VALUATION OF STRUCTURED FINANCIAL PRODUCTS BY ADAPTIVE MULTIWAVELET METHODS IN HIGH DIMENSIONS

RÜDIGER KIESEL, ANDREAS RUPP, AND KARSTEN URBAN

Abstract. We introduce a new numerical approach to value structured financial products.

These financial products typically feature a large number of underlying assets and require the explicit modelling of the dependence structure of these assets. We follow the approach of Kraft and Steffensen (2006,[26]), who explicitly describe the possible value combinations of the assets via a Markov chain with a portfolio state space.

As the number of states increases exponentially with the number of assets in the portfolio, this model so far has been – despite its theoretical appeal – not computational tractable.

The price of a structured financial product in this model is determined by a coupled system of parabolic PDEs, describing the value of the portfolio for each state of the Markov chain depending on the time and macroeconomic state variables. A typical portfolio of n assets leads to a system of $N = 2^n$ coupled parabolic partial differential equations. It is shown that this high number of PDEs can be solved by combining an adaptive multiwavelet method with the Hierarchical Tucker Format. We present numerical results for $n = 128$.

1. Introduction

The inadequate pricing of Asset-backed securities (ABS) and in particular Collateralized Debt Obligations (CDOs), on which we focus, is widely viewed as a main trigger of the financial crisis that started in 2007, [6, 17].

The lack of adequate mathematical models to capture the (dependency) risk structure, [23], of these assets is consistently identified as the main reason for the inaccurate pricing. Due to the complexity of a CDO portfolio, which arises from the high number of possible default combinations, drastic simplifications of the underlying portfolio structure had to be made in order to compute a price, [7, 9, 43].

We consider the CDO model of [26] where the value of a CDO portfolio is determined by a system of coupled parabolic PDEs, each PDE describing the portfolio value for a specific default situation. These default situations are characterized by a discrete Markov chain, where each state in the Markov chain stands for a default state of the portfolio. Therefore, for a portfolio of n assets, there are $N = 2^n$ possible combinations of defaults and, therefore, 2^n states in the Markov chain. It will later turn out to be convenient to label the states in the index set $\mathcal{N} := \{0, \dots, N-1\}$. The value of the CDO portfolio in [26] is described by the function $\mathbf{u}(t,y) = (u^0(t,y), \dots, u^{N-1}(t,y))^T$ that satisfies the partial differential equation for all $t \in (0,T)$ ($T > 0$ being the maturity) and all $y \in \Omega \subset \mathbb{R}^M$. The y variables are used to incorporate M economic market factors which describe the state of the economy.

$$
\begin{aligned}
u^j_t(t,y) &= -\frac{1}{2}\nabla\cdot(\mathbf{B}(t)\nabla u^j(t,y)) - \boldsymbol{\alpha}^T(t)\nabla\, u^j(t,y) + r(t,y)u^j(t,y) \\
&\quad - \sum_{k\in\mathcal{N}\setminus\{j\}} q^{j,k}(t,y)(a^{j,k}(t,y) + u^k(t,y) - u^j(t,y)) - c^j(t,y), \qquad \text{(1.1a)}
\end{aligned}
$$

$$\mathbf{u}(t,y) = 0,\ t\in(0,T),\ \ y\in\partial\Omega, \qquad \text{(1.1b)}$$

$$\mathbf{u}(T,y) = (u^0_T(y),\dots,u^{N-1}_T(y))^T,\ y\in\Omega, \qquad \text{(1.1c)}$$

for all $j\in\mathcal{N}$. The differential operator ∇ is to be understood w.r.t. the variable y. Often the bounded domain Ω arises from localizing the problem from $\mathbb{R}^M$ to a bounded domain by truncation.

This is a generalized Black-Scholes PDEs with a linear coupling, homogeneous Dirichlet boundary conditions (1.1b) in y (possibly after localization) and terminal condition (1.1c). The remaining parameters can be interpreted as follows:

- The space variables $y\in\Omega\subset\mathbb{R}^M$ describe the current market situation by means of variables which describe the market influence on the CDO portfolio. This could be for example interest rates, foreign exchange rates, macroeconomic factors and other factors depending on the composition of the portfolio. These space variables are modelled via a market process $dY(t) = \boldsymbol{\alpha}(t)dt + \boldsymbol{\beta}(t)dW(t)$, where $W(t)$ is a M-dimensional standard Brownian motion, the drift $\boldsymbol{\alpha}(t)$ is a M-dimensional vector and the volatility $\boldsymbol{\beta}(t)\in\mathbb{R}^{M\times M}$. Then, we abbreviate $\mathbf{B}(t) := \boldsymbol{\beta}(t)\boldsymbol{\beta}(t)^T$. By normalization, we may assume w.l.o.g. $\Omega = [0,1]^M$.
- $\mathcal{N}$ is the state space of a Markov chain, where each state is a possible combination of defaults of the underlying portfolio.
- The function $r(t,y)$ describes the relevant market interest rate.
- The parameters $q^{j,k}\geq 0$ are the transition intensities, which is the instantaneous change in the transition probabilities, from state j into state k, where $j,k\in\mathcal{N}$. Moreover, for any state j, all intensities sum up to zero, i.e., $q^{j,j} := -\sum_{k\in\mathcal{N}\setminus\{j\}} q^{j,k}$. The default probability is assumed to increase over time, see [26].
- The payments $c^j(t,y)$, $j\in\mathcal{N}$, made by the CDO are assumed to be continuous in time.
- The recovery payment, i.e., the distribution of the remaining funds of the defaulted firm, is denoted by $a^{j,k}(t,y)$. It depends on the transition from state j to state k, which means on the defaulted firm.
- Final payments at maturity can also be included. They also depend on the state and the current market situation and are denoted by $u^j_T(y)$.

All together, (1.1) is a system of $N = 2^n$ coupled time-dependent parabolic PDEs each in dimension M. The difficulty of this pricing approach is primarily the high number $N = 2^n$ of states in the Markov chain and, hence, the high number of coupled partial differential equations. In the following it will be shown, that under reasonable conditions, the high dimensionality resulting from the Markov chain can be separated as a time dependent factor from the actual solution of the partial differential equation. This allows to represent the system of coupled partial differential equations in variational form as the variational formulation of a high dimensional parabolic partial differential equation. We propose to use orthogonal multiwavelet bases to develop an equivalent discrete but infinite-dimensional system. This particular choice allows to write the system as a tensor product, which in turns leads to decoupling the Markov chain ingredients from the market parameters, i.e., the high dimensionality is separated from the integrals of the test and trial spaces. The hierarchical Tucker Format (HTF), is then applied to this tensor structure. To numerically approximate a solution for this system, multiwavelets ensure small condition numbers regardless of the dimension of the process. Moreover, this choice allows for asymptotically optimal adaptive schemes, see e.g. [24].

In the context of wavelet approximations of solutions of partial differential equations, the term "high dimensional" commonly refers to the dimension M of the space variable, say $M \geq 5$. In our problem at hand, we also have a huge number $N = 2^n$ of coupled equations. As already mentioned, we will show that we can separate both ingredients, namely the Markov chain state space $\mathcal{N}$ and the macroeconomic model $\Omega \subseteq \mathbb{R}^M$. The latter one will be discretized by a tensor product multiwavelet bases. In general, the dimension of the basis grows exponentially with M – the curse of dimensionality. Thus, the number of macroeconomic variables that can be used is often strongly limited by the available memory. This can be seen in [13], where the number of degrees of freedom in 10 dimensions is not enough to reach the optimal convergence rate. In [34] it can also be seen, that the number of degrees of freedom which can be used in 5 dimensions is strongly limited. By applying principal component analysis, [35], the authors are able to solve a problem in 30 dimensions essentially by a reduction to 5 dimensions. In [22], 8 dimensions are reached for a full rank Black Scholes model and 16 dimensions, when a stochastic volatility model is considered.

The remainder of this paper is organized as follows. In Section 2, we derive a variational formulation to (1.1) and prove its well-posedness. Section 3 is devoted to the description of well-known multiwavelet bases and the collection of the main properties that are needed here. The discretization in Section 4 is done in three steps. First, we use the multiwavelet basis in order to derive an equivalent discrete but infinite-dimensional system. We also show that this approach allows to decouple the market variables from the Markov chain state space in terms of a tensor product. The next two steps involve the discretization in time and market variables. Due to the mentioned separation, we can handle large portfolios of companies by the so-called Hierarchical Tucker Format (HTF) which is briefly reviewed in Section 5 concentrating on those properties that are relevant here. Finally, in Section 6, we report on some numerical experiments for realistic market scenarios. We collect some auxiliary facts in Appendix A.

2. Variational formulation

We start by deriving a variational formulation of the original system (1.1). We start with some remarks on *systems* of elliptic partial differential equations. Let $V \hookrightarrow H \hookrightarrow V'$ be a Gelfand triple and $\mathbf{V} := V^N$ be the tensor product space. For $\mathbf{u} = (u^0, \dots, u^{N-1})^T$, $\mathbf{v} = (v^0, \dots, v^{N-1})^T \in \mathbf{V}$, let $a^j : \mathbf{V} \times V \to \mathbb{R}$ be a multilinear form and $f^j : \mathbf{V} \to \mathbb{R}$, $j = 0, \dots, N-1$, a linear form. Then,

$$\mathbf{u} \in \mathbf{V} : \quad a^j(\mathbf{u}, v) = f^j(v) \quad \forall\, v \in V,\, j \in \mathcal{N}, \tag{2.1}$$

is a coupled linear system of N equations. Defining $\mathbf{a}:\mathbf{V}\times\mathbf{V}\to\mathbb{R}$, $\mathbf{f}:\mathbf{V}\to\mathbb{R}$ we

$$\mathbf{a}(\mathbf{u},\mathbf{v}) := \sum_{j\in\mathcal{N}} a^j(\mathbf{u},v^j), \qquad \mathbf{f}(\mathbf{v}) := \sum_{j\in\mathcal{N}} f^j(v^j), \quad \mathbf{u},\mathbf{v}\in\mathbf{V},$$

obtain a variational problem

$$\mathbf{u}\in\mathbf{V}: \quad \mathbf{a}(\mathbf{u},\mathbf{v}) = \mathbf{f}(\mathbf{v}) \quad \forall\,\mathbf{v}\in\mathbf{V}, \tag{2.2}$$

which is well-posed provided the well-known Nečas conditions are valid, [32]. Note that (2.1) and (2.2) are equivalent since using the test functions $v^j\boldsymbol{\delta}_j$, where $\boldsymbol{\delta}_j = (\delta_{j,j'})^T_{j'\in\mathcal{N}}$ ($\delta_{i,j}$ denoting the Kronecker delta) in (2.2) yields (2.1); the other direction is trivial.

Next, we need to separate the high dimensional Markov chain parts from the variational formulation. This means that the state dependent variables are compound functions of a state dependent part (which might also depend on the time t) and a mutual factor depending on the space variables y. Hence, we assume that there exist functions $\tilde{q}^{j,k},\tilde{a}^{j,k},\tilde{c}^j:[0,T]\to\mathbb{R}$, constants $\tilde{a}^j\in\mathbb{R}$ and functions $h_q,h_a,h_c,h_{a(T)}:\Omega\to\mathbb{R}$ such that

$$q^{j,k}(t,y) = \tilde{q}^{j,k}(t)\,h_q(y), \qquad a^{j,k}(t,y) = \tilde{a}^{j,k}(t)\,h_a(y), \tag{2.3a}$$

$$c^j(t,y) = \tilde{c}^j(t)\,h_c(y), \qquad a^j(y) = \tilde{a}^j\,h_{a(T)}(y), \tag{2.3b}$$

for all $j,k\in\mathcal{N}$, $t\in[0,T]$ and $y\in\Omega$. This is a reasonable assumption from the financial point of view since it states that the dependency on the market process is the same for all points in time and for all states in the Markov chain. The fact, that changes of the state of the Markov chain cannot alter the dependency of the market process Y means that default of single firms in the CDO portfolio will not change the market situation. Finally, we remark that there are methods available in order to obtain an approximate representation of the form (2.3) even in cases where the functions do not directly allow such a separation of variables, see e.g. [4, 33].

We are now going to derive a variational formulation. We need one more abbreviation: If $v:(0,T)\times\Omega\to\mathbb{R}$ is a function in time and space, we will always abbreviate $v(t):\Omega\to\mathbb{R}$, where $v(t)(y) := v(t,y)$, $y\in\Omega$.

Definition 2.1. *Given assumption* (2.3), *a function* $\mathbf{u}\in\mathbf{X} := L_2(0,T;H^1_0(\Omega)^N)\cap H^1(0,T;H^{-1}(\Omega)^N)$ *is called* weak solution *of* (1.1) *if*

$$\begin{cases} (\mathbf{u}_t(t),\mathbf{v})_{0;\Omega} + \mathbf{a}(\mathbf{u}(t),\mathbf{v}) = (\mathbf{f}(t),\mathbf{v})_{0;\Omega} \text{ for all } \mathbf{v}\in H^1_0(\Omega)^N, t\in[0,T] \\ \mathbf{u}(T,y) = \mathbf{u}_T(y) := (u^0_T(y),\dots,u^{N-1}_T(y))^T \end{cases} \tag{2.4}$$

where $(\mathbf{w},\mathbf{v})_{0;\Omega} = \sum_{j\in\mathcal{N}}(w^j,v^j)_{0;\Omega}$, $\mathbf{a}(\mathbf{w},\mathbf{v}) := \sum_{j\in\mathcal{N}} a^j(\mathbf{w},v^j)$ *with*

$$a^j(\mathbf{w}(t),v) := \frac{1}{2}(\nabla w^j(t),\mathbf{B}(t)\,\nabla v)_{0;\Omega} - (\boldsymbol{\alpha}(t)^T\nabla w^j(t) + \boldsymbol{\gamma}^j(t)^T\mathbf{w}(t),v)_{0;\Omega} \tag{2.5}$$

the reaction coefficient $(j,k\in\mathcal{N})$

$$\gamma^j_k(t,y) := (\boldsymbol{\gamma}^j(t,y))_k := \begin{cases} -\tilde{q}^{j,k}(t)\,h_q(y) & \text{if } k\neq j, \\ r(t) - \sum_{k'\in\mathcal{N}\setminus\{j\}}\tilde{q}^{j,k'}(t)\,h_q(y) & \text{if } k = j. \end{cases}$$

and the right-hand side $(\mathbf{f}(t),\mathbf{v})_{0;\Omega} := \sum_{j\in\mathcal{N}}(f^j(t),v^j)_{0;\Omega}$ *with* $f^j(t) := -\tilde{c}^j(t)\,h_c(y) - \sum_{k\in\mathcal{N}\setminus\{j\}}\tilde{q}^{j,k}(t)\,\tilde{a}^{j,k}(t)\,h_a(y)\,h_q(y)$.

