PADOVA UNIVERSITY PRESS

Dolomites Research Notes on Approximation

Special Issue SA2022 – *Software for Approximation 2022*, Volume 15 · 2022 · Pages 13–24

MatLab Toolbox for the numerical solution of linear Volterra integral equations arising in metastatic tumor growth models*

Iulia Martina Bulai[c] · Maria Carmela De Bonis[a] · Concetta Laurita[a] · Valeria Sagaria[a]

Abstract

This paper introduces VIE Toolbox composed by fourteen MatLab functions used for the numerical resolution of Volterra Integral Equations (VIEs) of the second kind on infinite intervals. An application to metastatic tumor growth models is also considered, assuming five different tumor growth laws, e.g. exponential, power-law, Gompertz, generalized logistic and von Bertalanffy-West laws, for lung and breast tumors data.

1 Introduction

Here we propose a MatLab Toolbox for the numerical resolution of linear Volterra Integral Equations (VIEs) of the second kind on the positive semiaxis of the following type

$$F(t) - \int_0^t k(s,t)F(s)\,ds = G(t), \quad t \geq 0, \tag{1}$$

where the kernel $k(s,t)$ is a given function defined on $\Delta = \{(s,t)\,|\,t \geq 0,\, 0 \leq s \leq t\}$, $G(t)$ is a known function on $\mathbb{R}^+$ and $F(t)$ is the unknown solution. The software implements a method developed in the following two very recent papers [1, 2]. The use of the software is twofold. First of all, it can be employed for the numerical resolution of the general VIE introduced above. Secondly, it addresses the need to introduce alternative and efficient numerical methods for the resolution of Partial Differential Equation (PDE) models, describing the metastatic tumor growth, reformulated in terms of VIEs of the second kind [4]. In the last case the unknown of the VIEs is a biological observable such as the total metastatic mass and the cumulative number of metastases larger than a certain volume.

As a major contribution of this work, VIE toolbox aims at equipping researchers with a readily implemented tool to solve VIEs with minimal effort and high efficiency. Moreover, it can be easily expanded in order to be applied to more general PDE models that can be reformulated into VIEs.

The toolbox is licensed under the GNU General Public License v3.0 and is freely available through https://github.com/IuliaMartinaBulai/VIE_Toolbox, which aims at providing easy access and high visibility. The code is under active development, further MatLab functions will be added for the 2D case and generalizing the method for tumor growth laws for which no analytical expression of the solution is known. Moreover, the community can add functionalities or report issues within the git framework. This project shall foster communication and collaborations across scientific disciplines and, at the same time, increase the functionalities of VIE Toolbox further.

Details of the implemented numerical method are introduced in Section 2, while Section 3 presents its application for the efficient resolution of models describing the tumor growth and the metastatic spreading. In Section 4 the main MatLab functions forming the VIE Toolbox are described and in Section 5 through five demos we show some of the potentialities of the Toolbox.

2 Description of the implemented numerical method

The numerical method consists of the following steps. We first make the change of variable $s = te^{-z}$ in order to extend the integration interval to $[0, +\infty)$. In this way the Laguerre weight function naturally appears under the integral sign and the VIE (1) becomes

$$F(t) - t\int_0^{+\infty} k(te^{-z}, t)F(te^{-z})e^{-z}\,dz = G(t), \quad t \geq 0. \tag{2}$$

Then, we multiply both sides of equation (2) by the weight $u(t) = t^{\gamma}(1+t)^{\delta}e^{-\frac{t}{2}}$, $\gamma, \delta \geq 0$, and we approximate the integral using a truncated product quadrature rule based on Laguerre nodes along with an additional point. More precisely, the quadrature formula is obtained by replacing the function F appearing under the integral sign by the following truncated Lagrange polynomial (see, also, [1, 2]):

$$L^*_{n+1}(F,t) = \sum_{i=1}^{j} \ell_{n+1,i}(t)F(z_{n,i}), \tag{3}$$

[*] The preface of this special issue to which the article belongs is given in [3].

[a] Department of Mathematics, Computer Science and Economics, University of Basilicata, Via dell'Ateneo Lucano 10, 85100 Potenza, Italy

[c] Dipartimento di Scienze Chimiche, Fisiche, Matematiche e Naturali, University of Sassari, Via Vienna 2, 7100 Sassari, Italy

where $z_{n,i}$, $i=1,\dots,n$, are the zeros of the Laguerre polynomial of degree n, $p_n(w,x), w(x)=e^{-x}$,

$$j=\min_{i=1,\dots,n}\{i\ :\ z_{n,i}\geq 4\theta n\}, \tag{4}$$

with $0<\theta<1$ fixed, and

$$\ell_{n+1,i}(t)=l_{n,i}(t)\frac{4n-t}{4n-z_{n,i}},\quad i=1,\dots,j,$$

where $l_{n,i}(t)=\dfrac{p_n(w,t)}{p'_n(w,z_{n,i})(t-z_{n,i})}$ is the i–th fundamental Lagrange polynomial associated with the system of interpolation nodes $\{z_{n,i}\}_{i=1,\dots,n}$. We get the following new equation

$$(uF_n)(t)-u(t)\int_0^{+\infty} tk(te^{-z},t)L^*_{n+1}(F_n,te^{-z})e^{-z}dz=(uG)(t),$$

in the new unknown F_n, i.e.

