The Multivariate Decomposition Method

Dirk Nuyens
NUMA, KU Leuven, Belgium

On joint work with
Alec Gilbert, Frances Kuo, Dong Nguyen,
Leszek Plaskota, Ian Sloan, Greg Wasilkowski.

Numerical Analysis and Approximation Theory Meets Data Science
Banff International Research Station
April 2018

The Multivariate Decomposition Method (MDM)

Aim: approximate infinite dimensional integral (or function)

$$I_\infty(f) = \int_{[-1/2,1/2]^{\mathbb{N}}} f(y_1, y_2, \ldots)\, \mathrm{d}\boldsymbol{y},$$

The Multivariate Decomposition Method (MDM)

Aim: approximate infinite dimensional integral (or function)

$$I_\infty(f) = \int_{[-1/2,1/2]^{\mathbb{N}}} f(y_1, y_2, \ldots) \, \mathrm{d}\boldsymbol{y},$$

where f can be decomposed by a multivariate decomposition method

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}), \qquad f_{\mathfrak{u}} \in F_{\mathfrak{u}},$$

The Multivariate Decomposition Method (MDM)

Aim: approximate infinite dimensional integral (or function)

$$I_\infty(f) = \int_{[-1/2,1/2]^{\mathbb{N}}} f(y_1, y_2, \ldots)\,\mathrm{d}\boldsymbol{y} = \sum_{|\mathfrak{u}|<\infty} I_{\mathfrak{u}}(f_{\mathfrak{u}}),$$

where f can be decomposed by a multivariate decomposition method

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}), \qquad f_{\mathfrak{u}} \in F_{\mathfrak{u}},$$

The Multivariate Decomposition Method (MDM)

Aim: approximate infinite dimensional integral (or function)

$$I_\infty(f) = \int_{[-1/2,1/2]^{\mathbb{N}}} f(y_1, y_2, \ldots)\, \mathrm{d}\boldsymbol{y} = \sum_{|\mathfrak{u}|<\infty} I_{\mathfrak{u}}(f_{\mathfrak{u}}),$$

where f can be decomposed by a multivariate decomposition method

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}), \qquad f_{\mathfrak{u}} \in F_{\mathfrak{u}},$$

specifically: using the anchored decomposition:

$$f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = f(\boldsymbol{y}_{\mathfrak{u}}; 0) - \sum_{\mathfrak{v}\subset\mathfrak{u}} f_{\mathfrak{v}}(\boldsymbol{y}_{\mathfrak{v}}) = \sum_{\mathfrak{v}\subseteq\mathfrak{u}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} f(\boldsymbol{y}_{\mathfrak{v}}; 0).$$

The Multivariate Decomposition Method (MDM)

Aim: approximate infinite dimensional integral (or function)

$$I_\infty(f) = \int_{[-1/2,1/2]^{\mathbb{N}}} f(y_1, y_2, \ldots) \,\mathrm{d}\boldsymbol{y} = \sum_{|\mathfrak{u}|<\infty} I_{\mathfrak{u}}(f_{\mathfrak{u}}),$$

where f can be decomposed by a multivariate decomposition method

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}), \qquad f_{\mathfrak{u}} \in F_{\mathfrak{u}},$$

specifically: using the anchored decomposition:

$$f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = f(\boldsymbol{y}_{\mathfrak{u}}; 0) - \sum_{\mathfrak{v}\subset\mathfrak{u}} f_{\mathfrak{v}}(\boldsymbol{y}_{\mathfrak{v}}) = \sum_{\mathfrak{v}\subseteq\mathfrak{u}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} f(\boldsymbol{y}_{\mathfrak{v}}; 0).$$

Method: approximate $I_{\mathfrak{u}}$ on sets of important dimensions $\mathfrak{u} \in \mathcal{U}_\epsilon$ by cubature formulae $Q_{\mathfrak{u},n_{\mathfrak{u}}}$ using function values.

Baldeaux, Gilbert, Gnewuch, Hefter, Hickernell, Hinrichs, Kuo, Müller-Gronbach, N., Plaskota, Ritter, Sloan, Wasilkowski, Woźniakowski, ...

Anchored decomposition and point evaluations

Prototype infinite variate function as depending on a series decomposition, e.g.,

$$f(\boldsymbol{y}) = g\left(\sum_{j\geq 1} y_j\, c_j\right),$$

Anchored decomposition and point evaluations

Prototype infinite variate function as depending on a series decomposition, e.g.,

$$f(\boldsymbol{y}) = g\left(\sum_{j \geq 1} y_j \, c_j\right) ,$$

so anchoring f at zero is practical and computable:

$$f(\boldsymbol{y}_{\mathfrak{u}}; 0) = g\left(\sum_{j \in \mathfrak{u}} y_j \, c_j\right) ,$$

Anchored decomposition and point evaluations

Prototype infinite variate function as depending on a series decomposition, e.g.,

$$f(\boldsymbol{y}) = g\left(\sum_{j\geq 1} y_j\, c_j\right),$$

so anchoring f at zero is practical and computable:

$$f(\boldsymbol{y}_{\mathfrak{u}};0) = g\left(\sum_{j\in\mathfrak{u}} y_j\, c_j\right),$$

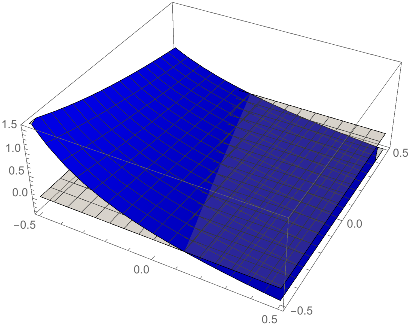
and the cost of evaluating $f_{\mathfrak{u}}$, denoted later by $\pounds(\mathfrak{u})$,

