SAFE method for analytical approximation of multidimensional diffusion, and applications

emmanuel.gobet@polytechnique.edu

Centre de Mathématiques Appliquées and FiME,
Ecole Polytechnique and CNRS

With the support of La FONDATION du RISQUE

Joint work with R. Bompis (PhD thesis).

SAFE = Stochastic Approximation + Finite Elements

▷ **Diffusion model**: a d-dimensional stochastic differential equation

$$X_t = x_0 + \sum_{j=1}^{q} \int_0^t \sigma_j(s, X_s) \mathrm{d}W_s^j + \int_0^t b(s, X_s) \mathrm{d}s.$$

Aim: for fixed $T > 0$, analytical computation of

$$\mathbb{E}[h(X_T)].$$

Standing assumption: h is (at least) Lipschitz continuous

▷ **Applications:**

✓ alternative numerical method to Monte-Carlo or PDE methods

✓ useful for dynamic programming problems (optimal stopping, BSDEs, ...)

Advanced (different) methods: cubature on Wiener space by Kusuoka-Lyons-Victoir (2004), splitting method by Ninomiya-Victoir (2008), quantization method...

▷ **Ingredient 1:** Gaussian proxy

$$X_t^P = x_0 + \sum_{j=1}^{q} \int_0^t \sigma_j(s, x_0) \mathrm{d}W_s^j + \int_0^t b(s, x_0) \mathrm{d}s.$$

Justification (heuristics): b, σ small, or $\nabla_x b, \nabla_x \sigma$ smal, or T small.

▷ **Ingredient 2:** suitable interpolation of h by a function $\hat{h}$

▷ **Final approximation:**

$$\mathbb{E}[h(X_T)] \approx \mathbb{E}[\hat{h}(X_T^P)] + \sum_{|\alpha| \leq 3} w_{\alpha,T} \partial^{|\alpha|}_{\epsilon_{\alpha_1} \dots \epsilon_{\alpha_{|\alpha|}}} \left(\mathbb{E}[\hat{h}(X_T^P + \epsilon)] \right) \Big|_{\epsilon=0}.$$

Constraints in the derivation: $\epsilon \mapsto \mathbb{E}[\hat{h}(X_T^P + \epsilon)]$ has to be explicit (as a closed-form function), and its derivatives as well.

➠ $\hat{h}$ is built depending on the law of X_T^P.

Agenda of the talk

- ✓ Stochastic approximation step (similar to Call price expansions, with a multi-dimensional extension)
- ✓ Finite Elements step (Lagrange type, multilinear and multiquadratic interpolations)
- ✓ Final approximation, complexity analysis (improved efficiency up to dimension 10)
- ✓ Numerical tests

1st Ingredient: Stochastic approximation

Property (Gaussian proxy). Define

$$X_t^P = x_0 + \sum_{j=1}^{q} \int_0^t \sigma_j(s, x_0) \mathrm{d}W_s^j + \int_0^t b(s, x_0) \mathrm{d}s$$

and set $b_t = b(t, x_0)$, $\sigma_{j,t} = \sigma_j(t, x_0)$, $\Sigma_t = \sigma_t \sigma_t^*$.

✓ The distribution of X_T^P is normal with mean $m_T^P = x_0 + \int_0^T b_t \mathrm{d}t$ and covariance matrix $\mathcal{V}_T^P = \int_0^T \sigma_t \sigma_t^* \mathrm{d}t$.

Standing assumptions for discussing the accuracy

$(\mathcal{H})$-i) σ and b are bounded and C_b^2 in space + a bit of regularity of $D_x^2 b, D_x^2 \sigma$.

We define $\mathcal{M}_0(\sigma, b)$ and $\mathcal{M}_1(\sigma, b)$ as follows:

$$\mathcal{M}_i(\sigma, b) = \sum_{\alpha : i \le |\alpha| \le 2} (|\partial^\alpha \sigma|_\infty + |\partial^\alpha b|_\infty).$$

To allow non smooth test function: local ellipticity condition.

Recalling the covariance matrix $\mathcal{V}_T^P = \int_0^T \sigma_t \sigma_t^* \mathrm{d}t$, write

$$\mathcal{V}_{\mathbf{T}}^{\mathbf{P}} = \mathcal{U}_{\mathcal{V}} \mathcal{D}_{\mathbf{T}}^{\mathbf{P}} \mathcal{U}_{\mathcal{V}}^{-\mathbf{1}}$$

where $\mathcal{D}_{\mathbf{T}}^{\mathbf{P}} := \mathrm{diag}(\lambda_{\mathbf{1}}^{\mathbf{2}}\mathbf{T}, \dots, \lambda_{\mathbf{d}}^{\mathbf{2}}\mathbf{T})$ and for an orthogonal matrix $\mathcal{U}_{\mathcal{V}}$.

