Interpolation and function representation with grids in stochastic control

FIME

Xavier Warin

EDF R&D & FiME, Laboratoire de Finance des Marchés de l'Energie (www.fime-lab.org)

September 2015

Schedule

Where is it possible to get software with effective methods in stochastic control ?

`https://gitlab.com/stochastic-control/libstoc`

- Should be open for the end of this year,
- Most of what is presented below is included
  1. C++ library, multi OS
  2. MPI and threaded version
  3. Python binding for python users
  4. Extensive documentation.....

Including the following resolution methods

1. Regression methods (adaptive local grids, sparse grids, global polynomials)
2. Dynamic Programming with Longstaff Schwartz dealing with stocks (for storage etc...)
3. Semi Lagrangian methods for HJB equations (with linear interpolator, high order interpolators for full grids and sparse grids)
4. Stochastic Dual Dynamic Programming methods for multi stock optimization
5. Framework for optimization and simulation of the optimal control.

Should be considered in future versions

- Adaptation for sparse grids (local adaptation and dimension adaptation)
- Finite Difference for singular problems,

When open, don't hesitate to test and send feed back..

Interest of the representation/interpolation of a function in stochastic control

- Calculation of conditional expectation (regressions ...)
- Evaluation of a function depending on stocks (interpolation due to dynamic programming)
- Semi Lagrangian methods (characteristics ...) needs interpolation on a grid....

Linear interpolation

The most common interpolation : given some regular meshes, interpolation linearly between the values on the mesh.

- Loved the theorist (permit to keep monotone schemes, stable etc...)
- easy to tensorize in dimension d
- Slow convergence of the interpolator $I_{1,\Delta x}$: mesh $\Delta x = (\Delta x^1, ..., \Delta x^d)$, $f \in C^{k+1}(\mathbf{R}^d)$ with $k \leq 1$

$$||f - I_{1,\Delta x} f||_\infty \leq c \sum_{i=1}^{d} \Delta x_i^{k+1} \sup_{x \in [-1,1]^d} |\frac{\partial^{k+1} f}{\partial x_i^{k+1}}|$$

If f is only Lispchitz

$$||f - I_{1,\Delta x} f||_\infty \leq K \sup_i \Delta x_i$$

Slow convergence while refining

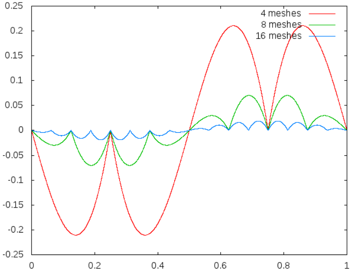


Figure: Linear interpolator error for $\sin(2\pi x)$

Spectral finite elements interpolator

On a grid $[-1, 1]$, $N+1$ points $X = (x_0, .., x_N)$, interpolate by the unique polynomial of degree N such that

$$I_N^X(f)(x_i) = f(x_i), 0 \leq i \leq N$$

.

Introduce the Lagrange polynomials $l_i^X, 0 \leq i \leq N$ (satisfying $l_i^X(x_j) = \delta_{i=j}$)

$$I_N^X(f)(x) = \sum_{i=0}^{N} f(x_i) l_i^X(x)$$

Stability and convergence of the interpolator

Standard results with $\lambda_N(X) = max_{x\in[-1,1]} \sum_{i=0}^{N} |l_i^X(x)|$ the Lebesgue constant :

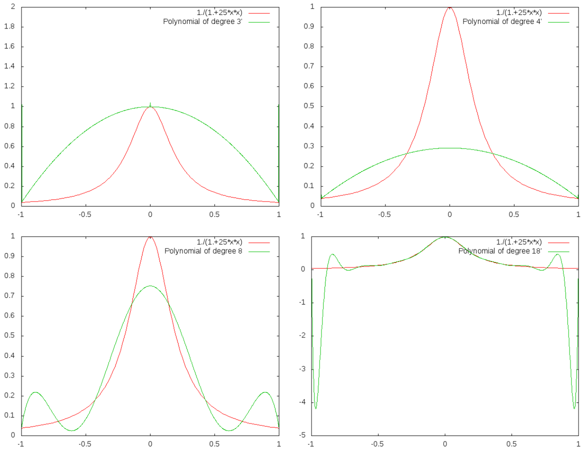
- $$||I_N^X(f)(x)||_\infty \leq \lambda_N(X)||f||_\infty \text{ Difficult to control } ||||_\infty$$
- $$||I_N^X(f)(x) - f||_\infty \leq C\lambda_N(X)w(f, \frac{1}{N})$$
  where w is the modulus of continuity
  $$w(f,\delta) = \sup_{\substack{x_1, x_2 \in [-1,1] \\ |x_1 - x_2| < \delta}} |f(x_1) - f(x_2)|$$

