VALUATION OF GUARANTEED MINIMUM ACCUMULATION BENEFITS (GMAB) WITH PHYSICS INSPIRED NEURAL NETWORKS

Donatien Hainaut

LIDAM Discussion Paper ISBA
2023 / 29

UCLouvain

LIDAM
ISBA | Louvain Institute of Data Analysis and Modeling in economics and statistics

ISBA

Voie du Roman Pays 20 - L1.04.01

B-1348 Louvain-la-Neuve

Email : lidam-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/isba/publication.html

Valuation of guaranteed minimum accumulation benefits (GMAB) with physics inspired neural networks

Donatien Hainaut*
UCLouvain, LIDAM

September 18, 2023

Guaranteed Minimum Accumulation Benefits (GMAB) are retirement savings vehicles which protect the policyholder against downside market risk. This article proposes a valuation method of these contracts based on physics inspired neural networks (PINN's), in presence of multiple financial and biometric risk factors. A PINN integrates principles from physics into its learning process to enhance its efficiency in solving complex problems. In this article, the driving principle is the Feynman-Kac (FK) equation which is a partial differential equation (PDE) ruling the GMAB price in an arbitrage-free market. In our context, the FK PDE depends on multiple variables and is hard to solve by classical finite difference approximations. In comparison, PINN's constitute a efficient alternative which furthermore can evaluate GMAB's with various specifications without retraining. To illustrate this, we consider a market with four risk factors. We first find a closed form expression for the GMAB that serves as benchmark for the PINN. Next, we propose a scaled version of the FK equation that we solve with a PINN. Pricing errors are analyzed in a numerical illustration.

1 Introduction

A physics-inspired neural network (PINN) integrates principles from physics into its learning process. It enhances its efficiency in solving complex scientific problems. Compared to existing approaches, a PINN mitigates the need for large datasets and provides accurate predictions even with sparse or noisy measurements. Researchers have employed PINN's to tackle a diverse range of problems, including fluid dynamics, solid mechanics, heat transfer, and quantum mechanics. In this work, we show that PINN's can be used for the valuation and design of particular variable annuities (VA's), called Guaranteed Minimum Accumulation Benefits (GMAB). This retirement savings vehicle provides at expiry, the maximum from the account value and the guaranteed amount in case of survival. The proposed model includes the following risk factors: interest rate, equity and mortality rates. We consider four classes of assets: cash, zero-coupon bonds, stocks and mortality linked bonds. Our objective is to estimate a single neural network capable to price GMAB's with various specifications including investment strategies and maturities.

To summarize, a PINN is an approximated solution of a non-linear partial differential (PDE)

*Corresponding author. Postal address: Voie du Roman Pays 20, 1348 Louvain-la-Neuve (Belgium). E-mail to: donatien.hainaut(at)uclouvain.be

explaining the dynamics of a model. The neural network is trained by minimizing the error of approximation at sampled points inside the PDE domain and on its border. This method finds its roots in the article of Lee and Kang (1990) and has been later applied to non-linear partial differential equations. For instance, Raissi M. et al. (2019) use PINN's for solving two main classes of mathematical problems: data-driven solution and data-driven discovery of partial differential equations. In physics, Carleo and Troyer (2017) or Cai (2018) use PINN's to approximate with high accuracy quantum many-body wave functions. We refer the interested reader to Cuomo et al. (2022) for a literature review on other recent developments and applications of PINN's.

The literature about the valuation of financial and actuarial liabilities with neural network is relatively recent. Hejazi and Jackson (2016) develop a neural network to price and estimate the "Greeks" of a large portfolio of variable annuities. Doyle & Groendyke (2019) price and hedge equity-linked contracts with neural networks. They build a dataset of variable annuity prices with various features, using Monte-Carlo simulations. The neural network is next estimated by minimizing the errors of prediction with a scaled conjugate gradient descent. Sirignano and Spiliopoulos (2018) propose a Deep Galerkin Method (DGM) based on a network with an architecture inspired from long short-term memory (LSTM) neural cells. Their approach is tested on a class of high-dimensional free boundary partial differential equations (PDE's) and on a high-dimensional Hamilton-Jacobi-Bellman PDE and Burgers' equation. Gatta et al. (2018) evaluate a suitable PINN for the evaluation of American multi-asset options and propose a novel algorithmic trick for free boundary training. Al-Aradi et al.(2022) extend the DGM in several directions for solving Fokker-Planck and Hamilton-Jacobi-Bellman equations. Recently, Jiang et al. (2023) prove under mild assumptions, the convergence of Deep Galerkin and PINNs method for solving PDE. Glau and Wunderlich (2022) formalise and analyse the deep parametric PDE method to solve high-dimensional parametric partial differential equations.

A close alternative to PINN is developed in Weinan et al. (2017) who solve with neural networks, parabolic PDE's in high dimension using their connection to BSDE's. This approach is based on two separate networks for the solution and its gradient. They apply their algorithm to price option in a multivariate Black and Scholes model. Beck et al. (2019) introduce a similar method for solving high-dimensional PDE based on a connection between fully nonlinear second-order PDEs and second-order backward stochastic differential equations. They illustrate the accuracy of their method with the Black and Scholes and Hamilton-Jacobi-Bellman equations. Using this principle, Barigou and Delong (2022) evaluate equity-linked life insurances with neural networks solving a backward stochastic differential equation (BSDE), based on previous results of Delong et al. (2019). Compared to PINN's, BSDE methods rely on two networks instead of one and any modification of contract specifications requires a retraining.

Another valuation approach based on neural networks relies on simulations. Buehler et al. (2019) present a framework for pricing and hedging derivatives based on neural network using simulated sample paths of risk factors. The neural network approximates in this case the optimal investment strategy and the value is determined by averaging discounted payoffs obtained by simulations. In a similar manner, Horvath et al. (2021), study the performance of deep hedging under rough volatility models. Biagini et al. (2023) develop a neural network approximation using simulations for the superhedging price and replicating strategy. As for BSDE approaches, any modification of contract specifications requires a retraining. We refer the reader to Germain et al. (2021) for a review of algorithms based on neural networks for stochastic control and PDE's in finance.

In the financial or insurance industry, four dominant numerical methods are used for pricing contingent claim contracts: Monte-Carlo (MC) simulations, bi- or trinomial trees, PDE solving

or Fast Fourier Transform (FFT) inversion. For more than 2 risks factors, trees, PDE solving or FFT are too complex and MC simulations is the only viable alternative. In comparison, the valuation with a PINN does not require to simulate sample paths of underlying risk factors. Furthermore, once the PINN is trained, the valuation is quasi-instantaneous. A last interesting feature is the possibility to parameterize a PINN with contract features such as asset-mix or contract durations. This offers a quick solution to understand the impact of Asset-Liability Management (ALM) on pricing without training again the network for each setting. This is a serious advantage compared to BSDE or deep hedging approaches.

The contribution of this article are the following. Firstly, we develop a closed-form expression for the price of a GMAB with participation in a market with five classes of assets, including a mortality bond for hedging the exposure to longevity risk. This analytical formula allows us to evaluate the accuracy of the pricing with PINN's on a large set of contracts without any recourse to Monte-Carlo simulations. Next, we propose a particular type of neural networks in which intermediate layers are fed with the output of the previous layer and the initial input vector. Such a network better replicates the non-linear behaviour of GMAB prices than classical feed-forward network. Thirdly, we consider risk factors (interest and mortality rates) driven by mean reverting processes whereas the existing literature on PINN mainly focuses on high-dimensional geometric processes. Fourthly, we train the network for various investment strategies, asset-mix and contract durations. In this sense, our model is parametric and allows a great flexibly compared to BSDE or deep hedging methods that require a retraining for each setting.

The outline of the article is the following. The next section introduces the financial and biometric risk factors. We consider a multivariate Brownian setting to preserve the analytical tractability needed to benchmark the PINN. In Section 3, we develop the closed-form expression of zero-coupon bonds, survival probabilities and pure endowments. The specifications of the GMAB are detailed in Section 4. We assume that assets are invested into a portfolio of cash, bonds, stocks and into mortality bonds. We provide in this framework, the analytical formula for pricing the GMAB. Section 5 presents a scaled version of the Feynman-Kac equation fulfilled by GMAB prices. In section 6, we develop the architecture of the neural network solving this PDE and explain the training procedure. The last section provides a detailed analysis of pricing errors on a validation set of 500 000 contracts with various features.

2 Risk factors

Our objective is to illustrate the utility of PINN's for valuing and setting up an investment strategy for GMAB's. To validate the methodology, we compare exact GMAB prices to neural network estimates. For this reason, we consider Brownian financial and biometric risk factors and find a closed form expression for the GMAB. An alternative validation procedure would be to evaluate GMAB by Monte-Carlo simulations but this approach is more computing intensive and less accurate. This section introduces the dynamics of risk factors. Section 3 provides analytical expressions for bond prices and survival probabilities. The analytical value of the GMAB is finally obtained in Section 4.

The Guaranteed Minimum Accumulation Benefits (GMAB's) provide policyholders with a promise that the value of their investment, conditionally to their survival, will reach a minimum predetermined amount at expiry, regardless of how the underlying investments perform. The value of GMAB's depends both on biometrical and financial risk factors that we detail in this section. We consider an hybrid market in which we invest in cash, stocks, interest rate bonds and mortality bonds. The stock indice, the interest rate are respectively denoted by $(S_t)_{t\geq 0}$, $(r_t)_{t\geq 0}$. The insured is x_μ-year old and the reference force of mortality is denoted by $(\mu_{x+t})_{t\geq 0}$. The insurer

can also invest in mortality bonds written on a reference population of age x_λ and mortality rates, $(\lambda_{x_\lambda+t})_{t\geq 0}$.

The financial and biometric processes are defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ associated to four independant Brownian motions under $\mathbb{P}$, $\tilde{\boldsymbol{W}}_t = \left(\tilde{W}_t^{(1)}, \tilde{W}_t^{(2)}, \tilde{W}_t^{(3)}, \tilde{W}_t^{(4)}\right)_{t\geq 0}$. The state variables $\left(S_t, r_t, \mu_{x_\mu+t}, \lambda_{x_\lambda+t}\right)$ are driven by the following SDE's:

$$d\begin{pmatrix} S_t \\ r_t \\ \mu_{x_\mu+t} \\ \lambda_{x_\lambda+t} \end{pmatrix} = \begin{pmatrix} (r_t + \nu_S\sigma_S)\, S_t \\ \kappa_r\left(\gamma_r(t) - \nu_r\frac{\sigma_r}{\kappa_\kappa} - r_t\right) \\ \kappa_\mu\left(\gamma_\mu(t) - \nu_\mu\frac{\sigma_\mu(t)}{\kappa_\mu} - \mu_{x_\mu+t}\right) \\ \kappa_\lambda\left(\gamma_\lambda(t) - \nu_\lambda\frac{\sigma_\lambda(t)}{\kappa_\lambda} - \lambda_{x_\lambda+t}\right) \end{pmatrix} dt + \begin{pmatrix} S_t\sigma_S\Sigma_S^\top \\ \sigma_r\Sigma_r^\top \\ \sigma_\mu(t)\Sigma_\mu^\top \\ \sigma_\lambda(t)\Sigma_\lambda^\top \end{pmatrix} d\tilde{\boldsymbol{W}}_t\,, \qquad (1)$$

where κ_r, κ_μ, κ_λ, σ_S and σ_r belong to $\mathbb{R}^+$ whereas $\gamma_r(t)$, $\gamma_\mu(t)$, $\gamma_\lambda(t)$, $\sigma_\mu(t)$ and $\sigma_\lambda(t)$ are positive functions of time. $\gamma_r(t)$ $\gamma_\mu(t)$ and $\gamma_\lambda(t)$ are fitted to term structures of interest and mortality rates. Details about the estimation procedure are provided in Appendix. Notice that $\gamma_\mu(t)$, $\gamma_\lambda(t)$, $\sigma_\mu(t)$ and $\sigma_\lambda(t)$ depends on the initial age of the insured and of the hedging population; $x_\mu, x_\lambda \in [0, x_{max})$, $x_{max} > 0$. The parameters ν_S, ν_r, ν_μ and ν_λ tunes the risk premiums $\nu_S\sigma_S$, $-\nu_r\sigma_r$, $-\nu_\mu\sigma_\mu(t)$, $-\nu_\lambda\sigma_\lambda(t)$ of processes. Furthermore, we assume that the standard deviation of mortality rates are related to age through the relations $\sigma_\mu(t) = \alpha_\mu e^{\beta_\mu(x_\mu+t)}$ and $\sigma_\lambda(t) = \alpha_\lambda e^{\beta_\lambda(x_\lambda+t)}$. Σ_S, Σ_r, Σ_μ and Σ_λ are vectors such that $\Sigma = \left(\Sigma_S^\top, \Sigma_r^\top\Sigma_\mu^\top, \Sigma_\lambda^\top\right)$is the (upper) Choleski decomposition of the correlation matrix:

$$\Sigma = \begin{pmatrix} \Sigma_S^\top \\ \Sigma_r^\top \\ \Sigma_\mu^\top \\ \Sigma_\lambda^\top \end{pmatrix} = \begin{pmatrix} \epsilon_{SS} & \epsilon_{Sr} & \epsilon_{S\mu} & \epsilon_{S\lambda} \\ 0 & \epsilon_{rr} & \epsilon_{r\mu} & \epsilon_{r\lambda} \\ 0 & 0 & \epsilon_{\mu\mu} & \epsilon_{\mu\lambda} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad , \quad \begin{pmatrix} 1 & \rho_{Sr} & \rho_{S\mu} & \rho_{S\lambda} \\ \rho_{Sr} & 1 & \rho_{r\mu} & \rho_{r\lambda} \\ \rho_{S\mu} & \rho_{r\mu} & 1 & \rho_{\mu\lambda} \\ \rho_{S\lambda} & \rho_{r\lambda} & \rho_{\mu\lambda} & 1 \end{pmatrix} = \Sigma\Sigma^\top\,,$$

where $\rho_{Sr}, \rho_{S\mu}$, $\rho_{S\lambda}, \rho_{r\mu}, \rho_{r\lambda}$ and $\rho_{\mu\lambda} \in (-1, 1)$. This model incorporates the correlation between financial and mortality shocks, which can be relevant in the context of events such as a pandemic like Covid-19. We assume that costs of risk, noted $\boldsymbol{\theta} = (\theta_j)_{j=1,\ldots,4}^\top$, are constant, i.e. the Brownian motions with drift under $\mathbb{P}$,

$$dW_t^{(j)} = d\tilde{W}_t^{(j)} + \theta_j dt\,,$$

are Brownian motions under $\mathbb{Q}$. To keep identical dynamics under $\mathbb{P}$ and $\mathbb{Q}$, the risk premium parameters, stored in a vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)^\top$ are such that

$$\begin{aligned} \nu_S &= \epsilon_{SS}\theta_1 + \epsilon_{Sr}\theta_2 + \epsilon_{S\mu}\theta_3 + \epsilon_{S\lambda}\theta_4\,, \\ \nu_r &= -\left(\epsilon_{rr}\theta_2 + \epsilon_{r\mu}\theta_3 + \epsilon_{r\lambda}\theta_4\right)\,, \\ \nu_\mu &= -\left(\epsilon_{\mu\mu}\theta_3 + \epsilon_{\mu\lambda}\theta_4\right)\,, \\ \nu_\lambda &= -\theta_4\,. \end{aligned}$$

Under the risk neutral measure $\mathbb{Q}$, financial and biometric processes are ruled by the following SDE's:

$$d\begin{pmatrix} S_t \\ r_t \\ \mu_{x_\mu+t} \\ \lambda_{x_\lambda+t} \end{pmatrix} = \begin{pmatrix} r_t\, S_t \\ \kappa_r\left(\gamma_r(t) - r_t\right) \\ \kappa_\mu\left(\gamma_\mu(t) - \mu_{x_\mu+t}\right) \\ \kappa_\lambda\left(\gamma_\lambda(t) - \lambda_{x_\lambda+t}\right) \end{pmatrix} dt + \begin{pmatrix} S_t\sigma_S\Sigma_S^\top \\ \sigma_r\Sigma_r^\top \\ \sigma_\mu(t)\Sigma_\mu^\top \\ \sigma_\lambda(t)\Sigma_\lambda^\top \end{pmatrix} d\tilde{\boldsymbol{W}}_t\,. \qquad (2)$$

We will use later these equations to build the valuation equation fulfilled by the GMAB. As already mentioned, we consider a pure Brownian model because we want to benchmark exact GMAB prices to PINN predictions. The next two sections present useful intermediate analytical results, needed to infer a closed form expression for the GMAB.