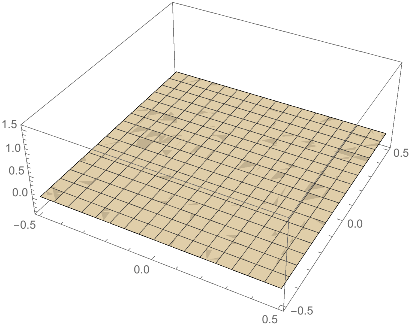
$$f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = \sum_{\mathfrak{v}\subseteq\mathfrak{u}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|}\, f(\boldsymbol{y}_{\mathfrak{v}};0),$$

can be bounded as $\pounds(\mathfrak{u}) \leq 2^{|\mathfrak{u}|}\$(\mathfrak{u})$ (with here $\$(\mathfrak{u}) \sim |\mathfrak{u}|$).

Example of anchored decomposition (on $[-1/2, 1/2]^{\mathbb{N}}$)

$$f(\boldsymbol{y}) = \left(1 + \sum_{j\geq 1} j^{-2}\, y_j\right)^{-1} - 1 \qquad f_\emptyset$$

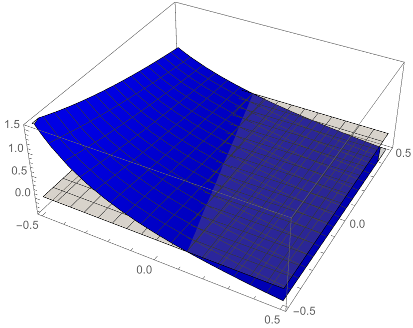


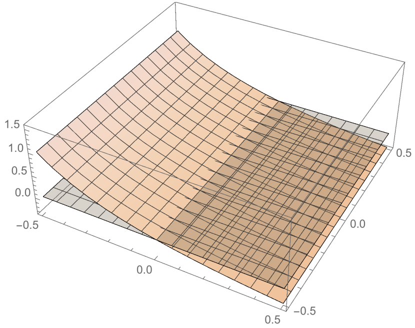


$$f_\emptyset = f(0, 0, \ldots)$$

Example of anchored decomposition (on $[-1/2, 1/2]^{\mathbb{N}}$)

$$f(\boldsymbol{y}) = \Big(1 + \sum_{j\geq 1} j^{-2}\, y_j\Big)^{-1} - 1 \qquad f_{\{1\}}(y_1)$$

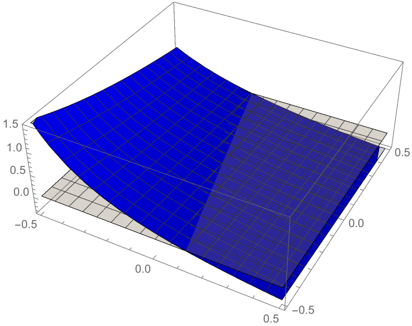


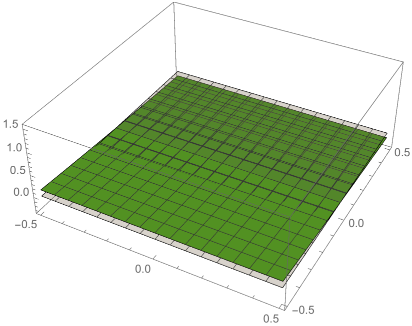


$$f_{\{1\}}(y_1) = f(y_1, 0, 0, \ldots) - f(0, 0, \ldots)$$

Example of anchored decomposition (on $[-1/2, 1/2]^{\mathbb{N}}$)

$f(\boldsymbol{y}) = \left(1 + \sum_{j\geq 1} j^{-2} y_j\right)^{-1} - 1$ $\qquad$ $f_{\{2\}}(y_2)$

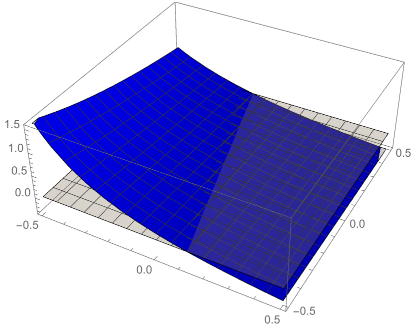


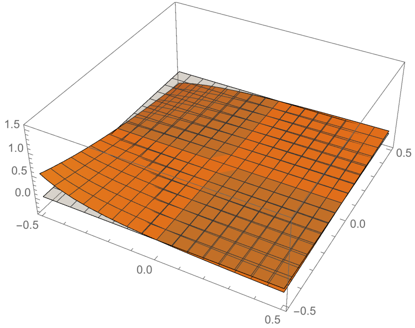


$$f_{\{2\}}(y_2) = f(0, y_2, 0, \ldots) - f(0, 0, \ldots)$$

Example of anchored decomposition (on $[-1/2, 1/2]^{\mathbb{N}}$)

$$f(\boldsymbol{y}) = \Big(1 + \sum_{j\geq 1} j^{-2}\, y_j\Big)^{-1} - 1 \qquad f_{\{1,2\}}(y_1, y_2)$$