Oscillation/ellipticity assumption (only at x_0):

($\mathcal{H}$)-ii) There is a constant $\overline{C}_{\mathcal{V}} \geq 1$ such that

$$\overline{C}_{\mathcal{V}} \mathcal{M}_0(\sigma, b) \geq \max_{i \in \{1,\dots,d\}} \lambda_i \geq \min_{i \in \{1,\dots,d\}} \lambda_i \geq (\overline{C}_{\mathcal{V}})^{-1} \mathcal{M}_0(\sigma, b).$$

All subsequent estimates depend only on $\mathcal{M}_0(\sigma, b), \mathcal{M}_1(\sigma, b), \overline{C}_{\mathcal{V}}, T$.

Stochastic expansion

Interpolated process: for $\eta \in [0,1]$, set

$$X_t^\eta = x_0 + \sum_{j=1}^{q} \int_0^t \sigma_j(s, \eta X_s^\eta + (1-\eta)x_0)\mathrm{d}W_s^j + \int_0^t b(s, \eta X_s^\eta + (1-\eta)x_0)\mathrm{d}s,$$

so that $X^{\eta=1} = X$ and $X^{\eta=0} = X^P$.

$\dot{X} := \dot{X}^{\eta=0}$ is solution of the SDE:

$$\dot{X}_t = \sum_{j=1}^{q} \int_0^t \sigma'_{j,s}\ (X_s^P - x_0)\mathrm{d}W_s^j + \int_0^t b'_s\ (X_s^P - x_0)\mathrm{d}s$$

where $b'_t = \nabla_x b(t, x_0)$, $\sigma'_{j,t} = \nabla_x \sigma_j(t, x_0)$.

Taylor expansion (assuming here h is C^1):

$$\mathbb{E}[h(X_T)] = \mathbb{E}[h(X_T^P)] + \mathbb{E}[\nabla h(X_T^P)\dot{X}_T] + \mathrm{Error}_{2,h}^{\mathrm{SA}}.$$

Theorem (Second order weak approximation using the Gaussian proxy). For $\epsilon \in \mathbb{R}^d$, set $\overline{h}^P(\epsilon) = \mathbb{E}[h(X_T^P + \epsilon)]$.

Assume $(\mathcal{H})$ and suppose that $h \in \mathrm{Lip}(\mathbb{R}^{\mathrm{d}}, \mathbb{R})$. Then we have:

$$\mathbb{E}[\mathbf{h}(\mathbf{X_T})] = \mathbb{E}[\mathbf{h}(\mathbf{X_T^P})] + \mathrm{Cor}_{\mathbf{2,h}} + \mathrm{Error}^{\mathrm{SA}}_{\mathbf{2,h}},$$

where:

$$\mathrm{Cor}_{2,h} = \nabla\overline{h}^P(0)\int_0^T b'_t\Big(\int_0^t b_s \mathrm{d}s\Big)\mathrm{d}t + \sum_{i,j=1}^d \partial^2_{\epsilon_i,\epsilon_j}\overline{h}^P(0)\Big[\int_0^T (b^i_t)'\Big(\int_0^t \Sigma_{j,s}\mathrm{d}s\Big)\mathrm{d}t$$
$$+\frac{1}{2}\int_0^T (\Sigma^i_{j,t})'\Big(\int_0^t b_s\mathrm{d}s\Big)\mathrm{d}t\Big] + \frac{1}{2}\sum_{i,j,k=1}^d \partial^3_{\epsilon_i,\epsilon_j,\epsilon_k}\overline{h}^P(0)\int_0^T (\Sigma^i_{j,t})'\Big(\int_0^t \Sigma_{k,s}\mathrm{d}s\Big)\mathrm{d}t,$$

recalling $b'_t = \nabla_x b(t, x_0)$ and $(\Sigma^i_{j,t})' = \nabla_x[\sigma\sigma^*]^i_j(t, x_0)$.

