# A new algorithm (CoDeFi) for the curse of dimensionality in finance

Philippe G. LeFloch [a] and Jean-Marc Mercier [b]

[a] *Laboratoire Jacques-Louis Lions, Centre National de la Recherche Scientifique, Université Pierre et Marie Curie, 4 Place Jussieu, 75252 Paris, France.*

[b] *MPG-Partners, 23 rue Lavoisier, 75008 Paris, France.*

**Abstract**

We present a new algorithm (CoDeFi) to tackle the Curse Of Dimensionality In Finance and deal with a broad class of Partial Differential Equations (PDE's) including the Kolmogorov equations as, for instance, the Black and Scholes equations. As a main feature, our method allows one to solve the Kolmogorov equations in large dimensions and provides a very general framework to deal with risk measurements. In financial applications, the number of dimensions corresponds to the number of underlyings or risk sources.

**Résumé**

Nous présentons un nouvel algorithme (CoDeFi) pour aborder la malédiction dimensionnelle en mathématiques financière, en considérant une large classe d'équations aux dérivées partielles (EDP), incluant les équations de Kolmogorov, comme par exemple les équations de Black et Scholes. La principale caractéristique de notre méthode est de permettre la résolution des équations de Kolmogorov en dimension haute, et fournit ainsi un environnement très général pour la mesure des risques. Pour les applications en Finance, le nombre de dimensions correspond au nombre de sources de risques.

## Version française abrégée

La *malédiction dimensionnelle* (CoD - le terme provient d'un article de 1957 de R.E. Bellman) décrit des temps de calcul augmentant exponentiellement avec le nombre de sources de risques, et nous proposons dans cette note une solution à ce problème ouvert depuis longtemps. Notre approche est basée sur une combinaison de techniques provenant de la théorie des équations aux dérivés partielles : des trajectoires Monte-Carlo, une méthode numérique classique qui n'est pas "maudite" (non affectée par CoD), sont utilisées comme des grilles de calcul mouvantes. Ces grilles permettent de résoudre les équations de Kolmogorov, en utilisant des techniques de maillage non structuré conjointement avec la théorie du transport optimal, pour passer outre la malédiction. Nous utilisons également des techniques de calibration zero-erreur, pour une prise en compte parfaite des instruments de réplication ou les besoins de modélisation financière complexe. Toutes ces techniques sont regroupées dans un

*Email addresses:* contact@philippelefloch.org (Philippe G. LeFloch), jean-marc.mercier@mpg-partners.com (Jean-Marc Mercier).

environnement de calcul que nous appelons CoDeFi. CoDeFi peut être vu comme un environnement de mesure de risque très général, implémentant l'algorithmie présentée dans [3]–[4].

Cette note présente ces technologies et propose un premier benchmark pour le cas multi-dimensionnel, en le complétant jusquà la 64 ème dimension (ou 64 sources de risque). A notre connaissance, peu d'autres technologies pourraient compléter ce test. La quantification optimale [1] ou l'analyse par ondelette [9] pourraient être utilisés jusqu'à disons 10 dimensions. Au delà, les techniques type Monte-Carlo américain [7] pourraient peut-être fournir des bornes inférieures, ces méthodes étant connues pour fournir des exercices sub-optimaux.

Nous concluons en notant que ce même environnement de calcul est utilisé pour des simulations dans le cadre de problèmes hyperboliques non-linéaires [**?**].

## English version

## 1. Introduction

The *Curse of Dimensionality* (CoD) [1] refers to the fact that the computational time increases exponentially with the number of risk sources, and in the present note we propose a solution to this long-standing open problem. Our approach is based on a combination of techniques from the theory of partial differential equations (PDE's): Monte-Carlo trajectories, a classical numerical method that is not cursed (not affected by CoD) are used as moving-in-time grids. These grids allow to solve Kolmogorov equations, using unstructured mesh PDE techniques together with optimal transport theory, to overcome the cursing. We are also using zero-error calibration technics, for a perfect fit of replication instruments or complex financial modeling. All these technics are bundled together in a framework that we call CoDeFi: it can be seen as a general risk measurement framework based on the algorithm presented in [4]–[6].

