

Research Article

Solving Reaction-Diffusion and Advection Problems with Richardson Extrapolation

Tamás Mona,¹ István Lagzi,² and Ágnes Havasi³

¹Department of Meteorology, Eötvös Loránd University, Pázmány Péter sétány 1/A, Budapest 1117, Hungary

²Department of Physics, Budapest University of Technology and Economics, Budafoki út 8, Budapest 1111, Hungary

³Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University and MTA-ELTE Numerical Analysis and Large Networks Research Group, Pázmány Péter sétány 1/C, Budapest 1117, Hungary

Correspondence should be addressed to István Lagzi; istvanlagzi@gmail.com

Received 9 December 2014; Accepted 11 March 2015

Academic Editor: Henryk Kozłowski

Copyright © 2015 Tamás Mona et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Richardson extrapolation is a simple but powerful computational tool to enhance the accuracy of time integration methods. In the past years a few theoretical and partly practical works have been presented on this method. Detailed numerical applications of this method, however, are rarely found in the literature. Therefore, it is worth investigating whether this promising technique lives up to the expectations also in practice. In this paper we investigate the efficiency of the Richardson method in one-dimensional numerical (reaction-diffusion) problems.

1. Introduction

Phenomena occurring in nature and in laboratories can be understood and described by mathematical models. Most of those models are time-dependent ordinary or partial differential equations that cannot be solved analytically due to the highly nonlinear nature of the functions appearing in the equations. To overcome this problem, several numerical solving strategies have been developed in the past few decades, which can provide an accurate numerical solution to the original problem [1, 2].

Partial differential equations (PDEs), containing time (t) and space (x, y, z) as independent variables, can describe the spatiotemporal evolution of a set of physical quantities (physical system). This mathematical framework is used to understand and describe physical systems in various fields of science and technology. There are many processes (e.g., mass transport and pattern formation) which can be described and understood by reaction-diffusion systems. Generally, reaction-diffusion systems are mathematical models that describe the spatial and temporal variations of concentrations of chemical substances. From the mathematical point of view, the reaction-diffusion system is a set of parabolic PDEs [3].

The solution of PDEs should be accurate and the computational time should be small. There are several situations where the set of PDEs is used to make predictions such as in the case of weather prediction models, and thus the computational time should be dramatically reduced to use the solution in real-time applications. There are predominantly two strategies that can be used to increase the accuracy of the solution and reduce the computational time. First is developing sophisticated numerical schemes and methods that can increase the accuracy [4, 5]. This involves more computational effort. Second, computational time of mathematical models consisting of PDEs can be efficiently reduced, applying parallelization strategies on supercomputers, clusters, grids, and graphical processing units (video cards) [3, 6–10].

In this study, we show an efficient numerical method called Richardson extrapolation to increase the accuracy of the solution of various reaction-diffusion and advection systems. Four different problems are investigated and examined, including (i) simple diffusion of a chemically inert compound, (ii) diffusion with first- and second-order reaction of a chemical compound, (iii) advection process, and (iv) phase separation (pattern formation) in the wake of a moving diffusion front.

2. Richardson Extrapolation Methods

Richardson extrapolation is a powerful tool to increase the accuracy of any numerical method. It consists in applying the given numerical scheme with different discretization parameters (usually h and $h/2$) and combining the obtained numerical solutions by properly chosen weights. Namely, if p denotes the order of the selected numerical method, w_n denotes the numerical solution obtained by $h/2$, and z_n denotes that obtained by h , then the combined solution

$$y_n = \frac{2^p w_n - z_n}{2^p - 1} \quad (1)$$

has order $p + 1$. This method was first extensively used by Richardson, who called it “the deferred approach to the limit” [11]. The Richardson extrapolation is especially widely used for time integration schemes, when, as a rule, the results obtained by two different time step sizes are combined.

The Richardson extrapolation can be implemented in two different ways. During the passive Richardson extrapolation the combined solution is not used in the further computations, while in the active version it serves as an initial value for the next time step. Results concerning the stability and convergence of different numerical methods combined with the Richardson extrapolation can be found in [12–14]. In this paper, we investigate both the passive and the active Richardson methods in the selected reaction-diffusion and advection problems.

3. Model Applications

In the following sections we investigate the efficiency of the passive and active Richardson extrapolations on several one-dimensional model problems. First the problems were discretized in space using second-order finite differences and a small spatial step size. Then the obtained systems of time-dependent ordinary differential equations were solved by the forward Euler method, both with and without Richardson extrapolation. In this manner the errors resulting from the spatial discretization can be assumed to be much smaller than those arising from the time discretization. Since the forward Euler method is of first order, the Richardson extrapolation should result in a second-order time discretization method.

In all the simulations, each model problem was solved on the same time interval using different time step sizes. In each case we measured the running time, which can be considered as the computational time needed by the given method, since the optimal exploitation of the computer memory was not an issue in these simple models. The codes were written in C programming language. The running times were measured by the `clock()` function contained by the `time.h` header. All the computations were performed on the same Toshiba Satellite L750-In3 laptop by Ubuntu 12.04 LTS operational system, and the C compiler `g++` was used, which is easy to apply on a Linux terminal.

The model results were compared with a reference solution obtained by the forward Euler method with a very small time step size ($\Delta t = 10^{-7}$). The grid step size in all simulation was fixed ($\Delta x = 1$).

