A numerical method for a nonlinear spatial population model with a continuous delay

Jörg Frochte

Hochschule Bochum, Höseler Platz 2, 42579 Heiligenhaus, Germany, joerg.frochte@hs-bochum.de

Abstract. A numerical method for a time-dependent nonlinear partial integro-differential equation (PIDE) is considered. This PIDE describes a spatial population model that includes a given carrying capacity and the memory effect of this environment. To deal with this issue an adaptive method of third order in time is considered to save storage data in smooth parts of the solution. Beyond this, a post-processing step adaptively thins out the history data.

Keywords: finite element method, partial integro differential equations, adaptivity in time, higher order, population dynamics, delay
PACS: 02.30.Ks, 02.60.Cb, 2.60Nm, 2.70Dh

INTRODUCTION

We will consider numerical methods for equations with memory or delay. Such equations can, of course, arise as ordinary or partial differential equations. One of the major problems is the data storage during the simulation because the total history data is needed to compute a new time step. So the practical experience with the ODE type is considerably higher and includes a wide range of topics (see e.g. [1], [2]). In this paper we will consider a numerical method for a nonlinear time dependent partial integro-differential equation (PIDE). The existence and uniqueness of the solution in L_1 of the population model was proved by Ruess in [3].

$$\begin{aligned} \frac{du}{dt} - \varepsilon \cdot \nabla^2 u(x,t) &= f + r \cdot u(x,t)(1 - \frac{1}{k} \cdot u(x,t) - \int_0^t K(s) \cdot u(x,s)\, ds) \\ u &= 0 \quad \text{on } \partial\Omega \times (0,T], u = u_0 \quad \text{in } \Omega, \text{for } t = 0 \end{aligned} \tag{1}$$

Let $T > 0$ and $\Omega_T = \Omega \times (0,T]$ where Ω is an open bounded region in $\mathbb{R}^n$. u is the density of a population. $r > 0$ defines the growth rate and $k > 0$ is the carrying capacity of the environment. The integral on the right represents a memory effect. It includes a continuous part or maybe the whole history weighted by the kernel function K.
In order to solve a sequence of linear initial value problems

$$\begin{aligned} \frac{d}{dt}u + Lu &= f + \int_0^t K(s-t)u(s)\, ds \quad \text{in } \Omega_T, \\ u &= \hat{u} \quad \text{on } \partial\Omega \times (0,T], u = u_0 \quad \text{in } \Omega, \text{for } t = 0\, , \end{aligned} \tag{2}$$

instead of the nonlinear one, we first of all have to linearize the PIDE.
One of the earliest publications concerning this kind of equations was published by Sloan and Thomée [4].

Up to the present the research in this area has mainly focused on linear and semi-linear cases (see e.g. [5], [6], [7], [8], [9]). We will use the higher order techniques for linear PIDE presented in [10] as ansatz for the non-linear task. It uses adaptivity in time and a post-processing technique to thin out the memory. This will help to face the problem of data storage during simulation. Beyond this, it is a higher order technique and we hope that the order three can be reached for the non-linear problem as well.

FIX POINT APPROACHES FOR THE NON-LINEAR TERM

We will use a kind of fixed point scheme for the linearization. This is a quite common ansatz which has also been applied to the nonlinear term of the Navier-Stokes-Equation (see e.g. [11]). To do this, we first turn to the semi-discrete case using a BDF(3) scheme:

$$\beta_0 u^{n+1} - \varepsilon\Delta t \nabla^2 u^{n+1} = \underbrace{\sum_{i=1}^{j} \beta_i u^{n+1-i}}_{:=f_2} + \Delta t f + r\Delta t \cdot u^{n+1}(1 - \frac{1}{K} \cdot u^{n+1} - \hat{I}_u)$$

$$\left[(\beta_0 - r\Delta t + \frac{r\Delta t}{K} \cdot u^{n+1})I - \varepsilon\Delta t \cdot \nabla^2\right] u^{n+1} = f_2 + \Delta t f - r\Delta t \cdot u^{n+1}\hat{I}_u$$

Now we will invert the operator on the left. This is possible as long as the stability condition $\beta_0 \geq r\Delta t(1 - \frac{1}{K} \cdot u^{n+1})$ is fulfilled. For Δt small enough it is always fulfilled, because all parameters are positive. So in fact this condition will only be violated for huge time steps. So, as a matter of fact, this condition will only be violated in the context of time steps.