This note reviews this technology and proposes a first benchmark to the multi-dimensional case, filling it up to 64 dimensions (or risk sources). Indeed, to our knowledge, few other technology could benchmark CoDeFi framework: optimal quantization [1] or wavelet analysis [9] might be used for up to, let say 10 dimensions. Above this limit, American-Monte Carlo methods [7] might provide lower bounds, as these methods are known to compute sub-optimal exercizing.

To conclude this introduction, we emphasize that the same computational framework is used to provide simulations for non linear hyperbolic problems [3].

## 2. General description of the method

### 2.1. *Preliminaries*

Let consider $0 < D \in \mathbb{N}$ underlyings or risk sources. Let us call $\mu(t,x)$, $t \geq 0$, $x \in \mathbb{R}^D$, the probability density measure defined by the underlying process. Let $\Lambda = [0,1]^D$ the unit cube. To understand the overall methodology, we start recalling some known results.

From a standard result (going back to Brenier [2]) we can define the quantile of any probability measure as a change of variable,

$$\mu(t,\cdot) = S(t,\cdot)_{\#}m, \quad S(t,\cdot) : \Lambda \mapsto \mathbb{R}^D \tag{1}$$

where $m$ is the Lebesgue measure over $\Lambda$, since $S = \nabla h$, $h$ convex, is unique. $S(t,\cdot)_{\#}m$ is an optimal transport notations, see [11] for a review of Optimal transport theory. For the needs of this note, it is enough to think to $S(t,\cdot)$ as a map, inducing a natural change of variable to Monte-Carlo methods: $S(t,\cdot)$ can be used for instance together with a random generator to sample the underlying process at time $t$.

We recall that *Fokker-Planck* equations are equations describing the underlyings dynamics:

$$\partial_t \mu + \mathcal{L}\mu = 0, \tag{2}$$

where $\mathcal{L}(t,\cdot)$ is a (usually *unknown*) *positive* differential operator of *parabolic* type.

1. This term comes from a 1957 paper by R.E. Bellman.

Backward *Kolmogorov* equations are adjoint of Fokker-Planck equations, submitted to terminal cauchy conditions at future dates $T > 0$ of any events as are cash-flows

$$\partial_t P + \mathcal{L}^* P = 0, \quad P(T, \cdot) = P_T(\cdot) \tag{3}$$

where $\mathcal{L}^*$ stands for adjoint operator. Backward Kolmogorov equations computes the future value of any contract (resp. portfolio of $M$ contracts) as a scalar valued function $P(t,x) \in \mathbb{R}$ (resp. vector-valued functions $P(t,x) \in \mathbb{R}^M$), $t \geq 0$, $x \in \mathbb{R}^D$, *knowing* $x$, i.e. hypothesizing that the underlying worths $x$ at time $t$.

Note that these equations are set in $\mathbb{R}^D$. Hence we localize them, *transporting* into the unit cube $\Lambda$, by using a change of variable defined by the quantile $S(t,\cdot)$: we call $P \circ S : \Lambda \mapsto \mathbb{R}$ a *transported* solution of backward Kolmogorov equation. Kolmogorov equations can be coupled with strategies $\mathcal{S}(t,x,P)$, summarized in the following system of equations, that is a PDE version of transported optimal-stopping problems

$$\max\Big((\partial_t P) \circ S + (\mathcal{L}^* P) \circ S, \mathcal{S}(t, S, P \circ S)\Big) = 0, \quad P(t,x) \in \mathbb{R}^M \tag{4}$$

### 2.2. *Scope of the CoDeFi algorithm*

CoDeFi proposes a framework to deal and solve equations of kind (4). It is almost a universal framework in the sense that, to our knowledge, all financial models (usually prescribing specific underlying dynamic, as are for instance normal, log-normal, stochastic volatility, local volatility or local correlation models...) fit into the Fokker-Planck setting (2). In the same vein, all classical risk measurements (e.g. pricing, hedging, what-if scenari, risk measurements as RWA, CVA, portfolio strategy evaluation), can be deduced from the knowledge of $P \circ S$, solution of (4).