$$(uF_n)(t)-u(t)\sum_{i=1}^{j}\frac{(uF_n)(z_{n,i})}{u(z_{n,i})}c_i(t)=(uG)(t),$$

where the integrals

$$c_i(t)=t\int_0^{+\infty} k(te^{-z},t)\ell_{n+1,i}(te^{-z})e^{-z}dz \tag{5}$$

are called modified moments. Since the analytical expressions of such integrals is not always available or their computation requires too much effort, we approximate them by a N-point truncated Gauss-Laguerre quadrature formula with $N\gg n$ obtaining

$$c_{i,N}(t)=t\sum_{\nu=1}^{J}\lambda_\nu k(te^{-z_{N,\nu}},t)\ell_{n+1,i}(te^{-z_{N,\nu}}), \tag{6}$$

where $z_{N,\nu}$ is the ν—th zero of the Laguerre polynomial $p_N(w)$, λ_ν is the corresponding Christoffel number and

$$J=\min_{\nu=1,\dots,N}\{\nu\ :\ z_{N,\nu}\geq 4\theta N\}, \tag{7}$$

with $0<\theta<1$ fixed. In this way the following fully discretized equation

$$(uF_{n,N})(t)-u(t)\sum_{i=1}^{j}\frac{(uF_{n,N})(z_{n,i})}{u(z_{n,i})}c_{i,N}(t)=(uG)(t), \tag{8}$$

in the unknown $F_{n,N}$ is obtained. Finally, collocating (8) at the Laguerre zeros $z_{n,i}$, $i=1,\dots,j$, we are led to solve the following linear system of order j

$$\sum_{i=1}^{j}\left[\delta_{r,i}-\frac{u(z_{n,r})}{u(z_{n,i})}c_{i,N}(z_{n,r})\right]a_i=(uG)(z_{n,r}),\quad r=1,\dots,j, \tag{9}$$

whose unknowns are $a_i=(uF_{n,N})(z_{n,i})$, $i=1,\dots,j$. Once the linear system is solved, the following approximation

$$F_{n,N}(t)=G(t)+\sum_{i=1}^{j}\frac{c_{i,N}(t)}{u(z_{n,i})}a_i \tag{10}$$

of the solution F of equation (1) is constructed. If $(a_1,\dots,a_j)$ is a solution of the system (9), then the so-called Nyström interpolating function $F_{n,N}$ is a solution of the equation (8) and vice versa.

We point out that we need to solve only one linear system whatever the number of evaluation points t is. Moreover, the computation of $c_{i,N}(z_{n,r})$, $i,r=1,\dots,j$, in the linear system (9) and of $c_{i,N}(t)$, $i=1,\dots,j$, in (10), has been implemented in VIE Toolbox (that will be described in Section 4) by dynamically choosing the truncation index J, according to the definition given in [2, (38)] which is equivalent to (7).

In [2, Theorems 2.1-2.3] sufficient conditions ensuring the unisolvence of the equation (1), the convergence of the just described method and the well conditioning of the linear systems (9) are given. Recalling the following function space definitions

$$C_u=\left\{F\in C((0,+\infty)) : \lim_{\substack{t\to 0\\ t\to+\infty}}(Fu)(t)=0\right\}$$

and

$$W_{r,u}=\{F\in C_u : F^{(r-1)}\in AC((0,+\infty)),\ \|F^{(r)}\varphi^r\|_u<+\infty\},\qquad r\geq 1,$$

where $\varphi(t)=\sqrt{t}$ and $AC(A)$ is the collection of all functions which are absolutely continuous on every closed subset $A\subseteq(0,+\infty)$, equipped with the following weighted norms

$$\|F\|_u=\sup_{t\geq 0}|(Fu)(t)|$$

and

$$\|F\|_{W_{r,u}}=\|F\|_u+\|F^{(r)}\varphi^r\|_u,$$

respectively, and setting $\log^+(t)=\log(\max\{1,t\}), t>0$, the results of Theorems 2.1-2.3 in [2] can be summarized as follows.

Theorem 2.1. *Let* $u(t) = (1+t)^{\delta} t^{\frac{1}{4}} e^{-\frac{t}{2}}, \delta \geq 0$, $w(t) = e^{-t}$ *and* $\varphi(t) = \sqrt{t}$. *Assuming that*

$$\sup_{t\geq 0} u(t) \left\| \frac{k_t}{u} \right\|_1 < +\infty,$$

$$\sup_{t\geq 0} u(t) \left\| \frac{k_t}{\sqrt{w\varphi}} \left(1 + \log^+ \left| \frac{k_t}{\sqrt{w\varphi}} \right| \right) \right\|_1 < +\infty,$$

$$\lim_{h\to 0} \sup_{t\geq 0} u(t) \left\| \frac{u(t+h)k_{t+h} - u(t)k_t}{\sqrt{w\varphi}} \left(1 + \log^+ \left| \frac{u(t+h)k_{t+h} - u(t)k_t}{\sqrt{w\varphi}} \right| \right) \right\|_1 = 0,$$

$$\sup_{0<t<1} u(t)\varphi(t) \left\| \frac{\varphi^i}{u} k_t^{(i)} \right\|_1 < +\infty, \quad i = 0, \dots, r, \quad r \geq 1,$$

$$\sup_{t\geq 1} u(t)\varphi^r(t) \left\| \frac{\varphi^i}{u} k_t^{(i)} \right\|_1 < +\infty, \quad i = 0, \dots, r, \quad r \geq 1,$$

and that $\mathrm{Ker}(I-K) = \{0\}$ *in* C_u, *then equation* (1) *admits a unique solution* $F \in C_u$, *for any right-hand side* $G \in C_u$.
Moreover, for all sufficiently large n *(say* $n \geq n_0$*) the coefficient matrix* $A_{n,N}$ *of linear system* (9) *is invertible and its condition number in the uniform norm satisfies*

$$\sup_{n\geq n_0} \mathrm{cond}(A_{n,N}) \leq \mathcal{C}, \qquad \mathcal{C} \neq \mathcal{C}(n). \tag{11}$$

If we further assume that both the exact solution F *and the right-hand side* G *of* (1) *belong to* $W_{r,u}$ *and that the method is stable, then the Nyström interpolant* $F_{n,N}$ *in* (10) *satisfies the following error estimate*

$$\|F - F_{n,N}\|_u \leq \mathcal{C} \max\left\{ \frac{1}{n^{\frac{r}{2}}}, \frac{\log n}{N^{\frac{r}{2}}} \right\}, \tag{12}$$

where $\mathcal{C}$ *is a positive constant independent of* n *and* N.

