{\gamma_1}^{n_{\gamma_1}+1}} \partial_1^{n_{\gamma_1}+1} f(\theta_1, x_2) w_{n_{\gamma_1}+1}(x_1)\\
&= \partial_1^{n_{\gamma_1}+1} f(\theta_1, x_2) \tilde{w}_{n_{\gamma_k}+1}(x_1).
\end{aligned}$$

where $\theta_1 \in X_1$. Since $f \in C^{n_{\gamma_1}+1}(X_1) \times C^{n_{\gamma_1}+1}(X_2)$ and X_1 is bounded, we have for any $x_2 \in X_2$

$$|\partial_1^{n_{\gamma_1}+1} f(\theta_1, x_2)| \le \|\partial_1^{n_{\gamma_1}+1} f\|_\infty.$$

Using the assumption that $\tilde{w}_{n+1}^k(x_k)$, $k = 1, 2$ are bounded, we obtain the desired result. □

For general choices of the unequally spaced grid G_γ, the condition

$$\tilde{w}_{n_{\gamma_k}+1}^k(x_k), k = 1, 2$$

is not necessarily to be bounded. However, for some specific grid points, we can prove the condition holds. Here we show the results for the CGL points and the grid points generated from a Kronrod scheme.

We only need to consider 1D grid because of the tensor product structure. Therefore, we omit the index k in the following analysis. In order to show

$\tilde{w}_{n_\gamma+1}(x)$ is bounded for CGL points, we need some results of the Chebyshev polynomial of the second kind.

Definition 2.18. The Chebyshev polynomial $U_n(x)$ of the second kind is a polynomial of degree n in x defined by

$$U_n(x) = \frac{\sin(n+1)\theta}{\sin\theta}, \; x = \cos\theta.$$

Using the definition, we can find $U_n(x)$ satisfies the recurrence relation

$$U_n(x) = 2xU_{n-1}(x) - U_{n-2}, \; n = 2, 3, \dots$$

with the initial conditions

$$U_0(x) = 1, \; U_1(x) = 2x.$$

From the recurrence relation, we can compute the leading term of $U_n(x)$ is $2^n x^n$.

Lemma 2.19. *The zeros of $U_n(x)$ are*

$$x = \cos\frac{i\pi}{(n+1)}, \; i = 1, 2, \dots, n.$$

Proof. When $\theta = \frac{i\pi}{n+1}$, we have $\sin(n+1)\theta = 0$ and thus $U_n(x) = 0$. By using the definition of the Chebyshev polynomial of the second kind, the zeros of $U_n(x)$ are

$$x = \cos\frac{i\pi}{(n+1)}, \; k = 1, 2, \dots, n.$$

□

Lemma 2.20. *If we take the following CGL points as our grid points*

$$x^i = \cos\frac{i\pi}{n}, \; i = 1, \dots, n.$$

Then the corresponding polynomial is

$$w_{n+1}(x) = \frac{1}{2^{n-1}}(1-x^2)U_{n-1}(x).$$

Proof. According to the definition, the polynomial

$$w_{n+1}(x) = (x - x^0)(x - x^1)\dots(x - x^n).$$

From Lemma 2.19, $x^1, \dots, x^{n-1}$ are zeros of $U_{n-1}(x)$. Thus, all the CCP points are the zeros of the polynomial

$$p_n(x) = (1-x^2)U_{n-1}(x).$$

The degrees of the polynomials $w_{n+1}(x)$ and $p_n(x)$ are both $n+1$. Therefore, we claim that

$$w_{n+1}(x) = Cp_n(x)$$

where C is a constant. Comparing the leading terms of $w_{n+1}(x)$ and $p_n(x)$, we have

$$w_{n+1}(x) = \frac{1}{2^{n-1}}(1-x^2)U_{n-1}(x).$$

□

Lemma 2.21. *The polynomial $w_{n+1}(x)$ is defined as in the previous Lemma, we have the following estimation*

$$|w_{n+1}(x)| \leq \frac{n}{2^{n-1}}, \ x \in [-1,1].$$

Proof. First, we compute $\max_{x\in[-1,1]} |U_{n-1}(x)|$. We first compute

$$\frac{d}{dx}U_n(x) = \frac{d}{dx}\frac{\sin n\theta}{\sin\theta} = \frac{-n\sin\theta\cos n\theta + \cos\theta\sin n\theta}{\sin^3\theta}.$$

The extreme value is taken when $\frac{d}{dx}U_n(x) = 0$. We get

$$\tan n\theta = n\tan\theta \neq 0.$$

Thus the extreme values of U_n are taken at $x = 1$ and $x = -1$. We have $\max_{x\in[-1,1]} |U_n(x)| = n+1$ and therefore

$$\max_{x\in[-1,1]} |U_{n-1}(x)| = n.$$

Using the result in the previous Lemma, we have

$$\begin{aligned} |w_{n+1}(x)| &\leq \frac{1}{2^{n-1}}|1-x^2||U_{n-1}(x)| \\ &\leq \frac{1}{2^{n-1}} \max_{x\in[-1,1]} |1-x^2| \max_{x\in[-1,1]} |U_{n-1}(x)| \\ &\leq \frac{n}{2^{n-1}}. \end{aligned}$$

□

Theorem 2.22. *If we take the following CGL points as our grid points*

$$x^i = \cos\frac{i\pi}{n_\gamma}, \ i = 0, \ldots, n_\gamma,$$

then the polynomial

$$\tilde{w}_{n_\gamma+1}(x) = \frac{1}{(n_\gamma+1)!\tilde{h}_\gamma^{n_\gamma+1}} w_{n_\gamma+1}(x)$$

is bounded on $[-1,1]$.

Proof. For large n_γ, we have the following Stirling's formula

$$(n_\gamma+1)! \sim \sqrt{2\pi(n_\gamma+1)}\left(\frac{n_\gamma+1}{e}\right)^{n_\gamma+1}.$$

Using the result in the previous Lemma, we have the following result

$$\begin{aligned}
|\tilde{w}_{n_\gamma+1}(x)| &\le \left|\frac{1}{(n_\gamma+1)!}\left(\frac{n_\gamma}{2}\right)^{n_\gamma+1}\frac{n_\gamma}{2^{n_\gamma-1}}\right| \\
&\sim \left|\frac{1}{\sqrt{2\pi(n_\gamma+1)}\left(\frac{n_\gamma+1}{e}\right)^{n_\gamma+1}}\left(\frac{n_\gamma}{2}\right)^{n_\gamma+1}\frac{n_\gamma}{2^{n_\gamma-1}}\right| \\
&= \left|\frac{4n_\gamma}{\sqrt{2\pi(n_\gamma+1)}}\left(\frac{e}{4}\right)^{n_\gamma+1}\left(\frac{n_\gamma}{n_\gamma+1}\right)^{n_\gamma+1}\right|.
\end{aligned}$$

Using the fact that

$$\left(1-\frac{1}{n_\gamma+1}\right)^{n_\gamma+1} \to \frac{1}{e}$$

as $n_\gamma \to \infty$. We get

$$\lim_{n_\gamma\to\infty}|\tilde{w}_{n_\gamma+1}(x)| = 0.$$

Therefore, there exists a constant M such that for any n_γ, we have

$$|\tilde{w}_{n_\gamma+1}(x)| \le M.$$

□

Next, we consider the grid points generated from a Kronrod scheme. We first show some properties of the Legendre polynomials and the Stieltjes polynomials.

We use the following expression of the Legendre polynomials

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n,\ n=1,2,\ldots$$

with the initial condition $P_0(x)=1$ where $x\in[-1,1]$.

Lemma 2.23. *$|P_n(x)|$ achieves maximum value at points 1 and -1. The maximum is 1.*

Definition 2.24. The Stieltjes polynomials E_{n+1} is defined by

$$\int_{-1}^{1} E_{n+1}(x)P_n(x)x^k\,dx = 0,\ k=0,1,\ldots,n.$$

We use the method in [34, 63] to normalise the Stieltjes polynomial. The polynomial $E_{n+1}(x)$ can be expanded as the following Chebyshev series

$$\begin{aligned}\frac{\Gamma_n}{2}E_{n+1}(\cos\theta) =& a_{0,n}\cos(n+1)\theta + a_{1,n}\cos(n-1)\theta \\ &+\cdots+\begin{cases} a_{\frac{n}{2},n}\cos\theta, & n \text{ even} \\ \frac{1}{2}a_{\frac{n+1}{2},n}, & n \text{ odd,}\end{cases}\end{aligned} \tag{2.8}$$

where $a_{0,n} = 1$ and

$$\Gamma_n = \sqrt{\pi}\frac{2^{2n+1}(n!)^2}{(2n+1)!}.$$

By using the definition of the Chebyshev polynomial of the first kind, the leading term of $E_{n+1}(x)$ appears in the first term of the expansion formula. The first term can be written as

$$\begin{aligned}&\frac{2}{\Gamma_n}a_{0,n}T_{n+1}(x) \\ =&\frac{2}{\Gamma_n}2^n x^{n+1} + p_n(x).\end{aligned}$$

where $p_n(x)$ here is a polynomial with degree less or equal than n. Therefore, the leading coefficient is $\frac{2^{n+1}}{\Gamma_n}$.

Lemma 2.25 ([34, 62]). *Under the normalisation* (2.8), *the Stieltjes polynomials are bounded by*

$$|E_{n+1}(x)| \leq \frac{4}{\Gamma_n},\ x \in [-1,1].$$

Theorem 2.26. *If we take the zeros of the Legendre polynomial $P_n(x)$ as the grid points, then the corresponding polynomial $\tilde{w}_n(x)$ is bounded on $x \in [-1,1]$.*

Proof. Suppose $x_0, \ldots, x_{n-1}$ are the zeros of $P_n(x)$. Then we have

$$w_n(x) = (x-x_0)\ldots(x-x_{n-1}) = \frac{2^n(n!)^2}{(2n)!}P_n(x).$$

Using the Lemma 2.23, we have

$$|w(x)| \leq \frac{2^n(n!)^2}{(2n)!}.$$

Therefore

$$|\tilde{w}_n(x)| \leq \frac{n!(n-1)^n}{(2n)!} \to 0$$

as $n \to \infty$. □

Theorem 2.27. *If we use the Kronrod scheme to extend the grid points in the Theorem* 2.26*, the newly added grid points are the zeros of the Stieljes polynomial* $E_{n+1}(x)$*. In addition, the polynomial* $\tilde{w}_{2n+1}(x)$ *is bounded on* $x \in [-1, 1]$*.*

Proof. Suppose $x_0, \ldots, x_{n-1}$ are the zeros of $P_n(x)$ and $y_0, \ldots, y_n$ are the zeros of $E_{n+1}(x)$. Then

$$\tilde{w}_{2n+1} = \frac{1}{(2n+1)!} n^{2n+1} (x - x_0) \ldots (x - x_{n-1})(x - y_0) \ldots (x - y_n).$$

Using the results in Lemma 2.23 and Lemma 2.25, we have

$$\begin{aligned} |\tilde{w}_{2n+1}(x)| &\leq \frac{n^{2n+1}}{(2n+1)!} \max_{x \in [-1,1]} \left| \prod_{i=0}^{n-1} (x - x_i) \right| \max_{y \in [-1,1]} \left| \prod_{j=0}^{n-1} (x - y_i) \right| \\ &\leq \frac{n^{2n+1}}{(2n+1)!} \frac{2^n (n!)^2}{(2n)!} \frac{\Gamma_n}{2^{n+1}} \frac{4}{\Gamma_n} \\ &= \frac{2n^{2n+1}(n!)^2}{(2n+1)!(2n)!} \\ &= \frac{2n^{2n+1}(n!)^2}{[(2n+1)\ldots(n+1)(2n)\ldots(n+1)](n!)^2} \to 0 \end{aligned}$$

as $n \to \infty$. □

For the Gauss-Patterson method, the analysis is much more complicated than that of the pure Kronrod scheme. We need to consider the problem that if the newly added grid points satisfy a polynomial of a certain type as we consider the Stieljes polynomial for Kronrod scheme. Then we need to study the properties of this type of polynomial in order to bound $|\tilde{w}_{n_\gamma + 1}(x)|$. However, it is still an open problem to find the polynomial [67].

2.3.3 Error of the Combination Technique

Based on the error splitting models, we can derive the error bound for the combination technique. The following lemma gives an upper bound for the error of the 2D classical sparse grid combination technique on equally spaced grid.

Theorem 2.28 ([43])**.** *Suppose* f *is a 2D function defined on domain* $[0, 1] \times [0, 1]$*. Let* f_γ *be an interpolant of* f *on the equally spaced grid* $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ *with the spacings* h_{γ_1} *and* h_{γ_2}*, and* $h_i = 2^{-i}$ *where* $i = \gamma_1, \gamma_2$*. Suppose* f_γ *satisfies the error expansion*

$$f - f_\gamma = C_1(x, h_{\gamma_1}) h_{\gamma_1}^p + C_2(x, h_{\gamma_2}) h_{\gamma_2}^p + C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^p h_{\gamma_2}^p$$

with $p > 0$ and for any $\gamma = (\gamma_1, \gamma_2)$, $|C_1(x, h_{\gamma_1})| \leq K$, $|C_2(x, h_{\gamma_2})| \leq K$ and $|C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})| \leq K$ for some $K > 0$. Then the error of the classical sparse grid combination technique f_n^c satisfies

$$|f - f_n^c| \leq (3 + (1 + 2^p)n)Kh_n^p.$$

Proof. First we notice that

$$\sum_{|\gamma|=n} f - \sum_{|\gamma|=n-1} f = (n+1)f - nf = f,$$

then we can write the error as

$$\begin{aligned} f - f_n^c &= f - \left(\sum_{|\gamma|=n} f_\gamma - \sum_{|\gamma|=n-1} f_\gamma \right) \\ &= \sum_{|\gamma|=n} (f - f_\gamma) - \sum_{|\gamma|=n-1} (f - f_\gamma). \end{aligned}$$

Now by using the error splitting model (2.5) and the fact $h_{n-1}^p = 2^p h_n^p$, we obtain the error

$$\begin{aligned} &f - f_n^c \\ =&\sum_{\gamma_1=0}^{n} C_1(x, h_{\gamma_1})h_{\gamma_1}^p - \sum_{\gamma_1=0}^{n-1} C_1(x, h_{\gamma_1})h_{\gamma_1}^p + \sum_{\gamma_2=0}^{n} C_2(x, h_{\gamma_2})h_{\gamma_2}^p - \sum_{\gamma_2=0}^{n-1} C_2(x, h_{\gamma_2})h_{\gamma_2}^p \\ &+ \sum_{\gamma_1+\gamma_2=n} C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})h_n^p - \sum_{\gamma_1+\gamma_2=n-1} C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})h_{n-1}^p \\ =&\left(C_1(x, h_n) + C_2(x, h_n) + \sum_{\gamma_1+\gamma_2=n} C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) - 2^p \sum_{\gamma_1+\gamma_2=n-1} C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) \right) h_n^p. \end{aligned}$$

Taking the absolute value of we have

$$\begin{aligned} |f - f_n^c| \leq &|C_1(x, h_n)|h_n^p + |C_2(x, h_n)|h_n^p \\ &+ \left(\sum_{\gamma_1+\gamma_2=n} |C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})| + 2^p \sum_{\gamma_1+\gamma_2=n-1} |C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})| \right) h_n^p \\ &\leq Kh_n^p + Kh_n^p + (n+1)Kh_n^p + 2^p nKh_n^p \\ &= (3 + (1 + 2^p)n)Kh_n^p, \end{aligned}$$

as claimed. □

For interpolation on unequally spaced grid, we again need to introduce the average spacing. The coefficients C_1, C_2 and $C_{1,2}$ depend on the choices of the grids for this case.

Theorem 2.29. *Suppose f is a 2D function defined on domain $[a_1, b_1] \times [a_2, b_2]$. Let f_γ be an interpolant of a function f on the unequally spaced grids $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$. The numbers of grid points are $n_{\gamma_1} + 1$ and $n_{\gamma_2} + 1$ with respect to different dimensions. The average spacings are defined as*

$$\tilde{h}_{\gamma_1} = \frac{b_1 - a_1}{n_{\gamma_1}}$$
$$\tilde{h}_{\gamma_2} = \frac{b_2 - a_2}{n_{\gamma_2}}.$$

We further assume the grids G_γ are hierarchical.

$$G_{\gamma_k} \subset G_{\gamma_k+1}$$
$$n_i = 2^i, \; i = \gamma_1, \gamma_2.$$

Suppose the interpolant f_γ satisfies the error expansion

$$f - f_\gamma = C_1(x, G_{\gamma_1})\tilde{h}^p_{\gamma_1} + C_2(x, G_{\gamma_2})\tilde{h}^p_{\gamma_2} + C_{1,2}(x, G_{\gamma_1}, G_{\gamma_2})\tilde{h}^p_{\gamma_1}\tilde{h}^p_{\gamma_2}$$

with $p > 0$ and for any $\gamma = (\gamma_1, \gamma_2)$, $|C_1(x, G_{\gamma_1})| \leq K$, $|C_2(x, G_{\gamma_2})| \leq K$ and $|C_{1,2}(x, G_{\gamma_1}, G_{\gamma_2})| \leq K$ for some $K > 0$. Then the error of the classical sparse grid combination technique f^c_n satisfies

$$|f - f^c_n| \leq (3 + (1 + 2^p)n)K\tilde{h}^p_n$$

where

$$\tilde{h}_n = \left(\frac{b_1 + b_2 - a_1 - a_2}{2}\right)^2 2^{-n}.$$

Proof. The proof is the same as the proof of the Theorem 2.28 if we replace the spacings h_{γ_1} and h_{γ_2} with the average spacings $\tilde{h}_{\gamma_1}$ and $\tilde{h}_{\gamma_2}$ and notice that

$$(b_1 - a_1)(b_2 - a_2) \leq \left(\frac{b_1 + b_2 - a_1 - a_2}{2}\right)^2.$$

□

Corollary 2.30. *Suppose $f \in C^2(X) \otimes C^2(X)$ and $X = [0, 1]$. If we use the piecewise linear interpolant $f_\gamma = \mathcal{K}_\gamma f$ to approximate the function f on equally spaced grid $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ with the spacings h_{γ_1} and h_{γ_2}, and $h_i = 2^{-i}$ where $i = \gamma_1, \gamma_2$. then in this case the error of the 2D classical sparse grid combination technique interpolant f^c_n*

$$f^c_n = \sum_{|\gamma|=n} f_\gamma - \sum_{|\gamma|=n-1} f_\gamma$$

satisfies

$$|f - f_n^c| \leq (3 + 5n)Kh_n^2$$

where

$$K = \max\left\{\frac{1}{4}\|\partial_1^2 f\|_\infty,\ \frac{1}{4}\|\partial_2^2 f\|_\infty,\ \left(\frac{1}{4}\right)^2\|\partial_1^2\partial_2^2 f\|_\infty\right\}.$$

Proof. Combine the results in the Theorem 2.6 and Theorem 2.28. □

Corollary 2.31. *Suppose $f \in C^{n_{\gamma_1}+1}(X_1) \otimes C^{n_{\gamma_2}+1}(X_2)$ and $X_1 = [a_1, b_1]$, $X_2 = [a_2, b_2]$. The function f is approximated on the grid G_γ where $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ where G_{γ_k} consists of the following grid points*

$$a_k \leq x_k^0 \leq x_k^1 \leq \cdots \leq x_k^{n_{\gamma_k}} \leq b_k,\ k = 1, 2.$$

Suppose we further assume the grids G_γ are hierarchical.

$$G_{\gamma_k} \subset G_{\gamma_k+1}$$
$$n_i = 2^i,\ i = \gamma_1, \gamma_2.$$

$f_\gamma = L_{n_\gamma} f$ is the 2D Lagrangian interpolant of f on the grid G_γ. We define the average spaceings

$$\tilde{h}_{\gamma_1} = \frac{b_1 - a_1}{n_{\gamma_1}}$$
$$\tilde{h}_{\gamma_2} = \frac{b_2 - a_2}{n_{\gamma_2}}.$$

We further define

$$\tilde{w}_{n_{\gamma_k}+1}(x_k) = \frac{1}{(n_{\gamma_k}+1)!\tilde{h}_{\gamma_k}^{n_{\gamma_k}+1}} w_{n_{\gamma_k}+1}(x_k)$$

where

$$w_{n_{\gamma_k}+1}(x_k) = (x_k - x_k^0)\ldots(x_k - x_k^{n_{\gamma_k}}).$$

$k = 1, 2$. If there exists $M_1 > 0$, $M_2 > 0$ such that

$$|\tilde{w}_{n+1}^k(x_k)| \leq M_k,\ k = 1, 2$$

then the error of the 2D classical sparse grid combination technique interpolant

$$f_n^c = \sum_{|\gamma|=n} f_\gamma - \sum_{|\gamma|=n-1} f_\gamma$$

satisfies

$$|f - f_n^c| \leq (3 + 5n)K\tilde{h}_n^{p+1}$$

where

$$\begin{aligned} p &= \min\left\{n_{\gamma_1},\ n_{\gamma_2}\right\}, \\ K &= \max\left\{M_1\|\partial_1^{p+1}f\|_\infty,\ M_2\|\partial_2^{p+1}f\|_\infty,\ M_1M_2\|\partial_1^{p+1}\partial_2^{p+1}f\|_\infty\right\}. \end{aligned}$$

and

$$\tilde{h}_n = \left(\frac{b_1+b_2-a_1-a_2}{2}\right)^2 2^{-n}.$$

Proof. Combine the results in the Theorem 2.15 and Theorem 2.29. □

2.3.4 d Dimensional Case

Here we review the result of interpolation on equally spaced grid in [76]. We also generalise the result to the interpolation on unequally spaced grid using the similar approach as we did in 2D case. In order to analyse the error for the d dimensional sparse grid, we need to first introduce the following notations and combinatorial identities. We first define

$$S(n,d) := \sum_{|\gamma|=n} f_\gamma$$

for the multi-index $\gamma \in \mathbb{N}^d$. Then we define the difference operator δ on a function $g : \mathbb{N} \to \mathbb{R}$,

$$\delta g(n) := g(n) - g(n-1).$$

Lemma 2.32 ([76])**.** *The combination formula in d dimensions is given by*

$$\delta^{d-1}S(n,d) = \sum_{k=0}^{d-1}(-1)^k\binom{d-1}{k}\sum_{|\gamma|=n-k} f_\gamma.$$

Proof. The original proof can be found in [76]. A more detailed proof is in [45]. □

Next, we show the consistency of the combination technique.

Lemma 2.33 ([76])**.** *Let*

$$N(n,d) := \sum_{|\gamma|=n} 1 = \binom{n+d-1}{d-1},$$

then

$$\delta^{d-1}N(n,d) = 1.$$

Proof. We refer [76] and [45] for the proof here. □

By using the Lemma 2.32 and Lemma 2.33, we have when $f_\gamma = 1$ for all γ

$$1 = \delta^{d-1}N(n,d) = \delta^{d-1}S(n,d) = \sum_{k=0}^{d-1}(-1)^k \binom{d-1}{k} \sum_{|\gamma|=n-k} 1.$$

[76] Next, we leave the following Leibniz rule for the difference operator.

Lemma 2.34. *Let g_1, $g_2 : \mathbb{N} \to \mathbb{R}$, then for all $k \leq n$*

$$\delta^k(g_1(n)g_2(n)) = \sum_{j=0}^{k} \binom{k}{j} (\delta^{k-1}g_1(n-j))(\delta^j g_2(n)).$$

Proof. We refer [76] and [45] for the proof here. □

Next, we have the following Lemma

Lemma 2.35 ([76])**.** *Let $d \in \mathbb{N}$ and $g:\ \mathbb{N} \to \mathbb{R}$ and define*

$$F(n) := \sum_{l=0}^{n} \binom{n-l+d-1}{d-1} g(l).$$

If $0 \leq k < d \leq n$ then $\delta^k F(n) = G_k(n) + H_k(n)$ where

$$G_k(n) := \begin{cases} 0 & k = 0 \\ \displaystyle\sum_{l=1}^{k} \binom{d-l-1}{k-l} g(n-k+l) & k \geq 1 \end{cases},$$

$$H_k(n) := \sum_{l=0}^{n-k} \binom{n-k-l+d-1}{d-k-1} g(l).$$

Additionally, $\delta^d F(n) = g(n)$.

Proof. We refer [76] and [45] for the proof here. □

The last lemma shows the combination of individual error terms in the error splitting.

Lemma 2.36 ([76])**.** *Let $m, d, p \geq 1$, $v : \mathbb{R}_+^m \to \mathbb{R}$ and for $n \in \mathbb{N}$*

$$F(n) := \sum_{|\alpha|_1 = n} v(2^{-\alpha_1}, \ldots, 2^{-\alpha_m}) 2^{-p\alpha_1} \ldots 2^{-p\alpha_m}$$

Then

$$\delta^{d-1}F(n) = 2^{-pn} \sum_{j=0}^{m-1} \binom{m-1}{j} (-2^p)^j s_{n-j}$$

where $s_l := \sum_{|\alpha|_1 = l} v(2^{-\alpha_1}, \ldots, 2^{-\alpha_m})$.

Proof. We refer [76] and [45] for the proof here. □

Theorem 2.37 ([76])**.** *For $\gamma \in \mathbb{N}^d$, let $f_\gamma : [0,1]^d \mapsto \mathbb{R}$ be an approximation to $f : [0,1]^d \mapsto \mathbb{R}$ satisfying the pointwise error expansion*

$$f - f_\gamma = \sum_{m=1}^{d} \sum_{\{j_1,\ldots,j_m\}\subset\{1,\ldots,d\}} v_{j_1,\ldots,j_m}(h_{\gamma_{j_1}},\ldots,h_{\gamma_{j_m}}) h^p_{\gamma_{j_1}} \ldots h^p_{\gamma_{j_m}}.$$

Additionally, suppose there exists some $K > 0$ such that

$$|v_{j_1,\ldots,j_m}(h_{\gamma_{j_1}},\ldots,h_{\gamma_{j_m}})| \le K,\ \forall 1 \le m \le d \text{ and } \forall \{j_1,\ldots,j_m\} \subset \{1,\ldots,d\}.$$

Then the combination

$$f_n^c = \sum_{k=0}^{d-1} (-1)^k \binom{d-1}{k} \sum_{|\gamma|=n-k} f_\gamma$$

satisfies the pointwise error bound

$$|f - f_n^c| \le K 2^{-pm} (1+2^p)^{d-1} \binom{n+2d-1}{d-1}.$$

Proof. Here we provide a sketch of the proof shown in [76] and [45]. The idea is similar as the idea of the proof of the Theorem 2.28. Applying the Lemma 2.33, we have that

$$\delta^{d-1} \sum_{|\gamma|=n} f = f \delta^{d-1} \sum_{|\gamma|=n} 1 = f \delta^{d-1} N(n,d) = f.$$

Combining the above equation and the result in the Lemma 2.32, we obtain

$$f - f_n^c = \delta^{d-1} \sum_{|\gamma|=n} f - f_\gamma.$$

Let

$$F_{j_1,\ldots,j_m}(n) = \sum_{|\gamma|=n} v_{j_1,\ldots,j_m}(h_{\gamma_{j_1}},\ldots,h_{\gamma_{j_m}}) h^p_{\gamma_{j_1}} \ldots h^p_{\gamma_{j_m}}.$$

Using the error expansion and changing the order of summation, we have

$$f - f_n^c = \sum_{m=1}^{d} \sum_{\{j_1,\ldots,j_m\}\subset\{1,\ldots,d\}} \delta^{d-1} F_{j_1,\ldots,j_m}(n).$$

Applying the Lemma 2.36, we get

$$f - f_n^c = 2^{-pn} \sum_{m=1}^{d} \sum_{\{j_1,\ldots,j_m\}\subset\{1,\ldots,d\}} \sum_{j=0}^{m-1} \binom{m-1}{j} (-2^p)^j s_{n-j,m}$$

where $s_{l,m} := \sum_{|\gamma|=l} v_{j_1,\ldots,j_m}$. Using the triangle inequality and the assumption $|v_{j_1,\ldots,j_m}| \leq K$, we have

$$\max_{j=0,\ldots,m-1} |s_{n-j,m}| \leq \binom{n+m-1}{m-1} K.$$

Therefore,

$$\begin{aligned}
|f - f_n^c| &\leq 2^{-pn} \sum_{m=1}^{d} \sum_{\{j_1,\ldots,j_m\}\subset\{1,\ldots,d\}} \left| \sum_{j=0}^{m-1} \binom{m-1}{j} (-2^P)^j s_{n-j,m} \right| \\
&\leq 2^{-pn} \sum_{m=1}^{d} \sum_{\{j_1,\ldots,j_m\}\subset\{1,\ldots,d\}} K \binom{n+m-1}{m-1} (1+2^p)^{m-1} \\
&\leq K 2^{-pn} (1+2^p)^{d-1} \sum_{m=1}^{d} \binom{d}{m} \binom{n+m-1}{m-1}.
\end{aligned}$$

Using the equality

$$\binom{n+2d-1}{d-1} = \sum_{m-1}^{d} \binom{d}{m} \binom{n+m-1}{m-1},$$

we finally obtain the result

$$|f - f_n^c| \leq K 2^{-pn} (1+2^p)^{d-1} \binom{n+2d-1}{d-1}.$$

□

Theorem 2.37 gives us the same asymptotic result as we get from Theorem 2.5 for 2D case. In [43], the authors also provide a bound for 3D case. Comparing their result with the result from Theorem 2.37 when $d = 3$, we also get the same asymptotic property.

Similar as we did for the 2D case, we can get the corresponding Theorem for approximation on unequally spaced grids if we further introduce the average spacing and have an error splitting model with respect to the average spacing.

Theorem 2.38. *Suppose f is a d dimensional function defined on the box domain $\prod_{k=1}^{d}[a_k, b_k]$. Let f_γ be an interpolant of a function f on the unequally spaced grids $G_\gamma = \prod_{k=1}^{d} G_{\gamma_k}$. The numbers of grid points are $n_{\gamma_k} + 1$, $k = 1, \ldots, d$. The average spacings are defined as*

$$\tilde{h}_{\gamma_k} = \frac{b_k - a_k}{n_{\gamma_k}}, \ k = 1, \ldots, d$$

We further assume the grids G_γ are hierarchical.

$$\begin{aligned} G_{\gamma_k} &\subset G_{\gamma_k+1} \\ n_i &= 2^i,\ i=\gamma_1,\gamma_2. \end{aligned}$$

Suppose the d dimensional interpolant f_γ satisfies the error expansion

$$f - f_\gamma = \sum_{m=1}^{d} \sum_{\{j_1,\dots,j_m\}\subset\{1,\dots,d\}} v_{j_1,\dots,j_m}(G_{\gamma_{j_1}},\dots,G_{\gamma_{j_m}})\tilde{h}^p_{\gamma_{j_1}}\dots\tilde{h}^p_{\gamma_{j_m}}.$$

Additionally, suppose there exists some $K>0$ such that

$$|v_{j_1,\dots,j_m}(G_{\gamma_{j_1}},\dots,G_{\gamma_{j_m}})| \le K,\ \forall 1\le m\le d \text{ and } \forall\,\{j_1,\dots,j_m\}\subset\{1,\dots,d\}\,.$$

Then the combination

$$f^c_n = \sum_{k=0}^{d-1}(-1)^k\binom{d-1}{k}\sum_{|\gamma|=n-k} f_\gamma$$

satisfies the pointwise error bound

$$|f-f^c_n| \le K\tilde{h}^p_n(1+2^p)^{d-1}\binom{n+2d-1}{d-1}$$

where

$$\tilde{h}_n = \left(\frac{\sum_{k=1}^{d}(b_k-a_k)}{d}\right)^d 2^{-n}.$$

Proof. The proof is the same as the proof of the Theorem 2.37 if we replace the spacings h_{γ_1} and h_{γ_2} with the average spacings $\tilde{h}_{\gamma_1}$ and $\tilde{h}_{\gamma_2}$ and notice that

$$\prod_{k=1}^{d}(b_k-a_k) \le \left(\frac{\sum_{k=1}^{d}(b_k-a_k)}{d}\right)^d.$$

□

2.4 Generalised Combination Technique

The 2D classical sparse grid combination technique is based on the error splitting model

$$f-f_\gamma = C_1(x,h_{\gamma_1})h^p_{\gamma_1} + C_2(x,h_{\gamma_2})h^p_{\gamma_2} + C_{1,2}(x,h_{\gamma_1},h_{\gamma_2})h^p_{\gamma_1}h^p_{\gamma_2}.$$

In this model, the error decays with the same rate in both dimensions. In fact, we can generalise the model to be

$$f - f_\gamma = C_1(x, h_{\gamma_1})h_{\gamma_1}^p + C_2(x, h_{\gamma_2})h_{\gamma_2}^q + C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q$$

with $p \neq q$. From the symmetry of the classical sparse grid, we know it can not be the 'optimal' grid to compute a problem with such generalised model. Even when we consider a problem with the standard error splitting model, sometimes the classical sparse grid is not the 'optimal' choice since the coefficients $C_1(x, h_{\gamma_1})$ and $C_2(x, h_{\gamma_2})$ can be largely different. In order to improve the sparse grid and take advantage of the asymmetry of the problem, more generalised combination technique is required. The generalised combination technique can be subdivided into two categories. In the first kind of combination technique [48, 45, 47], the combination scheme is decided before the computation. In the second kind of combination technique [48, 46, 38, 45, 49], the combination scheme is chosen during the computation, in other words, the (dimension) adaptive approach. The combination scheme for combination technique in both categories is chosen based on the error splitting model or equivalent surplus decay model.