3 Bonds and survival probabilities

We denote by $P(t,T)$ the value of a zero-coupon bond paying one monetary unit at expiry date, T. The survival probabilities of an $x_z + t$ year old individual up to time t are noted ${}_Tp^z_{x_z+t}$, for $z \in \{\mu, \lambda\}$. Pure endowments provide a lump sum payment at expiry in case of survival and nothing if the policyholder dies before the term. We denote them by ${}_TE^z_t$ for $z \in \{\mu, \lambda\}$. Zero-coupon bonds, survival probabilities, pure endowents are valued by the following expectations under the risk neutral measure:

$$\begin{cases} P(t,T) & = \mathbb{E}^{\mathbb{Q}}\left(e^{-\int_t^T r_s ds}|\mathcal{F}_t\right), \\ {}_Tp^z_{x_z+t} & = \mathbb{E}^{\mathbb{Q}}\left(e^{-\int_t^T z_{x_z+s} ds}\,|\,\mathcal{F}_t\right) \; z \in \{\mu,\lambda\}, \\ {}_TE^z_t & = N^\mu_t \mathbb{E}^{\mathbb{Q}}\left(e^{-\int_t^T (r_s + z_{x_z+s}) ds}\,|\,\mathcal{F}_t\right) \; z \in \{\mu,\lambda\}. \end{cases}$$

The model being affine, we derive the closed-form expressions of these products. In the remainder of this article, we adopt the following notation

$$B_y(t,T) = \frac{1 - e^{-y(T-t)}}{y},$$

where $y \in \mathbb{R}^+$ is a positive parameter. Various combined integrals of $B_y(t,T)$, $\sigma_\mu(t)$ and $\sigma_\lambda(t)$ are provided in Appendix B and serve to analytically evaluate the GMAB. From Proposition 4 in Appendix A, $\gamma_r(T)$ matches the initial yield curve of interest rate if

$$\gamma_r(T) \;=\; -\frac{1}{\kappa_r}\partial^2_T \ln P(0,T) - \partial_T \ln P(0,T) + \frac{\sigma_r^2}{2\kappa_r^2}\left(1 - e^{-2\kappa_r T}\right),$$

where $-\partial_T \ln P(0,T)$ is the instantaneous forward rate. In a similar manner, for a given initial mortality curve ${}_Tp^z_{x_z}$, the function $\gamma_z(u)$ is such that

$$\gamma_z(T) \;=\; -\frac{1}{\kappa_z}\partial^2_T \ln {}_Tp^z_{x_z} - \partial_T \ln {}_Tp^z_{x_z} + \frac{\alpha_z^2 e^{2\beta_z x_z}}{2\kappa_z(\kappa_z+\beta_z)}\left(e^{2\beta_z T} - e^{-2\kappa_z T}\right).$$

Bond prices, survival probabilities and endowments admit closed-form expressions presented in the next proposition.

Proposition 1. *The price at time $t \le T$ of a discount bond of maturity T is linked to the initial interest rate curve at time $t = 0$ by the relation*

$$\begin{aligned} P(t,T) \;=\; & \exp\left(-r_t B_{\kappa_r}(t,T) - \left(\partial_t \ln P(0,t)\right) B_{\kappa_r}(t,T) + \ln\frac{P(0,T)}{P(0,t)}\right) \\ & \times \exp\left(-\frac{\sigma_r^2}{4\kappa_r}\left(\left(1 - e^{-2\kappa_r t}\right) B_{\kappa_r}(t,T)^2\right)\right). \end{aligned} \tag{3}$$

In a similar manner, we can show that, when alive at age $x_z + t$, the survival probability up to time T depends on the initial survival term structure as follows for $z \in \{\mu, \lambda\}$

$$\begin{aligned} {}_Tp^z_{x+t} = & \exp\left(-z_{x_z+t} B_{\kappa_z}(t,T) - \left(\partial_t \ln {}_tp^z_{x_z}\right) B_{\kappa_z}(t,T) + \ln\frac{{}_Tp^z_{x_z}}{{}_tp^z_{x_z}}\right) \times \\ & \exp\left(\frac{\alpha_z^2 e^{2\beta_z(x_z+T)}}{2\kappa_z^2}\left(B_{2\beta_z}(t,T) - 2B_{2\beta_z+\kappa_z}(t,T) + B_{2\beta_z+2\kappa_z}(t,T)\right)\right) \times \\ & \exp\left(\frac{\alpha_z^2 e^{2\beta_z x_z}}{2\kappa_z(\kappa_z+\beta_z)}\left(e^{2\beta_z T} B_{2\beta_z+\kappa_z}(t,T) - e^{2\beta_z T} B_{2\beta_z}(t,T)\right)\right) \times \\ & \exp\left(\frac{\alpha_z^2 e^{2\beta_z x_z}}{2\kappa_z(\kappa_z+\beta_z)}\left(e^{-2\kappa_z t} B_{2\kappa_z}(t,T) - e^{-\kappa_z(T+t)} B_{\kappa_z}(t,T)\right)\right), \end{aligned} \tag{4}$$

whereas the pure endowment contracts are valued by:

$$
\begin{aligned}
{}_TE_t^{\mu} &= N_t^{\mu}\, {}_Tp_{x_\mu+t}^{\mu}\, P(t,T)\, \exp\left(\frac{\sigma_r \Sigma_r^\top \Sigma_\mu \alpha_\mu e^{\beta_\mu (x_\mu+T)}}{\kappa_\mu \kappa_r}\right) \\
&\quad \times \exp\left(B_{\beta_\mu}(t,T) - B_{\kappa_\mu+\beta_\mu}(t,T) - B_{\kappa_r+\beta_\mu}(t,T) + B_{\kappa_\mu+\kappa_r+\beta_\mu}(t,T)\right) .
\end{aligned} \tag{5}
$$

and

$$
\begin{aligned}
{}_TE_t^{\lambda} &= N_t^{\lambda}\, {}_Tp_{x_\lambda+t}^{\lambda}\, P(t,T)\, \exp\left(\frac{\sigma_r \Sigma_r^\top \Sigma_\lambda \alpha_\lambda e^{\beta_\lambda (x_\lambda+T)}}{\kappa_\lambda \kappa_r}\right) \\
&\quad \times \exp\left(B_{\beta_\lambda}(t,T) - B_{\kappa_\lambda+\beta_\lambda}(t,T) - B_{\kappa_r+\beta_\lambda}(t,T) + B_{\kappa_\lambda+\kappa_r+\beta_\lambda}(t,T)\right) .
\end{aligned} \tag{6}
$$

The sketch of the proof is provided in Appendix A. Using standard developments, the bond price dynamic is equal to

$$
\frac{dP(t,T)}{P(t,T)} = \left(r_t + B_{\kappa_r}(t,T)\sigma_r\nu_r\right) dt - B_{\kappa_r}(t,T)\sigma_r\Sigma_r^\top d\tilde{\boldsymbol{W}}_t\,. \tag{7}
$$

We assume that policyholder's savings is invested in cash, stocks, zero-coupon bonds, and in mortality bonds written on a reference population of age x_λ at time $t = 0$. We denote by $D(t,T)$, the value of this mortality linked bond expiring at time T. Its return depends on the benchmark mortality rate, $\lambda_{x_\lambda+t}$ and is valued at by the following expectation:

$$
\begin{aligned}
D(t,T) &= \mathbb{E}^{\mathbb{Q}}\left(e^{-\int_t^T r_s ds - \int_0^T \lambda_{x_\lambda+s} ds} | \mathcal{F}_t\right) \\
&= e^{-\int_0^t \lambda_{x_\lambda+s} ds}\, \mathbb{E}^{\mathbb{Q}}\left(e^{-\int_t^T (r_s+\lambda_{x_\lambda+s}) ds} | \mathcal{F}_t\right) .
\end{aligned}
$$

This product is design in such a manner that in case of over/under-mortality, the holder of this bond realizes a capital loss/gain at expiry. The dynamic of such a bond is described in the next proposition, proved in Appendix A.

Proposition 2. *Under the risk neutral measure $\mathbb{Q}$, the mortality bond earns on average the risk free rate and its price obeys to the SDE:*

$$
\frac{dD(t,T)}{D(t,T)} = r_t dt - \left(B_{\kappa_r}(t,T)\sigma_r\Sigma_r^\top + B_{\kappa_\lambda}(t,T)\sigma_\lambda(t)\Sigma_\lambda^\top\right) d\boldsymbol{W}_t \tag{8}
$$

Under $\mathbb{P}$, the risk premium of the mortality bond is proportional to interest and mortality volatilities:

$$
\begin{aligned}
\frac{dD(t,T)}{D(t,T)} &= \left(r_t + B_{\kappa_r}(t,T)\sigma_r\nu_r + B_{\kappa_\lambda}(t,T)\sigma_\lambda(t)\nu_\lambda\right) dt \\
&\quad - \left(B_{\kappa_r}(t,T)\sigma_r\Sigma_r^\top + B_{\kappa_\lambda}(t,T)\sigma_\lambda(t)\Sigma_\lambda^\top\right) d\tilde{\boldsymbol{W}}_t
\end{aligned} \tag{9}
$$

4 Contract and analytical valuation

Guaranteed Minimum Accumulation Benefits (GMAB) are retirement savings products which promises in case of survival, the maximum between investments and a guaranteed capital. The policyholder savings are invested in cash, stocks, risk-free and mortality bonds of maturity T. We denote by $(A_t)_{t\geq 0}$ the total value of these investments. The percentages of stocks, risk-free and mortality bonds are assumed constant and denoted by $\boldsymbol{\pi} = (\pi_S, \pi_P, \pi_D)_{t\geq 0}$. We assume that the investment policy is self-financed, the dynamic of assets under management is equal to

$$
\frac{dA_t}{A_t} = \pi_S \frac{dS_t}{S_t} + \pi_P \frac{dP_t}{P_t} + \pi_D \frac{dD_t}{D_t} + (1 - \pi_S - \pi_P - \pi_D)\, r_t\, dt\,,
$$

where dS_t, $dP(t,T)$ and $dD(t,T)$ are respectively provided in Equations (1), (7) and (9). We can therefore reformulate the differential of A_t under $\mathbb{P}$ as follows

$$\frac{dA_t}{A_t} = \left(r_t + \boldsymbol{\pi}^\top \boldsymbol{\nu}_A(t)\right) dt + \boldsymbol{\pi}^\top \Sigma_A(t) d\tilde{\boldsymbol{W}}_t$$

where $\boldsymbol{\nu}_A(t)$ is a time varying vector of risk premiums,

$$\boldsymbol{\nu}_A(t) = \begin{pmatrix} \sigma_S \nu_S \\ B_{\kappa_r}(t,T)\sigma_r \nu_r \\ B_{\kappa_r}(t,T)\sigma_r \nu_r + B_{\kappa_\lambda}(t,T)\sigma_\lambda(t)\nu_\lambda \end{pmatrix},$$

and $\Sigma_A(t)$ is a volatility 3×4-matrix :

$$\Sigma_A(t) = \begin{pmatrix} \sigma_S \Sigma_S^\top \\ -B_{\kappa_r}(t,T)\sigma_r \Sigma_r^\top \\ -B_{\kappa_r}(t,T)\sigma_r \Sigma_r^\top - B_{\kappa_\lambda}(t,T)\sigma_\lambda(t)\Sigma_\lambda^\top \end{pmatrix}.$$

Under the risk neutral measure, investments earn on average the risk-free rate and the differential of A_t is ruled by

$$\frac{dA_t}{A_t} = r_t dt + \boldsymbol{\pi}^\top \Sigma_A(t) d\boldsymbol{W}_t.$$

The contract is subscribed by an individual of age x_μ and guarantees at the expiry date, denoted by T^*, a payout equal to the maximum between a guaranteed capital G_m and the portfolio A_{T^*}, in the event of survival. However, the benefit is eventually bounded by G_M. In practice, G_m is often set to the initial investment, A_0, or to a percentage of A_0. Whereas, the participation is most of the time unbounded ($G_M = +\infty$). We denote by $\tau_\mu \in \mathbb{R}^+$, the random time of insured's death and $N_t^\mu = \mathbf{1}_{\{\tau_\mu \geq t\}}$. The payoff in case of survival is

$$H(A_{T^*}, G_m) = \left(G_{T^*} + (A_{T^*} - G_m)_+ - (A_{T^*} - G_M)_+\right) \tag{10}$$

The fair value of such a policy, denoted by L_t is equal to the expected discounted cash-flows under the chosen risk neutral measure, noted $\mathbb{Q}$. This is a function of all state variables:

$$\begin{aligned} L_t = {} & N_t^\mu \mathbb{E}^{\mathbb{Q}}\left(e^{-\int_t^{T^*}(r_s+\mu_{x_\mu+s})ds} G_m \,|\, \mathcal{F}_t\right) \\ & + N_t^\mu \mathbb{E}^{\mathbb{Q}}\left(e^{-\int_t^{T^*}(r_s+\mu_{x_\mu+s})ds} \left((A_{T^*} - G_m)_+\right) \,|\, \mathcal{F}_t\right) \\ & - N_t^\mu \mathbb{E}^{\mathbb{Q}}\left(e^{-\int_t^{T^*}(r_s+\mu_{x_\mu+s})ds} \left((A_{T^*} - G_M)_+\right) \,|\, \mathcal{F}_t\right). \end{aligned} \tag{11}$$

In order to obtain a closed form expression of the saving contract (10), we perform a change of measure using as Radon-Nykodym derivative:

$$\left.\frac{d\mathbb{F}}{d\mathbb{Q}}\right|_{T^*} = \mathbb{E}^{\mathbb{Q}}\left(\frac{d\mathbb{F}}{d\mathbb{Q}} | \mathcal{F}_{T^*}\right) = \frac{e^{-\int_0^{T^*}(r_s+\mu_{x_\mu+s})ds}}{\mathbb{E}^{\mathbb{Q}}\left(e^{-\int_0^{T^*}(r_s+\mu_{x_\mu+s})ds} | \mathcal{F}_0\right)}. \tag{12}$$

From Equations (37), this change of measure is rewritten as follows:

$$\begin{aligned}
\left.\frac{d\mathbb{F}}{d\mathbb{Q}}\right|_{T^*} &= \exp\left(-\frac{\sigma_r^2\epsilon_{rr}^2}{2}\int_0^{T^*} B_{\kappa_r}(u,T^*)^2 du - \sigma_r\epsilon_{rr}\int_0^{T^*} B_{\kappa_r}(u,T^*)\, dW_u^{(2)}\right)\\
&\times\exp\left(-\int_0^{T^*}\left(\sigma_r\epsilon_{r\mu}B_{\kappa_r}(u,T^*)+\sigma_\mu(u)\epsilon_{\mu\mu}B_{\kappa_\mu}(u,T^*)\right)dW_u^{(3)}\right)\\
&\times\exp\left(-\frac{1}{2}\int_0^{T^*}\left(\sigma_r\epsilon_{r\mu}B_{\kappa_r}(u,T^*)+\sigma_\mu(u)\epsilon_{\mu\mu}B_{\kappa_\mu}(u,T^*)\right)^2 du\right)\\
&\times\exp\left(-\int_0^{T^*}\left(\sigma_r\epsilon_{r\lambda}B_{\kappa_r}(u,T^*)+\sigma_\mu(u)\epsilon_{\mu\lambda}B_{\kappa_\mu}(u,T^*)+\sigma_\lambda(u)B_{\kappa_\lambda}(u,T^*)\right)dW_u^{(4)}\right)\\
&\times\exp\left(-\frac{1}{2}\int_0^{T^*}\left(\sigma_r\epsilon_{r\lambda}B_{\kappa_r}(u,T^*)+\sigma_\mu(u)\epsilon_{\mu\lambda}B_{\kappa_\mu}(u,T^*)+\sigma_\lambda(u)B_{\kappa_\lambda}(u,T^*)\right)^2 du\right)
\end{aligned}$$

We recognize a Doleans-Dade exponential and then under the measure $\mathbb{F}$. We denote by $\boldsymbol{W}_t^{\mathbb{F}}$ the vector of $\mathbb{F}$-Brownian motions $\left(W_t^{(1)\mathbb{F}}, W_t^{(2)\mathbb{F}}, W_t^{(3)\mathbb{F}}, W_t^{(4)\mathbb{F}}\right)$, defined by

$$d\boldsymbol{W}_t^{\mathbb{F}} \;=\; d\boldsymbol{W}_t + \left(\sigma_r B_{\kappa_r}(t,T^*)\Sigma_r + \sigma_\mu(t)B_{\kappa_\mu}(t,T^*)\Sigma_\mu + \sigma_\lambda(t)B_{\kappa_\lambda}(t,T^*)\Sigma_\lambda\right)dt\,. \tag{13}$$

The dynamic of the total asset under the $\mathbb{F}$- measure is then equal to

$$\begin{aligned}
\frac{dA_t}{A_t} &= r_t dt + \boldsymbol{\pi}^\top\Sigma_A(t)d\boldsymbol{W}_t^{\mathbb{F}} - \boldsymbol{\pi}^\top\Sigma_A(t)\times \\
&\left(\sigma_r B_{\kappa_r}(t,T^*)\Sigma_r + \sigma_\mu(t)B_{\kappa_\mu}(t,T^*)\Sigma_\mu + \sigma_\lambda(t)B_{\kappa_\lambda}(t,T^*)\Sigma_\lambda\right)dt
\end{aligned} \tag{14}$$

As $A_{T^*} = \frac{A_{T^*}}{P(T^*,T^*)}$, we focus on on the dynamics of $d\frac{A_t}{P(t,T^*)}$ in order to prove that this ratio is a log-normal process.