Obviously, (2.4) is a system of instationary convection-diffusion-reaction equation and the linear coupling is in the zero-order (reactive) term.

Theorem 2.2. *Let* (2.3) *hold. If* $\mathbf{u} \in C^1([0,T];(C^2(\Omega))^N)$ *is a classical solution of* (1.1)*, then it is also a weak solution in the sense of Definition 2.1. On the other hand, if* $\mathbf{u}$ *is a weak solution and additionally* $\mathbf{u} \in C^1([0,T];(C^2(\Omega))^N)$*, then* $\mathbf{u}$ *is also a classical solution of* (1.1).

Proof. We multiply (1.1) with some $v^j \in H_0^1(\Omega)$ and obtain

$$\begin{aligned}(u_t^j(t), v^j)_{0;\Omega} &= (r(t)u^j(t), v^j)_{0;\Omega} - (\boldsymbol{\alpha}(t)^T \nabla u^j(t), v^j)_{0;\Omega} \\ &\quad -\frac{1}{2}(\nabla \cdot (\mathbf{B}(t)\nabla u^j(t)), v^j)_{0;\Omega} \\ &\quad - \sum_{k\in\mathcal{N}\setminus\{j\}} \int_\Omega q^{j,k}(t,y)\,\big(u^k(t,y) - u^j(t,y)\big)v^j(y)\,dy \\ &\quad - \int_\Omega \Big\{c^j(t,y) + \sum_{k\in\mathcal{N}\setminus\{j\}} q^{j,k}(t,y)\,a^{j,k}(t,y)\Big\}v^j(y)\,dy.\end{aligned}$$

Using assumption (2.3), the (negative of the) last term reads

$$\tilde{c}^j(t)\int_\Omega h_c(y)\,v^j(y)\,dy + \sum_{k\in\mathcal{N}\setminus\{j\}} \tilde{q}^{j,k}(t)\,\tilde{a}^{j,k}(t)\int_\Omega h_q(y)\,h_a(y)\,v^j(y)\,dy = (f^j(t), v^j)_{0;\Omega}.$$

Integration by parts gives for the last term

$$\begin{aligned}&\frac{1}{2}(\mathbf{B}(t)\nabla u^j(t)), \nabla v^j)_{0;\Omega} - (\boldsymbol{\alpha}(t)^T\nabla u^j(t), v^j)_{0;\Omega} \\ &+ \int_\Omega \Big\{r(t)\,u^j(t,y) - \sum_{k\in\mathcal{N}\setminus\{j\}} \tilde{q}^{j,k}(t)\,h_q(y)\Big(u^k(t,y) - u^j(t,y)\Big)\Big\}v^j(y)\,dy,\end{aligned} \tag{2.6}$$

where the last term is equal to $(\boldsymbol{\gamma}^j(t)^T\mathbf{u}(t), v^j)_{0;\Omega}$. Summing over $j \in \mathcal{N}$ yields (2.4). The above derivation also proves the claim. □

The next step is to prove well-posedness of the variational problem.

Theorem 2.3. *If* $\mathbf{B}(t)$ *has rank* M*, then* (2.4) *is well-posed.*

Proof. We need to show that the bilinear form $\mathbf{a}(\cdot,\cdot)$ satisfies the Gårding inequality and is continuous. Then, the claim follows from the Lax-Milgram theorem. □

Remark 2.4. *Note that* (2.3) *is not needed for the well-posedness of* (2.4).

Finally, consider a space-time variational formulation of (2.4) by integrating over time. With

$$\begin{aligned}\mathbf{b}(\mathbf{u},\mathbf{v}) &:= \int_0^T [(\mathbf{u}_t(t), \mathbf{v}_1)_{0;\Omega} + \mathbf{a}(\mathbf{u}(t), \mathbf{v}_1)]\,dt + (\mathbf{u}(T), \mathbf{v}_2)_{0;\Omega} \\ \mathbf{f}(\mathbf{v}) &:= \int_0^T (\mathbf{f}(t), \mathbf{v}_1(t))_{0;\Omega} + (\mathbf{u}_T, \mathbf{v}_2)_{0;\Omega}\end{aligned}$$

for $\mathbf{u} \in \mathbf{X}$ and $\mathbf{v} \in \mathbf{Y} := L_2(0,T;H_0^1(\Omega)^N)\times L_2(\Omega)^N$, $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2) \in \mathbf{Y}$, the space-time variational formulation reads

$$\mathbf{u} \in \mathbf{X}: \qquad \mathbf{b}(\mathbf{u},\mathbf{v}) = \mathbf{f}(\mathbf{v}) \quad \forall\, \mathbf{v} \in \mathbf{Y}. \tag{2.7}$$

This latter problem is also well-posed following the arguments e.g. in [38, 46].

3. Multiwavelets

Since we want to use multiwavelets for the discretization of the macroeconomic variables, we briefly recall some facts of these function systems. A (standard, not multi-) wavelet system is a Riesz basis $\Psi := \{\psi_\lambda : \lambda \in \mathcal{J}\}$ of $L_2(\Omega)$, where $\lambda = (\ell, k)$, $|\lambda| := \ell \geq 0$ denotes the *level* (also steering the size of the support in the sense that $\operatorname{diam}(\operatorname{supp} \psi_\lambda) \sim 2^{-|\lambda|}$) and k indicates the position of $\operatorname{supp} \psi_\lambda$, e.g. the center of the support. Wavelets are (among other parameters) characterized by a certain order d of *vanishing moments*, i.e.,

$$\int_\Omega y^r \, \psi_\lambda(y)\, dy = 0 \quad \forall\, 0 \leq |r| \leq d-1 \quad \forall\, \lambda \in \mathcal{J}, |\lambda| > 0. \tag{3.1}$$

This means that wavelets necessarily oscillate which also explains the name. Note that (3.1) only holds for $|\lambda| > 0$. Those functions ψ_λ with $|\lambda| = 0$ are *not* wavelets but so-called *scaling functions* and those are generated by a single so-called *generator* $\varphi \in C_0(\Omega)$ in the sense that each ψ_λ, $|\lambda| = 0$, is a linear combination of (possibly to Ω restricted) shifts $\varphi(\cdot - k)$, $k \in \mathbb{Z}$. The wavelets ψ_λ, $|\lambda| > 0$, are linear combinations of dilated versions of scaling functions.

The difference of *multiwavelets* as opposed to wavelets is that linear combinations of shifts of *several* generators φ_i, $i = 1, \ldots, m$, are allowed. The main advantage is that corresponding multiwavelets may be constructed that are

- piecewise polynomial,
- L_2-orthogonal,
- compactly supported with small support size.

These three properties are quite useful for numercial methods since they allow an efficient evaluation of an approximation as well as well-conditioned and sparse system matrices.

We use B-spline multiple generators and wavelets as constructed in [16, 19]. These functions are also adapted to finite intervals and allow for homogeneous Dirichlet boundary conditions, the latter construction was introduced in [36]. We faced some difficulties with the realization of the construction in [16] in particular for higher regularity. However, we finally came up with a realization using Mathematica® for almost arbitrary regularity. Some functions are shown in Figures 1 and 2. Details can be found in [36].

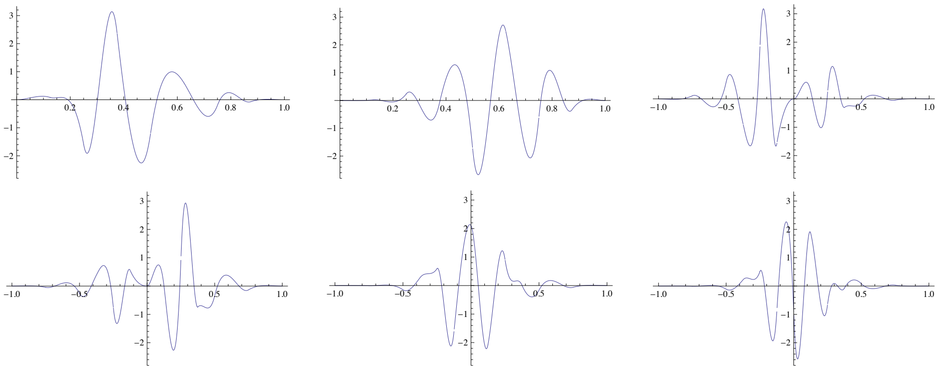

Figure 1: Wavelets generated by the piecewise cubic MRA having one continuous derivative.

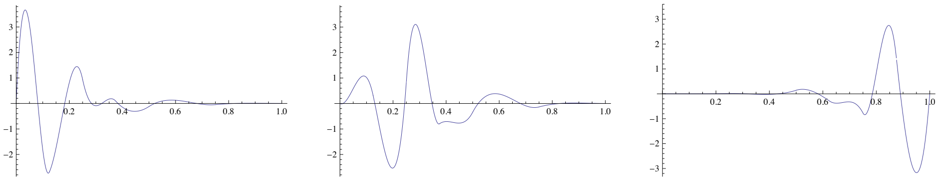

Figure 2: Wavelets with homogeneous Dirichlet boundary conditions generated by a piecewise cubic MRA on $[0,1]$ with one continuous derivative.

Let us summarize some properties that we will need in the sequel.

Proposition 3.1 ([16])**.** *Let* $\Psi = \{\psi_\lambda : \lambda \in \mathcal{J}\}$ *be a system of multiwavelets on* $\Omega = [0,1]$ *from* [16] *normalized in* $H^1(\Omega)$*, i.e.,* $\|\psi_\lambda\|_{1;\Omega} \sim 1$*. Then,*

(a) Ψ *is* L_2*-orthogonal, i.e.,* $(\psi_\lambda, \psi_\mu)_{0;\Omega} = \delta_{\lambda,\mu}\|\psi_\lambda\|^2_{0;\Omega}$, $\lambda, \mu \in \mathcal{J}$*;*

(b) $\psi_\lambda \in H^1_0(\Omega)$, $\lambda \in \mathcal{J}$*;*

(c) *The system* Ψ *is a Riesz basis for* $H^1_0(\Omega)$ *with* L_2*-orthogonal functions.*

Finally, denoting $\boldsymbol{\mathcal{J}} := \{(j,\lambda) : j \in \mathcal{N}; \lambda \in \mathcal{J}\} = \mathcal{N} \times \mathcal{J}$ (i.e., $|\boldsymbol{\mathcal{J}}| = N|\mathcal{J}|$), $\boldsymbol{\lambda} := (j,\lambda) \in \boldsymbol{\mathcal{J}}$ and $\boldsymbol{\psi}_{\boldsymbol{\lambda}} := \psi_\lambda \boldsymbol{\delta}_j$, $\boldsymbol{\lambda} = (j,\lambda) \in \boldsymbol{\mathcal{J}}$, the system $\boldsymbol{\Psi} := \{\boldsymbol{\psi}_{\boldsymbol{\lambda}} : \boldsymbol{\lambda} \in \boldsymbol{\mathcal{J}}\}$ is a tensor product Riesz basis for $H^1_0(\Omega)^N$.

4. Discretization

4.1. **An equivalent ℓ_2-problem.**

The first step towards an adaptive multiwavelet method is to rewrite the variational problem (2.4) and (2.7) in a discrete equivalent problem on the sequence space $\ell_2(\boldsymbol{\mathcal{J}})$ for the multiwavelet expansion coefficients. It turns out that the assumption (2.3) is particularly useful here, since it allows for a separation of state and space (and time), so that the discrete operators are of tensor product form. This also allows for an efficient numercial realization, also for the space-time variational formulation [25] and in particular for larger M.

Using $\boldsymbol{\Psi}$ as defined in Section 3, the solution $\mathbf{u}$ of (2.4) has a unique expansion of the form

$$\mathbf{u}(t,y) = \sum_{\boldsymbol{\lambda}\in\boldsymbol{\mathcal{J}}} \mathbf{x}_{\boldsymbol{\lambda}}(t)\, \boldsymbol{\psi}_{\boldsymbol{\lambda}}(y), \quad t \in (0,T),\, y \in \Omega,$$

where $\mathbf{x}_{\boldsymbol{\lambda}}(t) = x^j_\lambda(t)$, $\boldsymbol{\lambda} = (j,\lambda)$, $\mathbf{x}^j(t) = (x^j_\lambda(t))_{\lambda\in\mathcal{J}} \in \ell_2(\mathcal{J})$. The above sum is to be understood componentwise, i.e., $u^j(t,y) = \sum_{\lambda\in\mathcal{J}} x^j_\lambda(t)\,\psi_\lambda(y)$ for $j \in \mathcal{N}$. Then, for $\boldsymbol{\lambda} = (j,\lambda) \in \boldsymbol{\mathcal{J}}$, we get

$$\begin{aligned} a^j(\mathbf{u}(t),\psi_\lambda) &= \sum_{\mu\in\mathcal{J}} x_\mu(t)\Big\{\frac{1}{2}(\nabla\psi_\mu, \mathbf{B}(t)\,\nabla\psi_\lambda)_{0;\Omega} - (\boldsymbol{\alpha}(t)^T\,\nabla\psi_\mu, \psi_\lambda)_{0;\Omega}\Big\} \\ &\quad + \sum_{k\in\mathcal{N}} (\gamma^j_k(t)u^k(t), \psi_\lambda)_{0;\Omega}. \end{aligned}$$