In this section, we first introduce an abstract framework which can be used to define generalised combination technique in both categories. Then, we review some important theoretic results. Finally, we give a few numerical examples.

We begin with introducing the concept of partially ordered set.

Definition 2.39. Suppose I is a set and $\prec$ is a homogeneous binary relation defined on set I. The set I is a partially ordered set if for any element α, β and γ in I satisfy

- reflexive: $\alpha \prec \alpha$,
- antisymmetry: if $\alpha \prec \beta$ and $\beta \prec \alpha$, then $\alpha = \beta$,
- transitivity: if $\alpha \prec \beta$ and $\beta \prec \gamma$, then $\alpha \prec \gamma$.

For example, the set $\mathbb{N}^d$ of integer tuples is a partially ordered set if we define the following partial order on it,

$$\alpha \prec \beta \text{ if } \alpha_k \leq \beta_k, \ k = 1, \ldots, d.$$

In the previous chapter, we have already denoted this particular partial order $\alpha \prec \beta$ on lattice $\mathbb{N}^d$ as $\alpha \leq \beta$. We will follow this notation here. If any two elements in a partially ordered set have a least upper bound and a greatest lower bound, then it forms a lattice [24]. For example, $\mathbb{N}^d$ is a lattice, we have

$$\eta = \alpha \vee \beta : \ \eta_k = \max\{\alpha_k, \beta_k\}, \ k = 1, \ldots, d,$$
$$\gamma = \alpha \wedge \beta : \ \gamma_k = \min\{\alpha_k, \beta_k\}, \ k = 1, \ldots, d.$$

Suppose we have a function $f \in U$. We assume α, β, $\gamma \in \mathbb{N}^d$ are d-dimensional multi-indices. We say finite dimensional spaces $U_\gamma \subset U$ are hierarchical if $U_\alpha \subset U_\beta$ when $\alpha \leq \beta$. Here we actually defined a partial order on a lattice of function spaces $\{U_\gamma\}_{\gamma \in \mathbb{N}^d}$. The partial order relation $(\{U_\gamma\}_{\gamma \in \mathbb{N}^d}, \subset)$ is lifted from the partial order relation $(\mathbb{N}^d, \leq)$. Similarly, we can define the least upper bound and the greatest lower bound for any two spaces U_α and U_β as

$$U_\alpha \cup U_\beta = U_{\alpha \vee \beta},$$
$$U_\alpha \cap U_\beta = U_{\alpha \wedge \beta}.$$

Then, we denote projections from $U \to U_\gamma$ as P_γ and $f_\gamma = P_\gamma(f)$. The operator P_γ is the tensor product of the 1D projection operator P_{γ_k}, $k = 1, \ldots, d$, i.e. $P_\gamma = \bigotimes_{k=1}^d P_{\gamma_k}$. We have the following proposition for the projection operator P_γ.

Proposition 2.40 ([46])**.** *For every lattice space generated from a tensor product hierarchical space we have*

- *There are linear operators P_α on U with range $R(P_\alpha) = U_\alpha$ and $P_\alpha P_\beta = P_{\alpha \wedge \beta}$.*
- *Consequently, $P_\alpha P_\alpha = P_\alpha$ and $P_\alpha P_\beta = P_\beta P_\alpha$.*

Proof. Using the tensor product structure of the operator P_α, P_β and the definition of the projection operator. □

When we define a generalised combination technique, we need the following concepts.

Definition 2.41. A multi-indices set I is a downset if when

$$\alpha \in I \text{ and } \beta \leq \alpha,$$

then $\beta \in I$. If a multi-indices set J is not a downset, we define the smallest downset that contains J as $J\downarrow$.

Definition 2.42. We denote the power set of the set of all multi-indices as $\mathcal{P}(\mathbb{N}^d)$ and its subset which only contains finite downsets as $\mathcal{D}(\mathbb{N}^d)$.

$(\mathcal{D}(\mathbb{N}^d), \subset)$ is a partially ordered set if we check reflexive, antisymmetry and transitivity. If we further define the least upper bound and greatest lower bound for any two multi indices sets I, $J \in \mathcal{D}(\mathbb{N}^d)$ as set union and set intersection, we can prove $(\mathcal{D}(\mathbb{N}^d), \subset)$ also forms a lattice.

Definition 2.43. We define $(\{U_I\}_{I\in\mathcal{D}(\mathbb{N}^d)}, \subset)$ as combination space lattice where

$$U_I := \sum_{\gamma\in I} U_\gamma.$$

The least upper bound and the greatest lower bound for any two spaces V_I and U_J are defined as

$$U_I \cup U_J = U_{I\cup J},$$
$$U_I \cap U_J = U_{I\cap J}.$$

The lifting from $\mathcal{D}(\mathbb{N}^d)$ to $\{U_I\}_{I\in\mathcal{D}(\mathbb{N}^d)}$ is similar as the lifting from $\mathbb{N}^d$ to $\{V_\gamma\}_{\gamma\in\mathbb{N}^d}$.

Proposition 2.44 ([46]). *Let the lattice $\{U_\gamma\}_{\gamma\in I}$ have the projections P_γ as in Proposition* 2.40 *then there are linear operators P_I on U with range $R(P_I) = U_I$ such that $P_I P_J = P_{I\cap J}$. Conversely, if P_I is a family of projections with these properties then $P_\gamma := P_{\{\gamma\}\downarrow}$ defines a family of projections as in Proposition* 2.40.

Proof. We first define the following linear operators

$$P_I = 1 - \prod_{\gamma\in I}(1 - P_\gamma)$$

and define the set $\max I$ as

$$\max I = \{\gamma \in I : \text{ If there is no } \alpha \in I \setminus \{\gamma\} \text{ such that } \gamma \leq \alpha\}.$$

Let $\beta \in I \setminus \max I$. We have

$$\begin{aligned} 1 - \prod_{\gamma\in I}(1 - P_\gamma) &= 1 - (1 - P_\beta) \prod_{\gamma\in I\setminus\{\beta\}} (1 - P_\gamma) \\ &= 1 - \prod_{\gamma\in I\setminus\{\beta\}} (1 - P_\gamma) + P_\beta \prod_{\gamma\in I\setminus\{\beta\}} (1 - P_\gamma). \end{aligned}$$

Since $\beta \in I \setminus \max I$, there exists an $\alpha \in \max I$ such that $\beta \leq \alpha$ and therefore

$$P_\beta(1 - P_\alpha) = 0.$$

Hence

$$P_\beta \prod_{\gamma \in I \setminus \{\beta\}} (1 - P_\gamma) = 0.$$

By repeating the computation above for all other $\eta \in I \setminus \max I$, we finally obtain that

$$P_I = 1 - \prod_{\gamma \in \max I} (1 - P_\gamma).$$

Using the properties of P_γ in Proposition 2.40, we first have $P_\alpha P_\beta = P_{\alpha \wedge \beta}$ and therefore

$$P_I = \sum_{\gamma \in I} c_\gamma P_\gamma \tag{2.9}$$

where the combination coefficients $c_\gamma = 1$ if $\gamma \in \max I$ and $c_\gamma = 0$ if γ is not in the sublattice generated by $\max I$. Therefore, the range of P_I is U_I.

Next, we prove $P_I \cap P_J = P_{I \cap J}$. Let $Q = P_I P_J - P_{I \cap J}$. The range of operator Q is $U_{I \cap J}$. We have

$$\begin{aligned} Q^2 &= (P_I P_J - P_{I\cap J})(P_I P_J - P_{I \cap J}) \\ &= P_I P_J - 2P_{I \cap J} + P_{I \cap J} \\ &= P_I P_J - P_{I \cap J} = Q. \end{aligned}$$

Therefore $Q = 0$. The converse follows directly. □

According to the proof of Proposition 2.44, we can actually write P_I as a linear combination of the projections P_γ, $\gamma \in I$. The projection P_I can be computed once we know the combination coefficients. The following Proposition provides another method to compute the projection P_J when J is a covering of a known I.

Proposition 2.45 ([46]). *Let $J = I \cup \{\alpha\}$ be a covering element of I and let P_I be the family of projections as in the Proposition 2.44 and $P_\gamma = P_{\{\gamma\}\downarrow}$. Then one has*

$$P_J - P_I = \sum_{\gamma \in J} d_\gamma P_\gamma$$

where $d_\alpha = 1$ and for $\gamma \in I$, we have

$$d_\gamma = - \sum_{\beta \in I_{\gamma|\alpha}} c_\beta$$

with $I_{\gamma|\alpha} := \{\beta \in I : \ \alpha \wedge \beta = \gamma\}$.

Proof. Using the definition in the proof of the previous Proposition, we have

$$\begin{aligned}
P_J - P_I &= (1 - (1 - P_\alpha)\prod_{\gamma\in I}(1 - P_\gamma)) - (1 - \prod_{\gamma\in I}(1 - P_\gamma)) \\
&= P_\alpha \prod_{\gamma\in I}(1 - P_\gamma) \\
&= P_\alpha - P_\alpha(1 - \prod_{\gamma\in I}(1 - P_\gamma)) \\
&= P_\alpha - P_\alpha P_I.
\end{aligned}$$

Using the combination formula (2.9), we further have

$$\begin{aligned}
P_J - P_I &= P_\alpha - P_\alpha \sum_{\gamma\in I} c_\gamma P_\gamma \\
&= P_\alpha - \sum_{\gamma\in I} c_\gamma P_{\alpha\wedge\gamma} \\
&= P_\alpha - \sum_{\gamma} P_\gamma \sum_{\beta\in I_{\gamma|\alpha}} c_\beta \\
&= \sum_{\gamma\in J} d_\gamma P_\gamma.
\end{aligned}$$

In the second equation, we use the Proposition 2.40. In the third equation, we group the all the terms with P_γ. □

We can compute P_I by using the result in either Proposition 2.44 or Proposition 2.45. However, both propositions do not provide a concrete expression used to compute the combination coefficients c_γ. In order to get the expressions for the coefficients, we first introduce the hierarchical surpluses operator and study their connections with the projections P_γ. We define the hierarchical surpluses operator Δ^d_α as followings

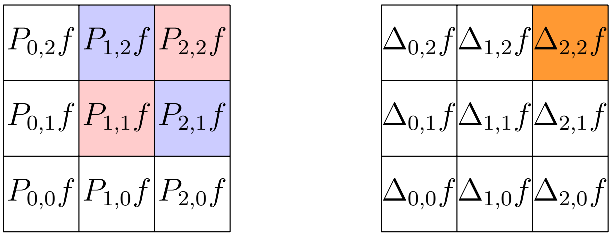
$$\Delta^d_\alpha := \Delta^1_{\alpha_1} \otimes \cdots \otimes \Delta^1_{\alpha_d} \tag{2.10}$$

where $\Delta^1_{\alpha_k}$, $k = 1, \ldots, d$ are 1D hierarchical surplus operators

$$\Delta^1_{\alpha_k} = P_{\alpha_k} - P_{\alpha_k - 1}.$$

In Figure 3.1, we show how to compute the 2D surplus $\Delta_{2,2}$ by the projections.

By following proposition, we can also reconstruct the projection P_γ by the hierarchical surpluses operators.



$P_{0,2}f$	$P_{1,2}f$	$P_{2,2}f$
$P_{0,1}f$	$P_{1,1}f$	$P_{2,1}f$
$P_{0,0}f$	$P_{1,0}f$	$P_{2,0}f$

$\Delta_{0,2}f$	$\Delta_{1,2}f$	$\Delta_{2,2}f$
$\Delta_{0,1}f$	$\Delta_{1,1}f$	$\Delta_{2,1}f$
$\Delta_{0,0}f$	$\Delta_{1,0}f$	$\Delta_{2,0}f$

Figure 2.2: Take $d = 2$ and $\alpha = (2, 2)$.By using the definition, we have $\Delta^2_{2,2} = \Delta^1_2 \otimes \Delta^1_2 = (P_2 - P_1) \otimes (P_2 - P_1) = P_{2,2} - P_{1,2} - P_{2,1} + P_{1,1}$.

Proposition 2.46 ([48]). *Suppose $f \in U$. Let U_γ be a sequence of finite dimensional subspaces of U with $\gamma \in \mathbb{N}^d$ such that $U_\alpha \subset U_\beta$ if $\alpha \leq \beta$ and let P_γ be the projections from $U \to U_\gamma$. Δ_α are defined in* (2.10)*. Then we have*

$$\sum_{\alpha\leq\gamma} \Delta_\alpha(f) = P_\gamma(f)$$

Moreover, the Δ_γ are uniquely determined and one has

$$\Delta_\gamma(f) = \sum_{\gamma\in B(\alpha)} (-1)^{|\alpha-\gamma|} P_\gamma(f)$$

where $B(\alpha) = \{\gamma \geq 0 \,|\, \alpha - 1 \leq \gamma \leq \alpha\}$.

Proof. By using the defination of the surplus operator

$$\begin{aligned}
\Delta_\gamma &= \bigotimes_{k=1}^{d} \Delta_{\gamma_k} = \bigotimes_{k=1}^{d} (P_{\gamma_k} - P_{\gamma_{k-1}}) \\
&= P_{\gamma_1} \otimes \dots \otimes P_{\gamma_d} - P_{\gamma_1 - 1} \otimes \dots \otimes P_{\gamma_d} + \dots \\
&+ (-1)^d P_{\gamma_1 - 1} \otimes \dots \otimes P_{\gamma_d - 1} \\
&= \sum_{\gamma\in B(\alpha)} (-1)^{|\alpha-\gamma|} P_\gamma.
\end{aligned}$$

By direct computation, we have

$$\begin{aligned}
\sum_{\alpha\leq\gamma} \Delta^d_\alpha(f) &= \sum_{\alpha\leq\gamma} \bigotimes_{k=1}^{d} \Delta_{\alpha_k} = \sum_{\alpha\leq\gamma} \bigotimes_{k=1}^{d} (P_{\alpha_k} - P_{\alpha_{k-1}}) \\
&= \bigotimes_{k=1}^{d} \sum_{\alpha\leq\gamma} (P_{\alpha_k} - P_{\alpha_{k-1}}) = \bigotimes_{k=1}^{d} P_{\gamma_k} = P_\gamma.
\end{aligned}$$

□

Based on the result in the previous proposition, we define the (generalised) combination technique as follow

Definition 2.47. Given $I \in \mathcal{P}(\mathbb{N}^d)$ where $\mathcal{P}(\mathbb{N}^d)$ is the power set of the set of all multi-indices. Then the (generalised) combination technique on set I is

$$P_I^{'}(f) = \sum_{\alpha \in I} \Delta_\alpha(f). \tag{2.11}$$

Though there is no restriction for the choices of set I, we usually take I as a downset. We say I is a downset if $\alpha \in I$ and $\beta \leq \alpha$, then $\beta \in I$. In particular, if $I = \{\gamma \,|\, |\gamma| \leq n\}$, we get level n classical sparse grid.

Lemma 2.48 ([45]). *Given $I \in \mathcal{D}(\mathbb{N}^d)$. then $P_I^{'} = P_I$.*

Proof. Using the result in Proposition 2.44, we have

$$P_I = 1 - \prod_{\gamma \in \max I} (1 - P_\gamma).$$

Expand the product and use the Proposition 2.40 and the Proposition 2.46, we get

$$\begin{aligned} 1 - \prod_{\gamma \in \max I} (1 - P_\gamma) &= 1 - \sum_{J \subset \max I} (-1)^{|J|} \prod_{\alpha \in J} P_\alpha \\ &= 1 - \sum_{J \subset \max I} (-1)^{|J|} P_{\wedge_{\alpha \in J} \alpha} \\ &= \sum_{J \subset \max I, J \neq \emptyset} (-1)^{|J|+1} \sum_{\beta \leq \wedge_{\alpha \in J} \alpha} \Delta_\beta. \end{aligned} \tag{2.12}$$

Therefore, after rearranging the summations, we have

$$P_I = \sum_{\beta \in I} d_\beta \Delta_\beta$$

where d_β, $\beta \in I$ are some coefficients. We only need to show $d_\beta = 1$, $\beta \in I$. First we notice that for any η and ϵ, we have

$$\Delta_\eta \Delta_\epsilon = \begin{cases} \Delta_\eta, & \text{if } \eta = \epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

This can be verified directly by using the definition. Based on this result, we get for any $\eta \in I$

$$\Delta_\eta P_I = \sum_{\beta \in I} d_\beta \Delta_\eta \Delta_\beta = d_\eta \Delta_\eta.$$

Using (2.12), we also have for any $\eta \in I$

$$\begin{aligned}
\Delta_\eta P_I &= \sum_{J \subset \max I, J \neq \emptyset} (-1)^{|J|+1} \sum_{\beta \leq \wedge_{\alpha \in J} \alpha} \Delta_\eta \Delta_\beta \\
&= \sum_{m=1}^{|J_\eta|} \sum_{\{\beta_1 \dots \beta_m \subset J_\eta\}} (-1)^{m+1} \Delta_\eta \\
&= \Delta_\eta \sum_{m=1}^{|J_\eta|} \sum_{\{\beta_1 \dots \beta_m \subset J_\eta\}} (-1)^{m+1} \\
&= \Delta_\eta \sum_{m=1}^{|J_\eta|} \binom{|J_\eta|}{m} (-1)^{m+1} \\
&= \Delta_\eta \left(1 - \sum_{m=0}^{|J_\eta|} \binom{|J_\eta|}{m} (-1)^{m+1} \right) \\
&= \Delta_\eta (1 - (1-1)^{|J_\eta|}) = \Delta_\eta.
\end{aligned}$$

Therefore

$$d_\eta \Delta_\eta = \Delta_\eta P_I = \Delta_\eta$$

and hence $d_\eta = 1$. □

Using the result in Lemma 2.48, we can further compute the combination coefficients in (2.9).

Lemma 2.49 ([45]). *Given $I \in \mathcal{D}(\mathbb{N}^d)$. The projection P_I can be written in the form*

$$P_I = \sum_{\gamma \in I} c_\gamma P_\gamma.$$

For any $\gamma \in I$, the coefficient

$$c_\gamma = 1 - \sum_{\gamma < \alpha \in I} c_\alpha.$$

Proof. We first apply Proposition 2.46

$$P_I = \sum_{\gamma \in I} c_\gamma P_\gamma = \sum_{\gamma \in I} c_\gamma \sum_{\alpha \leq \gamma} \Delta_\alpha.$$

Next, using Lemma 2.48, for any $\eta \in I$, we get

$$\Delta_\eta = \Delta_\eta P_I = \sum_{\gamma \in I} c_\gamma \sum_{\alpha \leq \gamma} \Delta_\eta \Delta_\alpha = \sum_{\eta \leq \gamma \in I} c_\gamma \Delta_\eta$$

and therefore

$$1 = \sum_{\eta \leq \gamma \in I} c_\gamma = c_\eta + \sum_{\eta < \gamma \in I} c_\gamma.$$

After changing the indices, we get the desired result. □

Moreover, if we further define the characteristic function of a set of indices I,

$$\chi_I(\gamma) := \begin{cases} 1, & \text{if } \gamma \in I, \\ 0, & \text{otherwise,} \end{cases}$$

we can express the combination coefficients using the characteristic function.

Proposition 2.50 ([48, 45])**.** *Given $I \in \mathcal{D}(\mathbb{N}^d)$. The projection P_I can be written in the form*

$$P_I = \sum_{\gamma \in I} c_\gamma P_\gamma.$$

For any $\gamma \in I$, the coefficient

$$c_\gamma = \sum_{\gamma \leq \alpha \leq \gamma + 1} (-1)^{|\alpha - \gamma|} \chi_I(\alpha).$$

Proof. First we have for any $\gamma \in \mathbb{N}^d$

$$\chi_I(\gamma) = \sum_{\gamma \leq \alpha} c_\alpha \chi_I(\alpha).$$

This is a direct result from Lemma 2.49

$$1 = \sum_{\gamma \leq \alpha \in I} c_\alpha = \sum_{\gamma \leq \alpha} c_\alpha \chi_I(\alpha).$$

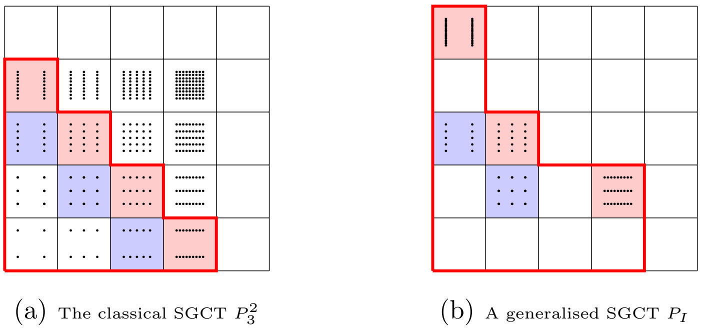
By direct computation, we have

$$\begin{aligned} \sum_{\gamma \leq \alpha \leq \gamma+1} (-1)^{|\alpha-\gamma|} \chi_I(\alpha) &= \sum_{\gamma \leq \alpha \leq \gamma+1} (-1)^{|\alpha-\gamma|} \sum_{\alpha \leq \eta} c_\eta \chi_I(\eta) \\ &= \left(\sum_{\eta_1 = \gamma_1}^{\infty} - \sum_{\eta_1 = \gamma_1 + 1}^{\infty} \right) \cdots \left(\sum_{\eta_d = \gamma_d}^{\infty} - \sum_{\eta_d = \gamma_d + 1}^{\infty} \right) c_\eta \chi_I(\eta) \\ &= c_\gamma \chi_I(\gamma). \end{aligned}$$

□

In particular, if $d = 2$ and $I = \{\gamma \,|\, |\gamma| \leq n\}$, we will have

$$P_I(f) = P_n^c(f) = \sum_{\gamma_1 + \gamma_2 = n+1} P_{\gamma_1, \gamma_2}(f) - \sum_{\gamma_1 + \gamma_2 = n} P_{\gamma_1, \gamma_2}(f)$$


(a) The classical SGCT P_3^2

(b) A generalised SGCT P_I

Figure 2.3: The figures show how to apply the proposition 2.50 to the computation of (generalised) combination technique approximations. In each case, the final approximation is obtained by adding all the red block approximations and subtracting all the blue block approximations. The red contour line shows the blocks with indices included in the set I in (2.11).

which is the 2D classical sparse grid combination technique. In Figure 2.3, we compare it (when $n = 3$) with a generalised 2D combination technique.

We can also compute the updating combination coefficients in Proposition 2.45 by using the surplus operator.

Proposition 2.51 ([45]). *Let I, $J \in \mathcal{D}(\mathbb{N}^d)$ such that $J = I \cup \{\gamma\}$. Then*

$$P_J - P_I = \sum_{\gamma-1 \leq \alpha \leq \gamma} (-1)^{|\gamma-\alpha|} P_\alpha$$

where $P_\alpha := 0$ if any of the $j_k \leq 0$.

Proof. Using the definition of the surplus operator, we have

$$\begin{aligned} P_J - P_I = \Delta_\gamma = \bigotimes_{k=1}^{d} \Delta_{\gamma_k} &= \bigotimes_{k=1}^{d} (P_{\gamma_k} - P_{\gamma_k - 1}) \\ &= \sum_{\alpha \leq 1} (-1)^\alpha \bigotimes_{k=1}^{d} P_{\gamma_k - \alpha_k} = \sum_{\alpha \leq 1} (-1)^{|\alpha|} P_{\gamma-\alpha} \\ &= \sum_{\gamma-1 \leq \alpha \leq \gamma} (-1)^{|\gamma-\alpha|} P_\alpha. \end{aligned}$$

□

For the combination coefficients, we have the following properties.

Remark 2.52. The combination coefficients $c_\gamma = 0$, $\gamma \in I$ if $\gamma + 1 \in I$. We have $\alpha \in I$ for any $\alpha \leq \gamma + 1$ according to the condition. Using proposition 2.50, we have

$$c_\gamma = \sum_{\gamma \leq \alpha \leq \gamma+1} (-1)^{|\alpha-\gamma|} \chi_I(\alpha) = \sum_{\gamma \leq \alpha \leq \gamma+1} (-1)^{|\alpha-\gamma|} = 0.$$

Remark 2.53. If $\emptyset \neq I \in \mathcal{D}(\mathbb{N}^d)$, then the combination coefficients satisfy

$$\sum_{\gamma\in I} c_\gamma = 1.$$

Since $I \neq \emptyset$, we have $0 \in I$. Use the fact $\Delta_0 = P_0$

$$P_0 = \Delta_0 = \Delta_0 P_I = P_0 P_I = P_0 \sum_{\gamma\in I} c_\gamma P_\gamma = P_0 \sum_{\gamma\in I} c_\gamma$$

and therefore $\sum_{\gamma\in I} c_\gamma = 1$.

Remark 2.54. In particular, if we take $I = \{\gamma \,|\, |\gamma| \leq n\}$ in the generalised combination technique $P_I = \sum_{\gamma\in I} c_\gamma P_\gamma$, the combination technique can be computed using Proposition 2.50

$$c_\gamma = \sum_{\gamma\leq\alpha\leq\gamma+1} (-1)^{|\alpha-\gamma|}\chi_I(\alpha) = \sum_{l=0}^{n-|\gamma|} (-1)^l \binom{d}{l}.$$

Using the recursion relation

$$\binom{d}{l} - \binom{d-1}{l-1} = \binom{d-1}{l},$$

we finally get

$$c_\gamma = (-1)^{n-|\gamma|}\binom{d-1}{n-|\gamma|}.$$

Next, we show the convergence of the generalised combination technique under suitable assumptions.

Proposition 2.55. *Suppose $f \in U \subset L^2(\Omega)$, $\Omega \subset \mathbb{R}^d$. Let U_γ, $\gamma \in \mathbb{N}^d$ be a sequence of hierarchical finite dimensional subspaces of U. Let P_γ be orthogonal projections from $U \to U_\gamma$. Furthermore, we assume the surpluses $\Delta_\gamma f$ satisfy*

$$\Delta_\gamma(f)(x) = C(x,\gamma)\, 2^{-\gamma_1 p_1} \dots 2^{-\gamma_d p_d}$$

for some $p_k \in \mathbb{R}^+$, $k = 1, \dots, d$ and $|C(x,\gamma)| \leq K$ for any $x \in \Omega$ and $\gamma \in \mathbb{N}^d$. Then we have the following error bound for the generalised combination technique.

$$\|P_I(f) - f\|_2 \leq K \left(\prod_{k=1}^{d} \frac{2^{p_k}}{2^{p_k}-1} - \sum_{\gamma\in I} 2^{-\gamma_1 p_1} \dots 2^{-\gamma_d p_d} \right).$$

Proof. We first have

$$\|P_I(f) - f\|_2 = \|\sum_{\gamma\in I} \Delta_\gamma(f) - \sum_{\gamma\in\mathbb{N}^d} \Delta_\gamma(f)\|_2 \leq \sum_{\gamma\in\mathbb{N}^d\setminus I} \|\Delta_\gamma(f)\|_2.$$

Using the assumption on surpluses, we have $\|\Delta_\gamma(f)\|_2 \leq K2^{-\gamma_1 p_1}\dots 2^{-\gamma_d p_d}$ and therefore

$$\begin{aligned}\|P_I(f) - f\|_2 &\leq K \sum_{\gamma\in\mathbb{N}^d\setminus I} 2^{-\gamma_1 p_1}\dots 2^{-\gamma_d p_d}\\ &= K\left(\sum_{\gamma\in\mathbb{N}^d} 2^{-\gamma_1 p_1}\dots 2^{-\gamma_d p_d} - \sum_{\gamma\in I} 2^{-\gamma_1 p_1}\dots 2^{-\gamma_d p_d}\right)\\ &= K\left(\prod_{k=1}^{d} \frac{2^{p_k}}{2^{p_k}-1} - \sum_{\gamma\in I} 2^{-\gamma_1 p_1}\dots 2^{-\gamma_d p_d}\right).\end{aligned}$$

as required. □

If we have a sequence of downsets $I_t \in \mathcal{D}(\mathbb{N}^d)$, $t \in \mathbb{N}$ such that

$$I_1 \subset I_2 \subset \dots \subset I_t \subset \dots,$$

$P_{I_t}(f)$, $t \in \mathbb{N}$ does not necessarily converge to f as $t \to \infty$ or $|I_t| \to \infty$. For example, in 2D case, if we take

$$I_t = \{\gamma, \gamma \leq (0,t) \,|\, t \in \mathbb{N}\}$$

and assume we have the following error splitting model

$$f(x) - P_\gamma(f)(x) = C_1(x,\gamma_1)2^{-\gamma_1 p} + C_2(x,\gamma_2)2^{-\gamma_2 p} + D_{1,2}(x,\gamma_1,\gamma_2)2^{-\gamma_1 p}2^{-\gamma_2 p},$$

then when $t \to \infty$ or $|I_t| \to \infty$, we have

$$\lim_{t\to\infty} \|f - P_{I_t}(f)\| = \|f - P_{(0,\infty)}\| = \|C_1(\cdot,0)\|_2.$$

$\|C_1(\cdot,0)\|_2$ does not necessarily equal to 0. Therefore, $P_{I_t}(f)$, $t \in \mathbb{N}$ can not converge to f. In order to exclude such cases and further study the convergence of the generalised combination technique, we define the following effective sequence of downsets

Definition 2.56. Suppose we have a sequence of downsets $I_t \in \mathcal{D}(\mathbb{N}^d)$, $t \in \mathbb{N}$ such that

$$I_1 \subset I_2 \subset \dots \subset I_t \subset \dots.$$

If there exists a subsequence I_{t_r}, $r \in \mathbb{N}$ of I_t such that

$$S_0^d \subset I_{t_0} \subset S_1^d \subset I_{t_1} \subset S_2^d \subset I_{t_2} \subset \dots$$

where the auxiliary sequence of downsets S_r^d, $r \in \mathbb{N}$ is defined as

$$S_r^d = \{\gamma \,|\, \gamma \leq r\mathbf{1}_d\},$$

then the sequence $I_t \in \mathcal{D}(\mathbb{N}^d)$, $t \in \mathbb{N}$ will be called an effective sequence of downsets.

By using this concept, we can prove the convergence of the generalised combination technique.

Proposition 2.57. *Suppose $f \in U \subset L^2(\Omega)$, $\Omega \subset \mathbb{R}^d$. Let U_γ, $\gamma \in \mathbb{N}^d$ be a sequence of hierarchical finite dimensional subspaces of U. Let P_γ be orthogonal projections from $U \to U_\gamma$. Furthermore, we assume $P_\gamma(f)$ satisfy*

$$f(x) - P_\gamma(f)(x) = \sum_{k=1}^{d} \sum_{\{j_1,\dots,j_k\}\subset\{1,\dots,d\}} C_{j_1\dots j_m}(x, \gamma_{j_1}, \dots, \gamma_{j_m}) 2^{-\gamma_{j_1}p} \dots 2^{-\gamma_{j_k}p}. \tag{2.13}$$

for some $p \in \mathbb{R}^+$ and

$$|C_{j_1\dots j_m}(x, \gamma_{j_1}, \dots, \gamma_{j_m})| \leq K$$

for any $x \in \Omega$ and $\gamma \in \mathbb{N}^d$. Suppose $I_t \in \mathcal{D}(\mathbb{N}^d)$, $t \in \mathbb{N}$ is an effective sequence of downsets. The generalised combination technique with downset I_t is

$$P_{I_t} = \sum_{\gamma_t \in I_t} c_{\gamma_t} P_{\gamma_t}.$$

Then we have

$$\lim_{t\to\infty} \|f - P_{I_t}(f)\|_2 = 0.$$

Proof. According to the definition, we first have

$$P_{r-1}(f) - f + \sum_{\gamma \in I_{t_{r-1}} \setminus S_{r-1}^d} \Delta_\gamma(f) = P_{I_{t_{r-1}}}(f) - f$$

$$P_{I_{t_{r-1}}}(f) - f + \sum_{\gamma \in S_r^d \setminus I_{t_{r-1}}} \Delta_\gamma(f) = P_r(f) - f.$$

Using the triangle inequality and the reverse triangle inequality, we have

$$\|P_r(f) - f\|_2 - \left\| \sum_{\gamma \in S_r^d \setminus I_{t_{r-1}}} \Delta_\gamma(f) \right\|_2 \leq \|P_{I_{t_{r-1}}}(f) - f\|_2$$

$$\|P_{I_{t_{r-1}}}(f) - f\|_2 \leq \|P_{r-1}(f) - f\|_2 + \left\| \sum_{\gamma \in I_{t_{r-1}} \setminus S_{r-1}^d} \Delta_\gamma(f) \right\|_2 .$$

Applying the triangle inequality to the sum of the surpluses and using

$$\begin{gathered} S_r^d \setminus I_{t_{r-1}} \subset S_r^d \setminus S_{r-1}^d \\ I_{t_{r-1}} \setminus S_{r-1}^d \subset S_r^d \setminus S_{r-1}^d, \end{gathered}$$

we get

$$\begin{gathered} \|P_r(f) - f\|_2 - \sum_{\gamma \in S_r^d \setminus S_{r-1}^d} \|\Delta_\gamma(f)\|_2 \leq \|P_{I_{t_{r-1}}}(f) - f\|_2 \\ \|P_{I_{t_{r-1}}}(f) - f\|_2 \leq \|P_{r-1}(f) - f\|_2 + \sum_{\gamma \in S_r^d \setminus S_{r-1}^d} \|\Delta_\gamma(f)\|_2. \end{gathered}$$

Using the error splitting model, we can prove that[‡]

$$\|\Delta_\gamma(f)\|_2 \leq L 2^{-\gamma_1 p} \ldots 2^{-\gamma_d p}.$$

for some $L \in \mathbb{R}^+$. Based on this bound, the largest $\|\Delta_\gamma(f)\|$ on $S_r^d \setminus S_{r-1}^d$ is 2^{-rp} and therefore

$$\sum_{\gamma \in S_r^d \setminus S_{r-1}^d} \|\Delta_\gamma(f)\|_2 \leq L[r^d - (r-1)^d]2^{-rp}.$$

As $r \to \infty$, by using squeeze Theorem [78], we have

$$\|P_{I_{t_r}}(f) - f\|_2 \to 0.$$

Now we consider the sequence I_t, $t \in \mathbb{N}$. For any $t \in \mathbb{N}$, let

$$\begin{gathered} i := \max_k \{k \in \mathbb{N}, I_{t_k} \subset I_t\} \\ j := \min_k \{k \in \mathbb{N}, I_t \subset I_{t_k}\}. \end{gathered}$$

Then we have

$$S_{t_{i-1}}^d \subset I_{t_i} \subset I_t \subset I_{t_j} \subset S_{t_{j+1}}^d.$$

Consequently, we have

$$\begin{gathered} \|P_{I_{t_j}} f - f\|_2 - \sum_{\gamma \in S_{t_{j+1}}^d \setminus S_{t_{i-1}}^d} \|_2 \Delta_\gamma f\| \leq \|P_{I_t}(f) - f\|_2 \\ \|P_{I_t}(f) - f\|_2 \leq \|P_{I_{t_i}} f - f\|_2 + \sum_{\gamma \in S_{t_{j+1}}^d \setminus S_{t_{i-1}}^d} \|_2 \Delta_\gamma f\| \end{gathered}$$

and

$$\sum_{\gamma \in S_{t_{j+1}}^d \setminus S_{t_{i-1}}^d} \|\Delta_\gamma f\|_2 \leq L((j+1)^d - (i-1)^d)2^{-(i-1)p}.$$

[‡] We give a proof for 2D case in Chapter 3. d dimensional case can be proved by using induction.