Proposition 3. *The quantity $\frac{A_t}{P(t,T^*)}$ has a log-normal distribution, $\ln\frac{A_{T^*}/P(T^*,T^*)}{A_t/P(t,T^*)} \sim N\left(\mu_{\mathbb{F}}(t), v_{\mathbb{F}}(t)\right)$, with a time dependent variance equal to*

$$v_{\mathbb{F}}(t) = \int_t^{T^*}\left(B_{\kappa_r}(u,T^*)\sigma_r\Sigma_r^\top + \boldsymbol{\pi}^\top\Sigma_A(u)\right)\left(B_{\kappa_r}(u,T^*)\sigma_r\Sigma_r + \Sigma_A(u)^\top\boldsymbol{\pi}\right)du\,, \tag{15}$$

and a time dependent mean given by:

$$\begin{aligned}
\mu_{\mathbb{F}}(t) = &-\frac{1}{2}v_{\mathbb{F}}(t) - \sigma_r\Sigma_r^\top\Sigma_\mu\int_t^{T^*}\sigma_\mu(u)B_{\kappa_r}(u,T^*)B_{\kappa_\mu}(u,T^*)\,du\\
&-\sigma_r\Sigma_r^\top\Sigma_\lambda\int_t^{T^*}\sigma_\lambda(u)B_{\kappa_r}(u,T^*)B_{\kappa_\lambda}(u,T^*)du\\
&-\boldsymbol{\pi}^\top\int_t^{T^*}\Sigma_A(u)\sigma_\mu(u)B_{\kappa_\mu}(u,T^*)du\,\Sigma_\mu\\
&-\boldsymbol{\pi}^\top\int_t^{T^*}\Sigma_A(u)\sigma_\lambda(u)B_{\kappa_\lambda}(u,T^*)du\,\Sigma_\lambda\,.
\end{aligned} \tag{16}$$

The proof is provided in Appendix A whereas the the full analytical expressions of $v_{\mathbb{F}}(t)$ and $\mu_{\mathbb{F}}(t)$ are detailed in Appendix C. As $\frac{A_t}{P(t,T^*)}$ is log-normal under the forward measure, we can evaluatecall options embedded in the GMAB.

Corollary 1. *Let us denote by $\Phi(.)$ is the cumulative distribution function (cdf) of a $N(0,1)$. If we define*

$$\begin{cases} d_2(t) = \frac{\ln\left(\frac{G_{T^*}}{A_t/P(t,T^*)}\right) - \mu_{\mathbb{F}}(t)}{\sqrt{v_{\mathbb{F}}(t)}} & , d_1(t) = d_2(t) - \sqrt{v_{\mathbb{F}}(t)} \\ d_2^M(t) = \frac{\ln\left(\frac{G_M}{A_t/P(t,T^*)}\right) - \mu_{\mathbb{F}}(t)}{\sqrt{v_{\mathbb{F}}(t)}} & , d_1^M(t) = d_2^M(t) - \sqrt{v_{\mathbb{F}}(t)} \end{cases} \tag{17}$$

then expected positive differences between A_{T^*}, G_{T^*} and G_M under the forward measure are given by

$$\mathbb{E}^{\mathbb{F}}\left((A_{T^*} - G_m)_+ \,|\mathcal{F}_t\right) = \frac{A_t}{P(t,T^*)} e^{\mu_{\mathbb{F}}(t) + \frac{v_{\mathbb{F}}(t)}{2}} \Phi\left(-d_1(t)\right) - G_m \Phi\left(-d_2(t)\right) ,$$
$$\mathbb{E}^{\mathbb{F}}\left((A_{T^*} - G_M)_+ \,|\mathcal{F}_t\right) = \frac{A_t}{P(t,T^*)} e^{\mu_{\mathbb{F}}(t) + \frac{v_{\mathbb{F}}(t)}{2}} \Phi\left(-d_1^M(t)\right) - G_M \Phi\left(-d_2^M(t)\right) .$$

This corollary is proven using standard Black & Scholes developments. This last result allows us to infer that the market value of the GMAB (11) is given by

$$\begin{aligned} L_t = {}& {}_{T^*}E_t^{\mu}\, G_m \\ & + {}_{T^*}E_t^{\mu}\left[\frac{A_t e^{\mu_{\mathbb{F}}(t) + \frac{v_{\mathbb{F}}(t)}{2}}}{P(t,T^*)} \Phi\left(-d_1(t)\right) - G_m\, \Phi\left(-d_2(t)\right)\right] \\ & - {}_{T^*}E_t^{\mu}\left[\frac{A_t e^{\mu_{\mathbb{F}}(t) + \frac{v_{\mathbb{F}}(t)}{2}}}{P(t,T^*)} \Phi\left(-d_1^M(t)\right) - G_M \Phi\left(-d_2^M(t)\right)\right] , \end{aligned} \tag{18}$$

where ${}_{T^*}E_t^{\mu}$ is given by Equation (5). This closed-form is used to measure the accuracy of the PINN for predicting GMAB prices.

5 Scaled Feynman-Kac equation

A physics-inspired neural network (PINN) integrates principles from physics, such as a PDE ruling the behaviour of state variables. In a financial context, we use as principle the Feynman-Kac (FK) equation. This is a PDE fulfilled by all assets traded in an arbitrage-free market. In this section, we build this equation for the model considered in this article. In the next section, we solve it with a neural network for various investment strategies, contract and bond maturities. To lighten future developments, we denote the vector of state variables, augmented with the total asset value, by

$$\boldsymbol{y}_t = \left(S_t, r_t, \mu_{x_\mu + t}, \lambda_{x_\lambda + t}, A_t\right)^\top .$$

Under the risk neutral measure $\mathbb{Q}$, this multivariate process is ruled by the following SDE:

$$\mathrm{d}\boldsymbol{y}_t \;\; = \;\; \boldsymbol{\mu_y}(t, \boldsymbol{y}_t)dt + \Sigma_{\boldsymbol{y}}(t, \boldsymbol{y}_t)d\boldsymbol{W}_t \,,$$

where $\boldsymbol{\mu_y}(.)$ is a vector of dimension 5 and $\Sigma_{\boldsymbol{y}}(.)$ is a 5×4 matrix:

$$\boldsymbol{\mu_y}(t, \boldsymbol{y}_t) = \begin{pmatrix} r_t\, S_t \\ \kappa_r\left(\gamma_r(t) - r_t\right) \\ \kappa_\mu\left(\gamma_\mu(t) - \mu_{x_\mu + t}\right) \\ \kappa_\lambda\left(\gamma_\lambda(t) - \lambda_{x_\lambda + t}\right) \\ r_t A_t \end{pmatrix} , \; \Sigma_{\boldsymbol{y}}(t, \boldsymbol{y}_t) = \begin{pmatrix} S_t \sigma_S \Sigma_S^\top \\ \sigma_r \Sigma_r^\top \\ \sigma_\mu(t) \Sigma_\mu^\top \\ \sigma_\lambda(t) \Sigma_\lambda^\top \\ A_t \boldsymbol{\pi}^\top \Sigma_A(t) \end{pmatrix} .$$

The GMAB is a function of time and state variables parameterized by the investment policy $\boldsymbol{\pi}$, the duration of the contract T^* and the maturity of bonds, T. As we aim to understand the relation between assets and liabilities, we emphasize the dependence to investment parameters by denoting the contract price as follows:

$$L_t = V\left(t, \boldsymbol{y}_t, N_t^{\mu} \,|\, \boldsymbol{\pi}, T^*, T\right) .$$

Under the assumption of arbitrage-free market, all traded securities, including the insurance contract, earn on average the risk free rate, i.e. :

$$\mathbb{E}\left(dL_{t+} \,|\, \mathcal{F}_t\right) = L_t \, r_t \, dt \,. \tag{19}$$

Let us respectively denote the gradient and the Hessian of $V(.)$ with respect to $\boldsymbol{y}$ by $\nabla_{\boldsymbol{y}} V$ and $\mathcal{H}_{\boldsymbol{y}}(V)$:

$$\nabla_{\boldsymbol{y}} V = \begin{pmatrix} \partial_S V \\ \partial_r V \\ \partial_\mu V \\ \partial_\lambda V \\ \partial_A V \end{pmatrix} \quad , \quad \mathcal{H}_{\boldsymbol{y}}(V) = \begin{pmatrix} \partial_{SS} V & \partial_{Sr} V & \partial_{S\mu} V & \partial_{S\lambda} V & \partial_{SA} V \\ \partial_{Sr} V & \partial_{rr} V & \partial_{r\mu} V & \partial_{r\lambda} V & \partial_{rA} V \\ \partial_{S\mu} V & \partial_{r\mu} V & \partial_{\mu\mu} V & \partial_{\mu\lambda} V & \partial_{\mu A} V \\ \partial_{S\lambda} V & \partial_{r\lambda} V & \partial_{\mu\lambda} V & \partial_{\lambda\lambda} V & \partial_{\lambda A} V \\ \partial_{SA} V & \partial_{rA} V & \partial_{\mu A} V & \partial_{\lambda A} V & \partial_{AA} V \end{pmatrix} .$$

Applying the Itô's lemma to $V\,(.)$ allows us to rewrite (19) as a partial differential valuation equation:

$$\begin{aligned} 0 =& \partial_t V - (r_t + \mu_{x+t}) \; V + \boldsymbol{\mu}_{\boldsymbol{y}}(t, \boldsymbol{y}_t)^\top \nabla_{\boldsymbol{y}} V \\ &+ \frac{1}{2} \mathrm{tr}\left(\Sigma_{\boldsymbol{y}}(t, \boldsymbol{y}_t) \Sigma_{\boldsymbol{y}}(t, \boldsymbol{y}_t)^\top \mathcal{H}_{\boldsymbol{y}}(V) \right) . \end{aligned} \tag{20}$$

This last expression is called the "Feynman-Kac (FK) equation". The boundary constraints on $V(.)$ are

$$\begin{cases} V(T^*, \boldsymbol{y}_{T^*}, 1 \,|\, \boldsymbol{\pi}, T^*, T) &= H\left(A_{T^*}, G_m, G_M\right) \,, \\ V(t, \boldsymbol{y}_t, 0 \,|\, \boldsymbol{\pi}, T^*, T) &= 0 \,. \end{cases} \tag{21}$$

Notice that we have 5 state variables and 4 Brownian motions. The model is over-specified and we can remove one state variable from $\boldsymbol{y}$, e.g. the stock price which is redundant with respect to the total asset value, A_t. Nevertheless, we have kept all state variables in the numerical illustration to see how the PINN manages a naïve over-specification.

Due to the number of risk factors, solving numerically the PDE (20) is numerically challenging. Implementing an explicit or implicit finite difference method requires to build a 6-dimensional grid in the cross space of time and state variables, with relatively small meshes to ensure convergence. In practice, this is intractable. We instead approximate the solution of the FK equation (20) by a neural network taking as input the time, the state variables, the asset mix and durations of GMAB and bonds. The mathematical definition of a neural network is recalled in the next section but we can already mention that a neural network uses bounded activation functions (e.g. sigmoid or hyperbolic tangents). To avoid convergence issues related to what is called the vanishing gradient effect, we need to center and scale the inputs of the network. This pretreatment slightly modifies the FK equation. Let us define the vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^5$ as follows

$$\begin{aligned} \boldsymbol{a} &= (a_S, a_r, a_\mu, a_\lambda, a_A)^\top \,, \\ \boldsymbol{b} &= (b_S, b_r, b_\mu, b_\lambda, b_A)^\top \,. \end{aligned} \tag{22}$$

$\boldsymbol{a}$ and $\boldsymbol{b}$ are chosen to normalize a random sample of state variables. This point is discussed in the next section. The vector of centered and scaled state variables is denoted by

$$\tilde{\boldsymbol{y}}_t = \boldsymbol{a} + \boldsymbol{b} \odot \boldsymbol{y}_t \,, \tag{23}$$

where $\odot$ is the elementwise product. In a similar manner, we center and scale the time with coefficients a_h and b_h that will depends on the horizon of valuation. The centered scaled time and durations are defined as follows

$$\begin{cases} \tilde{t} &= a_h + b_h t \,, \\ \tilde{T} &= a_h + b_h T \,, \\ \tilde{T}^* &= a_h + b_h T^* . \end{cases} \tag{24}$$

We do not need to scale the vector $\boldsymbol{\pi} \in [0,1]^3$. The normalized state variables are ruled by the SDE:

$$d\tilde{\boldsymbol{y}}_t \quad = \quad \boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}(t,\tilde{\boldsymbol{y}}_t)dt + \Sigma_{\tilde{\boldsymbol{y}}}(t,\tilde{\boldsymbol{y}}_t)d\boldsymbol{W}_t\,,$$

where $\mu_{\tilde{y}}(.)$ is a 5×vector and $\Sigma_{\tilde{\boldsymbol{y}}}(.)$ is a 5×4 matrix:

$$\boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}(t,\tilde{\boldsymbol{y}}_t) = \begin{pmatrix} \left(\tilde{S}_t - a_S\right)\frac{\tilde{r}_t - a_r}{b_r} \\ \kappa_r\left(b_r\gamma_r(t) + a_r - \tilde{r}_t\right) \\ \kappa_\mu\left(b_\mu\gamma_\mu(t) + a_\mu - \tilde{\mu}_{x_\mu+t}\right) \\ \kappa_\lambda\left(b_\lambda\gamma_\lambda(t) + a_\lambda - \tilde{\lambda}_{x_\lambda+t}\right) \\ \left(\tilde{A}_t - a_A\right)\frac{\tilde{r}_t - a_r}{b_r} \end{pmatrix}, \; \Sigma_{\tilde{\boldsymbol{y}}}(t,\tilde{\boldsymbol{y}}_t) = \begin{pmatrix} \left(\tilde{S}_t - a_S\right)\sigma_S\Sigma_S^\top \\ b_r\sigma_r\Sigma_r^\top \\ b_\mu\sigma_\mu(t)\Sigma_\mu^\top \\ b_\lambda\sigma_\lambda(t)\Sigma_\lambda^\top \\ \left(\tilde{A}_t - a_A\right)\boldsymbol{\pi}_t^\top\Sigma_A(t) \end{pmatrix}.$$

The value of the contract may be seen as a function of time and rescaled state variables, $L_t = V(\tilde{t},\tilde{\boldsymbol{y}}_t,N_t^\mu,|\,\boldsymbol{\pi},\tilde{T}^*,\tilde{T})$. The valuation equation (20) is then rewritten in rescaled form:

$$\begin{aligned} 0 =& b_h\,\partial_{\tilde{t}}V - \left(\frac{\tilde{r}_t - a_r}{b_r} + \frac{\tilde{\mu}_{x+t} - a_\mu}{b_\mu}\right)V + \boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}(t,\tilde{\boldsymbol{y}}_t)^\top\nabla_{\tilde{\boldsymbol{y}}}V \\ &+ \frac{1}{2}\mathrm{tr}\left(\Sigma_{\tilde{\boldsymbol{y}}}(t,\tilde{\boldsymbol{y}}_t)\Sigma_{\tilde{\boldsymbol{y}}}(t,\tilde{\boldsymbol{y}}_t)^\top\mathcal{H}_{\tilde{\boldsymbol{y}}}(V)\right), \end{aligned} \tag{25}$$

where $\nabla_{\tilde{\boldsymbol{y}}}V$ and $\mathcal{H}_{\tilde{\boldsymbol{y}}}(V)$ are respectively the gradient and the Hessian of V with respect to standardized state variables, $\tilde{\boldsymbol{y}}$. The rescaled version of boundary conditions (21) is:

$$\begin{cases} V\left(\tilde{T}^*,\tilde{\boldsymbol{y}}_{T^*},1\,|\,\boldsymbol{\pi},\tilde{T}^*,\tilde{T}\right) & = H\left(\left(\tilde{A}_{T^*} - a_A\right)/b_A, G_{T^*}\right), \\ V\left(\tilde{t},\tilde{\boldsymbol{y}}_t,0\,|\,\boldsymbol{\pi},\tilde{T}^*,\tilde{T}\right) & = 0\,. \end{cases} \tag{26}$$

The next section explains how to solve Eq. (25) with a neural network.