We remark that the above error estimate holds true assuming the stability of the method. Anyway, the numerical results (see Section 5) show that the condition numbers of the coefficient matrices of the solved linear systems do not grow for increasing dimensions supporting both theoretical result (11) and that the method is actually stable.

3 VIEs arising in metastatic tumor growth models

The above described method can be, in particular, applied for the numerical solution of the following special VIE studied in [2]

$$F(t) - \int_0^t k(s,t)F(s)\,ds = \int_0^t h(s,t)\,ds \quad t \geq 0, \tag{13}$$

where

$$k(s,t) = \beta_m(v_m(t-s)), \quad h(s,t) = \phi(v_m(t-s))\beta_p(v_p(s)),$$

which reformulates the following PDE differential model describing the tumor growth and the metastatic spreading

$$\begin{cases} \dfrac{\partial}{\partial t}\rho(v,t) + \dfrac{\partial}{\partial v}[g_m(v)\rho(v,t)] = 0, & v \in [v_{m,0}, b),\ t \geq 0, \\ g_m(v_{m,0})\rho(v_{m,0},t) = \beta_p(v_p(t)) + \displaystyle\int_{v_{m,0}}^b \beta_m(v)\rho(v,t)\,dv, & t \in (0,+\infty), \\ \rho(v,0) = 0, & v \in [v_{m,0}, b). \end{cases}$$

Here $v_p(t), v_m(t)$ represent the volume of the primary tumor at time t and of the metastases emitted at time t, respectively, $v_{m,0} = v_m(0)$, and $\beta_p(v), \beta_m(v)$ represent the rates with which the primary tumor and the metastases, respectively, of size v emit new metastases. According to the choice of the function $\phi(v)$ involved in the definition of $h(s,t)$, the unknown solution F of the VIE represents a different biological observable of interest. In particular

$$F(t) = \begin{cases} M(t) & \text{if } \phi(v) = v \\ N_{\bar{v}}(t) & \text{if } \phi(v) = \chi_{v\geq\bar{v}}(v) \end{cases}, \tag{14}$$

where $M(t)$ is the total metastatic burden at time t and $N_{\bar{v}}(t)$ is the cumulative number of metastases whose volume is larger than $\bar{v}$ ($\chi_{v\geq\bar{v}}$ denotes the characteristic function of the interval $[\bar{v}, +\infty)$).

The volumes $v_p(t)$ and $v_m(t)$ are solutions of the following Cauchy problems

$$\begin{cases} \dfrac{dv_p(t)}{dt} = g_p(v_p(t)), & t \geq 0 \\ v_p(0) = v_{p,0} \end{cases}$$

and

$$\begin{cases} \dfrac{dv_m(t)}{dt} = g_m(v_m(t)), \quad t \geq 0 \\ v_m(0) = v_{m,0} \end{cases},$$

where $v_{p,0}$ denotes the volume of the primary tumor at time $t = 0$, $g_p(v)$ is the growth rate of the primary tumor and $g_m(v)$ is the assumed growth rate of the new metastases.

Since in (13) the right-hand side is an integral, i.e.

$$G(t) = \int_0^t h(s,t)\,ds \tag{15}$$

we approximate it applying the same N-point truncated Gauss-Laguerre quadrature formula used for approximating the modified moments (5), obtaining

$$G_N(t) = t\sum_{\nu=1}^{J} \lambda_\nu h(te^{-z_{N,\nu}}, t), \tag{16}$$

where the index J has been detected in VIE Toolbox according to the definition given in [2, (39)] . Consequently, the linear system (9) and the Nyström interpolating function (10) take the following form

$$\sum_{i=1}^{j} \left[\delta_{r,i} - \frac{u(z_{n,r})}{u(z_{n,i})} c_{i,N}(z_{n,r}) \right] a_i = (uG_N)(z_{n,r}), \quad r = 1, \dots, j, \tag{17}$$

and

$$F_{n,N}(t) = G_N(t) + \sum_{i=1}^{j} \frac{c_{i,N}(t)}{u(z_{n,i})} a_i, \tag{18}$$

respectively.

We note that the assumption $G \in W_r(u)$, of Theorem 2.1, is fulfilled if the function $h(s,t)$ satisfies

$$\lim_{t\to+\infty} u(t) \int_0^t h(s,t)ds = 0,$$

$$\sup_{t\geq 0} u(t)\,\|h_t\|_{L^1([0,t])} < +\infty,$$

$$\sup_{0<t<1} u(t)\varphi(t)\,\|h_t\|_{W_r^1([0,t])} < +\infty,$$

and

$$\sup_{t\geq 1} u(t)\varphi^r(t)\,\|h_t\|_{W_r^1([0,t])} < +\infty,$$

where

$$W_r^1 = \left\{ f \in L^1 : \ f^{(r-1)} \in AC(0,+\infty), \ \|f^{(r)}\varphi^r\|_1 < +\infty \right\},$$

is the Sobolev subspace of L^1 of order $r \in \mathbb{R}^+$ equipped with the norm

$$\|f\|_{W_r^1} = \|f\|_1 + \|f^{(r)}\varphi^r\|_1.$$

4 VIE Toolbox

The VIE Toolbox we are going to propose is composed by fourteen novel MatLab functions, plus extra three recalled from existing toolboxes. In Figure 1 we have represented, starting from the top of the tree to the bottom, the core functions on the first level, e.g. `vie.m` and `tumorGrowth.m`, and the functions that are needed to run them on the underlying levels.