Defining $\mathbf{A}(t) := \Big(\frac{1}{2}(\nabla\psi_\mu, \mathbf{B}(t)\,\nabla\psi_\lambda)_{0;\Omega} - (\boldsymbol{\alpha}(t)^T\,\nabla\psi_\mu, \psi_\lambda)_{0;\Omega}\Big)_{\lambda,\mu\in\mathcal{J}}$, the first sum can be abbreviated as $\mathbf{A}(t)\mathbf{x}^j(t)$. The remaining term can be further detailed as

$$\sum_{k\in\mathcal{N}} (\gamma^j_k(t)u^k(t), \psi_\lambda)_{0;\Omega} = \sum_{k\in\mathcal{N}}\sum_{\mu\in\mathcal{J}} x^k_\mu(t)(\gamma^j_k(t)\psi_\mu, \psi_\lambda)_{0;\Omega} = \Big[\sum_{k\in\mathcal{N}} \mathbf{C}^{j,k}(t)\mathbf{x}^k(t)\Big]_\lambda,$$

where we set $\mathbf{C}^{j,k}(t) := \big((\gamma_k^j(t)\psi_\lambda,\psi_\mu)_{0;\Omega}\big)_{\lambda,\mu\in\mathcal{J}} \in \mathbb{R}^{\mathcal{J}\times\mathcal{J}}$.[1] Next,

$$(u_t^j(t),\psi_\lambda)_{0;\Omega} = \sum_{\mu\in\mathcal{J}} \dot{x}_\mu(t)\,(\psi_\mu,\psi_\lambda)_{0;\Omega} = \dot{x}_\lambda(t),$$

if the multiwavelets are L_2-orthonormalized. Thus, we obtain

$$\dot{\mathbf{x}}^j(t) + \mathbf{A}(t)\,\mathbf{x}^j(t) + \sum_{k\in\mathcal{N}} \mathbf{C}^{j,k}(t)\mathbf{x}^k(t), \quad j\in\mathcal{N},$$

or, written as a system

$$\dot{\mathbf{x}}(t) + (\boldsymbol{\mathcal{A}}(t) + \boldsymbol{\mathcal{C}}(t))\,\mathbf{x}(t) = \boldsymbol{f}(t), \tag{4.1}$$

where $\boldsymbol{\mathcal{A}}(t), \boldsymbol{\mathcal{C}}(t) \in \mathbb{R}^{\boldsymbol{\mathcal{J}}\times\boldsymbol{\mathcal{J}}}$ are given by $\boldsymbol{\mathcal{A}}(t) = \operatorname{diag}(\mathbf{A}(t)\ \ldots, \mathbf{A}(t))$, $(\boldsymbol{\mathcal{C}}(t))_{j,k} = \mathbf{C}^{j,k}(t)$ and $\boldsymbol{f}(t) = \big((\mathbf{f}(t),\boldsymbol{\psi_\mu})_{0;\Omega}\big)_{\boldsymbol{\mu}\in\boldsymbol{\mathcal{J}}}$. Obviously, (4.1) is a coupled system of ODEs in the sequence space $\ell_2(\boldsymbol{\mathcal{J}})$. We will now show that (4.1) is in fact a tensor product problem. For that, we need to review some facts on tensor products, which can be found in Appendix A. We detail the coupling term

$$\begin{aligned}
[\mathbf{C}^{j,k}(t)]_{\lambda,\mu} &= (\gamma_k^j(t)\psi_\lambda,\psi_\mu)_{0;\Omega} \\
&= \begin{cases} -\tilde{q}^{j,k}(t)\,(h_q\psi_\lambda,\psi_\mu)_{0;\Omega}, & \text{if } k \neq j, \\ r(t)\delta_{\lambda,\mu} - \sum_{k\in\mathcal{N}\setminus\{j\}} \tilde{q}^{j,k'}(t)\,(h_q\psi_\lambda,\psi_\mu)_{0;\Omega}, & \text{if } k = j, \end{cases} \\
&=: d^{j,k}(t)\mathbf{M}^q_{\lambda,\mu} + r(t)\,\delta_{j,k}\,\delta_{\lambda,\mu},
\end{aligned}$$

where $(\mathbf{M}^q)_{\lambda,\mu} := (h_q\psi_\lambda,\psi_\mu)_{0;\Omega}$ is a weighted mass matrix and

$$\mathbb{R} \ni d^{j,k}(t) := \begin{cases} -\tilde{q}^{j,k}(t), & \text{if } k \neq j, \\ -\sum_{m\in\mathcal{N}\setminus\{j\}} \tilde{q}^{j,m}(t), & \text{if } k = j. \end{cases} \tag{4.2}$$

We denote $\mathbf{D}(t) := (d^{j,k}(t))_{j,k\in\mathcal{N}}$.

Theorem 4.1. *Let Assumption* (2.3) *hold and assume that* Ψ *satisfies the properties in Proposition 3.1. Then,* (2.4) *is equivalent to*

$$(\mathbf{I}_\mathcal{N}\otimes\mathbf{I}_\mathcal{J})\dot{\mathbf{x}}(t) + [(\mathbf{I}_\mathcal{N}\otimes[\mathbf{A}(t)+r(t)\mathbf{I}_\mathcal{J}]) + (\mathbf{D}(t)\otimes\mathbf{M}^q)]\,\mathbf{x}(t) = \mathbf{b}(t)\otimes\mathbf{g}^1 - \tilde{\mathbf{c}}(t)\otimes\mathbf{g}^2, \tag{4.3}$$

where $\mathbf{I}_\mathcal{I}$ *denotes the identity w.r.t. an index set* $\mathcal{I}$, $\mathbf{b}(t) = (b^j(t))_{j\in\mathcal{N}}$, $b^j(t) := -\sum_{k\in\mathcal{N}\setminus\{j\}} \tilde{q}^{j,k}(t)\,\tilde{a}^{j,k}(t)$, $\tilde{\mathbf{c}}(t) := (\tilde{c}^j(t))_{j\in\mathcal{N}}$, $\mathbf{g}^1 = (g^1_\lambda)_{\lambda\in\mathcal{J}}$, $g^1_\lambda := (h_q\,h_a,\psi_\lambda)_{0;\Omega}$ *and* $\mathbf{g}^2 = (g^2_\lambda)_{\lambda\in\mathcal{J}}$ *with* $g^2_\lambda := (h_c,\psi_\lambda)_{0;\Omega}$.

Proof. Let $j\in\mathcal{N}$ and $\lambda\in\mathcal{J}$ so that $\boldsymbol{\lambda} = (j,\lambda)\in\boldsymbol{\mathcal{J}}$. Then,

$$\begin{aligned}
(\boldsymbol{\mathcal{C}}(t)\,\mathbf{x}(t))_{\boldsymbol{\lambda}} &= \sum_{k\in\mathcal{N}}\sum_{\mu\in\mathcal{J}} [\mathbf{C}^{j,k}(t)]_{\lambda,\mu}\,x^k_\mu(t) \\
&= \sum_{k\in\mathcal{N}}\sum_{\mu\in\mathcal{J}} [d^{j,k}(t)\mathbf{M}^q_{\lambda,\mu} + r(t)\delta_{j,k}\delta_{\lambda,\mu}]x^k_\mu(t) \\
&= \sum_{k\in\mathcal{N}}\sum_{\mu\in\mathcal{J}} [\mathbf{M}^q_{\lambda,\mu}x^k_\mu(t)d^{j,k}(t) + r(t)\delta_{\lambda,\mu}x^k_\mu(t)\delta_{j,k}] \\
&= \Big(\mathbf{M}^q\,\mathbf{x}(t)\mathbf{D}(t)^T + r(t)\mathbf{I}_\mathcal{J}\,\mathbf{x}(t)\mathbf{I}_\mathcal{N}\Big)_{\boldsymbol{\lambda}} \\
&= \Big([(\mathbf{D}(t)\otimes\mathbf{M}^q) + r(t)(\mathbf{I}_\mathcal{N}\otimes\mathbf{I}_\mathcal{J})]\mathbf{x}(t)\Big)_{\boldsymbol{\lambda}},
\end{aligned}$$

[1] With a slight abuse of notation, we set $\mathbb{R}^\mathcal{I} = \ell_2(\mathcal{I})$ for any countable (possibly infinite) set $\mathcal{I}$ as well as $\mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ as the set of linear operators from $\ell_2(\mathcal{I})$ into $\ell_2(\mathcal{I})$.

where we have used Lemma A.4 in the last step. Note that $\mathbf{D}(t) \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $\mathbf{M}^q \in \mathbb{R}^{\mathcal{J}\times\mathcal{J}}$, thus $(\mathbf{D}(t)\otimes\mathbf{M}^q) \in \mathbb{R}^{(\mathcal{N}\times\mathcal{J})\times(\mathcal{N}\times\mathcal{J})} = \mathbb{R}^{\boldsymbol{\mathcal{J}}\times\boldsymbol{\mathcal{J}}}$. Finally, we consider the right-hand side. We obtain

$$\begin{aligned}(\mathbf{b}(t)\otimes\mathbf{g}^1 - \tilde{\mathbf{c}}(t)\otimes\mathbf{g}^2)_{(j,\lambda)} &= \; b^j(t)\, g^1_\lambda - \tilde{c}^j(t)\, g^2_\lambda \\ &= \; -\sum_{k\in\mathcal{N}\setminus\{j\}} \tilde{q}^{j,k}(t)\,\tilde{a}^{j,k}(t)\,(h_q\, h_a, \psi_\lambda)_{0;\Omega} - \tilde{c}^j(t)\,(h_c, \psi_\lambda)_{0;\Omega} \\ &= \; (f^j(t), \psi_\lambda)_{0;\Omega},\end{aligned}$$

so that the claim is proven. □

4.2. **Temporal discretization.** So far (4.3) is equivalent to the original PDE. As a first step towards a (finite) discretization, we consider the time interval $[0,T]$, fix some $K\in\mathbb{N}$ and set $\Delta t := \frac{1}{K}$ as well as $t^k := k\,\Delta t$, $k = 0,\ldots,K$. Obviously, (4.3) takes the form $\mathscr{M}\dot{x}(t) + \mathscr{A}(t)\,x(t) = f(t)$, $x(T) = x_T$, to which we apply the standard θ-scheme ($\theta\in[0,1]$) to derive an approximation $x^k \approx x(t^k)$ by solving

$$\begin{aligned} x^K &= x_T, \\ \frac{1}{\Delta t}\mathscr{M}(x^{k+1} - x^k) + \theta\mathscr{A}(t^{k+1})x^{k+1} + (1-\theta)\mathscr{A}(t^k)x^k &= \theta f(t^{k+1}) + (1-\theta) f(t^k),\end{aligned}$$

for $k = K-1,\ldots,0$. Obviously, the second equation amounts solving the following linear system in each time step

$$(\mathscr{M} + \Delta t(\theta-1)\mathscr{A}(t^k))x^k = (\mathscr{M} + \Delta t\,\theta\mathscr{A}(t^{k+1}))x^{k+1} + \theta f(t^{k+1}) + (1-\theta)f(t^k). \tag{4.4}$$

4.3. **Wavelet Galerkin methods.** The last step towards a fully discrete systems in space and time is the discretization with respect to the economic variable $y\in\Omega$. After having transformed (2.4) into the discrete but infinite-dimensional system (4.3), this can easily be done by selecting a finite index set $\Lambda\subset\mathcal{J}$. Hence, we obtain

$$\mathscr{M}_\Lambda := \mathbf{I}_\mathcal{N}\otimes\mathbf{I}_\Lambda, \quad \mathscr{A}_\Lambda(t) := \mathbf{I}_\mathcal{N}\otimes[\mathbf{A}_\Lambda(t) + r(t)\mathbf{I}_\Lambda] + \mathbf{D}(t)\otimes\mathbf{M}^q_\Lambda,$$

which is then inserted into (4.4) for $\mathscr{M}$ and $\mathscr{A}(t)$, respectively, in order to get a finite system. We denote by $\mathbf{A}_\Lambda(t) := \mathbf{A}(t)|_\Lambda = (a_{\lambda,\mu}(t))_{\lambda,\mu\in\Lambda}$ the restriction of the original bi-infinite operator $\mathbf{A}(t)$ to a finite index set $\Lambda\subset\mathcal{J}$, $|\Lambda|<\infty$ (and similarly $\mathbf{I}_\Lambda$, $\mathbf{M}^q_\Lambda$). The choice of Λ is done in an adaptive manner, i.e., we get a sequence $\Lambda^{(0)}\to\Lambda^{(1)}\to\Lambda^{(2)}\to\cdots$ by one of the known adaptive wavelet schemes that have been proven to be asymptotically optimal, [11, 10, 12, 24, 37, 45].

5. The Hierarchical Tucker Format (HTF)

In this section, we briefly recall the main properties of the *Hierarchical Tucker Format (HTF)* and describe key features of our implementation. We concentrate on those issues needed for the pricing problem under consideration and refer e.g. to [20, 21, 27, 36] for more details.

We call $\mathbf{w}\in\mathbb{R}^\mathcal{K}$, $\mathcal{K} = \times_{j\in\mathcal{N}}\mathcal{K}_j$ with entries $\mathbf{w}_\mathbf{i}\in\mathbb{R}$, $\mathbf{i} = (i_0,\ldots,i_{N-1})^T = (i_j)_{j\in\mathcal{N}}$, $i_j\in\mathcal{K}_j$, a *tensor* of *order* N.[2] Note that we will consider the cases $\mathcal{K}_j = \mathcal{I}_j$ (a *vector-tensor*) as well as $\mathcal{K}_j = \mathcal{J}_j\times\mathcal{I}_j$ (a *matrix-tensor*), where $\mathcal{I}_j$ and $\mathcal{J}_j$ are suitable (possibly adaptively chosen) index sets. Storing and numerically manipulating tensors exactly is extremely expensive since the amount of storage and work grows

[2] The indexation here is adapted to our problem at hand and thus differs from the standard literature on the Hierarchical Tucker Format.

exponentially with the order. Hence, one wishes to approximate a tensor $\mathbf{w}$ (or $\operatorname{vec}(\mathbf{w})$, which denotes the vector storage of $\mathbf{w}$ using reverse lexicographical order w.r.t. the indices) by some efficient format. One example is the *Tucker Format*, [44], where one aims at determining an approximation

$$\operatorname{vec}(\mathbf{w}) \approx \mathbf{V} \operatorname{vec}(\mathbf{c}) = (V_{N-1} \otimes \cdots \otimes V_0) \operatorname{vec}(\mathbf{c}), \quad V_j \in \mathbb{R}^{\mathcal{I}_j \times \mathcal{J}_j},\ j \in \mathcal{N}, \tag{5.1}$$

with the so called *core tensor* $\mathbf{c} \in \mathbb{R}^{\mathcal{J}}$, $\mathcal{J} = \times_{j \in \mathcal{N}} \mathcal{J}_j$. It is known that the *High-Order Singular Value Decomposition (HOSVD)* (see (5.2) below) yields a 'nearly' optimal solution to the approximation problem (5.1) which is also easily numerically realizable, [36]. However, the storage amount for the core tensor $\mathbf{c}$ still grows exponentially with N. This is the reason to consider alternative formats such as the HTF which provides an efficient multilevel format for the core tensor $\mathbf{c}$. In order to be able to describe the HTF, it is useful to introduce the concept of matricization as well as to describe the HOSVD in some more detail.