Therefore, using the convergence of the subsequence and applying the squeeze theorem again, we have

$$\lim_{t\to\infty} \|f - P_{I_t}(f)\|_2 = 0.$$

□

Remark 2.58. In practice, the sequence of downsets I_t, $t \in \mathbb{N}$ can be selected by

$$I_t = \{\gamma \mid \|\Delta_\gamma(f)\|_2 \geq \epsilon_t\}$$

where ϵ_t, $t \in \mathbb{N}$ is a sequence of given decreasing thresholds. When we generate I_t, $t \in \mathbb{N}$ by the above strategy, we also need to make sure the newly generated I_t is still a downset. The algorithm can be find in [38, 46].

Remark 2.59. In fact, we can also replace the auxiliary sequence of downsets S^d_r, $r \in \mathbb{N}$ with other sequences of downsets in the definition, e.g. the sequence of rectangular downsets

$$R^d_\alpha = \left\{\gamma \in \mathbb{N}^d \mid \gamma \leq \alpha\right\},$$

or even a sequence of downsets of a sparse grid

$$SG^d_n = \left\{\gamma \in \mathbb{N}^d \mid |\gamma| \leq n\right\}.$$

If we choose I_t, $t \in \mathbb{N}$ using the strategy in Remark 2.58, a good choice of the auxiliary sequence of downsets can help us estimate the convergence rate of the generalised combination technique theoretically according to the proof of Proposition 2.57. We can construct such auxiliary sequence of downsets based on the error splitting model of the problem if we know it before the computation.

We can obtain a concrete (generalised) combination technique after we choose specific spaces U_γ and projections P_γ. Commonly used choices of U_γ and P_γ are shown as examples below. The first example is the Lagrangian interpolation. Suppose the domain of function f is $\Omega = [0,1]^d$ and G_γ is regular d dimensional grid with spacing $h_{\gamma_k} = 2^{-\gamma_k}$ in the kth coordinate. The d dimensional basis function is defined as the tensor product of 1D Lagrangian basis functions l_{γ_k,i_k}

$$l_{\gamma,i}(x) := \otimes_{k=1}^d l_{\gamma_k,i_k}(x_k).$$

U_γ is the space spanned by $l_{\gamma,i}$. We take the projections P_γ as Lagrangian interpolation operators L_γ

$$L_\gamma(f) = \sum_{0\leq i\leq 2^\gamma} f(x_{\gamma,i}) l_{\gamma,i}(x).$$

Therefore the generalised combination technique for the Lagrangian interpolation is

$$L_I(f) = \sum_{\gamma \in I} c_\gamma L_\gamma(f).$$

The second example is the Newton Cotes formulas. The Newton Cotes formulas are quadrature rules of interpolatory type, i.e.

$$Q_\gamma(f) = \int_\Omega L_\gamma(f)(x)\,dx$$

where L_γ here are the Lagrangian interpolation associated to the Newton Cotes formulas Q_γ. Using linearity of integral, we can define the generalised sparse grid combination technique for computing the integral

$$Q_I(f) = \int_\Omega P_I(f)(x)\,dx = \int_\Omega \sum_{\gamma \in I} c_\gamma P_\gamma f(x)\,dx = \sum_{\gamma \in I} c_\gamma Q_\gamma(f).$$

2.5 Conclusions

In this Chapter, we review the classical combination technique and the generalised combination technique. The convergence of the combination technique, either the classical one or the generalised one, is based on the error splitting models. A lot of research has been done to find the error splitting models for approximations on equally spaced grids. Here we explore a little on error models on unequally spaced grids. For frequently used CGL points and the Kronrod's scheme, we prove we can construct an error splitting model with bounded coefficients. For the generalised combination technique, we show its convergence is closely related to the convergence of the classical combination technique(sparse grid). Therefore, find an error splitting model is also crucial to prove its convergence.

Chapter 3

Sparse Grid Combination Technique applied to Functionals

In many real world problems, people are more interested in some important functionals(Quantities of interest [85]) of the solution of a problem rather than the solution itself. For example, in an uncertainty quantification(UQ [85]) problem

$$A(u(\cdot,w),w)=0,\ w\in W \tag{3.1}$$

where W is a parameter space and $u(\cdot,w):\Omega\to\mathbb{R}$ is the solution to the problem when we take parameter w, people are more interested in computing the following moments

$$F(u)(x)=\mathbb{E}_W(u^k(x,\cdot)),\ k\in\mathbb{Z}^+.$$

In the above uncertainty quantification problem, in order to obtain an accurate approximation of these moments, we need to solve the equation many times. This makes the whole UQ problem expensive to compute when W is high dimensional. Numerical methods such as Quasi Monte Carlo methods [57, 41, 56], sparse grid methods and sparse grid combination techniques [79, 85] are widely used here to reduce the cost of the computation of such UQ problems. While functionals related to a randomised problem are expensive to compute, for some deterministic problems, the computational cost of the related functionals can also be unaffordable. In addition, such computation appears frequently in real world applications.

In this Chapter, we discuss the following two-stage approximation problem. Suppose $u\in U\subset X$ is the solution of a partial differential equation

$$A(u)=0.$$

We compute the following functional

$$\begin{aligned} T : U &\to \mathbb{R} \\ u &\mapsto T(u). \end{aligned}$$

Such functional can be important quantity used to capture feature of the problem. It widely appears in applied mathematics and Physics. When the problem is multidimensional, the computational cost of the functional can be large since the numerical solution of a multidimensional partial differential equation(e.g. 3D to 100D) is usually expensive to compute. We apply the generalised combination technique to the problem. The generalised combination technique can reduce the computational cost of the functional. It depends on the error splitting model of the problem. We study the error splitting models of such problem when numerical schemes used to compute the PDEs and the functionals are known. Besides the error splitting model, it is more convenient to design and analyse the generalised combination technique by studying the decay of the surpluses for many problems. We explore the connections between the error splitting model and the surpluses decay model. By using the connection, we can also apply the generalised combination technique to the problems when we only know the computed surpluses.

This Chapter is organised as follows. We first introduce the space we work with, the Banach space and the Hilbert space and the functional defined on these spaces. We then further study the differential calculus and the Taylor expansion on these spaces. Next, we work through the PDE examples and numerical schemes to solve the PDEs. We also review some important results on the error splitting models for the PDE examples. After that, we study the error splitting model for the two-stage approximation problem. We then build the connection between the surpluses decay model and the error splitting model. Finally, we show a few numerical experiments to illustrate our method.

3.1 Differential Calculus in Banach Space and Hilbert Space

3.1.1 Banach Space

Definition 3.1. A Banach space is a complete vector space V with a norm $\|\cdot\|$. The norm is a real valued function defined on space V such that

- $\|v\| \geq 0, v \in V$,
- $\|av\| = |a|\|v\|,\ v \in V,\ a \in \mathbb{R}$,
- $\|v + w\| \leq \|v\| + \|w\|,\ v, w \in V$,
- If $\|v\| = 0,$ then $v = 0$.

Definition 3.2. Let X, Y be two Banach space, we denote the set of all the linear continuous operators from X to Y as $L(X, Y)$. In particular, if $Y = \mathbb{R}$, then $L(X, \mathbb{R})$ contains all functionals on Banach space X.

As we define the differential and the derivatives in the Euclidean spaces, we define the following Fréchet differential and Fréchet derivatives in the Banach spaces.

Definition 3.3. Let X, Y be two Banach space. F maps X to Y. Suppose U is an open subset of X and $u \in U$. Then F is Fréchet differentiable at u if there exists $B \in L(X, Y)$ such that, if we set

$$R(h) = F(u + h) - F(u) - B(h),$$

then

$$R(h) = o(\|h\|),$$

i.e.

$$\frac{\|R(h)\|}{\|h\|} \to 0 \text{ as } \|h\| \to 0.$$

The operator B is unique and is called the Fréchet differential of F at u and denoted by

$$B = dF(u).$$

If F is differentiable at all $u \in U$, then F is said to be differentiable in U.

Definition 3.4. Suppose $F : U \to Y$ is differentiable in U. We define

$$\begin{aligned} F^{'} : U &\to L(X, Y) \\ u &\mapsto dF(u) \end{aligned}$$

as the Fréchet derivative of F. We further define $C^1(U, Y)$ as a set which contains all $F : U \to Y$ with continuous Fréchet derivative in U

The following properties of Fréchet derivative are the same as those of the derivative in Euclidean space.

- Linearity: Given Fréchet differentiable functions F, $G : U \to Y$, $u \in U$. Then for any $a, b \in \mathbb{R}$
$$(aF + bG)^{'}(u) = aF^{'}(u) + bG^{'}(u).$$

- Chain rule: Given $F : U \to Y$ and $G : V \to Z$ with $F(U) \subset V$. U and V are open subsets of X and Y, respectively. F is differentiable at $u \in U$ and G is differentiable at $v := F(u) \in V$. Then the composite operator $F \circ G$ is also differentiable and
$$(G \circ F)^{'}(u) = G^{'}(v)F^{'}(u).$$

The proofs of these two properties are similar as the proof in Euclidean space and can be found in [7].

We can also define higher order Fréchet differential and derivative. We start with the definition of the twice differential and second derivative.

Definition 3.5. Let X, Y be two Banach space. F maps X to Y. Suppose U is an open subset of X and $u \in U$. Let F be differentiable in U. Then F is twice Fréchet differentiable at u if $F^{'}$ is differentiable at u. The second Fréchet differential of F at u is defined as
$$d^2F(u) = dF^{'}(u).$$
If F is twice differentiable at all $u \in U$, then F is said to be differentiable in U

From the definition, we know that $d^2F(u)$ is a linear continuous map from X to $L(X, Y)$, therefore we have
$$d^2F(u) \in L(X, L(X, Y)).$$
If we further define $L_2(X, Y)$ as the space of all continuous bilinear maps from $X \times X$ to Y, we can prove that the space $L(X, L(X, Y))$ and the space $L_2(X, Y)$ are isometric [2]. Therefore, the twice differential operator $d^2F(u)$ can also be viewed as a continuous bilinear map. The value of $d^2F(u)$ at pair (h, k) is denoted by $d^2F(u)[h, k]$.

Definition 3.6. Suppose F is twice differentiable in U. We define
$$F^{''} : U \to L_2(X, Y)$$
$$u \mapsto d^2F(u)$$
as the second Fréchet derivative of F. We define $C^2(U, Y)$ as a set which contains all $F : U \to Y$ with continuous second Fréchet derivative in U.

If the nth derivative is given, we can define the $(n+1)$th Fréchet derivative by induction. Let $L_n(X,Y)$ be the space of all continuous n-linear map from $X \times X \times \cdots \times X(n$ times) to Y. If $F : U \to Y$ is n times differentiable in U, we denote its nth Fréchet derivative as

$$\begin{aligned} F^{(n)} : U &\to L_n(X,Y) \\ u &\mapsto d^n F(u). \end{aligned}$$

The $(n+1)$th differential at u is defined by induction

$$d^{(n+1)}F(u) = dF^n(u).$$

According to the definition, we know that $d^{n+1}F(u) \in L(X, L_n(X,Y))$. Let $L_{n+1}(X,Y)$ be the space of all continuous $(n+1)$-linear map from $X \times X \times \cdots \times X$((n+1) times) to Y. We can similarly prove the space $L(X, L_n(X,Y))$ and the space $L_{n+1}(X,Y)$ are isometry.

If the nth Fréchet derivative is continuous, then we denote $F \in C^n(U,Y)$. The value of $d^nF(u)$ at $(h_1, \ldots, h_n)$ is

$$d^nF(u)[h_1, \ldots, h_n].$$

In particular, if $h_1 = \cdots = h_n = h$, we write it in a conciser form $d^nF(u)[h]^n$.

In Euclidean spaces, we use Taylor's theorem to approximate an n times differentiable function around a given point by a polynomial of degree n. In Banach spaces, we also have similar Taylor's theorem.

Theorem 3.7 (Adapted from [2]). *Suppose $F \in C^n(Q,Y)$ and u, $u+v \in Q$ such that the interval $[u, u+v] \subset Q$. Then we have the following Taylor expansion*

$$F(u+v) = F(u) + dF(u)[v] + \frac{1}{2!}d^2F(u)[v]^2 + \cdots + \frac{1}{n!}d^nF(u)[v]^n + \epsilon(u,v)[v]^n$$

where the operator $\epsilon(u,v)$ in the remainder term is defined as

$$\epsilon(u,v) = \frac{1}{(n-1)!}\int_0^1 (1-t)^{n-1}[d^{(n)}F(u+tv) - d^{(n)}F(u)]\,dt, \qquad (3.2)$$

and

$$\epsilon(u,v) \to 0 \text{ as } v \to 0. \qquad (3.3)$$

Proof. Let $\gamma(t) = u + tv$, $t \in [0,1]$ and define function

$$\begin{aligned} \Phi : [0,1] &\to Y \\ t &\mapsto F(\gamma(t)). \end{aligned}$$

Using the chain rule and the definition of the higher derivatives, we have

$$\begin{aligned} \Phi^{'}(t) &= dF(u+tv)[v], \\ \Phi^{''}(t) &= d^2F(u+tv)[v]^2, \\ &\vdots \\ \Phi^{n}(t) &= d^nF(u+tv)[v]^n. \end{aligned} \tag{3.4}$$

Applying Taylor expansion of function Φ at 0 in $\mathbb{R}$, we have

$$\begin{aligned} \Phi(1) =& \Phi(0) + \Phi^{'}(0) + \frac{1}{2!}\Phi^{''}(0) + \cdots + \frac{1}{(n-1)!}\Phi^{(n)}(0) \\ &+ \frac{1}{(n-1)!}\int_0^1 (1-t)^{n-1}\Phi^{(n)}(t)\, dt. \end{aligned} \tag{3.5}$$

Therefore, combining (3.4) and (3.5), we have

$$\begin{aligned} F(u+v) =& F(u) + dF(u)[v] + \frac{1}{2!}d^2F(u)[v]^2 + \cdots + \dots \\ & \frac{1}{(n-1)!}\int_0^1 (1-t)^{n-1}d^{(n)}F(u+tv)[v]^n\, dt. \end{aligned}$$

Using definition (3.2), the last integral can be written as

$$\begin{aligned} & \frac{1}{(n-1)!}\int_0^1 (1-t)^{n-1}d^{(n)}F(u+tv)dt[v]^n \\ =& \frac{1}{n!}d^nF(u)[v]^n + \epsilon(u,v)[v]^n. \end{aligned}$$

Since $F \in C^n(Q,Y)$, $d^{(n)}F(u)$ is continuous. Therefore, we have (3.3). □

3.1.2 Hilbert Space

In many applications, we work with a special type of the Banach Space, the Hilbert Space.

Definition 3.8. A Hilbert space is a complete vector space V with an inner product $\langle \cdot, \cdot \rangle$. The inner product is a binary operation defined on the space V such that

- $\langle u, v\rangle = \langle v, u\rangle,\ u, v \in V,$
- $\langle au_1 + bu_2, v\rangle = a\langle u_1, v\rangle + b\langle u_2, v\rangle,\ u_1, u_2, v \in V,$
- $\langle v, v\rangle > 0,\ \text{if } v \neq 0, \langle v, v\rangle = 0,\ \text{if } v = 0$

The norm on a Hilbert space can be induced by the inner product on it, i.e.

$$\|v\| = \langle v, v\rangle^{\frac{1}{2}},\ v \in V.$$

One can check such norm satisfies the norm conditions in definition 3.1. Hence a Hilbert space is a Banach space. Therefore, the differential calculus and the Taylor expansion also hold in a Hilbert space.

Definition 3.9. If X is a normed space, we denote

$$X^* = L(X, \mathbb{R})$$

as the dual space of X. X^* is itself a normed space with the operator norm.

Theorem 3.10 (Riesz representation theorem)**.** *If T is a bounded linear functional on a Hilbert space H, then there is a unique element $y \in H$ such that*

$$T(x) = \langle y, x\rangle,\ \forall x \in H.$$

Proof. See in [12]. □

Using the Riesz representation theorem, we can prove that a real Hilbert space is self-dual [87]. In particular, we consider a functional $T : U \to \mathbb{R}$ where $U \subset H$ and H is a Hilbert space. Suppose T is Fréchet differentiable, then according to the definition, we have

$$\begin{aligned} T^{'} :\ & U \to L(H, \mathbb{R}) \\ & u \mapsto T^{'}(u). \end{aligned}$$

Since we have $H = L(H, \mathbb{R})$, the Fréchet derivative $T^{'}(u) \in H$. If the functional T is twice Fréchet differentiable, then we have

$$\begin{aligned} T^{''} :\ & U \to L(H, L(H, \mathbb{R})) \\ & u \mapsto T^{''}(u). \end{aligned}$$

Therefore the second Fréchet derivative $T^{''}(u) \in L(H, H)$. By using induction, the nth Fréchet derivative $T^n(u) \in L_{(n-1)}(H, H)$.

In the previous Chapters, the spaces $C^s(X)$, $C^s_{mix}(X)$ are Banach spaces. The spaces $H^s(X)$, $H^s_{mix}(X)$, $H^s_0(X)$,, $L_2(X)$ are Hilbert spaces.

3.2 Examples and Numerical Schemes

In this section, we discuss three examples we will use in numerical experiments and the numerical methods to solve these examples.

3.2.1 Galerkin Method

The first one is the following 2D Poisson equation with Dirichlet boundary condition.

$$\begin{cases} -\Delta u = f \text{ in } [0,1]^2 \\ \quad u = 0 \text{ on } \partial[0,1]^2 \end{cases}$$

where $f \in L^2([0,1]^2)$. We use Galerkin method to solve this problem. The weak form of the problem is we find $u \in H_0^1([0,1]^2)$ such that

$$\langle \nabla u, \nabla v \rangle = \langle f, v \rangle, \ \forall v \in H_0^1([0,1]) \times H_0^1([0,1]). \tag{3.6}$$

In order to compute it, we first look into the underlying abstract problem, i.e. find $u \in H$ where H is a Hilbert space such that

$$a(u,v) = b(v), \ \forall v \in H.$$

Here $a(u,v)$ is a continuous bilinear form defined on $H \times H$ and $b(v)$ is a continuous linear functional on H. If we further assume the bilinear form is H-elliptic, then we can prove the abstract problem has a unique solution.

Definition 3.11. A bilinear form is H-elliptic(strict coercivity) if there exists an $\alpha > 0$ such that for all $v \in H$ one has

$$a(v,v) \geq \alpha \|v\|_H^2.$$

Theorem 3.12 (Lax-Milgram)**.** *Let H be a Hilbert space, $a(\cdot,\cdot)$ a continuous H-elliptic bilinear form on H and $b \in H^*$. Then there exists exactly one $u \in H$ such that*

$$a(u,v) = b(v), \ \forall v \in H.$$

Proof. See in [12]. □

According to the Lax-Milgram theorem, in order to prove the existence and the uniqueness of the solution to the weak form of 2D Poisson equation, we only need to check if the bilinear form

$$a(v,v) = \langle \nabla v, \nabla v \rangle, \ \forall v \in H \tag{3.7}$$

is H-elliptic.

Theorem 3.13 (Poincaré-Friedrichs Inequality)**.** *Suppose Ω is contained in an n-dimensional cube with side length s. Then*

$$\|v\|_2 \leq s\|v\|_{H_0^1}, \ \forall v \in H_0^1(\Omega). \tag{3.8}$$

Proof. See in [12]. □

The H-ellipticity of the bilinear form a follows immediately from the Poincaré-Friedrichs Inequality. Therefore, the solution of the weak problem (3.6) exists and it is unique.

For the abstract formulation, suppose H_γ is a finite dimension subspace of H. Our aim is to find the finite dimension approximation $u_\gamma \in H_\gamma$ such that it satisfies the following Galerkin equation

$$a(u_\gamma, v_\gamma) = b(v_\gamma),\ \forall v_\gamma \in H_\gamma.$$

In order to find suitable finite dimension space for the 2D Poisson problem, we first consider the 1D case

$$\begin{cases} -u^{''} = f \text{ in } [0,1] \\ \quad u = 0 \text{ on } \partial[0,1]. \end{cases}$$

Its weak form is

$$\langle u^{'}, v^{'} \rangle = \langle f, v \rangle, \forall v \in H_0^1([0,1]). \tag{3.9}$$

We first define the 1D grid on $[0,1]$. Suppose $[0,1]$ is equally spaced by the following grid points

$$0 = x_{\gamma,0} < x_{\gamma,1} < \cdots < x_{\gamma,2^\gamma} = 1.$$

We denote the set contains all these grid points as grid G_γ. The spacing is $h_\gamma = 1/2^\gamma$. We consider the following finite dimensional space

$$H_\gamma = \text{span}\,\{\phi_{\gamma,i},\ i = 0, \dots, 2^\gamma\}$$

where $\phi_{\gamma,i}$ are the nodal basis functions defined on given grid G_γ. Using these basis functions, the left-hand side of the weak form (3.9) can be expressed as

$$\langle u^{'}, v^{'} \rangle = \langle \sum_{i=0}^{2^\gamma} u_{\gamma,i}\phi^{'}_{\gamma,i}, \sum_{i=0}^{2^\gamma} v_{\gamma,i}\phi^{'}_{\gamma,i} \rangle = v^T K u \tag{3.10}$$

where

$$K = \begin{bmatrix} \langle \phi^{'}_{\gamma,0}, \phi^{'}_{\gamma,0} \rangle & \dots & \langle \phi^{'}_{\gamma,0}, \phi^{'}_{\gamma,2^\gamma} \rangle \\ \vdots & \ddots & \vdots \\ \langle \phi^{'}_{\gamma,2^\gamma}, \phi^{'}_{\gamma,0} \rangle & \dots & \langle \phi^{'}_{\gamma,2^\gamma}, \phi^{'}_{\gamma,2^\gamma} \rangle \end{bmatrix}, \quad u = \begin{bmatrix} u_{\gamma,0} \\ \vdots \\ u_{\gamma,2^\gamma} \end{bmatrix}, \quad v = \begin{bmatrix} v_{\gamma,0} \\ \vdots \\ v_{\gamma,2^\gamma} \end{bmatrix}.$$

K is the (global) stiffness matrix. Using different nodal basis functions can result in different stiffness matrices K. Here we consider two cases: linear basis and quadratic basis. The right-hand side can be written as

$$\langle f, v\rangle = \langle f, \sum_{i=0}^{2^\gamma} v_i \phi_{\gamma,i}\rangle = v^T F \tag{3.11}$$

where $F = [\langle f, \phi_{\gamma,i}\rangle, \dots, \langle f, \phi_{\gamma,2^\gamma}\rangle]^T$. Therefore, combining (3.10) and (3.11), we get

$$v^T K u = v^T F,\ \forall v \in H_\gamma. \tag{3.12}$$

The unknowns in this equation actually include the boundary points. However, we know the value on the boundary from the Dirichlet boundary condition. Thus we need to further deal with these boundary points. First, we write the stiffness matrix K as the following block matrix

$$\Big[k_0, \dots, k_{2^\gamma}\Big]$$

where k_j, $j = 0, \dots, 2^\gamma$ is the jth column of the stiffness matrix K. Then we have

$$\begin{aligned} v^T K u = (v^T K) u &= [v^T k_0, \dots, v^T k_{2^\gamma}] u \\ &= v^T k_0 u_0 + \sum_{j=1}^{2^\gamma - 1} v^T k_j u_j + v^T k_{2^\gamma} u_{2^\gamma}. \end{aligned}$$

Hence (3.12) can be written as

$$v^T \sum_{j=0}^{2^\gamma - 1} k_j u_j = v^T (F - k_0 u_0 - k_{2^\gamma} u_{2^\gamma}),\ \forall v \in H_\gamma.$$

Therefore,

$$[k_1, \dots, k_{2^\gamma - 1}] \begin{bmatrix} u_1 \\ \vdots \\ u_{2^\gamma - 1} \end{bmatrix} = F - k_0 u_0 - k_{2^\gamma} u_{2^\gamma} = F.$$

Removing the rows with the boundary information in the matrix $[k_1, \cdots, k_{2^\gamma - 1}]$ and vector F, we finally get the linear system

$$K_{1:2^\gamma - 1, 1:2^\gamma - 1} u_{1:2^\gamma - 1} = F_{1:2^\gamma - 1}.$$

Now we consider the 2D problem. The weak form (3.6) can be further written into

$$\langle u_x, v_x\rangle + \langle u_y, v_y\rangle = \langle f, v\rangle,\ \forall v \in H_0^1([0,1]) \times H_0^1([0,1]). \tag{3.13}$$

We compute the problem on the grid $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ where G_{γ_i}, $i = 1, 2$ is defined as that in the 1D problem. The 2D nodal basis function is the tensor product of 1D nodal basis function. Therefore, the 2D finite dimension subspace H_γ of $H_0^1([0,1]) \times H_0^1([0,1])$ is

$$H_\gamma = H_{\gamma_1,\gamma_2} = \operatorname{span}\{\phi_{\gamma_1,i_1} \otimes \phi_{\gamma_2,i_2} \,|\, i_1 = 0, \dots 2^{\gamma_1},\ i_2 = 0, \dots, 2^{\gamma_2}\}.$$

By using the property of the tensor product, we have

$$\langle \phi'_{\gamma_1,i_1} \otimes \phi_{\gamma_2,i_2}, \phi'_{\gamma_1,j_1} \otimes \phi_{\gamma_2,j_2} \rangle = \langle \phi'_{\gamma_1,i_1}, \phi'_{\gamma_1,j_1} \rangle \otimes \langle \phi_{\gamma_2,i_2}, \phi_{\gamma_2,j_2} \rangle.$$

and

$$\langle \phi_{\gamma_1,i_1} \otimes \phi'_{\gamma_2,i_2}, \phi_{\gamma_1,j_1} \otimes \phi'_{\gamma_2,j_2} \rangle = \langle \phi_{\gamma_1,i_1}, \phi_{\gamma_1,j_1} \rangle \otimes \langle \phi'_{\gamma_2,i_2}, \phi'_{\gamma_2,j_2} \rangle.$$

Therefore, the left-hand side of the weak form can be written as

$$(\operatorname{vec} v)^T (K_{\gamma_1} \otimes M_{\gamma_2} + M_{\gamma_1} \otimes K_{\gamma_2}) \operatorname{vec} u$$

where $\operatorname{vec} v$ and $\operatorname{vec} u$ are vectorisation of the matrix $[v_{ij}]_{2^\gamma+1, 2^\gamma+1}$ and $[u_{i,j}]_{2^\gamma+1,2^\gamma+1}$ respectively. K_{γ_i}, $i = 1, 2$ is the 1D stiffness matrix and M_{γ_i}, $i = 1, 2$ is the 1D (global) mass matrix where

$$M_\gamma = \begin{bmatrix} \langle \phi_{\gamma,0}, \phi_{\gamma,0} \rangle & \cdots & \langle \phi_{\gamma,0}, \phi_{\gamma,2^\gamma} \rangle \\ \vdots & \ddots & \vdots \\ \langle \phi_{\gamma,2^\gamma}, \phi_{\gamma,0} \rangle & \cdots & \langle \phi_{\gamma,2^\gamma}, \phi_{\gamma,2^\gamma} \rangle \end{bmatrix}$$

The right-hand side of the weak form[*] can be written as

$$\langle f, v \rangle = (\operatorname{vec} v)^T \operatorname{vec}([\langle f, \phi_{\gamma_1,j_1} \otimes \phi_{\gamma_2,j_2} \rangle]_{2^{\gamma+1},2^{\gamma+1}})$$

Similar as we did in 1D problem, we move the left-hand side weak form terms which include boundary points to the right and remove the corresponding rows. Finally we obtain a solvable linear system.

3.2.2 Finite Difference Method

We study the following 2D advection equation

$$\frac{\partial u}{\partial t} + a_1 \frac{\partial u}{\partial x} + a_2 \frac{\partial u}{\partial y} = 0,\ (x, y) \in \mathbb{R}^2,\ t > 0.$$

[*] The notation $\otimes$ here is the tensor product of two matrices. We also use $\otimes$ as the tensor product of two functions. Both of them are standard notations

with the initial condition

$$u(x,y,0) = u_0(x,y),\ (x,y) \in \mathbb{R}^2,$$

and the 2D diffusion equation

$$\frac{\partial u}{\partial t} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right),\ (x,y) \in \mathbb{R}^2,\ t > 0,\ \nu > 0$$

with the initial condition

$$u(x,y,0) = u_0(x,y),\ (x,y) \in \mathbb{R}^2.$$

In the computation, concrete boundary conditions are required for both PDEs. For the advection equation, we use the periodic boundary condition on $[0,1]^2$ for both directions. This is generally equivalent to the Cauchy problem with periodic initial data [58]. For the diffusion equation, we restrict the domain on $[0,1]^2$ and use zero boundary condition.

We use the upwind scheme to compute the advection equation. Without loss of generality, we assume a_1, $a_2 > 0$. The upwind scheme is

$$\frac{u_{l,j}^{k+1} - u_{l,j}^k}{\tau} + a_1 \frac{u_{l,j}^k - u_{l-1,j}^k}{h_1} + a_2 \frac{u_{l,j}^k - u_{l,j-1}^k}{h_2} = 0$$

where the domain $[0,1]^2$ is equally spaced in both directions with spacing h_1 and h_2 respectively. τ is the time step size and

$$u_{l,j}^k = u(lh_1, jh_2, k\tau).$$

The truncation error is $O(\tau + h_1 + h_2)$.

We use the forward Euler central difference scheme to compute the diffusion equation. It is

$$\frac{u_{l,j}^{k+1} - u_{l,j}^k}{\tau} = \nu \left(\frac{u_{l+1,j}^k - 2u_{l,j}^k + u_{l-1,j}^k}{h_1^2} + \frac{u_{l,j+1}^k - 2u_{l,j}^k + u_{l,j-1}^k}{h_2^2} \right)$$

The grid setting is the same as that of the upwind scheme. The truncation error is $O(\tau + h_1^2 + h_2^2)$.