Try the find the grids with the best λ_N. Erdös theorem :

$$\lambda_N(X) > \frac{2}{\Pi} log(N+1) - C$$

Runge effect :uniform grid X_u is not optimal

$$\lambda_N(X_u) \simeq \frac{2^{N+1}}{eNlnN}$$



Interpolation with optimal Lebesgue constant : Gauss Legendre Lobatto

The grids in $[-1,-1]$: $\eta_1 = -1, \eta_{N+1} = 1$, and η_i $(i = 2, ..., N)$ zeros of $L_N^{'}$. Legendre polynomials satisfies :

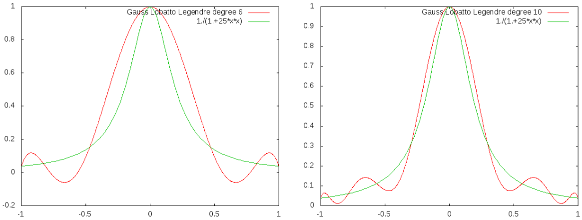
$$(N+1)L_{N+1}(x) = (2N+1)xL_N(x) - NL_{N-1}(x)$$

Lebesgue constant $\lambda_N(X_{GLL}) \simeq \frac{2}{\Pi} ln(N+1)$.
Interpolation formula on $[-1,1]$

$$\begin{aligned} I_N(f) &= \sum_{k=0}^{N} \tilde{f}_k L_k(x), \\ \tilde{f}_k &= \frac{1}{\gamma_k} \sum_{i=0}^{N} \rho_i f(\eta_i) L_k(\eta_i), \\ \gamma_k &= \sum_{i=0}^{N} L_k(\eta_i)^2 \rho_i, \end{aligned}$$

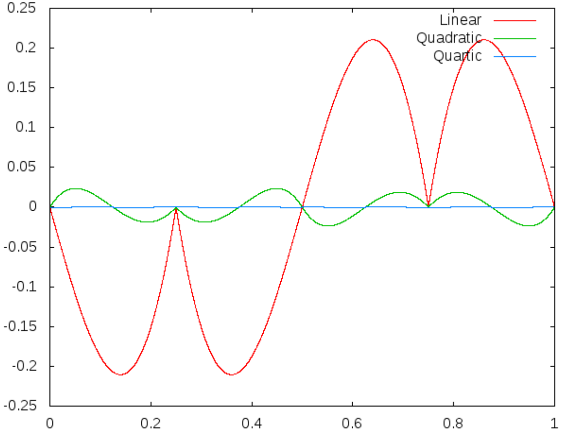
and $\rho_i = \frac{2.}{(M+1)ML_M^2(\eta_i)}, 1 \leq i \leq N+1.$

Runge effect on Gauss Legendre Lobatto



Back to the problem of the meshes with GLL interpolator

Keep 4 meshes and increase the polynomial approximation on each mesh.



Mesh construction dimension d

Define approximation by tensorization :

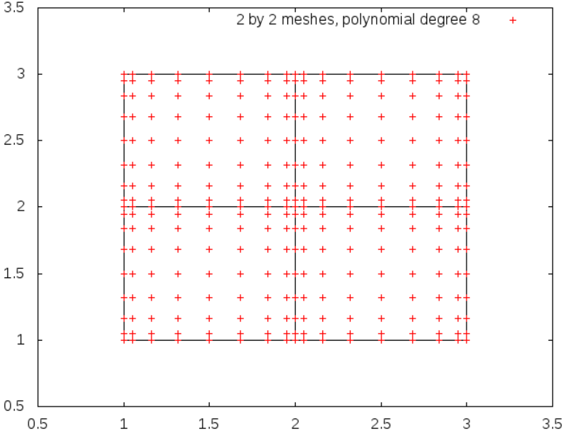


Figure: Gauss Legendre Lobatto points on 2×2 meshes

Interpolation results in dimension d with meshes of size $\Delta x = (\Delta x_1, \Delta x_2 .., \Delta x_d)$