6 Neural networks

We approximate the value function solving Equation (25) by a particular type of neural network. This network takes as input a vector of dimension 11, denoted by $\boldsymbol{z}$, containing the scaled time and state variables, the asset mix vector and scaled contract and bond maturities:

$$\boldsymbol{z} \quad := \quad \left(\tilde{t},\tilde{\boldsymbol{y}}_t,\boldsymbol{\pi},\tilde{T}^*,\tilde{T}\right)^\top. \tag{27}$$

The neural network, denoted by $F(.)$, approximates the price of the GMAB, i.e. $F(\boldsymbol{z}) \approx V(t,\tilde{\boldsymbol{y}}_t,1|\boldsymbol{\pi},\tilde{T}^*,\tilde{T})$. There does not exist any systematic procedure for determining an optimal structure for the neural network. From Hornik (1991), we just know that single layer neural networks approximate regular function arbitrarily well but the number of neurons may be high to achieve a reasonable accuracy. An alternative is offered by feed-forward networks in which the flow of information passes forward through several neural layers. Nevertheless, this category of networks struggles to replicate non-linear function with a limited number of layers. For this reason, Sirignano and Spiliopoulos (2018) proposes a variant of LSTM, called the Deep Galerkin Network. This structure replicates prices but the calibration is time consuming due to the complexity of neural cells. For this reason, we consider a simpler architecture in which intermediate layers are fed with the output of the previous layer and the initial input vector. Figure 1 presents the structure of such networks with what is called "skip connections". We can classify this model in the category of residual neural networks which have been successfully applied in many deep learning applications.

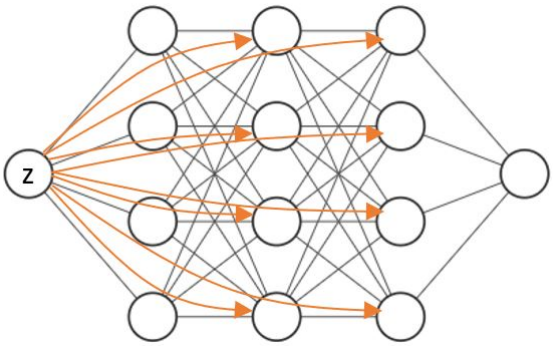

Figure 1: Feed forward network with skip connections toward intermediate layers.

Definition Let l, n_0, n_1, ..., $n_l \in \mathbb{N}$ be respectively the number of layers and neurons in each layer. n_0 is also the size of input vector. The activation function of layer $k = 1, 2, ..., l$ is noted $\phi_k(.) : \mathbb{R} \to \mathbb{R}$. Let $C_1 \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_0}$,$\boldsymbol{c}_1 \in \mathbb{R}^{n_1}$, $C_k \in \mathbb{R}^{n_k} \times \mathbb{R}^{n_0+n_{k-1}}$, $\boldsymbol{c}_k \in \mathbb{R}^{n_k}$ for $k = 2, ..., l-1$, $C_l \in \mathbb{R}^{n_l} \times \mathbb{R}^{n_{l-1}}$,$\boldsymbol{c}_l \in \mathbb{R}^{n_l}$be neural weights defining the input, intermediate and output layers. We define the following functions

$$\begin{cases} \psi_k(\boldsymbol{x}) = \phi_k\left(C_k \boldsymbol{x} + \boldsymbol{c}_k\right) , & k = 1, l \\ \psi_k(\boldsymbol{x}, \boldsymbol{z}) = \phi_k\left(C_k \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{pmatrix} + \boldsymbol{c}_k\right) , & k = 2, ..., l-1 \end{cases}$$

where activation functions $\phi_k(.)$ are applied componentwise. The residual neural network is is a function $F : \mathbb{R}^{n_0} \to \mathbb{R}^{n_l}$ defined by

$$F(\boldsymbol{z}) \quad = \quad \psi_{l-1} \circ \psi_{l-1}\left(., \boldsymbol{z}\right) ... \circ \psi_2\left(., \boldsymbol{z}\right) \circ \psi_1(\boldsymbol{z}) .$$

After having chosen a network achitecture, the model is trained by minimizing a loss function that is proportional to errors of approximation. This error is measured by replacing $V(.)$ with $F(.)$ in the scaled FK equation (25), at random points in the domain.

At time $t \in [0, T^*]$, the domain of the state vector, $\boldsymbol{y}_t = \left(S_t, r_t, \mu_{x_\mu + t}, \lambda_{x_\mu + t}, A_t\right)^\top$is $\mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}^{+3}$. We approximate this domain by a closed convex subspace:

$$\mathcal{D}^{\boldsymbol{y}} : [S_l, S_u] \times [r_l, r_u] \times [\mu_l, \mu_u] \times [\lambda_l, \lambda_u] \times [A_l, A_u] .$$

In order to fit the neural network, we draw a sample of $n_{\mathcal{D}}$ realizations of $\boldsymbol{y}_t$ in $\mathcal{D}^{\boldsymbol{y}}$, at random times. We also sample parameters $\left(\boldsymbol{\pi}_j, T_j^*, T_j\right)_{j=1,...,n_{\mathcal{D}}}$ under the constraints:

$$\begin{cases} \boldsymbol{\pi}_j \in [0,1] \times [0,1] \times [0,1] , \\ T_j^* \in [0, T_{max}^*] , \, T_j \in [0, T_{max}] \\ t_j \leq T_j^* \leq T_j . \end{cases}$$

The set of sampled state variables and parameters is noted $\mathcal{S}_{\mathcal{D}} = \left(t_j, \boldsymbol{y}_j, \pi_j, T_j^*, T_j\right)_{j=1,...,n_{\mathcal{D}}}$. We next center and scale state variables and times. The mean and standard deviation of sampled state variables are computed with the standard formulas:

$$\bar{\boldsymbol{y}} = \frac{1}{n_{\mathcal{D}}} \sum_{j=1}^{n_{\mathcal{D}}} \boldsymbol{y}_j \quad , \quad \boldsymbol{S}_y = \sqrt{\frac{1}{n_{\mathcal{D}} - 1} \sum_{j=1}^{n_{\mathcal{D}}} \left(\boldsymbol{y}_j - \bar{\boldsymbol{y}}\right)^2} .$$

The scaling vectors (22) are defined by $\boldsymbol{a} = -\frac{\bar{\boldsymbol{y}}}{\boldsymbol{S_y}}$ and $\boldsymbol{b} = \frac{1}{\boldsymbol{S_y}}$. For $1, ..., n_{\mathcal{D}}$, we define $\tilde{\boldsymbol{y}}_j = \boldsymbol{a} + \boldsymbol{b} \odot \boldsymbol{y}_j$. The times, contract and bond maturities are also rescaled with relations (24) and coefficients

$$a_h = -\tfrac{1}{2}, \quad b_h = \frac{1}{T^*_{max}}.$$

The set of sampled normalized state variables and parameters is denoted by:

$$\tilde{\mathcal{S}}_{\mathcal{D}} = \left(\tilde{t}_j, \tilde{\boldsymbol{y}}_j, \pi_j, \tilde{T}^*_j, \tilde{T}_j\right)_{j=1,...,n_{\mathcal{D}}}.$$

During the training phase, the error of approximation is measured for all points of this sample sets with the scaled FK equation (26). The error at expiry is measured with another sample set, denoted by $\tilde{\mathcal{S}}_{T^*}$, of n_T realizations of state variables and parameters:

$$\tilde{\mathcal{S}}_{T^*} = \left(\tilde{\boldsymbol{y}}_k, \pi_k, \tilde{T}^*_k, \tilde{T}_k\right)_{k=1,...,n_{T^*}}.$$

Let us denote by $\boldsymbol{\Theta}$, the vector containing all neural weights $(C_k, \boldsymbol{c}_k)_{k=1,...,l}$. At points of $\tilde{\mathcal{S}}_{\mathcal{D}}$, we define for $j = 1, ..., n_{\mathcal{D}}$, the error in Equation (25) when V is replaced by the neural network as follows:

$$\begin{aligned} e^{\mathcal{D}}_j(\boldsymbol{\Theta}) = b_h\, \partial_{t_j} F - \left(\frac{\tilde{r}_j - a_r}{b_r} + \frac{\tilde{\mu}_j - a_\mu}{b_\mu}\right) F \\ + \boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}(t_j, \tilde{\boldsymbol{y}}_j)^\top \nabla_{\tilde{\boldsymbol{y}}} F + \frac{1}{2}\mathrm{tr}\left(\Sigma_{\tilde{\boldsymbol{y}}}(t_j, \tilde{\boldsymbol{y}}_j)\Sigma_{\tilde{\boldsymbol{y}}}(t_j, \tilde{\boldsymbol{y}}_j)^\top \mathcal{H}_{\tilde{\boldsymbol{y}}}(F)\right). \end{aligned} \tag{28}$$

We refer to $e^{\mathcal{D}}_j$ as the error on the interior domain since it measures the goodness of fit before expiry. The average quadratic loss on $\tilde{\mathcal{S}}_{\mathcal{D}}$ is the first component of the total loss function used to fit the neural network,

$$\mathcal{L}_D(\boldsymbol{\Theta}) = \frac{1}{n_{\mathcal{D}}} \sum_{j=1}^{n_{\mathcal{D}}} e^{\mathcal{D}}_j(\boldsymbol{\Theta})^2. \tag{29}$$

As the error $e^{\mathcal{D}}_j(\boldsymbol{\Theta})$ depends on first and second order partial derivatives, a particular attention must be grant to the accuracy of their calculation. Computing them with a standard finite difference method may be the source of numerical unstabilities. We opt for an alternative approach that is called automatic or algorithmic differentiation. Automatic differentiation exploits the fact that every computer calculation executes a sequence of elementary arithmetic operations and elementary functions to compute partial derivatives with an accuracy to the working precision. Automatic differentiation is implemented in TensorFlow functions GradientTape(.) and tape.gradient(.). We refer the reader to van Merriënboer et al. (2018) for an introduction and perspectives.

On the boundary sample set $\tilde{\mathcal{S}}_{T^*}$, we define the error $e^{T^*}_k(\boldsymbol{\Theta})$ for $k = 1, ..., n_{T^*}$, as the difference between the output of the neural network and the payoff at expiry:

$$e^{T^*}_k(\boldsymbol{\Theta}) \;=\; F(t_k, \tilde{\boldsymbol{y}}_k, \boldsymbol{\pi}_k, \tilde{T}^*_k, \tilde{T}_k) - H\left(\left(\tilde{A}^*_k - a_S\right)/b_S, G_k\right).$$

The average quadratic loss at expiry is the second component of the total loss function.

$$\mathcal{L}_{T^*}(\boldsymbol{\Theta}) = \frac{1}{n_{T^*}} \sum_{k=1}^{n_{T^*}} e^{T^*}_k(\boldsymbol{\Theta})^2. \tag{30}$$

The optimal network weights are found by minimizing the losses in $\tilde{\mathcal{S}}_{\mathcal{D}}$ and $\tilde{\mathcal{S}}_{T^*}$,

$$\boldsymbol{\Theta}_{opt} = \arg\min_{\boldsymbol{\Theta}} \left[\mathcal{L}_D(\boldsymbol{\Theta}) + \mathcal{L}_{T^*}(\boldsymbol{\Theta})\right]. \tag{31}$$

In practice, the minimization is performed with a gradient descent algorithm. For an introduction, we refer e.g. to Denuit et al. (2019), section 1.6.

7 Numerical illustration

We fit a Nelson-Siegel model to the Belgian state yield curve[1] on the 1/9/23. Initial survival probabilities, ${}_tp_x^{\mu}$ and $\sigma_\mu(t)$, are fitted to male Belgian mortality rates[2]. Details are provided in Appendix D. Parameters of survival probabilities ${}_tp_x^{\lambda}$, of the reference population for the mortality bond, are assumed proportional to these of ${}_tp_x^{\mu}$. Reference ages, x_μ and x_λ, are set to 50 years. Other market parameters are estimated from daily time series of the Belgian stock index BEL 20 and of the 1 year Belgian state yield from the 1/9/10 to the 1/9/23. The correlations $\rho_{S\,\mu}$ and $\rho_{r\,\mu}$ are set to -5% and 0%. Parameter estimates are reported in Table 1.

Parameters			
κ_r	0.20482	σ_S	0.18061
ρ_{Sr}	-0.02301	σ_r	0.00786
r_0	0.03550		
x_μ	50	β_μ	0.11094
α_μ	8.5277e-7	κ_μ	0.83925
μ_0	3.325e-03	$\rho_{S\,\mu}$	-0.05000
$\rho_{r\mu}$	0.00000	$\rho_{\mu\lambda}$	0.85000
x_λ	50	β_λ	0.9 β_μ
α_μ	0.9 α_u	κ_λ	0.9 κ_μ
λ_0	0.9 μ_0	$\rho_{S\,\lambda}$	-0.05000
$\rho_{r\lambda}$	0.00000	$\rho_{\mu\lambda}$	0.85000

Table 1: Financial and biometric parameters.

Parameters for sampling contracts and risk factors				
	Minimum	Maximum	Distribution	Additional info.
t	> 0	10 years	uniform	-
T^*	$\geq t$	10 years	exponential$\left(\frac{10-t}{6}\right)$	T^* capped to 10
T	$\geq T^*$	15 years	exponential$\left(\frac{15-t}{3}\right)$	T capped to 15
A_t	23.82	414.15	uniform	$G_m = 100$, $G_M = \infty$
S_t	23.82	414.15	uniform	-
r_t	0.00	0.08	uniform	$r_0 = 4.4187\%$
μ_{x+t}	0.00	0.01	uniform	$\mu_0 = 0.1365\%$
λ_{x+t}	0.00	0.01	uniform	$\lambda_0 = 0.1229\%$
π_S	0.00	1	uniform	-
π_P	0.00	1	uniform	-
π_D	0.00	1	uniform	-

Table 2: Model parameters and features of contracts.