The goal of the two main functions `vie.m` and `tumorGrowth.m` of the introduced toolbox is, for the former one, to approximate the solutions of a VIE such as the one introduced in (1) and, for the second one, to compute the observables M and $N_{\bar{v}}$ defined in (14). For the sake of brevity, next we will report only the introductory part of the MatLab functions, where one can see information such as the name of the file, the goal of the function, how it should to be used, the input and output variables, and last, the functions that are recalled by the function itself. For more details about the core of the codes see https://github.com/IuliaMartinaBulai/VIE_Toolbox.

Below we start with the description of `vie.m`:

```
%--------------------------------------------------------------------------------------------
% File: vie.m
%
% Goal: Compute the solution of the following Volterra Integral Equation (VIE) of the second
%       kind on infinite intervals
%             t
%      f(t) - | K(s,t)f(s)ds=g(t),    t>0,
%             0
```

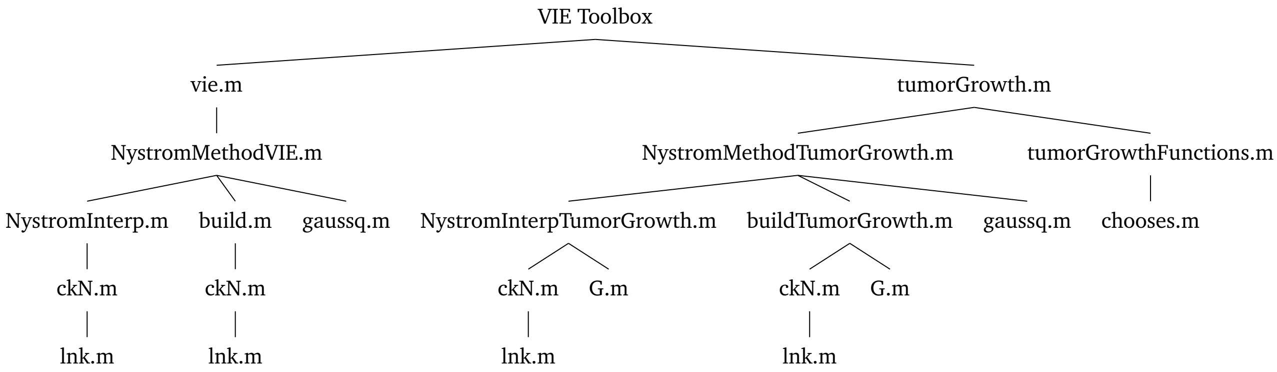

Figure 1: The MatLab functions of the toolbox VIE.

```
% and the corresponding weighted approximation error
%
% Use: [f,err,j,C] = vie(t,K,g)
%
% Input: t - row array of the evaluation points
%        K - kernel of the VIE
%        g - right-hand side of the VIE
%
% Output: f - solution of the VIE at t
%         err - weighted absolute error
%         j - size of the solved linear system
%         C - condition number of the solved linear system
%
% Recalls: NystromMethodVIE.m
%--------------------------------------------------------------------------------------------
```

The function `vie.m` recalls the function `NystromMethodVIE.m` implementing the method described in Section 2.

```
%--------------------------------------------------------------------------------------------
% File: NystromMethodVIE.m
%
% Goal: Compute the solution of the VIE using the Nystrom method in [2]
%
% Use: [f,j,C] = NystromMethodVIE(t,K,g,n,theta)
%
% Input:  t - row array of the evaluation points
%         K - kernel of the VIE
%         g - right-hand side of the VIE
%         n - number of knots
%         theta - truncation parameter
%
% Output: f - solution of the VIE
%         j - size of the solved linear system
%         C - condition number of the solved linear system
%
% Recalls: gaussq.m, build.m, NystromInterp.m
%--------------------------------------------------------------------------------------------
```

In order to run `NystromMethodVIE.m` a direct call to three different functions are needed, the first one is `NystromInterp.m`:

```
%--------------------------------------------------------------------------------------------
% File: NystromInterp.m
%
% Goal: Compute the Nystrom interpolant, solution of the VIE
%
% Use: [f] = NystromInterp(t,K,g,a,x,w1,E,U)
%
% Input: t - row array of the evaluation points
%        K - kernel of the VIE
%        g - right-hand side of the VIE
%        a - solution of the linear system
%        x - Laguerre zeros
%        w1 - weights of the N-point Gauss-Laguerre rule with N=2048
%        E - array exp(-x1) with x1 array of N=2048 Laguerre zeros
%        U - u(x) weight function u computed at the Laguerre zeros x
%
% Output: f - Nystrom interpolant, solution of the VIE
%
% Recalls: ckN.m
%---------------------------------------------------------------------------------------------
```

the second one is `build.m`:

```
%---------------------------------------------------------------------------------------------
% File: build.m
%
% Goal: Build the matrix and the right-hand side term of the linear system
%
% Use: [A,b] = build(K,g,j,x,w1,E,U)
%
% Input: K - kernel of the VIE
%        g - right-hand side of the VIE
%        j - size of the solved linear system
%        x - Laguerre zeros
%        w1 - weights of the N-point Gauss-Laguerre rule with N=2048
%        E - array exp(-x1) with x1 array of N=2048 Laguerre zeros
%        U - u(x) weight function u computed at the Laguerre zeros x
%
% Output: A - matrix of the linear system
%         b - right-hand side term of the linear system
%
% Recalls: ckN.m
%---------------------------------------------------------------------------------------------
```

and the last one is `gaussq.m`, already existing in Fortran environment (http://www.netlib.org/go/gaussq.f) and, then, translated in MatLab language. The last two functions to be recalled, corresponding to the last two levels of the tree in Figure 1, are `ckN.m`:

```
%---------------------------------------------------------------------------------------------
% File: ckN.m
%
% Goal: Approximate the integrals
%               +oo
%    c_k(t) = t | K(t exp(-z),t)l_{n+1,k}(t exp(-z)) exp(-z)dz,  k=1,...,j
%               0
%       using the N-point truncated Gauss-Laguerre quadrature rule
%
% Use: [c] = ckN(t,K,j,x,w1,Y,U)
%
% Input: t - row array of the evaluation points
%        K - kernel of the VIE
%        j - size of the solved linear system
%        x - Laguerre zeros
%        w1 - weights of the N-point Gauss-Laguerre rule
%        Y - matrix t.*exp(-x1)' with x1 array of N Laguerre zeros
%        U - u(x) weight function u computed at the Laguerre zeros x
%
% Output: c - matrix (c_k(t_i))_{i=1,...,length(t), k=1,...,j}
%
% Recalls: lnk.m
%---------------------------------------------------------------------------------------------
```