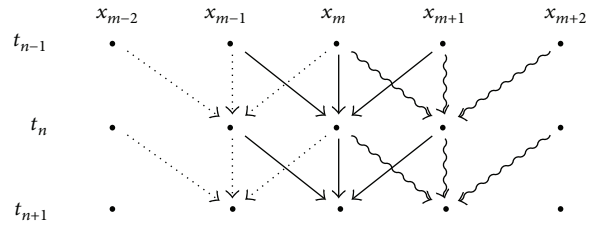


FIGURE 1: Discretization scheme of the diffusion problem.

4. Diffusion

Our basic model problem was the one-dimensional diffusion equation

$$\frac{\partial c}{\partial t} - D \frac{\partial^2 c}{\partial x^2} = 0, \quad x \in [0, 1000], \quad t \in [0, 1000] \quad (2)$$

equipped with Neumann boundary condition and an initial condition describing a peak (constant zero function with the exception of the middle point of the spatial domain where the value was 1). The value of the diffusion coefficient D was equal to 1 for simplicity. We generated a mesh with step sizes Δx and Δt on the space-time domain of the problem. Then, after a standard spatial discretization of the second-order spatial derivative, we applied the forward Euler method for time integration, which resulted in the formula

$$\begin{aligned} \tilde{c}(x_m, t_{n+1}) &= \tilde{c}(x_m, t_n) + D \\ &\cdot \frac{\tilde{c}(x_{m-1}, t_n) - 2 \cdot \tilde{c}(x_m, t_n) + \tilde{c}(x_{m+1}, t_n)}{\Delta x^2} \\ &\cdot \Delta t. \end{aligned} \quad (3)$$

Here $\tilde{c}(x_m, t_n)$ stands for the approximation of the exact solution c at the mesh point x_m and time layer t_n . This finite difference scheme is illustrated in Figure 1.

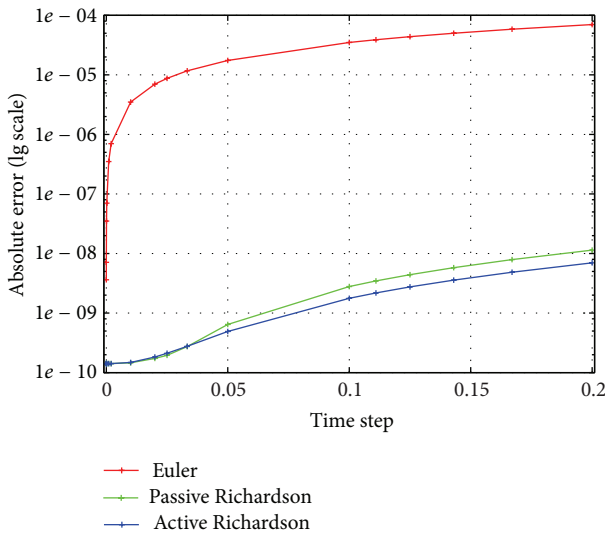
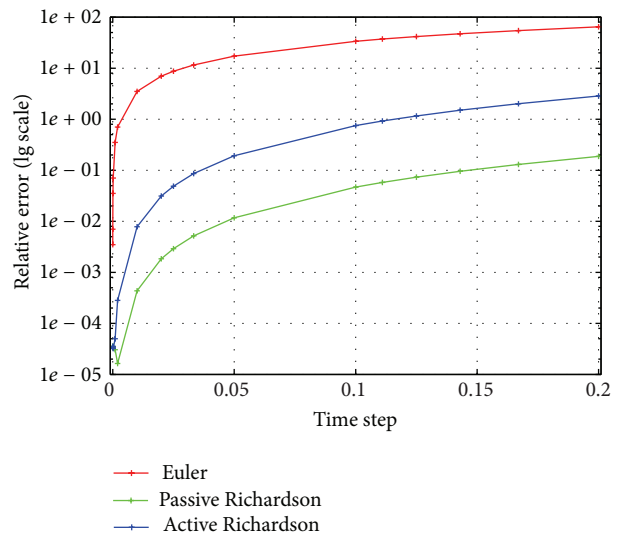
The numerical solutions obtained by Figure 1 with and without Richardson extrapolation were compared to the reference solution, and the absolute and relative errors were evaluated. Table 1 shows the absolute errors for decreasing time step sizes. Here and in the further tables the values given in parentheses in the i th row of the table show the estimated error order computed from the error obtained for the given time step (Δt_i) and the error obtained for the previous time step (Δt_{i-1}) according to the formula

$$p \approx \frac{\ln(\text{error}_{i-1}/\text{error}_i)}{\ln(\Delta t_{i-1}/\Delta t_i)}. \quad (4)$$

For a first-order method we should obtain values of p around 1, while for a second-order formula we should obtain values of p around 2. One can see that the forward Euler method behaves like a first-order method, while both Richardson extrapolations increased the order by one, according to the expectations. The higher order can be observed only for larger time steps, and then the decrease of the errors slows down. This can be explained by the fact that the accuracy of

TABLE 1: Absolute error of the diffusion problem without and with passive/active Richardson extrapolation (with the estimated convergence order in parentheses).