If $f \in C^{k+1}([-1,1]^d)$, $k \leq N$

$$||f - I^X_{N,\Delta x} f||_\infty \leq c \frac{(1+\lambda_N(X))^d}{N^k} \sum_{i=1}^{d} \Delta x_i^{k+1} \sup_{x \in [-1,1]^d} |\frac{\partial^{k+1} f}{\partial x_i^{k+1}}|$$

- Of course, accuracy limited by the regularity of the solution.
- Is there a more effective way than tensorization to deal with functions in dimension d ?

Linear hierarchical representation in 1D

- Hat function : $\phi^{(L)}(x) = \max(1 - |x|, 0)$,
- By dilatation (level l) and translation (given by i) $\phi^{(L)}_{l,i}(x) = \phi^{(L)}(2^l x - i)$
- $$W_l^{(L)} \quad := \quad span\left\{\phi^{(L)}_{l,i}(x) : 1 \leq i \leq 2^l - 1, i \text{ odd}\right\}$$
- Hierarchical space
$$V_n \quad = \quad \bigoplus_{l \leq n} W_{\underline{l}}^{(L)}$$
- Nodal equivalent representation :
$$V_n \quad = \quad span\left\{\phi^{(L)}_{n,i}(x) : 1 \leq i \leq 2^l - 1\right\}$$

Example in 1 D

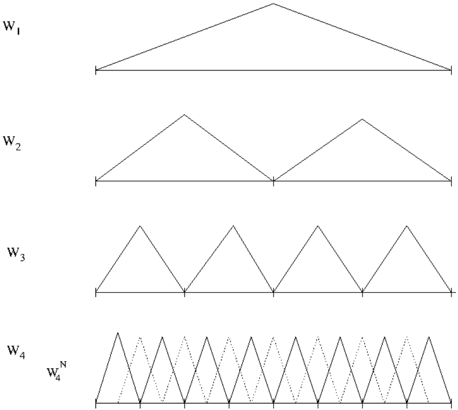


Figure: One dimensional $W^{(L)}$ spaces : $W_1^{(L)}$, $W_2^{(L)}$, $W_3^{(L)}$, $W_4^{(L)}$ and the nodal representation $W_4^{(L,N)}$

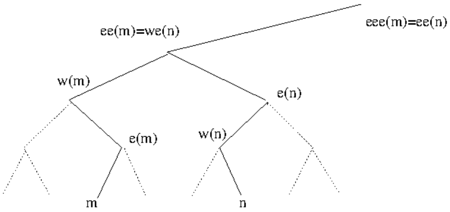
Interpolation in 1 D

$$I^{(L)}(f)(x) = \sum_{l \le n, 1 \le i \le 2^l - 1, \quad i \text{ odd}} \alpha_{l,i}^{(L)} \phi_{l,i}^{(L)}(x)$$

where for $m = x_{l,i}$

$$\alpha^{(L)}(m) := \alpha_{l,i}^{(L)} = f(m) - 0.5(f(e(m)) + f(w(m)))$$

where $e(m)$ is the east neighbor of m and $w(m)$ the west one.



Nodal versus hierarchical approach

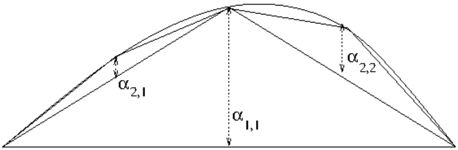


Figure: Example of hierarchical coefficients

- Hierarchical values : estimation of the discrete second derivative of the function : adaptation possible by refining at node with highest hierarchical values,
- Same solution as linear interpolation in 1 D,
- But basis function supports intersect a lot : full matrix appearing in numerical methods.