and `lnk.m`:

```
%---------------------------------------------------------------------------------------------
% File: lnk.m
%
% Goal: Compute the k-th fundamental Lagrange polynomial based on the
%       interpolation knots x at the point t
%
% Use: [l] = lnk(k,x,t)
%
% Input: k - index of the fundamental Lagrange polynomial
%        x - interpolation knots
%        t - evaluation point
%
% Output: l - k-th fundamental Lagrange polynomial at t
%-------------------------------------------------------------------------------------------
```

These last two functions together with gaussq.m are common to both branches of the tree in Figure 1. Let us now focus on the right hand side of Figure 1. As done for vie.m, we introduce tumorGrowth.m:

```
%-------------------------------------------------------------------------------------------
% File: tumorGrowth.m
%
% Goal: Compute the volume of the metastatic mass and the cumulative number of metastases as
%       solutions of Volterra Integral Equations
%
% Use: [M,err_rel_M,Nv,err_rel_Nv,j,C] = tumorGrowth(T,varargin{:})
%
% Input: T - final time point expressed in days
%        varargin - 1X16 cell containing 8 couples:
%                   'grow_p',value1 - growth law of the primary tumor
%                   'grow_m',value2 - growth law of the metastases
%                   'emission_p',mu_p - colonization coefficient of the primary tumor
%                   'emission_m',mu_m  - colonization coefficient of the metastases
%                   'v_p0', vp0 - volume of the primary tumor at time t=0
%                   'v_m0', vm0 - volume of the newly created metastases
%                   'Vbar', vbar - lower bound of the volume of the metastases whose
%                                  cumulative number Nv is computed
%                   'tumor_type', value3 - can assume values 'lung' and 'breast'
%
% Output: M - 1xT array of the approximate values of the metastatic mass at t=[1:T] days
%         err_rel_M - 1xT array of the relative errors related to the approximate values in M
%         Nv - 1xT array of the approximate values of the cumulative number of metastases
%              whose volume is larger than Vbar at t=[1:T] days
%         err_rel_Nv - 1xT array of the relative errors related to the approximate values in
%                      Nv
%         j - dimension of the solved linear system
%         C - condition number of the solved linear system
%
% Recalls: tumorGrowthFunctions.m, NystromMethodTumorGrowth.m
%-------------------------------------------------------------------------------------------
```

In contrast to vie.m, besides NystromMethodTumorGrowth.m, not reported here for the sake of brevity, tumorGrowth.m recalls also tumorGrowthFunctions.m:

```
%-------------------------------------------------------------------------------------------
% File: tumorGrowthFunctions.m
%
% Goal: Create a database of possible tumor growth laws both for primary and metastatic
%       growth, for lung and breast case studies
%
% Use: [s,gamma,g] = tumorGrowthFunctions(varargin{:})
%
% Input: varargin - 1X16 cell containing 8 couples:
%                   'grow_p',value1 - growth law of the primary tumor
%                   'grow_m',value2 - growth law of the metastases
%                   'emission_p',mu_p - colonization coefficient of the primary tumor
%                   'emission_m',mu_m - colonization coefficient of the metastases
%                   'v_p0', vp0 - volume of the primary tumor at time t=0
%                   'v_m0', vm0 - volume of the newly created metastases
%                   'Vbar', vbar - lower bound of the volume of the metastases whose
%                                  cumulative number Nv is computed
%                   'tumor_type',value3 - can assume values 'lung' and 'breast'
```

```
%
% Output: g - 1X4 cell with
%             g{1} integrand h(s,t) in the right-hand side of the VIE  for computing
%             the volume of the metastatic mass
%             g{2} integrand h(s,t) of the right-hand side of the VIE for computing
%             the cumulative number of metastases
%             g{3} kernel K(s,t) of the VIEs
%             g{4} weight function u(s) of the weighted space
%         gamma - upper limit of integration of the integral at the right-hand side of the
%                 VIE for computing the cumulative number of metastases
%         s - scale parameter of the time t
%
% Recalls: chooses.m
%--------------------------------------------------------------------------------------------
```

which contains a data-set of five different tumor growth laws, such as, exponential, power-law, Gompertz, generalized logistic and von Bertalanffy-West laws, for lung and breast tumors. If the input variable `varargin` is not provided, the following default values are assigned:

```
% default growth law for the primary tumor
value1 = 'gomp';
% default growth law for the metastases
value2 = 'gomp';
% default case study
value3 = 'breast';
% default parameter values
al = 2/3;
mu_p = 10^-3; % day^(-1)mm^(-3 al)
mu_m = 10^-3; % day^(-1)mm^(-3 al)
vp0 = 1; % mm^3
vm0 = 10^-6; % mm^3
vbar = 10^-6; % mm^3
```

Notice that the user can directly add in `tumorGrowthFunctions.m` different case studies by simply putting a new branch in the code

```
switch tumor_type
    case 'lung'
        switch grow_p
            case 'gomp'
            .
            .
            .
        end
        switch grow_m
            case 'gomp'
            .
            .
            .
       end
end
```

and substituting `lung` and `gomp` with opportune tumor type and growth laws, respectively.
The function `tumorGrowthFunctions.m` recalls the function `chooses.m` which returns the scale parameter `s` of the time. As explained in [1, p. 484] [2, p. 12], a suitable choice of `s` improves the performance of the method for large numbers of days. However, finding a criterion for making the best choice of `s` is still an open problem. In function `chooses.m` we report the values we have found in order to have, for the considered case studies, approximations of the observables in a time interval of 400 days, with an error less than $0.5e-6$. We point out that the given values might not be adequate when two different tumor growth laws, for primary and secondary tumors, are combined or when other case studies (different from the ones implemented here) are considered. Nevertheless, the user could insert in the code `chooses.m` his own choices.

