Available online at http://scik.org
Eng. Math. Lett. 2017, 2017:2
ISSN: 2049-9337

IMPLEMENTING SOME MATHEMATICAL OPERATORS FOR A CONTINUOUS-IN-TIME FINANCIAL MODEL

TARIK CHAKKOUR

Univ. Bretagne-Sud, UMR 6205, LMBA, F-56000 Vannes, France

Abstract. This paper considers the development of some mathematical operators as convolution and primitive for continuous-in-time financial model. This development is given in form of API (Application Programming Interface) with showing concept of its computation. The model is based on using measures and fields. The work we report here addresses the fundamental issue of how measures and fields are implemented for the software. The originality of this API lie in the fact that it will be used by the company MGDIS.

Keywords: API; convolution; primitive; discretization; integration; software tool.

2010 AMS Subject Classification: 34A45, 65M22, 68N30.

1. Introduction

Time is the central element that influence financial economic behavior. The continuous-in-time financial model constitutes a powerful tool for studying the development of continuous-in-time methods in finance. We refer to papers [1, 2], which are dealing with continuous-in-time financial model. These papers develop the mathematics and economic theory of finance from the perspective of a model in which agents can revise their decisions continuously in time. At the same time, we have seen an explosion in the use of algorithms for computation methods to

Received November 29, 2016

implement continuous-time models. The covered methods include convolution and primitive has been one of the most effective and widely-used of these methods. They began to be studied and applied systematically in various branches of modern science as in finance. We refer to [3] that presents an approach for implementing continuous-time adaptive recursive filters for convolution operator.

Within this paper, SOFI [4] is a software tool marketed by the company MGDIS. It is designed to the public institutions such local communities to set out multiyear budgets. SOFI is based on a discrete financial modeling. Currently, the mathematical objects involved in SOFI are suites and series. The discrete model generates outcomes in the form of tables. We showed in previous work [5] the default of this discrete model. We build a new model with using an other paradigm in [5]. This new model is based on continuous-in-time model and uses the mathematical tools such convolution and integration to describe loan scheme, reimbursement scheme and interest payment scheme. In [6] we have shown some results about improving one of the continuous-in-time financial models built in paper [5]. We use in [6] a mathematical framework to discuss an inverse problem of the continuous-in-time model.

This article describes implementing the continuous-in-time financial model. Mainly, we focus on concept of computation in API. This API is to be integrated in SOFI in order to produce the continuous software, and is restricted to certain measures and fields. The purpose of computing integration of measures over a time interval is to compute loan scheme, reimbursement scheme, etc; and the purpose of computing evaluation of fields at an instant is to compute current debt amount, where current debt field is a function that, at any time t, gives the capital amount still to be repaid.

Since some measures and fields could not been implemented continuously, we discretize them. Indeed, some computations in API need discretization. Next, we use these discrete values for obtaining the continuous values. The original motivation for this paper comes from a desire to understand the concept of computation in API with establishing mathematical relation between discrete measure and integrated measure. In addition, convolution and primitive operators are fundamental operations in the model. We use convolution in order to compute capital repayment measure with the Fast Fourier Transform method. We use primitive to compute current debt field at an instant t with accumulating measures between initial time and time t. The primitive of measure is defined as a field in spite of it is undefined in the Radon measure space. In this work, we describe how we impelement and check these operators.

The rest of this paper contains three sections. The first one introduces time steps that are involved in the models in order to show concept of computation in API. In the second we review numeric choices linked to API, where we define a field as continuous function by superior value. The last one shows implementation details about convolution and primitive.

2. Concept of computation in API

This section is devoted to explain time steps that are involved in the model and the relations between them. We give the time scales to integrate measure over interval which are shown in Figure 1. We introduce T_{min} which is the time scale below which nothing coming from the model will be observed. To be more precise, we say that a measure $\tilde{m}$ is observed over time interval $[t_1, t_2]$ if

$$\int_{t_1}^{t_2} \tilde{m}, \tag{2.1}$$

is computed. And, we will always, choose times t_1 and t_2 such that $t_2 - t_1 > T_{\text{min}}$. In order to observe models, we need an observation step T_{obs} which is strictly superior to minimal observation step T_{min}