Both numerical schemes are consistent. According to the Lax equivalence theorem [58], in order to make sure the numerical schemes are convergent, we also require stability of both schemes. Here we follow the standard Von Neumann analysis [58] and use it to find out the stability condition for both numerical schemes. The idea of the Von Neumann approach is to use the Fourier analysis. The core is to compute the amplification factor [58] $g(\xi)$ where ξ is a single wave number. In 2D case[†], the amplification factor is $g(\xi_1, \xi_2)$. ξ_s, $s = 1, 2$ is single wave number of different direction. If we set

$$u_{lj}^k = e^{ilh_1\xi_1} e^{ijh_2\xi_2}$$

and expect

$$u_{lj}^{k+1} = g(\xi_1, \xi_2) e^{ilh_1\xi_1} e^{ijh_2\xi_2},$$

the amplification factor of the 2D upwind scheme is

$$g(\xi_1, \xi_2) = 1 - \lambda_1(1 - e^{-ih_1\xi_1}) - \lambda_2(1 - e^{-ih_2\xi_2})$$

where

$$\lambda_1 = \frac{a_1\tau}{h_1}, \ \lambda_2 = \frac{a_2\tau}{h_2}.$$

The modulus of the amplification factor is

$$\begin{aligned} &|g(\xi_1, \xi_2)|^2 \\ =&1 - 4(\sqrt{\lambda_1(1-\lambda_1)} \sin\frac{h_1\xi_1}{2} - \sqrt{\lambda_2(1-\lambda_2)} \sin\frac{h_2\xi_2}{2})^2 \\ &- 8 \sin\frac{h_1\xi_1}{2} \sin\frac{h_2\xi_2}{2} (\sqrt{\lambda_1(1-\lambda_1)\lambda_2(1-\lambda_2)} - \lambda_1\lambda_2 \cos(\frac{h_1\xi_1 - h_2\xi_2}{2})) \end{aligned}$$

when $\sin\frac{h_1\xi_1}{2} \sin\frac{h_2\xi_2}{2} \geq 0$, and it is

$$\begin{aligned} &|g(\xi_1, \xi_2)|^2 \\ =&1 - 4(\sqrt{\lambda_1(1-\lambda_1)} \sin\frac{h_1\xi_1}{2} + \sqrt{\lambda_2(1-\lambda_2)} \sin\frac{h_2\xi_2}{2})^2 \\ &+ 8 \sin\frac{h_1\xi_1}{2} \sin\frac{h_2\xi_2}{2} (\sqrt{\lambda_1(1-\lambda_1)\lambda_2(1-\lambda_2)} + \lambda_1\lambda_2 \cos(\frac{h_1\xi_1 - h_2\xi_2}{2})) \end{aligned}$$

when $\sin\frac{h_1\xi_1}{2} \sin\frac{h_2\xi_2}{2} < 0$. For both cases, in order to achieve $|g(\xi_1, \xi_2)|^2 \leq 1$, we require

$$\begin{aligned} &\lambda_1(1-\lambda_1) \geq 0 \\ &\lambda_2(1-\lambda_2) \geq 0 \\ &\sqrt{\lambda_1(1-\lambda_1)\lambda_2(1-\lambda_2)} \geq \lambda_1\lambda_2. \end{aligned}$$

Since $\lambda_1 > 0$ and $\lambda_2 > 0$, we finally get

$$\lambda_1 + \lambda_2 = \frac{a_1\tau}{h_1} + \frac{a_2\tau}{h_2} \leq 1.$$

[†] Since the computation of the amplification factor of the 2D upwind scheme is not in [58] and it is not trivial, the computation is included here.

Similarly, the amplification factor of the 2D forward Euler central difference scheme is

$$g(\xi_1, \xi_2) = 1 - 4\mu_1 \sin^2 \frac{h_1\xi_1}{2} - 4\mu_2 \sin^2 \frac{h_2\xi_2}{2}$$

where

$$\mu_1 = \frac{\nu\tau}{h_1^2},\ \mu_2 = \frac{\nu\tau}{h_2^2}.$$

If

$$\mu_1 + \mu_2 = \frac{\nu\tau}{h_1^2} + \frac{\nu\tau}{h_2^2} \le \frac{1}{2},$$

then the modulus of the amplification factor $|g(\xi_1, \xi_2)| \le 1$.

3.2.3 Gyrokinetic Equations and GENE

In plasma physics, the following Gyrokinetic equations(Vlasov equation)

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} + q(E(f) + \frac{v}{c} \times B(f)) \cdot \frac{\partial f}{\partial v} = 0$$

is used to describe the evolution of the specie in the plasma. Here f is the particle distribution function of the specie at a certain position in real and velocity space(the phase space). It is a seven dimensional function with 3D in real space, 3D in velocity space and 1D in time. c is the speed of light. E and B are electric field and magnetic field respectively. E and B here should also depend on x, v and t. They can be computed from Maxwell's equations. Since it is a high dimensional problem, the time cost to compute this equation is usually very expensive.

We use an existing code GENE to compute it. It is a software package developed to solve the nonlinear gyrokinetic equations in a flux-tube domain. As mentioned in the GENE user's manual [88], it first reduces the original six dimensional phase space to a five dimensional one by removing the fast gyromotion. Then the 'method of lines' is used to solve the reduced PDEs. During the computation, different dimensions are treated differently, e.g. for one of the dimensions in real space, they use the fourth-order Arakawa scheme [3] while for the other two dimensions, they apply a pseudospectral approach. The whole numerical scheme is too complex to do the analysis as we did previously for the 2D Galerkin method and finite difference method.

In many cases, some quantities of interest are more important than the solution of the gyrokinetic equation itself. They can help us better understand the state of the plasma. However, these quantities of interest are related to the solution of the gyrokinetic equation which means we have to compute the high dimensional PDEs first then we can compute the quantities of interest. This sometimes makes the computation cost unaffordable. We will further discuss a concrete example in the numerical results.

3.2.4 Error Splitting Models

For the Galerkin method, we define the following operator P_γ

$$\begin{aligned} P_\gamma :& H \to H \\ & u \mapsto u_\gamma \end{aligned}$$

where we follow the notation in the previous section. u is the exact solution and u_γ is the solution computed from the Galerkin Method. According to the weak formulation, we have

$$a(u - u_\gamma, v_\gamma) = 0, \ \forall v_\gamma \in H_\gamma.$$

Therefore, the operator P_γ is a projection. For the 2D finite difference method, we define the following operator P_{τ,h_1,h_2}

$$\begin{aligned} P_{\tau,h_1,h_2} :& X \to X \\ & u \mapsto u_{\tau,h_1,h_2} \end{aligned}$$

where u_{τ,h_1,h_2} is the interpolant of the data computed from the linear finite difference method. We can check $P_{\tau,h_1,h_2} u_{\tau,h_1,h_2} = u_{\tau,h_1,h_2}$. Thus P_{τ,h_1,h_2} is a projection. Therefore, for both methods we can apply the generalised combination technique directly to its solution.

In order to make sure the convergence of the generalised combination technique, we further require the error splitting models. Now we review some results of the error splitting models for the Galerkin Method and the finite difference method. For the Gyrokinetic equations, it is hard to get an error splitting model for the solution since the underlying numerical scheme is too complex. For the 2D Galerkin method, the error splitting model [17, 74, 73] is

$$u(x,y) - u_\gamma(x,y) = C_1(x,y,h_{\gamma_1})h_{\gamma_1}^p + C_2(x,y,h_{\gamma_2})h_{\gamma_2}^p + C_{1,2}(x,y,h_{\gamma_1},h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^p$$

where h_{γ_1} and h_{γ_2} are the spacings. When the linear basis is used in the approximation, we have $p = 2$. When the quadratic basis is used, we have $p = 3$. The coefficients $C_1(x, h_{\gamma_1})$, $C_2(x, h_{\gamma_2})$ and $C_{12}(x, h_{\gamma_1}, h_{\gamma_2})$ are bounded under suitable assumptions of regularity. The error splitting model [76] for upwind scheme applied to the 2D advection equation is

$$\begin{aligned}u(p)-u_\gamma(p)=&C_1(p,\tau_{\gamma_1})\tau_{\gamma_1}+C_2(p,h_{\gamma_2})h_{\gamma_2}+C_3(p,h_{\gamma_3})h_{\gamma_3}\\&+C_{1,2}(p,\tau_{\gamma_1},h_{\gamma_2})\tau_{\gamma_1}h_{\gamma_2}+C_{1,3}(p,\tau_{\gamma_1},h_{\gamma_3})\tau_{\gamma_1}h_{\gamma_3}\\&+C_{2,3}(p,h_{\gamma_2},h_{\gamma_3})h_{\gamma_2}h_{\gamma_3}\end{aligned}\tag{3.14}$$

where $p=(t,x,y)$ and $\gamma=(\gamma_1,\gamma_2,\gamma_3)$. The error splitting model [76] for forward Euler central difference scheme applied to the 2D diffusion equation is

$$\begin{aligned}u(p)-u_\gamma(p)=&C_1(p,\tau_{\gamma_1})\tau_{\gamma_1}+C_2(p,h_{\gamma_2})h_{\gamma_2}^2+C_3(p,h_{\gamma_3})h_{\gamma_3}^2\\&+C_{1,2}(p,\tau_{\gamma_1},h_{\gamma_2})\tau_{\gamma_1}h_{\gamma_2}^2+C_{1,3}(p,\tau_{\gamma_1},h_{\gamma_3})\tau_{\gamma_1}h_{\gamma_3}^2\\&+C_{2,3}(p,h_{\gamma_2},h_{\gamma_3})h_{\gamma_2}^2h_{\gamma_3}^2\end{aligned}\tag{3.15}$$

where $p=(t,x,y)$ and $\gamma=(\gamma_1,\gamma_2,\gamma_3)$. For both cases, the coefficients are bounded if the solution of the PDEs satisfies the suitable regularity assumption [76].

3.3 Error Splitting Model for Functionals

For a complicated real world problem, multi-stage approximations can be required during the computation of the quantities related to the problem(e.g. solution, norms, functionals of the solution, etc.). In order to obtain the corresponding error splitting model of these quantities, we need to keep track of all those approximations used in the computation. Here, we consider a specific two-stage approximation problem which can be used as a model for computation of some important quantities of a given partial differential equation or system of partial differential equations. The two-stage approximation problem is as follow. Suppose $u\in U\subset X$ is the solution of partial differential equation

$$A(u)=0.$$

We denote the discretisation on a grid G_γ of the operator A as A_γ and the numerical solution of the discretised problem as u_γ. We further assume that we have the following error splitting model of u_γ

$$u(x)-u_\gamma(x)=C_1(x,h_{\gamma_1})h_{\gamma_1}^p+C_2(x,h_{\gamma_2})h_{\gamma_2}^q+C_{1,2}(x,h_{\gamma_1},h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q.\tag{3.16}$$

We compute the following functional

$$\begin{aligned}T:\ U&\to\mathbb{R}\\u&\mapsto T(u).\end{aligned}$$

We denote the discretisation of the functional T on grid G_τ as T_τ and further assume that for any $u \in U$, we have the error splitting model

$$T(u) - T_\tau(u) = V_1(u, h_{\tau_1})h_{\tau_1}^r + V_2(u, h_{\tau_2})h_{\tau_2}^s + W(u, h_{\tau_1}, h_{\tau_2})h_{\tau_1}^r h_{\tau_2}^s. \tag{3.17}$$

Our purpose is to find out the error splitting model for

$$T(u) - T_\tau(u_\gamma).$$

In fact, we have

$$T(u) - T_\tau(u_\gamma) = (T(u) - T(u_\gamma)) + (T(u_\gamma) - T_\tau(u_\gamma)). \tag{3.18}$$

Therefore, we first need to know the error splitting models for $T(u) - T(u_\gamma)$ and $T(u_\gamma) - T_\tau(u_\gamma)$.

We first the consider the first term $T(u) - T(u_\gamma)$. We consider a special case when

$$T = T_0 \circ \cdots \circ T_t.$$

T_i, $i = 0, \ldots, t$ is either an integral operator or a polynomial operator. In order to get the error splitting model for $T(u) - T(u_\gamma)$, we need the following Lemmas

Lemma 3.14. *Suppose $u, u_\gamma \in U \subset \mathbb{R}^{\Omega_1 \times \Omega_2}$, Ω_1 and Ω_2 are bounded. u_γ satisfies the error splitting model in* (3.16). *Then we have*

$$\begin{aligned}\int_{\Omega_1} u(x_1, x_2)\,dx_1 - \int_{\Omega_1} u_\gamma(x_1, x_2)\,dx_1 =& \tilde{C}_1(x_2, h_{\gamma_1})h_{\gamma_1}^p + \tilde{C}_2(x_2, h_{\gamma_2})h_{\gamma_2}^q \\ &+ \tilde{C}_{1,2}(x_2, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q\end{aligned}$$

where

$$\begin{aligned}\tilde{C}_1(x_2, h_{\gamma_1}) &= \int_{\Omega_1} C_1(x, h_{\gamma_1})\,dx_1 \\ \tilde{C}_2(x_2, h_{\gamma_2}) &= \int_{\Omega_1} C_1(x, h_{\gamma_2})\,dx_1 \\ \tilde{C}_{1,2}(x_2, h_{\gamma_1}, h_{\gamma_2}) &= \int_{\Omega_1} C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})\,dx_1.\end{aligned}$$

Furthermore, if for some $K > 0$, $|C_1(x, h_{\gamma_1})| \le K$, $|C_2(x, h_{\gamma_2})| \le K$ and $|C_{1,2}(x\ , h_{\gamma_1}, h_{\gamma_2})| \le K$, then

$$|\tilde{C}_1(x_2, h_{\gamma_1})| \le K,\ |\tilde{C}_2(x_2, h_{\gamma_2})| \le K,\ |\tilde{C}_{1,2}(x_2, h_{\gamma_1}, h_{\gamma_2})| \le K.$$

Proof. Using the linearity of the integration. □

Lemma 3.15. *Suppose M is a polynomial operator of degree t, i.e. for function $u \in U$, we have*

$$M(u) = \sum_{k=0}^{t} a_k u^k$$

where $a_k \in \mathbb{R}$ for $k = 0, \ldots, t$. Suppose $u_\gamma \in U$. Then we have

$$M(u_\gamma) - M(u) = \sum_{j=1}^{t} g_j (u_\gamma - u)^j$$

where

$$g_j = \sum_{k=j}^{t} \binom{k}{j} a_k u^{k-j}.$$

Proof. Using Taylor expansion or direct computation. □

Lemma 3.16.

$$(x+y+z)^k = \sum_{i=0}^{k} \sum_{j=0}^{i} \binom{k}{i} \binom{i}{j} x^j y^{i-j} z^{k-i}$$

Proof. Applying binomial theorem twice. □

Lemma 3.17. *Suppose M is a polynomial operator as defined in Lemma 3.15. Suppose u_γ satisfies the error splitting model in (3.16). Then we have*

$$M(u)(x) - M(u_\gamma)(x) = \tilde{C}_1(x, h_{\gamma_1}) h_{\gamma_1}^p + \tilde{C}_2(x, h_{\gamma_2}) h_{\gamma_2}^q + \tilde{C}_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^p h_{\gamma_2}^q.$$

Furthermore, if for some $K > 0$, $|C_1(x, h_{\gamma_1})| \leq K$, $|C_2(x, h_{\gamma_2})| \leq K$ and $|C_{1,2}(x , h_{\gamma_1}, h_{\gamma_2})| \leq K$, and for some K', $|u| \leq K'$, then there exists some constant $K''(K, K')$, such that

$$|\tilde{C}_1(x, h_{\gamma_1})| \leq K'', \; |\tilde{C}_2(x, h_{\gamma_2})| \leq K'', \; |\tilde{C}_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})| \leq K''.$$

Proof. Sketch of the proof. Combining the result in Lemma 3.15 and the error splitting model, we have

$$M(u_\gamma) - M(u) = \sum_{j=1}^{t} g_j (u_\gamma - u)^j$$

$$= \sum_{j=1}^{t} g_j (C_1(x, h_{\gamma_1}) h_{\gamma_1}^p + C_2(x, h_{\gamma_2}) h_{\gamma_2}^q + C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^p h_{\gamma_2}^q)^j.$$

Then we apply Lemma 3.16 to expand all brackets in the summation. After rearranging the terms, we can get the expressions of $\tilde{C}_1(x, h_{\gamma_1})$, $\tilde{C}_2(x, h_{\gamma_2})$ and $\tilde{C}_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})$. Here we take $\tilde{C}_1(x, h_{\gamma_1})$ for an example. We have

$$\tilde{C}_1(x, h_{\gamma_1}) = \sum_{j=1}^{t} g_j h_{\gamma_1}^{(j-1)p} C_1^j(x, h_{\gamma_1}).$$

When $u(x)$ and $C_1(x, h_{\gamma_1})$ are bound for any $x \in \Omega$, we can prove that $\tilde{C}_1(x, h_{\gamma_1})$ is also bounded for any $x \in \Omega$. □

Combining the results in Lemma 3.14 and Lemma 3.17, we actually proved that when error splitting (3.16) holds, the first term in the right-hand side of (3.18) have the following error splitting

$$T(u) - T(u_\gamma) = \tilde{C}_1(h_{\gamma_1})h_{\gamma_1}^p + \tilde{C}_2(h_{\gamma_2})h_{\gamma_2}^q + \tilde{C}_{1,2}(h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q. \tag{3.19}$$

Here the coefficients do not have x dependency since T is assumed to be a functional. We still use the notation $\tilde{C}$ and $\tilde{C}$ for simplicity though they will change every time when we composite a new operator T_k, $k = 0, \ldots, t$. The coefficients are bounded when the coefficients in (3.16) are bounded.

Now we consider more general choice of the functional T. If $T \in C^k(U, \mathbb{R})$, we have the following Taylor expansion in Banach space X,

$$\begin{aligned} T(u_\gamma) - T(u) =& dT(u)[u - u_\gamma] + \frac{1}{2!}d^2T(u)[u - u_\gamma]^2 + \ldots \\ &+ \ldots \frac{1}{k!}d^kT(u)[u - u_\gamma]^k + \epsilon(u, u_\gamma)[u - u_\gamma]^k \end{aligned}$$

where $d^iT(u) \in L_i(X, \mathbb{R})$, $i = 1, \ldots, k$ is the ith Fréchet derivative. It is an i-linear map from $X \times \cdots \times X$(i times) to $\mathbb{R}$. If we plug in the error splitting model (3.16), the first term in the expansion can be written as

$$\begin{aligned} dT(u)[u - u_\gamma] =& dT(u)[C_1(\cdot, h_{\gamma_1})h_{\gamma_1}^p + C_2(\cdot, h_{\gamma_2})h_{\gamma_2}^q + C_{1,2}(\cdot, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q] \\ =& dT(u)[C_1(\cdot, h_{\gamma_1})]h_{\gamma_1}^p + dT(u)[C_2(\cdot, h_{\gamma_2})]h_{\gamma_2}^q \\ &+ dT(u)[C_{1,2}(\cdot, h_{\gamma_1}, h_{\gamma_2})]h_{\gamma_1}^p h_{\gamma_2}^q \end{aligned}$$

Since $dT(u)$ is a bounded operator, we get an error splitting model for the first term. Similarly, we can compute other terms in the Taylor expansion and add them together. We get an error splitting model for the first k term of the Taylor expansion.

Next we consider the term $T(u_\gamma) - T_\tau(u_\gamma)$ in (3.18). Using the error splitting (3.29), we first have

$$T(u_\gamma) - T_\tau(u_\gamma) = V_1(u_\gamma, h_{\tau_1})h_{\tau_1}^r + V_2(u_\gamma, h_{\tau_2})h_{\tau_2}^s + W_{1,2}(u_\gamma, h_{\tau_1}, h_{\tau_2})h_{\tau_1}^r h_{\tau_2}^s. \quad (3.20)$$

Since $V_1(\cdot, h_{\tau_1})$ and $V_2(\cdot, h_{\tau_2})$ are continuous functionals, we have

$$\begin{aligned}
&V_1(u_\gamma, h_{\tau_1}) - V_1(u, h_{\tau_1})\\
=&E_1(h_{\tau_1}, h_{\gamma_1})h_{\gamma_1}^p + E_2(h_{\tau_1}, h_{\gamma_2})h_{\gamma_2}^q + F_{1,2}(h_{\tau_1}, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q,\\
&V_2(u_\gamma, h_{\tau_2}) - V_2(u, h_{\tau_2})\\
=&G_1(h_{\tau_2}, h_{\gamma_1})h_{\gamma_1}^p + G_2(h_{\tau_2}, h_{\gamma_2})h_{\gamma_2}^q + H_{1,2}(h_{\tau_2}, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q,\\
&W_{1,2}(u_\gamma, h_{\tau_1}, h_{\tau_2}) - W_{1,2}(u, h_{\tau_1}, h_{\tau_2})\\
=&I_1(h_{\tau_1}, h_{\tau_2}, h_{\gamma_1})h_{\gamma_1}^p + I_2(h_{\tau_1}, h_{\tau_2}, h_{\gamma_2})h_{\gamma_2}^q + J_{1,2}(h_{\tau_1}, h_{\tau_2}, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q.
\end{aligned} \quad (3.21)$$

Substitute expansions in (3.21) into expansion (3.20). We obtain

$$\begin{aligned}
&T(u_\gamma) - T_\tau(u_\gamma)\\
=&V_1 h_{\tau_1}^r + E_1 h_{\gamma_1}^p h_{\tau_1}^r + E_2 h_{\gamma_2}^q h_{\tau_1}^r + F_{1,2} h_{\gamma_1}^p h_{\gamma_2}^q h_{\tau_1}^r + V_2 h_{\tau_2}^s + G_1 h_{\gamma_1}^p h_{\tau_2}^s + G_2 h_{\gamma_2}^q h_{\tau_2}^s\\
&+ H_{1,2} h_{\gamma_1}^p h_{\gamma_2}^q h_{\tau_2}^s + W_{1,2} h_{\tau_1}^r h_{\tau_2}^s + I_1 h_{\gamma_1}^p h_{\tau_1}^r h_{\tau_2}^s + I_2 h_{\gamma_2}^q h_{\tau_1}^r h_{\tau_2}^s + J_{1,2} h_{\gamma_1}^p h_{\gamma_2}^q h_{\tau_1}^r h_{\tau_2}^s.
\end{aligned} \quad (3.22)$$

Here we omit the variables in the coefficients for a concise expression. Although (3.22) is quite complicated and has connections with both (3.16) and (3.29), we will usually choose

$$h_{\gamma_1} = h_{\tau_1},$$
$$h_{\gamma_2} = h_{\tau_2}$$

in practice. This is because we can directly use the computed function values in computing the functional value without further interpolation or extrapolation. In order to make sure the final error splitting model achieves the highest possible accuracy, we will choose

$$r \geq p$$
$$s \geq q.$$

We also want to reduce the computational cost. Therefore, a reasonable choice is $r = p$ and $s = q$. In this case, we have

$$T(u_\gamma) - T_\gamma(u_\gamma) = \tilde{V}_1(h_{\gamma_1})h_{\gamma_1}^p + \tilde{V}_2(h_{\gamma_2})h_{\gamma_2}^q + \tilde{W}_{1,2}(h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q. \quad (3.23)$$

Therefore, combining (3.18), (3.19) and (3.23), we finally get

$$T(u) - T_\gamma(u_\gamma) = X_1(h_{\gamma_1})h_{\gamma_1}^p + X_2(h_{\gamma_2})h_{\gamma_2}^q + Y_{1,2}(h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q. \quad (3.24)$$

It is possible that for different functionals, the convergence processes are also different due to the different coefficients in the error splitting models.

3.4 Error Splitting Model and Surpluses Decay

Using the error splitting model, we can further develop a corresponding model describe the decay of surpluses. The following Theorem explains the idea for 2D functions.

Theorem 3.18. *[93] Suppose $u \in U = \mathbb{R}^{[0,1]\times[0,1]}$. Let $u_\gamma \in U$ be an approximation of u on the equally spaced grid $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ with the spacings*

$$h_{\gamma_k} = 2^{-\gamma_k}, \ k = 1, 2.$$

u_γ satisfies a more general error splitting model

$$u(x) - u_\gamma(x) = C_1(x, h_{\gamma_1})h_{\gamma_1}^p + C_2(x, h_{\gamma_2})h_{\gamma_2}^q + C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q$$

for p, $q \in \mathbb{N}$. Then the surpluses

$$\Delta_\gamma(u)(x) = u_{\gamma_1,\gamma_2}(x) - u_{\gamma_1-1,\gamma_2}(x) - u_{\gamma_1,\gamma_2-2}(x) + u_{\gamma_1-1,\gamma_2-1}(x) \tag{3.25}$$

satisfy

$$\Delta_\gamma(u)(x) = \Theta(x, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q$$

for p, $q \in \mathbb{N}$. If the coefficients $|C_1(x, h_{\gamma_1})| \le K$, $|C_2(x, h_{\gamma_2})| \le K$ and $|C_{1,2}(x, h_1, h_2)| \le K$ for some $K > 0$, then

$$|\Theta(x, h_{\gamma_1}, h_{\gamma_2})| \le K(1 + 2^p)(1 + 2^q).$$

Proof. Using the definition of the surplus and the error splitting model, we have

$$\begin{aligned}
&\Delta_\gamma(u)(x) = \Delta_{\gamma_1,\gamma_2}(u)(x)\\
=&(P_{\gamma_1,\gamma_2} - P_{\gamma_1-1,\gamma_2} - P_{\gamma_1,\gamma_2-1} + P_{\gamma_1-1,\gamma_2-1})(u)(x)\\
=&[(P_{\gamma_1,\gamma_2} - I) + (I - P_{\gamma_1-1,\gamma_2}) + (I - P_{\gamma_1,\gamma_2-1}) + (P_{\gamma_1-1,\gamma_2-1} - I)](u)(x)\\
=&\Theta(x, h_{\gamma_1}, h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q
\end{aligned}$$

where

$$\begin{aligned}
&\Theta(x, h_{\gamma_1}, h_{\gamma_2})\\
=&[-C_{1,2}(x, h_{\gamma_1}, h_{\gamma_2}) + C_{1,2}(x, h_{\gamma_1-1}, h_{\gamma_2})2^p\\
&+ C_{1,2}(x, h_{\gamma_1},, h_{\gamma_2-1})2^q - C_{1,2}(x, h_{\gamma_1-1}, h_{\gamma_2-1})2^{p+q}].
\end{aligned}$$

Since the coefficients in the error splitting model are bounded, we have

$$|\Theta(x, h_{\gamma_1}, h_{\gamma_2})| \le K(1 + 2^p)(1 + 2^q)$$

by using the triangle inequality. □

Conversely, if we know the model which describes the decay of surpluses, i.e.

$$\Delta_\gamma(u)(x) = \Theta(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^p h_{\gamma_2}^q,$$

we can also rebuild the error splitting model.

Theorem 3.19. *[93] Suppose $u \in U = \mathbb{R}^{[0,1]\times[0,1]}$. Let $u_\gamma \in U$ be an approximation of u on the equally spaced grid $G_\gamma = G_{\gamma_1} \times G_{\gamma_2}$ with the spacings*

$$h_{\gamma_k} = 2^{-\gamma_k}, \ k = 1, 2.$$

If the surpluses defined in (3.25) *satisfy*

$$\Delta_\gamma(u)(x) = \Theta(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^p h_{\gamma_2}^q, \ \forall \gamma,$$

then we have the following error splitting model for u_α, $\forall \alpha$

$$u(x) - u_\alpha(x) = C_1(x, h_{\alpha_1}) h_{\alpha_1}^p + C_1(x, h_{\alpha_2}) h_{\alpha_2}^q + C_{1,2}(x, h_{\alpha_1}, h_{\alpha_2}) h_{\alpha_1}^p h_{\alpha_2}^q.$$

Furthermore, if for some $K > 0$

$$|\Theta(x, h_{\gamma_1}, h_{\gamma_2})| \leq K, \ \forall \gamma, \tag{3.26}$$

the coefficients in the error splitting model are also bounded, i.e.

$$\begin{aligned}
|C_1(x, h_{\alpha_1})| &\leq \frac{K 2^{-p}}{(1-2^{-q})(1-2^{-p})} \\
|C_2(x, h_{\alpha_2})| &\leq \frac{K 2^{-q}}{(1-2^{-q})(1-2^{-p})} \\
|C_{1,2}(x, h_{\alpha_1}, h_{\alpha_2})| &\leq \frac{K 2^{-p} 2^{-q}}{(1-2^{-q})(1-2^{-p})}.
\end{aligned}$$

Proof. Using the inclusion-exclusion principle, we have

$$\begin{aligned}
u(x) - P_\alpha(u)(x) &= u(x) - P_{\alpha_1,\alpha_2}(u)(x) \\
&= \left(\sum_{\gamma_1 > \alpha_1} \sum_{\gamma_2 = 0}^{\infty} + \sum_{\gamma_1 = 0}^{\infty} \sum_{\gamma_2 > \alpha_2} - \sum_{\gamma_1 > \alpha_1} \sum_{\gamma_2 > \alpha_2} \right) \Delta_{\gamma_1,\gamma_2}(u)(x) \\
&= \left(\sum_{\gamma_1 > \alpha_1} \sum_{\gamma_2 = 0}^{\infty} + \sum_{\gamma_1 = 0}^{\infty} \sum_{\gamma_2 > \alpha_2} - \sum_{\gamma_1 > \alpha_1} \sum_{\gamma_2 > \alpha_2} \right) \Theta(x, h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^p h_{\gamma_2}^q.
\end{aligned} \tag{3.27}$$

If we denote

$$\begin{aligned}\Theta_1(x,h_{\gamma_1}) &= \sum_{\gamma_2=0}^{\infty}\Theta(x,h_{\gamma_1},h_{\gamma_2})h_{\gamma_2}^q\\ C_1(x,h_{\alpha_1}) &= \sum_{\gamma_1>\alpha_1}\Theta_1(x,2^{(\alpha_1-\gamma_1)p}h_{\alpha_1}^p)2^{(\alpha_1-\gamma_1)p}\end{aligned}$$

the first term in the right hand side of (3.27) can be written as

$$\begin{aligned}\sum_{\gamma_1>\alpha_1}\sum_{\gamma_2=0}^{\infty}\Theta(x,h_{\gamma_1},h_{\gamma_2})h_{\gamma_2}^q =& \sum_{\gamma_1>\alpha_1}\Theta_1(x,h_{\gamma_1})h_{\gamma_1}^p\\ =&h_{\alpha_1}^p\sum_{\gamma_1>\alpha_1}\Theta_1(x,2^{(\alpha_1-\gamma_1)p}h_{\alpha_1}^p)2^{(\alpha_1-\gamma_1)p}\\ =&C_1(x,h_{\alpha_1})h_{\alpha_1}^p.\end{aligned}$$

For the second term in the right hand side of (3.27), we have similar result. For the last term. if we denote

$$C_{1,2}(x,h_{\gamma_1},h_{\gamma_2}) = \sum_{\gamma_1>\alpha_1}\sum_{\gamma_2>\alpha_2}\Theta(x,2^{(\alpha_1-\gamma_1)p}h_{\alpha_1},2^{(\alpha_2-\gamma_2)p}h_{\alpha_2})2^{(\alpha_1-\gamma_1)p}2^{(\alpha_2-\gamma_2)p},$$

we have

$$\sum_{\gamma_1>\alpha_1}\sum_{\gamma_2>\alpha_2}\Theta(x,h_{\gamma_1},h_{\gamma_2})h_{\gamma_1}^p h_{\gamma_2}^q = C_{1,2}(x,h_{\alpha_1},h_{\alpha_2})h_{\alpha_1}^p h_{\alpha_2}^q.$$

If (3.26) holds, we have

$$|\Theta_1(x,h_{\gamma_1})| \le K\sum_{\gamma_2=0}^{\infty}h_{\gamma_2}^q = K\sum_{\gamma_2=0}^{\infty}2^{-\gamma_2 q} = \frac{K}{1-2^{-q}}$$

and

$$|C_1(x,h_{\alpha_1})| \le \frac{K}{1-2^{-q}}\sum_{\gamma_1>\alpha_1}2^{(\alpha_1-\gamma_1)p} = \frac{K2^{-p}}{(1-2^{-q})(1-2^{-p})}.$$

Using similar method, we can also compute bound for $C_2(x,h_{\alpha_2})$. We have

$$|C_1(x,h_{\alpha_1})| \le \frac{K2^{-q}}{(1-2^{-q})(1-2^{-p})}$$

For $C_{1,2}(x,h_{\alpha_1},h_{\alpha_2})$, we have

$$|C_{1,2}(x,h_{\alpha_1},h_{\alpha_2})| \le K\sum_{\gamma_1>\alpha_1}2^{(\alpha_1-\gamma_1)p}\sum_{\gamma_2>\alpha_2}2^{(\alpha_2-\gamma_2)p} = \frac{K2^{-p}2^{-q}}{(1-2^{-q})(1-2^{-p})}.$$

□

Theorem 3.18 and Theorem 3.19 give the connection between the error splitting model and the decay of the surpluses. If we know the error splitting model, the result in Theorem 3.18 can be used to check the numerical result obtained from computation. On the other hand, if we solve a complicated problem and implement multiple times of approximations during our computation, it will be hard for us to get a concrete error splitting model. In this case, we can first compute the surpluses use the numerical scheme and then use Theorem 3.19 to study the corresponding error splitting model. Also, due to the connection between the error splitting model and the decay of the surpluses, one can check the convergence of the numerical scheme and design a more sophisticated combination technique by directly studying the computed surpluses instead of the unknown error splitting model.

The result in Theorem 3.18 and Theorem 3.19 can be easily generalised to functionals. Suppose we have an error splitting model (3.24) with bounded coefficients. By using similar idea in the proofs of the Theorems, we can get its error decay model is

$$\Delta_\gamma(T(u)) = \Theta(h_{\gamma_1}, h_{\gamma_2}) h_{\gamma_1}^p h_{\gamma_2}^q$$

where

$$\begin{aligned}\Delta_\gamma(T(u)) =& T_{\gamma_1,\gamma_2}(u_{\gamma_1,\gamma_2}) - T_{\gamma_1-1,\gamma_2}(u_{\gamma_1-1,\gamma_2}) \\ &- T_{\gamma_1,\gamma_2-1}(u_{\gamma_1,\gamma_2-1}) + T_{\gamma_1-1,\gamma_2-1}(u_{\gamma_1-1,\gamma_2-1})\end{aligned}$$

and the coefficient is bounded.

3.5 Numerical Results

In the following, we show numerical experiments of the examples in the section 3.2.

3.5.1 Poisson Problem

The 2D Poisson equation

$$\begin{aligned}-\Delta u &= f \quad in\ [0,1]^2, \\ u &= 0 \quad on\ \partial[0,1]^2.\end{aligned}$$

In particular, we take $f = 1$ here. We consider the following functional

$$T(u) = \int_\Omega u(x,y)\, dxdy. \tag{3.28}$$

We use the Galerkin method to solve the Poisson equation. Linear nodal basis functions and quadratic nodal basis functions are used in the computation. Also, we use two different quadrature rules, trapezoidal rule and Simpson's rule to compute the integral in (3.28).