Time step	Absolute error					
	Euler		Passive Richardson		Active Richardson	
2.00×10^{-1}	7.00×10^{-5}	—	1.14×10^{-8}	—	6.98×10^{-9}	—
1.67×10^{-1}	5.84×10^{-5}	(1.000)	7.91×10^{-9}	(2.019)	4.86×10^{-9}	(1.991)
1.43×10^{-1}	5.00×10^{-5}	(1.000)	5.79×10^{-9}	(2.027)	3.58×10^{-9}	(1.987)
1.25×10^{-1}	4.38×10^{-5}	(1.000)	4.41×10^{-9}	(2.035)	2.75×10^{-9}	(1.978)
1.11×10^{-1}	3.89×10^{-5}	(1.000)	3.47×10^{-9}	(2.043)	2.18×10^{-9}	(1.971)
1.00×10^{-1}	3.50×10^{-5}	(1.000)	2.79×10^{-9}	(2.054)	1.77×10^{-9}	(1.961)
5.00×10^{-2}	1.75×10^{-5}	(1.000)	6.44×10^{-10}	(2.117)	4.91×10^{-10}	(1.852)
3.33×10^{-2}	1.17×10^{-5}	(1.000)	2.74×10^{-10}	(2.109)	2.77×10^{-10}	(1.413)
2.50×10^{-2}	8.75×10^{-6}	(1.000)	1.95×10^{-10}	(1.183)	2.12×10^{-10}	(0.934)
2.00×10^{-2}	7.00×10^{-6}	(1.000)	1.73×10^{-10}	(0.520)	1.83×10^{-10}	(0.650)
1.00×10^{-2}	3.50×10^{-6}	(1.000)	1.46×10^{-10}	(0.252)	1.48×10^{-10}	(0.308)
2.00×10^{-3}	7.00×10^{-6}	(1.000)	1.42×10^{-10}	(0.016)	1.42×10^{-10}	(0.024)
1.00×10^{-3}	3.50×10^{-7}	(1.000)	1.42×10^{-10}	(−0.004)	1.42×10^{-10}	(−0.001)
2.00×10^{-4}	7.01×10^{-8}	(0.999)	1.42×10^{-10}	(0.000)	1.42×10^{-10}	(0.000)
1.00×10^{-4}	3.51×10^{-8}	(0.997)	1.42×10^{-10}	(−0.001)	1.42×10^{-10}	(0.001)
2.00×10^{-5}	7.12×10^{-9}	(0.991)	1.42×10^{-10}	(0.000)	1.42×10^{-10}	(0.000)
1.00×10^{-5}	3.62×10^{-9}	(0.975)	1.42×10^{-10}	(0.002)	1.42×10^{-10}	(0.000)

FIGURE 2: Absolute error as a function of time step for the Euler method without and with passive and active Richardson extrapolation, by grid step $\Delta x = 1$ and coefficient $D = 1.0$. The reference solution was calculated using $\Delta t = 10^{-7}$.FIGURE 3: Relative error as a function of time step for the Euler method without and with passive and active Richardson extrapolation, by grid step $\Delta x = 1$ and coefficient $D = 1.0$. The reference solution was calculated using $\Delta t = 10^{-7}$.

the computation is limited by the accuracy of the reference solution, which is a numerical solution obtained by a very small time step and the same spatial grid size. This maximal accuracy is achieved by the Richardson extrapolation rather early, which shows the robustness of this method. Note that the accuracy that can be achieved by the pure forward Euler method by a time step size of 1.00×10^{-5} can be achieved by the passive Richardson method already by using a time step size of 1.25×10^{-1} and by the active Richardson method by a time step size of 1.43×10^{-1} . Figures 2 and 3 also show

that both Richardson methods are significantly more accurate than the forward Euler method in itself when the same time step is used. However, it is not worth reducing the time step size beyond all bounds.

Accuracy is not the only point of view that we should consider. It is also important that the results should be achieved within reasonable computational time. Table 2 shows the running times for the studied methods and the step lengths. One can see that the extrapolation methods roughly take twice as much time as the pure Euler method for the same

TABLE 2: Computational times of the diffusion problem without and with passive/active Richardson extrapolation.

Time step	Computational time (s)		
	Euler	Passive Richardson	Active Richardson
0.20	0.05	0.12	0.14
0.17	0.07	0.13	0.17
0.14	0.08	0.15	0.20
0.13	0.08	0.18	0.23
0.11	0.09	0.20	0.25
0.10	0.10	0.22	0.29
0.05	0.19	0.43	0.56
0.03	0.29	0.66	0.83
0.03	0.38	0.87	1.11
0.02	0.48	1.08	1.39
0.01	0.96	2.16	2.76
0.002	4.76	10.79	14.36
0.001	9.57	21.64	27.62
0.0002	47.95	106.92	138.50
0.0001	94.58	213.67	275.30
0.00002	474.86	1077.50	1381.50
0.00001	953.78	2153.56	2885.96

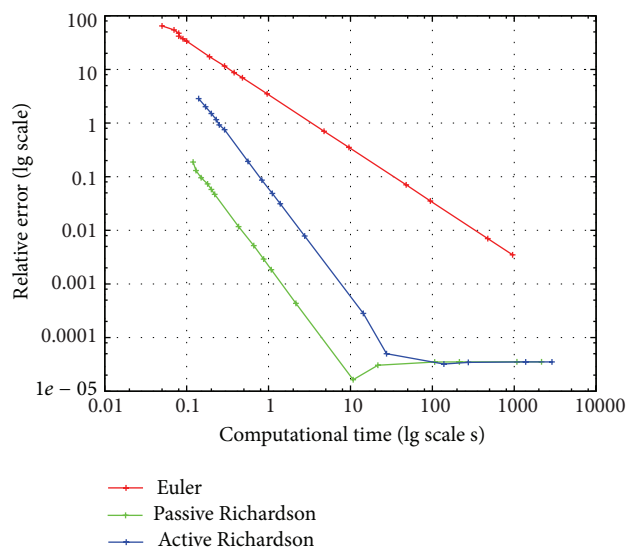


FIGURE 4: Relative error as a function of computational time for the Euler method without and with passive and active Richardson extrapolation, by grid step $\Delta x = 1$ and coefficient $D = 1.0$. The reference solution was calculated using $\Delta t = 10^{-7}$.

time step; however, as Figure 4 tells us, they are much more accurate.