The stochastic approximation error term is estimated as follows:

$$|\mathrm{Error}^{\mathrm{SA}}_{\mathbf{2,h}}| \leq_{\mathbf{c}} \mathrm{C}_{\mathrm{Lip},\mathbf{h}}\mathcal{M}_{\mathbf{1}}(\sigma,\mathbf{b})[\mathcal{M}_{\mathbf{0}}(\sigma,\mathbf{b})]^{\mathbf{2}}\mathbf{T}^{\frac{\mathbf{3}}{\mathbf{2}}}.$$

Proof. One has to transform $\mathbb{E}[\nabla h(X_T^P)\dot{X}_T]$ with

$$\dot{X}_t = \sum_{j=1}^{q}\int_0^t \sigma'_{j,s}\ (X_s^P - x_0)\mathrm{d}W_s^j + \int_0^t b'_s\ (X_s^P - x_0)\mathrm{d}s,$$

$$X_t^P = x_0 + \sum_{j=1}^{q}\int_0^t \sigma_{j,s}\mathrm{d}W_s^j + \int_0^t b_s\mathrm{d}s.$$

To switch from dW-integrals to dt-integrals, use the Malliavin calculus integration by parts formula (for deterministic process A):

$$\mathbb{E}\big[\psi(\int_0^T A_t\mathrm{d}W_t)\int_0^T c_t\mathrm{d}W_t\big] = \mathbb{E}\big[\nabla\psi(\int_0^T A_t\mathrm{d}W_t)\int_0^T A_t c_t^*\mathrm{d}t\big],$$

$$\mathbb{E}\big[\big(\psi_1(\int_0^T A_t\mathrm{d}W_t),\dots,\psi_\mathrm{d}(\int_0^T A_t\mathrm{d}W_t)\big)\int_0^T C_t\mathrm{d}W_t\big]$$

$$= \sum_{i,j=1}^{d}\mathbb{E}\big[\partial_{x_j}\psi_i(\int_0^T A_t\mathrm{d}W_t)\int_0^T (A_t C_t^*)_i^j\mathrm{d}t\big].$$

Theorem (Monte-Carlo representation of the expansions).

$$\mathbb{E}[h(X_T)] = \mathbb{E}\Big[h(X_T^P)\Big\{1 + \mathcal{W}[\Sigma, b; x_0]_0^T\big([\mathcal{V}_0^T]^{-1}(X_T^P - x_0 - \int_0^T b_t \mathrm{d}t)\big)\Big\}\Big] + \mathrm{Error}_{2,h}^{\mathrm{SA}},$$

where we set, for $\mathbf{Y} \in \mathbb{R}^d$,

$$\mathcal{W}[\Sigma, b; x_0]_0^T(\mathbf{Y}) = < \mathbf{Y}, \int_0^T b_t'\big(\int_0^t b_s \mathrm{d}s\big)\mathrm{d}t >$$

$$+ \sum_{i,j=1}^d \Big\{\mathbf{Y}^i\mathbf{Y}^j - ([\mathcal{V}_0^T]^{-1})_j^i\Big\}\Big[\int_0^T (b_t^i)'\big(\int_0^t \Sigma_{j,s}\mathrm{d}s\big)\mathrm{d}t + \frac{1}{2}\int_0^T (\Sigma_{j,t}^i)'\big(\int_0^t b_s \mathrm{d}s\big)\mathrm{d}t\Big]$$

$$+ \frac{1}{2}\sum_{i,j,k=1}^d \Big\{\mathbf{Y}^i\mathbf{Y}^j\mathbf{Y}^k - \mathbf{Y}^k([\mathcal{V}_0^T]^{-1})_j^i - \mathbf{Y}^j([\mathcal{V}_0^T]^{-1})_k^i - \mathbf{Y}^i([\mathcal{V}_0^T]^{-1})_k^j\Big\}\int_0^T (\Sigma_{j,t}^i)'\big(\int_0^t \Sigma_{k,s}\mathrm{d}s\big)\mathrm{d}t.$$

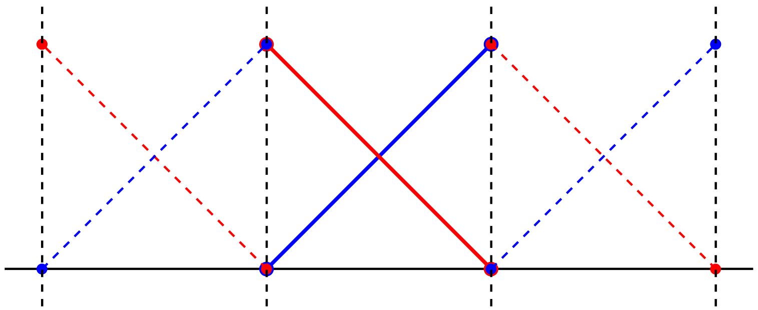
Consequence for Monte-Carlo methods:

✓ to compute $\mathbb{E}(h(X_T))$, simulate the Gaussian r.v. X_T^P and average out $h(X_T^P)\Big\{1 + \cdots(X_T^P)\Big\}$. Tested later as so-called Monte-Carlo proxy.