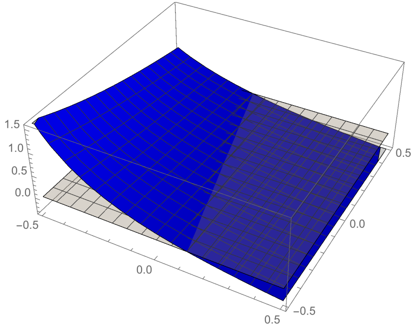


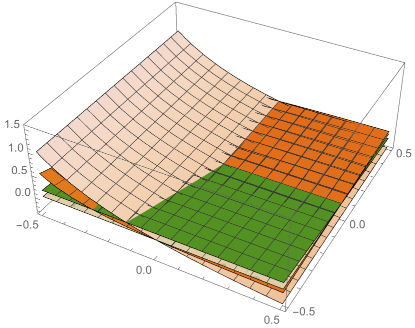


$$f_{\{1,2\}}(y_1, y_2) = f(y_1, y_2, 0, \ldots) - f(y_1, 0, \ldots) - f(0, y_2, 0, \ldots) + f(0, 0, \ldots)$$

Example of anchored decomposition (on $[-1/2, 1/2]^{\mathbb{N}}$)

$$f(\boldsymbol{y}) = \left(1 + \sum_{j\geq 1} j^{-2} y_j\right)^{-1} - 1 \qquad f_{\emptyset},\ f_{\{1\}},\ f_{\{2\}},\ f_{\{1,2\}}$$





$$f(y_1, y_2, 0, 0, \ldots) = f_{\emptyset} + f_{\{1\}}(y_1) + f_{\{2\}}(y_2) + f_{\{1,2\}}(y_1, y_2)$$

Standard problem

Consider the following elliptic PDE

$$-\nabla \cdot \big(a(\boldsymbol{x}, \boldsymbol{y}) \nabla u(\boldsymbol{x}, \boldsymbol{y})\big) = w(\boldsymbol{x}), \qquad \text{for } \boldsymbol{x} \in D, \text{ a.s. } \boldsymbol{y} \in [-1/2, 1/2]^{\mathbb{N}},$$

and $u(\boldsymbol{x}, \boldsymbol{y}) = 0$ for $\boldsymbol{x} \in \delta D$, where

- $D \subset \mathbb{R}^d$ is a "nice" bounded physical domain, $d = 1, 2, 3$,
- $\boldsymbol{y}$ is parameter/parametrization of random field,
- $a(\boldsymbol{x}, \boldsymbol{y})$ is a scalar random field, e.g.,

$$a(\boldsymbol{x}, \boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{j \ge 1} y_j \, \varphi_j(\boldsymbol{x}), \qquad y_j \sim U[-1/2, 1/2],$$

Standard problem

Consider the following elliptic PDE

$$-\nabla \cdot \big(a(\boldsymbol{x},\boldsymbol{y}) \nabla u(\boldsymbol{x},\boldsymbol{y})\big) = w(\boldsymbol{x}), \qquad \text{for } \boldsymbol{x} \in D, \text{ a.s. } \boldsymbol{y} \in [-1/2, 1/2]^{\mathbb{N}},$$

and $u(\boldsymbol{x},\boldsymbol{y}) = 0$ for $\boldsymbol{x} \in \delta D$, where

- $D \subset \mathbb{R}^d$ is a "nice" bounded physical domain, $d = 1, 2, 3$,
- $\boldsymbol{y}$ is parameter/parametrization of random field,
- $a(\boldsymbol{x},\boldsymbol{y})$ is a scalar random field, e.g.,

$$a(\boldsymbol{x},\boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{j \ge 1} y_j \, \varphi_j(\boldsymbol{x}), \qquad y_j \sim U[-1/2, 1/2],$$

- and $\sum_{j\ge1} \|\varphi_j\|_{L_\infty}^p < \infty$ for some $0 < p < 1$.

Lots of references: Adcock, Babuska, Brugiapaglia, Chkifa, Cohen, Dahmen, DeVore, Dexter, Ghanem, Gittelson, Graham, Griebel, Hoang, Karniadakis, Nobile, Scheichl, Schwab, Spanos, Tempone, Todor, Webster, Xiu, Zhang, EVERYBODY, ...

Calculate expected value (could also approximate function)

For a continuous linear functional G we wish to approximate

$$\mathbb{E}[G(u)] = I_\infty(G(u)) = \int_{[-1/2,1/2]^{\mathbb{N}}} G(u(\boldsymbol{x}, \boldsymbol{y}))\, \mathrm{d}\boldsymbol{y}.$$

From cubature point of view (MC, QMC or sparse grid, but QMC biased) there are 3 options with corresponding (dimension independent) error analysis (= truncation error + FEM error + cubature error) for parametrized PDE:

Calculate expected value (could also approximate function)

For a continuous linear functional G we wish to approximate

$$\mathbb{E}[G(u)] = I_\infty(G(u)) = \int_{[-1/2,1/2]^{\mathbb{N}}} G(u(\boldsymbol{x},\boldsymbol{y}))\,\mathrm{d}\boldsymbol{y}.$$

From cubature point of view (MC, QMC or sparse grid, but QMC biased) there are 3 options with corresponding (dimension independent) error analysis (= truncation error + FEM error + cubature error) for parametrized PDE:

1. Truncation algorithm: Dick, Gantner, Hereman, Kuo, Le Gia, N., Sloan, Schwab

Calculate expected value (could also approximate function)

For a continuous linear functional G we wish to approximate

$$\mathbb{E}[G(u)] = I_\infty(G(u)) = \int_{[-1/2,1/2]^{\mathbb{N}}} G(u(\boldsymbol{x},\boldsymbol{y}))\,\mathrm{d}\boldsymbol{y}.$$