For example, if a relative error of 0.7 is required, then the forward Euler method without Richardson extrapolation needs 4.76 s, with passive Richardson extrapolation 0.12 s, and with the active one 0.29 s, to achieve this goal. The latter method results in an almost fortyfold acceleration, while the previous one results in a 16-fold acceleration in

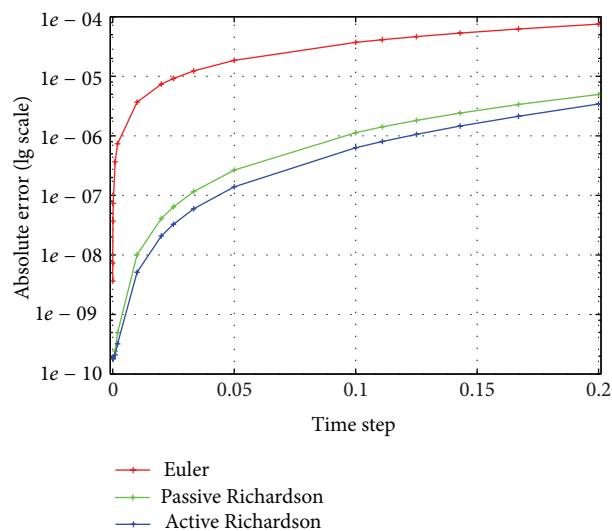


FIGURE 5: Absolute error as a function of time step in case of a quadratic reaction term for the Euler method without and with passive and active Richardson extrapolation, by grid step $\Delta x = 1$ and coefficient $D = 1.0$. The reference solution was calculated using $\Delta t = 10^{-7}$.

the computation. The passive extrapolation is the most efficient method from the three. Interestingly, it has a local minimum in the relative error (here the solution was obtained in 10.79 s). The result of this run is more accurate by more than two magnitudes than the most accurate Eulerian solution that we could obtain.

5. Diffusion with Linear or Quadratic Reaction Terms

For a more extensive analysis of the diffusion problem we supplemented the diffusion term with (i) a linear and (ii) a quadratic reaction term; that is, the following equations were considered:

$$\begin{aligned} \frac{\partial c}{\partial t} - D \frac{\partial^2 c}{\partial x^2} + k \cdot c &= 0, \quad x \in [0, 1000], \quad t \in [0, 1000], \\ \frac{\partial c}{\partial t} - D \frac{\partial^2 c}{\partial x^2} + k \cdot c^2 &= 0, \quad x \in [0, 1000], \quad t \in [0, 1000], \end{aligned} \quad (5)$$

where the reaction rate constant k was equal to 1. The diffusion coefficient, the meshes, and the supplementary conditions were the same as in the pure diffusion problem studied in the previous section.

The obtained absolute errors are given in Tables 3 and 4. In the case of the extra linear term, similar results have been obtained as in the pure diffusion problem; that is, the extrapolation methods significantly enhanced the efficiency. However, the addition of the quadratic term did cause some slight differences in the appearance of the error curves. Figure 5 shows no stationary section in the domain of the small time step sizes that were observed in the pure diffusion case. However, a thorough analysis reveals that it is not worth using extremely small time steps in this case, either.

TABLE 3: Absolute error of the diffusion problem with linear reaction term without and with passive/active Richardson extrapolation (with the estimated convergence order in parentheses).

Time step	Absolute error					
	Euler		Passive Richardson		Active Richardson	
2.00×10^{-1}	6.90×10^{-5}	—	1.14×10^{-8}	—	6.90×10^{-9}	—
1.67×10^{-1}	5.75×10^{-5}	(1.000)	7.89×10^{-9}	(2.012)	4.80×10^{-9}	(1.990)
1.43×10^{-1}	4.93×10^{-5}	(1.000)	5.78×10^{-9}	(2.016)	3.53×10^{-9}	(1.986)
1.25×10^{-1}	4.32×10^{-5}	(1.000)	4.42×10^{-9}	(2.021)	2.71×10^{-9}	(1.982)
1.11×10^{-1}	3.84×10^{-5}	(1.000)	3.48×10^{-9}	(2.027)	2.15×10^{-9}	(1.976)
1.00×10^{-1}	3.45×10^{-5}	(1.000)	2.81×10^{-9}	(2.034)	1.75×10^{-9}	(1.970)
5.00×10^{-2}	1.73×10^{-5}	(1.000)	6.63×10^{-10}	(2.083)	4.60×10^{-10}	(1.926)
3.33×10^{-2}	1.15×10^{-5}	(1.000)	2.66×10^{-10}	(2.247)	2.23×10^{-10}	(1.786)
2.50×10^{-2}	8.63×10^{-6}	(1.000)	1.34×10^{-10}	(2.402)	1.42×10^{-10}	(1.561)
2.00×10^{-2}	6.90×10^{-6}	(1.000)	7.88×10^{-11}	(2.364)	1.07×10^{-10}	(1.263)
1.00×10^{-2}	3.45×10^{-6}	(1.000)	4.68×10^{-11}	(0.751)	6.75×10^{-11}	(0.668)
2.00×10^{-3}	6.90×10^{-7}	(1.000)	6.24×10^{-11}	(−0.179)	6.33×10^{-11}	(0.040)
1.00×10^{-3}	3.45×10^{-7}	(1.000)	6.30×10^{-11}	(−0.015)	6.33×10^{-11}	(0.001)
2.00×10^{-4}	6.91×10^{-8}	(1.000)	6.33×10^{-11}	(−0.002)	6.32×10^{-11}	(0.000)
1.00×10^{-4}	3.46×10^{-8}	(0.999)	6.32×10^{-11}	(0.000)	6.33×10^{-11}	(0.000)
2.00×10^{-5}	6.95×10^{-9}	(0.997)	6.32×10^{-11}	(0.000)	6.32×10^{-11}	(0.000)
1.00×10^{-5}	3.50×10^{-9}	(0.990)	6.32×10^{-11}	(0.001)	6.32×10^{-11}	(0.001)

TABLE 4: Absolute error of the diffusion problem with quadratic reaction term without and with passive/active Richardson extrapolation (with the estimated convergence order in parentheses).