Extension of the Sparse grid to dimension d

- Basis functions : $\phi^{(L)}_{\underline{l},\underline{i}}(x) = \prod_{j=1}^{d} \phi^{(L)}_{l_j,i_j}(x_j)$ for $\underline{x} = (x_1, ...x_d)$, a multi-level $\underline{l} := (l_1, .., l_d)$ and a multi-index $\underline{i} := (i_1, .., i_d)$.
- with

$$B_{\underline{l}} := \left\{ \underline{i} : 1 \leq i_j \leq 2^{l_j} - 1, i_j \text{ odd }, 1 \leq j \leq d \right\}$$

$$W^{(L)}_{\underline{l}} := span \left\{ \phi^{(L)}_{\underline{l},\underline{i}}(\underline{x}) : \underline{i} \in B_{\underline{l}} \right\}$$

- Sparse grid

$$V_n = \bigoplus_{|\underline{l}|_1 \leq n+d-1} W^{(L)}_{\underline{l}}$$

- Full grid $V^F_n = \bigoplus_{|\underline{l}|_\infty \leq n} W^{(L)}_{\underline{l}}$
- Possible representation of V_n in term of nodal basis.

2 Full and Sparse basis

Representation of the W subspace for $\underline{l} \leq 3$ in dimension 2.

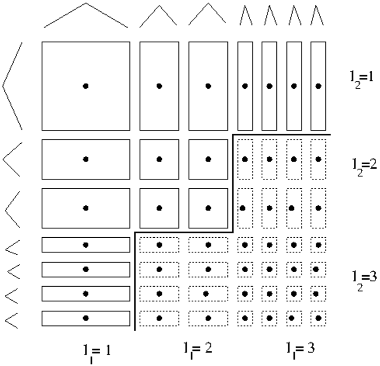


Figure: The two dimensional subspace $W_{\underline{l}}^{(L)}$ up to $l = 3$. Additional hierarchical functions corresponding to the full grid in dashed lines.

Error associated to linear space grid

With roughly $a_{n,d} = 2^n \frac{n^{d-1}}{(d-1)!}$ points, if $||\frac{\partial^{2d} u}{\partial x_1^2 \ldots \partial x_d^2}||_\infty < \infty$, [3, 5, 6]

$$||f - I^1(f)||_\infty = O(2^{-2n} log(2^n)^{d-1}) \quad (1)$$

to compare to the full grid interpolator

$$||f - I^1(f)||_\infty = O(2^{-2n}) \quad (2)$$

with 2^{dn} points.
The number of points increases slowly with the dimension with sparse grids with an error slowly above the full grid.

Incorporation boundary points

- Either add boundary points
- either modify the basis functions near the boundary.

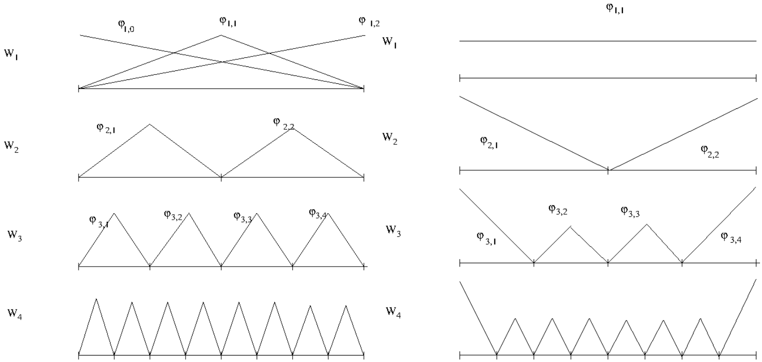


Figure: One dimensional $W^{(L)}$ spaces with linear functions with "exact " boundary (left) and "modified " boundary (right) : $W_1^{(L)}$, $W_2^{(L)}$, $W_3^{(L)}$, $W_4^{(L)}$

Grids with boundary points

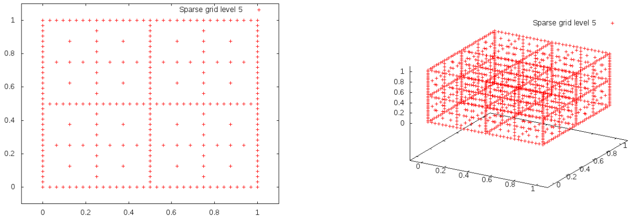
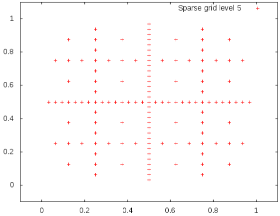