A typical problem solving in CoDeFi comes in two phases

— *Calibration* : this phase corresponds to find out a probability measure $\mu(t,\cdot)$ that fit market prices of replication type instruments;

— *Valuation* : once the probability measure $\mu(t,\cdot)$ is calibrated, then find $\mathcal{L}$, solves (4), and output risk measurements, computed from the surface $\{t, x, P(t,x)\}$;

In the rest of this overview, we illustrate CoDeFi methodology with numerical results.

## 3. Implementation and numerical results

### 3.1. *Preliminaries*

For numerical applications, there are two essentials parameters, central in all notations, and main limiting factors in term of computational cost:

— $D$ is the number of underlyings or risk sources.

— $N$ is the number of grid points.

The overall complexity of CoDeFi algorithms can be bound to $\mathcal{O}((N+D)^3)$ operations. For such a number of operations, the relative error commited to solve Kolmogorov equations (4) is $\epsilon = \mathcal{O}(1/N)\%$. To have some figures in mind, CoDeFi solves today Kolmogorov equations (4) with $D = 64$, $N = 512$, with time of order 15 minutes, for a precision of order $1/512 \sim 0.2\%$, using one core of the eight of an Intel 4770 processor on a Laptop computer.

### 3.2. *Generation of the computational grid*

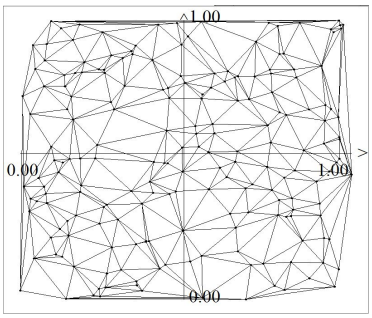
PDE methods relies on grids. We use grid of points belonging to the unit cube $\Lambda = [0,1]^D$, and denotes the grid as a matrix $Y = \{Y_n\}_{n=1..N} \in \mathbb{R}^{N \times D}$. CoDeFi is primarily a Monte-Carlo method, and can use any kind of sampling of the uniform law over $\Lambda$ as grid. For instance Figure (2) plots a two-dimensional grid of 200 points, using the Mersenne Twister generator mt19937, see [8]. We recall that Pseudo-random generator owns a statistical convergence rate of order $\epsilon = \mathcal{O}(1/\sqrt{N})\%$.

Another alternative is to use quasi-random sequences. Figure (2) plots a 200 points grid using a Sobol, see [10], low-discrepancy generator. We recall that Sobol generators have a convergence rate of order $\epsilon = \mathcal{O}((\ln N)^D/N)\%$.

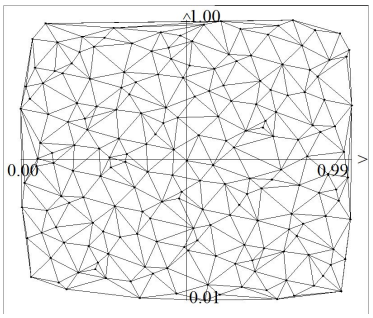
Table 1
Market values of European Call SX5E Mat 3M. Spot 3064.03

| Strike % | 0.8 | 0.9 | 0.95 | 0.975 | 1 | 1.025 | 1.05 | 1.1 | 1.2 |
|---|---|---|---|---|---|---|---|---|---|
| Value | 559.2 | 292.6 | 180.7 | 133.6 | 93.76 | 61.59 | 37.99 | 11.34 | 0.31 |

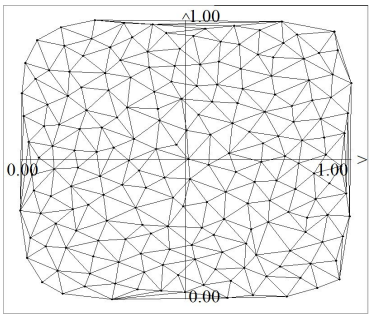
Finally, since mesh repartition is an important feature for PDE methods, we propose to use special sequences, that we call optimal discrepancy sequences [2] , to generate the grid. Such sequences reaches the above cited rate of convergence, of order $\epsilon = \mathcal{O}(1/N)\%$, but require $\mathcal{O}(N^2 D)$ operations to compute. Since PDE methods works usually with few grid points ($N = 512$ is enough), we prefer sacrifying computational time to mesh quality.