From cubature point of view (MC, QMC or sparse grid, but QMC biased) there are 3 options with corresponding (dimension independent) error analysis (= truncation error + FEM error + cubature error) for parametrized PDE:

1. Truncation algorithm: Dick, Gantner, Hereman, Kuo, Le Gia, N., Sloan, Schwab
2. Multi level algorithm: Dick, Gantner, Hereman, Kuo, Le Gia, Schwab

Calculate expected value (could also approximate function)

For a continuous linear functional G we wish to approximate

$$\mathbb{E}[G(u)] = I_\infty(G(u)) = \int_{[-1/2,1/2]^{\mathbb{N}}} G(u(\boldsymbol{x},\boldsymbol{y}))\,\mathrm{d}\boldsymbol{y}.$$

From cubature point of view (MC, QMC or sparse grid, but QMC biased) there are 3 options with corresponding (dimension independent) error analysis (= truncation error + FEM error + cubature error) for parametrized PDE:

1. Truncation algorithm: Dick, Gantner, Hereman, Kuo, Le Gia, N., Sloan, Schwab
2. Multi level algorithm: Dick, Gantner, Hereman, Kuo, Le Gia, Schwab
3. The MDM algorithm: ...

Calculate expected value (could also approximate function)

For a continuous linear functional G we wish to approximate

$$\mathbb{E}[G(u)] = I_\infty(G(u)) = \int_{[-1/2,1/2]^{\mathbb{N}}} G(u(\boldsymbol{x}, \boldsymbol{y}))\,\mathrm{d}\boldsymbol{y}.$$

From cubature point of view (MC, QMC or sparse grid, but QMC biased) there are 3 options with corresponding (dimension independent) error analysis (= truncation error + FEM error + cubature error) for parametrized PDE:

1. Truncation algorithm: Dick, Gantner, Hereman, Kuo, Le Gia, N., Sloan, Schwab
2. Multi level algorithm: Dick, Gantner, Hereman, Kuo, Le Gia, Schwab
3. The MDM algorithm: ...

NB: both first order and higher order convergence with QMC.

The general MDM setting (no PDE for now)

Remember we take the anchored decomposition

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = f_{\emptyset} + f_{\{1\}}(y_1) + f_{\{1\}}(y_1) + \cdots + f_{\{1,2\}}(y_1, y_2) + \cdots,$$

The general MDM setting (no PDE for now)

Remember we take the anchored decomposition

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = f_{\emptyset} + f_{\{1\}}(y_1) + f_{\{1\}}(y_1) + \cdots + f_{\{1,2\}}(y_1, y_2) + \cdots,$$

then the MDM approximates $I_\infty(f)$ up to an error $\epsilon > 0$ by

$$I_\infty(f) = \sum_{|\mathfrak{u}|<\infty} I_{\mathfrak{u}}(f_{\mathfrak{u}})$$

The general MDM setting (no PDE for now)

Remember we take the anchored decomposition

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = f_{\emptyset} + f_{\{1\}}(y_1) + f_{\{1\}}(y_1) + \cdots + f_{\{1,2\}}(y_1, y_2) + \cdots,$$

then the MDM approximates $I_\infty(f)$ up to an error $\epsilon > 0$ by

$$I_\infty(f) = \sum_{|\mathfrak{u}|<\infty} I_{\mathfrak{u}}(f_{\mathfrak{u}}) \approx \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_{\mathfrak{u}}(f_{\mathfrak{u}})$$

The general MDM setting (no PDE for now)

Remember we take the anchored decomposition

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = f_{\emptyset} + f_{\{1\}}(y_1) + f_{\{1\}}(y_1) + \cdots + f_{\{1,2\}}(y_1, y_2) + \cdots,$$

then the MDM approximates $I_\infty(f)$ up to an error $\epsilon > 0$ by

$$I_\infty(f) = \sum_{|\mathfrak{u}|<\infty} I_{\mathfrak{u}}(f_{\mathfrak{u}}) \approx \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_{\mathfrak{u}}(f_{\mathfrak{u}}) \approx \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} Q_{\mathfrak{u},n_{\mathfrak{u}}}(f_{\mathfrak{u}})$$

The general MDM setting (no PDE for now)

Remember we take the anchored decomposition

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = f_{\emptyset} + f_{\{1\}}(y_1) + f_{\{1\}}(y_1) + \cdots + f_{\{1,2\}}(y_1, y_2) + \cdots,$$

then the MDM approximates $I_\infty(f)$ up to an error $\epsilon > 0$ by

$$I_\infty(f) = \sum_{|\mathfrak{u}|<\infty} I_{\mathfrak{u}}(f_{\mathfrak{u}}) \approx \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_{\mathfrak{u}}(f_{\mathfrak{u}}) \approx \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} Q_{\mathfrak{u},n_{\mathfrak{u}}}(f_{\mathfrak{u}})$$

with error

$$|I_\infty(f) - Q_\epsilon(G)| \leq \underbrace{\sum_{\mathfrak{u}\notin\mathcal{U}_\epsilon} |I_{\mathfrak{u}}(f_{\mathfrak{u}})|}_{\lesssim \epsilon/2} +$$

The general MDM setting (no PDE for now)