Figure: Sparse grid in dimension 2 and 3 with boundary points

Grids with extrapolation



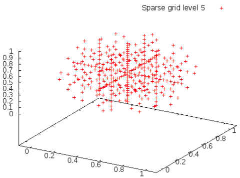


Figure: Sparse grid in dimension 2 and 3 without boundary points

Incorporation boundary points

- first method : explosion of the number of points with the dimension
  - in dimension 5, 2.8 millions points for 6401 inside the domain,
  - can't deal with some very high dimension problems
- second method : only effective if boundary points non important

Depending on the problem,

- stocks problems need accurate boundary treatment (dimension limited to 5) : first method needed
- PDE resolution in infinite domain, regression for conditional expectation will use the second method (dimension 7 to 10)

Quadratic sparse grids

How to upgrade the convergence properties of sparse grids without adding points...

On $[2^{-l}(i-1), 2^{-l}(i+1)]$ by

$$\phi^{(Q)}_{l,i}(x) = \phi^{(Q)}(2^l x - i)$$

with $\phi^{(Q)}(x) = 1 - x^2$.

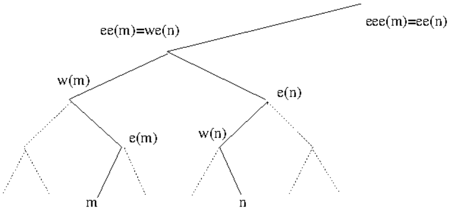
$$I^{(Q)}(f)(x) \quad = \sum_{l \le n, 1 \le i \le 2^l - 1, i \text{ odd}} \alpha^{(Q)}_{l,i} \phi^{(Q)}_{l,i}(x)$$

quadratic (cont)

where for $m = x_{l,i}$

$$
\begin{aligned}
\alpha(m)^{(Q)} &= f(m) - (\frac{3}{8}f(w(m)) + \frac{3}{4}f(e(m)) - \frac{1}{8}f(ee(m))) \\
&= \alpha(m)^{(L)}(m) - \frac{1}{4}\alpha(m)^{(L)}(e(m)) \\
&= \alpha(m)^{(L)}(m) - \frac{1}{4}\alpha(m)^{(L)}(df(m))
\end{aligned}
$$

where $df(m)$ is the direct father of the node m in the tree.



Quadratic basis functions incorporating boundary points

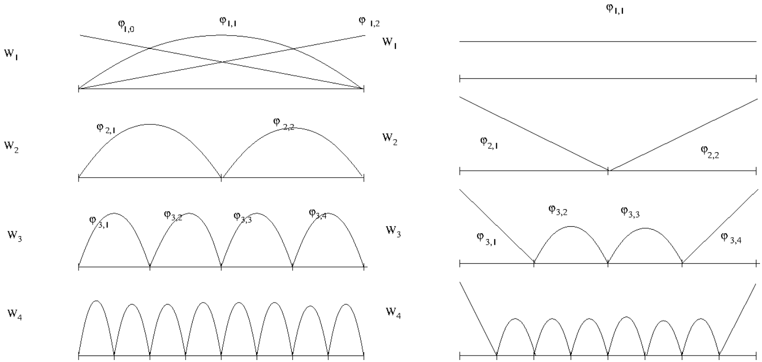


Figure: One dimensional $W^{(Q)}$ spaces with quadratic with "exact" boundary (left) and "modified" boundary (right) : $W_1^{(Q)}$, $W_2^{(Q)}$, $W_3^{(Q)}$, $W_4^{(Q)}$

High order error

- The methodology can be used for Cubic, Quartic .... interpolators,
- if $\sup_{\alpha_i \in \{2,..,p+1\}} \left\{ || \frac{\partial^{\alpha_1 + .. + \alpha_d} u}{\partial x_1^{\alpha_1} ... \partial x_d^{\alpha_d}} ||_\infty \right\} < \infty$ then for $I^2 := I^{(Q)}$, $I^3 := I^{(C)}$ by :

$$||f - I^p(f)||_\infty = O(2^{-n(p+1)} log(2^n)^{d-1}), \quad p = 2, 3$$

Adaptation for sparse grids

Two strategies:

- Local adaptation :
  - Refine nodes with highest surplus adding sons (left and right) in all directions ,
  - Derefine (if use in temporal problem) if surplus calculated to small,
  - Not sure to refine/derefine at the good points,
  - Necessity to be sure that all added nodes have fathers in all directions (if not, add points)
- Dimension adaptation: select a multilevel according to an error estimation and refine all the points at this level in all directions.