We first build a sample set, $\mathcal{S}_\mathcal{D}$, of contract features, asset-mixes and durations. We draw randomly $n_\mathcal{D} = 100\,000$ combinations of risk factors and investment-duration parameters. Most of these quantities are drawn from uniform distributions whose characteristics are reported in Table 2. The contract and bond maturities are drawn from exponential distributions to ensure a smoother allocation of durations than the one obtained with uniform laws. As π_S, π_P and π_D are randomly distributed in $[0, 1]$, we allow for short positions in cash. Using the same distributions as for $\mathcal{S}_\mathcal{D}$, we generate a boundary sample set $\mathcal{S}_{T^*}$ of size, $n_{T^*} = 50\,000$. The last step consists in

[1]Source: national bank of Belgium (www.nbb.be).
[2]Source: Human mortality database (www.mortality.org)

scaling and centering the sample sets, as described in the previous section. Notice that setting $G_m = 100$ is not constraining. As the payoff is piecewise linear, we can use the rule of thumb to evaluate any contract with another minimum capital. If $V(A_t, G_m)$ is the GMAB price at time t, for an asset value A_t and a guarantee G_m, the value of a contract with a guaranteed capital $G_m^{'} \neq G_m$ is equal to:

$$V(A_t, G_m^{'}) = \frac{G_m^{'}}{G_m} V\left(\frac{G_m}{G_m^{'}} A_t, G_m\right).$$

The neural network is implemented in Tensorflow under Python. As previously mentioned, the partial derivatives needed for evaluating the inner loss, are computed by automatic differentiation. This procedure, implemented in functions GradientTape(.) and tape.gradient(.), allows to calculate derivatives with a high accuracy as it is not based on finite differences. To train the network, we use the ADAM optimizer and random batches containing 34 000 and 17 000 samples from $\tilde{\mathcal{S}}_{\mathcal{D}}$ and $\tilde{\mathcal{S}}_{T^*}$. Over one epoch, neural weights are therefore updated three times. We run 26 000 epochs and adjust the ADAM learning rate to speed up the calibration according to the pattern provided in Table 3. The training is performed on a laptop equiped with a processor AMD Ryzen 7 5825U and 16 Gb of RAM (without cuda compatible GPU). The training time depends upon the network configuration. As reported in Table 4, running 200 epochs respectively takes respectively 12 and 18 minutes long for a network with 3 and 5 five hidden layers with 32 neurons. The total calibration time varies from 26 up to 40 hours. Choosing 32 neurons per hidden layer is a nice trade-off between accuracy and training time. Tests performed with 16 neurons over 8000 epochs reveals that this configuration underperforms compared to those with 32 neurons.

Epochs	Learning rate
4 000	0.01
4 001 - 8 000	0.005
8 001 - 14 000	0.001
14 001 - 20 000	0.0005
20 001 - 26 000	0.0001

Table 3: Learning rate in function of Epochs

Number of hidden layers	Neurons per layer	Times in sec., 200 epochs	Number of parameters	Total loss	$\mathcal{L}_D(\boldsymbol{\Theta})$	$\mathcal{L}_{T^*}(\boldsymbol{\Theta})$
3	32	685	4641	0.0437	0.0269	0.0168
4	32	853	6049	0.0093	0.0055	0.0038
5	32	1012	7457	0.0077	0.0046	0.0031

Table 4: Information about training time and loss values after training for three neural configurations.

Table 4 reports the losses on $\tilde{\mathcal{S}}_{\mathcal{D}}$, $\tilde{\mathcal{S}}_{T^*}$ and their total, computed with equations (29) and (30). We observe a significant reduction of losses between 3 and 4 hidden layers models with 32 neurons. The reduction is relatively smaller when we switch from a 4 to a 5 layers model.

Table 5 presents statistics about relative errors[3] between PINN and exact prices obtained with formula (18). These statistics are computed for the 100 000 contracts in the training data set $\tilde{\mathcal{S}}_{\mathcal{D}}$ and for a validation set of same size and built in the same manner. The similarity of figures on both sets does not reveal that we overfit the training dataset. The average relative errors

[3]By relative absolute error, we mean $\frac{|\text{Exact price} - \text{PINN price}|}{\text{Exact price}}$.

are small and around 0.15% for 4 and 5 hidden layers models. The 95% confidence level is also below 1% for these networks which is quite acceptable.

Relative errors, training set							
Number of hidden layers	Mean	Standard Deviation	Percentiles 5%	25%	50%	75%	95%
3	0.0027	0.0071	2.05e-05	10.88e-05	30.30e-05	147.51e-05	1.51e-02
4	0.0014	0.0057	7.63e-06	3.90e-05	9.23e-05	32.60e-05	0.74e-02
5	0.0016	0.0057	7.39e-06	3.96e-05	10.40e-05	52.50e-05	0.87e-02
Relative errors, validation set							
Number of hidden layers	Mean	Standard Deviation	Percentiles 5%	25%	50%	75%	95%
3	0.0027	0.0070	2.03e-05	10.84e-05	30.44e-05	148.94e-05	1.50e-02
4	0.0014	0.0055	7.73e-06	3.91e-05	9.28e-05	32.93e-05	0.73e-02
5	0.0016	0.0056	7.45e-06	3.93e-05	10.43e-05	52.45e-05	0.86e-02

Table 5: Relative errors computed on the 100 000 contracts of the training set $\tilde{\mathcal{S}}_{\mathcal{D}}$ and on a validation set of same size.

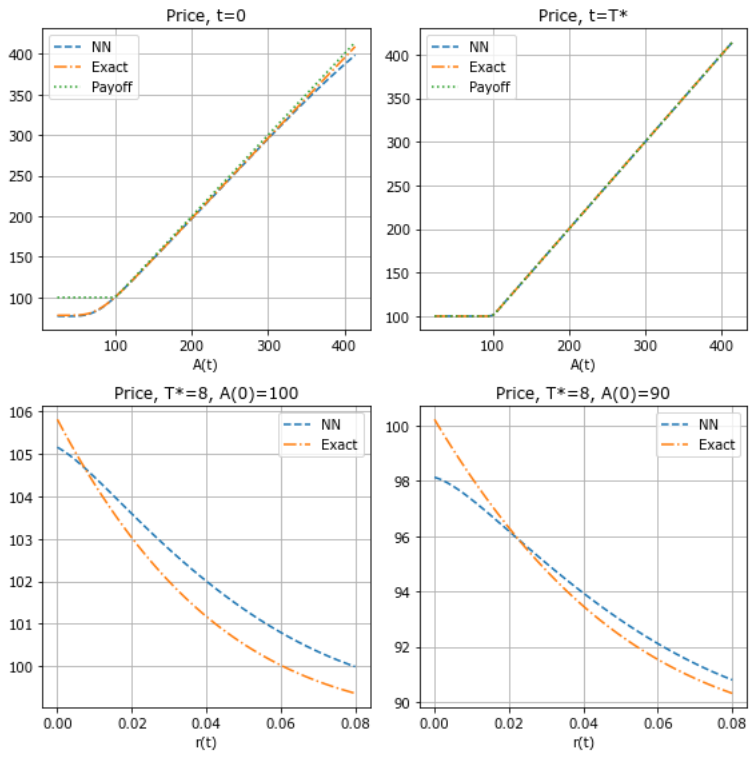

Figure 2: Comparison of exact and PINN prices of a $T^* = T = 8$ years GMAB with: $S_0 = 100$, $\boldsymbol{\pi} = (0.50, 0.25, 0.25)$, (r_0, μ_0, λ_0) of Table 2. Upper plots: values at issuance and expiry, for $A_t \in [23.82, 414.15]$. Lower plots: values at issuance with $A_0 = 100$ and 90, for $r_0 \in [0.00, 0.08]$.

In the remainder of this section, we focus on the network with 4 hidden layers since it yields the lowest relative errors on training and validation sets. Figure 2 compares exact and approached PINN prices of a 8 years GMAB. Upper plots show the value at issuance and at expiry for various values of the total asset. Market conditions are these on the 1/9/23. For a wide range of asset values, exact and PINN prices are very close. The largest deviations are observed for the lowest

and highest values of A_t. The lower plots present the value at issuance of the same contract with $A_0 \in \{90, 100\}$ for r_0 between 0% and 8%. We observe that spreads between true and approximated prices depends on both parameters. We nevertheless notice that the error stays under 1% in most of cases.

To understand in which circumstances, relative errors are the highest, we analyse it by sub-groups of contracts. In order to count enough representants in each sub-category, we regenerate a third validation set with this time 500 000 contracts instead of 100 000. Table 6 presents the averages and standard deviations of relative errors for 20 classes of contracts, regrouped by asset values. We observe that the highest errors are obtained with the lowest asset values. The left heatmap in Figure 3 shows relative errors per categories of asset values and residual contract maturities, $T^* - t$. We draw two conclusions from this graph. Firstly, for $A_t \leq 80$, the error increases with the residual contract duration and may be significant. Secondly, the errors in the usual pricing range, around the guarantee level $G_m = 100$, is close to 0.01% whatever the time horizon. Notice that for low asset values, GMAB is deeply out of the money and the contract price is very close to the one of a pure endowment for which we have closed-form expression. A way to improve the accuracy would be to add a third penalty term to the loss function (31). This term would penalize the quadratic spread between PINN and endowment prices for low asset values. We deliberately avoided this solution because we wanted to see how the network behaves when provided with the minimum amount of information possible.

$A_t \in$	[23.8, 43.3)	[43.3, 62.8)	[62.8, 82.3)	[82.3, 101.8)	[101.8, 121.4)
mean	0.0075	0.0072	0.0067	0.0039	0.001
std	0.0133	0.0129	0.0114	0.0062	0.0021
$A_t \in$	[121.4, 140.9)	[140.9, 160.4)	[160.4, 179.9)	[179.9, 199.5)	[199.5, 219.0)
mean	0.0003	0.0002	0.0001	0.0001	0.0001
std	0.0009	0.0006	0.0004	0.0002	0.0002
$A_t \in$	[219.0, 238.5)	[238.5, 258.0)	[258.0, 277.5)	[277.5, 297.0)	[297.0, 316.5)
mean	0.0001	0.0001	0.0001	0.0001	0.0001
std	0.0001	0.0001	0.0002	0.0002	0.0003
$A_t \in$	[316.5, 336.1)	[336.1, 355.6)	[355.6, 375.1)	[375.1, 394.6)	[394.6, 414.2)
mean	0.0001	0.0001	0.0002	0.0003	0.0004
std	0.0005	0.0005	0.0008	0.0012	0.0015

Table 6: Mean and standard deviations (std) of relative errors for contracts grouped by asset values.

$r_t \in$ (in %)	[0.0, 0.8)	[0.8, 1.6)	[1.6, 2.4)	[2.4, 3.2)	[3.2, 4.0)
mean	0.0017	0.0016	0.0015	0.0014	0.0014
std	0.0067	0.0063	0.0058	0.0058	0.0055
$r_t \in$(in %)	[4.0, 4.8)	[4.8, 5.6)	[5.6, 6.4)	[6.4, 7.2)	[7.2, 8.0)
mean	0.0014	0.0013	0.0013	0.0014	0.0013
std	0.0055	0.0052	0.0054	0.0054	0.0052

Table 7: Mean and standard deviations (std) of relative errors for contracts grouped by interest rates.

Table 7 reports means and standard deviations of errors for 10 groups of contracts, classed by ranges of interest rates, r_t. Whatever the category, the average relative error remains close to 0.14%. The right heatmap of Figure 3 seems to reveal that on average the errors increases

with the maturity. This conclusion must nevertheless be qualified by the observation of a high variability of errors between consecutive interest rate classes for the same maturity. This indicates that the increase is partly caused by other factors such as the low asset values present in each of interest rate classes.

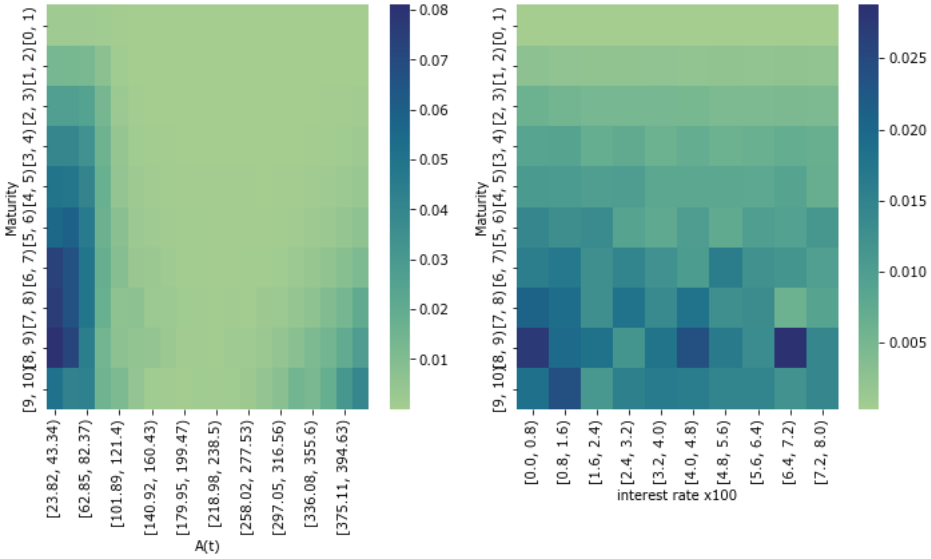

Figure 3: Left plot: heatmap of relative errors per asset and maturity classes. Right plot: heatmap of relative errors per interest rate and maturity classes.

$\mu_t \in$	[0.0, 1.0)	[1.0, 2.0)	[2.0, 3.0)	[3.0, 4.0)	[4.0, 5.0)
mean	0.0015	0.0014	0.0015	0.0014	0.0015
std	0.0056	0.0057	0.0059	0.0053	0.0059
$\mu_t \in$	[5.0, 6.0)	[6.0, 7.0)	[7.0, 8.0)	[8.0, 9.0)	[9.0, 10.0)
mean	0.0014	0.0014	0.0014	0.0014	0.0014
std	0.0058	0.0058	0.0057	0.0055	0.0057

Table 8: Mean and standard deviations (std) of relative errors for contracts grouped by mortality rates μ_{x+t}.

$\lambda_t \in$	[0.0, 1.0)	[1.0, 2.0)	[2.0, 3.0)	[3.0, 4.0)	[4.0, 5.0)
mean	0.0014	0.0014	0.0014	0.0014	0.0014
std	0.0057	0.0055	0.0055	0.0056	0.0055
$\lambda_t \in$	[5.0, 6.0)	[6.0, 7.0)	[7.0, 8.0)	[8.0, 9.0)	[9.0, 10.0)
mean	0.0014	0.0015	0.0015	0.0014	0.0015
std	0.0058	0.006	0.0059	0.0058	0.0059

Table 9: Mean and standard deviations (std) of relative errors for contracts grouped by hedging mortality rates λ_{x+t}.

Tables 8 and 9 provide information about errors for contracts grouped by, classes of mortality rates. As for interest rate, the average error is small and close to 0.14% whatever the class. The heatmaps of Figure 4 shows that errors slightly rise with the contract time to expiry. Again, we observe a higher variability of errors for longest maturities. As earlier mentioned, this is the signal that the increase of errors is also caused by other factors.