$$T_{\text{obs}} > T_{\text{min}}. \tag{2.2}$$

We define the discrete step T_{dM} as a smaller step than step T_{min} to discretize measures:

$$T_{\text{dM}} \leq T_{\text{min}}. \tag{2.3}$$

For instance, we are setting discrete step T_{dM} by following equality:

$$T_{\text{dM}} = \frac{T_{\text{min}}}{20}. \tag{2.4}$$

Observation step T_{obs} is partitioned into n_D discrete step T_{dM} defined by:

$$n_D = \left[\frac{T_{\text{obs}}}{T_{\text{dM}}}\right]. \tag{2.5}$$

A field is evaluated between inferior value a and superior value b with discrete step T_{dF} satisfying:

$$T_{\text{dF}} < b - a. \tag{2.6}$$

Since measures and fields compose API, they are shared in two levels which are shown in Figure 2. First is high level and is created for business reasons. The computation in high level is designed to the SOFI users. Second is low level which is only used by the high level. The computation in low level is designed to the high level users. Notice that low level doesn't use its high. We say that high level implements its low. High and low levels contain non-discrete measures, non-discrete fields defined on $\mathbb{R}$, discrete measures and discrete fields. Some computations in high level need discretization. For instance, if we want to discretize a measure in high level, we create its copy in low level. Then we discretize it in order to rise up its values to high level.

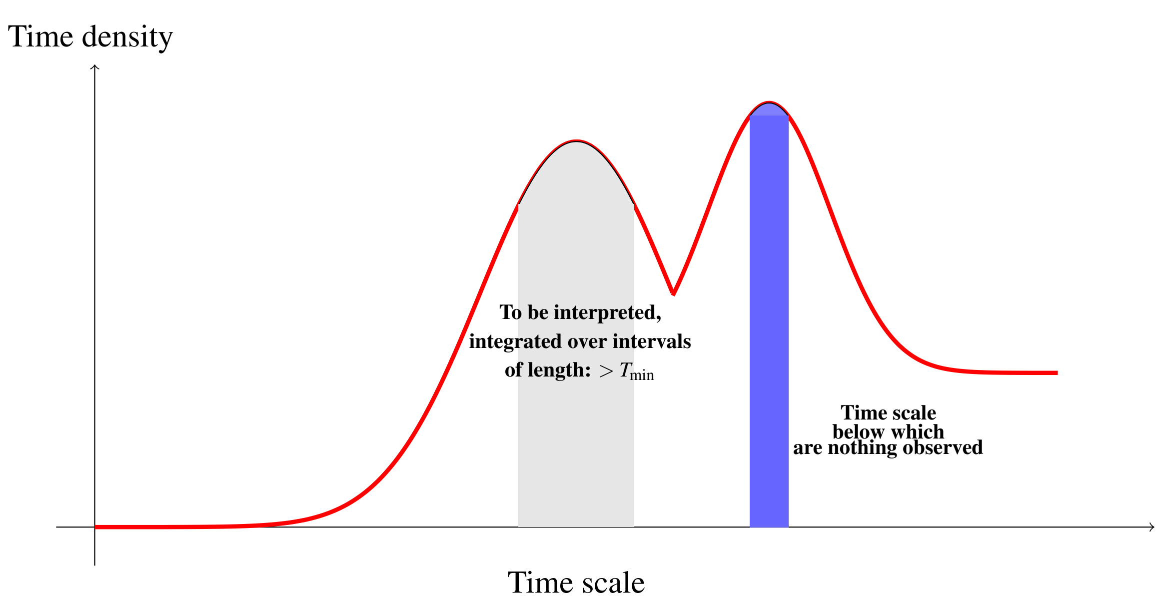

FIGURE 1. Different time steps.

The aim here is to explain how measures are integrated and how fields are evaluated. A non-discrete measure in low level is integrated between inferior bound a and superior bound b with minimal observation step T_{min} and observation step T_{obs}. It follows that discrete step T_{dM} is computed with relation (2.4). Whereas in high level it is integrated between inferior bounds a and superior bound b. A non-discrete field in low level is evaluated between inferior value a and superior value b with discrete step T_{dF}. Yet, its evaluation in high level is done only between inferior value a and superior value b.