Time step	Absolute error					
	Euler		Passive Richardson		Active Richardson	
2.00×10^{-1}	7.51×10^{-5}	—	4.97×10^{-6}	—	3.43×10^{-6}	—
1.67×10^{-1}	6.23×10^{-5}	(1.027)	3.37×10^{-6}	(2.134)	2.13×10^{-6}	(2.596)
1.43×10^{-1}	5.32×10^{-5}	(1.019)	2.42×10^{-6}	(2.138)	1.46×10^{-6}	(2.469)
1.25×10^{-1}	4.65×10^{-5}	(1.015)	1.82×10^{-6}	(2.135)	1.06×10^{-6}	(2.388)
1.11×10^{-1}	4.13×10^{-5}	(1.012)	1.42×10^{-6}	(2.128)	8.06×10^{-7}	(2.330)
1.00×10^{-1}	3.71×10^{-5}	(1.011)	1.13×10^{-6}	(2.121)	6.33×10^{-7}	(2.288)
5.00×10^{-2}	1.85×10^{-5}	(1.007)	2.66×10^{-7}	(2.091)	1.39×10^{-7}	(2.185)
3.33×10^{-2}	1.23×10^{-5}	(1.003)	1.16×10^{-7}	(2.056)	5.95×10^{-8}	(2.098)
2.50×10^{-2}	9.21×10^{-6}	(1.002)	6.43×10^{-8}	(2.040)	3.28×10^{-8}	(2.065)
2.00×10^{-2}	7.37×10^{-6}	(1.002)	4.09×10^{-8}	(2.030)	2.08×10^{-8}	(2.047)
1.00×10^{-2}	3.68×10^{-6}	(1.001)	1.01×10^{-8}	(2.014)	5.13×10^{-9}	(2.019)
2.00×10^{-3}	7.36×10^{-7}	(1.000)	4.97×10^{-10}	(1.873)	3.23×10^{-10}	(1.719)
1.00×10^{-3}	3.68×10^{-7}	(1.000)	2.42×10^{-10}	(1.035)	2.10×10^{-10}	(0.618)
2.00×10^{-4}	7.36×10^{-8}	(1.000)	1.84×10^{-10}	(0.171)	1.83×10^{-10}	(0.086)
1.00×10^{-4}	3.68×10^{-8}	(1.001)	1.83×10^{-10}	(0.011)	1.83×10^{-10}	(0.006)
2.00×10^{-5}	7.32×10^{-9}	(1.002)	1.82×10^{-10}	(0.002)	1.83×10^{-10}	(0.000)
1.00×10^{-5}	3.64×10^{-9}	(1.007)	1.83×10^{-10}	(−0.007)	1.83×10^{-10}	(−0.005)

TABLE 5: Absolute error of the advection problem without and with passive/active Richardson extrapolation (with the estimated convergence order in parentheses).

Time step	Absolute error					
	Euler		Passive Richardson		Active Richardson	
2.00×10^{-1}	9.79×10^{-3}	—	1.01×10^{-5}	—	7.13×10^{-5}	—
1.67×10^{-1}	8.14×10^{-3}	(1.009)	7.01×10^{-6}	(2.000)	4.93×10^{-5}	(2.019)
1.43×10^{-1}	6.97×10^{-3}	(1.008)	5.15×10^{-6}	(2.000)	3.62×10^{-5}	(2.016)
1.25×10^{-1}	6.10×10^{-3}	(1.007)	3.94×10^{-6}	(2.000)	2.76×10^{-5}	(2.013)
1.11×10^{-1}	5.41×10^{-3}	(1.006)	3.11×10^{-6}	(2.000)	2.18×10^{-5}	(2.012)
1.00×10^{-1}	4.87×10^{-3}	(1.005)	2.52×10^{-6}	(2.000)	1.76×10^{-5}	(2.010)
5.00×10^{-2}	2.43×10^{-3}	(1.004)	6.31×10^{-7}	(2.000)	4.39×10^{-6}	(2.007)
3.33×10^{-2}	1.62×10^{-3}	(1.002)	2.80×10^{-7}	(1.999)	1.95×10^{-6}	(2.002)
2.50×10^{-2}	1.21×10^{-3}	(1.001)	1.58×10^{-7}	(1.998)	1.10×10^{-6}	(1.998)
2.00×10^{-2}	9.70×10^{-4}	(1.001)	1.01×10^{-7}	(1.996)	7.03×10^{-7}	(1.995)
1.00×10^{-2}	4.85×10^{-4}	(1.001)	2.59×10^{-8}	(1.965)	1.78×10^{-7}	(1.981)
2.00×10^{-3}	9.69×10^{-5}	(1.000)	4.99×10^{-9}	(1.023)	1.08×10^{-8}	(1.744)
1.00×10^{-3}	4.85×10^{-5}	(1.000)	4.86×10^{-9}	(0.037)	6.12×10^{-9}	(0.814)
2.00×10^{-4}	9.69×10^{-6}	(1.000)	4.85×10^{-9}	(0.002)	4.89×10^{-9}	(0.139)
1.00×10^{-4}	4.84×10^{-6}	(1.001)	4.85×10^{-9}	(0.000)	4.86×10^{-9}	(0.011)
2.00×10^{-5}	9.64×10^{-7}	(1.002)	4.85×10^{-9}	(0.000)	4.85×10^{-9}	(0.001)
1.00×10^{-5}	4.80×10^{-7}	(1.007)	4.85×10^{-9}	(0.000)	4.85×10^{-9}	(0.000)