Remember we take the anchored decomposition

$$f(\boldsymbol{y}) = \sum_{|\mathfrak{u}|<\infty} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}}) = f_{\emptyset} + f_{\{1\}}(y_1) + f_{\{1\}}(y_1) + \cdots + f_{\{1,2\}}(y_1, y_2) + \cdots,$$

then the MDM approximates $I_\infty(f)$ up to an error $\epsilon > 0$ by

$$I_\infty(f) = \sum_{|\mathfrak{u}|<\infty} I_{\mathfrak{u}}(f_{\mathfrak{u}}) \approx \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_{\mathfrak{u}}(f_{\mathfrak{u}}) \approx \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} Q_{\mathfrak{u},n_{\mathfrak{u}}}(f_{\mathfrak{u}})$$

with error

$$|I_\infty(f) - Q_\epsilon(G)| \le \underbrace{\sum_{\mathfrak{u}\notin\mathcal{U}_\epsilon} |I_{\mathfrak{u}}(f_{\mathfrak{u}})|}_{\lesssim\epsilon/2} + \underbrace{\sum_{\mathfrak{u}\notin\mathcal{U}_\epsilon} |I_{\mathfrak{u}}(f_{\mathfrak{u}}) - Q_{\mathfrak{u},n_{\mathfrak{u}}}(f_{\mathfrak{u}})|}_{\lesssim\epsilon/2} \lesssim \epsilon.$$

$\Rightarrow$ Need properties on $f_{\mathfrak{u}} \in F_{\mathfrak{u}}$ to use optimal algorithms.

$\Rightarrow$ error = truncation error + cubature errors.

Truncation error

From application:
Assume that $f_{\mathfrak{u}} \in F_{\mathfrak{u}}$ with $\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \le B_{\mathfrak{u}}$ and $\|I_{\mathfrak{u}}\| \le C_{\mathfrak{u}}$.

Construct the active set $\mathcal{U}_\epsilon$ such that

$$\sum_{\mathfrak{u} \notin \mathcal{U}_\epsilon} |I_{\mathfrak{u}}(f_{\mathfrak{u}})| \le \sum_{\mathfrak{u} \notin \mathcal{U}_\epsilon} \|I_{\mathfrak{u}}\| \|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \le \sum_{\mathfrak{u} \notin \mathcal{U}_\epsilon} C_{\mathfrak{u}} B_{\mathfrak{u}} \le \frac{\epsilon}{2}.$$

Truncation error

From application:
Assume that $f_{\mathfrak{u}} \in F_{\mathfrak{u}}$ with $\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \le B_{\mathfrak{u}}$ and $\|I_{\mathfrak{u}}\| \le C_{\mathfrak{u}}$.

Construct the active set $\mathcal{U}_\epsilon$ such that

$$\sum_{\mathfrak{u} \notin \mathcal{U}_\epsilon} |I_{\mathfrak{u}}(f_{\mathfrak{u}})| \le \sum_{\mathfrak{u} \notin \mathcal{U}_\epsilon} \|I_{\mathfrak{u}}\| \|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}} \le \sum_{\mathfrak{u} \notin \mathcal{U}_\epsilon} C_{\mathfrak{u}} B_{\mathfrak{u}} \le \frac{\epsilon}{2}.$$

Moreover there is a decay with respect to increasing $|\mathfrak{u}|$:

$$\alpha_0 := \sup \left\{ \alpha : \sum_{|\mathfrak{u}| < \infty} (C_{\mathfrak{u}} B_{\mathfrak{u}})^{1/\alpha} < \infty \right\} > 1.$$

Compare this with ℓ_p summability with $0 < p < 1$ where $p = 1/\alpha$.

Constructing the active set $\mathcal{U}_\epsilon$ to control truncation error

Remember $|I_{\mathfrak{u}}(f_{\mathfrak{u}})| \le C_{\mathfrak{u}} B_{\mathfrak{u}}$, then for any $\alpha \in (1, \alpha_0)$ we may define

$$\mathcal{U}_\epsilon = \mathcal{U}_\epsilon(\alpha) := \left\{ \mathfrak{u} \subset \mathbb{N} : (C_{\mathfrak{u}} B_{\mathfrak{u}})^{1-1/\alpha} > \frac{\epsilon/2}{\sum_{|\mathfrak{v}|<\infty} (C_{\mathfrak{v}} B_{\mathfrak{v}})^{1/\alpha}} \right\}.$$

Constructing the active set $\mathcal{U}_\epsilon$ to control truncation error

Remember $|I_{\mathfrak{u}}(f_{\mathfrak{u}})| \le C_{\mathfrak{u}} B_{\mathfrak{u}}$, then for any $\alpha \in (1, \alpha_0)$ we may define

$$\mathcal{U}_\epsilon = \mathcal{U}_\epsilon(\alpha) := \left\{ \mathfrak{u} \subset \mathbb{N} : (C_{\mathfrak{u}} B_{\mathfrak{u}})^{1-1/\alpha} > \frac{\epsilon/2}{\sum_{|\mathfrak{v}|<\infty} (C_{\mathfrak{v}} B_{\mathfrak{v}})^{1/\alpha}} \right\}.$$

We have

$$|\mathcal{U}_\epsilon(\alpha)| < \left(\frac{2}{\epsilon}\right)^{\frac{1}{\alpha-1}} \left[\sum_{|\mathfrak{u}|<\infty} (C_{\mathfrak{u}} B_{\mathfrak{u}})^{\frac{1}{\alpha}} \right]^{\frac{\alpha}{\alpha-1}}.$$

Constructing the active set $\mathcal{U}_\epsilon$ to control truncation error

Remember $|I_{\mathfrak{u}}(f_{\mathfrak{u}})| \le C_{\mathfrak{u}} B_{\mathfrak{u}}$, then for any $\alpha \in (1, \alpha_0)$ we may define

$$\mathcal{U}_\epsilon = \mathcal{U}_\epsilon(\alpha) := \left\{ \mathfrak{u} \subset \mathbb{N} : (C_{\mathfrak{u}} B_{\mathfrak{u}})^{1-1/\alpha} > \frac{\epsilon/2}{\sum_{|\mathfrak{v}|<\infty} (C_{\mathfrak{v}} B_{\mathfrak{v}})^{1/\alpha}} \right\}.$$