$$u^{n+1} \quad = \quad \left[(\beta_0 - r\Delta t + \frac{r\Delta t}{K} \cdot u^{n+1})I - \varepsilon\Delta t \cdot \nabla^2\right]^{-1} \left(f_2 + \Delta t f - r\Delta t \cdot u^{n+1}\hat{I}_u\right) = \phi(u^{n+1})$$

Now we have got a fixed point problem: $u^{n+1} = \phi(u^{n+1})$ As usual, we now have to check whether ϕ is a contraction. It can be shown that for a sufficiently small Δt this is a contraction. Thus we can use this fixed point equation as an iteration technique. This fixed point approach is equivalent to substituting $r \cdot \overline{u}(x,t)$ for $r \cdot u(x,t)$ and solving the following equation:

$$\frac{du}{dt} - \varepsilon \cdot \nabla^2 u(x,t) + (\frac{r}{K}\overline{u}(x,t) - r) \cdot u(x,t) = f - r \cdot \overline{u}(x,t) \cdot \int_0^t K(s) \cdot u(x,s)\, ds \tag{3}$$

In the first iteration, $\overline{u}$ is an extrapolation of u based on the last time steps, whereas in the following iterations, $\overline{u}$ is the equal to the value of the last iteration. An alternative to approach I (3) is approach II (4):

$$\frac{du}{dt} - \varepsilon \cdot \nabla^2 u(x,t) + \frac{r}{K} \cdot \overline{u}(x,t)u(x,t) = f + r \cdot \overline{u}(x,t) - r \cdot \overline{u}(x,t) \cdot \int_0^t K(s) \cdot u(x,s)\, ds \tag{4}$$

Approach II in (4) is slightly closer to the original problem. In (3), however, less terms are linearized. Accordingly, one could expect a faster convergence. The argument that (4) is a contraction, is analogous to (3) and requires a sufficiently small Δt as well. In contrast to the formulation (3), we do not have to care about the stability condition, but as mentioned before, this is only a small advantage in the case of huge time step sizes.

NUMERICAL RESULTS

All numerical tests were computed using the FEM with linear base functions on triangles. For benchmark problem I

$$\frac{du}{dt} - \nabla^2 u(x,t) \quad = \quad f + 2 \cdot u(x,t)(1 - \cdot u(x,t) - \int_0^t \sin^2(s) \cdot u(x,s)\, ds) \tag{5}$$

$$u \quad = \quad u_0 \quad \text{in } \Omega, \text{for } t = 0 \tag{6}$$

we chose the Dirichlet boundary conditions and f in a way that $u = \left(\frac{3t}{16} + \frac{\sin(3\pi t)}{4}\right)(1 - (x-0.5)^2 - (y-0.5)^2)$ is the exact solution. The computation was performed over the time interval $[0,4]$. The fixed point iteration was stopped when the difference between two consecutive steps was smaller than $\varepsilon_{fix} = 10^{-6}$ in the maximum norm. In table 1 one can see that the third order in time could also be reached for this nonlinear problem. Thinking of increasing CPU-costs caused by reducing the time step size, one should keep the number of iterations per time step in mind. Each iteration means solving a linear PIDE. For $\Delta t = 1/8$ this was done 160 times, while for $\Delta t = 1/32$ it was 512 times. So the costs did not quadruple, the factor is just 3.2. Comparing the approaches I and II, we observe that with respect to accuracy both suggested fixed point approaches are almost identical. They do, however, differ in the number of fixed point iterations. One quite interesting fact is that the second approach in this example takes less iterations than the first approach.

TABLE 1. Nonlinear test case with fix point approaches I (left) and II (right) (33025 degrees of freedom in space)

Δt	$\|u-u_h\|_{L_2}^{\max}$	quotient	iterations per time step	Δt	$\|u-u_h\|_{L_2}^{\max}$	quotient	iterations per time step
1/8	2.682e-2	-	5.09	1/8	2.682e-2	-	4.68
1/16	4.336e-3	6.19	4.50	1/16	4.336e-3	6.19	4.33
1/32	5.557e-4	7.80	4.05	1/32	5.557e-4	7.80	3.81
1/64	6.921e-5	8.02	3.57	1/64	6.921e-5	8.02	3.41
1/128	8.981e-6	7.71	3.16	1/128	9.755e-6	7.09	3.06