The adaptation to choose should depend on the problem....

Benchmarks for bermudean options

- d assets with same characteristics $S_0^i = 1$, $\sigma^i = 0.2$, no correlation
- Maturity $T = 1$, interest rate $r = 0.05$, strike $K = 1.$,
- $\Delta t = \frac{1}{10}$,
- Exercise dates $j\Delta t$, $j = 1, T/\Delta t$,
- pay off $(K - \frac{1}{d}\sum_{i=1}^{d} S_T^i)^+$ basket american put. Reference calculated in [7].

Use sparse grids with approximated boundary treatment.

First results dimension 1, 3

Type	LINEAR		QUADRATIC	
Nb particles	1e5	1e5	1e5	1e5
Sparse grid level	4	5	4	5
Results	0.06032	0.06032	0.06031	0.060335
Error	1.8e-05	1.e-5	7.9e-06	2.50174e-05
Time (seconds)	1.9	2.17	1.9	2.18

Table: Results in dimension 1

Type	LINEAR			QUADRATIC		
Nb particles	5e5	5e5	5e5	5e5	5e5	5e5
Sparse grid level	4	5	6	4	5	6
Results	0.02902	0.02933	0.02950	0.029029	0.02933	0.02951
Error	4.4e-04	1.3e-4	3e-5	4.4e-04	1.3e-4	4e-5
Time (seconds)	29	50	87	29	51	87

Table: Results in dimension 3

No interest in high order approximation : solution not enough regular ?

First results dimension 5, 6 (Linear sparse grids)

Nb particles	2.5e6	2.5e6	2.5e6
Sparse grid level	4	5	6
Results	0.01994	0.02009	0.02031
Error	5e-4	3.e-4	1.4e-4
Time (seconds)	448	1240	3907

Table: Results in dimension 5

Nb particles	12.5e6	12.5e6	12.5e6
Sparse grid level	4	5	6
Results	0.01733	0.01751	0.01765
Error	0.0010	0.0008	0.00065
Time (seconds)	3741	12949	126096

Table: Results in dimension 6

First conclusion for the use of sparse grids for regressions

- Till dimension 6, provide good results for american style basket options,
- No reference in dimension above 6,
- Comparing to [7], less particles seem to be necessary. Not sure sparse grids for regressions are competitive (till dimension 6),
- Cost of the method is in the matrix construction (nearly a full one).

HJB problem

$$\begin{aligned}
\frac{\partial v}{\partial t}(t,x) \quad - \quad & \inf_{a \in A}\Big(\frac{1}{2}tr(\sigma_a(t,x)\sigma_a(t,x)^T D^2 v(t,x)) + b_a(t,x)Dv(t,x) \\
& + c_a(t,x)v(t,x) + f_a(t,x)\Big) = 0 \text{ in } Q \\
v(0,x) \quad = \quad & g(x) \text{ in } \mathbf{R}^d
\end{aligned} \tag{3}$$

where

- $Q := (0,T] \times \mathbf{R}^d$, A complete metric space,
- $\sigma_a(t,x)$ is a $d \times q$ matrix,
- b_a and f_a coefficients functions defined on Q in $\mathbf{R}^d$ and R.

HJB associated to a controlled process $X_s^{t,x}$ with W_s a q dimensional Brownian motion:

$$dX_s^{t,x} = b_a(s, X_s^{t,x})dt + \sigma_a(t, X_s^{t,x})dW_s \tag{4}$$

1D example of Semi Lagrangian scheme

Using development of ϕ

$$\phi(x + b_a h + \sigma_a \sqrt{h}) = \phi(x) + h b_a D\phi(x) + \sqrt{h}\sigma_a D\phi + \frac{\sigma_a^2 h}{2} D^2\phi + \frac{\sigma_a^2 h^{3/2}}{6} D^3\phi + O(h^2)$$

$$\phi(x + b_a h - \sigma_a \sqrt{h}) = \phi(x) + h b_a D\phi(x) - \sqrt{h}\sigma_a D\phi + \frac{\sigma_a^2 h}{2} D^2\phi - \frac{\sigma_a^2 h^{3/2}}{6} D^3\phi + O(h^2)$$