We have

$$|\mathcal{U}_\epsilon(\alpha)| < \left(\frac{2}{\epsilon}\right)^{\frac{1}{\alpha-1}} \left[\sum_{|\mathfrak{u}|<\infty} (C_{\mathfrak{u}} B_{\mathfrak{u}})^{\frac{1}{\alpha}} \right]^{\frac{\alpha}{\alpha-1}}.$$

In implementation we assume product and order dependent (POD)

$$C_{\mathfrak{u}} B_{\mathfrak{u}} = (|\mathfrak{u}|!)^{b_1} \mu \prod_{j \in \mathfrak{u}} (\kappa j^{-b_2}), \quad \text{with } b_2 > \max(b_1, 0), \mu > 0, \kappa > 0,$$

then for $\epsilon \to 0$

$$d(\epsilon) := \max_{\mathfrak{u} \in \mathcal{U}_\epsilon} |\mathfrak{u}| = O(\ln(1/\epsilon) \,/\, \ln(\ln(1/\epsilon))).$$

Cubature errors: worst case error

Assume we have cubature rules $Q_{\mathfrak{u},n_\mathfrak{u}}$ which for all $f_\mathfrak{u} \in F_\mathfrak{u}$:

$$|Q_{\mathfrak{u},n_\mathfrak{u}}(f_\mathfrak{u}) - I_\mathfrak{u}(f_\mathfrak{u})| \le \frac{G_{\mathfrak{u},q} \|f_\mathfrak{u}\|_{F_\mathfrak{u}}}{(n_\mathfrak{u}+1)^q}, \qquad n_\mathfrak{u} = 0, 1, 2, \ldots,$$

e.g., QMC or sparse grid rules, where $\|I_\mathfrak{u}\|_{F_\mathfrak{u}} \le C_\mathfrak{u} \le G_{\mathfrak{u},q}$ such that this also holds for the zero algorithm $Q_{\mathfrak{u},0} = 0$.
Note $G_{\mathfrak{u},q}$ may be exponential in $|\mathfrak{u}|$, for SG we modify to hide log.

Cubature errors: worst case error

Assume we have cubature rules $Q_{\mathfrak{u},n_\mathfrak{u}}$ which for all $f_\mathfrak{u} \in F_\mathfrak{u}$:

$$|Q_{\mathfrak{u},n_\mathfrak{u}}(f_\mathfrak{u}) - I_\mathfrak{u}(f_\mathfrak{u})| \le \frac{G_{\mathfrak{u},q} \|f_\mathfrak{u}\|_{F_\mathfrak{u}}}{(n_\mathfrak{u}+1)^q}, \qquad n_\mathfrak{u} = 0, 1, 2, \ldots,$$

e.g., QMC or sparse grid rules, where $\|I_\mathfrak{u}\|_{F_\mathfrak{u}} \le C_\mathfrak{u} \le G_{\mathfrak{u},q}$ such that this also holds for the zero algorithm $Q_{\mathfrak{u},0} = 0$.
Note $G_{\mathfrak{u},q}$ may be exponential in $|\mathfrak{u}|$, for SG we modify to hide log.

Then, again with $\|f_\mathfrak{u}\|_{F_\mathfrak{u}} \le B_\mathfrak{u}$, we need

$$\sum_{\mathfrak{u} \in \mathcal{U}_\epsilon} |I_\mathfrak{u}(f_\mathfrak{u}) - Q_{\mathfrak{u},n_\mathfrak{u}}(f_\mathfrak{u})| \le \sum_{\mathfrak{u} \in \mathcal{U}_\epsilon} \frac{G_{\mathfrak{u},q} B_\mathfrak{u}}{(n_\mathfrak{u}+1)^q} \le \frac{\epsilon}{2}.$$

The number of cubature samples $n_\mathfrak{u}$ now follows from a Lagrange multiplier argument minimizing the cost to reach error ϵ.

Synopsis

Main result, e.g., Kuo, N., Plaskota, Sloan, Wasilkowski (2017),

$$Q_\epsilon(f) = \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} Q_{\mathfrak{u},n_\mathfrak{u}}(f_\mathfrak{u}) = \sum_{\mathfrak{v}\in\mathcal{U}_\epsilon^{\mathrm{ext}}} \sum_{\substack{\mathfrak{u}\in\mathcal{U}_\epsilon \\ \text{s.t. } \mathfrak{v}\subseteq\mathfrak{u}}} (-1)^{|\mathfrak{u}|-|\mathfrak{v}|} Q_{\mathfrak{u},n_\mathfrak{u}}(f(\cdot_\mathfrak{v};0)).$$

$$\begin{aligned}\mathrm{cost}(Q_\epsilon) &\le \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon(\alpha)} n_\mathfrak{u}\,\pounds(\mathfrak{u}) \le \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon(\alpha)} n_\mathfrak{u}\, 2^{|\mathfrak{u}|}\$(\mathfrak{u}) \\ &\le \left(\frac{2}{\epsilon}\right)^{1/q} \left(\sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} (G_{\mathfrak{u},q} B_\mathfrak{u})^{1/(q+1)}\right)^{1+1/q} \max_{\mathfrak{u}\in\mathcal{U}_\epsilon} 2^{|\mathfrak{u}|}\$(\mathfrak{u}).\end{aligned}$$