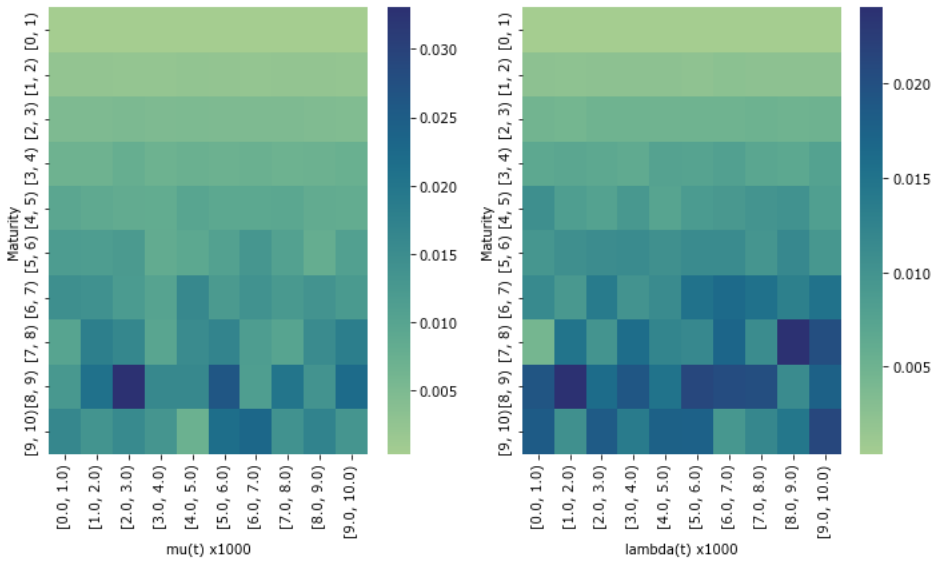

Figure 4: Left plot: heatmap of relative errors per μ_{x+t} (in °/oo) and maturity classes. Right plot: heatmap of relative errors per λ_{x+t} (in °/oo) and maturity classes.

$\pi_S \in$	$[0.0, 0.1)$	$[0.1, 0.2)$	$[0.2, 0.3)$	$[0.3, 0.4)$	$[0.4, 0.5)$
mean	0.0014	0.0014	0.0014	0.0014	0.0014
std	0.0059	0.0059	0.0057	0.0058	0.0057
$\pi_S \in$	$[0.5, 0.6)$	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 0.9)$	$[0.9, 1.0)$
mean	0.0014	0.0014	0.0015	0.0015	0.0015
std	0.0057	0.0056	0.0057	0.0055	0.0055

Table 10: Mean and standard deviations (std) of relative errors for contracts grouped by fraction of stocks

$(\pi_P + \pi_D) \in$	$[0.0, 0.2)$	$[0.2, 0.4)$	$[0.4, 0.6)$	$[0.6, 0.8)$	$[0.8, 1.0)$
mean	0.0016	0.0015	0.0015	0.0014	0.0014
std	0.006	0.0058	0.0058	0.0056	0.0057
$(\pi_P + \pi_D) \in$	$[1.0, 1.2)$	$[1.2, 1.4)$	$[1.4, 1.6)$	$[1.6, 1.8)$	$[1.8, 2.0)$
mean	0.0014	0.0014	0.0014	0.0014	0.0014
std	0.0057	0.0056	0.0056	0.0055	0.0053

Table 11: Mean and standard deviations (std) of relative errors for contracts grouped by fraction of bonds

Tables 10 and 11 provide statistics about relative errors for contracts classed by fractions of stocks and bonds (interest rate and mortality bonds). The left plot of Figure 5 shows the heatmap of errors according to these two parameters. We do not observe any particular trend and the pricing error is acceptable whatever the asset-mix. This indicates that the PINN can be used for ALM purposes and in particular to approximately estimates the impact of a change of asset-mix on contract fair values. Nevertheless, the right heatmap of Figure 5 reveals that the error rises with the time to expiry. To illustrate this, we compare in Figure 6, the exact and PINN prices of a 5 and 8 years contract, for π_S ranging from 0 to 1 and $\pi_D = \pi_M = \frac{1-\pi_S}{2}$. This graph reveals PINN captures the trend of the relationship between prices and fraction of stocks but the error may reach 1% for some combinations of parameters and risk factors. This error may appear relatively high but this must be put into perspective in relation to existing

alternatives. For instance, in the context of risk management, a common approach to value a contract at a future date relies on the Least Square Monte-Carlo (LSMC) method. As shown in Hainaut and Akbaraly (2023) for a similar contract, the average valuation error is often above 1% and even climbs above 3% in extreme cases. Substituting the PINN to the LSMC in such a context, would clearly reduce the global valuation error.

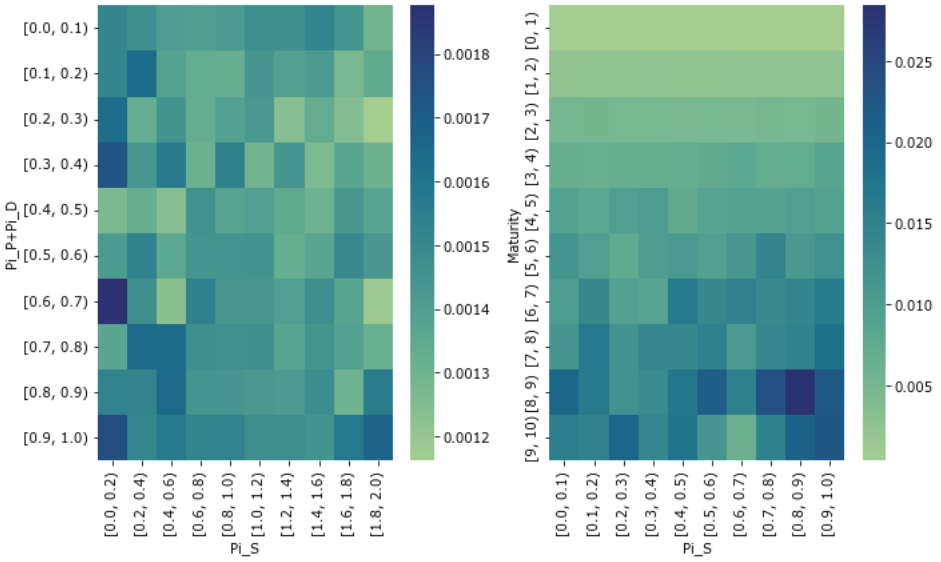

Figure 5: Heatmap of relative errors per π_S and $(\pi_P + \pi_D)$ asset allocation classes.

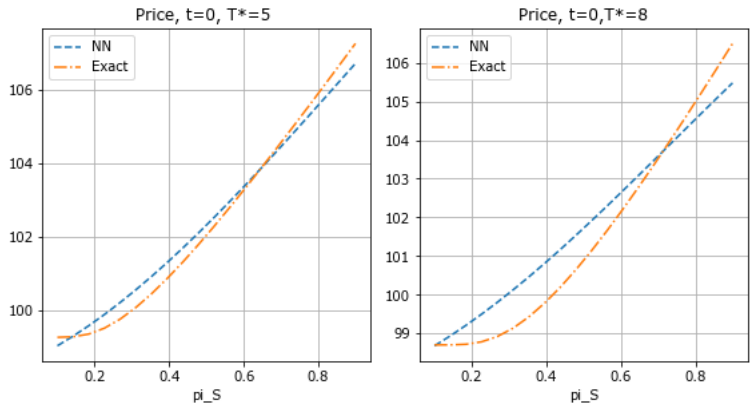

Figure 6: Comparison of exact and PINN prices of a 5 and 8 years GMAB with: $S_0 = 100$, $\pi_S \in [0, 1]$ and $\pi_D = \pi_M = \frac{1-\pi_S}{2}$. (r_0, μ_0, λ_0) of Table 2. Upper plots: values at issuance and expiry, for $A_t \in [23.82, 414.15]$. Lower plots: values at issuance with $A_0 = 100$ and 90, for $r_0 \in [0.00, 0.08]$.

We conclude from the error analysis that the average accuracy of a PINN is sufficient for risk management and ALM purposes. The global benefit of using such approach is the considerable gain of time once the calibration is performed. Computing prices on a sample set of 500 000 contracts, with different durations, asset-mix and market conditions, takes less than 45 seconds. Evaluating the same portfolio with a standard Monte-Carlo method would require a considerable amount of time as we need to regenerate a sufficient number of sample paths for each initial combination of risk factors. The PINN model is able to price an impressive range of contracts and with more computing power, it is probably possible to improve both the accuracy and the variety of contracts.

8 Conclusions

We explore in this work the capacity of physics inspired neural networks (PINN's) to price guaranteed minimum accumulation benefits. The first part of the article introduces market and biometric risk factors. We consider a Gaussian framework and develop an analytical formula for GMAB pricing to benchmark the PINN. We next propose a scaled version of the Feynman-Kac (FK) equation, which allows to limit the phenomenon of vanishing gradient during the training. An approximate solution is built with a neural network and skip connections. We conclude by a detailed analysis of pricing errors.

This study reveals that PINN's present several advantages. The first one is their ability to evaluate products exposed to multiple risk factors. Alternative procedures, e.g. based on finite differences or FFT, are difficult to implement and time consuming for more than 2 risk factors. Secondly, the PINN can be parameterized with e.g. the contract duration or asset-mix. This allows a great flexibility compared to other methods that must be run again for every different combination of contract features. Thirdly, after training, the valuation is quasi-instantaneous. We appraise in less than a minute, a portfolio of 500 000 GMAB's with various market conditions and features. The same exercise with a Monte-Carlo method would require to simulate sample paths for each combination of initial risk factors. Nevertheless, PINN's present drawbacks. The first one is the absence of systematic procedure to determine the optimal architecture of the network and the size of the training set. After several trials, we opt of a network with direct connections between input and hidden layers. Secondly, the training time is high but may probably be reduced by using parallelization on GPU's. On the other hand, we may expect that the update of the network to new market conditions, would be less time consuming when initialized with neural weights obtained after a first training.

Numerical results are clearly encouraging. In the usual pricing range of risk factors, a PINN with four intermediate layers approximates the price with a relative error smaller than ten basis points. The PINN also captures the dependence between GMAB prices, risk factors and asset-mix. The PINN accuracy is sufficient to perform a sensitivity analysis to contract specifications and to test different asset-allocation strategies. Nevertheless, we observe non-negligible errors for very low values of assets. For these values, the embedded call in the GMAB is deeply out of the money and the contract price is very close to the one of a pure endowment. We also notice that errors slightly rise with the contract maturity. In certain configurations, errors may appear relatively high but this must be put into perspective in relation to existing alternatives. For instance, in the context of risk management, a common approach to value a contract at a future date relies on the Least Square Monte-Carlo (LSMC) method which is much more inaccurate (see Hainaut and Akbaraly 2023).

This article paves the way of future experiments. Firstly, errors for deep out of the money contracts can be reduced by adding a third penalty term to the loss function (31). As mentioned above, the GMAB value is in this case nearly equal to the value of a pure endowment, which admits an analytical formula. We can therefore add a term to the loss, penalizing the quadratic spread between PINN and endowment prices when the asset value is below a certain threshold. We deliberately avoided this solution because we wanted to see how the network behaves when provided with the minimum amount of information possible. Secondly, it would be interesting to extend the set of parameters and contract features to generalize the procedure. Finally, we haven't explored all the possible neural architectures and maybe that we can optimize it to improve accuracy and reduce computing time.

Appendix A

The next proposition allows us to infer the expressions of $\gamma_r(t)$ and $\gamma_{z\in\{\mu,\lambda\}}(t)$ matching the initial term structures of interest and mortality rates.

Proposition 4. *At time $0 \leq t \leq T$, the value of the discount bond of maturity T is equal to*

$$\begin{aligned} P(t,T) &= \exp\left(-r_t B_{\kappa_r}(t,T) - \int_t^T \gamma_r(u)\left(1-e^{-\kappa_r(T-u)}\right)du\right) \\ &\quad \times \exp\left(\frac{\sigma_r^2}{2}\int_t^T B_{\kappa_r}(u,T)^2 du\right), \end{aligned} \tag{32}$$

The survival probability up to time T for $z \in \{\mu,\lambda\}$, is given by

$$\begin{aligned} {}_Tp^z_{x_z+t} &= \exp\left(-z_{x_z+t} B_{\kappa_z}(t,T) - \int_t^T \gamma_z(u)\left(1-e^{-\kappa_z(T-u)}\right)du\right) \\ &\quad \times \exp\left(\frac{1}{2}\int_t^T \left(\sigma_z(u)B_{\kappa_z}(u,T)\right)^2 du\right), \end{aligned} \tag{33}$$

The pure endowments admit the following expressions:

$$\begin{aligned} {}_TE^{\mu}_t &= \mathbf{1}_{\{\tau_\mu \geq t\}} \exp\left(-r_t B_{\kappa_r}(t,T) - \mu_{x_\mu+t} B_{\kappa_\mu}(t,T) + \frac{\sigma_r^2}{2}\int_t^T B_{\kappa_r}(u,T)^2 du\right) \times \\ &\exp\left(-\int_t^T \gamma_r(u)\left(1-e^{-\kappa_r(T-u)}\right)du - \int_t^T \gamma_\mu(u)\left(1-e^{-\kappa_\mu(T-u)}\right)du\right) \times \\ &\exp\left(\sigma_r \Sigma_r^\top \Sigma_\mu \int_t^T \sigma_\mu(u) B_{\kappa_\mu}(u,T) B_{\kappa_r}(u,T) du + \frac{1}{2}\int_t^T \left(\sigma_\mu(u) B_{\kappa_\mu}(u,T)\right)^2 du\right), \end{aligned} \tag{34}$$

and

$$\begin{aligned} {}_TE^{\lambda}_t &= \mathbf{1}_{\{\tau_\lambda \geq t\}} \exp\left(-r_t B_{\kappa_r}(t,T) - \lambda_{x_\lambda+t} B_{\kappa_\lambda}(t,T) + \frac{\sigma_r^2}{2}\int_t^T B_{\kappa_r}(u,T)^2 du\right) \times \\ &\exp\left(-\int_t^T \gamma_r(u)\left(1-e^{-\kappa_r(T-u)}\right)du - \int_t^T \gamma_\lambda(u)\left(1-e^{-\kappa_\lambda(T-u)}\right)du\right) \times \\ &\exp\left(\sigma_r \Sigma_r^\top \Sigma_\lambda \int_t^T \sigma_\lambda(u) B_{\kappa_\lambda}(u,T) B_{\kappa_r}(u,T) du + \frac{1}{2}\int_t^T \left(\sigma_\lambda(u) B_{\kappa_\lambda}(u,T)\right)^2 du\right), \end{aligned} \tag{35}$$

Sketch of the proof. We can show by direct differentation that interest and mortality rates are equal to

$$\begin{aligned} \begin{pmatrix} r_s \\ \mu_{x_\mu+s} \\ \lambda_{x_\lambda+s} \end{pmatrix} &= \begin{pmatrix} e^{-\kappa_r(s-t)} r_t \\ e^{-\kappa_\mu(s-t)} \mu_{x_\mu+t} \\ e^{-\kappa_\lambda(s-t)} \lambda_{x_\lambda+t} \end{pmatrix} + \begin{pmatrix} \kappa_r \int_t^s \gamma_r(u) e^{-\kappa_r(s-u)} du \\ \kappa_\mu \int_t^s \gamma_\mu(u) e^{-\kappa_\mu(s-u)} du \\ \kappa_\lambda \int_t^s \gamma_\lambda(u) e^{-\kappa_\lambda(s-u)} du \end{pmatrix} \\ &\quad + \begin{pmatrix} \sigma_r \Sigma_r^\top \int_t^s e^{-\kappa_r(s-u)} d\boldsymbol{W}_u \\ \Sigma_\mu^\top \int_t^s \sigma_\mu(u) e^{-\kappa_\mu(s-u)} d\boldsymbol{W}_u \\ \Sigma_\lambda^\top \int_t^s \sigma_\lambda(u) e^{-\kappa_\lambda(s-u)} d\boldsymbol{W}_u \end{pmatrix} \end{aligned} \tag{36}$$