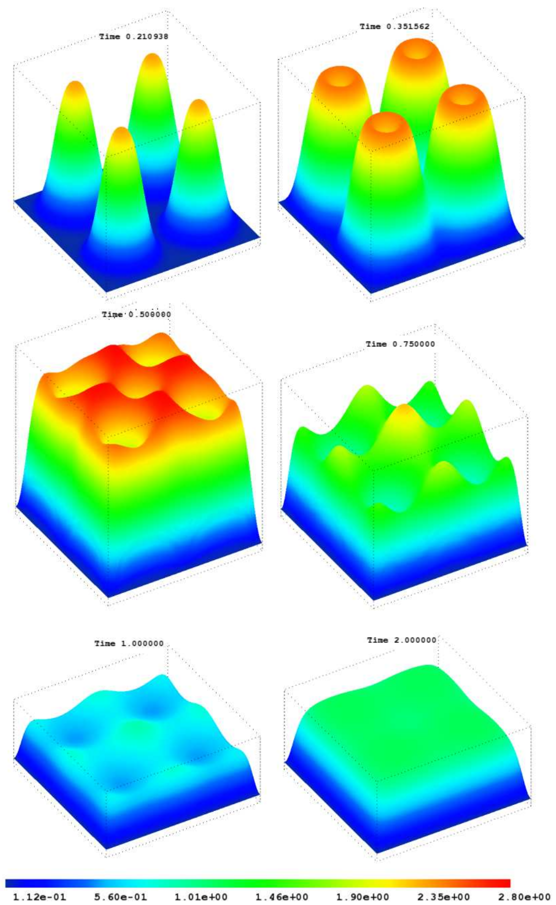
For the benchmark problem II the diffusion is set up with $\varepsilon = 10^{-2}$, the reproduction $r = 15$ and $K(s) = \exp(-s^2)$. On the right, it has no artificial function f and the exact solution is unknown.

$$\frac{du}{dt} - \frac{1}{100} \cdot \nabla^2 u(x,t) = 15 \cdot u(x,t)(1 - \frac{u(x,t)}{10} - \int_0^t K(s-t) \cdot u(x,s)\, ds) \tag{7}$$

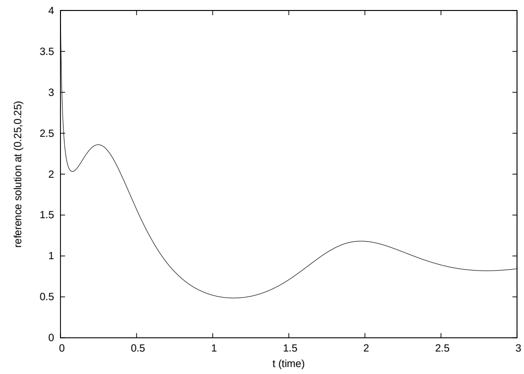
$$u = 0 \quad \text{on } \partial\Omega \times (0,3], ; \; u = u_0 \quad \text{in } \Omega, \text{for } t = 0 \tag{8}$$

As initial condition we set up four probe samples of a population on a kind of square island like this:

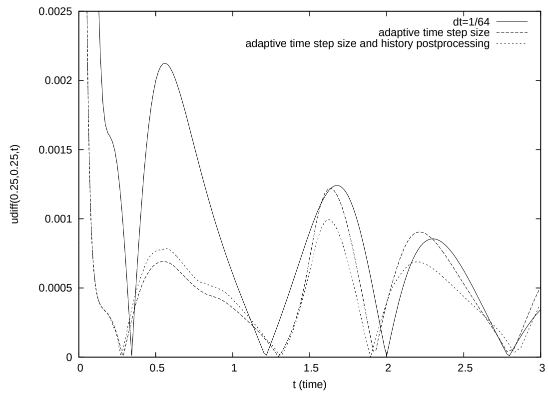
$$u_0(x,y) = u(x,y,0) = \frac{1}{\cosh(50(x-0.25))\cosh(50(y-0.25))} + \frac{1}{\cosh(50(x-0.25))\cosh(50(y-0.75))} + \frac{1}{\cosh(50(x-0.75))\cosh(50(y-0.25))} + \frac{1}{\cosh(50(x-0.75))\cosh(50(y-0.75))}$$