Given the symmetry between the two branches of (1) we will not report here the functions that have been generalized starting from the left-hand side of the tree. More details can be found at https://github.com/IuliaMartinaBulai/VIE_Toolbox. The last function we introduce is `G.m`, recalled by both `NystromInterpTumorGrowth.m` and `buildTumorGrowth.m`

```
%--------------------------------------------------------------------------------------------
% File: G.m
%
% Goal: Approximate the integral
%              t+gamma
%        G(t) = | h(s,t)ds,
%                0
%        using the N-point truncated Gauss-Laguerre quadrature rule
%
% Use: [gt] = G(t,h,w,Y,gamma)
%
% Input: t - row array of the evaluation times expressed in days
%         h - integrand of the function G(t)
%         w - weights of the classical Gauss-Laguerre rule
%         Y - matrix ((t(i)+gamma)Exp(-x1(k))_{k=1,...,N, i=1,...,length(t)}
%             with x1 array of N Laguerre zeros
%         gamma - real number. Its value is 0 in the computation of the total metastatic
%                 mass and of the total number of metasteses (vbar=10^-6)
%
% Output: gt - approximation of G(t)
%-------------------------------------------------------------------------------------
```

In Table 1 we summarize (in alphabetic order) the .m files of VIE toolbox, giving their names, input and output variables. In order to give some further details about the mode of operation of the toolbox, five demo files, vie_demoi.m, i $= 1,\ldots,5$, are provided in addition to the .m files in Table 1. The numerical results of some of such demo files are reported in the next section.

MatLab Function	Input	Output
build.m	(K,g,j,x,w1,E,U)	[A,b]
buildTumorGrowth.m	(K,g1,g2,j,x,w1,E,U,gamma)	[A,b1,b2]
chooses.m	(varargin{:})	[s]
ckN.m	(t,K,j,x,w1,Y,U)	[c]
G.m	(t,h,w,Y,gamma)	[gt]
lnk.m	(k,x,t)	[l]
NystromInterp.m	(t,K,g,a,x,w1,E,U)	[f]
NystromInterpTumorGrowth.m	(t,K,g1,g2,a1,a2,x,w1,E,U,gamma)	[M,Nv]
NystromMethodTumorGrowth.m	(t,g,n,gamma,theta)	[M,Nv,j,C]
NystromMethodVIE.m	(t,K,g,n,theta)	[f,j,C]
tumorGrowth.m	(T,varargin{:})	[M,err_rel_M,Nv,err_rel_Nv,j,C]
tumorGrowthFunctions.m	(varargin{:})	[s,gamma,g]
vie.m	(t,K,g)	[f,err,j,C]

Table 1: The MatLab functions of VIE Toolbox

5 Demos

In this section we show the potentialities of the toolbox reporting the outputs of the execution of the codes vie_demoi.m, i $= 1,2,3$. In particular, the given approximation errors have been computed as follows

$$\mathtt{err} = e_{n,N}(F) := \begin{cases} \max\limits_{t\in\{1,\ldots,T\}} \dfrac{|F_{512,N}(t) - F_{256,N}(t)|}{|F_{512,N}(t)|}, & \texttt{vie\_demoi.m},\ \texttt{i} = 1,2, \\ \max\limits_{t\in\{1,\ldots,T\}} |(uF_{512,N})(t) - (uF_{256,N})(t)|, & \texttt{vie\_demo3.m}, \end{cases}$$

with $N = 2048$. Notice that the demo folder in VIE Toolbox contains two further demo files, vie_demoi.m, i $= 4,5$. Using the data-set provided in the Data subfolder of demo, vie_demo4.m and vie_demo5.m reproduce Figures 1 and 2 in [2], respectively. For the simulations we have used MatLab R2021b Update 2 64-bit (Win64) on a personal computer with the following characteristics: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 1.69 GHz RAM 8.00 GB.

5.1 vie_demo1.m

This file contains the code to compute the observables (14) using the default values for the variable varargin. Then, the volume of the metastatic mass and the total number of metastases are provided in the case of a breast tumor for which both the primary tumor and the newly created metastases grow according to the Gompertz growth law gomp. Setting T=60, the outputs returned by the function are the following:

```
Welcome to VIE demo #1
Solve the VIE for default values
Case study: breast
```