The parallelism of discrete measures and fields in low level is based on the concept of a task. Tasks provide much benefits: more efficient computation and robustness API. Precisely, the Task Parallel Library [7] is used to entail execution and development speed. It is shown in [8] that this library makes it easy to take advantage of potential parallelism in a program. It relies heavily on generics and delegate expressions. Paper [9] shows several strategies that can be applied in large-scale discrete distribution clustering tasks.

In what follows, we build the unidimensional mesh called DAS (DiscretizedAxeSegment presented in Figure 3) for two reasons. First is to better structure the low level. Second is to compute discrete convolution due to the impossibility for computing it with variable step using the Fast Fourier Transform. Mesh DAS associated to discrete step T_{dM} is defined by a set of points $(x_k)_{k\in\mathbb{Z}}$ that are its multiple

$$\text{DAS}_{\text{T}_{\text{dM}}} = \{x_k = k \times T_{\text{dM}}, k \in \mathbb{Z}\}. \tag{2.7}$$

Integration of measure m_d in low level between inferior bound a and superior bound b with minimal observation step T_{min} returns its integration between new inferior bound x_a and new superior bound x_b with discrete step T_{dM}, where

$$x_a = n_a \times T_{\text{dM}}, \tag{2.8}$$

such that:

$$n_a = \left[\frac{a}{T_{\text{dM}}}\right], \tag{2.9}$$

and where

$$x_b = n_b \times T_{\mathrm{dM}}, \tag{2.10}$$

such that:

$$n_b = \begin{cases} \frac{b}{T_{\mathrm{dM}}} & \text{if } T_{\mathrm{dM}} \text{ is divisible by } b, \\ \left[\dfrac{b}{T_{\mathrm{dM}}}\right] + 1 & \text{else.} \end{cases} \tag{2.11}$$

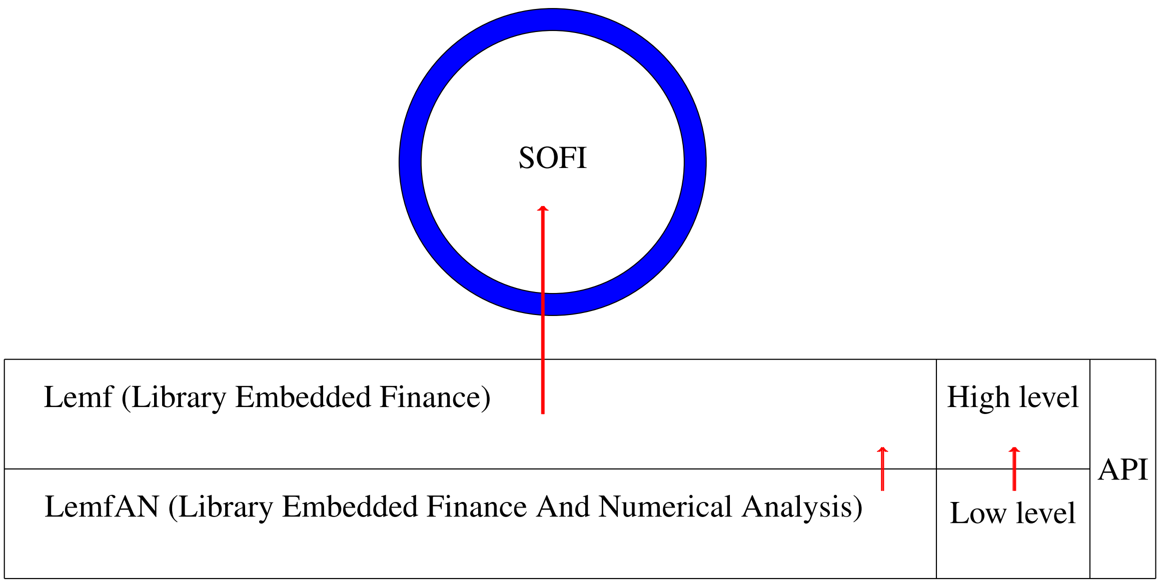

FIGURE 2. API composition.

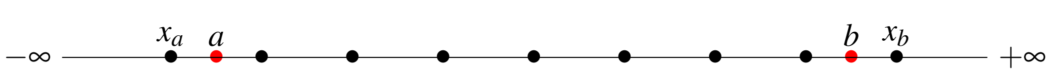

FIGURE 3. Mesh DAS defined on $\mathbb{R}$.

