

## A Sylvester-Based IMEX Method via Differentiation Matrices for Solving Nonlinear Parabolic Equations

Francisco de la Hoz<sup>1,\*</sup> and Fernando Vadillo<sup>1</sup>

<sup>1</sup> *Department of Applied Mathematics, Statistics and Operations Research, Faculty of Science and Technology, University of the Basque Country UPV/EHU, Barrio Sarriena S/N, 48940 Leioa, Spain.*

Received 5 June 2012; Accepted (in revised version) 18 January 2013

Communicated by John Boyd

Available online 18 April 2013

---

**Abstract.** In this paper we describe a new pseudo-spectral method to solve numerically two and three-dimensional nonlinear diffusion equations over unbounded domains, taking Hermite functions, sinc functions, and rational Chebyshev polynomials as basis functions. The idea is to discretize the equations by means of differentiation matrices and to relate them to Sylvester-type equations by means of a fourth-order implicit-explicit scheme, being of particular interest the treatment of three-dimensional Sylvester equations that we make. The resulting method is easy to understand and express, and can be implemented in a transparent way by means of a few lines of code. We test numerically the three choices of basis functions, showing the convenience of this new approach, especially when rational Chebyshev polynomials are considered.

**AMS subject classifications:** 65M20, 65M70

**PACS:** 02.70.Hm, 02.30.Jr

**Key words:** Semi-linear diffusion equations, pseudo-spectral methods, differentiation matrices, Hermite functions, sinc functions, rational Chebyshev polynomials, IMEX methods, Sylvester equations, blow-up.

---

## 1 Introduction

One of the most remarkable properties that distinguish nonlinear evolution problems from linear ones is the possibility of an eventual occurrence of singularities in a finite time  $T$ , starting from perfectly smooth data. One of the simplest forms of spontaneous singularities in nonlinear problems appears when one or more of the dependent variables

---

\*Corresponding author. *Email addresses:* francisco.delahoz@ehu.es (F. de la Hoz), fernando.vadillo@ehu.es (F. Vadillo)

tend to infinity as  $t \rightarrow T$ , where  $T \in (0, \infty)$  is called the blow-up time. A singularity represents often an important change in the properties of the models, which explains why it is important to analyze them and to reproduce them accurately by a numerical method.

In this paper, we consider the classical semi-linear parabolic equation

$$u_t = \Delta u + u^p, \quad p > 1, \quad \mathbf{x} \in \mathbb{R}^n, \quad t > 0, \quad (1.1)$$

with initial condition

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad (1.2)$$

where  $u_0(\mathbf{x})$  is continuous, non-negative and bounded.

The local existence in time of positive solutions of (1.1)-(1.2) follows from standard results, but the solution may develop singularities in finite time. More precisely, we have the following theorem [13, 23]:

**Theorem 1.1.** *Let  $p_c(n) = 1 + \frac{2}{n}$ .*

1. *If  $1 < p < p_c(n)$ , for any non-trivial solution of (1.1)-(1.2), there exists a finite time  $T$ , such that*

$$\limsup_{t \nearrow T} \left( \sup_{\mathbf{x} \in \mathbb{R}^n} u(\mathbf{x}, t) \right) = +\infty.$$

2. *If  $p > p_c(n)$ , there exists a global positive solution, if the initial values are sufficiently small (less than a small Gaussian).*

In the first case, we say that  $u(x, t)$  blows up in a finite time  $T$ , which is called the blow-up time of  $u$ ; and  $p_c$  is the critical exponent of the problem.

Another important question is to determine the asymptotic behavior of the solution, as the blow-up is approached. There are several references on this topic: in [18], the different possible asymptotic behaviors for  $n = 1$  are described, and in [33], the case  $n > 1$  is studied.

In the following pages, we will develop and test a new matrix-based pseudo-spectral method for (1.1) in two and three spatial dimensions. References on spectral methods can be found in [5, 7, 11, 29, 31, 32], together with the more classical [6, 15]. One of the main difficulties of dealing with (1.1) is the unboundedness of the spatial domain; nevertheless, according to Boyd [5, p. 338], the many options for unbounded domains fall into three broad categories:

1. Domain truncation (approximation of  $x \in \mathbb{R}$  by  $[-L, L]$ , with  $L \gg 1$ );
2. Basis functions intrinsic to an infinite interval (Hermite functions, sinc functions);
3. Mapping of the unbounded interval to a finite one, followed by application of Chebyshev polynomials or a Fourier series.