The direction indices $0, \dots, N-1$ of a tensor $\mathbf{w} \in \mathbb{R}^{\mathcal{I}}$ are also called *modes*. Consider a splitting of the set of all modes $\{0, \dots, N-1\} = \mathcal{N}$ into disjoint sets, i.e., $\mathcal{N} = t \cup s$, $t = \{t_1, \dots, t_k\}$, $s = \{s_1, \dots, s_{N-k}\} = t^{\mathsf{C}}$, then the *matricization* $\mathbf{w}^{(t)} \in \mathbb{R}^{\mathcal{I}_t \times \mathcal{I}_t^{\mathsf{C}}}$ of the tensor $\mathbf{w}$ w.r.t. the modes t is defined as follows

$$\mathcal{I}_t := \bigtimes_{i=1}^{k} \mathcal{I}_{t_i}, \quad \mathcal{I}_t^{\mathsf{C}} := \bigtimes_{i=1}^{N-k} \mathcal{I}_{s_i}, \quad (\mathbf{w}^{(t)})_{(i_{t_1}, \dots, i_{t_k}), (i_{s_1}, \dots, i_{s_{N-k}})} := \mathbf{w}_{\mathbf{i}}.$$

Note that $\operatorname{vec}(\mathbf{w}) = \mathbf{w}^{(\mathcal{N})}$. A special case is the *μ-matricization* for $\mu \in \mathcal{N}$, where $t = \{\mu\}$ and $\mathcal{I}_\mu^{\mathsf{C}} = \mathcal{I}_0 \times \cdots \times \mathcal{I}_{\mu-1} \times \mathcal{I}_{\mu+1} \times \cdots \times \mathcal{I}_{N-1}$. We set $r_\mu := \operatorname{rank}(\mathbf{w}^{(\mu)})$ and call $\mathbf{r} = (r_0, \dots, r_{N-1})^T$ the *rank* of $\mathbf{w}$.

One idea to obtain an approximation $\tilde{\mathbf{w}}$ of $\mathbf{w}$ requiring less storage is a low-rank approximation, i.e., to determine a tensor $\tilde{\mathbf{w}}$ of rank $\tilde{\mathbf{r}}$ with $\tilde{r}_\mu \le r_\mu \le \#\mathcal{I}_\mu$. This can be achieved by a truncated SVD of each $\mathbf{w}^{(\mu)}$ in the sense that $\mathbf{w}^{(\mu)} \approx U_\mu \Sigma_\mu V_\mu^T$, i.e., $U_\mu \in \mathbb{R}^{\mathcal{I}_\mu \times \tilde{r}_\mu}$ contains the most significant $\tilde{r}_\mu$ left singular vectors of $\mathbf{w}^{(\mu)}$. Then

$$\operatorname{vec}(\mathbf{w}) \approx \operatorname{vec}(\tilde{\mathbf{w}}) := (U_{N-1} \otimes \cdots \otimes U_0) \operatorname{vec}(\mathbf{c}) \tag{5.2}$$

with the core tensor $\operatorname{vec}(\mathbf{c}) := (U_{N-1}^T \otimes \cdots \otimes U_0^T) \operatorname{vec}(\mathbf{w}) \in \mathbb{R}^{\tilde{r}_0 \times \tilde{r}_{N-1}}$ and the *mode frames* U_μ, $\mu \in \mathcal{N}$. The approximation in (5.2) is precisely the HOSVD and this choice of $\mathbf{c}$ can easily be shown to minimize $\|\mathbf{w} - \tilde{\mathbf{w}}\|_2$ for given orthonormal matrices U_j. Moreover,

$$\|\mathbf{w} - \tilde{\mathbf{w}}\|_2 \le \sqrt{N} \inf\{\|\mathbf{w} - \mathbf{v}\|_2 : \mathbf{v} \in \mathbb{R}^{\mathcal{I}},\ \operatorname{rank}(\mathbf{v}^{(\mu)}) \le \tilde{r}_\mu, \mu \in \mathcal{N}\}.$$

The main idea behind the HTF is to construct a hierarchy of matricizations and to make use of the arising multilevel structure. It has been shown in [20, Lemma 17] that for $t = t_\ell \cup t_r$, $t \subseteq \mathcal{N}$, we have $\operatorname{span}(\mathbf{w}^{(t)}) \subseteq \operatorname{span}(\mathbf{w}^{(t_r)} \otimes \mathbf{w}^{(t_\ell)})$.[3] This result has the following consequence: If we consider $\mathbf{w}^{(t)}$, $\mathbf{w}^{(t_\ell)}$ and $\mathbf{w}^{(t_r)}$ as defined above and denote by U_t, U_{t_ℓ} and U_{t_r} any (column-wise) bases for the corresponding column spaces, then the result ensures the existence of a so called *transfer matrix* $B_t \in \mathbb{R}^{r_{t_\ell} r_{t_r} \times r_t}$ such that $U_t = (U_{t_r} \otimes U_{t_\ell}) B_t$, where r_t, r_{t_ℓ} and r_{t_r} denote the ranks of the corresponding matricizations. Obviously t_r and t_ℓ provide a subdivision of a mode t. If one recursively applies such a subdivision to $\operatorname{vec}(\mathbf{w}) = U_{\mathcal{N}}$, one obtains a multilevel-type hierarchy. One continues until t_ℓ, t_r become singletons.

[3] By $\operatorname{span}(A)$ we denote the linear span of the column vectors of a matrix A.

The corresponding general decomposition is formulated in terms of a so called dimension tree.

Definition 5.1 ([27, Def. 2.3]). *A binary tree $\mathcal{T}_{\mathcal{N}}$ with each node represented by a subset of $\mathcal{N}$ is called a* dimension tree *if the root is $\mathcal{N}$, each leaf node is a singleton, and each parent node is the disjoint union of its two children. We denote by $\mathcal{L}(\mathcal{T}_{\mathcal{N}})$ the set of all leave nodes and by $\mathcal{I}(\mathcal{T}_{\mathcal{N}}) := \mathcal{T}_N \setminus \mathcal{L}(\mathcal{T}_N)$ the set of inner nodes.*

Now, we define those tensors that can exactly be represented in HTF, called Hierarchical Tucker Tensor (HTT).

Definition 5.2. *Let $\mathcal{T}_{\mathcal{N}}$ be a dimension tree and let $\mathbf{r} := (r_t)_{t \in \mathcal{T}_{\mathcal{N}}} \in \mathbb{N}^{\mathcal{T}_{\mathcal{N}}}$, $r_t \in \mathbb{N}$ be a family of non-negative integers. A tensor $\mathbf{w} \in \mathbb{R}^{\mathcal{I}}$, $\mathcal{I} = \times_{j \in \mathcal{N}} \mathcal{I}_j$, is called* Hierarchical Tucker Tensor (HTT) *of rank $\mathbf{r}$ if there exist families*

(i) *$\mathbf{U} := (U_t)_{t \in \mathcal{L}(\mathcal{T}_{\mathcal{N}})}$ of matrices $U_t \in \mathbb{R}^{\mathcal{I}_t \times r_t}$, rank$(U_t) \le r_t$ (a nested frame tree),*

(ii) *$\mathbf{B} := (B_t)_{t \in \mathcal{I}(\mathcal{T}_{\mathcal{N}})}$ of matrices (the transfer tensors),*

such that vec$(\mathbf{w}) = U_{\mathcal{N}}$ and for each inner node $t \in \mathcal{I}(\mathcal{T}_{\mathcal{N}})$ with children t_ℓ, t_r it holds that $U_t = (U_{t_r} \otimes U_{t_\ell}) B_t$ with $B_t \in \mathbb{R}^{r_{t_\ell} r_{t_r} \times r_t}$.

In order to keep track of the dependencies, we will write $\mathbf{w} = (\mathcal{T}_{\mathcal{N}}^{\mathbf{w}}, \mathbf{r}^{\mathbf{w}}, \mathbf{U}^{\mathbf{w}}, \mathbf{B}^{\mathbf{w}})$.

Remark 5.3.

(a) *By Definition 5.2, we obtain a Tucker representation vec$(\mathbf{w}) = (U_{\{N-1\}} \otimes \cdots \otimes U_{\{0\}})$ vec$(\mathbf{c})$ with vec$(\mathbf{c})$ formed as a multilevel product of the transfer tensors.*

(b) *For the numerical realization, it turns out that it is useful to consider different representation of tensors in terms of different matricizations. This can be seen as different views on the tensor with the same data:*
   (i) *The 'standard' view, i.e., $B_t \in \mathbb{R}^{k_{t\ell} k_{t_r} \times k_t}$;*
   (ii) *The 'tensor view', $\mathscr{B}_t \in \mathbb{R}^{k_t \times k_{t_\ell} \times k_{t_r}}$, where $B_t = \mathscr{B}_t^{(\{2,3\})}$;*

(c) *The storage of an HTT thus requires N matrices $U_\mu \in \mathbb{R}^{\mathcal{I}_\mu \times k_\mu}$, $\mu \in \mathcal{N}$ and $(N-1) = \#\mathcal{I}(\mathcal{T}_{\mathcal{N}})$ transfer tensors B_t, $t \in \mathcal{I}(\mathcal{T}_{\mathcal{N}})$.*

(d) *From (i) it follows that rank$(\mathbf{w}^{(\mu)}) \le r_\mu$.*

(e) *An HTT for a vector $\mathbf{w} \in \mathbb{R}^{\mathcal{I}}$ is called* HT-vector *and for a matrix $\mathbf{A} \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$* HT-matrix*.*

Computing with hierarchical tensors. We are now going to describe some of the algebraic operations for HTT's that are required for the numerical realization of an adaptive wavelet method. Some issues are similar to existing software [27], some are specific due to our wavelet discretization. Our numerical realization is described in detail in [36], where also the source code is available.

Lemma 5.4 ([20]). *Let $\mathbf{v} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathbf{v}}, \mathbf{U}^{\mathbf{v}}, \mathbf{B}^{\mathbf{v}})$ and $\mathbf{w} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathbf{w}}, \mathbf{u}^{\mathbf{w}}, \mathbf{B}^{\mathbf{w}})$ be HTT's of order N w.r.t. the same dimension tree $\mathcal{T}_{\mathcal{N}}$. Then, the sum reads $\mathbf{v} + \mathbf{w} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathbf{v}} + \mathbf{r}^{\mathbf{w}}, \mathbf{U}^{\mathbf{v}+\mathbf{w}}, \mathbf{B}^{\mathbf{v}+\mathbf{w}})$, where $U_t^{\mathbf{v}+\mathbf{w}} = (U_t^{\mathbf{v}} U_t^{\mathbf{w}}) \in \mathbb{R}^{\mathcal{I}_t^{\mathbf{v}} \cup \mathcal{I}_t^{\mathbf{w}} \times (r_t^{\mathbf{v}} + r_t^{\mathbf{w}})}$ and $\mathscr{B}_t^{\mathbf{v}+\mathbf{w}} \in \mathbb{R}^{(r_t^{\mathbf{v}} + r_t^{\mathbf{w}}) \times (r_{t_\ell}^{\mathbf{v}} + r_{t_\ell}^{\mathbf{w}}) \times (r_{t_r}^{\mathbf{v}} + r_{t_r}^{\mathbf{w}})}$ for $t = t_\ell \,\dot\cup\, t_r$ is given by*

$$(\mathscr{B}_t^{\mathbf{v}+\mathbf{w}})_{i,j,k} = \begin{cases} (\mathscr{B}_t^{\mathbf{v}})_{i,j,k}, & 1 \le i \le r_t^{\mathbf{v}}, 1 \le j \le r_{t_\ell}^{\mathbf{v}}, 1 \le k \le r_{t_r}^{\mathbf{v}}, \\ (\mathscr{B}_t^{\mathbf{w}})_{i,j,k}, & r_t^{\mathbf{v}} < i \le r_t^{\mathbf{v}} + r_t^{\mathbf{w}}, r_{t_\ell}^{\mathbf{v}} < j \le r_{t_\ell}^{\mathbf{v}} + r_{t_\ell}^{\mathbf{w}}, r_{t_r}^{\mathbf{v}} < k \le r_{t_r}^{\mathbf{v}} + r_{t_r}^{\mathbf{w}}, \\ 0, & \text{otherwise,} \end{cases}$$

for $t \in \mathcal{I}(\mathcal{T}_{\mathcal{N}}) \setminus \mathcal{N}$ *and*

$$(\mathscr{B}^{\mathbf{v}+\mathbf{w}}_{\mathcal{N}})_{1,j,k} = \begin{cases} (\mathscr{B}^{\mathbf{v}}_{\mathcal{N}})_{1,j,k}, & 1 \le j \le r^{\mathbf{v}}_{t_\ell}, 1 \le k \le r^{\mathbf{v}}_{t_r}, \\ (\mathscr{B}^{\mathbf{w}}_{\mathcal{N}})_{1,j,k}, & r^{\mathbf{v}}_{t_\ell} < j \le r^{\mathbf{v}}_{t_\ell} + r^{\mathbf{w}}_{t_\ell}, r^{\mathbf{v}}_{t_r} \le k < r^{\mathbf{v}}_{t_r} + r^{\mathbf{w}}_{t_r}, \\ 0, & \text{otherwise.} \end{cases}$$

at the root node $t = \mathcal{N}$.

It is particularly worth mentioning that the HTF of the sum of two HTF only requires a reorganization of the data and no additional computational work. On the other hand, however, we see that the rank of the sum is the sum of the ranks. We will come back to that point later. Let us now consider the matrix-vector multiplication.