6. Advection

One-dimensional advection is described by the equation

$$\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} = 0, \quad x \in [0, 1000], \quad t \in [0, 1000], \quad (6)$$

with advection velocity u , which was chosen to be 1. The initial condition was the same peak as in the diffusion model, and Neumann-type boundary condition was used at the inflow boundary. The first-order spatial derivative was discretized by a forward difference scheme, and then the forward Euler method was applied for time integration. This procedure leads to the relation

$$\tilde{c}(x_m, t_{n+1}) = \tilde{c}(x_m, t_n) - u \cdot \frac{\tilde{c}(x_m, t_n) - \tilde{c}(x_{m+1}, t_n)}{\Delta x} \cdot \Delta t. \quad (7)$$

Figure 6 illustrates the applied discretization scheme.

Table 5 and Figures 7 and 8 show the absolute and relative errors of the applied methods. Note that both extrapolation methods produced smaller absolute and relative errors than the Euler method. For large time steps the active Richardson extrapolation is only slightly better than the Eulerian scheme, and then the Richardson extrapolation becomes more and more accurate until the errors cannot decrease any further.

Figure 9 shows the relative errors as a function of the running time. Although the active Richardson extrapolation performs better than the pure Euler method, it is far less efficient than the passive Richardson method, which only needs 19.16 s to produce the most accurate numerical solution. This solution is 500 times more accurate than the best Eulerian solution.

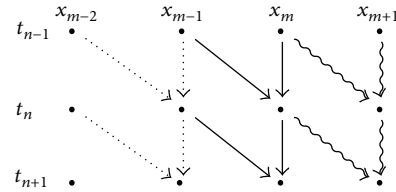


FIGURE 6: Discretization scheme for the advection problem.

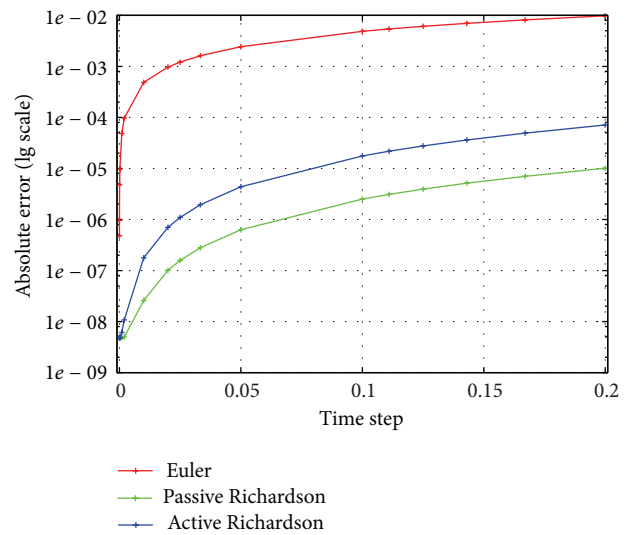


FIGURE 7: Absolute error as a function of time step in case of the advection problem for the Euler method without and with passive and active Richardson extrapolation. In the simulation the grid step was $\Delta x = 1$ and advection velocity $u = 0.1$. The reference solution was calculated using $\Delta t = 10^{-7}$.

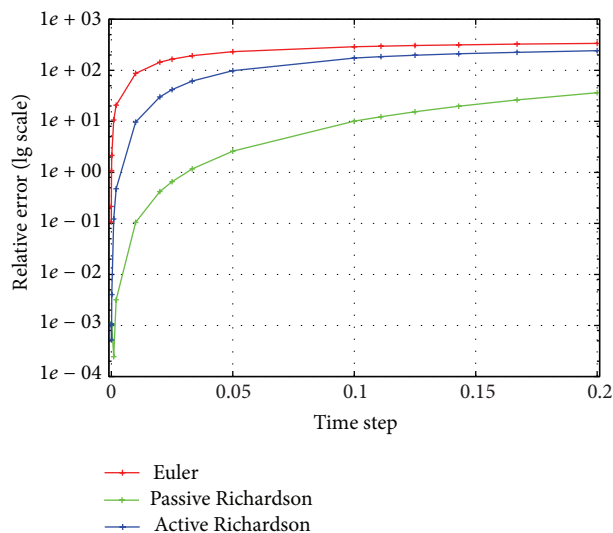


FIGURE 8: Relative error as a function of time step for the Euler method without and with passive and active Richardson extrapolation. In the simulation the grid step was $\Delta x = 1$ and the advection velocity $u = 0.1$. The reference solution was calculated using $\Delta t = 10^{-7}$.