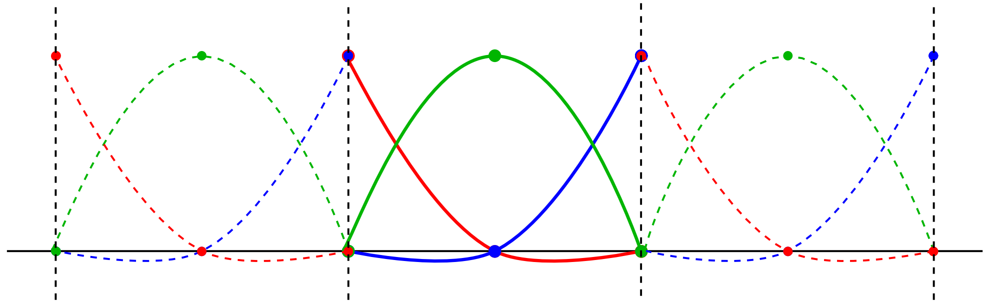
2nd Ingredient: Finite Elements

Lagrange type: only pointwise values of h are used to build $\hat{h}$.

Shape functions in dimension 1

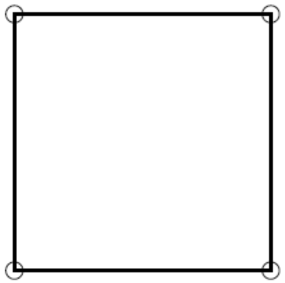


Linear interpolation

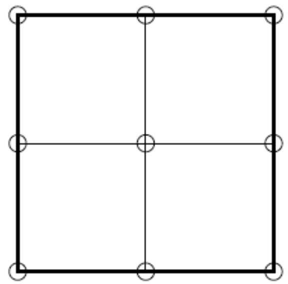


Quadratic interpolation

In dimension 2 (Tensor products **[Brenner-Scott 2008]**)



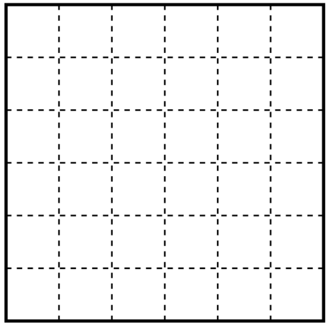
Bilinear interpolation



Biquadratic interpolation

The usual case of multilinear interpolation

Start from a regular d-dimensional grid s.t.



- ✓ covering $[-R, R]^d$, R large?
- ✓ N nodes in each direction
- ✓ mesh size $\delta = 2R/N$

Multilinear interpolation of h based on hat functions Λ built on the grid:

$$h(x) \approx \hat{h}(x) := \sum_{j_1,\dots,j_d \in \{0,\dots,N\}} h(x^{j_1,\dots,j_d}) \prod_{i=1}^{d} \Lambda^{\delta}_{x_i^{j_i}}(x_i), \qquad x \in \mathbb{R}^d.$$

Proposition (Brenner-Scott).

$$\sup_{x\in[-R,R]^d} \left| h(x) - \hat{h}(x) \right| \le c_0 \begin{cases} \mathrm{C}_{\mathrm{Lip},h}\ \delta, & \text{if } h \in \mathrm{Lip}(\mathbb{R}^{\mathrm{d}}, \mathbb{R}), \\ \left(\sum_{\alpha:|\alpha|=2} |\partial^\alpha h|_\infty \right) \delta^2 & \text{if } h \in \mathcal{C}^2_b(\mathbb{R}^d, \mathbb{R}), \end{cases}$$

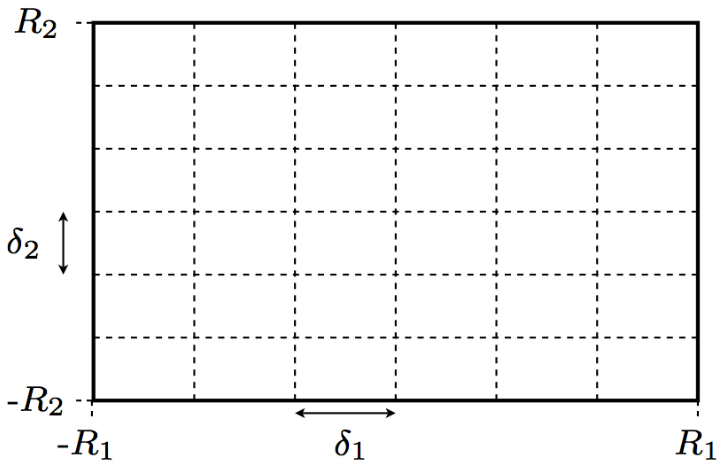
where c_0 is a universal constant.

The case of multilinear interpolation on d-parallelotope

In order to make $\mathbb{E}(\hat{h}(X_T^P))$ explicitly computable...