In a previous work [9], we applied successfully the domain truncation technique to (1.1). There, we assumed periodic solutions over a large enough rectangular domain, in such a way that the effects due to periodicity were minimal, but one could take advantage of the Fast Fourier Transform (FFT) to develop an efficient pseudo-spectral method, using a fourth-order exponential-time differencing (ETDRK4) scheme [22] to advance in time.

On the other hand, in all the theoretical results about (1.1), there always appear Hermite functions,

$$\psi_n(x) = \frac{1}{\pi^{1/4} \sqrt{2^n n!}} e^{-x^2/2} H_n(x), \quad (1.3)$$

where  $H_n(x)$  are the Hermite polynomials (see Appendix A.1), which encouraged us to develop a new pseudo-spectral method based on them. Furthermore, since the devised method can be implemented quite independently from the choice of the class of basis functions, we have also considered sinc functions (see Appendix A.2),

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}, \quad (1.4)$$

as well as a relevant particular case of mapped Chebyshev polynomials  $T_n(x)$ , the so called rational Chebyshev polynomials  $TB_n(x)$  (see Appendix A.3),

$$TB_n(x) \equiv T_n\left(\frac{x}{\sqrt{1+x^2}}\right). \quad (1.5)$$

In this paper, we test the proposed method using Hermite functions, sinc functions, and rational Chebyshev polynomials; moreover, we compare the results with those obtained in [9]. In this way, all the three cases mentioned by Boyd are covered. The structure of the paper is as follows. In Section 2, we explain the numerical method for the two-dimensional case. Section 3 is devoted to numerical tests in two dimensions. In Section 4, the ideas of Section 2 are extended to three-dimensional problems. Numerical tests in three dimensions are carried in Section 5. In Section 6 we draw the main conclusions, highlighting the most relevant aspects of the work and some possible applications to other equations. We finish the paper with two appendices. In Appendix A, we offer a brief introduction to the classes of functions used in the paper, as well as the differentiation matrices associated to them, incorporating also some new results. Appendix B, which is of particular interest, is about solving three-dimensional Sylvester equations.

All the algorithms have been implemented in MATLAB, Version 7.14 (R2012a), and executed in a Dell Precision T7500. They are available upon request.

## 2 Description of the numerical method in two spatial-dimensions

Let us consider the two-dimensional semi-linear parabolic equation

$$u_t = u_{xx} + u_{yy} + u^p, \quad p > 1, \quad (x, y, t) \in \mathbb{R}^2 \times \mathbb{R}^+, \quad (2.1)$$

with initial data

$$u(x,y,0) = u_0(x,y), \quad (x,y) \in \mathbb{R}^2, \tag{2.2}$$

and critical exponent  $p_c = 2$ . The idea of pseudo-spectral methods over unbounded domains is to approximate the solution  $u$  by a finite sum:

$$u(x,y,t) \approx \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \hat{u}_{ij}(t) \phi_i(L_x x) \phi_j(L_y y), \quad (x,y,t) \in \mathbb{R}^2 \times \mathbb{R}^+, \tag{2.3}$$

where  $L_x, L_y$  are two positive constants, and  $\phi_i, \phi_j$  are basis functions; in our case, Hermite functions, sinc functions, or rational Chebyshev polynomials. We introduce this expression into (2.1), making the residual equal to zero at  $(N_x + 1) \times (N_y + 1)$  conveniently chosen nodes  $(x_i, y_j)$ , which are the points at which we calculate the evolution of  $u$ . Then, (2.1) becomes a coupled system of ordinary differential equations for the coefficients  $\hat{u}_{ij}(t)$ .

Equivalently, we can adopt a matrix representation for  $u$ , defining  $\mathbf{U} \equiv [U_{ij}]$ , which is the matrix formed by the values of  $u$  at the nodes  $(x_i, y_j)$ , i.e.,  $U_{ij}(t) = u(x_i, y_j, t)$ . In that case, (2.1) becomes

$$\mathbf{U}_t = \mathbf{D}_x^{(2)} \cdot \mathbf{U} + \mathbf{U} \cdot (\mathbf{D}_y^{(2)})^T + \mathbf{U}^p, \tag{2.4}$$

where  $T$  denotes the transpose, and  $\mathbf{D}_x^{(2)}$  and  $\mathbf{D}_y^{(2)}$  are the second-order differentiation matrices with respect to  $x$  and  $y$ , respectively (see Appendix A):

$$u_{xx}(x_i, y_j) \approx \mathbf{D}_x^{(2)} \cdot \mathbf{U}, \quad u_{yy}(x_i, y_j) \approx \mathbf{U} \cdot (\mathbf{D}_y^{(2)})^T; \tag{2.5}$$

of course, if the amount of nodes in the axes  $x$  and  $y$ ,  $N_x + 1$  and  $N_y + 1$ , is the same, we can simply write  $\mathbf{D}^{(2)}$ . Introducing the approximations (2.5) into (2.1), it becomes discretized as a non-linear matrix differential equation:

$$\mathbf{U}_t = \mathbf{D}_x^{(2)} \cdot \mathbf{U} + \mathbf{U} \cdot (\mathbf{D}_y^{(2)})^T + \mathbf{U}^p. \tag{2.6}$$