The error splitting model for the Galerkin method

$$u(x) - u_\gamma(x) = C_1(x, h_{\gamma_1})h^p_{\gamma_1} + C_2(x, h_{\gamma_2})h^p_{\gamma_2} + C(x, h_{\gamma_1}, h_{\gamma_2})h^p_{\gamma_1}h^p_{\gamma_2}$$

as we discussed in Section 3.2.4. For the tensor product trapezoidal rule and Simpson's rule, we have the following error splitting model

$$T(u) - T_\tau(u) = V_1(u, h_{\tau_1})h^r_{\tau_1} + V_2(u, h_{\tau_2})h^r_{\tau_2} + W(u, h_{\tau_1}, h_{\tau_2})h^r_{\tau_1}h^r_{\tau_2} \qquad (3.29)$$

where h_{τ_1}, $\tau_1 = 0, 1, 2, \ldots$ are the length of the spacing in x and h_{τ_2}, $\tau_2 = 0, 1, 2, \ldots$ are the length of the spacing in y. The coefficients $V_1(u_1, h_{\tau_1})$, $V_2(u, h_{\tau_2})$ and $W_{12}(u, \tau_1, \tau_2)$ are bounded if u is smooth enough. $r = 2$ if T is the trapezoidal rule and $r = 3$ if T is the Simpson's rule.

If we further assume $h_{\gamma_1} = h_{\tau_1}$ and $h_{\gamma_2} = h_{\tau_2}$, according to the combined error splitting model (3.22), we have the following error splitting model for the case of quadratic basis and Simpson's rule

$$T(u) - T_\gamma(u_\gamma) = X_1(h_{\gamma_1})h^3_{\gamma_1} + X_2(h_{\gamma_2})h^3_{\gamma_2} + Y_{1,2}(h_{\gamma_1}, h_{\gamma_2})h^3_{\gamma_1}h^3_{\gamma_2}.$$

For other cases(linear basis and trapezoidal rule, linear basis and Simpson's rule and quadratic basis and trapezoidal rule), we have

$$T(u) - T_\gamma(u_\gamma) = X_1(h_{\gamma_1})h^2_{\gamma_1} + X_2(h_{\gamma_2})h^2_{\gamma_2} + Y_{1,2}(h_{\gamma_1}, h_{\gamma_2})h^2_{\gamma_1}h^2_{\gamma_2}.$$

The surpluses of the functional are $\Delta_\gamma(T(u))$. According to the Theorem 3.19, we have

$$\Delta_\gamma(T(u)) = O(h^3_{\gamma_1}h^3_{\gamma_2}) \text{ and } \Delta_\gamma(T(u)) = O(h^2_{\gamma_1}h^2_{\gamma_2})$$

respectively. In the Figure 3.1, we compute the exponents of the corresponding surpluses for four different cases. In all four plots, the absolute values of the surpluses decay when we increase the grid points in either x direction or y direction. We can see the convergence of all four methods from these decays. As we expected, the most accurate numerical scheme, using quadratic basis and Simpson's rule, achieves the fastest convergence. If we compare these plots in more details, we can actually find these absolute value of surpluses decrease with different patterns, especially when we compare the last plot with the other three. These differences lead to different 'optimal choice' of the downsets of the generalised combination technique.

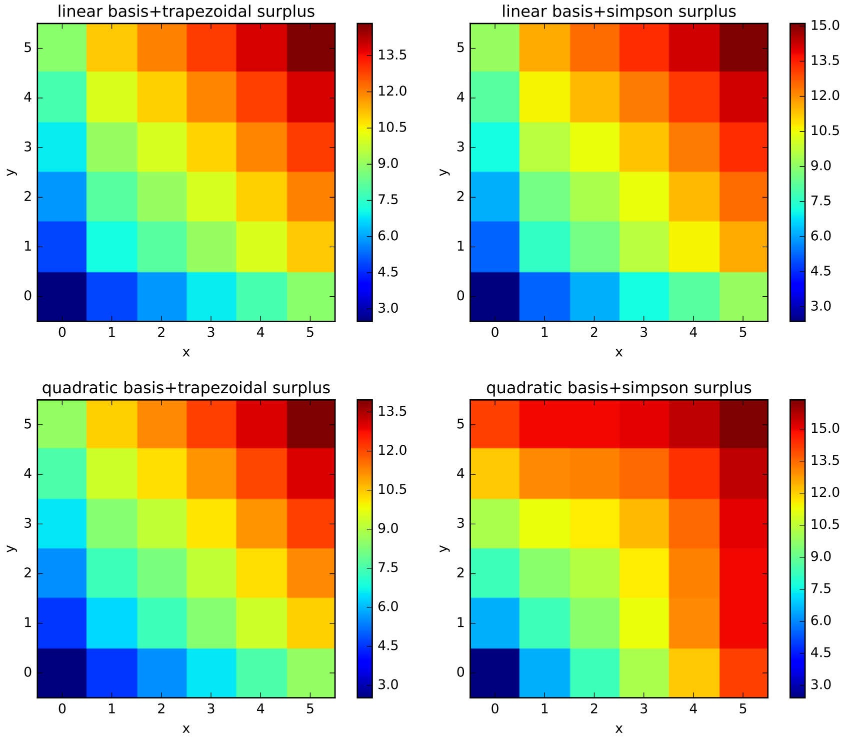


Figure 3.1: These figures show the absolute values of exponents of the corresponding surpluses, i.e. $|\log(\Delta_\gamma(T(u))/\log 4|$. The corsest grid surplus is at the left bottom corner. The grid size is 5 × 5. The finest grid surplus is at the top right corner. The grid size is 129 × 129. By comparing the marks on the colorbar, we can know the exponents of each surpluses.

3.5.2 Advection Equation and Diffusion Equation

Consider the 2D advection equation and 2D diffusion equation

$$\frac{\partial u}{\partial t} + a_1 \frac{\partial u}{\partial x} + a_2 \frac{\partial u}{\partial y} = 0,$$

and

$$\frac{\partial u}{\partial t} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right).$$

In the computation, the initial condition for the advection equation is

$$u_0(x, y) = (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2$$

and we use the periodic boundary condition on $[0, 1]^2$. Without loss of generality, we take $a_1 = 1$ and $a_2 = 2$ here. The initial condition for the diffusion equation is

$$u_0(x, y) = \frac{1}{0.04\pi} \exp \left\{ \frac{1}{0.04} [(x - \frac{1}{2})^2 + (y - \frac{1}{2})^2] \right\}.$$

We use zero boundary condition on $[0, 1]^2$ and set $\nu = 0.01$. We want to show for more complicated functionals T, we can also apply combination technique directly to the computed functionals. Here we tested the following functionals

$$\begin{aligned}
T^1(u) &= \int_{[0,1]^2} u^2(x, y, t)\, dxdy, \\
T^2(u) &= \left(\int_{[0,1]^2} u(x, y, t)\, dxdy \right)^2, \\
T^3(u) &= \int_{[0,1]^2} \cos(u(x, y, t))\, dxdy, \\
T^4(u) &= \cos(\int_{[0,1]^2} u(x, y, t)\, dxdy), \\
T^5(u) &= \int_{[0,1]^2} \exp(u(x, y, t))\, dxdy, \\
T^6(u) &= \exp(\int_{[0,1]^2} u(x, y, t)\, dxdy)
\end{aligned} \tag{3.30}$$

for fixed time t for both equations. If we use the same notation as in the (3.14) and (3.15) where τ_{γ_1} is the length of time step and h_{γ_2}, h_{γ_3} is the length of the spacings in x, y respectively, the error is

$$T(u) - T_{\gamma_2,\gamma_3}(u_{\gamma_1,\gamma_2,\gamma_3}) = T(u) - T(u_{\gamma_1}) + T(u_{\gamma_1}) - T_{\gamma_2,\gamma_3}(u_{\gamma_1,\gamma_2,\gamma_3}).$$

We consider the case when τ_{γ_1} is fixed and apply the combination technique in the space (x, y). Therefore, we need an error splitting model for

$$T(u_{\gamma_1}) - T_{\gamma_2,\gamma_3}(u_{\gamma_1,\gamma_2,\gamma_3}). \tag{3.31}$$

In fact, the first two functionals in (3.30) fit in the setting when T can be written as compositions of integral operators and polynomial operators. Therefore, according to our theory, we can exactly derive an error splitting model for (3.31) from the error splitting model of the solutions of the PDEs(with τ_{γ_1} fixed) and the numerical methods used to compute the integrals. For the other four functionals, we can use the Taylor expansion and obtain an error splitting model for the first k terms of it. Since these functionals are smooth, we can take a sufficiently large k. Thus, the error splitting model for the first k terms will be close to the true error splitting model.

We can also study the computed surpluses of these functionals. Here we use the trapezoidal rule to compute the integrals. In the Figure 3.2, we show the result for the advection equation. In all six plots, the absolute values of the surpluses decay when we increase the grid points in either x or y direction. This means the surpluses of each functional satisfy a decay model as in the Theorem 3.18 and 3.19. Thus, we also have an error splitting model for the (3.31). Therefore, the (generalised) combination techniques for the functionals are convergent. If we further look into the details of all six plots, we will find they show similar decay pattern. If we truncate the row 0 and the column 0, the absolute values of surpluses along all diagonals from top-left to bottom-right are almost the same(almost same color in the plots). This suggests a truncated classical combination technique is a good choice used to compute all six functionals. In the Figure 3.3, we show the result for the diffusion equation. We also have error decay models for all six functionals which suggests the combination techniques are convergent. However, the decay patterns are different for each case. Therefore, in order to achieve fast convergence, different generalised combination techniques need to be used.

3.5.3 Quantity of Interest from GENE Experiment

We now consider computing the quantities of interest of the Gyrokinetic equation. Here we focus on those spatially averaged normalized fluctuating quantities which can be written into high dimensional integrals. In particular, we take following

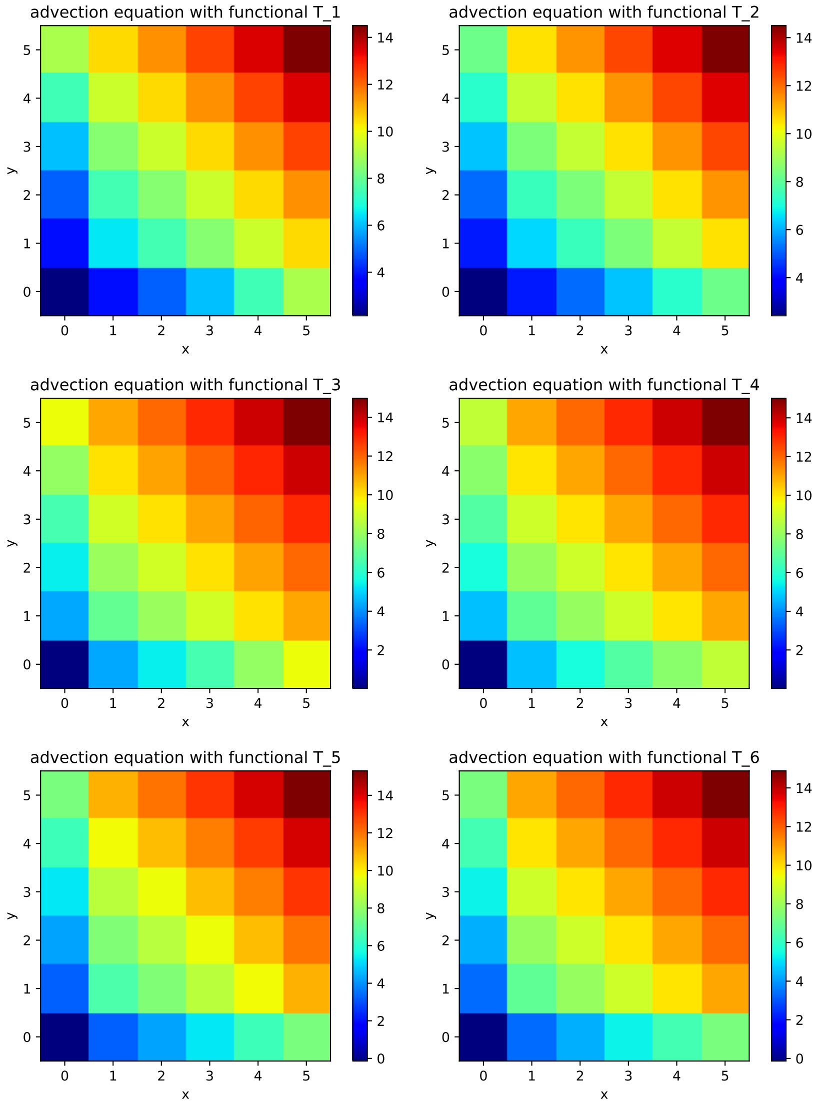


Figure 3.2: Here is the result for the advection equation. These figures show the absolute values of exponents of the corresponding surpluses on the anisotropic grids. The surpluses are in phase space (x, y) when we take $T = 1$ in the functionals. The coarsest grid surplus is at the left bottom corner. The grid size is 5×5. The finest grid surplus is at the top right corner. The grid size is 129×129.

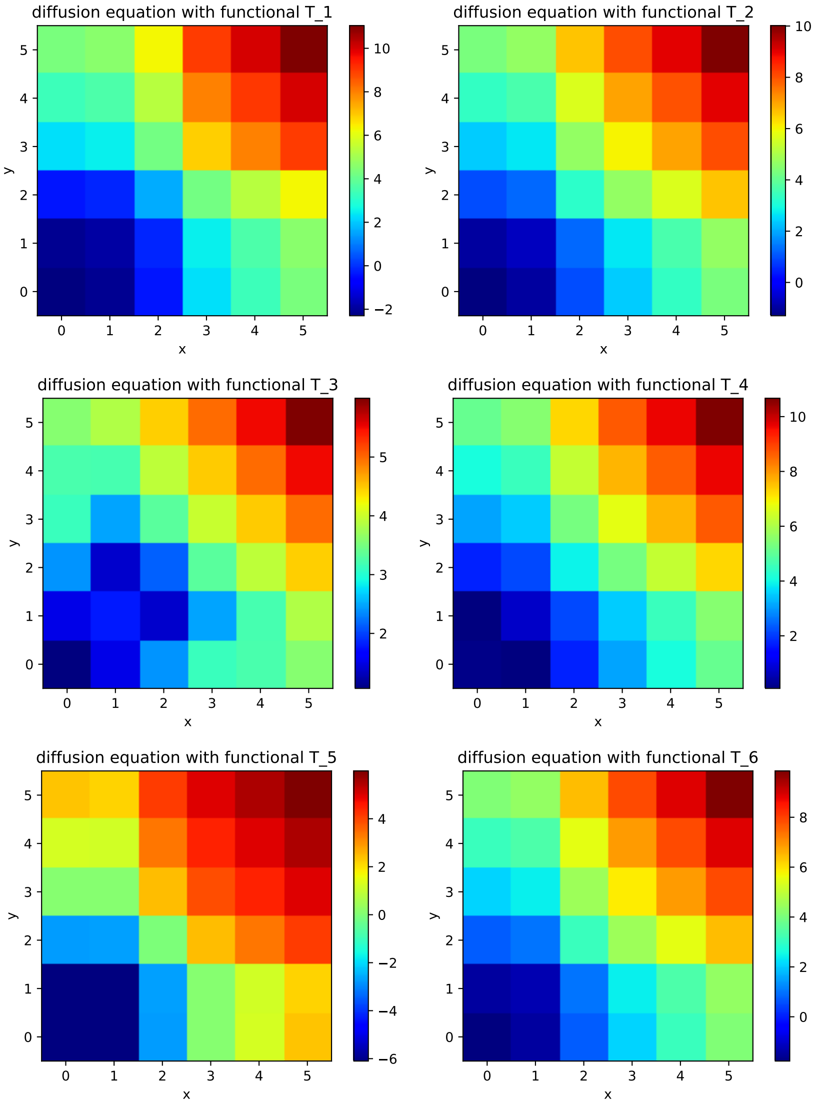


Figure 3.3: Here is the result for the diffusion equation. These figures show the absolute values of exponents of the corresponding surpluses on the anisotropic grids. The surpluses are in phase space (x, y) when we take $T = 3$ in the functionals. The coarsest grid surplus is at the left bottom corner. The grid size is 3×3. The finest grid surplus is at the top right corner. The grid size is 65×65.

spatially averaged, normalized fluctuating quantity as an example

$$\frac{\langle |n_1|^2 \rangle}{(n_0 \rho^*_{ref})^2}, \tag{3.32}$$

other physical quantities can be similarly computed by our method. In (3.32), n_0 is the equilibrium density of the corresponding species and ρ^*_{ref} is the reference gyroradius-to-machine-size ratio[‡],

$$n_1 = \int_V f_1^{(pc)}(x, v, t)\, dv$$

is the velocity space average/moments of the fluctuating part of a time-dependent particle distribution function $f^{(pc)}$. Here $f^{(pc)}$ can be obtained by solving the nonlinear gyrokinetic equations [13]. The phase space of the distribution function $f^{(pc)}$ is 6D of which 3D in real space and 3D in velocity space. $\langle\cdot\rangle$ is the spatial average over real space X. In fact, the 6D phase space is reduced into 5D in GENE simulation since the fast gyromotion can be removed from the nonlinear gyrokinetic equations. Thus, if we ignore the constant n_0, ρ^*_{ref} and keep using the notation $f_1^{(pc)}$ and $f^{(pc)}$ after fast gyromotion is removed. (3.32) can be written explicitly as

$$T(f_1^{(pc)}(:, :, t)) = \int_X \left(\int_V f_1^{(pc)}(x, v, t)\, dv \right)^2 dx.$$

It is a time-dependent quantity. x describes the position of the gyrocenter in 3D real space and v is a 2D space of which two coordinates are the parallel velocity and the magnetic moment.

Similar as the Poisson problem, GENE problem also requires two stage approximations. GENE first computes $f_{1,\gamma}^{(pc)}$ as an approximation of $f_1^{(pc)}$. Then it uses a quadrature rule T_τ to compute the quantity of interest. These two approximations are also treated on the same grid for one simulation, therefore $\gamma = \tau$. However, unlike the Poisson problem, it is not easy to obtain an error splitting model for

$$T(f_1^{(pc)}(:, :, t)) - T_\gamma(f_{1,\gamma}^{(pc)}(:, :, t))$$

because the whole computation process in GENE is too complex for us to obtain an accurate analysis. Since GENE can provide us with the solution of the nonlinear gyrokinetic equations and the value of quantities of interest on anisotropic full grid, the surpluses are easy to obtain from these data. Therefore, using the

[‡] Details can be found on [88] page 45.

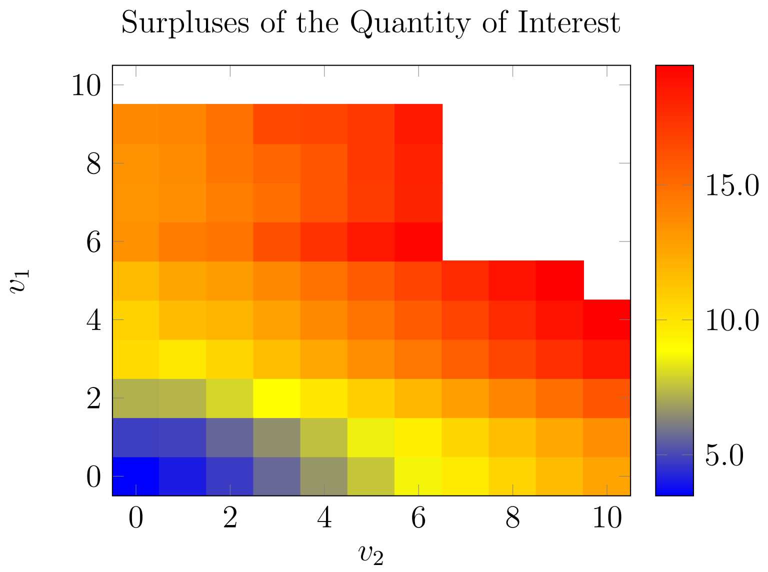


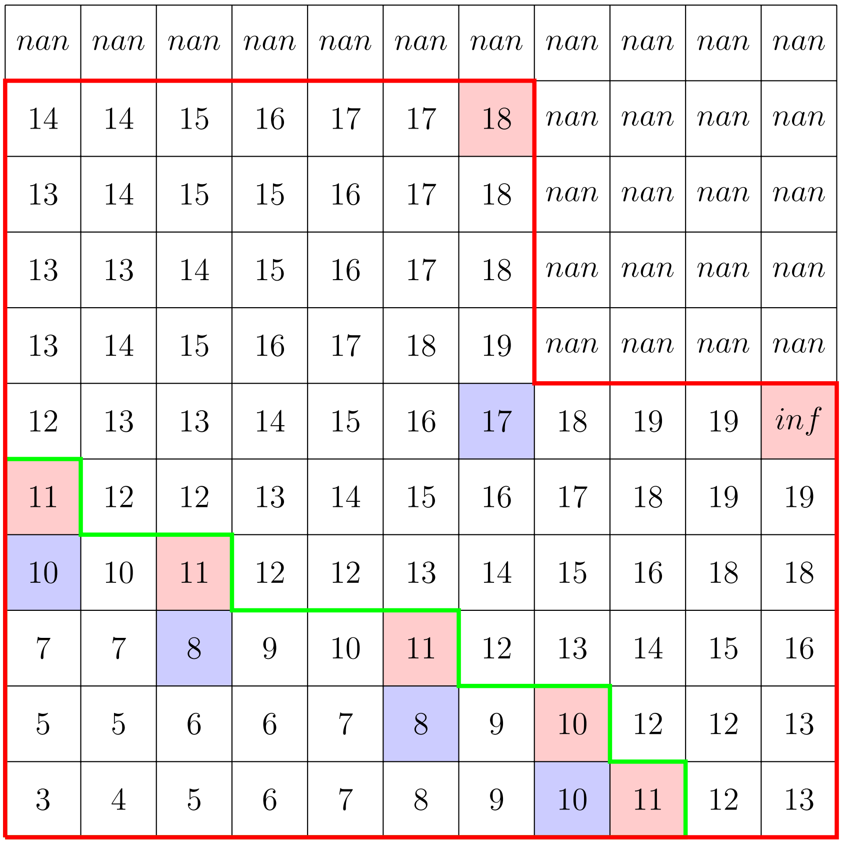
Figure 3.4: We compute the absolute values of exponents of all the surpluses for our quantity of interest. Compare the color to the color bar, the absolute value of a surplus is around 4^{-c} where c is its value on the color bar. The data of missing blocks is expensive to compute and not available. The left bottom block shows the data computed from a grid with size 9×9.

equivalence of the error splitting model and surpluses decay model shown in Theorem 3.18 and 3.19, we switch to use the surpluses decay model for this problem. In order to visualise the surpluses, we only consider the combination in the velocity space, which physicists are most interested in. Let $\eta = (\eta_1, \eta_2)$ be the new multi-index. The generalised combination technique on velocity space is

$$S_I(f_1^{(pc)}(:,:,t)) = \sum_{\eta \in I} c_\eta S_\eta(f_1^{(pc)}(:,:,t)) = \sum_{\eta \in I} c_\eta T_\eta(f_{1,\eta}^{(pc)}(:,:,t)).$$

By studying the surpluses decay, we are able to design combination technique with high accuracy.

The surpluses of the quantity of interest are shown in the Figure 3.4. As the surpluses of the Poisson problem, the absolute value of the surpluses (almost) decay while we increase the grid in either v_1 and v_2 direction. Unlike the previous problems, the absolute value of surpluses decay much faster in one direction(v_1) than the other(v_2). In Figure 3.5, we show two generalised combination techniques according to the observation in the previous surpluses plot. In the first (generalised) combination technique, we combine all the blocks when the absolute value of the hierarchical surpluses are greater than 4^{-11}. The downset I of the combination technique is not symmetric. In the second (generalised) combination technique, we combine these three blocks on the top right corner which is the best



nan	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*	*nan*
14	14	15	16	17	17	18	*nan*	*nan*	*nan*	*nan*
13	14	15	15	16	17	18	*nan*	*nan*	*nan*	*nan*
13	13	14	15	16	17	18	*nan*	*nan*	*nan*	*nan*
13	14	15	16	17	18	19	*nan*	*nan*	*nan*	*nan*
12	13	13	14	15	16	17	18	19	19	*inf*
11	12	12	13	14	15	16	17	18	19	19
10	10	11	12	12	13	14	15	16	18	18
7	7	8	9	10	11	12	13	14	15	16
5	5	6	6	7	8	9	10	12	12	13
3	4	5	6	7	8	9	10	11	12	13

Figure 3.5: We show two (generalised) combination techniques for computing the quantity of interest according to the previous surpluses figure. The green and red borders show downsets of two different (generalised) combination techniques. Number in each block is the absolute value of exponent of the surplus. inf means the surplus on the block is very small. The block with number nan is expensive to compute and not available.

combination for the given data.

3.6 Conclusions

In this Chapter, we actually covered two topics. The first one is how to obtain an error splitting model for the two-stage approximation problem. It is common to check if there is an error splitting model for the problem before we implement the (generalised) combination technique. This is because the error splitting model is the basis for the convergence of the (generalised) combination technique as shown in the Chapter 2. However, for many real world problems, this is impossible or very difficult. First, a real world problem can be a multi-stage approximation problem which is more difficult to analyse than a standard textbook problem. Second, one may use many different legacy codes to solve a complex problem without knowing any details about the algorithms of the legacy codes. Therefore, the (generalised) combination technique which can be used without knowing much information about the numerical scheme is required. Our second topic is studying the basis for such combination technique. By studying the connections between the error splitting model and the surpluses decay model, we can directly implement the (generalised) combination technique when we only know the computed surpluses of the functionals. This is convenient and can be widely used in many applications.

Chapter 4

The Application of Sparse Grid Method in Stochastic Optimisation

Stochastic optimisation is a useful tool in decision making and has many applications [10, 80, 11]. It minimises the expectation of a random cost function. In this Chapter, we will mostly focus on unconstrained stochastic optimisation problems since techniques for constrained problem are often extensions of unconstrained methods(though we include several simple constrained stochastic optimisation examples in our numerical experiments).

Stochastic optimisation can be solved by either randomised algorithms or deterministic algorithms. Randomised algorithms are aimed to solve problems with both high dimensions in optimisation space and probability space. Randomised algorithms have been widely used in many real world problems. The first and one of the most important randomised optimisation algorithm is the stochastic gradient descent(SGD) method [11, 50, 66]. The deterministic gradient vectors in ordinary gradient decent method are replaced by stochastic gradient vectors. Although SGD is a simple algorithm, it is the standard choice for most optimisation problems in data science even today. Another frequently used randomised algorithm is the adaptive gradient method(AdaGrad) [11, 26]. AdaGrad is an improved version of SGD by using some curvature information while retaining computational efficiency. Higher-order methods, such as Newton or quasi-Newton methods are not commonly used in randomised algorithms because of some practical difficulties and lack of convergence theory. However, people are getting more and more interested in this topic because their potential fast convergence.

In a deterministic algorithm, the objective function is treated as a high dimensional integral and quadrature rules are used to discretised the integrals which appear in the algorithm. Then the stochastic optimisation problem is solved by using common optimisation algorithm for deterministic optimisation problem. There are two advantages of such deterministic algorithms. First, the problem can be solved with high accuracy if suitable quadrature rules and optimisation algorithms are used. Second, theory of the convergence of such deterministic algorithm can be build upon the convergence theory of the quadrature methods and optimisations methods used.

In this Chapter, we will only focus on the deterministic algorithms for stochastic optimisation problem. There are two categories of detereministic approaches to solve the stochastic optimisation problems. One is based on the idea of 'discretise then optimise'(DTOM) while the other is based on 'optimise then discretise'(OTDM). The methods in the first category are also known as the surrogate methods or scenario generation methods. The main idea of these methods is to first approximate the expectation in the cost function, which is an integral, using some quadrature rules, then minimise the discretised cost function. Thus accurate quadrature rules are required in order to get accurate minimiser and minimum of the optimisation problem. Monte Carlo [77, 61, 51] and quasi Monte Carlo methods [64, 65, 81, 30, 82, 84] has been commonly used in the 'discretise then optimise' method. Promising methods based on classical sparse grids [20] were shown to display high quadrature accuracy for smooth integrands. But they have negative quadrature weights which potentially destroy the convexity of the objective and thus may lead to totally wrong results. We prove here that, due to their high accuracy, sparse grids maintain the convexity of the objective for sufficiently fine grids. In order to further increase the accuracy, we also explore applying the dimension adaptive approach in solving high dimensional stochastic optimisation problems. The main idea of the 'optimise then discretise' method is we first solve the optimisation problem by some algorithms, which can be either gradient based or Hessian based, then we discretised the objective, gradient and Hessian which appear in the algorithm. For methods in the second category, different numerical schemes are allowed to evaluate the objective and each component of the gradient and the Hessian. The 'discretise then optimise' method is a special case of the the 'optimise then discretise' method when we fix to use the same numerical scheme when computing the objective, gradient and Hessian. The application of the dimension adaptive approaches is also possible in the 'optimise then discretise' methods, which help us further reduce the computational cost. Applications are provided to demonstrate the superiority of our approaches over the classical Monte Carlo and product rule based approaches.

4.1 Stochastic Optimisation

The general form of a stochastic optimisation problem is

$$\min_{u \in U} \mathbb{E}[h(u, W)], \tag{4.1}$$

where W is a d dimensional random vector which is defined on the probability space $(\Omega, \mathcal{B}, \mathbb{P})$, $\mathcal{B}$ is the Borel σ-algebra and $\mathbb{P}$ is the associate probability measure. U is a subset of $\mathbb{R}^n$ which contains all possible decisions[*].

If the random vector W subjects to a probability density[†] $p(w)$ on $\mathbb{R}^d$ the objective is of the form

$$F(u) := \mathbb{E}\left[h(u, W)\right] = \int_{\mathbb{R}^d} h(u, w) p(w)\, dw = \int_{\mathbb{R}^d} f(u, w)\, dw \tag{4.2}$$

where $f(u, w) = h(u, w)p(w)$.

First we begin with introducing some basic concepts of optimisation.

Definition 4.1. u^* is a global minimizer of F if $F(u) \geq F(u^*)$ for all $u \in \mathbb{R}^n$.

Definition 4.2. F is convex when F satisfies

$$F(tu + (1-t)v) \leq tF(u) + (1-t)F(v), \quad \forall u, v \in \mathbb{R}^n,\ t \in [0, 1].$$

F is γ-strongly convex when there exists $\gamma > 0$ such that

$$F(tu+(1-t)v) \leq tF(u)+(1-t)F(v) - \frac{1}{2}\gamma t(1-t)\|u-v\|_2^2, \quad \forall u, v \in \mathbb{R}^n,\ t \in [0, 1].$$

We then make some smoothness assumptions on the cost function F such that the global minimiser of the stochastic optimisation problem exists.

[*] The u here is a vector. It is different from the function u we used in previous Chapter. In most standard textbooks, the notation u denotes the solution of a PDE. Also, in Stochastic Optimisation and Stochastic Optimal Control, people use u as a standard notation for the solution or the control.

[†] $p(\omega)$ is a density function respect to the probability measure. (4.2) gives a general method to solve the stochastic optimisation problem. Detailed methods used to calculate the integral are discussed in examples on page 139 and page 159 in later sections. For different density functions, we use different quadrature methods.

Definition 4.3. A function $g : \mathbb{R}^p \to \mathbb{R}^q$ is Lipschitz continuous with constant $L > 0$ if

$$\|g(x) - g(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in \mathbb{R}^q.$$

Assumption 4.3.1. *The function F is continuously differentiable and ∇F is Lipschitz continuous with constant $L > 0$. In this case, we call F L-smooth.*

By using Taylor expansion, we have $\|\nabla^2 F(u)\|_2 \leq L$ if F is L-smooth and twice continuously differentiable.

The following Lemma gives connection between convexity of a function and its smoothness.

Lemma 4.4 ([66])**.** *If F is continuously differentiable, then F is convex if and only if F lies on or above any tangent line:*

$$F(v) \geq F(u) + \nabla F(u)^T (v - u), \quad \forall u, v \in \mathbb{R}^n.$$

Also, F is γ-strongly convex if and only if

$$F(v) \geq F(u) + \nabla F(u)^T (v - u) + \frac{\gamma}{2}\|v - u\|_2^2, \quad \forall u, v \in \mathbb{R}^n.$$

If F is twice continuously differentiable, then F is convex if and only if $\nabla^2 F(w)$ is positive semidefinite for every $w \in \mathbb{R}^n$. Also, F is γ-strongly convex if and only if $\nabla^2 F(w) \geq \gamma I$.

By using this Lemma, we can show the existence and uniqueness of global minimizers for strongly convex functions.