The grid is adjusted to eigenvalues of $\mathcal{V}_T^P = \int_0^T \Sigma_t \mathrm{d}t = \mathcal{U}_{\mathcal{V}}\ \mathrm{diag}(\lambda_1^2 T, \dots, \lambda_d^2 T)\ \mathcal{U}_{\mathcal{V}}^{-1}$:

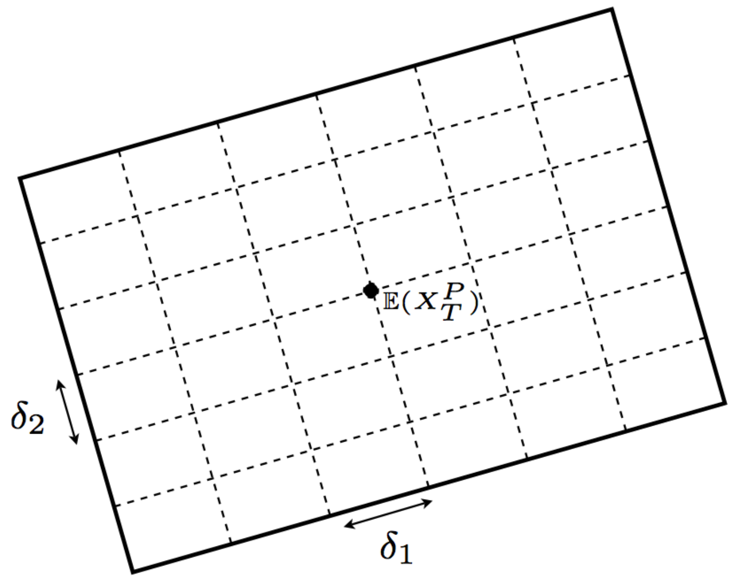
- ✓ $\mathcal{Y} = (y_i^j)_{(i,j)\in\{1,\dots,d\}\times\{0,\dots,N\}}$
- ✓ $y_i^j = -\mathrm{R}_i + j\delta_i$, $\mathrm{R}_i = \mathrm{R}\lambda_i\sqrt{T}$
- ✓ $\delta_i = \delta\lambda_i\sqrt{T}$, $\delta = \frac{2\mathrm{R}}{N}$



Affine transformation:

$$x^{j_1,\dots,j_d} := \mathcal{A}\begin{pmatrix} y_1^{j_1} \\ \dots \\ y_d^{j_d} \end{pmatrix}$$

where $\mathcal{A}\ x := \mathbb{E}(X_T^P) + \mathcal{U}_{\mathcal{V}} x$.



Final multilinear interpolation of h:

$$h(x) \approx \hat{h}(x) := \sum_{j_1,\dots,j_d \in \{0,\dots,N\}} h(x^{j_1,\dots,j_d}) \prod_{i=1}^{d} \Lambda^{\delta_i}_{y_i^{j_i}} \big((\mathcal{U}_{\mathcal{V}}^{-1}(x - m_T^P))^i \big).$$

Computation cost: $\mathcal{C}^{\text{FE}-\text{MultiLinear}}_{calculus} = \mathcal{O}((N+1)^d)$

Proposition. $\mathbb{E}[\hat{h}(X_T^P + \varepsilon)]$ is explicit, and its ε-derivatives too.

➠ we can apply the stochastic approximation to $\hat{h}$ instead of h.

Proof. One has to compute $\mathbb{E}\big[\prod_{i=1}^{d} \Lambda^{\delta_i}_{y_i^{j_i}} \big((\mathcal{U}_{\mathcal{V}}^{-1}(X_T^P + \varepsilon - m_T^P))^i \big)\big]$. Because of the axis-rotations,

$$\mathbb{E}\Big[\prod_{i=1}^{d} \Lambda^{\delta_i}_{y_i^{j_i}} \big((\mathcal{U}_{\mathcal{V}}^{-1}(X_T^P - m_T^P))^i \big)\Big] = \prod_{i=1}^{d} \mathbb{E}\big[\Lambda^{\delta_i}_{y_i^{j_i}} (\lambda_i \sqrt{T} W_1^1)\big] = \prod_{i=1}^{d} \mathbb{E}\big[\Lambda^{\delta}_{y_0^{j_i}} (W_1^1)\big] = \prod_{i=1}^{d} \beta^{\delta}_{j_i}(y_0),$$

where

$$y_0 = (y_0^j)_{j \in \{-1,\dots,N+1\}} := (-\mathrm{R} + j\delta)_{j \in \{-1,\dots,N+1\}},$$

$$\beta^{\delta}_{j}(y_0) := \frac{\beta(y_0^{j+1}) - 2\beta(y_0^{j}) + \beta(y_0^{j-1})}{\delta}, \qquad \beta(x) := x\mathcal{N}(x) + \mathcal{N}'(x).$$

Global error analysis

Superposition of

- ✓ stochastic approximation error (available for any Lipschitz h)
- ✓ truncation error + interpolation error (depends much on regularity of h)

(H1): $h \in \mathrm{Lip}(\mathbb{R}^{\mathrm{d}}, \mathbb{R})$.