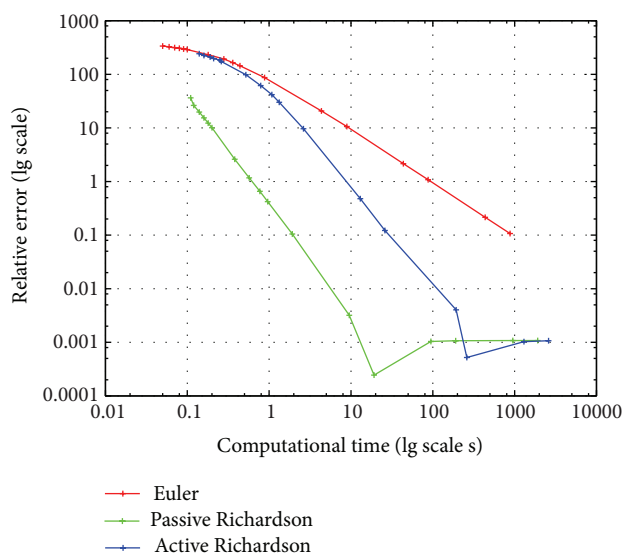


FIGURE 9: Relative error as a function of computational time for the Euler method without and with passive and active Richardson extrapolation. In the simulation the grid step was $\Delta x = 1$ and the advection velocity $u = 0.1$. The reference solution was calculated using $\Delta t = 10^{-7}$.

To summarize, both extrapolation methods are suitable for solving the advection problem, but the passive method is significantly more efficient.

7. The Cahn-Hilliard Model

Finally, we consider a more complex reaction-diffusion model problem, the so-called Cahn-Hilliard model [15–17]. It is suitable for describing the phenomenon of phase separation, which is used for the simulation of chemical

processes as well as for the formation of certain biological patterns and nucleation in meteorology [18, 19].

We chose the above model because it contains a fourth-order derivative of the unknown function. Our aim was to investigate whether the Richardson extrapolation methods show any noticeable difference due to the presence of the higher-order derivative in a complex reaction-diffusion system. We consider a pattern formation (phase separation) in the wake of a reaction front. This front emerges due to a strongly inhomogeneous initial distribution of the chemical species A and B . The reaction takes place in a domain occupying the half-space $x > 0$ and, initially, the inner electrolyte B is homogeneously distributed in it [$b(x > 0, t = 0) = b_0$]. Chemical species A with a much higher initial concentration [$a(x > 0, t = 0) = 10b_0$] is brought into contact with the domain at $t = 0$. Assuming a chemical reaction between reactants $A + B \rightarrow C$, the phase separation (pattern formation) behind the advancing front can be described by the set of equations

$$\begin{aligned}\frac{\partial a}{\partial t} &= D_A \frac{\partial^2 a}{\partial x^2} - kab, \\ \frac{\partial b}{\partial t} &= D_B \frac{\partial^2 b}{\partial x^2} - kab, \\ \frac{\partial c}{\partial t} &= -\lambda \nabla^2 (\epsilon c - \gamma c^3 + \sigma \nabla^2 c) + kab, \\ x &\in [0, 1000], \quad t \in [0, 40000],\end{aligned}\tag{8}$$

where k is the reaction rate constant and, for simplicity, the diffusion constants of the reagents are taken to be equal ($D_A = D_B = D$), and λ , ϵ , γ , and σ characterize the phase separation process.

Solving the equations above, we can simulate the pattern formation of chemical species C in the wake of the reaction front. The reference solution was computed again using the forward Euler method with a very small time step ($\Delta t = 10^{-7}$).

According to the results, the methods are not sensitive to the appearance of the higher-order derivatives. Both extrapolation methods can handle the problem very well and produce much better results than the application of the forward Eulerian method without extrapolation (Figure 10).

8. Conclusion

In this study, we showed a powerful integration method called Richardson extrapolation, which can be efficiently used to solve various reaction-diffusion and advection problems. This method increases the accuracy of the numerical results at the cost of slightly increased computational time. However, the Richardson extrapolation can provide an increase of around two orders of magnitude in the computational time to achieve the same error tolerance compared to a numerical scheme without using the extrapolation method. Combining this cost-efficient method with parallel computing techniques can provide a powerful method to solve partial differential equations.