Theorem 4.5 ([66])**.** *If $F : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and strongly convex, then it has a unique global minimizer.*

If the function f satisfies assumptions in Theorem 4.5, we can make sure the stochastic optimisation problem is well defined. Next, we consider the numerical solvers to solve the problem. In order not to be too general, we will use Newton-type methods as our solvers. According to the optimality condition, solving the stochastic optimisation problem is equivalent to solve the following system of equations

$$G(u) = \nabla_u F(u) = 0. \tag{4.3}$$

The Newton-type methods generate following sequence $\{u_p\}$

$$u_{p+1} = u_p - A_p^{-1} G(u_p), \quad p = 0, 1, 2, \ldots \tag{4.4}$$

and one expects the limit of this sequence will be the solution of (4.3). In the iteration, $A_p \in \mathcal{L}(U)$ is an approximation to the derivative $G^{'}(u)$, namely, the Hessian of F. A_p can be generated in many different ways and different choices of A_p lead to different kinds of Newton-type methods. For example, if we take

$$A_p = \nabla_u^2 F(u_p),$$

this is exactly Newton's method. If we take

$$A_p = \alpha_p^{-1} B_p,$$

where α_p is chosen by exact/inexact line search and B_p is updated from the previously computed value B_{p-1}

$$B_p = B_{p-1} + \frac{y_{p-1} y_{p-1}^T}{s_{p-1}^T y_{p-1}} - \frac{(B_{p-1} s_{p-1})(B_{p-1} s_{p-1})^T}{s_{p-1}^T B_{p-1} s_{p-1}},$$

where $B_0 = I$, $s_p := u_{p+1} - u_p$ and $y_p := \nabla F(u_{p+1}) - \nabla F(u_p)$, then the iteration (4.4) becomes the BFGS method [52, 53], one of the most frequently used Quasi Newton methods.

If we further assume $\nabla_u^2 F(u)$ is definite, bounded and Lipschitz continuous, see details in [52, 53], the Newton method is quadratically convergent when the initial value is close enough to the minimiser while the BFGS method is superlinearly convergent. We can similarly assume the positive definiteness and boundedness of $\nabla_u^2 f(u, w)$, $\forall w \in \Omega$ and the Lipschitz continuity of $\nabla_u^2 f(u, w)$, $\forall w \in \Omega$ to make sure Newton method and BFGS method are convergent.

When we consider high dimensional problems, the difficulty lies in the approximations of $F(u)$, $\nabla_u F(u)$ and $\nabla_u^2 F(u)$.The objective $F(u)$ and each component of $\nabla_u F(u)$ and $\nabla_u^2 F(u)$ are high dimensional integrals in such case. Moreover, we have to compute them at each iteration in the solvers. This will result in the curse of dimensionality. Thus, we need to find efficient way to compute these integrals.

4.2 Surrogate Method and Convexity

Suppose the objective $F(u) = \mathbb{E}[h(u, W)]$ satisfies assumptions in Theorem 4.5. Here we approximate this objective using sparse grid quadrature. Unlike Monte Carlo (MC) or Quasi Monte Carlo (QMC) methods, the quadrature weights of sparse grids can be negative. As a consequence, the surrogate $F_l(u)$ may no longer be convex. The surrogate is defined as

$$F_l(u) = \sum_{i=1}^{N} c_i f(u, w_i)$$

where c_i and w_i are the level l sparse grid weights and quadrature points, respectively. We choose sparse grids as they have superior approximation properties for smooth integrands compared to MC and QMC. Using the linearity of the derivatives, the gradient and Hessian used in the computation is

$$\nabla F_l(u) = \sum_{i=1}^{N} c_i \nabla_u f(u, w_i),$$

and

$$\nabla^2 F_l(u) = \sum_{i=1}^{N} c_i \nabla_u^2 f(u, w_i).$$

Therefore, the weights and the grid points are the same when we compute the objective and each component of the gradient and Hessian.

In order to address the problem on nonconvexity of the sparse grid surrogate (for a convex objective) , we define the following norm in $C^2(U)$:

$$\|F\|_N = \sup_{u \in U} \left[|F(u)|^2 + \nabla F(u)^T \nabla F(u) + \text{trace } (\nabla^2 F(u)^T \nabla^2 F(u)) \right]^{1/2}$$

for $F \in C^2(U)$. We will show that for sufficiently large l, $F_l(u)$ is convex under certain conditions.

Theorem 4.6 ([92]). *Let $U \subset \mathbb{R}^n$ be convex and compact and $\Omega = [-1, 1]^n$, and let*

(i) $F_l \in C^2(U)$ and $F \in C^2(U)$,

(ii) $F_l \to F$ in the $C^2(U)$ norm for $l \to \infty$ and

(iii) $\nabla^2 F(u) > \gamma I$ for all $u \in U$ for some $\gamma > 0$ independent of u.

Then there exists $l_0 > 0$ such that $F_l(u)$ is strongly convex for $u \in U$ and all $l \geq l_0$.

Proof. According to (iii), we have

$$\nabla^2 F_l(u) > \gamma I + \nabla^2 F_l(u) - \nabla^2 F(u). \tag{4.5}$$

We use the Frobenius norm as our matrix norm here, then we get

$$\nabla^2 F_l(u) - \nabla^2 F(u) \leq \|\nabla^2 F_l(u) - \nabla^2 F(u)\|_F I. \tag{4.6}$$

Using the definition of the Frobenius norm and the norm we defined, we have

$$\begin{aligned} &\sup_{u \in U} \|\nabla^2 F_l(u) - \nabla^2 F(u)\|_F \\ =&\sup_{u \in U} \left[\text{trace } (\nabla^2 F_l(u) - \nabla^2 F(u))^T (\nabla^2 F_l(u) - \nabla^2 F(u))\right]^{1/2} \\ \leq&\sup_{u \in U} \ \left[|F(u) - F_l(u)|^2 + \nabla(F(u) - F_l(u))^T (\nabla F(u) - F_l(u)) \right. \\ &\left. + \text{trace } (\nabla^2 (F(u) - F_l(u))^T \nabla^2 (F(u) - F_l(u))\right]^{1/2} \\ =&\|F - F_l\|_N. \end{aligned} \tag{4.7}$$

Combining the result in (4.5), (4.6), (4.7) and applying the reverse triangle inequality, we have

$$\nabla^2 F_l(u) \geq (\gamma - \|F - F_l\|_N)\, I.$$

Then the result follows directly from (ii). □

In Figure 4.1 we compute a simple stochastic optimisation problem with cost function

$$h(u, W) = u^2 + (W_0^2 + 10W_1^2)\, u, \tag{4.8}$$

where $u \in U = [-5, 5]$ and $W_i, i = 1, 2$ are i.i.d. random variables satisfying

$$W_i \sim \text{Beta}(\alpha, \beta).$$

Rewriting the cost function explicitly in integral form, the problem becomes computing

$$\min_{u \in U} \int_0^1 \int_0^1 [u^2 + (w_0^2 + 10w_1^2)u] p(w_0) p(w_1) \ dw_0 dw_1, \tag{4.9}$$

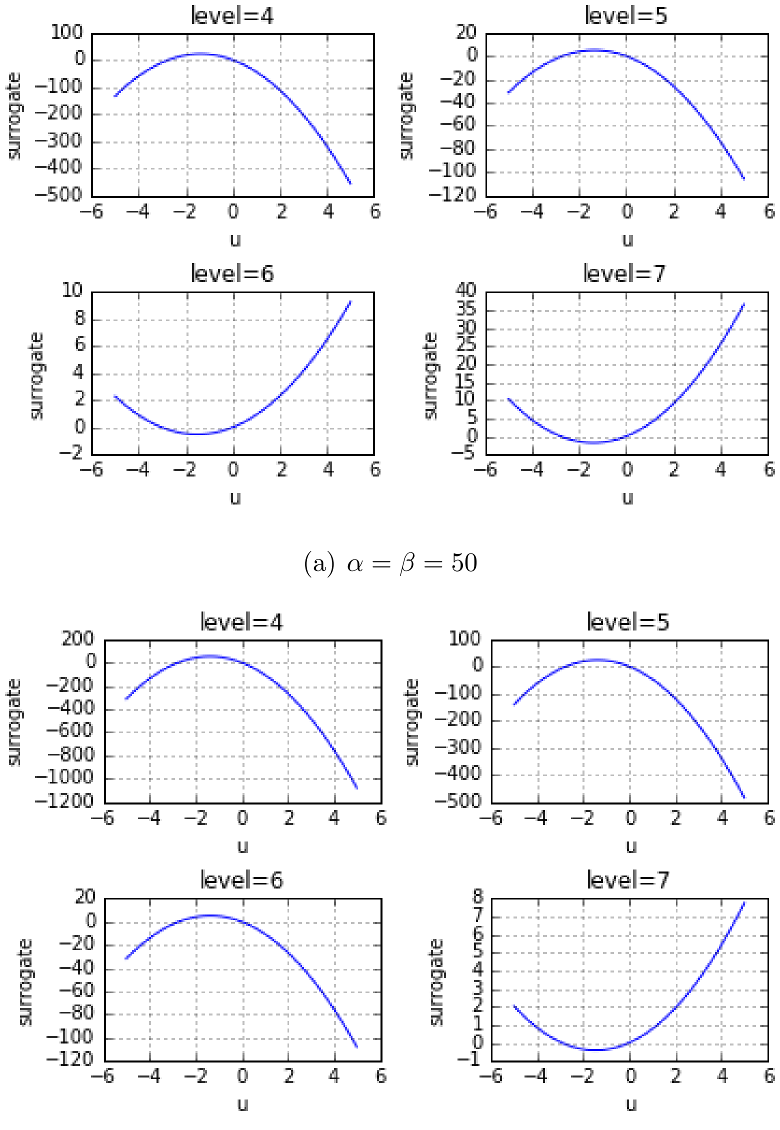
where p is the probability density function of W_i.

The objective function is strictly convex over U and $\frac{\partial^2 F}{\partial u^2} = 2$ at any point of U. The exact solution u^* is

$$u^* = -\frac{1}{2}(\mathbb{E}\left[W_0^2\right] + 10\,\mathbb{E}\left[W_1^2\right]). \tag{4.10}$$

We consider the following two cases. When $\alpha = \beta = 50$, the exact solution of (4.9) is $u^* \approx -1.38861$. When $\alpha = \beta = 100$, the exact solution is $u^* \approx -1.38184$.

Figure 4.1 shows the sparse grid surrogates of the example with levels $4, 5, 6$ and 7 for the above two cases. The univariate quadrature rule we used in this



(a) $\alpha = \beta = 50$

(b) $\alpha = \beta = 100$

Figure 4.1: The dependence of the convexity of the sparse grid surrogate on the level

example is Gauss–Patterson rule. In the first case, we see from Figure 4.1(a) that the surrogate function is concave when $l = 4, 5$ and becomes convex when the level increases to $l = 6, 7$. The minimizer of the surrogate function with $l = 4, 5$ is 5 on the boundary of U which is far from the the exact solution u^* which is an interior point. However, the computed minimizers are -1.50965 and -1.39318 for $l = 6, 7$ respectively which are much better approximations for u^*. Figure 4.1(b) presents a more extreme example. It shows the same pattern of the change of convexity as that in Figure 4.1(a). The computational results in Figure 4.1 agree with Theorem 4.6 since the sparse grid surrogate becomes strictly convex when the level l is large enough under given assumptions.

4.3 Dimension Adaptive Quadrature

In section 2.7, we have defined the generalised combination technique with downset I as

$$P_I(f) = \sum_{\alpha\in I} \Delta_\alpha(f)$$

where Δ_α, $\alpha \in I$ is a d dimensional surplus operator which is the tensor product of the 1D surplus operators $\Delta_{\alpha_k} = P_{\alpha_k} - P_{\alpha_k - 1}$, $k = 1, \dots, d$. P_{α_k} is the projection as defined in section 2.7. For the quadrature rules of Lagrangian interpolatory type on domain $\Omega = \Omega_1 \times \cdots \times \Omega_d$, we have

$$Q_I(f) := \int_\Omega L_I(f)\,dx = \int_\Omega \sum_{\alpha\in I} \Delta_\alpha(f)(x)\,dx = \sum_{\alpha\in I} \delta_\alpha(f) \tag{4.11}$$

where

$$\begin{aligned}\delta_\alpha(f) :&= \int_\Omega \Delta_\alpha(f)(x)\,dx = \int_\Omega (\bigotimes_{k=1}^d \Delta_{\alpha_k})(f)(x)\,dx \\ &= \int_{\Omega_1} \cdots \int_{\Omega_d} \prod_{k=1}^d \underbrace{(I \otimes \cdots \otimes \Delta_{\alpha_k} \otimes \cdots \otimes I)}_{\text{kth slot}}(f)(x)\,dx \\ &= \prod_{k=1}^d [E_k \circ \underbrace{(I \otimes \cdots \otimes \Delta_{\alpha_k} \otimes \cdots \otimes I)}_{\text{kth slot}}](f).\end{aligned}$$

Here E_k is the integral operator defined by

$$E_k(f) = \int_{\Omega_k} f(x)\,dx_k.$$

If we define the 1D quadrature surplus for a 1D function g as

$$\delta_{\alpha_k}(g) := \int_{\Omega_k} \Delta_{\alpha_k}(g)(x_k)\,dx_k,$$

then we have

$$\begin{aligned}\delta_\alpha &= \prod_{k=1}^{d} \underbrace{(I \otimes \cdots \otimes \delta_{\alpha_k} \otimes \cdots \otimes I)}_{\text{kth slot}} \\ &= \bigotimes_{k=1}^{d} \delta_{\alpha_k}.\end{aligned}$$

From the definition of the 1D quadrature surplus, we can also write it as

$$\delta_{\alpha_k} = Q_{\alpha_k} - Q_{\alpha_k - 1}$$

where Q_{α_k} is the quadrature rule derived from the 1D interpolant L_{α_k}.

If the downset I is chosen before we compute the quadrature, the equation in (4.11) computes a generalised combination technique. A dimension adaptive combination technique/sparse grid is a special kind of generalised combination technique when the downset I is decided during the computation according to the importance of each dimension. We have mentioned the idea of it in the Remark 2.58. We now look into details.

There are two important things which need to be considered before designing the dimension adaptive algorithm. First, when we add a new surplus $\Delta_\alpha f$ to the sum, we need to make sure the newly generated index set $I \cup \{\alpha\}$ is still a downset. This is because we need to use the method of differences to compute the telescope sum and thus every index β which have smaller entries than α in at least one dimension must be included in I. Second, the algorithm is required to detect the 'important dimension' and do refinement first in 'the most important dimension' at each iteration.

Algorithm 1 Dimension Adaptive Sparse Grid Quadrature

```
Initialize $I = \{\underline{1}\}$ and $s = \delta_\alpha f$
while Termination condition not reached do
    Consider all possible covering elements to $I$ and put them in a heap $\mathcal{A}$
    Select $\alpha$ from heap $\mathcal{A}$ with largest $\delta_\alpha f$
    $s = s + \delta_\alpha f$
end while
```

The termination condition we considered here is

$$|\delta_\alpha f| < \epsilon \text{ or } \alpha_k > \gamma_k, \text{ for some } \ k. \tag{4.12}$$

Therefore, the corresponding downset I is

$$I = \{\alpha \leq \gamma \,|\, |\delta_\alpha f| \geq \epsilon\}.$$

The termination condition $|\delta_\alpha f| < \epsilon$ has been used in many dimension adaptive sparse grid algorithms to stop the while loop [46, 45, 49, 75]. The additional condition $\alpha \leq \gamma$ we added here is aimed at avoiding excessive refinement in some dimensions. Here, we say $\{\alpha\}$ is a covering element of a downset I if $I \cup \{\alpha\}$ is also a downset.

We use $\mathcal{Q}_{\gamma,\epsilon}$ as the operator for the dimension-adaptive sparse grid quadrature in Algorithm 1 with the termination condition (4.12). The choice of the downset I depends on f, γ, ϵ, so we have

$$\mathcal{Q}_{\gamma,\epsilon}(f) = Q_{I(f,\gamma,\epsilon)}(f).$$

It should be noted here we have to use different notations for the quadrature method($\mathcal{Q}_{\gamma,\epsilon}$) and the computing formula(Q_I). This is because when the same quadrature method applied to approximate different integrals, e.g. integrals with integrand f and g, respectively, we can get different downsets I_f and I_g,

$$I_f = I(f, \gamma, \epsilon) \neq I(g, \gamma, \epsilon) = I_g.$$

Thus, the formulas used to approximate the integrals are different, that is,

$$Q_{I_f} \neq Q_{I_g}.$$

For the non-adaptive approach, we don't have such problem. We use the same notation(Q) to denote both quadrature method and computing formula.

4.3.1 1D quadrature rules

We will use 1D quadrature rules introduced in the Section 1.2 to build the dimension adaptive sparse grid quadrature. They are the trapezoidal rule/Newton Cotes formulas, Clenshaw-Curtis rule and Gauss-Patterson rule. All of these quadrature rules are quadrature rules of interpolatory type. Therefore, the definition (4.11) is applicable to all three cases.

4.3.2 Accuracy of the Dimension Adaptive Quadrature

We have studied the convergence of generalised combination technique when the error splitting model (2.13) is given in the Section 2.6. Here we further study the error of the dimension adaptive sparse grid quadrature with the termination condition (4.12). In [48], the author gives an error analysis of the dimension-adaptive sparse grid interpolation. Similar as their analysis, we first have the following a priori bound on $|Q_L f - Q_I f|$.

Proposition 4.7. *(A priori error bound) Let $I = \{\alpha \leq \gamma \,|\, |\Delta_\alpha f| \geq \epsilon\}$ and $Q_L f - Q_I f$ be the error of the dimension adaptive sparse grid quadrature on set I relative to $Q_\gamma f$. Here $L = \{\alpha \,|\, \alpha \leq \gamma\}$. Then we get the bound*

$$|Q_L f - Q_I f| \leq |L|\epsilon.$$

Proof. According to the definition, we have

$$\begin{aligned}|Q_L f - Q_I f| &= |\sum_{\alpha\in L}\Delta_\alpha f - \sum_{\alpha\in I}\Delta_\alpha f| = |\sum_{\alpha\in L\backslash I}\Delta_\alpha f| \\ &\leq \sum_{\alpha\in L\backslash I}|\Delta_\alpha f| \leq \sum_{\alpha\in L\backslash I}\epsilon \leq \sum_{\alpha\in L}\epsilon = |L|\epsilon.\end{aligned}$$

The first inequality follows from the triangle inequality. The second inequality holds because $L\backslash I = \{\alpha \leq \gamma \,|\, |\Delta_\alpha f| < \epsilon\}$. The third inequality follows by the fact $I \subset L$. □

From the proof of the proposition 4.7, we do not use any information about the computational process of $Q_I f$. The error bound can be derived before computing $Q_I f$. However, after we compute $Q_I f$ by the dimension-adaptive sparse grid method, we will know exactly what the downset I is. This can help us improve this error bound.

Proposition 4.8. *(Posteriori error bound) Suppose the downset I is known after we computed the $Q_I f$. The error bound in* 4.7 *can be improved by*

$$|Q_L f - Q_I f| \leq (|L| - |I|)\epsilon,$$

where the set $L = \{\alpha \,|\, \alpha \leq \gamma\}$.

Proof. Since $I = \{\alpha \leq \gamma \,|\, |\Delta_\alpha f| \geq \epsilon\}$ is a subset of L, we have

$$|Q_L f - Q_I f| = |\sum_{\alpha\in L\backslash I}\Delta_\alpha f| \leq \sum_{\alpha\in L\backslash I}|\Delta_\alpha f| = (|L| - |I|)\epsilon.$$

□

When we compute a high dimensional integral, we do not set a very small ϵ, e.g. 10^{-15}, in the termination condition since the computational cost is usually unaffordable for such small ϵ in most examples. Thus, neither a priori error bound nor the posteriori error bound is accurate when we consider a high dimensional problem since $|L|$ will grow exponentially when the dimension increases while the ϵ can't be chosen as small as possible. In order to get a more accurate error bound, we need to utilise the smoothness of the integrand f. Recall in section 1.4 Lemma 1.34. We proved that if $f \in F_d^s$, then the quadrature surplus with multi-index α satisfies

$$|\delta_\alpha(f)| = \left| \int_X f_\alpha^h \, dx \right| \leq C_{s,d} 2^{-|\alpha| s} \|f\|_d^s$$

where the space F_d^s and the norm defined on this space are defined in (1.29) and (1.30). Therefore, we have

Theorem 4.9. *Under the conditions of Lemma* 1.35*, we can further improve the posteriori bound to*

$$|Q_L f - Q_I f| \leq K \sum_{\alpha \in L \backslash I} 2^{-r|\alpha|}. \tag{4.13}$$

Proof. The Theorem (4.9) follows directly from the proposition (4.8) and the Lemma (1.34). $K = C_{r,d} \|f\|_d^s$ is a constant. □

Corollary 4.10. *Under the conditions of Lemma*1.35*, if m is one of the indices such that $|m| \leq |\alpha|, \forall \alpha \in L \backslash I$, then the posterior bound is*

$$|Q_L f - Q_I f| \leq \frac{\epsilon}{\rho} \sum_{\alpha \in L \backslash I} 2^{r(|m| - |\alpha|)} \tag{4.14}$$

where $\rho K 2^{-r|m|} = \epsilon$ and

$$\rho_{\min} := \frac{\sum_{\alpha \in L \backslash I} 2^{r(|m| - |\alpha|)}}{|L| - |I|} \leq \rho < 2^r.$$

Proof. We first notice we can rewrite (4.13) as

$$|Q_L f - Q_I f| \leq K 2^{-r|m|} \sum_{\alpha \in L \backslash I} 2^{r(|m| - |\alpha|)}.$$

By using $\rho K 2^{-r|m|} = \epsilon$, we get the inequality (4.14). For the lower bound of ρ, we expect the error bound (4.14) is not worse than the posterior error bound in the proposition (4.8),otherwise we can use the latter one. Thus, we have

$$\frac{\epsilon}{\rho} \sum_{\alpha \in L \backslash I} 2^{r(|m| - |\alpha|)} \leq \epsilon(|L| - |I|),$$

This leads to the lower bound of ρ. For the upper bound, if we denote the kth unit vector as $e_k = [0, \ldots, 1, \ldots, 0]$, then according to the definition of m, there exists an index $m - e_k \in I$, otherwise we should choose $m - e_k$ instead of m in the theorem. According to the Theorem 4.9 and the definition of the downset I, we have

$$\epsilon < |\Delta_{m-e_k} f| \leq K 2^{-r(|m|-1)}.$$

Thus, using the definition of ρ, we have

$$\rho = \frac{\epsilon}{K 2^{-r|m|}} \leq \frac{K 2^{-r(|m|-1)}}{K 2^{-r|m|}} = 2^r.$$

□

By using the error estimation in the Theorem 4.10 and the error bound for d-dimensional product rule, we can obtain a bound on $|If - Q_I f|$ by using trapezoidal rule, that is

$$|If - Q_I f| \leq |If - Q_L f| + |Q_L f - Q_I f| \leq c_d 2^{-lr} + K \sum_{\alpha \in L \setminus I} 2^{-r|\alpha|} \tag{4.15}$$

In the above bound, isotropic grid $\mathcal{G}_\gamma$, $\gamma_i = l$ is used in the comparison. In [46], the authors get an optimised priori error bound for $|If - Q_I f|$ by balancing error bounds for the term $|If - Q_L f|$ and $|Q_L f - Q_I f|$. The ϵ need to be chosen very small in order to achieve the optimized bound for high dimensional problems. Here we are more interested in the case when

$$|If - Q_L f| \ll |Q_L f - Q_I f|$$

which means the approximation of the integral need to be accurate to some extent on the corresponding full grid, otherwise we can not expect the dimensional adaptive sparse grid method which uses a subset of quadrature points on the full grid provides a good approximation. Larger ϵ is allowed in this situation.

4.3.3 The Dimension Adaptive Surrogate

According to the definition of the dimension adaptive quadrature, a dimension adaptive surrogate of the stochastic optimisation problem (4.1) with parameters ϵ and γ can be defined as

$$F_I(u) = \mathcal{Q}_{\gamma,\epsilon}(f(u, \cdot)).$$

However, since the function f also depends on u, we can not obtain a concrete downset I. In fact, the choice of I also depends on the value of u, i.e.

$$I = I(f(u, \cdot), \gamma, \epsilon).$$

One way to deal with this problem is we take a specific $u \in U$. We can try the initial value u_0 in the optimisation algorithm. However, if the initial value u_0 is very far away from the exact solution u^*, the accuracy of the dimension adaptive quadrature will be bad since we do not use the downset at u^*, i.e. $I(f(u^*, \cdot), \gamma, \epsilon)$. Therefore, it is better to use different surrogates at different iterations. This leads to the idea of the 'optimise then discretise' method.

4.4 The 'Optimise then Discretise' Method

We will show the framework of the 'optimise then discretise' method and test a 2D example to illustrate its performance. For simplicity, we first use Newton method as our optimisation algorithm. It is shown in the Algorithm 2. The Algorithm 3, 4, 5, 6 and 7 are discretised versions of the Newton method in Algorithm 2 from simple to complex. In the Algorithm 3 and 4, we use non-adaptive quadrature Q to compute the integrals. In the Algorithm 6 and 7, we use the dimension adaptive quadrature $\mathcal{Q}$ to compute the integrals. The notation D_i denotes the ith discretised derivative and thus D^2_{ij} is the ijth second order discretised derivative. D denotes the discretised gradient and D^2 denotes the discretised Hessian.

In Algorithm 3, we use different non-adaptive surrogate at each iteration. The non-adaptive quadrature operator Q and the discretised derivative operators D_i, D_{ij} are commutative, i.e.

$$D_i Q_p = Q_p D_i \text{ and } D^2_{ij} Q_p = Q_p D^2_{ij}.$$

This is because both two operators are fixed finite summations. In Algorithm 4, we further allow different choices of the non-adaptive quadrature for the objective and different component of gradient and Hessian.

The Algorithm 5 looks almost the same as the Algorithm 3 except the quadrature method is dimension adaptive. However, they have essential differences. The dimension adaptive operator $\mathcal{Q}$ and the discretised derivative operators D_i, D_{ij} are not commutative. This observation leads to a new Algorithm 6. The reason why the dimension adaptive operator and the discretised derivative operators are not commutative is because the downsets used in the computation are not equal, i.e.

$$I_{(D_i f, \epsilon_p, \gamma_p)} \neq I_{(f, \epsilon_p, \gamma_p)}$$
$$I_{(D^2_{ij} f, \epsilon_p, \gamma_p)} \neq I_{(f, \epsilon_p, \gamma_p)}.$$

One can also generalise the Algorithm 6 to the Algorithm 7 by allowing the usage of different parameters in dimension adaptvie quadrature for the objective and different component of gradient and Hessian. Though Algorithm 7 will be more flexible than Algorithm 6, we mostly use Algorithm 6 in practice because it is usually hard to get information used for choose different parameters and the algorithm 7 is too complex.

Algorithm 2 OPTIMISE

```
Take an initial $u_0 \in \mathbb{R}^n$ and $p := 0$
Compute $G_0 = \nabla F(u_0)$
while $\|G_p\| > \epsilon$ do
    Compute the Hessian $H_p = \nabla^2 F(u_p)$
    Update
        $u_{p+1} = u_p - H_p^{-1} G_p$
    Set $p := p + 1$
    Compute $G_p = \nabla F(u_p)$
end while
Output $u_p$ and $F(u_p)$
```

Algorithm 3 DISCRETISED VERSION

```
Take an initial $\bar{u}_0 \in \mathbb{R}^n$ and $p := 0$
Compute the approximation of the gradient $\bar{G}_0 = DQ_0(f(\bar{u}_0, \cdot))$
while $\|\bar{G}_p\| > \epsilon$ do
    Compute the approximation of the Hessian $\bar{H}_p = D^2 Q_p(f(\bar{u}_p, \cdot))$
    Update
        $\bar{u}_{p+1} = \bar{u}_p - \bar{H}_p^{-1} \bar{G}_p$        (4.16)
    Set $p := p + 1$
    Compute the approximation of the gradient $\bar{G}_p = DQ_p(f(\bar{u}_p, \cdot))$
end while
Output $\bar{u}_p$ and $\bar{F}_p := Q_p(f(\bar{u}_p, \cdot))$
```

For more complicated quasi Newton methods, we take BFGS method with the exact line search as an example. The optimise algorithm and its adaptive discretised version are shown in the Algorithm 8 and Algorithm 9 which are extensions of the Algorithm 2 and the Algorithm 6, respectively. In practice, the exact line search (4.21) is replaced with inexact line search for efficiency.

Algorithm 4 GENERAL DISCRETISED VERSION

```
Take an initial $\bar{u}_0 \in \mathbb{R}^n$ and $p := 0$
Compute the approximation of the gradient $\bar{G}_0 = [Q^G_{0,i}(D_i f(\bar{u}_0, \cdot))]_{n \times 1}$
while $\|\bar{G}_p\| > \epsilon$ do
    Compute the approximation of the Hessian $\bar{H}_p = [Q^H_{p,i,j}(D^2_{ij} f(\bar{u}_p, \cdot))]_{n \times n}$
    Update
        $$\bar{u}_{p+1} = \bar{u}_p - \bar{H}_p^{-1} \bar{G}_p \qquad (4.17)$$
    Set $p := p + 1$
    Compute the approximation of the gradient $\bar{G}_p = [Q^G_{p,i}(D_i f(\bar{u}_p, \cdot))]_{n \times 1}$
end while
Output $\bar{u}_p$ and $\bar{F}_p := Q^O_p(f(\bar{u}_p, \cdot))$
```

Algorithm 5 DISCRETISED VERSION(ADAPTIVE)

```
Take an initial $\bar{u}_0 \in \mathbb{R}^n$ and $p := 0$
Compute the approximation of the gradient $\bar{G}_0 = D\mathcal{Q}_{\epsilon_0,\gamma_0}(f(\bar{u}_0, \cdot))$
while $\|\bar{G}_p\| > \epsilon$ do
    Compute the approximation of the Hessian $\bar{H}_p = D^2\mathcal{Q}_{\epsilon_p,\gamma_p}(f(\bar{u}_p, \cdot))$
    Update
        $$\bar{u}_{p+1} = \bar{u}_p - \bar{H}_p^{-1} \bar{G}_p \qquad (4.18)$$
    Set $p := p + 1$
    Compute the approximation of the gradient $\bar{G}_p = D\mathcal{Q}_{\epsilon_p,\gamma_p}(f(\bar{u}_p, \cdot))$
end while
Output $\bar{u}_p$ and $\bar{F}_p := \mathcal{Q}_{\epsilon_p,\gamma_p}(f(\bar{u}_p, \cdot))$
```

Algorithm 6 MODIFIED DISCRETISED VERSION(ADAPTIVE)

```
Take an initial $\bar{u}_0 \in \mathbb{R}^n$ and $p := 0$
Compute the approximation of the gradient $\bar{G}_0 = [\mathcal{Q}_{\epsilon_0,\gamma_0}(D_i f(\bar{u}_0, \cdot))]_{n \times 1}$
while $\|\bar{G}_p\| > \epsilon$ do
    Compute the approximation of the Hessian $\bar{H}_p = [\mathcal{Q}_{\epsilon_p,\gamma_p}(D^2_{ij} f(\bar{u}_p, \cdot))]_{n \times n}$
    Update
        $$\bar{u}_{p+1} = \bar{u}_p - \bar{H}_p^{-1} \bar{G}_p \qquad (4.19)$$
    Set $p := p + 1$
    Compute the approximation of the gradient $\bar{G}_p = [\mathcal{Q}_{\epsilon_p,\gamma_p}(D_i f(\bar{u}_p, \cdot))]_{n \times 1}$
end while
Output $\bar{u}_p$ and $\bar{F}_p := \mathcal{Q}_{\epsilon_p,\gamma_p}(f(\bar{u}_p, \cdot))$
```

Algorithm 7 GENERAL DISCRETISED VERSION(ADAPTIVE)

```
Take an initial $\bar{u}_0 \in \mathbb{R}^n$ and $p := 0$
Compute the approximation of the gradient
    $\bar{G}_0 = [\mathcal{Q}^G_{\epsilon_{0,i},\gamma_{0,i}}(D_i f(\bar{u}_0, \cdot))]_{n\times 1}$
while $\|\bar{G}_p\| > \epsilon$ do
    Compute the approximation of the Hessian
        $\bar{H}_p = [\mathcal{Q}^H_{\epsilon_{p,i,j},\gamma_{p,i,j}}(D^2_{ij} f(\bar{u}_p, \cdot))]_{n\times n}$
    Update
        $\bar{u}_{p+1} = \bar{u}_p - \bar{H}_p^{-1}\bar{G}_p \qquad (4.20)$
    Set $p := p + 1$
    Compute the approximation of the gradient
        $\bar{G}_p = [\mathcal{Q}^G_{\epsilon_{p,i},\gamma_{p,i}}(D_i f(\bar{u}_p, \cdot))]_{n\times 1}$
end while
Output $\bar{u}_p$ and $\bar{F}_p := \mathcal{Q}^O_{\epsilon_p,\gamma_p}(f(\bar{u}_p, \cdot))$
```

The commonly used inexact line search is strong Wolfe's rule. The sequence u_p generated by BFGS with Wolfe's rule is proved to converge to the exact minimizer u^* superlinearly [54]. It should be noted that we need to compute the objective F in the line search methods(include the strong Wolfe's rule used in practice)in each iteration while this is not required if we use Newton method.