Then for all f with $\|f_\mathfrak{u}\|_{F_\mathfrak{u}} \le B_\mathfrak{u}$:

$$e(Q_\epsilon;\mathcal{F}) := \sup_{f\in\mathcal{F}} |I_\infty(f) - Q_\epsilon(f)| \le \epsilon.$$

Error = truncation error + FEM error + cubature error

Back to the PDE example, the total error then is

$$I_\infty(G(u)) - \sum_{\mathfrak{u} \in \mathcal{U}_\epsilon} Q_{\mathfrak{u},n_\mathfrak{u}}(G(u^{h_\mathfrak{u}}(\boldsymbol{x}, (\boldsymbol{y}_\mathfrak{u}; 0))))$$

$$=$$

Error = truncation error + FEM error + cubature error

Back to the PDE example, the total error then is

$$I_\infty(G(u)) - \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} Q_{\mathfrak{u},n_\mathfrak{u}}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))$$

$$= \left(I_\infty(G(u)) - \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0)))) \right)$$

$$+$$

Error = truncation error + FEM error + cubature error

Back to the PDE example, the total error then is

$$I_\infty(G(u)) - \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} Q_{\mathfrak{u},n_\mathfrak{u}}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))$$

$$= \left(I_\infty(G(u)) - \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0)))) \right)$$

$$+ \left(\sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0)))) - I_\mathfrak{u}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0)))) \right)$$

$$+$$

Error = truncation error + FEM error + cubature error

Back to the PDE example, the total error then is

$$
\begin{aligned}
I_\infty(G(u)) - \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} Q_{\mathfrak{u},n_\mathfrak{u}}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))
&= \left(I_\infty(G(u)) - \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))\right) \\
&\quad + \left(\sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0)))) - I_\mathfrak{u}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))\right) \\
&\quad + \left(\sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0)))) - Q_{\mathfrak{u},n_\mathfrak{u}}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))\right).
\end{aligned}
$$

Error = truncation error + FEM error + cubature error

Back to the PDE example, the total error then is

$$I_\infty(G(u)) - \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} Q_{\mathfrak{u},n_\mathfrak{u}}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))$$

$$= \left(I_\infty(G(u)) - \sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))\right)$$

$$+ \left(\sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0)))) - I_\mathfrak{u}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))\right)$$

$$+ \left(\sum_{\mathfrak{u}\in\mathcal{U}_\epsilon} I_\mathfrak{u}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0)))) - Q_{\mathfrak{u},n_\mathfrak{u}}(G(u^{h_\mathfrak{u}}(\boldsymbol{x},(\boldsymbol{y}_\mathfrak{u};0))))\right).$$

$\Rightarrow$ Balance errors such that total error is ϵ.

Function spaces

Now we need to choose the function spaces.
The details are a story for another day... but

Use [Cohen, De Vore, Schwab (2010)]:

$$\|\partial^{\boldsymbol{\nu}} u(\cdot,\boldsymbol{y})\|_V \le |\boldsymbol{\nu}|!\, \boldsymbol{b}^{\boldsymbol{\nu}} \, \frac{\|w\|_{V^*}}{a_{\min}} \qquad \text{with } b_j := \frac{\|\varphi_j\|_{L_\infty}}{a_{\min}},$$

and combine, e.g., with the (first order convergence) norm

$$\|f_{\mathfrak{u}}\|_{F_{\mathfrak{u}}}^2 = \int_{[-1/2,1/2]^{|\mathfrak{u}|}} \left|\partial_{\boldsymbol{y}_{\mathfrak{u}}}^{|\mathfrak{u}|} f_{\mathfrak{u}}(\boldsymbol{y}_{\mathfrak{u}})\right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}} = \int_{[-1/2,1/2]^{|\mathfrak{u}|}} \left|\partial_{\boldsymbol{y}_{\mathfrak{u}}}^{|\mathfrak{u}|} f(\boldsymbol{y}_{\mathfrak{u}};0)\right|^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}},$$

for $f(\boldsymbol{y}_{\mathfrak{u}};0) = G(u^{h_{\mathfrak{u}}}(\boldsymbol{x},(\boldsymbol{y}_{\mathfrak{u}};0)))$.

Test function

Consider $f : [-1/2, 1/2]^{\mathbb{N}} \to \mathbb{R}$ given by

$$f(\boldsymbol{y}) = \frac{1}{1 + \sum_{j \geq 1} j^{-\beta}\, y_j},$$

for $\beta > 1$.
Note: β is the decay (i.e., ℓ_p summability with $p > 1/\beta$).

We use

- (Randomly shifted) lattice rule with tent-transform,
- Sparse grid based on trapezoidal rule (starting with 1-point),

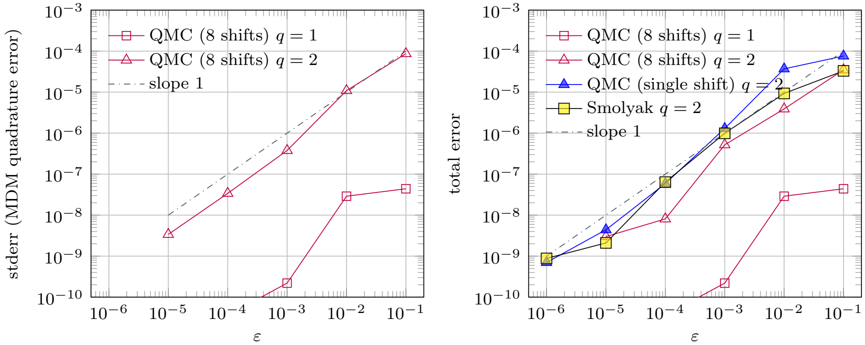
both have order 2 (deterministic).