Interval $[x_a, x_b]$ is partitioned into $\mathcal{N}_a^b$ subintervals of equal length, where $\mathcal{N}_a^b$ is given by:

$$\mathcal{N}_a^b = n_b - n_a, \tag{2.12}$$

where integers n_a and n_b are defined respectively in relations (2.9) and (2.11). Now, we will define a discrete measure of measure m_d. For any integer j from 1 to $\mathcal{N}_a^b$, we call $(n_a + j - 1)^{\mathrm{nd}}$

discrete value, the integration of measure m_d between inferior bound $(n_a+j-1)\times T_{\text{dM}}$ and superior bound $(n_a+j)\times T_{\text{dM}}$ given by following equality:

$$\forall j \in [\![1;\mathcal{N}_a^b]\!], m_d(n_a+j-1) = \int_{(n_a+j-1)\times T_{\text{dM}}}^{(n_a+j)\times T_{\text{dM}}} m_d. \tag{2.13}$$

For any integer i from 1 to $\left[\dfrac{\mathcal{N}_a^b}{n_D}\right]$, we define quantity $m_d^{\text{obs}}(i)$ as observed measure over time interval that its length is T_{obs} between inferior bound $n_a\times T_{\text{dM}}+(i-1)\times T_{\text{obs}}$ and superior bound $n_a\times T_{\text{dM}}+i\times T_{\text{obs}}$. Formally, $m_d^{\text{obs}}(i)$ is defined as:

$$\forall i \in \left[\!\left[1;\left[\frac{\mathcal{N}_a^b}{n_D}\right]\right]\!\right], m_d^{\text{obs}}(i) = \int_{n_a\times T_{\text{dM}}+(i-1)\times T_{\text{obs}}}^{n_a\times T_{\text{dM}}+i\times T_{\text{obs}}} m_d, \tag{2.14}$$

which is decomposed with Chasles relation as:

$$\forall i \in \left[\!\left[1;\left[\frac{\mathcal{N}_a^b}{n_D}\right]\right]\!\right], m_d^{\text{obs}}(i) = \sum_{k=1}^{n_D} \int_{(n_a+k-1)\times T_{\text{dM}}+(i-1)\times T_{\text{obs}}}^{(n_a+k-n_D)\times T_{\text{dM}}+i\times T_{\text{obs}}} m_d. \tag{2.15}$$

Because of (2.5) and of the fact that $l=k+(i-1)\times n_D$, relation (2.15) implies that:

$$\forall i \in \left[\!\left[1;\left[\frac{\mathcal{N}_a^b}{n_D}\right]\right]\!\right], m_d^{\text{obs}}(i) = \sum_{l=1+(i-1)\times n_D}^{i\times n_D} \int_{(n_a+l-1)\times T_{\text{dM}}}^{(n_a+l)\times T_{\text{dM}}} m_d. \tag{2.16}$$

From this and according to (2.16), we conclude that observed value $m_d^{\text{obs}}(i)$ is a sum of values $m_d(n_a+l-1)$ for integer l from $1+(i-1)\times n_D$ to $i\times n_D$

$$\forall i \in \left[\!\left[1;\left[\frac{\mathcal{N}_a^b}{n_D}\right]\right]\!\right], m_d^{\text{obs}}(i) = \sum_{l=1+(i-1)\times n_D}^{i\times n_D} m_d(n_a+l-1). \tag{2.17}$$

There are two situations for computing observed values. If $\mathcal{N}_a^b$ is divisible by n_D, then $\left[\dfrac{\mathcal{N}_a^b}{n_D}\right]$ observed values are computed with relation (2.17). Else, $\left[\dfrac{\mathcal{N}_a^b}{n_D}\right]$ observed values are computed with relation (2.17) such that the observed value $m_d^{\text{obs}}\left(\left[\dfrac{\mathcal{N}_a^b}{n_D}\right]+1\right)$ is computed with following relation:

$$m_d^{\text{obs}}\left(\left[\frac{\mathcal{N}_a^b}{n_D}\right]+1\right)=\sum_{k=n_D\times\left[\frac{\mathcal{N}_a^b}{n_D}\right]+1}^{\mathcal{N}_a^b} m_d(n_a+k-1). \tag{2.18}$$