(H2): $h \in \mathrm{Lip}(\mathbb{R}^{\mathrm{d}}, \mathbb{R})$, piecewise $\mathcal{C}^2$, with ∇h discontinuous across finitely many hypersurfaces

(H3): $h \in \mathcal{C}_b^2(\mathbb{R}^d, \mathbb{R})$.

Theorem (SAFE method with multilinear finite elements).

$$|\text{Global Error}| \leq c(h)\mathcal{M}_1(\sigma,b)\mathcal{M}_0(\sigma,b)^2 T^{3/2} + c(h)\exp(-R^2/4)$$

$$+\, c(h) \begin{cases} \delta + \mathcal{M}_0(\sigma,b)\sqrt{T} & \text{under } \textbf{(H1)}, \\ \delta \mathcal{M}_0(\sigma,b)\sqrt{T}\Big[\delta + \mathcal{M}_0(\sigma,b)\sqrt{T}\Big] & \text{under } \textbf{(H2)}, \\ \delta^2[\mathcal{M}_0(\sigma,b)\sqrt{T}]^2 & \text{under } \textbf{(H3)}. \end{cases}$$

Target accuracy:

$$\mathcal{E} = [\mathcal{M}_0(\sigma, b)\sqrt{T}]^3.$$

Theorem (optimal parameters of SAFE). Choose parameters R and δ as follows:

$$\mathrm{R} := 2\sqrt{\log(1/\mathcal{E})}, \qquad \delta := c\begin{cases} [\max_i \lambda_i \sqrt{T}]^2 & \text{under } \textbf{(H1)}, \\ \max_i \lambda_i \sqrt{T} & \text{under } \textbf{(H2)}, \\ \left[\max_i \lambda_i \sqrt{T}\right]^{\frac{1}{2}} & \text{under } \textbf{(H3)}. \end{cases}$$

Then, the global error is of order 3 w.r.t. $\mathcal{M}_0(\sigma, b)\sqrt{T}$:

$$\mathbb{E}[h(X_T)] = \mathbb{E}[\hat{h}(X_T^P)] + \mathrm{Cor}_{2,\hat{h}} + \mathcal{O}\big([\mathcal{M}_0(\sigma, b)\sqrt{T}]^3\big).$$

Complexity analysis: cost versus accuracy

Recall

- ✓ $\mathcal{C}^{\mathrm{FE-MultiLinear}}_{calculus} = \mathcal{O}\Big((N+1)^d\Big)$,
- ✓ $N = 2\mathrm{R}/\delta$

Corollary. With the previous notations and assumptions, as $\mathcal{E} \to 0$ we have

$$\mathcal{C}^{\mathrm{FE-MultiLinear}}_{calculus} = \begin{cases} \mathcal{O}\Big([\log(1/\mathcal{E})]^{d/2}\mathcal{E}^{-\frac{2d}{3}}\Big) & \text{under } \textbf{(H1)}, \\ \mathcal{O}\Big([\log(1/\mathcal{E})]^{d/2}\mathcal{E}^{-\frac{d}{3}}\Big) & \text{under } \textbf{(H2)}, \\ \mathcal{O}\Big([\log(1/\mathcal{E})]^{d/2}\mathcal{E}^{-\frac{d}{6}}\Big) & \text{under } \textbf{(H3)}. \end{cases}$$

Compared to Monte-Carlo ($\mathcal{C}^{\mathrm{MC}}_{calculus} = \mathcal{O}(\mathcal{E}^{-2})$), SAFE is more efficient up to dimension

$d = 3$ under **(H1)**, $d = 6$ under **(H2)**, $d = 12$ under **(H3)**.

Theorem (SAFE method with multi-quadratic elements). Assume **(H3)**. As $\mathcal{E} \to 0$ we have $\mathcal{C}^{\mathrm{FE-MultiQuadratic}}_{calculus} = \mathcal{O}\Big([\log(1/\mathcal{E})]^{d/2}\Big)$.

Numerical tests (more experiments in Bompis PhD thesis

Diffusion model $X = (X^i)_{i\in\{1,\dots,d\}}$:

$$\sigma(x) = \frac{1+x^2}{1+x+x^2}, \quad \mathrm{d}X_t^i = \big(\mu + \frac{1}{2}\nu^2\sigma^{(1)}(X_t^i)\big)\sigma(X_t^i)\mathrm{d}t + \nu\sigma(X_t^i)\mathrm{d}W_t^i, \quad X_0^i = 0.$$

Lamperti-transformation: explicit solution of the form $X_t^i = g(f(x_0) + \mu t + \nu W_t^i)$.