1: Density u at different times



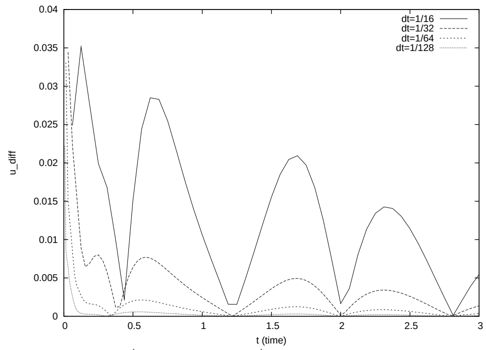
2: $u_{ref}(0.25, 0.25, t)$



3: $u_{diff}(0.25, 0.25, t)$ adaptive and fixed step size

TABLE 2. Results for different adaptive and non-adaptive techniques referring to a nonlinear test case in time (33025 degrees of freedom in space).

adaptivity in time	adaptive post-proc.	averaged Δt	averaged history size	quotient	No. of solved linear systems	quotient
no	no	0.015625	99.5	-	963 (5.015)	-
yes	no	0.012552	121	1.22	1150 (4.862)	1.19
yes	yes	0.012552	79.74	0.80	1150 (4.862)	1.19



4: $u_{diff}(0.25, 0.25, t)$ for different Δt

Every term represents one of four probes samples, and because $f = 0$ the population model should keep its symmetry during simulation. Figure 1 illustrates the behavior of the solution. The reference solution was computed with $\Delta t = 1/512$. To achieve a rating scale, the point (0.25,0.25), which is the peak of the probe sample in the lower left corner, was chosen. In figure 2, its development is displayed. At this point we will use the difference $u_{diff}(0.25, 0.25, t) = |u_{ref}(0.25, 0.25, t) - u_h(0.25, 0.25, t)|$ between the reference solution and the other approximations as an indicator of the achieved accuracy. For a fixed time step size, figure 4 reveals the difference. As expected, the error is high at the beginning and lower at the end of the simulation. The errors arising in the initial phase influence the simulation for some time and then their impact fades. Thus, in the beginning this problem requires small time steps, and later on it is possible to increase them. Using an adaptive technique with $Atol = 1e-5$ and $Rtol = 1e-4$, we can obtain the results displayed in figure 3. For comparison, the results for the fixed step size $\Delta t = 1/64$ were added to the figure. What might seem a bit odd at a first glance is that the difference for the thinned out history produced after about $t = 1.3$ is a bit smaller than the one that was computed without thinning. But this is just a comparison at one single point. It would presumably be the other way around for the L_2-norm. Table 2 gives an impression of the costs in the context of CPU and memory. In the sixth column the number of linear systems to be solved is denoted. The value in parenthesis is the average number per time step. It is not easy to compare the techniques in lines one and two because the adaptive approach is generally more exact, especially at the beginning, but it also requires about 20% more memory and linear systems to be solved. The full adaptive approach using the history post-processing still requires about 20% more linear systems but enables us to cut the memory usage for the saved history to 80 %. Hence it would allow a longer simulated time with a stable accuracy compared to a fixed time step approach.

CONCLUSION AND FUTURE PROSPECTS

An adaptive third order algorithm for a class of partial integro differential equations was applied to a class of nonlinear population dynamic models where higher order in time could be reached and techniques like the adaptive thinning out of the saved history works as well. A paper including more results and an analysis of the presented methods is in review. Due of the 'local' nature of the equation and its strong need for memory, distributed computation in style of domain decomposition methods should be considered in the near future.

REFERENCES

1. E. Buckwar, *Monte Carlo Methods and Applications* **10**, 235–244 (2004).
2. L. Göllmann, D. Kern, and H. Maurer, *Optimal Control Applications and Methods* **30**, 341–365 (2009).
3. W. Ruess, *Advances in Differential Equations* **4**, 843–876 (1999).
4. H. Sloan, and V. Thomée, *SIAM J. Numer. Anal.* **23** (1986).
5. V. Thomée, and N.-V. Zhang, *Mathematics of Computation* **53**, 121–139 (1989).
6. N.-Y. Zhang, *Mathematics of Computation* **60**, 133–166 (1993).
7. Y. Lin, V. Thomée, and L. B. Wahlbin, *SIAM J. Num. Ana.* **28**, 1047–1070 (1991).
8. A. K. Pani, V. Thomée, and L. B. Wahlbin, *Journal of Integral Equations and Applications* **4**, 231–252 (1992).
9. A. K. Pani, and T. E. Peterson, *SIAM Journal on Numerical Analysis* **33**, 1084–1105 (1996).
10. J. Frochte, *Numerical analysis and applied mathematics. AIP Conference Proceedings* **1048**, 213–216 (2008).
11. J. Frochte, and W. Heinrichs, *J. Comput. Appl. Math.* **228**, 373–390 (2009).