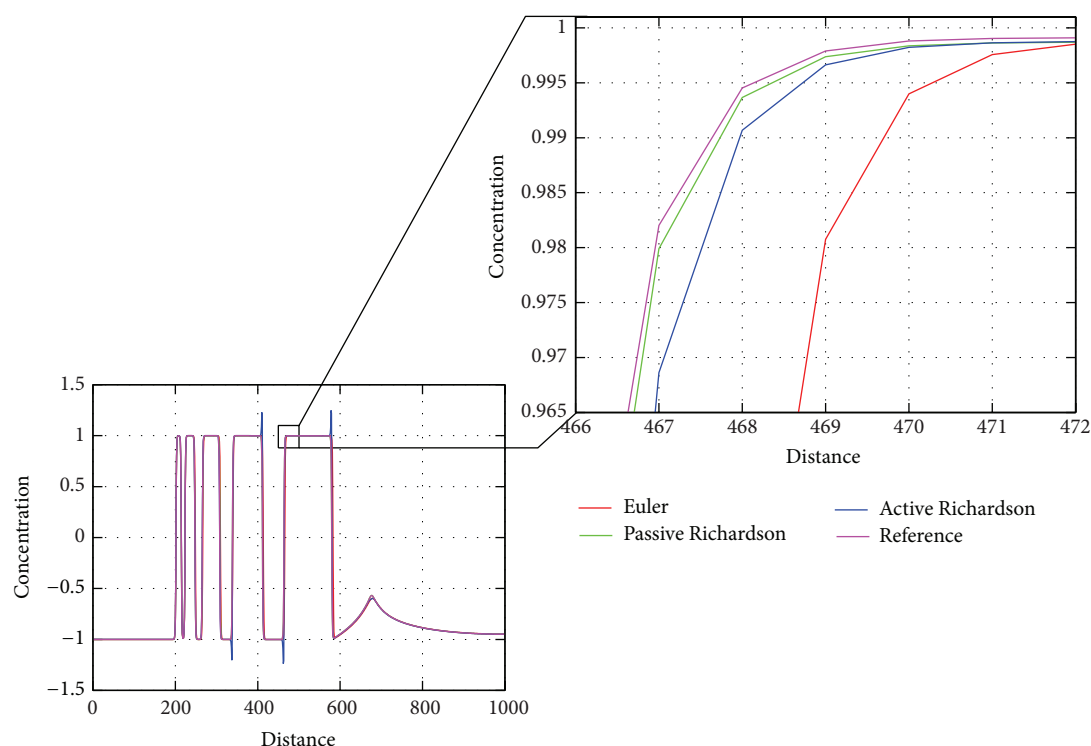


FIGURE 10: Results of the numerical simulations for the Cahn-Hilliard type reaction-diffusion model obtained by the Euler method without and with passive and active Richardson extrapolations. We used the following parameters in the simulations: $\Delta x = 1$, $D_A = D_B = 1.0$, $a_0 = 10.0$, $b_0 = 1.0$, and $\lambda = \epsilon = \gamma = \sigma = 1.0$. The reference solution was calculated using $\Delta t = 10^{-7}$.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Hungarian Research Found (OTKA K104666), the European Union, and the State of Hungary and cofinanced by the European Social Fund in the framework of TÁMOP 4.2.4. A/1-11-1-2012-0001 National Excellence Program. The European Union and the European Social Fund have provided financial support to the project under the Grant Agreement no. TÁMOP 4.2.1/B-09/1/KMR-2010-0003.

References

- [1] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, John Wiley & Sons, New York, NY, USA, 1991.
- [2] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons, 2008.
- [3] F. Molnár Jr., F. Izsák, R. Mészáros, and I. Lagzi, "Simulation of reaction—diffusion processes in three dimensions using CUDA," *Chemometrics and Intelligent Laboratory Systems*, vol. 108, no. 1, pp. 76–85, 2011.
- [4] P. A. Zegeling and H. P. Kok, "Adaptive moving mesh computations for reaction-diffusion systems," *Journal of Computational and Applied Mathematics*, vol. 168, no. 1-2, pp. 519–528, 2004.
- [5] I. Lagzi, D. Kármán, T. Turányi, A. S. Tomlin, and L. Haszpra, "Simulation of the dispersion of nuclear contamination using an adaptive Eulerian grid model," *Journal of Environmental Radioactivity*, vol. 75, no. 1, pp. 59–82, 2004.
- [6] F. Molnár Jr., T. Szakály, R. Mészáros, and I. Lagzi, "Air pollution modelling using a graphics processing unit with CUDA," *Computer Physics Communications*, vol. 181, no. 1, pp. 105–112, 2010.
- [7] M. Januszewski and M. Kostur, "Accelerating numerical solution of stochastic differential equations with CUDA," *Computer Physics Communications*, vol. 181, no. 1, pp. 183–188, 2010.
- [8] A. R. Sanderson, M. D. Meyer, R. M. Kirby, and C. R. Johnson, "A framework for exploring numerical solutions of advection-reaction-diffusion equations using a GPU-based approach," *Computing and Visualization in Science*, vol. 12, no. 4, pp. 155–170, 2009.
- [9] V. N. Alexandrov, W. Owczarz, P. G. Thomson, and Z. Zlatev, "Parallel runs of a large air pollution model on a grid of Sun computers," *Mathematics and Computers in Simulation*, vol. 65, no. 6, pp. 557–577, 2004.
- [10] C. Németh, G. Dózsa, R. Lovas, and P. Kacsuk, "The p-grade grid portal," in *Computational Science and Its Applications—ICCSA 2004*, pp. 10–19, Springer, 2004.
- [11] L. F. Richardson and J. A. Gaunt, "The deferred approach to the limit. Part I. Single lattice. Part II. Interpenetrating lattices," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 226, pp. 299–361, 1927.

- [12] I. Farag, G. Havasi, and Z. Zlatev, "Efficient implementation of stable Richardson extrapolation algorithms," *Computers & Mathematics with Applications*, vol. 60, no. 8, pp. 2309–2325, 2010.
- [13] Z. Zlatev, I. Faragó, and Á. Havasi, "Stability of the Richardson Extrapolation applied together with the θ -method," *Journal of Computational and Applied Mathematics*, vol. 235, no. 2, pp. 507–517, 2010.
- [14] I. Faragó, Á. Havasi, and Z. Zlatev, "The convergence of explicit Runge-Kutta methods combined with Richardson extrapolation," in *Applications of Mathematics*, pp. 99–106, GCSE, 2012.
- [15] J. W. Cahn and J. E. Hilliard, "Free energy of a nonuniform system. I. Interfacial free energy," *The Journal of Chemical Physics*, vol. 28, no. 2, pp. 258–267, 1958.
- [16] J. W. Cahn and J. E. Hilliard, "Spinodal decomposition: a reprise," *Acta Metallurgica*, vol. 19, no. 2, pp. 151–161, 1971.
- [17] C. M. Elliott and D. A. French, "Numerical studies of the Cahn-Hilliard equation for phase separation," *IMA Journal of Applied Mathematics*, vol. 38, no. 2, pp. 97–128, 1987.
- [18] Z. Rácz, "Formation of Liesegang patterns," *Physica A: Statistical Mechanics and its Applications*, vol. 274, no. 1, pp. 50–59, 1999.
- [19] T. Antal, M. Droz, J. Magnin, A. Pekalski, and Z. Rácz, "Formation of Liesegang patterns: simulations using a kinetic Ising model," *The Journal of Chemical Physics*, vol. 114, no. 8, pp. 3770–3775, 2001.