Lemma 5.5 ([27])**.** *Let* $\mathbf{A} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathbf{A}}, \mathbf{U}^{\mathbf{A}}, \mathbf{B}^{\mathbf{A}}) \in \mathbb{R}^{\mathcal{J}\times\mathcal{I}}$ *be a matrix HTT and* $\mathbf{w} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathbf{w}}, \mathbf{U}^{\mathbf{w}}, \mathbf{B}^{\mathbf{w}}) \in \mathbb{R}^{\mathcal{I}}$ *be a vector HTT w.r.t. the same dimension tree* $\mathcal{T}_{\mathcal{N}}$*. Then, the matrix-vector product reads* $\mathbf{A}\mathbf{w} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathbf{A}\mathbf{w}}, \mathbf{U}^{\mathbf{A}\mathbf{w}}, \mathbf{B}^{\mathbf{A}\mathbf{w}})$*, where*

- $r^{\mathbf{A}\mathbf{w}}_t = r^{\mathbf{A}}_t r^{\mathbf{w}}_t$, $t \in \mathcal{T}_{\mathcal{N}}$;
- $V^{(i)}_t \in \mathbb{R}^{\mathcal{J}_t\times\mathcal{I}_t}$ *is chosen such that* $(U^{\mathbf{A}}_t)_i = vec(V^{(i)}_t)$ *for* $t \in \mathcal{L}(\mathcal{T}_N)$ *(reinterpretation of the columns of the leaf bases as matrices);*
- $U^{\mathbf{A}\mathbf{w}}_t = [V^{(1)}_t U^{\mathbf{w}}_t, \dots, V^{(r^{\mathbf{A}}_t)}_t U^{\mathbf{w}}_t] \in \mathbb{R}^{\mathcal{J}_t \times r^{\mathbf{A}}_t r^{\mathbf{w}}_t}$;
- $B^{\mathbf{A}\mathbf{w}}_t = B^{\mathbf{A}}_t \otimes B^{\mathbf{w}}_t$, $t \in \mathcal{I}(\mathcal{T}_N)$.

Again, the computational work is almost negligible. Note again, that the rank grows and is the product of the two original ranks. It should be noted that the HT-matrix $\mathbf{A}$ has to be represented in the *same* hierarchical order as the HT-vector $\mathbf{w}$ (i.e., w.r.t. the same dimension tree). This might require a conversion of a given matrix into the HTF given by $\mathbf{w}$, see Figure 3. If we can efficiently realize a conversion from $\mathbb{R}^{\mathcal{J}\times\mathcal{I}}$ to $\mathbb{R}^{\mathcal{J}\cdot\mathcal{I}}$, where $\mathcal{J}\cdot\mathcal{I} := \times_{j\in\mathcal{N}}(\mathcal{J}_j \times \mathcal{I}_j)$, then we can use a standard matrix-vector-multiplication for each $j \in \mathcal{N}$. Such a transformation

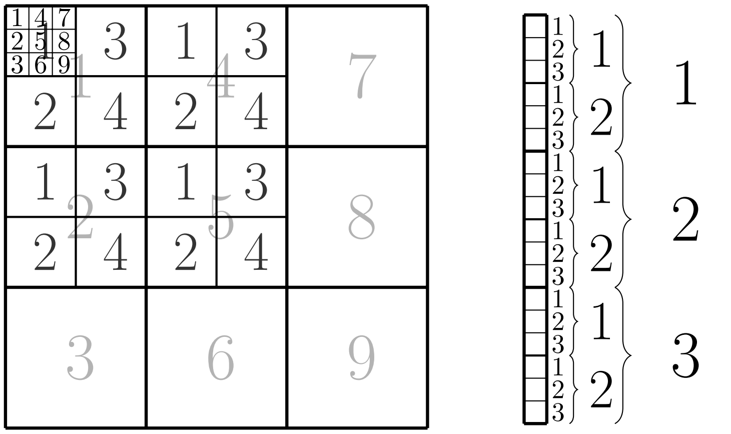

FIGURE 3. Conversion of a matrix. The left picture shows the hierarchical order of the matrix blocks which have to be converted into vectors, the right picture the hierarchical order of a vector that is multiplied with the matrix.

can be done by a reverse lexicographical ordering of the indices, i.e., applying the vec-routine to a matrix-tensor.

Lemma 5.6. *Let* $\mathbf{A} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathbf{A}}, \mathbf{U}^{\mathbf{A}}, \mathbf{B}^{\mathbf{A}}) \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ *be a square HTT-matrix, then* $\mathbf{D}_{\mathbf{A}} = diag(\mathbf{A}) \in \mathbb{R}^{\mathcal{I}}$ *is given as* $\mathbf{D}_{\mathbf{A}} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathbf{A}}, \mathbf{U}^{\mathbf{D}_{\mathbf{A}}}, \mathbf{B}^{\mathbf{A}})$*, where* $U_t^{\mathbf{D}_{\mathbf{A}}} \in \mathbb{R}^{\mathcal{I}_t \times r_t^{\mathbf{A}}}$ *is obtained from* $U_t^{\mathbf{A}} \in \mathbb{R}^{(\mathcal{I}_t\times\mathcal{I}_t)\times r_t^{\mathbf{A}}}$*,* $t \in \mathcal{T}_{\mathcal{N}}$*, by* $(U_t^{\mathbf{D}_{\mathbf{A}}})_{i,k} := (U_t^{\mathbf{A}})_{(i,i),k}$*,* $k = 1, \dots, r_t^{\mathbf{A}}$*,* $i \in \mathcal{I}_t$*.*

Lemma 5.7. *Let* $\mathscr{A} = \sum_{k=1}^{m} \bigotimes_{\mu\in\mathcal{N}} A_{k,\mu} \in \mathbb{R}^{\mathcal{K}}$ *be a Kronecker tensor, then* $\mathscr{A} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}^{\mathscr{A}}, \mathbf{U}^{\mathscr{A}}, \mathbf{B}^{\mathscr{A}})$*, where*

- $U_\mu^{\mathscr{A}} := A_{k,\mu}$*,* $k = 1, \dots, m$*,* $\mu \in \mathcal{L}(\mathcal{T}_{\mathcal{N}})$*,* $r_\mu^{\mathscr{A}} := k$*;*
- *for* $t \in \mathcal{I}(\mathcal{T}_{\mathcal{N}}) \setminus \{\mathcal{N}\}$*:* $B_t^{\mathscr{A}} \in \mathbb{R}^{m\times m\times m}$*,* $r_t^{\mathscr{A}} := m$ *and*

$$(B_t^{\mathscr{A}})_{i,j,\ell} := \begin{cases} 1, & \textit{if } i = j = \ell, \\ 0, & \textit{else;} \end{cases}$$

- $(B_{\mathcal{N}}^{\mathscr{A}})_{i,j,\ell} = \delta_{j,\ell}$*,* $B_{\mathcal{N}}^{\mathscr{A}} \in \mathbb{R}^{1\times k\times k}$*,* $r_{\mathcal{N}}^{\mathscr{A}} := 1$*.*

Remark 5.8. *Since the Kronecker sum in Definition A.6 is a special case of a Kronecker tensor, Lemma 5.7 also provides an HTF for Kronecker sums.*

Truncation of Tensors. We have seen that vector-vector addition and matrix-vector multiplication can be efficiently done for tensors in HTF. However, by Lemmata 5.4 and 5.5 the hierarchical rank grows with each addition or multiplication, so that only a certain number of such operations can be done in a numerical (iterative) scheme until the resulting HTTs get too large to be handled efficiently. Thus, a truncation is required.

The basic idea is to apply a singular value decomposition (SVD) on the matricizations $\mathbf{w}^{(t)}$ of the tensor $\mathbf{w}$ and restrict these to the dominant singular values. This can be realized without setting up $\mathbf{w}^{(t)}$ explicitly. Since, by construction, the columns of the mode frames U_t contain a basis for the column span of $\mathbf{w}^{(t)}$ there is a matrix $V_t \in \mathbb{R}^{\mathcal{I}^{(t)}\times r_t}$ such that $\mathbf{w}^{(t)} = U_t V_t^T$. Only the left singular vectors of the SVD of $\mathbf{w}^{(t)}$ are thus needed. Thus, the symmetric singular value decomposition of $\mathbf{w}^{(t)}(\mathbf{w}^{(t)})^T = U_t V_t^T V_t U_t^T =: U_t G_t U_t^T$ yields the same result. The matrices $G_t := V_t^T V_t \in \mathbb{R}^{r_t\times r_t}$ are called *reduced Gramian.* They are always of small size and can be computed recursively within the tree structure. In [20, Lemma 4.6] it is shown, that the reduced Gramians correspond to the accumulated transfer tensors for orthogonal HTTs. This statement also holds for general HTTs, see also [27].

The truncation of an HTT can then be computed by the computation of the QR decomposition $U_t = Q_t R_t$ for $t \in \mathcal{L}(T_d)$ or $(\hat{S}_{t_l}^T R_{t_l} \otimes \hat{S}_{t_r}^T R_{t_r}) B_t = Q_t R_t$ for $t \in \mathcal{I}(T_d)$. Subsequently, the symmetric eigenvalue decomposition of $R_t G_t R_t^T = S_t \Sigma^2 S_t^T$ is computed and the truncated matrix is then given by $U_t = Q_t \hat{S}_t$, $t \in \mathcal{L}(T_d)$, or $B_t = Q_t \hat{S}_t$, $t \in \mathcal{I}(T_d)$, where $\hat{S}_t$ is a restriction of S_t to the first r_t columns. Finally, we recall a well-known estimate for the truncation error.

Lemma 5.9. *Let* $\mathbf{w} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}, \mathbf{U}, \mathbf{B})$ *be an HTT and let* $\tilde{\mathbf{w}} = (\mathcal{T}_{\mathcal{N}}, \tilde{\mathbf{r}}, \tilde{\mathbf{U}}, \tilde{\mathbf{B}})$ *be the truncation of* $\mathbf{w}$ *such that* $\operatorname{rank}(\tilde{\mathbf{w}}^{(t)}) = \tilde{r}_t \le r_t$*. Then, for* $\mathcal{I}_t = \{1, \dots, n_t\}$*, we have*

$$\|\mathbf{w} - \tilde{\mathbf{w}}\|_2 \le \left(\sum_{i=r_t+1}^{n_t} \sigma_i^2 \right)^{1/2} \le \sqrt{2d-3} \inf_{\mathbf{v}\in\mathcal{H}(\tilde{\mathbf{r}})} \|\mathbf{w} - \mathbf{v}\|_2 ,$$

where $\mathcal{H}(\mathbf{r}) := \{\mathbf{v} = (\mathcal{T}_{\mathcal{N}}, \mathbf{r}, \mathbf{U}, \mathbf{B}) : \operatorname{rank}(\mathbf{v}^{(t)}) \leq r_t, t \in \mathcal{T}_{\mathcal{N}}\}$ *and* σ_i *are the sigular values of* $\mathbf{w}^{(t)}$ *such that* $\sigma_i \geq \sigma_j$ *if* $i < j$, $i = 1, \dots, n_t$.

Together with the vector-vector and matrix-vector addition as well as the truncation, the linear solver can be realized, [2]. The main difference is that after each addition or multiplication a truncation has to be made in order to keep the hierarchical rank small, which is not always easy to realize, [27]. For the HTF a Matlab implementation is available, [27]. We have developed an HT library in C++ in [36] based on BLAS, [8, 14, 15, 28] and LAPACK [1] routines, which are efficiently accessed via the FLENS interface, [29, 30, 31]. The reason for our implementation is that FLENS is the basis of the LAWA library (Library for Adaptive Wavelet Applications), [40]. The coupling of the proposed adaptive wavelet scheme with the HT structure could thus be efficiently realized. All subsequent numerical experiments have been performed with this software.

6. Numerical experiments

We report on numerical experiments, first (in order to describe some fundamental mechanisms) for a simple CDO and then in a more realistic framework.

6.1. **Encoding of defaults.**

We are given n assets and hence the state dimension is $N = 2^n$. Let $j \in \mathcal{N}$ and let $\mathtt{j} \in \mathtt{N} := \{0,1\}^n$ be its binary representation, i.e., a binary vector $(\mathtt{j}_1, \dots, \mathtt{j}_n)^T$ of length n, where $\mathtt{j}_i = 1$ means that asset number i is defaulted. For $j, k \in \mathcal{N}$ with binary representation $\mathtt{j}, \mathtt{k} \in \mathtt{N}$ and $\mathtt{j}|\mathtt{k}$ denoting the bitwise XOR, the number of ones in $\mathtt{j}|\mathtt{k}$ corresponds to the number of assets that change their state. This easy encoding is the reason why we used the labeling $\mathcal{N} = \{0, \dots, N-1\}$.

Once defaulted, always defaulted. For our numerical experiments, we assume for simplicity that an asset that is defaulted, stays defaulted for all future times, it cannot be reactivated (the theory and our implementation, however, is not restricted to this case). This means that $q^{j,k}(t,y) = 0$ if there exists an index $1 \leq i \leq n$ such that $\mathtt{j}_i = 1$, $\mathtt{k}_i = 0$. Both in the usual and in the binary ordering this last statement means $q^{j,k}(t,y) = 0$ if $j > k$, which in turns means that the $\mathbf{Q}(t,y) := (q^{j,k}(t,y))_{j,k \in \mathcal{N}}$ is an upper triangular matrix. Moreover, recall that $q^{j,j}(t,y) = -\sum_{k \in \mathcal{N}, k > j} q^{j,k}(t,y)$, so that $\mathbf{Q}$ can be stored as a strict lower triangular matrix, i.e., $\mathbf{Q} = (q^{j,k})_{k>j} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$.

Independent defaults. We will always assume that we have independent defaults. If defaults are independent, the transition of asset i from one state to another is independent of the state for all other assets as long as their states remain unchanged. Before we are going to formalize this, the following example may be helpful for the understanding.

Example 6.1. *If a portfolio consists of* 3 *assets, asset* 2 *defaults when changing from state* $0 = (000)_2$ *to state* $2 = (010)_2$*, from state* $1 = (001)_2$ *to state* $3 = (011)_2$*, from state* $4 = (100)_2$ *to state* $6 = (110)_2$ *and from state* $5 = (101)_2$ *to state* $7 = (111)_2$*. These are all transitions where only asset* 2 *defaults. When independent defaults are assumed, it follows that* $q^{0,2} = q^{1,3} = q^{4,6} = q^{5,7}$*. Note that* $0|2 = 1|3 = 4|6 = 5|7 = (010)_2$.