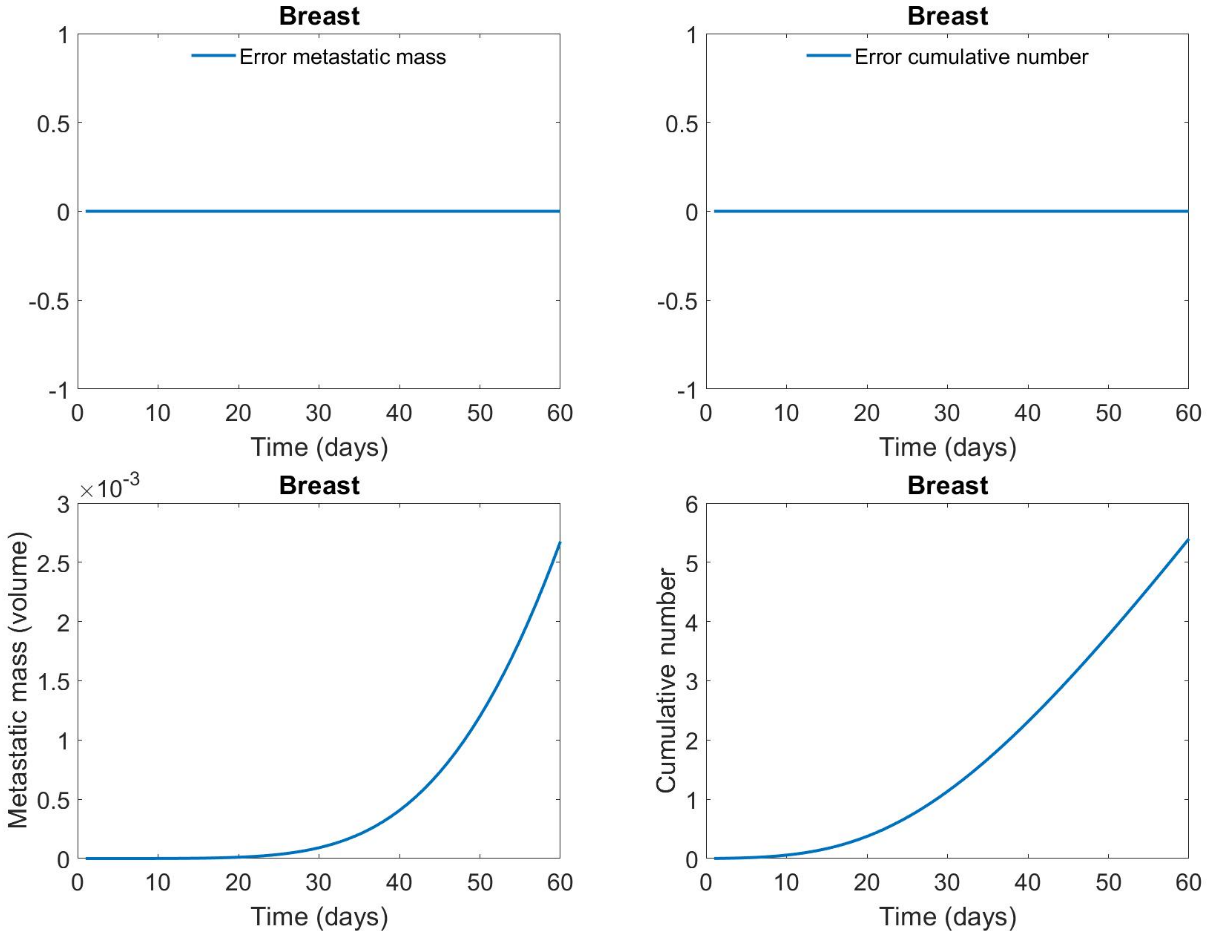

Figure 2: Relative errors and biological observable of `vie_demo1.m`

```
Index j and condition number C

ans =

1.560000000000000e+02     1.000297817015998e+00

Error for metastatic mass and for cumulative number of metastases

ans =

0     0

Plot the error for the metastatic mass
Plot the error for the cumulative number of metastases
Plot metastatic mass
Plot cumulative number of metastases
Elapsed time is 106.878419 seconds.
```

In Figure 2 the corresponding graphs are reported.

5.2 vie_demo2.m

The use of the toolbox for computing the observables (14) in the case where the growths of the primary tumor and of the metastases are described by different laws is presented in this file. In particular, we consider the case of a `lung` tumor and use the generalized logistic law, `gen_log`, for the primary tumor and the Gompertz law, `gomp`, for the new metastases. The results obtained in a time interval of 60 days (T=60) are the following:

```
Welcome to VIE demo #2
Solve the VIE
```

```
Generalized logistic growth law for the primary tumor
Gompertz growth law for the metastases
Case study: lung
Index j and condition number

ans =

1.560000000000000e+02     1.000836446552508e+00

Error for metastatic mass and for cumulative number of metastases

ans =

1.495323375912212e-13     1.944550558374719e-13

Plot the error for the metastatic mass
Plot the error for the cumulative number of metastases
Plot metastatic mass
Plot cumulative number of metastases
Elapsed time is 109.633700 seconds.
```