3. Numeric choices in API

We are concerned in this section about implementation choices providing for great flexibility in API. Given a continuous function ϕ, the Dirac measure δ_p at point p acts on the function ϕ. The value of this action is $\phi(p)$. The purpose is to maintain this action in API. For that, we will explain the numeric choices that we have made to achieve it due to the difficulty for describing the dual of vector space of continuous piecewise function with a finite number of pieces, continuous with superior values. For instance, the action of Dirac measure δ_p on fields $\mathbb{1}_{]-\infty,p]}$ and $\mathbb{1}_{[p,+\infty[}$ is undefined. Indeed, they integrals with respect to Dirac measure δ_p could not be computed. Formally, following integrals

$$\int_{-\infty}^{+\infty}\mathbb{1}_{]-\infty,p]}\,d\delta_p(x),\int_{-\infty}^{+\infty}\mathbb{1}_{[p,+\infty[}\,d\delta_p(x), \tag{3.1}$$

are undefined. In order to set the value of this action consistently, we make a choice on Dirac measure δ_p defined by:

$$<\delta_p,\phi>=\lim_{x\to p^+}\phi(x). \tag{3.2}$$

Consider a continuous function g with integral equals 1 over $\mathbb{R}$. To justify relation (3.2), we may restrict to support of function g defining function g_ε which approaches Dirac measure δ_p, and is defined as:

$$g_\varepsilon(x)=\frac{1}{\varepsilon}g\left(\frac{x-p}{\varepsilon}+\varepsilon p\right). \tag{3.3}$$

Dirac measure δ_p can be expressed as a limit of function g_ε

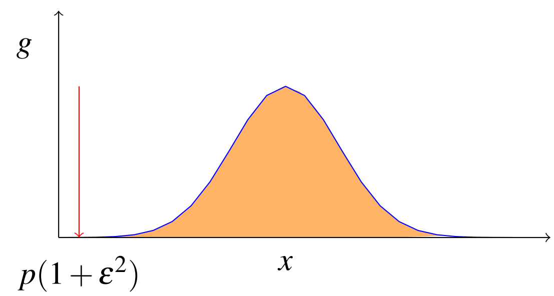

FIGURE 4. Restriction to support of function g defining function g_ε in relation (3.3).

$$\lim_{\varepsilon \to 0^+} g_\varepsilon = \delta_p. \tag{3.4}$$

To obtain relation (3.2), we require the following inclusion:

$$\mathrm{Supp}(g_\varepsilon) \subset]p, +\infty[, \tag{3.5}$$

because of:

$$\mathrm{Supp}(g) \subset]p, +\infty[. \tag{3.6}$$

Relation (3.6) provides restriction to support of function g illustrated in Figure 4 due to following equivalence:

$$\forall \varepsilon \in \mathbb{R}_+^*, x > p \iff \frac{1}{\varepsilon} \times \left(\frac{x-p}{\varepsilon} + \varepsilon p \right) > p. \tag{3.7}$$

4. Convolution and accumulation

This section covers the implementation of convolution and of primitive operators. We refer to papers [10, 11, 12], which are dealing with how convolution can be efficiently computed by FFT (the Fast Fourier Transform). For example, algorithms based on explicit computation and on FFT are described in [10]. Paper [11] presents a more efficient computation of the convolution between a compressed text and an uncompressed pattern. Schaller & Temnov estimates in [12] numerical errors of discrete FFT. In the model, Loan Measure $\tilde{\kappa}_E$ is defined such that the amount borrowed between times t_1 and t_2 is:

$$\int_{t_1}^{t_2} \tilde{\kappa}_E, \tag{4.1}$$

and Repayment Measure $\tilde{\rho}_{\mathscr{K}}$ is defined such that the amount borrowed between times t_1 and t_2 is:

$$\int_{t_1}^{t_2} \tilde{\rho}_{\mathscr{K}}. \tag{4.2}$$

Loan Measure $\tilde{\kappa}_E$ and Capital Repayment Measure $\tilde{\rho}_{\mathscr{K}}$ are connected by a convolution operator. It is required to implement it in order to compute repayment amount. Then the discrete convolution may be evaluated with the aid of FFT method. By the Fourier convolution theorem, the discrete Fourier transform of $\tilde{\kappa}_E \star \tilde{\gamma}$ may be computed as