Quasi Random Sequence



Sobol sequence



Optimal Discrepancy Sequence

Figure 1. Different grids with Delaunay meshes

### 3.3. *The calibration algorithm*

As quoted above, the calibration algorithm consists in finding a probability measure $\mu(t,\cdot)$ that fits market prices of replication type instruments. Due to Brenier result, this is equivalent to finding out a quantile $S(t,\cdot) : \Lambda \mapsto \mathbb{R}^D$, see (1), that fits prices. From a numerical point of view, it amounts finding a matrix $S^t = \{\nabla h(t, Y_n)\}_{n=1..N} \in \mathbb{R}^{N\times D}$, $h$ convex. For instance, consider the table (1), that includes real quotes of call options on index $SX5E$ having maturity 3 months. Then Figure (2) shows a quantile calibrated to these quotes.

Let $P(T,x)$, $x \in \mathbb{R}^D$, the scalar-valued payoff of any replication instrument having maturity $T$, and let $C \in \mathbb{R}$ its observed market price. The notion of zero-error calibration means precisely, for any time,

$$C = \frac{1}{N} \sum_{n=1..N} P(t, S_n^t), \quad 0 \le t \le T \tag{5}$$

where $P(t,\cdot)$ is the numerical solution, computed with CoDeFi, to the kolmogorov equation (3).

This calibration algorithm scales up to higher dimensions and numerous market data quite efficiently. To illustrate this feature, consider an european best-of option, having maturity $T = 10Y$, written over $D$ underlyings,

2. there might be a link, that remains to establish, between such sequences and optimal quantizers of the uniform law over $\Lambda$

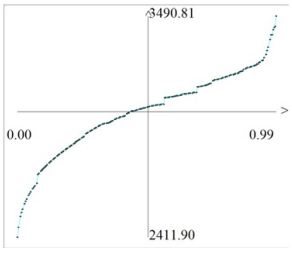



Figure 2. Calibrated SX5E quantile for $N = 256$, $D = 1$.

Table 2
Price of European Best-of options MAT 10Y

| | D=1 | D=4 | D=16 | D=64 |
|---|---|---|---|---|
| MC N = 1048576 | 0.12578 | 0.35941 | 0.68796 | 1.0166 |
| N=32 | 0.128275 | 0.340435 | 0.678092 | 0.927398 |
| N=128 | 0.126521 | 0.349573 | 0.693397 | 0.982611 |
| N=512 | 0.125921 | 0.359632 | 0.688982 | 1.0144 |

and payoff $P(T,x) = (|x|_\infty - K)^+$, where $|x|_\infty = \sup_{d=1..D}(|x_d|)$. Each underlying follows an independant log-normal process, with spot equal to one, volatility 10%, with no dividend and risk neutral set to zero. Table (2) presents prices with $D = 1, 4, 16, 64$, $N = 32, 128, 512$, of this option, computed as $\frac{1}{N}\sum_{n=1..N} P(T, S_n^T)$, where $S^T$ are calibrated quantiles. Table (2)-line MC- presents prices using a Monte-Carlo method with $1048576 \times D$ sequence of a pseudo-random Mersenne twister MT19937, that are confident with a relative error estimated to $0,1\% \sim 1/\sqrt{1048576}$.