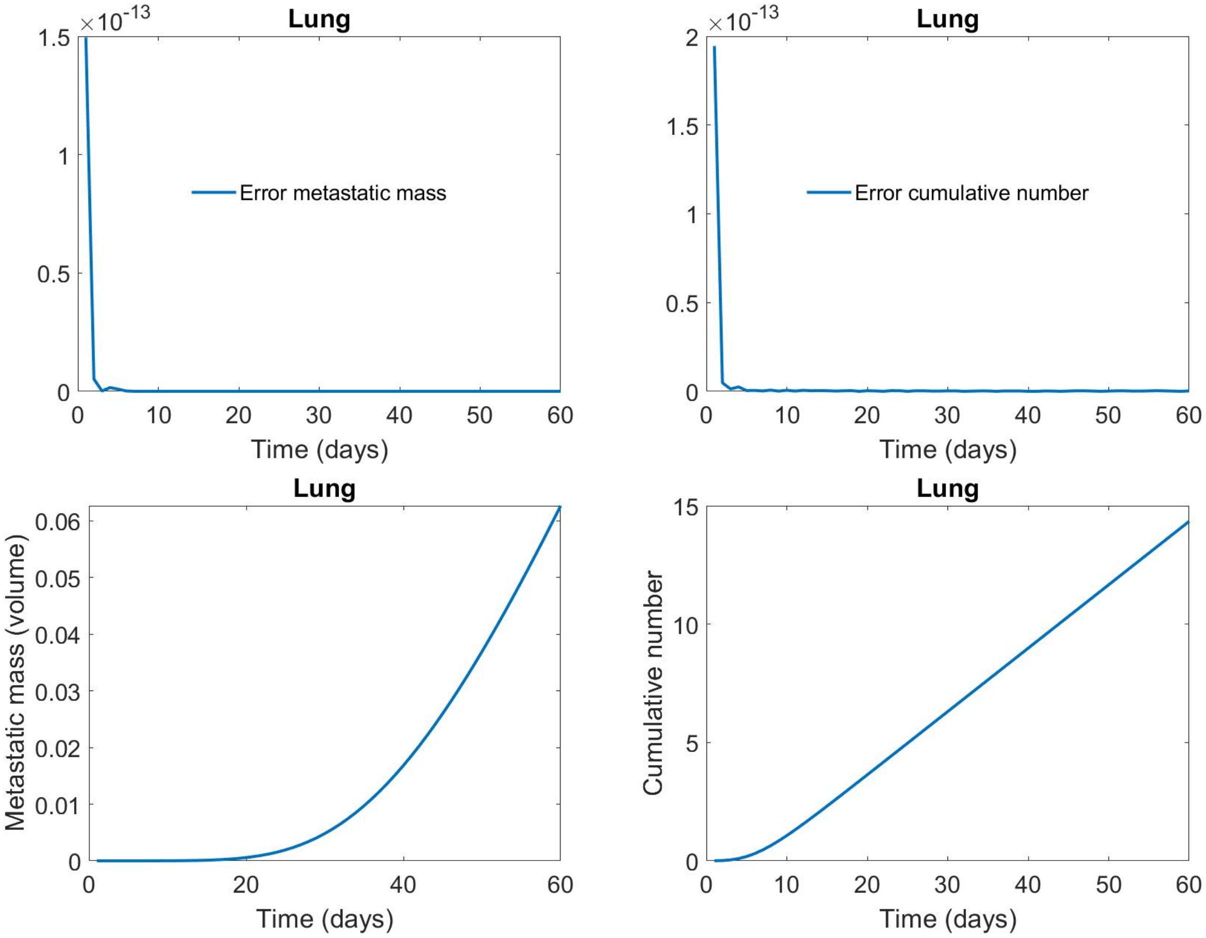

Figure 3: Relative errors and biological observable of `vie_demo2.m`

and the corresponding graphs are showed in Figure 3.

5.3 vie_demo3.m

Finally, this demo file is dedicated to the resolution of the following Volterra integral equation

$$f(t) - \int_0^t \frac{s}{s^2+t^2+1} f(s)\, ds = \sqrt{1+t^2}\arctan\frac{t}{\sqrt{1+t^2}}, \tag{19}$$

whose exact solution is the function $f(t) = t$. The following outputs are returned:

```
Welcome to VIE demo #3
Solve the VIE
Index j, condition number and error

ans =

1.560000000000000e+02     1.644523602637431e+00     4.440892098500626e-16

Plot the Nystrom interpolating function
Elapsed time is 84.802982 seconds.
```

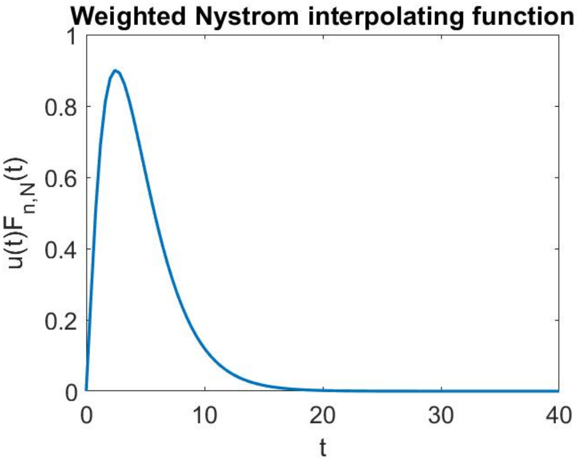

Figure 4: Weighted solution of equation (19).

In Figure 4 the graph of the Nyström interpolating function (10) in the interval $[0, 40]$ is reported.

6 Conclusions

In this paper, we have introduced a MatLab Toolbox, called VIE (Volterra Integral Equation), whose main purpose is to provide MatLab functions implemented to solve numerically linear Volterra integral equations of the second kind on the positive semiaxis.

Besides the use for the numerical solution of general VIEs, the toolbox has been designed as a useful tool to compute biological observables, such as the total metastatic mass and the cumulative number of metastasis, in the study of the evolution of metastatic tumors. These quantities are the unknowns of VIEs obtained from the reformulation of a generalized metastatic tumor growth model describing the primary tumor growth by means of an ordinary differential equation and the evolution of the metastatic density using a transport PDE.

One of the advantages of this toolbox is its versatility, in fact, on one side, one could assume different growth laws for the primary and secondary tumor growth. On the other side, also the emission rate of the metastases of the primary tumor may be different than the emission rate of the metastases themselves. As a second advantage, one could easily enrich the data-set of the case studies by adding tumor types and growth laws of personal interest.

Acknowledgements IMB has been supported by MUR through the grant PON-AIM Linea 1 (AIM1852570-1). MCDB and CL have been supported by University of Basilicata (local funds). MCDB, CL and VL have been supported by GNCS Project 2022 “Metodi e software per la modellistica integrale multivariata”. This research has been accomplished within RITA (Research ITalian network on Approximation) and the UMI Group TAA (Approximation Theory and Applications).

References

[1] M.C. De Bonis, C. Laurita, V. Sagaria. A numerical method for linear Volterra integral equations on infinite intervals and its application to the resolution of metastatic tumor growth. *Appl. Num. Math.* 172:475-496, 2022.

[2] I.M. Bulai, M.C. De Bonis, C. Laurita, V. Sagaria. Modeling metastatic tumor evolution, numerical resolution and growth prediction. *Mathematics and Computers in Simulation*, 2022.

[3] R. Cavoretto, A. De Rossi. Software for Approximation 2022 (SA2022). *Dolomites Res. Notes Approx.*, Special Issue SA2022, 15:i–ii, 2022.

[4] N. Hartung. Efficient resolution of metastatic tumor growth models by reformulation into integral equations. *Discrete and Continuous Dynamical Systems - Series B*, 20(2):445–467, 2015.