Algorithm 8 BFGS OPTIMISE

```
Take an initial $u_0 \in \mathbb{R}^n$ , an initial positive definite matrix $H_0$ and $p := 0$
while $\|G(u_p)\| > \epsilon$ do
    Compute the search direction $v_p = -H_p G(u_p)$
    Find the step length $\alpha_p$ by exact line search
        $$\min_{\alpha_p} F(u_p + \alpha_p v_p). \tag{4.21}$$
    The underlying $A_p^{-1}$ here is $\alpha_p H_p$.
    Update
        $$u_{p+1} = u_p + \alpha_p v_p$$
    Define $s_p := u_{p+1} - u_p$ and $y_p := G(u_{p+1}) - G(u_p)$
    Update
        $$H_{p+1} = \left(I - \frac{s_p y_p^T}{s_p^T y_p}\right) H_p \left(I - \frac{y_p s_p^T}{s_p^T y_p}\right) + \frac{s_p s_p^T}{s_p^T y_p}$$
    p:=p+1
end while
Output $u_p$ and $F(u_p)$
```

In order to illustrate our method, we provide the following 2D example. We will look into this example and it will also be used to explain the idea of the next two sections.

Example 4.10.1. *We consider the following minimization problem*

$$\min_{u \in \mathbb{R}} F(u)$$

where $F(u) = \mathbb{E}\left[u^2 + (W_1^2 + 10W_2^2)u\right]$. W_1 *and* W_2 *are i.i.d. random variables. Moreover, the objective function is strictly convex in this example, so we conclude that there is a unique minimizer of this problem. By using the linearity of the expectation, the minimizer of the problem is*

$$u^* = -\frac{\mathbb{E}[W_1^2] + 10\,\mathbb{E}[W_2^2]}{2}.$$

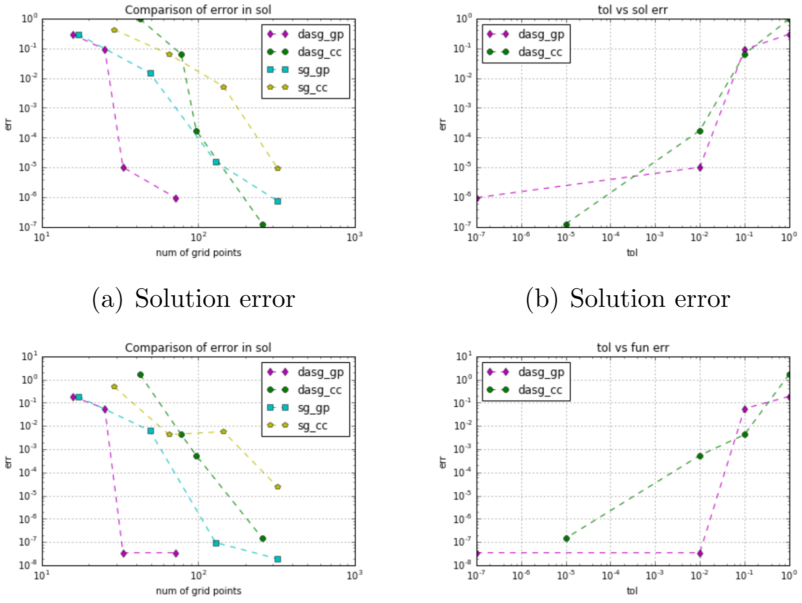
Algorithm 9 BFGS DISCRETISE

```
Take an initial $\bar{u}_0 \in \mathbb{R}^n$, an initial positive definite matrix $\bar{H}_0$ and $p := 0$
Compute $\bar{G}_0 = [\mathcal{Q}_{\epsilon_0,\gamma_0}(D_i f(\bar{u}_0, \cdot))]_{n\times 1}$
while $\|\bar{G}_p\| > \epsilon$ do
    Compute the search direction $\bar{v}_p = -\bar{H}_p \bar{G}_p$
    Find the step length $\bar{\alpha}_p$ by exact line search
        $$\min_{\bar{\alpha}_p} \bar{F}_p(\bar{u}_p + \bar{\alpha}_p \bar{v}_p)$$
    where $\bar{F}_p := \mathcal{Q}_{\epsilon_p,\gamma_p}(f(\bar{u}_p, \cdot))$ and the corresponding $\bar{A}_p^{-1}$ is $\bar{\alpha}_p \bar{H}_p$
    Update
        $$\bar{u}_{p+1} = \bar{u}_p + \bar{\alpha}_p \bar{v}_p$$
    Compute $\bar{G}_{p+1} = [\mathcal{Q}_{\epsilon_{p+1},\gamma_{p+1}}(D_i f(\bar{u}_{p+1}, \cdot))]_{n\times 1}$
    Define $\bar{s}_p := \bar{u}_{p+1} - \bar{u}_p$ and $\bar{y}_p := \bar{G}_{p+1} - \bar{G}_p$
    Update
        $$\bar{H}_{p+1} = \left(I - \frac{\bar{s}_p \bar{y}_p^T}{\bar{s}_p^T \bar{y}_p}\right) \bar{H}_p \left(I - \frac{\bar{y}_p \bar{s}_p^T}{\bar{s}_p^T \bar{y}_p}\right) + \frac{\bar{s}_p \bar{s}_p^T}{\bar{s}_p^T \bar{y}_p}$$
    Set p:=p+1
end while
Output $\bar{u}_p$ and $\bar{F}_p := \mathcal{Q}_{\epsilon_p,\gamma_p}(f(\bar{u}_p, \cdot))$
```

In particular, here we further assume

$$W_k \sim \text{ Beta}(\alpha, \beta),\ k = 1, 2.$$

with $\alpha = 5, \beta = 5$*. The exact minimizer is then* $u^* = -1.5$ *and the minimum is* -2.25*.*



(a) Solution error (b) Solution error

(c) Objective error (d) Objective error

Figure 4.2: Computational results for the 2D problem: (a) errors of computed minimizer vs. number of grid points used in each iteration(average for the dimension-adaptive sparse grid method).(b) errors of computed minimizer vs. ϵ in the termination condition of the dimension-adaptive algorithm. (c) errors of computed minimum vs. number of grid points used in each iteration. (d) errors of computed minimum vs. ϵ in the termination condition of the dimension-adaptive algorithm. We compare the dimension-adaptive sparse grid based on Gauss–Patterson(dasg_gp) and Clenshaw–Curtis(dasg_cc) with sparse grid based on Gauss–Patterson(sg_gp) and Clenshaw–Curtis(sg_cc).

In Figure 4.2, we apply the Algorithm 9 to solve the problem. We use forward difference to approximate the derivatives of the integrand. We compare the performance of the dimension-adaptive sparse grid quadrature and the sparse grid quadrature. For the OTDM based on the sparse grid quadrature, we fix the level l for each run which results in the same choices of the quadrature rule Q_p and $Q_{p,i}$ for any p. For the OTDM based on the dimension-adaptive sparse grid quadrature, we also fix the ϵ and γ in the while condition 4.12. However, the choices of the quadrature rule are no longer the same when we used to compute the objective and each component of the gradient and Hessian. This is because the underlying downsets are not necessary to be the same. We see in Figure 4.2 that the convergence rates are improved for both Clenshaw–Curtis and Gauss–Patterson when we apply the dimension-adaptive method. Especially, the average number of the grid points used in each iteration is substantially reduced for the same accuracy in the solution and the objective when Gauss– Patterson is used as 1D rule. For Clenshaw–Curtis, we can also see this pattern, moreover, higher accuracy is obtained for both computed minimizer and minimum.

4.5 The OTDM and the DTOM

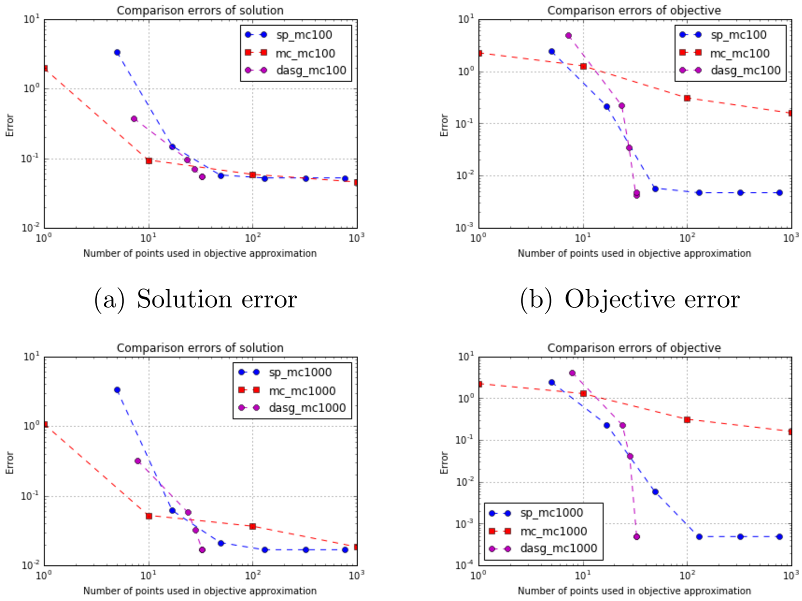
From the Algorithm 3, 4, 5, 6 and 7, we know that the OTDM is a kind of generalisation of the DTOM. Under the framework of OTDM, different discretisations to the integrals from the objective, the gradient/Hessian in different iterations are allowed. This will make the computation more flexible. First, since we can use different quadrature rules at different iterations, the dimension-adaptive sparse grid approach can achieve its full potential in saving computational cost. Otherwise, under the framework of the DTOM, we have to fix the downset I at all iteration and the fixed downset cannot be the 'best' choice for all iterations. Second, for some problems, we do not need very accurate approximation of the gradient/Hessian (e.g. when the minimiser is on the boundary). In this case, we can use low accuracy quadrature rule to approximate the gradient/Hessian while high accuracy quadrature to approximate the objective. Third, if we can get the exact value of the gradient/Hessian, we can make use of these information and devise better numerical scheme for the problem when we apply the OTDM.

The exact expression of the gradient in the previous 2D Example 4.10.1 is

$$\begin{aligned} G(u) &= \nabla \mathbb{E}[u^2 + (W_1^2 + 10W_2^2)u] \\ &= \int_0^1 \int_0^1 [2u + (w_1 + 10w_2)] p(w_1, \alpha, \beta) p(w_2, \alpha, \beta)\, dw_1 dw_2 \end{aligned} \tag{4.22}$$

where p is the probability density function.

In Figure 4.3, we again solved problem with the BFGS method. We use 100 points and 1000 points Monte Carlo method to approximate the integral $G(u)$ respectively. The sparse grid, the dimensional-adaptive sparse grid based on Gauss–Patterson 1D quadrature and the Monte Carlo are used in approximating the objective function. We intentionally choose the same random points for Monte Carlo in objective approximation with those in gradient approximation when number



(a) Solution error

(b) Objective error

(c) Solution error

(d) Objective error

Figure 4.3: Computational results for the toy problem. Here we use the exact expression of the gradient. Monte Carlo method is used to approximate the gradient and three different quadrature methods are used to approximate the objective function. (a) number of grid points vs. errors of computed minimizers(average of 10 runs of each method with gradient approximated by 100 points Monte Carlo method. (b)number of grid points vs. errors of computed minimum. (c)number of grid points vs. errors of computed minimizers(1000 points Monte Carlo). (d)number of grid points vs. errors of computed minimum.

of points equals to 100 in (a), (b) and 1000 in (c), (d). Thus, according to the propositions, the Monte Carlo surrogate methods are actually used. We can see from the Figure 4.3, both convergence performances of sparse grid and dimensional adaptive sparse grid are better than the Monte Carlo method (include those points which is actually surrogate method). The amount of work will be substantially reduced in the objective function if we apply the dimension-adaptive sparse grid in computing the objective. In addition, the dimension-adaptive sparse grid method performs better than sparse grid method in computing both solution and objective.

4.6 Convergence and Stopping Criterion

In the previous sections, we actually present a few optimisation algorithms in the presence of errors. The error in the objective, gradient and Hessian evaluations can affect the convergence of the original optimisation algorithms(Newton method and BFGS). Consider solving

$$\min_{u\in U} F(u)$$

where $F(u)$ is defined in (4.2). For the Newton method without error in discretisation, its convergence is based on the following assumptions [52, 53]. First, if u^* is a critical point of F, then $\nabla F(u^*) = 0$ and $\nabla^2 F(u^*)$ is positive definite. Second, $\nabla^2 F(u^*)$ is required to be Lipschitz continuous, i.e. there is $L > 0$ such that

$$\|\nabla^2 F(u) - \nabla^2 F(v)\|_2 \leq L\|u - v\|_2, \forall u, v \in U.$$

For the BFGS method without error in discretisation, its convergence requires the following assumptions [52, 53]. First, the objective function $F \in C^2(U)$. Second, for the initial value u_0, we require the sublevel set

$$L_F^{-1}(F(u_0)) := \{u \in U \subset \mathbb{R}^n \mid F(u) \leq F(u_0)\}$$

is convex and there exists constants $0 < m \leq M$ such that

$$m\|v\|_2^2 \leq v^T \nabla^2 F(u) v \leq M\|v\|_2^2$$

for all $v \in \mathbb{R}^n$ and $u \in L_F^{-1}(F(u_0))$.

In the Algorithm 6 and Algorithm 9, if we let

$$\epsilon_p \to 0,\ \gamma_p \to \infty \text{ as } p \to \infty, \tag{4.23}$$

then we will have

$$\bar{F}_p \to F_p,\ \bar{G}_p \to G_p\ \bar{H}_p \to H_p.$$

Therefore, in this case, the Algorithm 6/Algorithm 9 converges when the assumptions for Newton method/BFGS method(without errors) are satisfied. However, if we fixed $\epsilon_p = \epsilon$ and $\gamma_p = \gamma$ in all iterations of the Algorithm 6 and Algorithm 9, errors which appear in evaluations of the objective, gradient and Hessian will have influence on the convergence result. The convergence of the Newton method with errors is studied in [90, 28]. The result of the convergence of the BFGS method with errors can be found in a recent paper [89]. For both methods, when the errors are (uniformly) bounded, the sequence $\{\bar{u}_p\}$, $p \in \mathbb{N}$ generated by the algorithm converges to a neighbourhood of the exact solution that is determined by the size of the errors. In addition, after the sequence $\{\bar{u}_p\}$, $p \in \mathbb{N}$ reaches that neighbourhood, the behaviour of the sequence generated by the successive iterations is not predictable. It can be either converge to a point which is not equal to the exact solution or even diverge. Therefore, we need to avoid more iterations after the first time the computed sequence $\{\bar{u}_p\}$, $p \in \mathbb{N}$ reaches the neighbourhood. A proper stopping criterion is required.

Here we provide a numerical way to estimate the time to stop iterations in the Algorithms. Suppose $\bar{u}_p$ is the approximated minimizer generated by some Newton-type methods after pth iteration. When $\bar{u}_p$ is close enough to the exact minimizer u^*, we have the following Taylor expansion

$$F(\bar{u}_p) = F(u^*) + \nabla F(u^*)(\bar{u}_p - u^*) + (\bar{u}_p - u^*)^T \nabla^2 F(u^*)(\bar{u}_p - u^*) + o(\|\bar{u}_p - u^*\|^2).$$

Since u^* is the minimizer of F, we have $\nabla F(u^*) = 0$. F is convex, therefore $\nabla^2 F(u^*)$ is positive semidefinite. Thus, if $\|\bar{u}_p - u^*\|^2$ increases(decreases),

$$(\bar{u}_p - u^*)^T \nabla^2 F(u^*)(\bar{u}_p - u^*)$$

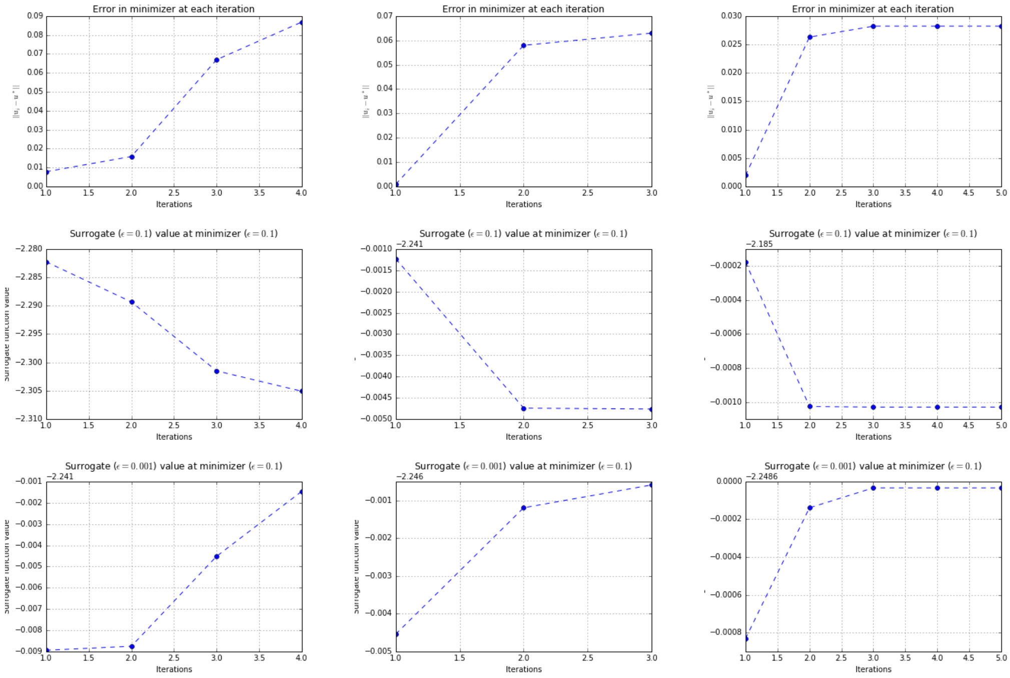
will not decrease(not increase) and therefore $F(\bar{u}_p)$ will not decrease (not increase). However, $F(\bar{u}_p)$ is a high dimensional integral in our case, we can not get the exact value in most cases. Therefore, in our approach, instead of using the exact function value at $\bar{u}_p$, we use the value of some high accuracy approximations of the function. If we denote the high accuracy approximation(surrogate of the objective) by dimension-adaptive sparse grid quadrature with ϵ^* and γ^* at u as

$$\bar{F}_{\epsilon^*,\gamma^*}(u) := \mathcal{Q}_{\epsilon^*,\gamma^*}(f(u,\cdot)),$$

then we can decide when to stop the algorithm based on Newton-type methods by studying the trend of $\bar{F}_{\epsilon^*,\gamma^*}(\bar{u}_p)$, $p \in \mathbb{N}$.

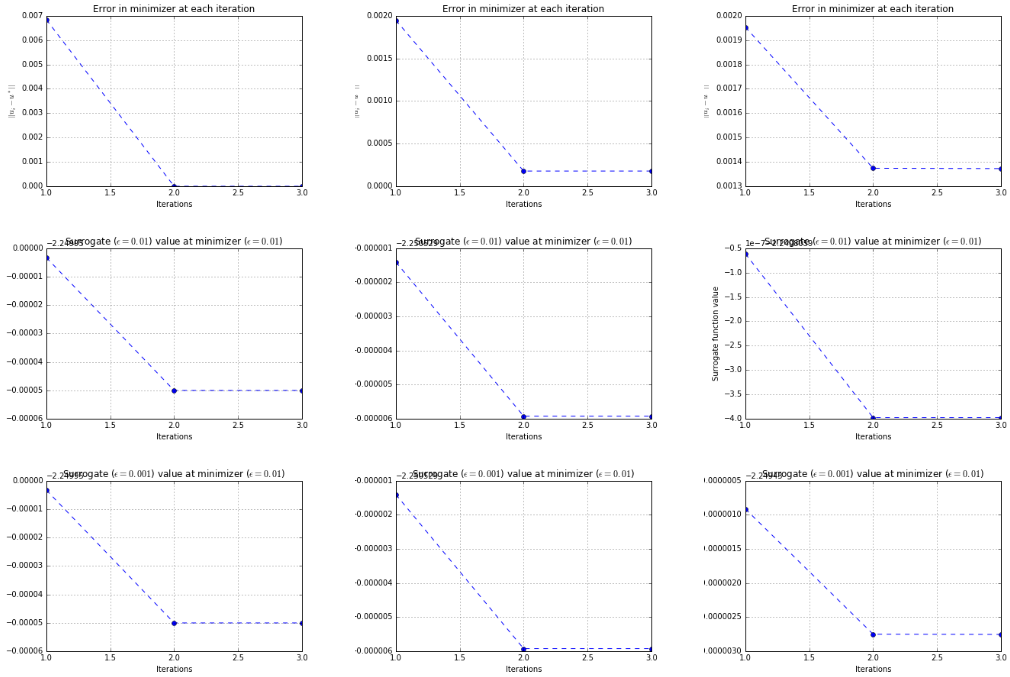
The advantage of this method is that we only need to compute $\bar{F}_{\epsilon^*,\gamma^*}(\bar{u}_p)$ with $\epsilon^* < \epsilon$ and $\gamma^* > \gamma$ with the same algorithm which used to compute the objective, gradient and Hessian in the iterations. Also, the additional computational cost for computing $\bar{F}_{\epsilon^*,\gamma^*}(\bar{u}_p)$ is affordable in most cases for even high dimensional problems. This is because we use Newton-type method as our optimisation solver. Thus the number of iterations will not be too large. Also, the computational cost of getting such stopping criterion is much lower than that of computing the gradient and Hessian in each iteration when the dimension of U is high.

In Figure 4.4 and Figure 4.5, we solve the Example 4.10.1 using surrogate method with fixed $\epsilon = 0.1$ and $\epsilon = 0.01$ respectively. We do not set any restriction on γ in this example which means $\gamma = \infty$. We show the results for three frequently used quadratures, i.e. the trapezoidal rule, the Clenshaw-Curtis rule and the Gauss-Patterson rule. For both two figures, the subfigures in the first row show the relation between error $|u^* - \bar{u}_p|$ versus the number of iterations. We can see



(a) Gauss Patterson (b) Clenshaw Curtis (c) Trapezoidal Rule

Figure 4.4: Solve the problem in the Example 4.10.1 with $\epsilon = 0.1$.



(a) Gauss Patterson (b) Clenshaw Curtis (c) Trapezoidal Rule

Figure 4.5: Solve the problem in the Example 4.10.1 with $\epsilon = 0.01$.

from these figures, we should stop the algorithm after the first iteration in the Figure 4.4 while after second iteration in the Figure 4.5. If we further increase the number of iterations, the errors will not decrease for both cases. The reason for this is the gradients are approximated with relatively low accuracy. From the convergence theory, we know the convergence of the Newton-type method with error might break down in this case. The subfigures in the second row present function values on the surrogate versus the number of iterations. As we expected, $F_{\epsilon,\gamma}(\bar{u}_p)$ are decreasing. The subfigures in the third row show the function values of the surrogate functions with $\epsilon = 0.001$ on point $\bar{u}_p$. Comparing the subfigures in the first row with the corresponding subfigures in the third row, we can see that the trends of the functions are the same for all three quadrature rules. Thus, we can predict when to stop the Newton-type method by studying the trends of the functions in the third row respectively. Our method successfully predicts the stopping times for all three quadrature rules in this example.

4.7 Numerical Experiments

4.7.1 A High Dimension Example

Consider the following minimization problem

$$\min_{u\in U} \mathbb{E}\left[\sum_{i=1}^{d} \exp\left(-u_i W_i^2\right)\right]$$

where W_i are i.i.d random variables which are subject to the uniform distribution on $[0,1]$ and $u \in U = [0,1]^d$. Thus the integral form of the objective function can be written as

$$F(u) = \int_{[0,1]^d} \sum_{i=1}^{d} e^{-u_i w_i^2}\, dw.$$

The gradient $G(u)$ is

$$\begin{aligned} G(u) &= \nabla_u F(u) \\ &= -\left[\int_{[0,1]^d} w_1^2 e^{-u_1 w_1^2}\, dw_1, \ldots, \int_{[0,1]^d} w_d^2 e^{-u_d w_d^2}\, dw_d\right], \end{aligned}$$

so we have $G(u) \leq \underline{0}$ for any $u \in U$ and thus the exact minimizer is $u^* = (1, \ldots, 1)$ for this problem.

The reference objective function value can be computed by

$$F(u) = \int_{[0,1]^d} \sum_{i=1}^{d} e^{-u_i w_i^2}\, dw = \sum_{i=1}^{d} \int_{[0,1]} e^{-u_i w_i^2}\, dw_i.$$

At the minimizer, we have the exact objective

$$F(u^*) = d \int_{[0,1]} e^{-w_i^2}\, dw_i.$$

which can be computed by using the cumulative distribution function of a transformed normal distribution.

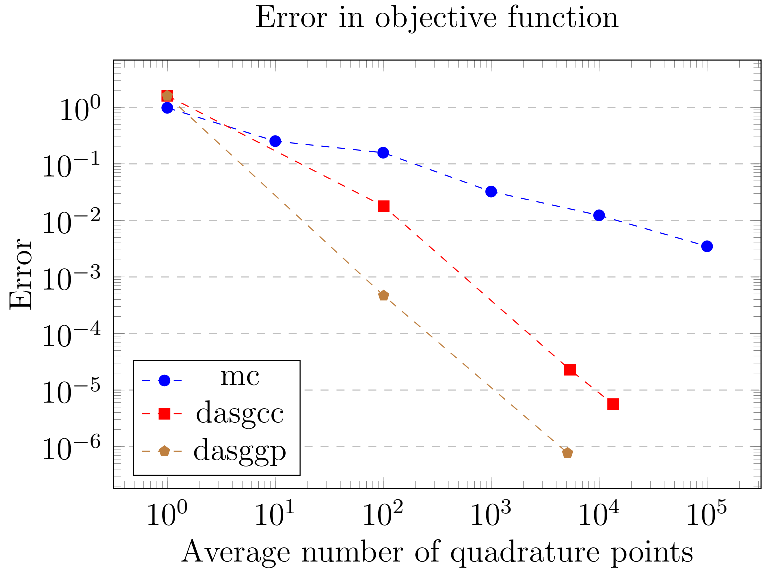


Figure 4.6: Compute the additive example with $d = 50$. Three different quadrature methods are used to approximate the objective function. For the Clenshaw-Curtis(dasgcc) and the Gauss-Patterson (dasggp) approaches, we average the number of quadrature points used in computing the objective functions in all iterations.

Here we apply the ’optimize then discretise’ approach to solve the problem. We use ’L-BFGS-B’ method in scipy.optimize package as our solver. We apply the dimension adaptive sparse grid method to computing the objective function while use the Monte-Carlo method to approximate the high dimensional integrals which appear in the gradient.

We have the approximated gradient $\bar{G}_p(u) \leq 0$ for any $u \in [0,1]^d$. This is because the integrand of each entry of the gradient is non-positive and only positive weights are used in the Monte Carlo method. Thus, no matter how many samples are used in the Monte Carlo method, we will always get a descent direction at each step during optimization process. It is noteworthy here that both low level sparse grid method and dimensional-adaptive sparse grid method with large ϵ and small γ may change the sign of the integral approximated and therefore lead to wrong search directions.

In the example, we only use the Monte Carlo method with 10 samples to compute the gradient components. In order to increase the accuracy in minimum, when we approximate the objective function, we increase samples in the Monte Carlo method. For the same reasons, we decrease ϵ and keep a fixed large γ in the termination condition of the dimension-adaptive sparse grid. The result in the Figure 4.6 shows all of the three methods achieve the exact minimizer as we expected. The errors of the minimums of two dimension adaptive approaches drop much faster than the Monte Carlo method.

4.7.2 Application to stochastic control

In this section, we illustrate our dimension-adaptive sparse grid method with an instance of a discrete time open-loop stochastic control problem. The general form of such a control problem can be found in [9, 8]. The control problem is described by the following discrete time dynamic system

$$x_{i+1} = \psi_i(x_i, u_i, w_i),\ i = 0, \ldots, d-1 \tag{4.24}$$

Here x_i and u_i are states and controls respectively where the initial state x_0 is given. w_i are disturbances. Here we only consider a special case when the states, the controls and the disturbances are in one dimensional space. When the disturbances in the system are unknown, we usually model them as i.i.d. random variables W_i with given probability density function. In this case, the open-loop means the controls u_i do not depend on the disturbances [69] and we can further write the dynamic system in its random form:

$$X_{i+1} = \psi_i(X_i, u_i, W_i),\ i = 0, \ldots, d-1. \tag{4.25}$$

If we further define the vectors of states,controls and noises, i.e.,

$$X = (x_0, \ldots, X_{d-1}),\ u = (u_0, \ldots, u_{d-1}),\ W = (W_0, \ldots, W_{d-1}),$$

then we can rewrite the dynamic system as

$$X = \Psi(X, u, W). \tag{4.26}$$

where Ψ is a function can be derived from ψ_i.

Our task now is to determine what is the 'best' control for the dynamic system (4.25) or (4.26) to minimize the expected cost

$$\mathbb{E}\left[\Phi(u, X)\right]$$

where Φ is a given function.

Here we focus on the case when X can be solved explicitly from the dynamic system (4.26), that is,

$$X = \xi(u, W).$$

In this case, the original problem can be reduced into the standard form of the stochastic optimization problem, namely,

$$\min_{u \in U} \mathbb{E}\left[h(u, W)\right],$$

where

$$h(u, W) = \Phi(u, \xi(u, W)).$$

The integral form of the expected cost and its surrogate with N quadrature points are

$$\int_{\mathbb{R}^d} \Phi(u, \xi(u, \underline{w}))p(\underline{w})\ d\underline{w} \approx \sum_{j=1}^{N} c_i \Phi(u, \xi(u, \underline{w}_j))p(\underline{w}_j).$$

In order to illustrate the computational performance of our approach, we consider a classical example with linear dynamic system

$$X = AX + Bu + CW + x_0\underline{e_0}. \tag{4.27}$$

and the quadratic objective function Φ

$$\Phi(u, \underline{x}) = u^T P u + \underline{x}^T Q \underline{x} \tag{4.28}$$

where A, B, C, P and Q are given $d \times d$ matrices and x_0 is the given initial value. By solving (4.27), we get $\xi(u, W) = (I - A)^{-1}(Bu + CW + x_0\underline{e_0})$. Combining the expression of $\xi(u, W)$ with (4.28), we know that $h(u, W)$ is again a quadratic function.

The exact solution can be derived by using the certainty equivalence principle [9]. According to the principle, the solution of the stochastic control problem is the same as that of a corresponding deterministic problem when the objective function is quadratic and the constraints are linear. That means we can get the reference solution by numerically solving the deterministic problem. Here is a concrete example.

Consider minimizing the following quadratic cost functional over u

$$\mathbb{E}\left[\left(\sum_{i=0}^{d-1}(p_i x_i^2 + q_i u_i^2)\right) + p_d x_d^2\right]$$

where the expectation is with respect to the random variables W_i and

$$x_{i+1} = a_i x_i + b_i u_i + c_i W_i, \quad i = 0, 1, \ldots, d-1.$$

x_i are state variables and u_i are control variables. x_0 refers to the initial state which is given. The disturbances W_i are independent continuous random variables with given probability density function $p(W_i)$. This is a linear-quadratic open loop control problem[69]. Similar to Feedback control, open loop control also has a wide variety of applications.