To compute the calibrated sequences $S^T$, the calibration algorithm considered exactly $2D + D(D-1)/2$ calibration instruments (i.e. 2144 instruments for $D = 64$) :

— *D Futures* : these are virtual instruments having payoff $F_d(T,x) = x_d$, $d = 1..64$;

— *D Variances* : these are virtual instruments having payoff $V_d(T,x) = \left(x_d - \bar{S}_d\right)^2$, $d = 1..64$, $\bar{S}_d$ being the expectation of the marginal process;

— *D(D-1)/2 Correlations* : these are virtual instruments having payoff $C_{d_1,d_2}(T,x) = \frac{(x_{d_1} - \bar{S}_{d_1})(x_{d_1} - \bar{S}_{d_2})}{V_{d_1} V_{d_2}}$, $1 \le d_1 < d_2 \le D$, $V_d$ being the variance of the marginal process;

For these tests, market prices of these virtual Futures were set to one, that is the theoretical marginal expectations, market prices of Variances were set to theoretical marginal variances of log-normal at time $T = 10Y$, and, since underlyings are following independant processes, the correlations were set to zero. In particular, with the calibration setting above, (5) implies that the correlation matrix satisfies (up to zero-error machine), at each time,

$$\frac{(\bar{S}^t)^* \times \bar{S}^t}{\|\bar{S}^t\|^2} = I, \quad 0 \le t \le T, \tag{6}$$

where $\|S\|^2 = Tr((S)^* \times S)$ is the standard Frobenius matrix norm, $\bar{S} := \{S_d^n - \bar{S}_d\} \in \mathbb{R}^{N \times D}$ is the averaged process, and $I \in \mathbb{R}^{D \times D}$, is the identity matrix; Note that the calibration step requires more points than dimensions, thus failed for the case $D = 64$, $N = 32$ of table (2).

Obviously, zero-error calibration is possible only if no arbitrage exists in input data markets. If arbitrage holds, the calibration remains stable but can't fit prices. Indeed, this calibration algorithm can detect market arbitrages within a large collection of instruments and underlyings, and might also propose a explanation to highlight them.

### 3.4. *The valuation algorithm*

This section is devoted to present some numerical results related to the computation of solutions to (4), once the calibration step has been performed. In this valuation phase, the quantile $S(t,\cdot)$ of the probability measure $\mu(t,\cdot)$ is known.

A solution to the optimal stopping problem (4) is approximated as $P^t \sim P(t, S^t) \in \mathbb{R}^N$, where $t$ belongs to a fixed time-grid, and $S^t \in \mathbb{R}^{N \times D}$ is a calibrated quantile. Let $s \le t$ two consecutive times of this time grid. To *retroprapagate* $P$ from $t$ to $s$, we proceed as follows:

$$P^s = \max\left(\Pi^{(t,s)} P^t, \mathcal{S}(t, S^s, \Pi^{(t,s)} P^t)\right), \quad \Pi^{(t,s)} := \left(\pi_{n,m}^{(t,s)}\right)_{n,m=1..N} \in \mathbb{R}^{N \times N}.$$

The matrix $\Pi^{(t,s)}$, that is the generator to the Kolmogorov equation (3), is a *bi-stochastic* matrix, and is computed explicitly at each step. This matrix can be interpreted in a Markov-chaining process setting: $\pi_{n,m}^{(t,s)}$ is the probability that the underlying jumps from the state $S_n^t$ to the state $S_m^s$. Once computed, computing Optimal