$$\mathscr{F}(\tilde{\rho}_{\mathscr{K}}) = \mathscr{F}(\tilde{\kappa}_E \star \tilde{\gamma}) = \mathscr{F}(\tilde{\kappa}_E) \bullet \mathscr{F}(\tilde{\gamma}), \tag{4.3}$$

where the Repayment Pattern Measure $\tilde{\gamma}$ expresses the way an amount 1 borrowed at $t = 0$ is repaid and where $\bullet$ denotes component-wise multiplication. Quantities $\mathscr{F}(\tilde{\kappa}_E)$ and $\mathscr{F}(\tilde{\gamma})$ define discrete Fourier transforms of $\tilde{\kappa}_E$ and of $\tilde{\gamma}$, respectively. The computation of discrete convolution $(\tilde{\kappa}_E \star \tilde{\gamma}(n_e + j - 1))_{1 \le j \le \mathscr{N}_e^f}$ with discrete measures $(\tilde{\kappa}_E(n_a + j - 1))_{1 \le j \le \mathscr{N}_a^b}$ and $(\tilde{\gamma}(n_c + j - 1))_{1 \le j \le \mathscr{N}_c^d}$ between points x_e and x_f of universel mesh $\mathrm{DAS}_{\mathrm{T_{dM}}}$ is summarized as follows:

- Determine the convex hull of the support of discrete measure $(\tilde{\kappa}_E(n_a + j - 1))_{1 \le j \le \mathscr{N}_a^b}$ supposed to be interval $[x_{a_1}, x_{b_1}]$;
- Determine the convex hull of the support of discrete measure $(\tilde{\gamma}(n_c + j - 1))_{1 \le j \le \mathscr{N}_c^d}$ supposed to be interval $[x_{c_1}, x_{d_1}]$;
- Complete by zero discrete measures $(\tilde{\kappa}_E(n_a + j - 1))_{1 \le j \le \mathscr{N}_a^b}$ and $(\tilde{\gamma}(n_c + j - 1))_{1 \le j \le \mathscr{N}_c^d}$ such that they have N values, where N is power of 2 and is smallest value satisfying $N \ge \mathscr{N}_a^b + \mathscr{N}_c^d$. Then, $(\tilde{\kappa}_E^1(n_a + j - 1))_{1 \le j \le N}$ and $(\tilde{\gamma}^1(n_c + j - 1))_{1 \le j \le N}$ are called the discrete values extended by zero;

- Compute discrete measures $(x(n_a+j-1))_{1\leq j\leq N}$ and $(y(n_c+j-1))_{1\leq j\leq N}$ by Fourier transform of discrete measures $(\tilde{\kappa}_E^1(n_a+j-1))_{1\leq j\leq N}$ and $(\tilde{\gamma}^1(n_c+j-1))_{1\leq j\leq N}$, respectively;
- Compute vector $z(j-1)_{1\leq j\leq N}$ defined by element-wise multiplication of $(x(n_a+j-1))_{1\leq j\leq N}$ by $(y(n_c+j-1))_{1\leq j\leq N}$;
- Compute vector $(h(j-1))_{1\leq j\leq N}$ defined by inverse Fourier transform of $(z(j-1))_{1\leq j\leq N}$;
- Build tabulated measure $\tilde{m}_{\text{Tabulated}}$ between inferior value $x_{a_1}+x_{c_1}$ and superior value $x_{b_1}+x_{d_1}$ with a set of first $\mathscr{N}_a^b+\mathscr{N}_c^d$ values of h;
- Discretize tabulated measure $\tilde{m}_{\text{Tabulated}}$ between points x_e et x_f with discrete step T_{dM} to get discrete values $(\tilde{\kappa}_E\star\tilde{\gamma}(n_e+j-1))_{1\leq j\leq\mathscr{N}_e^f}$.