So

$$(\phi(x + b_a h + \sigma_a \sqrt{h}) + \phi(x + b_a h - \sigma_a \sqrt{h}) - 2\phi(x)) \simeq 2h b_a D\phi(x) + h\sigma_a^2 D^2\phi + O(h^2)$$

And use explicit scheme

$$\begin{aligned}
v(t+h, x) &= v(t, x) + \inf_{a \in A} \frac{1}{2}[(v(t, \phi^{+}_{a,h}(t,x)) + v(t, \phi^{-}_{a,h}(t,x)) - 2v(t,x)) + \\
&\quad h c_a(t,x) v(t,x) + h f_a(t,x)] \\
\phi^{+}_{a,h}(t,x) &= x + b_a(t,x)h + \sigma_a(t,x)\sqrt{h} \\
\phi^{-}_{a,h}(t,x) &= x + b_a(t,x)h - \sigma_a(t,x)\sqrt{h}
\end{aligned}$$

Need for some interpolation

- $v(t,.)$ represented on a grid, for $v(t, \phi^{\pm}_{a,h}(t,x))$ interpolation on a grid is required,
- Regular full grids with high order interpolators are possible [8]
- Results with sparse grids [9]

Portfolio optimization problem

- $\{S_t, t \in [0, T]\}$ be an Ito process modeling the price evolution of n financial securities,
- $\{\theta_t, t \in [0, T]\}$ the investor strategy,
- Portfolio value

$$dX_t^\theta = \theta_t \cdot \frac{dS_t}{S_t} + (X_t^\theta - \theta_t \cdot \mathbb{1}) \frac{dS_t^0}{S_t^0} = \theta_t \cdot \frac{dS_t}{S_t} + (X_t^\theta - \theta_t \cdot \mathbb{1}) r_t dt,$$

- Maximize the portfolio value :

$$v_0 := \sup_{\theta \in \mathcal{A}} \mathbb{E}\left[-\exp\left(-\eta X_T^\theta \right) \right].$$

The two dimensional case ($r = 0$)

Asset dynamic (Heston model)

$$
\begin{aligned}
dS_t &= \mu S_t dt + \sqrt{Y_t} S_t dW_t^{(1)} \\
dY_t &= k(m - Y_t)dt + c\sqrt{Y_t}\left(\rho dW_t^{(1)} + \sqrt{1-\rho^2} dW_t^{(2)}\right),
\end{aligned}
$$

HJB equation :

$$
\begin{aligned}
v(T, x, y) &= -e^{-\eta x} \\
0 &= -v_t - k(m-y)v_y - \frac{1}{2}c^2 y v_{yy} - \\
& \sup_{\theta \in \mathbf{R}} \left(\frac{1}{2}\theta^2 y v_{xx} + \theta(\mu v_x + \rho c y v_{xy})\right)
\end{aligned}
$$

Quasi explicit solution (Zariphopoulou)

$$
v(t, x, y) = -e^{-\eta x} \left\| \exp\left(-\frac{1}{2}\int_t^T \frac{\mu^2}{\tilde{Y}_s} ds\right)\right\|_{L^{1-\rho^2}}
$$

where the process $\tilde{Y}$ is defined by

$$
\tilde{Y}_t = y \quad \text{and} \quad d\tilde{Y}_t = (k(m - \tilde{Y}_t) - \mu c \rho)dt + c\sqrt{\tilde{Y}_t} dW_t.
$$

Two dimensional case : no adaptation , exact boundary treatment

Parameters : $\eta = 1$, $\mu = 0.15$, $c = 0.2$, $k = 0.1$, $m = 0.3$, $Y_0 = m$, $\rho = 0$, $T = 1$, $X_0 = 1$, reference -0.3534.

Table: Portfolio optimization in dimension 2, no adaptation

	LINEAR		QUADRATIC		CUBIC	
Level	Solution	Time	Solution	Time	Solution	Time
6	-0.3678	5	-0.3622	5	-0.3629	5
7	-0.3670	14	-0.3433	15	-0.3360	16
8	-0.3565	40	-0.3555	40	-0.3565	43
9	-0.3550	105	-0.3533	109	-0.3531	116
10	-0.3539	274	-0.3535	283	-0.3535	304
11	-0.3536	700				
12	-0.3535	1757				

Two dimensional case : extrapolated boundary

Nearly same results but less expensive in computing time.