Let $j, k \in \mathcal{N}$, then $\mathtt{j}|\mathtt{k}$ indicates a state change of asset i if the i-th component of $\mathtt{j}|\mathtt{k}$ is one. Hence, if $\mathtt{j}_1|\mathtt{k}_1 = \mathtt{j}_2|\mathtt{k}_2$, then the same assets change their state. Since the change $1 \to 0$ is not allowed, we obtain that

$$\mathtt{j}_1|\mathtt{k}_1 = \mathtt{j}_2|\mathtt{k}_2 \quad \Rightarrow \quad q^{j_1,k_1} = q^{j_2,k_2}.$$

Only one default at a time. If only one asset can default at a time, the transition intensity $q^{j,k}$ is zero if $\mathtt{j}|\mathtt{k}$ has more than one "1". On the other hand, if $\mathtt{j}|\mathtt{k}$ has only one "1", then $\mathtt{j}|\mathtt{k}$ must be a power of 2. Since $q^{j,k}(t,y) = 0$ for $j > k$, it suffices to consider the case $k > j$ ($q^{j,j}$ is determined by the condition on the sum over all intensities). For $k > j$ being $\mathtt{j}|\mathtt{k}$ a power of 2 means that $\log_2(k-j) \in \mathbb{N}$. In this case, we have $\mathtt{j}|\mathtt{k} = 0|(\mathtt{k}|\mathtt{j})$, so that for $j, k \in \mathcal{N}$ we have

$$q^{j,k}(t,y) = \begin{cases} q^{0,k|j}(t,y), & \text{if } k > j \text{ and } \log_2(k-j) \in \mathbb{N}, \\ 0, & \text{otherwise.} \end{cases}$$

6.2. **A model problem.** The idea of our first numerical example is to showcase the numerical manageability, where the focus is on the combination of the multiwavelet components and the high dimensional Markov chain components. To this end, we start with a simplified CDO:

- The macroeconomic process Y is one dimensional with parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, that are constant in time. This implies that $\mathbf{A}(t) \equiv \mathbf{A}$ and $\mathbf{B}(t) \equiv \mathbf{B}$.
- The interest rate $r(t) \equiv r$ is constant in time and does not depend on the macroeconomic process Y.
- The state dependent parameters $q^{i,j}(t,y)$ and $c^j(t,y)$ are constant in time and do not depend on the macroeconomic process Y, i.e., $h_q(y) \equiv 1$ and $h_c(y) \equiv 1$, hence $q^{j,k}(t,y) \equiv \tilde{q}^{j,k}$ and $c^j(t,y) \equiv \tilde{c}^j$. This means that $\mathbf{Q}(t,y) \equiv \mathbf{Q} = \tilde{\mathbf{Q}}$, where $\tilde{\mathbf{Q}} = (\tilde{q}^{j,k})_{k>j;j,k\in\mathcal{N}}$.
- There is no recovery and no final payments, i.e., $a^{j,k}(t,y) \equiv 0$ for all $j,k \in \mathcal{N}$ and $a^j(y) = 0$ for $j \in \mathcal{N}$.
- There is only one tranche covering the entire CDO portfolio.

This means that all involved matrices are time-independent. In particular, we have $\mathbf{C}^{j,k}(t) \equiv \gamma_k^j((\psi_\lambda, \psi_\mu)_{0;\Omega})_{\lambda,\mu\in\mathcal{J}} = \gamma_k^j \mathbf{I}_{\mathcal{J}}$. Moreover $(\mathbf{M}^q)_{\lambda,\mu} = (h_q\psi_\lambda, \psi_\mu)_{0;\Omega} = \delta_{\lambda,\mu}$ and $\mathbf{D}(t) \equiv \mathbf{D} = (d^{j,k})_{j,k\in\mathcal{N}}$. Next, we have by $a^{j,k}(t,y) \equiv 0$ that $b^j(t) \equiv 0$, $\tilde{c}^j(t) \equiv \tilde{c}^j$, $g_\lambda^1 = g_\lambda^2 = (1, \psi_\lambda)_{0;\Omega}$ ($= 0$ for $|\lambda| > 0$) so that (4.3) simplifies to

$$(\mathbf{I}_{\mathcal{N}} \otimes \mathbf{I}_{\mathcal{J}})\dot{\mathbf{x}}(t) + [(\mathbf{I}_{\mathcal{N}} \otimes [\mathbf{A} + r\mathbf{I}_{\mathcal{J}}]) + (\mathbf{D} \otimes \mathbf{I}_{\mathcal{J}})]\,\mathbf{x}(t) = (-\tilde{\mathbf{c}}) \otimes ((1, \psi_\lambda)_{0;\Omega})_{\lambda\in\mathcal{J}}. \tag{6.1}$$

For later reference, recall that (4.2) in this case implies $d^{k,k} = -\sum_{m\in\mathcal{N}\setminus\{k\},m>k} \tilde{q}^{k,m}$. In turns, this means that

$$\mathbf{D} = \tilde{\mathbf{Q}} + \operatorname{diag}(\tilde{\mathbf{Q}}\,\mathbb{1}_{\mathcal{N}}), \quad \text{where} \quad \mathbb{1}_{\mathcal{N}} := (1, \dots, 1)^T \in \mathbb{R}^{\mathcal{N}}.$$

Note that even though $\mathbf{D}$ is time-independent, the huge dimension requires storage as an HT-matrix (in particular, it is impossible to store $\mathbf{D}$ directly). We use a standard implicit θ-scheme for the time-discretization of this Sylvester-type equation, [42]. The Barlets-Steward algorithm [5], is a well-known method for solving such Sylvester equations. It is based on a Schur decomposition. However, we cannot use this method here, since, to the best of our knowledge, there is no algorithm for the QR decomposition of HT-matrices available. Alternatively, an iterative scheme (CG, GMRES or BiCGStab) may be used. We have used BiCGStab as $\mathbf{D}$ is (in general) not symmetric. While generally GMRES yields faster convergence in terms of iteration numbers, it requires more computational steps and therefore, in the context of HT-matrices, more truncations are needed, which is computationally expensive. For systems with small condition numbers, BiCGStab requires only a few iterations, [41]. Using any iterative solver requires matrix-vector multiplications, here of the type $(\mathbf{I}_{\mathcal{N}} \otimes \mathbf{A} + \mathbf{D} \otimes \mathbf{I}_{\mathcal{J}})\mathbf{x}$, where $\mathbf{x} = \mathbf{x}_1 \otimes \mathbf{x}_2$ is also a Kronecker product of the appropriate dimension. Then, we obtain

$$(\mathbf{I}_{\mathcal{N}} \otimes \mathbf{A} + \mathbf{D} \otimes \mathbf{I}_{\mathcal{J}})(\mathbf{x}_1 \otimes \mathbf{x}_2) = \mathbf{x}_1 \otimes \mathbf{A}\mathbf{x}_2 + \mathbf{D}\mathbf{x}_1 \otimes \mathbf{x}_2,$$

which can be represented as a Kronecker product of an HT-matrix and a matrix. For details of the implementation, we refer to [36].

6.2.1. *Construction of the intensity matrix* $\mathbf{D}$. We describe the representation of $\mathbf{D}$ into HTF in case of independent defaults and only one default at a time. Recall that $\mathbf{D} = \tilde{\mathbf{Q}} + \operatorname{diag}(\tilde{\mathbf{Q}}\,\mathbb{1}_{\mathcal{N}})$ and $\tilde{\mathbf{Q}} = (\tilde{q}_{j,k})_{k>j} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$. Hence, we start by deriving a Kronecker sum representation for $\tilde{\mathbf{Q}}$.

Theorem 6.2. *In case of independent defaults and one default at a time, the matrix* $\tilde{\mathbf{Q}} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ *can be written as a Kronecker sum (see Definition A.6 below)*

$$\tilde{\mathbf{Q}} = \bigoplus_{k=1}^{n} \begin{pmatrix} 0 & q^{0,2^{n-k}} \\ 0 & 0 \end{pmatrix} = \sum_{k=1}^{n} \left\{ \bigotimes_{\ell=1}^{k-1} I_{2\times 2} \otimes \begin{pmatrix} 0 & q^{0,2^{n-k}} \\ 0 & 0 \end{pmatrix} \otimes \bigotimes_{\ell=k+1}^{n} I_{2\times 2} \right\}, \tag{6.2}$$

where $I_{2\times 2} \in \mathbb{R}^{2\times 2}$ *denotes the identity matrix of corresponding size.*

Proof. The k-th summand of the Kronecker sum on the right-hand side of (6.2) reads

$$\tilde{\mathbf{Q}}_k := \bigotimes_{\ell=1}^{k-1} I_{2\times 2} \otimes \begin{pmatrix} 0 & q^{0,2^{n-k}} \\ 0 & 0 \end{pmatrix} \otimes \bigotimes_{\ell=k+1}^{n} I_{2\times 2}.$$

It is readily seen that $\tilde{\mathbf{Q}}_k$ is a matrix having the entries $q^{0,n-k}$ at the positions

$$(2^{n-k+1}(\nu-1) + \mu,\ 2^{n-k+1}(\nu-1) + \mu + 2^{n-k}),$$

for $\nu = 1, \dots, 2^{k-1}$ and $\mu = 1, \dots, 2^{n-k}$, i.e., $2^{k-1} \cdot 2^{n-k} = 2^{n-1}$ entries. Note that 2^{n-1} is the number of all combinations with a state change of asset k. Since $2^{n-k+1}(\nu - 1) + \mu \neq 2^{n-k}$ for all possible ν and μ, we obtain that $\mathtt{j}|\mathtt{k}$ has exactly one "1" at position $i = n - k$. This, in turns, means that $q^{j,k} = q^{0,n-k}$. This shows that $\tilde{\mathbf{Q}}_k$ contains all transition intensities corresponding to asset k at the right positions.

It remains to show that the sum over all $\tilde{\mathbf{Q}}_k$ does not cause overlapping indices. Since $\bigoplus_{\ell=1}^{k-1} I_{2\times 2} \in \mathbb{R}^{2^{k-1}\times 2^{k-1}}$, each $\tilde{\mathbf{Q}}_k$ is a block matrix with blocks at different positions. Thus $\bigoplus_{k=1}^{n} \tilde{\mathbf{Q}}_k$ collects the transition intensities for all assets k. □

Having the Kronecker sum (6.2) at hand, the next step is to derive the HTF of $\mathbf{D}$. As we have seen in Lemma 5.7 and Remark 5.8, the HTF of $\tilde{\mathbf{Q}}$ can easily be derived from the Kronecker sum representation. Next, note that $\mathbb{R}^{\mathcal{N}} \ni \mathbb{1}_{\mathcal{N}} = \bigotimes_{k=1}^{n} (1,1)^T$, so that the HTF of $\tilde{\mathbf{Q}}\,\mathbb{1}_{\mathcal{N}}$ can be obtained via Lemma 5.6 and the HTF of $\mathbf{D}$ by Lemma 5.4. Finally, it is easily seen that $\tilde{\mathbf{c}} = \bigoplus_{k=1}^{n} (\tilde{c}_k, 0)^T$ so that the HTF of the right-hand side is easily be derived.

6.2.2. *Results.* We have used fictional data as marked data is generally not publicly available for CDOs. The fictional CDO portfolio consists of $n = 128$ assets, which is a reasonable size. The macroeconomic process is assumed to be one dimensional. This leads to a system of $N = 2^{128}$ coupled partial differential equations. The maturity is assumed to be $T = 1$ and the time interval is discretized into 20 time steps. For the Galerkin approximation, piecewise cubic L_2-orthonormal Multiwavelets with Dirichlet boundary conditions. We have fixed the lowest and highest level to $j_0 = 2$ and $J = 4$. This turned out to be sufficient in this example due to the smoothness of the right-hand side. For the θ-scheme, the parameter $\theta = 0.5$ has been chosen (Crank-Nicholson). To solve the linear system, BiCGStab has been used. The stopping criterion of the linear solver has been set as a relative error of the L_2-norm of the residual to 10^{-13} and HTTs are truncated to a rank of 5.

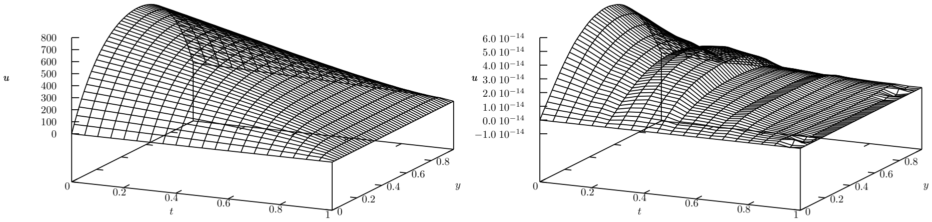

FIGURE 4. CDO portfolio value in a portfolio of 128 assets. First state (no defaults, left) and state (all firms have defaulted, right). Note the scaling of the vertical axis in the right figure.

In Figure 4, left, the portfolio value depending on the time t and the macroeconomic state y of the CDO in the first stage, where no firm has defaulted, is shown. Whenever a firm defaults, the Markov chain changes its state and the portfolio value jumps to a lower value due to the immediate loss of all future continuous payments. To illustrate the jumps of the portfolio value, when a default has occurred, the values of all states of a portfolio of 2 assets have been combined in Figure 5, left. The last stage is used for error analysis, since here the portfolio value has to be zero as all firms have defaulted. Figure 4 shows on the right the portfolio value in the last state of this simulation. This can be interpreted as the relative error arising from the Galerkin approximation with the L_2-orthonormal multiwavelets and the truncation of the HTTs after each addition or multiplication. It can be observed, that the relative error of this computation is smaller than 10^{-13}, which corresponds to the stopping criterion of the linear solver. In each time step, the BiCGStab algorithm took on average 54 iterations.

The computations were performed on a Dell XPS with Intel T9300 Dual Core CPU and 3GB storage and took about 1 hour, the runtime for different portfolio sizes is summarized in Figure 5, right. We observe a linear scaling with respect to n, i.e., only a logarithmic scaling compared to the number of equations $N = 2^n$.

6.3. **A realistic scenario.** The example presented in the previous section corresponds to a simplified and idealized situation. As we will show now, the extension to a realistic scenario (given sufficient data), is not too difficult. In fact:

- Observe, that time-dependent parameters do not affect the runtime of the pricing as the condition number of the matrix is not affected, assuming the parameters are sufficiently smooth in the time t.