We can rewrite the problem into matrix form

$$\min_u \mathbb{E}\left[x^T P x + u^T Q u\right]$$

with the linear system

$$x = Ax + Bu + CW + x_0.$$

where P is a $d \times d$ diagonal matrix with $(p_0, p_1, \ldots, p_{d-1})$ on its diagonal and $p_d = 0$ (boundary condition). Q is a $d \times d$ diagonal matrix with $(q_0, q_1, \ldots, q_{d-1})$ on its diagonal. A, B and C are $d \times d$ matrices that only have nonzero entries along their lower sub-diagonal.

$$A = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ a_0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & a_1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & a_{d-2} & 0 \end{bmatrix}, \quad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{d-2} \\ x_{d-1} \end{bmatrix}$$

B is a $d \times d$ diagonal matrix

$$B = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ b_0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & b_1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & b_{d-2} & 0 \end{bmatrix}, \quad u = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{d-2} \\ u_{d-1} \end{bmatrix}$$

C is a $d \times d$ diagonal matrix

$$C = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ c_0 & 0 & 0 & \dots & 0 & 0 \\ 0 & c_1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & c_{d-2} & 0 \end{bmatrix}, \quad W = \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ \vdots \\ W_{d-2} \\ W_{d-1} \end{bmatrix}$$

and x_0 is a $d \times 1$ vector

$$x_0 = \begin{bmatrix} x_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

By using the above matrix and vector notation, we can solve the linear system

$$x = (I - A)^{-1}(Bu + CW + x_0).$$

explicitly. Substituting the solution (9) into objective function, we get

$$\mathbb{E}\left[\left[(I - A)^{-1}(Bu + CW + x_0)\right]^T P \left[(I - A)^{-1}(Bu + CW + x_0)\right] + u^T Q u\right].$$

Next, we consider the numerical example. P and Q matrices are $d \times d$ identity matrices and matrices A, B and C are

$$A = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1+\Delta t & 0 & 0 & \dots & 0 & 0 \\ 0 & 1+\Delta t & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1+\Delta t & 0 \end{bmatrix},$$

$$B = C = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \Delta t & 0 & 0 & \dots & 0 & 0 \\ 0 & \Delta t & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & \Delta t & 0 \end{bmatrix},$$

where $\Delta t = \frac{1}{d}$. The disturbances $W_i,\ i = 0, 1, \ldots, d-1$ are i.i.d random variables subject to beta distribution $beta(2, 3)$. If we denote

$$K = (I - A)^{-1},$$

then the objective function can be further written as

$$\mathbb{E}\left[(\Delta tu + \Delta tW + \underline{x_0})^T K^T K(\Delta tu + \Delta tW + \underline{x_0}) + u^T u\right].$$

Using the notation defined in (4.1)

$$\begin{aligned} h(u, W) = {} & u^T[(\Delta t)^2 K^T K + I]u + 2\Delta tu^T K^T K\underline{x_0} \\ & + 2(\Delta t)^2 u^T K^T KW + 2(\Delta t)W^T K^T K\underline{x_0} \\ & + (\Delta t)^2 W^T K^T KW + \underline{x_0}^T K^T Kx_0 \end{aligned}$$

By using the linearity of the expectation, we have

$$\mathbb{E}\left[2(\Delta t)^2 u^T K^T KW\right] = 2(\Delta t)^2 u^T K^T K\,\mathbb{E}[W],$$

and

$$\mathbb{E}\left[2(\Delta t)W^T K^T K\underline{x_0}\right] = 2(\Delta t)(\mathbb{E}[W])^T K^T K\underline{x_0}.$$

Then our expected objective function can be written as

$$\begin{aligned} \mathbb{E}[h(u, W)] = {} & u^T[(\Delta t)^2 K^T K + I]u + 2u^T K^T K(\Delta t\underline{x_0} + \Delta t^2\,\mathbb{E}[W]) \\ & + \underline{x_0}^T K^T Kx_0 + 2(\Delta t)(\mathbb{E}[W])^T K^T K\underline{x_0} \\ & + \mathbb{E}\left[(\Delta t)^2 W^T K^T KW\right]. \end{aligned}$$

Since

$$u^T K^T Ku \geq 0,$$

$K^T K$ is a positive semidefinite matrix. Therefore, $(\Delta t)^2 K^T K + I$ is positive definite and the expected objective function is strictly convex. Since this function is also differentiable, the optimality condition for this problem is the first order derivative equals to zero.

$$2[(\Delta t)^2 K^T K + I]u + 2K^T K(\Delta t\underline{x_0} + \Delta t^2\,\mathbb{E}[W]) = 0.$$

Thus, the optimal solution of this example is

$$u = -[(\Delta t)^2 K^T K + I]^{-1} K^T K(\Delta t\underline{x_0} + \Delta t^2\,\mathbb{E}[W]).$$

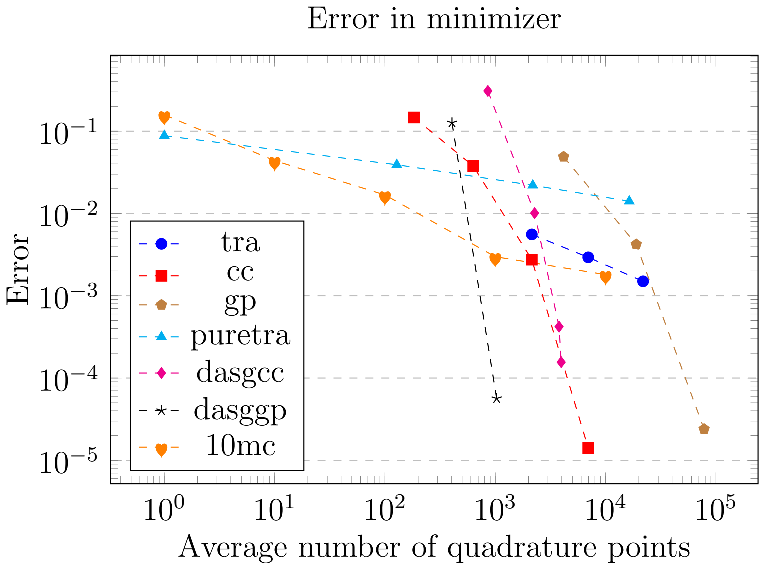


Figure 4.7: We test nine methods for the open loop control problem. They are classical sparse grid method generated by 1D trapezoidal rule(tra), Clenshaw-Curtis rule(cc) and Gauss-Patterson rule(gp), the product rule generated 1D trapezoidal rule(puretra), dimension adaptive sparse grid method generated by 1D Clenshaw-Curtis rule(dasgcc), Gauss-Patterson rule(dasggp) and the average of 10 runs of Monte Carlo method. For two dimension adaptive sparse grid approaches, we average the number of quadrature points which are used in computing the objective functions in all iterations.

Next, we consider following deterministic control problem,

$$\min_u (x^T P x + u^T Q u)$$

with the linear system

$$x = Ax + Bu + C\,\mathbb{E}\,[W] + x_0.$$

If we set the same parameters as that in the stochastic example and turn it into an unconstrained minimization problem, then we have

$$\min_u J(u),$$

where

$$J(u) = (\Delta t u + \Delta t\,\mathbb{E}\,[W] + \underline{x_0})^T K^T K(\Delta t u + \Delta t\,\mathbb{E}\,[W] + \underline{x_0}) + u^T u.$$

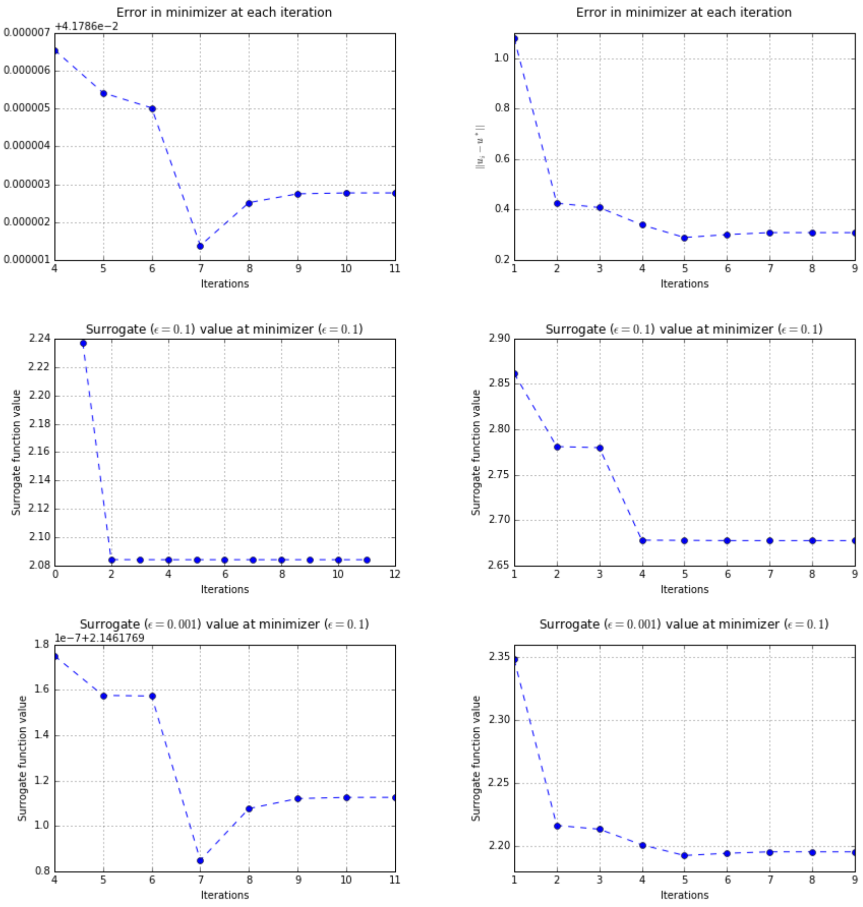
The optimality condition for this deterministic control problem is also

$$2[(\Delta t)^2 K^T K + I]u + 2K^T K(\Delta t \underline{x_0} + \Delta t^2\,\mathbb{E}\,[W]) = 0.$$

which means the above deterministic problem has the same minimizer as our stochastic problem. This is usually called certainty equivalence principle [9] in engineering. Here we actually provide a proof of the simple case of this principle. One can actually prove this holds for general quadratic objective function with linear constraints.

We test a 7 dimensional problem. We use an asymmetric distribution beta$(2, 3)$ here. We will get the exact solution with only rounding errors if we use a symmetric distribution. This is because the symmetric construction of the sparse grid will lead to the cancellation of the quadrature points pairs. We still use BFGS method as our optimisation solver. The computational results are shown in the Figure 4.7. We compare the errors of 9 different methods. They are product trapezoidal rule, the average of 10 runs Monte Carlo, three sparse grid method and three dimension adaptive sparse grid. We only record the data when sparse grid method and dimension adaptive sparse grid method start to converge. As can be seen from the Figure 4.7, the dimension adaptive sparse grid methods converge faster than classical sparse grid methods for both Clenshaw Curtis and Gauss Patterson approaches. The results of sparse gird methods are much better than the trapezoidal product rule and the Monte Carlo method.

In Figure 4.8, we test our stopping criterion for the stochastic control problem with quadratic cost function and linear dynamic system. Here we focus on the dimension adaptive sparse grid method generated by 1D Clenshaw Curtis rule. For both 6D and 7D examples, our method successfully predicts that we should stop at 7th iteration for 6D problem and 5th iteration for 7D problem.



(a) Clenshaw Curtis with $d = 6$ (b) Clenshaw Curtis with $d = 7$

Figure 4.8: Solve the open loop control problem.

4.8 Conclusions

We apply the Newton-type methods in solving the stochastic optimisation problem and the dimension adaptive sparse grid quadrature is used in approximating the integrals involved. We study two categories of approaches to solve the stochastic optimisation problem. They are 'discretise then optimise' method(DTOM) and the 'optimise then discretise' method(OTDM). We show that the OTDM is actually a kind of generalisation of the DTOM. In fact we can use more flexible discretisation scheme during the computation if we apply the OTDM. The dimension adaptive sparse grid quadrature can effectively reduce the computational cost when we use it to compute an integral of which the dimensions are not equally important. When we applied it to solve the stochastic optimisation problem, we find it is more suitable to be used in the OTDM compared with the DTOM. This is because the OTDM allows us to choose 'best' downset in the dimension adaptive sparse grid formula at each iteration and thus fully exploit the potential of the dimension-adaptive approach. The convergence of the OTDM can be make sure under the assumption (4.23). A good stopping criterion is crucial for reducing the computational cost when we solve high dimensional stochastic optimisation problems. We provide an accurate stopping criterion which only requires reasonable additional computation. We focus on the convex objective function here. For non-convex problems, our approach can only find an approximated local minimizer since Newton-type methods are locally convergent methods. In order to solve more general stochastic optimisation problems, other solvers will be taken into consideration in the future research. Also, we are currently unclear about how to choose a good decreasing sequence ϵ_p, $p \in \mathbb{N}$ and increasing sequence γ_p, $p \in \mathbb{N}$ for a practical problem. In most cases, when our stopping criterion predicts no need for more iteration, we can only choose a smaller ϵ and larger γ manually. We will consider how to generate the sequence ϵ_p, $p \in \mathbb{N}$ and γ_p, $p \in \mathbb{N}$ automatically based on the data from the previous iterations in future.

Bibliography

[1] Stefan Achatz and Christoph Zenger. "Higher order sparse grid methods for elliptic partial differential equations". In: *Curve and surface fitting (Saint-Malo, 2002).* Mod. Methods Math. Nashboro Press, Brentwood, TN, 2003, pp. 1–10.

[2] Antonio Ambrosetti and Giovanni Prodi. *A primer of nonlinear analysis.* Vol. 34. Cambridge Studies in Advanced Mathematics. Corrected reprint of the 1993 original. Cambridge University Press, Cambridge, 1995, pp. viii+171. ISBN: 0-521-48573-8.

[3] Akio Arakawa. "Computational design for long-term numerical integration of the equations of fluid motion: two-dimensional incompressible flow. I [J. Comput. Phys. **1** (1966), no. 1, 119–143]". In: vol. 135. 2. With an introduction by Douglas K. Lilly, Commemoration of the 30th anniversary {of J. Comput. Phys.}. 1997, pp. 101–114. DOI: `10.1006/jcph.1997.5697`. URL: `https://doi.org/10.1006/jcph.1997.5697`.

[4] Erlend Arge, Morten Dæhlen, and Aslak Tveito. "Approximation of scattered data using smooth grid functions". In: *J. Comput. Appl. Math.* 59.2 (1995), pp. 191–205. ISSN: 0377-0427. DOI: `10.1016/0377-0427(94)00033-W`. URL: `https://doi.org/10.1016/0377-0427(94)00033-W`.

[5] Robert Balder and Christoph Zenger. "The solution of multidimensional real Helmholtz equations on sparse grids". In: *SIAM J. Sci. Comput.* 17.3 (1996), pp. 631–646. ISSN: 1064-8275. DOI: `10.1137/S1064827593247035`. URL: `https://doi.org/10.1137/S1064827593247035`.

[6] Volker Barthelmann, Erich Novak, and Klaus Ritter. "High dimensional polynomial interpolation on sparse grids". In: vol. 12. 4. Multivariate polynomial interpolation. 2000, pp. 273–288. DOI: `10.1023/A:1018977404843`. URL: `https://doi.org/10.1023/A:1018977404843`.

[7] Jordan Bell. *Fréchet Derivatives and Gâteaux Derivatives.* https://individual.utoronto.ca/jordanbell/notes/frechetderivatives.pdf. Apr. 2014.

[8] Richard Bellman. *Dynamic programming.* Princeton Landmarks in Mathematics. Reprint of the 1957 edition, With a new introduction by Stuart Dreyfus. Princeton University Press, Princeton, NJ, 2010, pp. xxx+340. ISBN: 978-0-691-14668-3.

[9] Dimitri P. Bertsekas. *Dynamic programming and optimal control. Vol. I.* Third. Athena Scientific, Belmont, MA, 2005, pp. xvi+543. ISBN: 1-886529-26-4.

[10] John R. Birge and François Louveaux. *Introduction to stochastic programming.* Second. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2011, pp. xxvi+485. ISBN: 978-1-4614-0236-7. DOI: 10.1007/978-1-4614-0237-4. URL: https://doi.org/10.1007/978-1-4614-0237-4.

[11] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. "Optimization methods for large-scale machine learning". In: *SIAM Rev.* 60.2 (2018), pp. 223–311. ISSN: 0036-1445. DOI: 10.1137/16M1080173. URL: https://doi.org/10.1137/16M1080173.

[12] Dietrich Braess. *Finite elements.* Second. Theory, fast solvers, and applications in solid mechanics, Translated from the 1992 German edition by Larry L. Schumaker. Cambridge University Press, Cambridge, 2001, pp. xviii+352. ISBN: 0-521-01195-7.

[13] Alain Brizard and Taik Soo Hahm. "Foundations of nonlinear gyrokinetic theory". In: *Reviews of modern physics* 79.2 (2007), p. 421.

[14] H.-J. Bungartz and S. Dirnstorfer. "Multivariate quadrature on adaptive sparse grids". In: *Computing* 71.1 (2003), pp. 89–114. ISSN: 0010-485X. DOI: 10.1007/s00607-003-0016-4. URL: https://doi.org/10.1007/s00607-003-0016-4.

[15] Hans-Joachim Bungartz. "An adaptive Poisson solver using hierarchical bases and sparse grids". In: *Iterative methods in linear algebra (Brussels, 1991).* North-Holland, Amsterdam, 1992, pp. 293–310.

[16] Hans-Joachim Bungartz and Michael Griebel. "Sparse grids". In: *Acta Numer.* 13 (2004), pp. 147–269. ISSN: 0962-4929. DOI: 10.1017/S0962492904000182. URL: https://doi.org/10.1017/S0962492904000182.

[17] Hans-Joachim Bungartz, Michael Griebel, and Ulrich Rüde. "Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems". In: vol. 116. 1-4. ICOSAHOM'92 (Montpellier, 1992). 1994, pp. 243–252. DOI: 10.1016/S0045-7825(94)80029-4. URL: https://doi.org/10.1016/S0045-7825(94)80029-4.

[18] Hans-Joachim Bungartz et al. "Pointwise convergence of the combination technique for the Laplace equation". In: *East-West J. Numer. Math.* 2.1 (1994), pp. 21–45. ISSN: 0928-0200.

[19] Hans-Joachim Bungartz et al. "Two proofs of convergence for the combination technique for the efficient solution of sparse grid problems". In: *Domain decomposition methods in scientific and engineering computing (University Park, PA, 1993).* Vol. 180. Contemp. Math. Amer. Math. Soc., Providence, RI, 1994, pp. 15–20. DOI: 10.1090/conm/180/01952. URL: https://doi.org/10.1090/conm/180/01952.

[20] Michael Chen and Sanjay Mehrotra. *Epi-convergent scenario generation method for stochastic problems via sparse grid.* 2008.

[21] Ward Cheney. *Introduction to approximation theory.* Reprint of the second (1982) edition. AMS Chelsea Publishing, Providence, RI, 1998, pp. xii+259. ISBN: 0-8218-1374-9.

[22] Ward Cheney and Will Light. *A course in approximation theory.* Vol. 101. Graduate Studies in Mathematics. Reprint of the 2000 original. American Mathematical Society, Providence, RI, 2009, pp. xvi+359. ISBN: 978-0-8218-4798-5. DOI: 10.1090/gsm/101. URL: https://doi.org/10.1090/gsm/101.

[23] Charles W. Clenshaw and Alan R. Curtis. "A method for numerical integration on an automatic computer". In: *Numer. Math.* 2 (1960), pp. 197–205. ISSN: 0029-599X. DOI: 10.1007/BF01386223. URL: https://doi.org/10.1007/BF01386223.

[24] Brian Davey and Hilary Priestley. *Introduction to lattices and order.* Second. Cambridge University Press, New York, 2002, pp. xii+298. ISBN: 0-521-78451-4. DOI: 10.1017/CBO9780511809088. URL: https://doi.org/10.1017/CBO9780511809088.

[25] Philip J. Davis and Philip Rabinowitz. *Methods of numerical integration.* Second. Computer Science and Applied Mathematics. Academic Press Inc., Orlando, FL, 1984, pp. xiv+612. ISBN: 0-12-206360-0.

[26] Alexandre Defossez et al. *A Simple Convergence Proof of Adam and Adagrad.* 2020. arXiv: 2003.02395 [stat.ML].

[27] Franz-J. Delvos. "Boolean methods for double integration". In: *Math. Comp.* 55.192 (1990), pp. 683–692. ISSN: 0025-5718. DOI: 10.2307/2008441. URL: https://doi.org/10.2307/2008441.

[28] Peter Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms.* Vol. 35. Springer Science & Business Media, 2011.

[29] Ronald A. DeVore and George G. Lorentz. *Constructive approximation.* Vol. 303. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer-Verlag, Berlin, 1993, pp. x+449. ISBN: 3-540-50627-6. DOI: 10.1007/978-3-662-02888-9. URL: https://doi.org/10.1007/978-3-662-02888-9.

[30] Josef Dick, Frances Y Kuo, and Ian H Sloan. "High-dimensional integration: the quasi-Monte Carlo way". In: *Acta Numerica* 22 (2013), p. 133.

[31] T. Dornseifer and C. Pflaum. "Discretization of elliptic differential equations on curvilinear bounded domains with sparse grids". In: vol. 56. 3. International GAMM-Workshop on Multi-level Methods (Meisdorf, 1994). 1996, pp. 197–213. DOI: 10.1007/BF02238512. URL: https://doi.org/10.1007/BF02238512.

[32] Hartmut Ehlich and Karl Zeller. "Auswertung der Normen von Interpolationsoperatoren". In: *Math. Ann.* 164 (1966), pp. 105–112. ISSN: 0025-5831. DOI: 10.1007/BF01429047. URL: https://doi.org/10.1007/BF01429047.

[33] Sven Ehrich and Giuseppe Mastroianni. "On generalized Stieltjes polynomials and Lagrange interpolation". In: *Approximation theory and function series (Budapest, 1995).* Vol. 5. Bolyai Soc. Math. Stud. János Bolyai Math. Soc., Budapest, 1996, pp. 187–203.

[34] Sven Ehrich and Giuseppe Mastroianni. "Stieltjes polynomials and Lagrange interpolation". In: *Math. Comp.* 66.217 (1997), pp. 311–331. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-97-00808-9. URL: https://doi.org/10.1090/S0025-5718-97-00808-9.

[35] J. Garcke, M. Griebel, and M. Thess. "Data mining with sparse grids". In: *Computing* 67.3 (2001), pp. 225–253. ISSN: 0010-485X. DOI: 10.1007/s006070170007. URL: https://doi.org/10.1007/s006070170007.

[36] Jochen Garcke. “Sparse grids in a nutshell”. In: *Sparse grids and applications.* Vol. 88. Lect. Notes Comput. Sci. Eng. Springer, Heidelberg, 2013, pp. 57–80. DOI: 10.1007/978-3-642-31703-3. URL: https://doi.org/10.1007/978-3-642-31703-3.

[37] Jochen Garcke, Markus Hegland, and Ole Nielsen. “Parallelisation of sparse grids for large scale data analysis”. In: *ANZIAM J.* 48.1 (2006), pp. 11–22. ISSN: 1446-1811. DOI: 10.1017/S1446181100003382. URL: https://doi.org/10.1017/S1446181100003382.

[38] Thomas Gerstner and Michael Griebel. “Dimension–adaptive tensor–product quadrature”. In: *Computing* 71.1 (2003), pp. 65–87.

[39] Thomas Gerstner and Michael Griebel. “Numerical integration using sparse grids”. In: *Numer. Algorithms* 18.3-4 (1998), pp. 209–232. ISSN: 1017-1398. DOI: 10.1023/A:1019129717644. URL: https://doi.org/10.1023/A:1019129717644.

[40] Thomas Gerstner and Michael Griebel. “Numerical integration using sparse grids”. In: *Numer. Algorithms* 18.3-4 (1998), pp. 209–232. ISSN: 1017-1398. DOI: 10.1023/A:1019129717644. URL: https://doi.org/10.1023/A:1019129717644.

[41] Ivan G Graham et al. “Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications”. In: *Journal of Computational Physics* 230.10 (2011), pp. 3668–3694.

[42] M. Griebel and T. Schiekofer. “An adaptive sparse grid Navier Stokes solver in 3D based on the finite difference method”. In: *ENUMATH 97 (Heidelberg).* World Sci. Publ., River Edge, NJ, 1998, pp. 327–334.

[43] Michael Griebel, Michael Schneider, and Christoph Zenger. “A combination technique for the solution of sparse grid problems”. In: *Iterative methods in linear algebra (Brussels, 1991).* North-Holland, Amsterdam, 1992, pp. 263–281.

[44] Michael Griebel and Veronika Thurner. “The efficient solution of fluid dynamics problems by the combination technique”. In: *Internat. J. Numer. Methods Heat Fluid Flow* 5.3 (1995), pp. 251–269. ISSN: 0961-5539. DOI: 10.1108/EUM0000000004119. URL: https://doi.org/10.1108/EUM0000000004119.

[45] Brendan Harding. “Fault Tolerant Computation of Hyperbolic Partial Differential Equations with the Sparse Grid Combination Technique”. PhD thesis. The Australian National University, 2016.

[46] Markus Hegland. “Adaptive sparse grids”. In: *Anziam Journal* 44 (2002), pp. 335–353.

[47] Markus Hegland, Jochen Garcke, and Vivien Challis. “The combination technique and some generalisations”. In: *Linear Algebra Appl.* 420.2-3 (2007), pp. 249–275. ISSN: 0024-3795. DOI: 10.1016/j.laa.2006.07.014. URL: https://doi.org/10.1016/j.laa.2006.07.014.

[48] Markus Hegland et al. “Recent developments in the theory and application of the sparse grid combination technique”. In: *Software for exascale computing—SPPEXA 2013–2015*. Vol. 113. Lect. Notes Comput. Sci. Eng. Springer, [Cham], 2016, pp. 143–163.

[49] Markus Holtz. “Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance”. PhD thesis. university of Bonn, 2008.

[50] Rie Johnson and Tong Zhang. “Accelerating stochastic gradient descent using predictive variance reduction”. In: *Advances in neural information processing systems* 26 (2013), pp. 315–323.

[51] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo methods*. Second. Wiley-Blackwell, Weinheim, 2008, pp. xii+203. ISBN: 978-3-527-40760-6. DOI: 10.1002/9783527626212. URL: https://doi.org/10.1002/9783527626212.

[52] Carl Kelley. *Iterative methods for linear and nonlinear equations*. Vol. 16. Frontiers in Applied Mathematics. With separately available software. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995, pp. iv+165. ISBN: 0-89871-352-8. DOI: 10.1137/1.9781611970944. URL: https://doi.org/10.1137/1.9781611970944.

[53] Carl Kelley. *Solving nonlinear equations with Newton’s method*. Vol. 1. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003, pp. xiv+104. ISBN: 0-89871-546-6. DOI: 10.1137/1.9780898718898. URL: https://doi.org/10.1137/1.9780898718898.

[54] Carl Kelley. *Solving nonlinear equations with Newton’s method*. Vol. 1. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003, pp. xiv+104. ISBN: 0-89871-546-6. DOI: 10.1137/1.9780898718898. URL: https://doi.org/10.1137/1.9780898718898.

[55] Aleksandr Semenovich Kronrod. *Nodes and weights of quadrature formulas. Sixteen-place tables.* Authorized translation from the Russian. Consultants Bureau, New York, 1965, pp. vii+143.

[56] Frances Y Kuo, Christoph Schwab, and Ian H Sloan. "Multi-level quasi-Monte Carlo finite element methods for a class of elliptic PDEs with random coefficients". In: *Foundations of Computational Mathematics* 15.2 (2015), pp. 411–449.

[57] Frances Y Kuo, Christoph Schwab, and Ian H Sloan. "Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients". In: *SIAM Journal on Numerical Analysis* 50.6 (2012), pp. 3351–3374.

[58] Randall J. LeVeque. *Finite difference methods for ordinary and partial differential equations.* Steady-state and time-dependent problems. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007, pp. xvi+341. ISBN: 978-0-898716-29-0. DOI: `10.1137/1.9780898717839`. URL: `https://doi.org/10.1137/1.9780898717839`.

[59] C. B. Liem, T. Lü, and T. M. Shih. *The splitting extrapolation method.* Vol. 7. Series on Applied Mathematics. A new technique in numerical solution of multidimensional problems, With a preface by Zhong-ci Shi. World Scientific Publishing Co., Inc., River Edge, NJ, 1995, pp. xx+316. ISBN: 981-02-2217-3. DOI: `10.1142/2708`. URL: `https://doi.org/10.1142/2708`.

[60] John Mason and David Handscomb. *Chebyshev polynomials.* Chapman & Hall/CRC, Boca Raton, FL, 2003, pp. xiv+341. ISBN: 0-8493-0355-9.

[61] Nicholas Metropolis and Stanislaw Ulam. "The monte carlo method". In: *Journal of the American statistical association* 44.247 (1949), pp. 335–341.

[62] Giovanni Monegato. "An overview of results and questions related to Kronrod schemes". In: *Numerische Integration (Tagung, Math. Forschungsinst., Oberwolfach, 1978).* Vol. 45. Internat. Ser. Numer. Math. Birkhäuser, Basel-Boston, Mass., 1979, pp. 231–240.

[63] Giovanni Monegato. "Stieltjes polynomials and related quadrature rules". In: *SIAM Rev.* 24.2 (1982), pp. 137–158. ISSN: 0036-1445. DOI: `10.1137/1024039`. URL: `https://doi.org/10.1137/1024039`.

[64] William J. Morokoff and Russel E. Caflisch. "Quasi-Monte Carlo integration". In: *J. Comput. Phys.* 122.2 (1995), pp. 218–230. ISSN: 0021-9991. DOI: `10.1006/jcph.1995.1209`. URL: `https://doi.org/10.1006/jcph.1995.1209`.

[65] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[66] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006, pp. xxii+664. ISBN: 978-0387-30303-1; 0-387-30303-0.

[67] Sotirios E Notaris. "Gauss-Kronrod quadrature formulae-A survey of fifty years of research". In: *Electron. Trans. Numer. Anal* 45 (2016), pp. 371–404.

[68] Erich Novak and Klaus Ritter. "High-dimensional integration of smooth functions over cubes". In: *Numer. Math.* 75.1 (1996), pp. 79–97. ISSN: 0029-599X. DOI: `10.1007/s002110050231`. URL: `https://doi.org/10.1007/s002110050231`.

[69] Bernt Øksendal. *Stochastic differential equations*. Fifth. Universitext. An introduction with applications. Springer-Verlag, Berlin, 1998, pp. xx+324. ISBN: 3-540-63720-6. DOI: `10.1007/978-3-662-03620-4`. URL: `https://doi.org/10.1007/978-3-662-03620-4`.

[70] Thomas N.L. Patterson. "An algorithm for generating interpolatory quadrature rules of the highest degree of precision with preassigned nodes for general weight functions". In: *ACM Transactions on Mathematical Software (TOMS)* 15.2 (1989), pp. 123–136.

[71] Thomas N.L. Patterson. "Modified optimal quadrature extensions". In: *Numer. Math.* 64.4 (1993), pp. 511–520. ISSN: 0029-599X. DOI: `10.1007/BF01388702`. URL: `https://doi.org/10.1007/BF01388702`.

[72] Thomas N.L. Patterson. "The optimum addition of points to quadrature formulae". In: *Math. Comp. 22 (1968), 847–856; addendum, ibid.* 22.104, loose microfiche supp. (1968), pp. C1–C11. ISSN: 0025-5718. DOI: `10.2307/2004583`. URL: `https://doi.org/10.2307/2004583`.

[73] Christoph Pflaum. "Convergence of the combination technique for second-order elliptic differential equations". In: *SIAM J. Numer. Anal.* 34.6 (1997), pp. 2431–2455. ISSN: 0036-1429. DOI: `10.1137/S0036142993260294`. URL: `https://doi.org/10.1137/S0036142993260294`.

[74] Christoph Pflaum and Aihui Zhou. "Error analysis of the combination technique". In: *Numer. Math.* 84.2 (1999), pp. 327–350. ISSN: 0029-599X. DOI: 10.1007/s002110050474. URL: https://doi.org/10.1007/s002110050474.

[75] Dirk Pflüger, Benjamin Peherstorfer, and Hans-Joachim Bungartz. "Spatially adaptive sparse grids for high-dimensional data-driven problems". In: *J. Complexity* 26.5 (2010), pp. 508–522. ISSN: 0885-064X. DOI: 10.1016/j.jco.2010.04.001. URL: https://doi.org/10.1016/j.jco.2010.04.001.

[76] Christoph Reisinger. "Analysis of linear difference schemes in the sparse grid combination technique". In: *IMA J. Numer. Anal.* 33.2 (2013), pp. 544–581. ISSN: 0272-4979. DOI: 10.1093/imanum/drs004. URL: https://doi.org/10.1093/imanum/drs004.

[77] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method.* Vol. 10. John Wiley & Sons, 2016.

[78] Walter Rudin. *Principles of mathematical analysis.* Third. International Series in Pure and Applied Mathematics. McGraw-Hill Book Co., New York-Auckland-Düsseldorf, 1976, pp. x+342.

[79] Christoph Schwab and Claude Jeffrey Gittelson. "Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs". In: *Acta Numerica* 20 (2011), pp. 291–467.

[80] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory.* SIAM, 2014.

[81] Ian H Sloan and Henryk Woźniakowski. "When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?" In: *Journal of Complexity* 14.1 (1998), pp. 1–33.

[82] Ian H. Sloan. "QMC integration—beating intractability by weighting the coordinate directions". In: *Monte Carlo and quasi-Monte Carlo methods, 2000 (Hong Kong).* Springer, Berlin, 2002, pp. 103–123.

[83] Ian H. Sloan and W. E. Smith. "Product-integration with the Clenshaw-Curtis and related points. Convergence properties". In: *Numer. Math.* 30.4 (1978), pp. 415–428. ISSN: 0029-599X. DOI: 10.1007/BF01398509. URL: https://doi.org/10.1007/BF01398509.

[84] Ian H. Sloan and Henryk Woźniakowski. "When are quasi-Monte Carlo algorithms efficient for high-dimensional integrals?" In: *J. Complexity* 14.1 (1998), pp. 1–33. ISSN: 0885-064X. DOI: `10.1006/jcom.1997.0463`. URL: `https://doi.org/10.1006/jcom.1997.0463`.

[85] Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*. Vol. 12. Siam, 2013.

[86] Sergei Abramovich Smolyak. "Quadrature and interpolation formulas for tensor products of certain classes of functions". In: *Doklady Akademii Nauk*. Vol. 148. 5. Russian Academy of Sciences. 1963, pp. 1042–1045.

[87] Elias M. Stein and Rami Shakarchi. *Real analysis*. Vol. 3. Princeton Lectures in Analysis. Measure theory, integration, and Hilbert spaces. Princeton University Press, Princeton, NJ, 2005, pp. xx+402. ISBN: 0-691-11386-6.

[88] Gene Development Team. *The Gyrokinetic Plasma Turbulence Code Gene: User Manual*. 2013.

[89] Yuchen Xie, Richard H Byrd, and Jorge Nocedal. "Analysis of the BFGS Method with Errors". In: *SIAM Journal on Optimization* 30.1 (2020), pp. 182–209.

[90] Tjalling J. Ypma. "The effect of rounding errors on Newton-like methods". In: *IMA Journal of Numerical Analysis* 3.1 (1983), pp. 109–118.

[91] Christoph Zenger. "Sparse grids". In: *Parallel algorithms for partial differential equations (Kiel, 1990)*. Vol. 31. Notes Numer. Fluid Mech. Friedr. Vieweg, Braunschweig, 1991, pp. 241–251.

[92] Yuancheng Zhou and Markus Hegland. "The application of sparse grid quadrature in solving stochastic optimisation problems". In: *ANZIAM Journal* 60 (2018), pp. 16–32.

[93] Yuancheng Zhou and Markus Hegland. *The Combination Technique applied to Functionals*. ANZIAM Journal(accepted).