The integrals of interest and mortality rates are obtained by direct integration

$$\begin{aligned} \begin{pmatrix} \int_t^T r_s ds \\ \int_t^T \mu_{x_\mu+s} ds \\ \int_t^T \lambda_{x_\lambda+s} ds \end{pmatrix} &= \begin{pmatrix} r_t B_{\kappa_r}(t,T) \\ \mu_{x_\mu+t} B_{\kappa_\mu}(t,T) \\ \lambda_{x_\lambda+t} B_{\kappa_\lambda}(t,T) \end{pmatrix} + \begin{pmatrix} \int_t^T \gamma_r(u)\left(1-e^{-\kappa_r(T-u)}\right)du \\ \int_t^T \gamma_\mu(u)\left(1-e^{-\kappa_\mu(T-u)}\right)du \\ \int_t^T \gamma_\lambda(u)\left(1-e^{-\kappa_\lambda(T-u)}\right)du \end{pmatrix} \\ &\quad + \begin{pmatrix} \sigma_r \Sigma_r^\top \int_t^s B_{\kappa_r}(u,T)\, d\boldsymbol{W}_u \\ \Sigma_\mu^\top \int_t^s \sigma_\mu(u) B_{\kappa_\mu}(u,T)\, d\boldsymbol{W}_u \\ \Sigma_\lambda^\top \int_t^s \sigma_\lambda(u) B_{\kappa_\lambda}(u,T)\, d\boldsymbol{W}_u \end{pmatrix}. \end{aligned} \tag{37}$$

The results follow from the log-normality of $e^{-\int_t^T z_s ds}$ for $z \in \{r, \mu, \lambda\}$, from $\epsilon_{rr}^2 + \epsilon_{r\mu}^2 + \epsilon_{r\lambda}^2 = 1$ and $\epsilon_{\mu\mu}^2 + \epsilon_{\mu\lambda}^2 = 1$.

end

From Equation (32), we deduce that the function $\gamma_r(u)$ must satisfy the next relation to match the initial yield curve of zero-coupon bond:

$$\int_0^T \gamma_r(u) \left(1 - e^{-\kappa_r(T-u)}\right) du \quad = \quad -\ln P(0,T) - r_0 B_{\kappa_r}(0,T) + \frac{\sigma_r^2}{2} \int_0^T B_{\kappa_r}(u,T)^2 du \,. \tag{38}$$

Deriving twice this expression leads to the following useful reformulation of $\gamma_r(T)$:

$$\gamma_r(T) \quad = \quad -\frac{1}{\kappa_r} \partial_T^2 \ln P(0,T) - \partial_T \ln P(0,T) + \frac{\sigma_r^2}{2\kappa_r^2} \left(1 - e^{-2\kappa_r T}\right) , \tag{39}$$

where $-\partial_T \ln P(0,T)$ is the instantaneous forward rate. For a given initial mortality curve ${}_T p_{x_z}^z$, $z \in \{\mu, \lambda\}$, we show in a similar manner that the function $\gamma_z(u)$ satisfies the relation

$$\begin{aligned} \gamma_z(T) \quad &= \quad -\frac{1}{\kappa_z} \partial_T^2 \ln {}_T p_{x_z}^z - \partial_T \ln {}_T p_{x_z}^z + \frac{1}{\kappa_z} \int_0^T \sigma_z(u)^2 \left(e^{-2\kappa_z(T-u)}\right) du \\ &= \quad -\frac{1}{\kappa_z} \partial_T^2 \ln {}_T p_{x_z}^z - \partial_T \ln {}_T p_{x_z}^z + \frac{\alpha_z^2 e^{2\beta_z x_z}}{2\kappa_z(\kappa_z + \beta_z)} \left(e^{2\beta_z T} - e^{-2\kappa_z T}\right) . \end{aligned} \tag{40}$$

Equations (39) and (40) allows us to rewrite bond prices, survival probabilities and endowments as function of initial term structures of mortality and interest rates. Analytical expressions are provided in Proposition **1** the proof is summarized below.

Proposition 1 : sketch of the proof By direct integration of Equations (39) and (40), we obtain that

$$\begin{aligned} \int_t^T \gamma_r(u) \left(1 - e^{-\kappa_r(T-u)}\right) du \quad &= \quad \left(\partial_t \ln P(0,t)\right) B_{\kappa_r}(t,T) - \ln \frac{P(0,T)}{P(0,t)} + \frac{\sigma_r^2}{2\kappa_r^2}(T-t) \\ &\quad -\frac{\sigma_r^2}{2\kappa_r^2} B_{\kappa_r}(t,T) - \frac{\sigma_r^2}{4\kappa_r} e^{-2\kappa_r t} B_{\kappa_r}(t,T)^2 \,, \end{aligned}$$

and

$$\begin{aligned} &\int_t^T \gamma_z(u) \left(1 - e^{-\kappa_z(T-u)}\right) du = \left(\partial_t \ln {}_t p_{x_z}^z\right) B_{\kappa_z}(t,T) - \ln \frac{{}_T p_{x_z}^z}{{}_t p_{x_z}^z} + \frac{\alpha_z^2 e^{2\beta_z x_z}}{2\kappa_z(\kappa_z + \beta_z)} \\ &\times \left(e^{2\beta_z T} B_{2\beta_z}(t,T) - e^{-2\kappa_z t} B_{2\kappa_z}(t,T) - e^{2\beta_z T} B_{2\beta_z + \kappa_z}(t,T) + e^{-\kappa_z(T+t)} B_{\kappa_z}(t,T)\right) \end{aligned}$$

Combining these expressions with these of Proposition 4.

end

Proposition 2 : sketch of the proof Let us denote the mortality indexed discount factor by ${}_T m_t^\lambda = \mathbb{E}^{\mathbb{Q}} \left(e^{-\int_t^T (r_s + \lambda_{x_\lambda + s}) ds} | \mathcal{F}_t\right)$ such that $D(t,T) = e^{-\int_0^t \lambda_{x_\lambda + s} ds} {}_T m_t^\lambda$. The differential of $D(t,T)$ then equal to

$$dD(t,T) \quad = \quad -\lambda_{x_\lambda + t} {}_T m_t^\lambda \, dt + e^{-\int_0^t \lambda_{x_\lambda + s} ds} d\, {}_T m_t^\lambda \,. \tag{41}$$

From Proposition 5, we infer that the mortality indexed discount factor is ruled by the SDE:

$$\frac{d\, {}_T m_{x+t}^\lambda}{{}_T m_{x+t}^\lambda} = (r_t + \lambda_{x_\lambda + t}) \, dt - B_{\kappa_r}(t,T) \sigma_r \Sigma_r^\top d\boldsymbol{W}_t - B_{\kappa_\lambda}(t,T) \sigma_\lambda(t) \Sigma_\lambda^\top d\boldsymbol{W}_t \,.$$

From Eq. (41), we infer the dynamics under $\mathbb{Q}$, Eq. (8). As $d\boldsymbol{W}_t = d\tilde{\boldsymbol{W}}_t + \boldsymbol{\theta} dt$ with

$$\begin{aligned} \nu_r &= -\Sigma_r^\top \boldsymbol{\theta}\,, \\ \nu_\lambda &= -\Sigma_\lambda^\top \boldsymbol{\theta}\,, \end{aligned}$$

we obtain Eq. (9).

end

The next result presents the dynamics of the discount bond and endowment under the pricing measure. This is a direct consequence of the Itô's lemma applied to Proposition 4.

Proposition 5. *Under the risk neutral measure $\mathbb{Q}$, the dynamics of the zero-coupon bond and of the pure endowment at time $t \leq T$ are given by*

$$\begin{cases} \frac{dP(t,T)}{P(t,T)} & = r_t\, dt - B_{\kappa_r}(t,T)\sigma_r \Sigma_r^\top d\boldsymbol{W}_t\,, \\ \frac{d\,{}_T E_t^\mu}{{}_T E_t^\mu} & = \left(r_t + \mu_{x_\mu+t}\right) dt + d\mathbf{1}_{\{\tau_\mu \geq t\}} - B_{\kappa_r}(t,T)\sigma_r \Sigma_r^\top d\boldsymbol{W}_t \\ & \quad - B_{\kappa_\mu}(t,T)\sigma_\mu(t)\Sigma_\mu^\top d\boldsymbol{W}_t \\ \frac{d\,{}_T E_t^\lambda}{{}_T E_t^\lambda} & = \left(r_t + \lambda_{x+t}\right) dt + d\mathbf{1}_{\{\tau_\lambda \geq t\}} - B_{\kappa_r}(t,T)\sigma_r \Sigma_r^\top d\boldsymbol{W}_t \\ & \quad - B_{\kappa_\lambda}(t,T)\sigma_\lambda(t)\Sigma_\lambda^\top d\boldsymbol{W}_t \end{cases} \tag{42}$$

As $\mathbb{E}^{\mathbb{Q}}\left(d\mathbf{1}_{\{\tau_\mu \geq t\}}\right) = -\mu_{x+t}dt$ for , we check that the pure endowment has a return equal to the risk free rate: $\mathbb{E}^{\mathbb{Q}}\left(d\,{}_T E_t^\mu\right) = {}_T E_t^\mu\, r_t\, dt$.

Proposition 3 : sketch of the proof From the Itô's lemma, the dynamic of $\frac{1}{P(t,T^*)}$ under the risk neutral measure $\mathbb{Q}$ is equal to

$$d\frac{1}{P(t,T^*)} = -\frac{r_t}{P(t,T^*)}dt + \frac{B_{\kappa_r}(t,T^*)^2\sigma_r^2}{P(t,T^*)}dt + \frac{B_{\kappa_r}(t,T^*)}{P(t,T^*)}\sigma_r\Sigma_r^\top d\boldsymbol{W}_t\,.$$

From Equation (13), we infer that under $\mathbb{F}$, this ratio is ruled by

$$\begin{aligned} d\frac{1}{P(t,T^*)} &= -\frac{r_t}{P(t,T^*)}dt + \frac{B_{\kappa_r}(t,T^*)^2\sigma_r^2}{P(t,T^*)}dt + \frac{B_{\kappa_r}(t,T^*)}{P(t,T^*)}\sigma_r\Sigma_r^\top d\boldsymbol{W}_t^{\mathbb{F}} \\ &\quad -\frac{B_{\kappa_r}(t,T^*)}{P(t,T^*)}\sigma_r\left(\sigma_r B_{\kappa_r}(t,T^*) + \sigma_\mu(t)\Sigma_r^\top\Sigma_\mu B_{\kappa_\mu}(t,T^*) + \sigma_\lambda(t)\Sigma_r^\top\Sigma_\lambda B_{\kappa_\lambda}(t,T^*)\right)dt\,. \end{aligned} \tag{43}$$

On the other hand, the differential of $\frac{A_t}{P(t,T^*)}$ is related to dA_t and $d\left(1/P(t,T^*)\right)$ by the relation:

$$d\left(\frac{A_t}{P(t,T^*)}\right) = A_t\, d\left(\frac{1}{P(t,T^*)}\right) + \frac{dA_t}{P(t,T^*)} + \boldsymbol{\pi}^\top\Sigma_A(t)\Sigma_r\sigma_r B_{\kappa_r}(t,T^*)\frac{A_t}{P(t,T^*)}\,dt\,.$$

From Equations (43) and (14), we rewrite this last expression as follows:

$$\begin{aligned} \frac{d\left(A_t/P(t,T^*)\right)}{A_t/P(t,T^*)} &= -\boldsymbol{\pi}^\top\Sigma_A(t)\left(\sigma_\mu(t)B_{\kappa_\mu}(t,T^*)\Sigma_\mu + \sigma_\lambda(t)B_{\kappa_\lambda}(t,T^*)\Sigma_\lambda\right)dt \\ &\quad -B_{\kappa_r}(t,T^*)\sigma_r\left(\sigma_\mu(t)\Sigma_r^\top\Sigma_\mu B_{\kappa_\mu}(t,T^*) + \sigma_\lambda(t)\Sigma_r^\top\Sigma_\lambda B_{\kappa_\lambda}(t,T^*)\right)dt \\ &\quad +\left(B_{\kappa_r}(t,T^*)\sigma_r\Sigma_r^\top + \boldsymbol{\pi}^\top\Sigma_A(t)\right)d\boldsymbol{W}_t^{\mathbb{F}}\,. \end{aligned}$$

Applying the Itô's lemma to $\ln\left(A_t/P(t,T^*)\right)$, leads to the following result:

$$\begin{aligned} d\ln\left(A_t/P(t,T^*)\right) &= -\boldsymbol{\pi}^\top\Sigma_A(t)\left(\sigma_\mu(t)B_{\kappa_\mu}(t,T^*)\Sigma_\mu + \sigma_\lambda(t)B_{\kappa_\lambda}(t,T^*)\Sigma_\lambda\right)dt \\ &\quad -B_{\kappa_r}(t,T^*)\sigma_r\left(\sigma_\mu(t)\Sigma_r^\top\Sigma_\mu B_{\kappa_\mu}(t,T^*) + \sigma_\lambda(t)\Sigma_r^\top\Sigma_\lambda B_{\kappa_\lambda}(t,T^*)\right)dt \\ &\quad -\frac{1}{2}\left(B_{\kappa_r}(t,T^*)\sigma_r\Sigma_r^\top + \boldsymbol{\pi}^\top\Sigma_A(t)\right)\left(B_{\kappa_r}(t,T^*)\sigma_r\Sigma_r + \Sigma_A(t)^\top\boldsymbol{\pi}\right)dt \\ &\quad +\left(B_{\kappa_r}(t,T^*)\sigma_r\Sigma_r^\top + \boldsymbol{\pi}^\top\Sigma_A(t)\right)d\boldsymbol{W}_t^{\mathbb{F}}\,. \end{aligned}$$

This last equation emphasizes that $\ln \frac{A_{T^*}/P(T^*,T^*)}{A_t/P(t,T^*)} \sim N\left(\mu_{\mathbb{F}}(t), v_{\mathbb{F}}(t)\right)$.
end

Appendix B

This appendix provides various integrals used for calculating analytical prices. For $t \leq T^*$, $T^* \leq T$, $T^* \leq S$ and $y \in \mathbb{R}^+$, we have the following expressions:

$$\begin{aligned}
\int_t^{T^*} B_y(u,S)\,du &= \frac{1}{y}\left(T^*-t\right) - \frac{1}{y}e^{-y(S-T^*)}B_y(t,T^*)\,,\\
\int_t^{T^*} B_y(u,S)^2\,du &= \frac{1}{y^2}\Big[(T^*-t) - 2e^{-y(S-T^*)}B_y(t,T^*)\\
&\qquad + e^{-2y(S-T^*)}B_{2y}(t,T^*)\Big]\,,\\
\int_t^{T^*} B_y(u,S)B_y(u,T)\,du &= \frac{1}{y^2}\Big[(T^*-t) - e^{-y(S-T^*)}B_y(t,T^*)\\
&\qquad - e^{-y(T-T^*)}B_y(t,T^*) + e^{-y(S+T)}e^{2y\,T^*}B_{2y}(t,T^*)\Big]\,.
\end{aligned}$$

The combined integrals of $B_y(t,T)$, $\sigma_\mu(t)$ and $\sigma_\lambda(t)$ are:

$$\begin{aligned}
\int_t^{T^*} \sigma_z(u)B_y(u,S)du &= \frac{\alpha_z}{y}e^{\beta_z(x_z+T^*)}B_{\beta_z}(t,T^*)\\
&\qquad -\frac{\alpha_z}{y}e^{\beta_z(x_z+T^*)-y(S-T^*)}B_{\beta_z+y}(t,T^*)\,,\\
\int_t^{T^*} \sigma_z(u)B_y(u,S)B_{\kappa_z}(u,T)\,du &= \frac{\alpha_z e^{\beta_z(x_z+T^*)}}{y\kappa_z}\left(B_{\beta_z}(t,T^*)\right.\\
&\qquad -e^{-y(S-T^*)}B_{\beta_z+y}(t,T^*) - e^{-\kappa_z(T-T^*)}B_{\beta_z+\kappa_z}(t,T^*)\\
&\qquad \left.+e^{-y(S-T^*)-\kappa_z(T-T^*)}B_{y+\kappa_z+\beta_z}(t,T^*)\right)\,,\\
\int_t^{T^*} \sigma_z(u)^2B_{\kappa_z}(u,S)B_{\kappa_z}(u,T)\,du &= \left(\frac{\alpha_z e^{\beta_z(x_z+T^*)}}{\kappa_z}\right)^2\left(B_{2\beta_z}(t,T^*)\right.\\
&\qquad -e^{-\kappa_z(S-T^*)}B_{2\beta_z+\kappa_z}(t,T^*) - e^{-\kappa_z(T-T^*)}B_{2\beta_z+\kappa_z}(t,T^*)\\
&\qquad \left.+e^{-\kappa_z(S-T^*)-\kappa_z(T-T^*)}B_{2(\kappa_z+\beta_z)}(t,T^*)\right)\,,\\
\int_t^{T^*} \sigma_\mu(u)\sigma_\lambda(u)B_{\kappa_\mu}(u,S)B_{\kappa_\lambda}(u,T)\,du &= \frac{\alpha_\mu\alpha_\lambda e^{\beta_\mu(x_\mu+T^*)+\beta_\lambda(x_\lambda+T^*)}}{\kappa_\mu\kappa_\lambda}\\
&\qquad \Big(B_{\beta_\mu+\beta_\lambda}(t,T^*) - e^{-\kappa_\mu(S-T^*)}B_{\beta_\mu+\beta_\lambda+\kappa_\mu}(t,T^*)\\
&\qquad -e^{-\kappa_\lambda(T-T^*)}B_{\beta_\mu+\beta_\lambda+\kappa_\lambda}(t,T^*)\\
&\qquad +e^{-\kappa_\mu(S-T^*)-\kappa_\lambda(T-T^*)}B_{\kappa_\mu+\kappa_\lambda+\beta_\mu+\beta_\lambda}(t,T^*)\Big)\,.
\end{aligned}$$