At this point, it is straightforward to apply an explicit time-discretization to advance in time, although we will usually have a severe time-step restriction on  $\Delta t$ . Therefore, an implicit-explicit (IMEX) scheme [1], where we treat implicitly the linear parts and explicitly the non-linear ones, seems a good choice, because it is relatively easy to advance from  $t^n$  to  $t^{n+1}$ , yet much smaller constraints on  $\Delta t$  may be expected.

Let us apply, for instance, a first-order semi-implicit Euler discretization in time to (2.6), which is the most simple IMEX method:

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = \mathbf{D}_x^{(2)} \cdot \mathbf{U}^{n+1} + \mathbf{U}^{n+1} \cdot (\mathbf{D}_y^{(2)})^T + (\mathbf{U}^n)^p. \tag{2.7}$$

This equation can be rewritten as

$$\left( \frac{1}{2} \mathbf{I}_x - \Delta t \mathbf{D}_x^{(2)} \right) \cdot \mathbf{U}^{n+1} + \mathbf{U}^{n+1} \cdot \left( \frac{1}{2} \mathbf{I}_y - \Delta t \mathbf{D}_y^{(2)} \right)^T = \mathbf{U}^n + \Delta t (\mathbf{U}^n)^p, \tag{2.8}$$

i.e., a Sylvester equation of the type

$$\mathbf{A}_1 \cdot \mathbf{X} + \mathbf{X} \cdot \mathbf{A}_2^T = \mathbf{B}, \tag{2.9}$$

with  $\mathbf{A}_1 = \frac{1}{2}\mathbf{I}_x - \Delta t \mathbf{D}_x^{(2)}$ ,  $\mathbf{A}_2 = \frac{1}{2}\mathbf{I}_y - \Delta t \mathbf{D}_y^{(2)}$ .

Sylvester equations have been extensively studied (see for instance [19,20]). The standard procedure to solve them is based on the Bartels-Steward algorithm [2], which is widely implemented (let us mention, for instance, the command `lyap` in Matlab). Although this algorithm uses the complex Schur form, it is numerically safe to obtain real solutions [26]. On the other hand, if  $\mathbf{A}_1$  and  $\mathbf{A}_2$  are diagonalizable, which is our case, i.e.,  $\mathbf{A}_1 = \mathbf{P}_1 \cdot \mathbf{D}_1 \cdot \mathbf{P}_1^{-1}$ ,  $\mathbf{A}_2 = \mathbf{P}_2 \cdot \mathbf{D}_2 \cdot \mathbf{P}_2^{-1}$ , with  $\mathbf{P}_1$  and  $\mathbf{P}_2$  well conditioned, then, defining  $\mathbf{Y} = \mathbf{P}_1^{-1} \cdot \mathbf{X} \cdot (\mathbf{P}_2^{-1})^T$ , (2.9) can be safely transformed into

$$\mathbf{D}_1 \cdot \mathbf{Y} + \mathbf{Y} \cdot \mathbf{D}_2 = \mathbf{P}_1^{-1} \cdot \mathbf{B} \cdot (\mathbf{P}_2^{-1})^T = \mathbf{C}, \tag{2.10}$$