The integration of discrete measure $(\tilde{\kappa}_E\star\tilde{\gamma}(n_e+j-1))_{1\leq j\leq\mathscr{N}_e^f}$ in high level between inferior bound e and superior bound f is the sum of its values given by:

$$\sum_{j=1}^{\mathscr{N}_e^f}\tilde{\kappa}_E\star\tilde{\gamma}(n_e+j-1).$$

After defining convolution operator, we describe how the accumulation of measure is defined in API. The Current Debt Field $\mathscr{K}_{RD}$ is related to Loan Measure $\tilde{\kappa}_E$ and Repayment Measure $\tilde{\rho}_{\mathscr{K}}$ by the following Ordinary Differential Equation:

$$\frac{d\mathscr{K}_{RD}}{dt}=\kappa_E(t)-\rho_{\mathscr{K}}(t). \tag{4.4}$$

The solution of this ODE is expressed:

$$\mathscr{K}_{RD}(t)=\mathscr{K}_{RD}(t_{\mathtt{I}})+\int_{t_{\mathtt{I}}}^{t}\tilde{\kappa}_E-\int_{t_{\mathtt{I}}}^{t}\tilde{\rho}_{\mathscr{K}}. \tag{4.5}$$

To compute the Current Debt Field $\mathscr{K}_{RD}$ at an instant t, we define the method that is computing the primitive of a measure. This method is based on numerical approach which consists in accumulating a discrete measure in order to approximate it by a field. The primitive of measure m_d in low level that is zero at point x_c is a field F_d. Its discretization between inferior value x_a and superior value x_b with discrete step T_{dF} is defined by discrete field $\left(F_d^{\text{D}}(n_a+k-1)\right)_{1\leq k\leq \mathcal{N}_a^b+1}$ given by:

$$\forall k \in [\![1;\mathcal{N}_a^{\,b}+1]\!], F_d^{\text{D}}(n_a+k-1) = \int_{x_c}^{y_k} m_d, \tag{4.6}$$

where points $(y_k)_{1\leq k\leq \mathcal{N}_a^b+1}$ are defined as:

$$\forall k \in [\![1;\mathcal{N}_a^{\,b}+1]\!], y_k = x_a + (k-1)\times T_{\text{dF}}. \tag{4.7}$$

We distinguish three cases of computing discrete field $\left(F_d^{\text{D}}(n_a+k-1)\right)_{1\leq k\leq \mathcal{N}_a^b+1}$:

<u>First case $x_c < x_a$</u>

Measure m_d is discretized between points x_c and x_b with discrete step T_{dF} to compute discrete measure $\left(m_d(n_c+j-1)\right)_{1\leq j\leq \mathcal{N}_c^b}$. The integral defined in relation (4.6) is decomposed with Chasles relation to get:

$$\forall k \in [\![1;\mathcal{N}_a^{\,b}+1]\!], F_d^{\text{D}}(n_a+k-1) = \sum_{j=1}^{\mathcal{N}_c^{\,a}} \int_{x_c+(j-1)\times T_{\text{dF}}}^{x_c+j\times T_{\text{dF}}} m_d + \sum_{j=1}^{k-1} \int_{x_a+(j-1)\times T_{\text{dF}}}^{x_a+j\times T_{\text{dF}}} m_d. \tag{4.8}$$

Replacing x_a by $x_c + \mathcal{N}_c^{\,a} \times T_{\text{dF}}$ in relation (4.8), we obtain the following equality:

$$\forall k \in [\![1;\mathcal{N}_a^{\,b}+1]\!], F_d^{\text{D}}(n_a+k-1) = \sum_{j=1}^{\mathcal{N}_c^{\,a}} \int_{x_c+(j-1)\times T_{\text{dF}}}^{x_c+j\times T_{\text{dF}}} m_d + \sum_{j=1}^{k-1} \int_{x_c+(j-1+\mathcal{N}_c^a)\times T_{\text{dF}}}^{x_c+(j+\mathcal{N}_c^a)\times T_{\text{dF}}} m_d. \tag{4.9}$$

From this and using relation (2.13) which defines discrete measure, we get:

$$\forall k \in [\![1;\mathcal{N}_a^{\,b}+1]\!], F_d^{\text{D}}(n_a+k-1) = \sum_{j=1}^{\mathcal{N}_c^{\,a}} m_d(n_c+j-1) + \sum_{j=1+\mathcal{N}_c^a}^{k-1+\mathcal{N}_c^a} m_d(n_c+j-1). \tag{4.10}$$