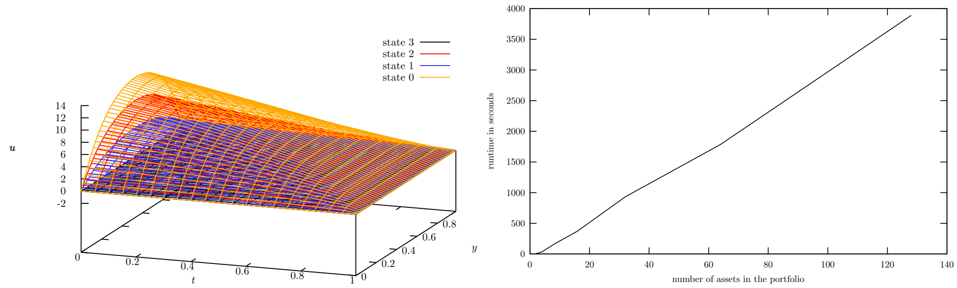

FIGURE 5. CDO portfolio value of all states in a portfolio of 2 assets (left) and runtime of the pricing of a CDO portfolio of $N \in \{2, 4, 8, 16, 32, 64, 128\}$ assets.

- The matrices $\mathbf{A}(t)$ and $\mathbf{D}(t)$ need to be setup for any time instant t^k. For $\mathbf{D}(t)$, this only amounts constructing an HTF exactly as described in the previous section. For the matrix $\mathbf{A}(t)$, we have to compute integrals of the type $(\nabla\psi_\mu, \mathbf{B}(t)\,\nabla\psi_\lambda)_{0;\Omega}$ and $(\boldsymbol{\alpha}(t)^T\,\nabla\psi_\mu, \psi_\lambda)_{0;\Omega}$ for any t^k followed by a transformation to HTF. All these operations can be performed efficiently.
- Space dependent parameters, however, will affect the condition of the linear system and possibly will require the use of a preconditioner. The construction of such a preconditioner in HTF will thus be discussed in the following subsection.

Furthermore, independent defaults are a very restrictive assumption. Other HTFs of dependency structures have to be developed for each given dependency structure. If no explicit HTF can be set up, the transition matrix can as well be approximated using the so-called *Black Box Algorithm*, [3, 18, 36].

6.3.1. *Preconditioning.* One of the features of wavelet Galerkin schemes is the availability of asymptotically optimal preconditioners, see e.g. [45, Ch. 6]. This means that the matrix $\mathbf{A}_\Lambda(t)$ from Section 4 has a uniformly bounded condition number, i.e., $\kappa_2(\mathbf{A}_\Lambda(t)) = \mathcal{O}(1)$ as $|\Lambda| \to \infty$. This, however, does not immediately imply that $\mathscr{A}_\Lambda(t)$ is well-conditioned. We propose to use a simple Jacobi-type preconditioner, i.e.,

$$\mathscr{D}_\Lambda(t) := \operatorname{diag}(\mathscr{A}_\Lambda(t))^{-1/2} = \Big[\big(\mathbf{I}_{\mathcal{N}} \otimes [\mathbf{A}_\Lambda(t) + r(t)\mathbf{I}_\Lambda] + \mathbf{D}(t) \otimes \mathbf{M}^q_\Lambda\big)^{-1/2}_{\lambda,\lambda}\Big]_{\lambda\in\Lambda} \tag{6.3}$$

One reason for this choice is the fact that the HTF of such a preconditioner can efficiently be derived. Moreover, the numerical performance has been quite satisfactory, at least in our experiments. The computation of the HTF of a diagonal matrix is provided by Lemma 5.6. The sum in (6.3) can be transformed into HTF by Lemma 5.4 so that we are left determining the HTF of the four matrices $\mathbf{I}_{\mathcal{N}}$, $\mathbf{D}_{\mathcal{N}}$ and $\mathbf{A}_\Lambda$, $\mathbf{M}_\Lambda$. The first two ones are trivial or have been derived above and for the second two ones we are using again the Black Box Algorithm. Finally, also the power $-1/2$ of a tensor can be transformed into HTF by the Black Box Algorithm.

6.3.2. *Computing CDO tranches.* So far, only single tranche portfolios were considered. However, in practice, a CDO is usually sold in tranches such that the first defaults only affect a certain tranche. Therefore, by construction, this is the riskiest tranche, the so called *equity tranche*. As soon as this first tranche has defaulted completely, subsequent defaults begin to affect a second trance. Therefore, this second tranche, called *mezzanine tranche*, is less risky than the first one. And finally, if this second tranche also has defaulted, the last tranche, the so called *senior tranche* is affected. Sometimes, these three tranches can be further split into sub-tranches. To compute the price of a CDO tranche, the cash flows of (1.1a) have to be adapted to the cash flows which affect the tranche under consideration. The construction of these adapted parameters is described in the following.

Given a portfolio of n assets with nominal values $\pi_1, \dots, \pi_n$, the S tranches are defined by their upper boundary b_s, $s = 1, \dots, S$, given in percentages $b_0 = 0 < b_1 < \dots < b_{S-1} < b_S = 1$ of the total portfolio nominal $\Pi := \sum_{i=1}^n \pi_i$. Let L^j be the accumulated loss in state $j \in \mathcal{N}$, i.e., $L^j = \sum_{i \in \mathscr{D}(j)} \pi_i$, where

$$\mathscr{D}(j) := \left\{ i \in \{1, \dots, n\} \;\middle|\; \exists x_k \in \{0,1\}, k = 1, \dots, n,\ x_i = 1 : j = \sum_{k=1}^n x_k 2^{k-1} \right\}$$

denotes the set of all defaulted firms in state j. Then, the cash flows are distributed as follows:

- The amount of the state dependent continuous payments c^j which is assigned to the s-th tranche, $s = 1, \dots, S$, in state $j \in \mathcal{N}$ can be computed as the percentage of the nominal of the tranche divided by the accumulated nominals of the assets not in default:
$$\tilde{c}_s^j(t) = \begin{cases} \tilde{c}^j(t) \frac{\Pi(b_s - b_{s-1})}{\Pi - L^j} & \text{if } L^j < \Pi b_{s-1} \\ \tilde{c}^j(t) \frac{\Pi(1-b_s) - L^j}{\Pi - L^j} & \text{if } \Pi b_{s-1} \le L^j < \Pi b_s, \\ 0 & \text{otherwise.} \end{cases} \tag{6.4}$$
- The final payments u_T^j are distributed to the tranches in the same way as the continuous payments $\tilde{c}$, i.e., (6.4) holds with $\tilde{c}_s^j(t)$ replaced by $u_{T,s}^j$, the final payment of tranche number s.
- The recovery payments are paid out as a single payment to the tranche in which the default occurred.

  If several tranches are affected, the recovery is paid out proportional to their nominals. This means
$$\tilde{a}_s^{j,k}(t) = \begin{cases} \tilde{a}^{j,k}(t) & \text{if } \Pi b_{s-1} < L^j < L^k \le \Pi b_s, \\ \tilde{a}^{j,k}(t) \frac{L^k - \Pi b_{s-1}}{L^k - L^j} & \text{if } L^j \le \Pi b_{s-1} < L^k \le \Pi b_s, \\ \tilde{a}^{j,k}(t) \frac{\Pi b_s - \Pi b_{s-1}}{L^k - L^j} & \text{if } L^j \le \Pi b_{s-1} < \Pi b_s < L^k, \\ \tilde{a}^{j,k}(t) \frac{\Pi b_s - L^j}{L^k - L^j} & \text{if } \Pi b_{s-1} < L^j \le \Pi b_s < L^k. \end{cases}$$
  In this setting only recoveries are considered, when a default occurs. However, the model also allows for firms to be in default only temporarily. This means, there can also be payments if $L^k < L^j$. These cases are omitted in the following as they can be handled exactly as the cases where $L^j < L^k$.

The difficulty of computing the payoffs of the s-th tranche is that certain specific states within the huge amount of states of the Markov chain have to be found. Therefore, we define vectors $\mathbb{1}_s^<$, $\mathbb{1}_s^=$ and $\mathbb{1}_s^>$, by

$$(\mathbb{1}_s^<)_j := \chi_{\{L^j < \Pi b_{s-1}\}}, \quad (\mathbb{1}_s^=)_j := \chi_{\{\Pi b_{s-1} \le L^j < \Pi b_s\}}, \quad (\mathbb{1}_s^>)_j := \chi_{\{L^j \ge \Pi b_s\}}, \quad j \in \mathcal{N}.$$

In order to explain the realization of some required operations, we introduce the following short-hand nottaion for HTTs. Let $\mathbf{w} \in \mathbb{R}^{\mathcal{K}}$ be a tensor with HTF $\mathbf{w} =$

$(\mathcal{T}_{\mathcal{N}}, \mathbf{r}, \mathbf{U}, \mathbf{B})$, then we abbreviate by $\mathcal{H}(\mathbf{w})$ its HTF without specifying the quantities involved in the HTF. For $\mathbf{A} \in \mathbb{R}^{\mathcal{K}\times\mathcal{K}}$ and $\mathbf{b} \in \mathbb{R}^{\mathcal{K}}$, we indicate by $\mathcal{H}(\mathbf{A})\,\mathcal{H}(\mathbf{u}) = \mathcal{H}(\mathbf{b})$ that $\mathbf{u} \in \mathbb{R}^{\mathcal{K}}$ is determined as the solution of the linear system $\mathbf{Au} = \mathbf{b}$ but only with numerical routines using the HTF-variants. Finally, we abbreviate $\mathcal{D}(\mathbf{w}) \coloneqq \mathcal{H}(\mathrm{diag}(\mathbf{w})) \in \mathbb{R}^{\mathcal{K}}$ for $\mathbf{w} \in \mathbb{R}^{\mathcal{K}}$, where $(\mathrm{diag}(\mathbf{w}))_{\mathbf{i},\mathbf{j}} \coloneqq \delta_{\mathbf{i},\mathbf{j}}\,\mathbf{w}_{\mathbf{i}}$, $\mathbf{i} \in \mathcal{K}$.

Now, denote $\mathbf{L} \coloneqq (L^j)_{j\in\mathcal{N}}$. Then, we need to compute the reciprocal value of each component of the vector $\mathbf{R} \coloneqq \Pi - \mathbf{L} \in \mathbb{R}^{\mathcal{N}}$ in HTF denoted by $\mathbf{R}^{(-1)}$. This can be achieved by solving the linear system $\mathcal{D}(\mathcal{H}(\mathbf{R}))x = \mathcal{H}(\mathbb{1})$ for x. Then $\mathcal{H}(\mathbf{R}^{(-1)}) \coloneqq x$ and

$$\begin{aligned}\mathcal{H}(\tilde{c}_s(t)) = {} & \mathcal{D}(\mathcal{H}(\tilde{c}(t))\mathcal{D}(\mathcal{H}(\mathbf{R}^{-1}))\mathcal{H}(\mathbb{1}_\mu^<)\Pi(b_s - b_{s-1}) \\ & + \mathcal{D}(\mathcal{H}(\tilde{c}(t))\mathcal{D}(\mathcal{H}(\mathbf{R}^{-1}))\mathcal{D}(\mathcal{H}(\mathbb{1}_\mu^=))(\Pi(1-b_s)\mathcal{H}(\mathbb{1}) - \mathcal{H}(\mathbf{L})).\end{aligned}$$

We obtain a similar formula for the final payments $(u_{T,s}^j)_{j\in\mathcal{N}}$. The HTF of the matrix $\tilde{\mathbf{A}}_s(t) \coloneqq (\tilde{a}_s^{j,k}(t))_{j,k\in\mathcal{N}}$ can be set up similarly. Therefore, the matrix $\mathbf{S} \coloneqq (L^k - L^j)_{j,k\in\mathcal{N}}$ is required. The HTF of this matrix can be obtained by $\mathcal{H}(\mathbf{S}) = \mathcal{H}(\mathbb{1}\cdot\mathbb{1}^T)\mathcal{D}(\mathcal{H}(\mathbf{L})) - \mathcal{D}(\mathcal{H}(\mathbf{L}))\mathcal{H}(\mathbb{1}\cdot\mathbb{1}^T)$. As before, the HTF of the matrix $\mathbf{S}^{(-1)}$ containing the reciprocals of the entries of $\mathbf{S}$ can be found by solving $\mathcal{D}(\mathbf{S})x = \mathcal{H}(\mathbb{1})$ for x and setting $\mathcal{H}(\mathbf{S}^{(-1)}) \coloneqq x$. Defining finally $\mathscr{A}(\mathbf{v},\mathbf{w}) \coloneqq \mathcal{D}(\mathcal{H}(\mathbf{v}))\mathcal{H}((\tilde{a}^{i,j}(t))_{i,j\in\mathcal{N}})\mathcal{D}(\mathcal{H}(\mathbf{w}))$, we obtain

$$\begin{aligned}\mathcal{H}(\tilde{\mathbf{A}}_s(t)) = {} & \mathscr{A}(\mathbb{1}_s^=, \mathbb{1}_s^=) \\ & + \mathcal{D}(\mathscr{A}(\mathbb{1}_s^<, \mathbb{1}_s^=))\mathcal{D}(\mathcal{H}(\mathbf{S}^{(-1)}))(\mathcal{H}(\mathbb{1}\cdot\mathbb{1}^T)(\mathcal{D}(\mathcal{H}(\mathbf{L})) - \mathcal{D}(\mathcal{H}(\mathbb{1}))\Pi b_{s-1})) \\ & + \mathcal{D}(\mathscr{A}(\mathbb{1}_s^<, \mathbb{1}_s^>))\mathcal{D}(\mathcal{H}(\mathbf{S}^{(-1)}))(\mathcal{H}(\mathbb{1}\cdot\mathbb{1}^T)(\mathcal{D}(\mathcal{H}(\mathbb{1}))\Pi(b_s - b_{s-1})) \\ & + \mathcal{D}(\mathscr{A}(\mathbb{1}_s^=, \mathbf{1}_s^>))\mathcal{D}(\mathcal{H}(\mathbf{S}^{(-1)}))(\mathcal{H}(\mathbb{1}\cdot\mathbb{1}^T)(\mathcal{D}(\mathcal{H}(\mathbb{1}))\Pi b_s) - \mathcal{D}(\mathcal{H}(\mathbf{L}))).\end{aligned}$$