Table: Portfolio optimization in dimension 2, no adaptation , extrapolated boundary treatment

	LINEAR		QUADRATIC		CUBIC	
Level	Solution	Time	Solution	Time	Solution	Time
6	-0.3678	3	-0.3576	3	-0.3575	4
7	-0.3668	8	-0.3522	9	-0.3519	9
8	-0.3579	24	-0.3536	24	-0.3536	26
9	-0.3551	64	-0.3535	67	-0.3535	71
10	-0.3539	173				
11	-0.3536	451				
12	-0.3535	1145				

two dimensional case : extrapolated boundary and adaptation

Table: Adaptation , extrapolated boundary , initial level 5

	LINEAR		QUADRATIC		CUBIC	
Precision	Solution	Time	Solution	Time	Solution	Time
0.001	-0.3593	17	-0.3495	13	-0.3535	10
0.00025	-0.3553	39	-0.3545	25	-0.3535	18
6.25e-05	-0.3542	84	-0.3536	50		
1.56e-05	-0.3537	157	-0.3535	90		

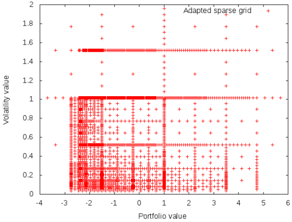


Figure: Example of adapted meshes in dimension 2

A 5 dimensional problem

- OU process for the rate

$$dr_t = \kappa(b - r_t)dt + \zeta dW_t^{(0)}.$$

- Heston model for the second security , CEV-SV model for the first security

$$\begin{aligned} dS_t^{(i)} &= \mu_i S_t^{(i)} dt + \sigma_i \sqrt{Y_t^{(i)}} {S_t^{(i)}}^{\beta_i} dW_t^{(i,1)}, \quad \beta_2 = 1, \\ dY_t^{(i)} &= k_i \left(m_i - Y_t^{(i)} \right) dt + c_i \sqrt{Y_t^{(i)}} dW_t^{(i,2)} \end{aligned}$$

- State $(t, X_t, r_t, S_t^{(1)}, Y_t^{(1)}, Y_t^{(2)})$, value function $v(t, x, r, s_1, y_1, y_2)$ solution of an HJB equation

Results in 5D

Space grids discretization for the commands (2 dimensional)

Table: No adaptation , extrapolated boundary treatment

	LINEAR		QUADRATIC		CUBIC	
Level	Solution	Time	Solution	Time	Solution	Time
6	-0.7167	50	-0.2889	51	-0.2933	52
7	-0.3326	230	-0.3035	227	-0.3044	233
8	-0.2980	1032	-0.3124	1047	-0.3132	1091
9	-0.3134	4641	-0.3092	4716	-0.3091	4980
10	-0.3112	21263	-0.3089	21238	-0.3089	22500

Table: adaptation , extrapolated boundary treatment, initial level equal to 7

	LINEAR		QUADRATIC		CUBIC	
Precision	Solution	Time	Solution	Time	Solution	Time
0.001	-0.3392	1311	-0.3085	1166	-0.3091	1170
0.00025	-0.3116	2874	-0.3098	2212	-0.3097	2192
6.25e-05	-0.3092	5667	-0.3101	3817	-0.3105	3738
1.56e-05	-0.3101	10115	-0.3095	6396	-0.3095	6262

References

- H.J. Bunged, M. Griebel, *Sparse Grids*, Acta Numerica, volume 13, (2004), pp 147-269
- D Pflüger, *Spatially Adaptive Sparse Grids for High-Dimension problems*, Dissertation, für Informatik, Technische Universität München, München (2010).
- H.-J. Bungartz.,*Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung*. Dissertation, Fakultät für Informatik, Technische Universität München, November 1992.
- T. Gerstner, M. Griebel ,*Dimension-Adaptive Tensor-Product Quadrature*, Computing 71, (2003) 89-114.
- H.-J. Bungartz.,*Concepts for higher order finite elements on sparse grids*, Proceedings of the 3.Int. Conf. on Spectral and High Order Methods, pp. 159-170 , (1996)