so we get the trivial equation

$$\Lambda \circ \mathbf{Y} = \mathbf{C}, \tag{2.11}$$

where  $\Lambda \equiv [\lambda_{ij}]$  is the matrix whose  $\lambda_{ij}$  element is the  $i$ -th eigenvalue of  $\mathbf{A}_1$  plus the  $j$ -th eigenvalue of  $\mathbf{A}_2$ , i.e.,  $\lambda_{ij} = \lambda_i(\mathbf{A}_1) + \lambda_j(\mathbf{A}_2)$ ; and  $\circ$  denotes the point-wise or Hadamard product between matrices. In our case, the eigenvalues of  $\mathbf{A}_1$  and  $\mathbf{A}_2$  are respectively  $\lambda_i = \frac{1}{2} - \Delta t \mu_i$  and  $\lambda_j = \frac{1}{2} - \Delta t \mu_j$ , where  $\mu_i$  and  $\mu_j$  are respectively the eigenvalues of  $\mathbf{D}_x^{(2)}$  and  $\mathbf{D}_y^{(2)}$ . On the one hand, for a small enough  $\Delta t$ ,  $\lambda_i$  and  $\lambda_j$  will always be positive; on the other hand, at least for the second-order differentiation matrices we are dealing with, we have **always**  $\mu_i < 0$  and  $\mu_j < 0$  (see for instance [34]). Therefore,  $\forall \Delta t > 0$ ,  $\lambda_i > 0$  and  $\lambda_j > 0$ , and we conclude that (2.8) has always a unique solution, because the sum of any eigenvalue of  $\frac{1}{2}\mathbf{I}_x - \Delta t \mathbf{D}_x^{(2)}$  and any eigenvalue of  $\frac{1}{2}\mathbf{I}_y - \Delta t \mathbf{D}_y^{(2)}$  is never equal to zero, i.e.,  $\lambda_i + \lambda_j = \lambda_{ij} \neq 0$  in (2.11).

In our numerical experiments, we have observed that, for our differentiation matrices,  $\frac{1}{2}\mathbf{I}_x - \Delta t \mathbf{D}_x^{(2)}$  and  $\frac{1}{2}\mathbf{I}_y - \Delta t \mathbf{D}_y^{(2)}$  are always diagonalizable, with **well-conditioned eigenvalue matrices**. Hence, it is numerically save to reduce (2.8) to (2.11). Moreover, since  $\frac{1}{2}\mathbf{I}_x - \Delta t \mathbf{D}_x^{(2)}$  and  $\frac{1}{2}\mathbf{I}_y - \Delta t \mathbf{D}_y^{(2)}$  are time-independent, their diagonal decompositions have to be computed just once for all the time-steps, obtaining an extremely efficient numerical scheme:

$$\mathbf{U}^{n+1} = \mathbf{P}_1 \cdot \left[ \left[ \mathbf{P}_1^{-1} \cdot [\mathbf{U}^n + \Delta t (\mathbf{U}^n)^p] \cdot (\mathbf{P}_2^{-1})^T \right] \circ \Lambda^{-1} \right] \cdot \mathbf{P}_2^T, \tag{2.12}$$

where  $\Lambda^{-1}$  denotes the point-wise inverse of  $\Lambda$ , i.e.,  $\Lambda^{-1} \equiv [1/(\lambda_i + \lambda_j)]$ . The computational cost is of just  $2N_x N_y (N_x + N_y)$  multiplications and  $N_x N_y$  divisions per time-step, i.e., of the same order of magnitude as an explicit scheme.

It is also possible to consider higher-order schemes. More precisely, in this paper we have chosen a fourth-order semi-implicit backward differentiation formula (SBDF4) [1],

which has good stability properties. Applying it to (2.6), we get the following numerical scheme:

$$\begin{aligned} & \frac{1}{\Delta t} \left( \frac{25}{12} \mathbf{U}^{n+1} - 4\mathbf{U}^n + 3\mathbf{U}^{n-1} - \frac{4}{3} \mathbf{U}^{n-2} + \frac{1}{4} \mathbf{U}^{n-3} \right) \\ &= \mathbf{D}_x^{(2)} \cdot \mathbf{U}^{n+1} + \mathbf{U}^{n+1} \cdot (\mathbf{D}_y^{(2)})^T + 4(\mathbf{U}^n)^p - 6(\mathbf{U}^{n-1})^p + 4(\mathbf{U}^{n-2})^p - (\mathbf{U}^{n-3})^p, \end{aligned} \quad (2.13)$$

which can be conveniently transformed into another Sylvester equation of the form

$$\begin{aligned} & \left( \frac{25}{24} \mathbf{I}_x - \Delta t \mathbf{D}_x^{(2)} \right) \cdot \mathbf{U}^{n+1} + \mathbf{U}^{n+1} \cdot \left( \frac{25}{24} \mathbf{I}_y - \Delta t \mathbf{D}_y^{(2)} \right)^T \\ &= 4\mathbf{U}^n - 3\mathbf{U}^{n-1} + \frac{4}{3} \mathbf{U}^{n-2} - \frac{1}{4} \mathbf{U}^{n-3} \\ & \quad + \Delta t \left[ 4(\mathbf{U}^n)^p - 6(\mathbf{U}^{n-1})^p + 4(\mathbf{U}^{n-2})^p - (\mathbf{U}^{n-3})^p \right]. \end{aligned} \quad (2.14)$$

Now, by the same arguments, the eigenvalues are  $\lambda_i = \frac{25}{24} - \Delta t \mu_i > 0$  and  $\lambda_j = \frac{25}{24} - \Delta t \mu_j > 0$ ,  $\forall \Delta t > 0$ , and, consequently, (2.14) has a unique solution. Moreover, (2.14) can be solved again computing only  $2N_x N_y (N_x + N_y)$  multiplications and  $N_x N_y$  divisions per time-step.