We take $\nu = 20\%$, $\mu = 0$, $T = 1$ and different functions:

$$h_1(x) = \frac{100 e^{\frac{1}{d}\sum_{i=1}^d x_i}}{1 + e^{\frac{1}{d}\sum_{i=1}^d x_i}} \ (\mathcal{C}^\infty \text{ with bounded derivatives, case } \mathbf{(H3)}),$$

$$h_2(x) = 100 e^{\frac{1}{d}\sum_{i=1}^d x_i} \ (\mathcal{C}^\infty, \text{ case } \mathbf{(H3)} \text{ with unbounded derivatives}),$$

$$h_3(x) = \frac{100}{d}\Big(\sum_{i=1}^d x_i\Big)_+ \ (\text{Lipschitz, case } \mathbf{(H2)}),$$

$$h_4(x) = 100 \max(x_1, \dots, x_d) \quad (\text{Lipschitz, case } \mathbf{(H1)}).$$

Table 1: Estimation of the expectations in **dimension 4** with MC, MC proxy, SAFE methods and execution time.

method / function	h_1	h_2	h_3	h_4	exec. time
`MC`	49.47 (±1.7E−3)	98.45 (±6.4E−3)	3.22 (±3.2E−3)	18.30 (±7.4E−3)	2m1s
`MC Proxy`	49.49 (±1.5E−2)	98.50 (±3.1E−2)	3.16 (±4.0E−3)	18.25 (±1.1E−2)	1m23s
`SAFE Lin (H1)`	49.48	98.48	3.17	18.28	1h16m
`SAFE Lin (H2)`	49.48	98.49	3.18	18.31	7s
`SAFE Lin (H3)`	49.48	98.50	3.23	18.47	0.3s
`SAFE Quad 1`	49.48	98.48	3.17	18.20	3s
`SAFE Quad 2`	49.48	98.48	3.17	17.98	0.2s
`SAFE Quad 3`	49.48	98.49	3.22	16.92	0.02s

Table 2: Estimation of the expectations in **dimension 6** with MC, MC proxy and SAFE methods and execution time.

method / function	h_1	h_2	h_3	h_4	exec. time
`MC`	49.47	98.27	2.47	22.42	2m58s
	(±1.3E−3)	(±5.3E−3)	(±2.5E−3)	(±6.6E−3)	
`MC Proxy`	49.47	98.29	2.43	22.25	2m2s
	(±1.9E−2)	(±3.7E−2)	(±3.3E−3)	(±1.3E−2)	
`SAFE Lin (H2)`	49.47	98.30	2.44	22.33	4h56m
`SAFE Lin (H3)`	49.48	98.31	2.49	22.54	2m7s
`SAFE Quad 1`	49.47	98.29	2.43	22.19	1h30m
`SAFE Quad 2`	49.48	98.30	2.43	21.90	1m30s
`SAFE Quad 3`	49.48	98.31	2.46	19.77	2s

Table 3: Estimation of the expectations in **dimension 8** with MC, MC proxy and SAFE methods and execution time.

method / function	h_1	h_2	h_3	h_4	exec. time
`MC`	49.47	98.17	2.03	25.04	3m57s
	(±1.2E−3)	(±4.6E−3)	(±2.1E−3)	(±6.1E−3)	
`MC Proxy`	49.46	98.18	1.99	24.74	2m41s
	(±2.2E−2)	(±4.3E−2)	(±2.9E−3)	(±1.5E−2)	
`SAFE Quad 3`	49.48	98.21	2.00	21.29	3m39s

Table 4: Estimation of the expectations in **dimension 10** with MC, MC proxy and SAFE methods and execution time.

method / function	h_1	h_2	h_3	h_4	exec. time
`MC`	49.47 (±1.0E−3)	98.12 (±4.1E−3)	1.73 (±1.9E−3)	26.93 (±5.8E−3)	4m50s
`MC Proxy`	49.49 (±2.4E−2)	98.18 (±4.8E−2)	1.70 (±2.7E−3)	26.52 (±1.8E−2)	3m15s
`SAFE Quad 3`	49.47	98.15	1.69	22.35	5h49m
`SAFE Quad 4`	49.48	98.16	1.82	13.32	1m
`SAFE Quad 5`	49.48	98.17	1.60	21.05	0.39s

Financial applications: pricing in CEV models $\mathrm{d}S_t = \nu S_t^{\beta} \mathrm{d}W_t$