Appendix C

This appendix, provides the full analytical expressions of the variance and mean of the lognormal random process, $\ln \frac{A_{T^*}/P(T^*,T^*)}{A_t/P(t,T^*)}$ (see Proposition 3). The variance $v_{\mathbb{F}}(t)$ is developed as the

following sum:

$$v_{\mathbb{F}}(t) = \int_t^{T^*} \sigma_r^2 B_{\kappa_r}(u,T^*)^2\, du + 2\boldsymbol{\pi}^\top \int_t^{T^*} \Sigma_A(u) B_{\kappa_r}(u,T^*)\, du\, \Sigma_r \sigma_r \qquad (44)$$
$$+\boldsymbol{\pi}^\top \int_t^{T^*} \Sigma_A(u)\Sigma_A(u)^\top\, du\, \boldsymbol{\pi}\,.$$

The integral in the second term of the right hand side is a 3×4 matrix equal to

$$\int_t^{T^*} \Sigma_A(u) B_{\kappa_r}(u,T^*)\, du = \qquad (45)$$
$$\begin{pmatrix} \sigma_S\Sigma_S^\top \int_t^{T^*} B_{\kappa_r}(u,T^*)\, du \\ -\sigma_r\Sigma_r^\top \int_t^{T^*} B_{\kappa_r}(u,T) B_{\kappa_r}(u,T^*)\, du \\ \begin{pmatrix} -\sigma_r\Sigma_r^\top \int_t^{T^*} B_{\kappa_r}(u,T) B_{\kappa_r}(u,T^*)\, du - \Sigma_\lambda^\top \\ \times \int_t^{T^*} \sigma_\lambda(u) B_{\kappa_r}(u,T^*) B_{\kappa_\lambda}(u,T) du \end{pmatrix} \end{pmatrix}.$$

The integral in the third term of the right hand side of Equation (3) is a symmetric 3×3 matrix:

$$\int_t^{T^*} \Sigma_A(u)\Sigma_A(u)^\top\, du = \begin{pmatrix} c_{11}(t) & c_{12}(t) & c_{13}(t) \\ c_{12}(t) & c_{22}(t) & c_{23}(t) \\ c_{13}(t) & c_{23}(t) & c_{33}(t) \end{pmatrix}, \qquad (46)$$

with the following entries:

$$\begin{aligned}
c_{11}(t) &= \sigma_S^2\,(T^*-t)\,, \\
c_{12}(t) &= -\sigma_S\Sigma_S^\top\Sigma_r\sigma_r \int_t^{T^*} B_{\kappa_r}(u,T)\, du\,, \\
c_{13}(t) &= -\sigma_S\Sigma_S^\top\Sigma_r\sigma_r \int_t^{T^*} B_{\kappa_r}(u,T) du - \sigma_S\Sigma_S^\top\Sigma_\lambda \int_t^{T^*} \sigma_\lambda(u) B_{\kappa_\lambda}(u,T) du\,, \\
c_{22}(t) &= \sigma_r^2 \int_t^{T^*} B_{\kappa_r}(u,T)^2 du\,, \\
c_{23}(t) &= \sigma_r^2 \int_t^{T^*} B_{\kappa_r}(u,T)^2 du + \sigma_r\Sigma_r^\top\Sigma_\lambda \int_t^{T^*} \sigma_\lambda(u) B_{\kappa_r}(u,T) B_{\kappa_\lambda}(u,T) du\,, \\
c_{33}(t) &= \sigma_r^2 \int_t^{T^*} B_{\kappa_r}(u,T)^2 du + \int_t^{T^*} \sigma_\lambda(u)^2 B_{\kappa_\lambda}(u,T)^2 du \\
&\quad +2\sigma_r\Sigma_r^\top\Sigma_\lambda \int_t^{T^*} \sigma_\lambda(u) B_{\kappa_r}(u,T) B_{\kappa_\lambda}(u,T) du\,.
\end{aligned}$$

The mean function, $\mu_{\mathbb{F}}(t)$, is developed as follows

$$\mu_{\mathbb{F}}(t) = -\frac{1}{2} v_{\mathbb{F}}(t) - \sigma_r\Sigma_r^\top\Sigma_\mu \int_t^{T^*} \sigma_\mu(u) B_{\kappa_r}(u,T^*) B_{\kappa_\mu}(u,T^*)\, du \qquad (47)$$
$$-\sigma_r\Sigma_r^\top\Sigma_\lambda \int_t^{T^*} \sigma_\lambda(u) B_{\kappa_r}(u,T^*) B_{\kappa_\lambda}(u,T^*) du$$
$$-\boldsymbol{\pi}^\top \int_t^{T^*} \Sigma_A(u)\sigma_\mu(u) B_{\kappa_\mu}(u,T^*) du\, \Sigma_\mu$$
$$-\boldsymbol{\pi}^\top \int_t^{T^*} \Sigma_A(u)\sigma_\lambda(u) B_{\kappa_\lambda}(u,T^*) du\, \Sigma_\lambda\,.$$

The integrals in the third and fourth terms of Equation (47) are respectively equal to

$$\int_t^{T^*} \Sigma_A(u)\sigma_\mu(u)B_{\kappa_\mu}(u,T^*)du = \tag{48}$$
$$\begin{pmatrix} \sigma_S\Sigma_S^\top \int_t^{T^*} \sigma_\mu(u)B_{\kappa_\mu}(u,T^*)\,du \\ -\sigma_r\Sigma_r^\top \int_t^{T^*} \sigma_\mu(u)B_{\kappa_r}(u,T)B_{\kappa_\mu}(u,T^*)\,du \\ \begin{pmatrix} -\sigma_r\Sigma_r^\top \int_t^{T^*} \sigma_\mu(u)B_{\kappa_r}(u,T)B_{\kappa_\mu}(u,T^*)du - \Sigma_\lambda^\top \\ \times \int_t^{T^*} \sigma_\mu(u)\sigma_\lambda(u)B_{\kappa_\mu}(u,T^*)B_{\kappa_\lambda}(u,T)\,du \end{pmatrix} \end{pmatrix},$$

and

$$\int_t^{T^*} \Sigma_A(u)\sigma_\lambda(u)B_{\kappa_\lambda}(u,T^*)du = \tag{49}$$
$$\begin{pmatrix} \sigma_S\Sigma_S^\top \int_t^{T^*} \sigma_\lambda(u)B_{\kappa_\lambda}(u,T^*)\,du \\ -\sigma_r\Sigma_r^\top \int_t^{T^*} \sigma_\lambda(u)B_{\kappa_r}(u,T)B_{\kappa_\lambda}(u,T^*)\,du \\ \begin{pmatrix} -\sigma_r\Sigma_r^\top \int_t^{T^*} \sigma_\lambda(u)B_{\kappa_r}(u,T)B_{\kappa_\lambda}(u,T^*)du - \Sigma_\lambda^\top \\ \times \int_t^{T^*} \sigma_\lambda(u)^2B_{\kappa_\lambda}(u,T^*)B_{\kappa_\lambda}(u,T)\,du \end{pmatrix} \end{pmatrix}.$$

The integrals in Equations (45), (46), (48) and (49) are detailed in Appendix B.

Appendix D

In the numerical illustration, we model the initial yield curve with the Nelson-Siegel (NS) model. In this framework, initial instantaneous forward rates are provided by the following function:

$$f(0,t) := -\partial_t \ln P(0,t) \quad = \quad b_0^{(r)} + \left(b_{10}^{(r)} + b_{11}^{(r)}t\right)\exp\left(-c_1^{(r)}t\right).$$

Parameters $\{b_0, b_{10}, b_{11}, c_1\}$ are estimated by minimizing the quadratic spread between market and model zero-coupon yields:

$$P(0,t) \quad = \quad \exp\left(b_0^{(r)} + \frac{1}{t}\frac{b_{10}^{(r)}}{c_1^{(r)}}\left(1-e^{-c_1^{(r)}t}\right) + \frac{1}{t}\frac{b_{11}^{(r)}}{\left(c_1^{(r)}\right)^2}\left(1-\left(c_1^{(r)}t+1\right)e^{-c_1^{(r)}t}\right)\right).$$

We fit the NS model to the yield curve of Belgian state bonds observed on the first of September 2023 and obtain estimates reported in Table 12.

Parameter	Value
$b_0^{(r)}$	0.040535
$b_{10}^{(r)}$	0.003652
$b_{11}^{(r)}$	-0.022349
$c_1^{(r)}$	0.449268

Table 12: Nelson-Siegel parameters, Belgian state bonds, 1/9/23.

The volatility of mortality rates is fitted by least square minimization of spreads between $\sigma_x(.)$ and empirical deviations of variations of mortality rates by cohort (ages between 20 and 90 years from 1950 to 2020). If $\mu_x^{(y)}$ is the observed mortality rates at age x during the calendar year y, we denote by $\Delta\mu_x^{(y)} = \mu_x^{(y)} - \mu_{x-1}^{(y-1)}$ and by S_x the standard deviation of $\Delta\mu_x^{(y)}$ for y=1950 to 2020. The α_μ and β_μ are obtained by minimizing the sum

$$\alpha_\mu, \beta_\mu = \arg\min \sum_{x=20}^{90} \left(S_x - \alpha_\mu e^{\beta_\mu x}\right)^2.$$

On the other hand, the initial curve of survival probabilities is described by a Makeham's model, i.e.

$$\begin{aligned} {}_tp_x^{\mu} &= \exp - \int_x^{x+t} \left(a^{(\mu)} + b^{(\mu)} \left(c^{(\mu)} \right)^s \right) ds \\ &= \exp(-a^{(\mu)} t) \exp\left(-\frac{b^{(\mu)}}{\ln c^{(\mu)}} \left(\left(c^{(\mu)} \right)^{x+t} - \left(c^{(\mu)} \right)^x \right) \right) . \end{aligned}$$

where $a^{(\mu)}, b^{(\mu)}, c^{(\mu)} \in \mathbb{R}^+$. These parameters and the reversion speed κ_u are obtained by least square minimization of spreads between prospective and model survival probabilities. Prospective survival probabilities are computed with a Lee-Carter model fitted to Belgian mortality rates from 1950 to 2020 for 0 to 105 years, male population. Model ${}_tp_x$ are computed with Equation (25) for $x = 20$ years old. Estimated parameters are provided in Table 13.

Parameters			
$a^{(\mu)}$	1.006349e-03	κ_μ	0.83925
$b^{(\mu)}$	2.790903e-07	α_μ	8.5277e-7
$c^{(\mu)}$	1.152292	β_μ	0.110938
$a^{(\lambda)}$	1.05 $a^{(\mu)}$	κ_λ	0.9 κ_μ
$b^{(\lambda)}$	1.05 $b^{(\mu)}$	α_λ	0.9 α_μ
$c^{(\lambda)}$	1.05 $c^{(\mu)}$	β_λ	0.9 β_μ

Table 13: Mortality parameters, Belgian male mortality rates, year 2020.

Acknowledgement

The first author thanks the FNRS (Fonds de la recherche scientifique) for the financial support through the EOS project Asterisk (research grant 40007517).

References

[1] Al-Aradi A., Correia A., Jardim G., de Freitas Naiff D., Saporito Y. 2022. Extensions of the deep Galerkin method. Applied Mathematics and Computation.

[2] Barigou K., Delong L. 2022. Pricing equity-linked life insurance contracts with multiple risk factors by neural networks. Journal of Computational and Applied Mathematics, 404, 113922.

[3] Beck C., Weinan E., Jentzen A. 2019. Machine learning approximation algorithms for high dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. Journal of Nonlinear Science, 29, 1563-1519.

[4] Biagini F., Gonon L., Reitsam T., 2023. Neural network approximation for superhedging prices. Mathematical Finance 33 (1), 146-184.

[5] Buehler H., Gonon L., Teichmann J., Wood B., 2019. Deep hedging. Quantitative Finance 19 (8), 1-21.

[6] Cai Z., 2018. Approximating quantum many-body wave-functions using artificial neural networks. Physical Review B, 97, 035116.

[7] Cuomo S., Schiano Di Cola V., Giampaolo F., Rozza G., Raissi M., Piccialli F., 2022. Scientific machine learning through Physics-Informed Neural Networks: where we are and what's next. Journal of Scientific Computing 92:88.

[8] Carleo G., Troyer M. 2017. Solving the quantum many-body problem with artificial neural networks. Science 355 (6325), 602–606.
Delong L., Dhaene J., Barigou K., 2019. Fair valuation of insurance liability cash-flow streams in continuous time: Theory. Insurance: Mathematics and Economics 88, p 196-208.

[9] Denuit M., Trufin J., Hainaut D. 2019. Effective statistical learning for actuaries III. Neural networks and extensions. Springer Nature Switzerland

[10] Doyle D.,Groendyke C. ,2019. Using neural networks to price and hedge variable annuity guarantees. Risks 7(1), 1.

[11] Gatta F., Di Cola V. S., Giampaolo F., Piccialli F. Cuomo S. 2023. Meshless methods for American option pricing through Physics-Informed Neural Networks. Engineering Analysis with Boundary Elements 151, 68-82.

[12] Germain M., Pham H., Warin X., 2021. Neural networks-based algorithms for stochastic control and PDE's in finance. Machine Learning And Data Sciences For Financial Markets: A Guide To Contemporary Practices. Edited by Agostino Capponi and Charles-Albert Lehalle, Cambridge University Press.

[13] Glau K., Wunderlich L., 2022. The deep parametric PDE method and applications to option pricing. Applied Mathematics and Computation. 432, 127355.

[14] Hainaut D., Akbaraly A. 2023. Risk management with local least-square Monte-Carlo. ASTIN Bulletin 53 (3), 489 - 514.

[15] Hejazi S.A., Jackson K.R. 2016. A neural network approach to efficient valuation of large portfolios of variable annuities, Insurance: Mathematics and Economics 70, p 169–181.

[16] Horvath B., Teichmann J., Zuric Z., 2021. Deep hedging under rough volatility. Risk 9 (7), 138.

[17] Jiang Q., Sirignano J., Cohen S., 2023. Global convergence of Deep Galerkin and PINNs method for solving PDE. arXiv preprint.

[18] Lee H., Kang I.S., 1990. Neural algorithm for solving differential equations. J. Comput. Phys. 91(1), 110-131.

[19] Raissi M., Perdikaris P., Karniadakis G.E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput. Phys. 378, 686–707.

[20] Sirignano J., Spiliopoulos K., 2018. DGM: a deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375, 1339–1364.

[21] van Merriënboer B., Breuleux O., Bergeron A. 2018. Automatic differentiation in ML: Where we are and where we should be going. 32th NeurIPS proceedings. Advances in Neural Information Processing Systems, 31, 1-11.

[22] Weinan E., Han J., Jentzen A., 2017. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Commun. Math. Stat. 5(4), 349– 380.