<u>Second case $x_c > x_b$</u>

Measure m_d is discretized between points x_a and x_c with discrete step T_{dF} to compute discrete measure $(m_d(n_a+j-1))_{1\leq j\leq \mathcal{N}_a^c}$. It follows that Chasles relation applied to relation (4.6) gives:

$$\forall k \in [\![1;\mathcal{N}_a^b+1]\!], F_d^{\mathrm{D}}(n_a+k-1) = -\sum_{j=k}^{\mathcal{N}_a^c} \int_{x_a+(j-1)\times T_{\mathrm{dF}}}^{x_a+j\times T_{\mathrm{dF}}} m_d, \tag{4.11}$$

which is reduced to following equality:

$$\forall k \in [\![1;\mathcal{N}_a^b+1]\!], F_d^{\mathrm{D}}(n_a+k-1) = -\sum_{j=k}^{\mathcal{N}_a^c} m_d(n_a+j-1). \tag{4.12}$$

<u>Third case $x_a \leq x_c \leq x_b$</u>

Determining integer $L \in [\![1;\mathcal{N}_a^b]\!]$ satisfying following inequalities:

$$y_L < x_c \leq y_{L+1}. \tag{4.13}$$

Since $x_c > y_k$ for each integer k from 1 to L, the result of second case implies that:

$$\forall k \in [\![1;L]\!], F_d^{\mathrm{D}}(n_a+k-1) = -\sum_{j=k}^{\mathcal{N}_a^c} m_d(n_a+j-1). \tag{4.14}$$

Replacing x_c by $x_a+\mathcal{N}_a^c \times T_{\mathrm{dF}}$ in relation (4.6) and employing Chasles relation, we obtain:

$$\forall k \in [\![L+1;\mathcal{N}_a^b+1]\!], F_d^{\mathrm{D}}(n_a+k-1) = \sum_{j=1+\mathcal{N}_a^c}^{k-1} m_d(n_a+j-1). \tag{4.15}$$

Conflict of Interests

The authors declare that there is no conflict of interests.

REFERENCES

[1] R.C. Merton, Theory of finance from the perspective of continuous time, Journal of Financial and Quantitative Analysis, 10(04) (1975), 659-674.

[2] R.C. Merton, P.A. Samuelson, Continuous-Time Finance, Blackwell Boston, 1st edition, (1992).

[3] D. Johns, W.M. Snelgrove, A.S. Sedra, Continuous-time lms adaptive recursive filters. IEEE Transactions on Circuits and Systems, 38(7) (1991), 769-778.

[4] SOFI, https://www.mgdis.fr/index.phppage=display_doma&class=article&object=sol_sofi_programmation_financiere&method=display_full&refo=001009, Accessed: 2016-08-16.

[5] E. Frénod, T. Chakkour, A continuous-in-time financial model. Mathematimathcal Finance Letters, 2016 (2016), Article ID 2.

[6] T. Chakkour, E. Frénod, Inverse problem and concentration method of a continuous-in-time financial model, International Journal of Financial Engineering, 3(2) (2016), 1650016-1650036.

[7] Task parallel library [online], https://msdn.microsoft.com/fr-fr/library/dd537609%28v=vs.110%29.aspx, Accessed: 2016-06-30.

[8] D. Leijen, W. Schulte, S. Burckhardt, The design of a task parallel library, Acm Sigplan Notices, 44(10) (2009), 227-242.

[9] Y. Zhang, J.Z. Wang, J. Li, Parallel massive clustering of discrete distributions, ACM Trans. Multimedia Comput. Commun. Appl., 1 (1) (2015), Article ID 1.

[10] T. Hearn, L. Reichel, Fast computation of convolution operations via low-rank approximation, Applied Numerical Mathematics, 75 (2014), 136-153.

[11] T. Tanaka, I. Tomohiro, S. Inenaga, H. Bannai, M. Takeda, Computing convolution on grammar-compressed text, In Data Compression Conference (DCC), IEEE, (2013), 451-460.

[12] P. Schaller, G. Temnov, Efficient and precise computation of convolutions: applying fft to heavy tailed distributions Computational Methods in Applied Mathematics Comput, Methods Appl. Math., 8(2) (2008), 187-200.