With these adapted payments, the value of a portfolio tranche can now be determined. A key point of this approach is the construction of the HTF of the vectors $\mathbb{1}_s^<$, $\mathbb{1}_s^=$ and $\mathbb{1}_s^>$. This can be computed by

$$\begin{aligned}\mathcal{H}(\mathbb{1}_s^<) &\coloneqq \max\{\Pi b_{s-1}\mathcal{H}(\mathbb{1}) - \mathcal{H}(\mathbf{L}), 0\}\,(\Pi b_{s-1}\mathcal{H}(\mathbb{1}) - \mathcal{H}(\mathbf{L}))^{-1}, \\ \mathcal{H}(\mathbb{1}_s^>) &\coloneqq \max\{\mathcal{H}(\mathbf{L}) - \Pi b_s\mathcal{H}(\mathbb{1}), 0\}\,(\mathcal{H}(\mathbf{L}) - \Pi b_s\mathcal{H}(\mathbb{1}))^{-1}, \\ \mathcal{H}(\mathbb{1}_s^=) &\coloneqq \mathcal{H}(\mathbb{1}) - \mathcal{H}(\mathbb{1}_s^<) - \mathcal{H}(\mathbb{1}_s^>).\end{aligned}$$

The component-wise maximum $\max\{\mathcal{H}(\cdot), 0\}$ of any HT-vector can be determined by the relation $\max\{\mathcal{H}(\cdot), 0\} = \frac{1}{2}(\mathcal{H}(\cdot) + |\mathcal{H}(\cdot)|)$. The absolute value $|\mathcal{H}(\mathbf{w})|$ can be computed by the component-wise Newton iteration

$$\mathcal{H}(\mathbf{w}^{(n+1)}) = \mathcal{H}(\mathbf{w}^{(n)}) - D((\mathcal{H}(\mathbf{w}^{(n)}))^{-1})\mathcal{D}(\mathcal{H}(\mathbf{w}))\mathcal{H}(\mathbf{w}).$$

Note, that each iteration step requires the component-wise inversion of an HT-vector, i.e., the solution of a linear system. Let ν be such that $\mathbf{w}^{(\nu)}$ is of the desired accuracy, then $|\mathcal{H}(\mathbf{w})| \approx \mathcal{H}(\mathbf{w}^{(\nu)})$. Essentially, this corresponds to $|x| = \sqrt{x^2}$, the well-known Babylonian method, which converges quadratically for non-zero values. Moreover, for vectors $\mathbf{w}^{(0)}$ with positive entries, it always converges to the positive solution.

Appendix A. The Kronecker Product

The Kronecker product is a well-known technique when dealing with high dimensional problems, as it often allows the decomposition of a high dimensional problem into a product of problems of low dimension. The following facts can be found in [39, 47].

Definition A.1. *The* Kronecker product $A \otimes B \in \mathbb{R}^{m_A m_B \times n_A n_B}$ *of two matrices* $A \in \mathbb{R}^{m_A \times n_A}$ *and* $B \in \mathbb{R}^{m_B \times n_B}$ *is defined by* $(A \otimes B)_{(\mu_1 - 1)m_B + \mu_2, (\nu_1 - 1)n_B + \nu_2} := A_{\mu_1,\nu_1} B_{\mu_2,\nu_2}$.

We note that the above definition and all subsequent properties can also be extended to (infinite) countable index sets $\mathcal{J}$.

Lemma A.2. *Let* $A \in \mathbb{R}^{m_A \times n_A}$, $B \in \mathbb{R}^{m_B \times n_B}$, $C \in \mathbb{R}^{m_C \times n_C}$, $D \in \mathbb{R}^{m_D \times n_D}$ *and* $\nu \in \mathbb{R}$. *Then,*

(1) $(A \otimes B)^T = A^T \otimes B^T$,
(2) $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$,
(3) $(A \otimes B)(C \otimes D) = AC \otimes BD$, *if* $n_A = m_C$ *and* $n_B = m_D$,
(4) $A \otimes (B \otimes C) = (A \otimes B) \otimes C$,
(5) $A \otimes (B + C) = A \otimes B + A \otimes B$ *and* $(B + C) \otimes A = B \otimes A + C \otimes A$,
(6) $\nu(A \otimes B) = (\nu A) \otimes B = A \otimes (\nu B)$,
(7) $\mathrm{tr}(A \otimes B) = \mathrm{tr}(A)\mathrm{tr}(B)$,
(8) *if* $m_A = n_A$ *and* $m_B = n_B$, *then* $\det(A \otimes B) = (\det(A))^{n_B} (\det(B))^{n_A}$,
(9) $\mathrm{rank}(A \otimes B) = \mathrm{rank}(A)\mathrm{rank}(B)$.

Definition A.3. *Let* $A \in \mathbb{R}^{m_A \times n_A}$, *then its* vectorization *is defined as* $\mathrm{vec}(A) = (A_{\cdot,1}^T, \dots, A_{\cdot,n_A}^T)^T$, *where* $A_{\cdot,i}$, $i \in \{1, \dots, n_A\}$, *is the i-th column of the matrix A.*

Lemma A.4 ([47, (2)])**.** *Let* $A \in \mathbb{R}^{m_A \times n_A}, B \in \mathbb{R}^{m_B \times n_B}, C \in \mathbb{R}^{m_B \times m_A}, X \in \mathbb{R}^{n_B \times n_A}$, *then* $(A \otimes B)\mathrm{vec}(X) = \mathrm{vec}(C) \Leftrightarrow BXA^T = C$.

Definition A.5. *Let* $A_{k,\mu} \in \mathbb{R}^{\mathcal{K}_\mu}$, $\mu \in \mathcal{N}$, $k = 1, \dots, m$, $m \in \mathbb{N}$. *Then, we define the* Kronecker tensor *as*

$$\mathscr{A} := \sum_{k=1}^{m} \bigotimes_{\mu \in \mathcal{N}} A_{k,\mu} \in \mathbb{R}^{\mathcal{K}}.$$

Definition A.6. *Let* $A_1, \dots, A_m$ *be matrices* $A_k \in \mathbb{R}^{n_k \times n_k}$. *Then, the* Kronecker sum *of* $A_1, \dots, A_m$ *is defined as*

$$\bigoplus_{k=1}^{m} A_k := \sum_{k=1}^{m} \left\{ \bigotimes_{\ell=1}^{k-1} I_{n_\ell \times n_\ell} \otimes A_k \otimes \bigotimes_{\ell=k+1}^{m} I_{n_\ell \times n_\ell} \right\} \in \mathbb{R}^{(n_1 \cdots n_m) \times (n_1 \cdots n_m)}. \tag{A.1}$$

Obviously, the Kronecker sum is a special case of the Kronecker tensor, where $A_{k,\mu} = \delta_{k,\ell} A_k + (1 - \delta_{k,\ell}) I_{n_\ell \times n_\ell}$

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide.* Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[2] J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numerical Linear Algebra with Applications*, 20(1):27–43, 2013.

[3] J. Ballani, L. Grasedyck, and M. Kluge. Black box approximation of tensors in hierarchical tucker format. *Linear Algebra and its Applications*, 438(2):639–657, 2013.

[4] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.

[5] R. H. Bartels and G. W. Stewart. Solution of the matrix equation ax + xb = c [f4]. *Commun. ACM*, 15(9):820–826, September 1972.

[6] P. Beaudry and A. Lahiri. Risk allocation, debt fueled expansion and financial crisis. Working Paper 15110, National Bureau of Economic Research, June 2009.

[7] T. R. Bielecki and M. Rutowski. *Credit Risk: Modeling, Valuation and Hedging*. Springer, 2002.

[8] L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley. An updated set of basic linear algebra subprograms (blas). *ACM Trans. Math. Softw.*, 28(2):135–151, June 2002. http://www.netlib.org/blas/.

[9] C. Bluhm and L. Overbeck. *Structured Credit Portfolio Analysis, Baskets and CDOs*. Chapman and Hall, 2007.

[10] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods for elliptic operator equations: convergence rates. *Math. Comp.*, 70(233):27–75, 2001.

[11] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods. II. Beyond the elliptic case. *Found. Comput. Math.*, 2(3):203–245, 2002.

[12] A. Cohen, W. Dahmen, and R. Devore. Adaptive wavelet schemes for nonlinear variational problems. *SIAM J. Numer. Anal.*, 41(5):1785–1823, 2003.

[13] T. J. Dijkema, C. Schwab, and R. Stevenson. An adaptive wavelet method for solving high-dimensional elliptic pdes. *Constructive Approximation*, 30:423–455, 2009.

[14] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff. A set of level 3 basic linear algebra subprograms. *ACM Trans. Math. Softw.*, 16(1):1–17, 1990. http://www.netlib.org/blas/.

[15] J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson. An extended set of fortran basic linear algebra subprograms. *ACM Trans. Math. Softw.*, 14:1–17, 1986. http://www.netlib.org/blas/.

[16] G. Donovan, J. Geronimo, and D. Hardin. Orthogonal polynomials and the construction of piecewise polynomial smooth wavelets. *SIAM J. Math. Anal.*, 30(5):1029–1056, 1999.

[17] G. P. Dwyer. Financial innovation and the financial crisis of 2007-2008. *Papeles de Economia Espana*, 130, 2012.

[18] M. Epsig, L. Grasedyck, and W. Hackbusch. Black box low tensor-rank approximation using fiber-crosses. *Constructive Approximation*, 30(3):557–597, 2009.

[19] J. S. Geronimo, D. P. Hardin, and P. R. Massopust. Fractal functions and wavelet expansions based on several scaling functions. *J. Approx. Theory*, 78(3):373–401, 1994.

[20] L. Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[21] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, 2009.

[22] N. Hilber, S. Kehtari, C. Schwab, and C. Winter. Wavelet finite element method for option pricing in highdimensional diffusion market models. Research Report 01, ETH Zürich, 2010.

[23] R. A. Jarrow. The role of abs, cds and cdos in the credit crisis and the economy. *Working Paper*, 2012.

[24] S. Kestler. *On the Adaptive Tensor Product Wavelet Galerkin Method with Applications in Finance*. PhD thesis, Ulm University, 2013.

[25] S. Kestler, K. Steih, and K. Urban. An efficient space-time adaptive wavelet galerkin method for time-periodic parabolic partial differential equations. Research Report 06, Ulm University, Fac. of Mathematics and Economics, August 2013.

[26] H. Kraft and M. Steffensen. Bankruptcy, counterparty risk, and contagion. *Review of Finance*, 11:209–252, 2006.

[27] D. Kressner and C. Tobler. htucker - a matlab toolbox for tensors in hierarchical tucker format. *Preprint*, 2012.

[28] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Trans. Math. Softw.*, 5(3):308–323, September 1979. http://www.netlib.org/blas/.

[29] M. Lehn. *FLENS - a flexible library for efficient numerical solutions*. PhD thesis, Ulm University, 2008.

[30] M. Lehn. Flexible Library for Efficient Numerical Solutions (FLENS), 2013. http://apfel.mathematik.uni-ulm.de/~lehn/FLENS/index.html.

[31] M. Lehn, A. Stippler, and K. Urban. Flens - a flexible library for efficient numerical solutions. In *Proceedings of Equadiff 11*, pages 467–473. Comenius University Press, 2007.

[32] R.H. Nochetto, K.G. Siebert, and A. Veeser. Theory of adaptive finite element methods: an introduction. In R. DeVore and A. Kunoth, editors, *Multiscale, nonlinear and adaptive approximation*, pages 409–542. Springer, Berlin, 2009.

[33] A. Nouy. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 199(23-24):1603–1626, 2010.

[34] D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. PhD thesis, Technische Universität München, 2010.

[35] C. Reisinger and G. Wittum. Efficient hierarchical approximation of high-dimensional option pricing problems. *SIAM J. Sci. Comput.*, 29(1):440–458, January 2007.

[36] A. J. Rupp. *High Dimensional Wavelet Methods for Structured Financial Products*. PhD thesis, University of Ulm, 2013.

[37] C. Schwab and R. Stevenson. Space-time adaptive wavelet methods for parabolic evolution problems. *Math. Comp.*, 78(267):1293–1318, 2009.

[38] C. Schwab and R. P. Stevenson. Fast evaluation of nonlinear functionals of tensor product wavelet expansions. *Numer. Math.*, 119(4):765–786, 2011.

[39] W. H. Steeb and T. K. Shi. *Matrix Calculus and Kronecker Product With Applications and C++ Programs*. World Scientific Publishing Company, Incorporated, 1997.

[40] A. Stippler. Lawa - library for adaptive wavelet applications, 2013. ”http://lawa.sourceforge.net/index.html”.

[41] F. D. Swesty, D. C. Smolarski, and P. E. Syalor. A comparison of algorithms for the efficient solution of the linear systems arising from multigroup flux-limited diffusion problems. *The Astrophysical Journal*, 153:369–387, 2004.

[42] J. J. Sylvester. Sur l'equations en matrices $px = xq$. *Comptes Redus Acad. Sci. Paris*, 99:67–71, 1884.

[43] J. M. Tavakoli. *Collateralized Debt Obligations and Structured Finance*. John Wiley and Sons, Inc., 2003.

[44] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[45] K. Urban. *Wavelet Methods for Elliptic Partial Differential Equations*. Oxford University Press, 2009.

[46] K. Urban and A. T. Patera. A new error bound for reduced basis approximation of parabolic partial differential equations. *C.R. Math.*, 350(3–4):203 – 207, 2012.

[47] C. F. van Loan. The ubiquitous kronecker product. *Journal of Computational and Applied Mathematics*, 123(1-2):85–100, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

Rüdiger Kiesel Lehrstuhl für Energiehandel und Finanzdienstleistungen, University of Duisburg-Essen, Universitätsstr. 12, D-45117 Essen, Germany
E-mail address: ruediger.kiesel@uni-due.de

Andreas Rupp, Institute for Numerical Mathematics, University of Ulm, Helmholtzstrasse 20, D-89069 Ulm, Germany
E-mail address: andreas.rupp@uni-ulm.de

Karsten Urban, Institute for Numerical Mathematics, University of Ulm, Helmholtzstrasse 20, D-89069 Ulm, Germany
E-mail address: karsten.urban@uni-ulm.de