Table 3
American Call SX5E Mat 3M. Spot 3064.03

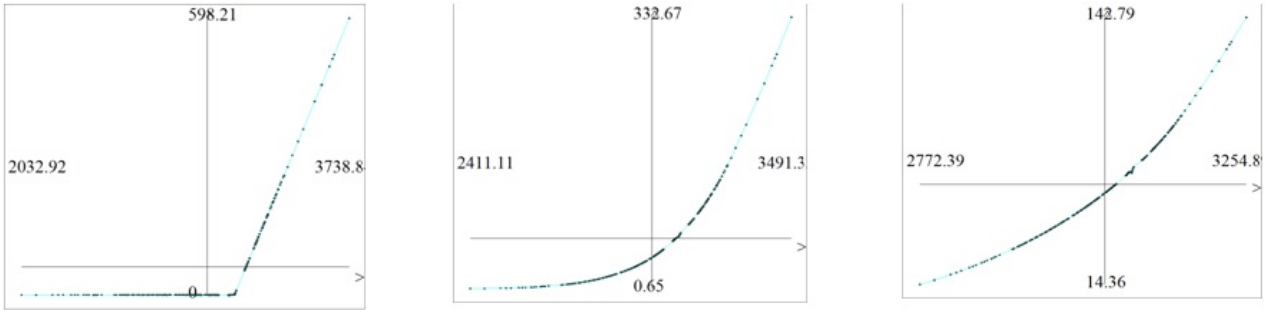
| Strike % | *Call values* | *N=16* | *N=64* | *N=256* | *N=1024* |
|---|---|---|---|---|---|
| 0.8 | 559.224 | 650.54 | 650.54 | 650.54 | 650.54 |
| 0.9 | 292.61 | 346.98 | 347.17 | 347.17 | 347.20 |
| 0.95 | 180.78 | 209.90 | 210.27 | 210.29 | 210.32 |
| 0.975 | 133.59 | 151.00 | 152.27 | 152.22 | 152.23 |
| 1 | 93.76 | 103.16 | 104.19 | 104.33 | 104.37 |
| 1.025 | 61.59 | 66.53 | 67.19 | 67.37 | 67.40 |
| 1.05 | 37.99 | 40.23 | 40.68 | 40.79 | 40.82 |
| 1.1 | 11.34 | 19.47 | 11.83 | 11.92 | 11.91 |
| 1.2 | 0.31 | 0.31 | 0.31 | 0.31 | 0.32 |
| comp. time | | 0.02 s. | 0.06 s. | 1.52 s. | 77 s. |

stopping problem over a portfolio is simply a matter of matrix-matrix multiplication. Indeed, we didn't notice any significant slow-down in computational time with portfolio of size $ND$ (i.e. $\sim 32000$ financial instruments for $N = 512$, $D = 64$), since matrix-matrix multiplication is usually a quite optimized code.

3.4.1. *Numerical experiments in the one-dimensional case*

Our first test is a one dimensional test, that consists in pricing american call options with maturity three months, and strike corresponding to data in table (1). We considered the data set of table (1) for calibration, so the quantile is the same as plotted in Figure (2). Prices are presented in table (3), corresponding to differents number of points $N$ - we recall that $N$ drives the precision of the computations. It has already been noted that transported schemes are very stable and accurate, see [5], as confirmed by this table. Computation time is also confirming the $N^3$ behavior of the algorithm. Note that for $D = 1$, we could have specialized the algorithm to get a complexity of order $\mathcal{O}(N)$, as has been done in [5]. However we prefer to use the same algorithm, independantly of the dimension $D$, for this review.

The qualitative properties of the solution are plotted in a serie of three figures, illustrating the retropagation steps of the american call having at-the-money strike 3064. The first one, Figure (3.4.1), represents the Payoff at maturity $T = 3M$. It plots more precisely the surface $\{S_n^{3M}, P(3M, S_n^{3M})\}_{n=1..256}$. The second one, figure (3.4.1), plots the solution at time $t = 1M$, $\{S_n^{1M}, P(1M, S_n^{1M})\}_{n=1..256}$. The third one plots the solution at time $t = 6D$, $\{S_n^{6D}, P(6D, S_n^{6D})\}_{n=1..256}$, that is the last computation time. Note that a tiny numerical artefact starts appearing at time 6D, very close to time origin, as operators are almost singulars there. We emphasize nonetheless that CoDeFi uses a numerical scheme that is unconditionally stable.




Payoff $T = 3M$ $T = 1M$ $T = 6D$

Figure 3. $P(T, S)$ at different retropopagation time, $N = 256$

Table 4
Price of Bermudean Best-of options MAT 10Y

| | D=1 | | D=4 | | D=16 | | D=64 | |
|---|---|---|---|---|---|---|---|---|
| MC N = 1048576 | 0.12578 | | 0.35941 | | 0.68796 | | 1.0166 | |
| N=32 | 0.128275 | (0 %, 0.14 s.) | 0.40047 | (8.1 %, 0.21s.) | 0.684418 | (0.46 %,2.3s.) | 0.927398 | (0.36 %, 152s.) |
| N=128 | 0.126521 | (0 %, 1.30 s.) | 0.38120 | (4.3 %, 2.49s.) | 0.711784 | (1.30 %, 14.4s.) | 0.985387 | (0.14 %, 304s.) |
| N=512 | 0.125921 | (0 %, 121 s.) | 0.39757 | (5.0 %, 148s.) | 0.699329 | (0.74 %, 280s.) | 1.01721 | (0.13 %, 1665s.) |