(We took great care in implementing this efficiently in reusing computations $\to$ see Gilbert, Kuo, N., Wasilkowski (201x).)

Active set constructions for different β and ϵ

	$\beta = 4$			$\beta = 3$			$\beta = 2.5$	
ϵ	1e-1	1e-2	1e-3	1e-1	1e-2	1e-3	1e-1	1e-2
T	1.4e-4	2.8e-6	6.4e-8	4.0e-6	3.6e-8	3.8e-10	1.5e-8	4.9e-11
max $\lvert\mathfrak{u}\rvert$	3	4	5	5	6	7	8	10
$\max_{\mathfrak{u}} \mathfrak{u}$	10	28	72	86	418	1907	2528	24724
size 1	9	26	68	76	370	1686	2019	19750
2	12	48	159	195	1285	7327	10077	126882
3	5	28	132	202	1828	13117	21996	354377
4	0	4	36	80	1234	11907	26258	559155
5	0	0	1	10	361	5578	17874	536133
6	0	0	0	0	32	1145	6513	313623
7	0	0	0	0	0	69	1088	106877
8	0	0	0	0	0	0	47	18582
9	0	0	0	0	0	0	0	1210
10	0	0	0	0	0	0	0	8

Convergence orders $\beta = 3$



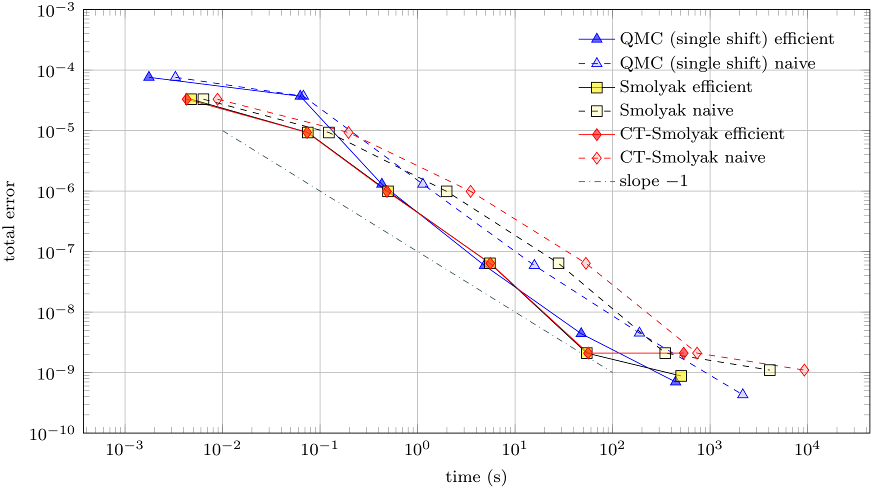
Error request against estimated standard error (lft) and total error (rgt).

Efficient implementations vs naive implementations

$\beta = 3$, reference value = 1.1011984577041

ε	method	efficient total error	efficient time (s)	naive total error	naive time (s)	speedup
1e-01	QMC	7.57e-05	0.0017576	7.57e-05	0.0032837	1.9
$t_1 = 0.000768$	Smolyak	3.26e-05	0.0047466	3.26e-05	0.0063622	1.3
$t_2 = 0.00339$	CT-Smolyak	3.26e-05	0.0042816	3.26e-05	0.0088774	2.1
1e-02	QMC	3.66e-05	0.062643	3.66e-05	0.067456	1.1
$t_1 = 0.00899$	Smolyak	9.34e-06	0.074826	9.34e-06	0.12321	1.6
$t_1 = 0.00899$	Smolyak	9.34e-06	0.074826	9.34e-06	0.12321	1.6
$t_2 = 0.049$	CT-Smolyak	9.34e-06	0.073692	9.34e-06	0.19568	2.7
1e-03	QMC	1.26e-06	0.4301	1.26e-06	1.1336	2.6
$t_1 = 0.0401$	Smolyak	9.92e-07	0.49712	9.92e-07	1.9859	4.0
$t_2 = 0.339$	CT-Smolyak	9.92e-07	0.48502	9.92e-07	3.4984	7.2
1e-04	QMC	5.90e-08	4.8547	5.90e-08	15.766	3.2
$t_1 = 0.34$	Smolyak	6.39e-08	5.5186	6.39e-08	27.89	5.1
$t_2 = 4.08$	CT-Smolyak	6.39e-08	5.5692	6.39e-08	53.191	9.6
1e-05	QMC	4.41e-09	47.64	4.51e-09	188.61	4.0
$t_1 = 2.79$	Smolyak	2.13e-09	54.331	2.12e-09	346.79	6.4
$t_2 = 41.7$	CT-Smolyak	2.11e-09	56.083	2.12e-09	734.87	13.1
1e-06	QMC	7.01e-10	442.8	4.31e-10	2163.1	4.9
$t_1 = 20.2$	Smolyak	8.76e-10	504.78	1.14e-09	4093.9	8.1
$t_2 = 435$	CT-Smolyak	2.08e-09	535.74	1.14e-09	9255.4	17.3

Total error against time



Thank you for your attention

- Implementation of the MDM for infinite dimensional integration.
- Overview papers for QMC and PDE: Kuo & N. (2016, 2018).
- Also function reconstruction using QMC point sets.
- And higher order convergence.

The Magic Point Shop!

QMC4PDE

See https://www.cs.kuleuven.be/~dirkn/qmc4pde/ and https://www.cs.kuleuven.be/~dirkn/qmc-generators/.