$dX_t^i = \sigma(X_t^i)[\mathrm{d}W_t^i - \frac{1}{2}\sigma(X_t^i)\mathrm{d}t], \qquad X_0^i = \log(100).$

$\sigma(x) = 0.2\exp(-0.2(x - \log(100)).$

Pricing of multi-asset options with payoffs:

- ✓ $(K - \frac{1}{d}\sum_{i=1}\exp(x_i))_+$ (`Basket`)
- ✓ $(K - \exp(\frac{1}{d}\sum_{i=1} x_i))_+$ (`Geo. mean`)
- ✓ $(K - \min_{i=1,\dots,d}\exp(x_i))_+$ (`Worst of`)
- ✓ $(K - \max_{i=1,\dots,d}\exp(x_i))_+$ (`Best of`)

We report results for $d = 6$.

payoff	method / strikes	90	95	100	105	110	exec. time
Basket	MC	0.38	1.31	3.27	6.41	10.50	6h52m
		(9.1E−4)	(1.8E−3)	(2.9E−3)	(3.8E−3)	(4.5E−3)	
	MC Proxy	0.38	1.30	3.27	6.41	10.50	4m36s
		(9.5E−4)	(1.8E−3)	(2.9E−3)	(3.9E−3)	(4.6E−3)	
	SAFE Lin **(H2)**	0.38	1.31	3.27	6.41	10.49	8h2m
	SAFE Lin **(H3)**	0.40	1.33	3.29	6.41	10.47	4m47s
	SAFE Quad 1	0.38	1.30	3.26	6.41	10.50	3h2m
	SAFE Quad 2	0.38	1.30	3.26	6.41	10.50	3m25s
	SAFE Quad 3	0.38	1.32	3.26	6.40	10.47	5s
Geo. Mean	MC	0.57	1.78	4.14	7.65	11.98	
		(1.1E−3)	(2.1E−3)	(3.2E−3)	(4.1E−3)	(4.6E−3)	
	MC Proxy	0.56	1.78	4.13	7.64	11.98	
		(1.2E−3)	(2.2E−3)	(3.3E−3)	(4.2E−3)	(4.8E−3)	
	SAFE Lin **(H2)**	0.57	1.79	4.14	7.65	11.97	
	SAFE Lin **(H3)**	0.60	1.82	4.19	7.68	11.99	
	SAFE Quad 1	0.56	1.78	4.13	7.64	11.97	
	SAFE Quad 2	0.56	1.77	4.13	7.64	11.97	
	SAFE Quad 3	0.58	1.76	4.16	7.62	11.98	

Worst of	MC	14.12	18.82	23.72	28.69	33.68	
		(5.9E−3)	(6.2E−3)	(6.3E−3)	(6.3E−3)	(6.4E−3)	
	MC Proxy	14.09	18.80	23.70	28.67	33.66	
		(6.7E−3)	(7.2E−3)	(7.5E−3)	(7.8E−3)	(8.0E−3)	
	SAFE Lin **(H2)**	14.11	18.81	23.71	28.67	33.66	
	SAFE Lin **(H3)**	14.31	18.93	23.84	28.79	33.78	
	SAFE Quad 1	14.05	18.72	23.62	28.58	33.57	
	SAFE Quad 2	13.73	18.55	23.42	28.39	33.36	
	SAFE Quad 3	12.94	17.56	22.33	27.32	32.31	
Best of	MC	3.2E−3	0.02	0.08	0.28	0.75	
		(7.5E−5)	(2.0E−4)	(4.6E−4)	(9.1E−4)	(1.6E−3)	
	MC Proxy	3.2E−3	0.02	0.08	0.28	0.75	
		(7.5E−5)	(2.0E−4)	(4.5E−4)	(8.8E−4)	(1.5E−3)	
	SAFE Lin **(H2)**	3.3E−3	0.02	0.09	0.29	0.75	
	SAFE Lin **(H3)**	5.1E−3	0.03	0.10	0.35	0.81	
	SAFE Quad 1	3.9E−3	0.02	0.09	0.31	0.80	
	SAFE Quad 2	8.1E−3	0.04	0.12	0.27	0.80	
	SAFE Quad 3	1.8E−2	0.03	0.17	0.50	0.84	

Conclusion

- ✓ Analytical approximation under the asymptotics b, σ small, or $\nabla b, \nabla \sigma$ small or T small.
- ✓ Quick and quite accurate.
- ✓ Improved accuracy for smoth payoffs.
- ✓ More competitive than Monte-Carlo up to medium dimension (10).
- ✓ Alternatively, accurate proxy Monte-Carlo methods (avoid Euler discretizations).
- ✓ Perspectives: speed-up of SAFE using sparse grid techniques to reduce number of FE shape functions.