3.4.2. *Multi-dimensional experiments*

To set a first benchmark for high-dimensional problems, we chose a prototypal financial contract for which we can compute a theoretical price: consider the Best-of options presented in the previous section dedicated to calibration. Consider that the buyer has the option to exercize anually at date $1Y, .., 10Y$ (Bermudean style option). Since Best-Of payoffs are convex, if underlyings do not pay any dividend, such products should never be exercized. Hence prices correspond to the european options, estimated in table (2).

Table (4) presents prices of this product, computed with CoDeFi. Note that prices should coincide with those computed with calibrated Monte-Carlo sequences, presented in table (2). Prices are a little bit higher, and we precised the relative error, as well as an indicative computational time. This is indicating a mismatched early exercizing: we are using a numerical scheme that is convergent and stable in a weak $L^2$ norm, but that is not convex (except in one dimension). Therefore, the solution presents small oscillations that led to early exercizing. We refer to this error as the **convexity** error, that is more visible in lower dimensions.

3.5. *Conclusions*

In this note we presented a methodology to solve parabolic / hyperbolic equations set in high dimensions. Our perception is that, even if perfectible, this technology can already confidently compute risk measurements. There are numerous potential applications linked to the curse of dimensionality. We identified some of them in the insurance industry, but mainly in the financial sector. Above pricing and hedging, allowing to issue new financial products, more adapted to clients needs, or market abitraging, we are adressing risk measurements: indeed, the test realized in this paper shown that CoDeFi could compute accurate Basel-based regulatory measures as WRA (Weighted Ration Asset) or CVA (Counterparty Value Adjustment) on a single laptop, whereas farm of thousand of computers are used on a daily basis today with approximation methods. Finally, such a framework could be helpful in systemic risk measurements, that can be accurately modelized through high dimensional Kolmogorov equations like (4). This could notably be useful for developing tools dedicated to economical crisis supervision.

## References


[1] V. Bally, G. Pages, J. Printems,First order schemes in the numerical quantization method,Rapport de recherche INRIA N 4424.

[2] Brenier Y., Polar factorization and monotone re-arrangements of vector-valued functions, Comm. Pure Appl. Math. 44 (1991), 375–417.

[3] P.G. LeFloch, J.M., Mercier, Revisiting the method of characteristics via a convex hull algorithm, J. Comput. Phys. 298 (2015), 95–112.

[4] P.G. LeFloch, J.M., Mercier, A Framework to solve high-dimensional Kolmogorov equations. In preparation.

[5] J.M., Mercier, Optimally transported schemes and applications in mathematical finance, November 2008, available at http://www.crimere.com/blog/jean-marc/?p=336

[6] J.M., Mercier, A high-dimensional pricing framework for financial instruments valuation, April 2014, available at http://ssrn.com/abstract=2432019.

[7] Longstaff, Francis A., Schwartz, Eduardo S, Valuing American Options by Simulation: A Simple Least-Squares Approach, The Review of Financial Studies, 14(1):113-147, 2001

[8] M. Matsumoto and T. Nishimura, Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Trans. on Modeling and Computer Simulation Vol. 8, No. 1, January pp.3-30 (1998) DOI:10.1145/272991.272995

[9] Matache A.M., Nitsche P.A. and Schwab C., Wavelet Galekin Pricing of American Options on Levy Driven Assets, Research reports N 2003/06, (2003).

[10] Sobol,I.M. (1967), Distribution of points in a cube and approximate evaluation of integrals, Zh. Vych. Mat. Mat. Fiz. 7: 784802 (in Russian); U.S.S.R Comput. Maths.

[11] C.Villani, Optimal transport, old and new, Springer Verlag, 2006.