Since this is a multi-step scheme, additional starting values are required. More specifically, the first value  $\mathbf{U}^0$  at  $t = t^0$  is the initial condition (2.2), but we also need fourth-order approximations of  $u$  at  $t \in \{t^1, t^2, t^3\}$ , i.e.,  $\mathbf{U}^1$ ,  $\mathbf{U}^2$  and  $\mathbf{U}^3$ , which are computed by means of a classical Richardson extrapolation [28]. For instance, to get  $\mathbf{U}^1$ ,

$$\mathbf{U}^1 = \frac{64}{21} \mathbf{U} \left( t^1, \frac{\Delta t}{8} \right) - \frac{8}{3} \mathbf{U} \left( t^1, \frac{\Delta t}{4} \right) + \frac{2}{3} \mathbf{U} \left( t^1, \frac{\Delta t}{2} \right) - \frac{1}{21} \mathbf{U} \left( t^1, \Delta t \right), \quad (2.15)$$

where  $\mathbf{U}(t^1, \Delta t/8)$  denotes the first-order approximation of  $u$  at  $t = t^1$ , using a time step equal to  $\Delta t/8$ , etc.

Since we are dealing with real positive solutions of (1.1), we round to zero those infinitesimally small negative values that might appear during the numerical simulations.

### 2.1 Working in the Fourier side

Sometimes, it can be preferable to consider the evolution of the coefficients  $\hat{u}_{ij}(t)$  in (2.3), instead of  $U_{ij}(t)$ . That happens when there is some sort of fast transform that converts  $\{\hat{u}_{ij}(t)\}$  into  $\{U_{ij}(t)\}$  and vice-versa, and the differentiation matrix for the  $\hat{u}_{ij}(t)$  coefficients is better structured and, preferably, sparse. As shown in Appendix A.3, these conditions are satisfied by the rational Chebyshev polynomials, so we use this approach in our numerical experiments for these functions.

Let  $\mathcal{F}$  be the discrete transform that converts  $\{U_{ij}(t)\}$  into  $\{\hat{u}_{ij}(t)\}$ , and  $\mathcal{F}^{-1}$  its inverse. We denote  $\hat{\mathbf{U}}(t)$  the matrix formed by  $\{\hat{u}_{ij}(t)\}$ . Then, applying  $\mathcal{F}$  to (2.4), we get

$$\hat{\mathbf{U}}_t = \mathbf{D}_x^{(2)} \cdot \hat{\mathbf{U}} + \hat{\mathbf{U}} \cdot (\mathbf{D}_y^{(2)})^T + \mathcal{F} \left[ \left( \mathcal{F}^{-1}(\hat{\mathbf{U}}) \right)^p \right], \quad (2.16)$$

where, with some notational abuse,  $\mathbf{D}_x^{(2)}$  and  $\mathbf{D}_y^{(2)}$  denote now the differentiation matrices in the Fourier side (A.23). At this point, (2.14) and all the other schemes previously defined can be applied **with absolutely no change**, and the computational cost is again of  $\mathcal{O}(N_x N_y (N_x + N_y))$  operations for solving the Sylvester equations of the form

$$\mathbf{A}_1 \cdot \mathbf{X} + \mathbf{X} \cdot \mathbf{A}_2^T = \mathbf{B} \tag{2.17}$$

that appear during the process, plus another  $\mathcal{O}(N_x N_y (\log(N_x) + \log(N_y)))$  operations, which are needed for computing  $\mathcal{F}$  and  $\mathcal{F}^{-1}$ .

Alternatively, given the sparse structure of  $\mathbf{A}_1$  and  $\mathbf{A}_2$ , we can consider an alternating direction implicit (ADI) iteration [27] to solve (2.17). Indeed, when working with rational Chebyshev polynomials, our problem closely resembles that in [17]. In our case, provided an initial guess  $\mathbf{X}^0$ , the iterational ADI scheme would be as follows:

$$(\omega_{1,\nu} \mathbf{I} + \mathbf{A}_1) \mathbf{X}^{\nu+1/2} = \mathbf{X}^\nu (\omega_{2,\nu} \mathbf{I} - \mathbf{A}_2^T) + \mathbf{B}, \tag{2.18a}$$

$$\mathbf{X}^{\nu+1} (\omega_{2,\nu} \mathbf{I} + \mathbf{A}_2^T) = (\omega_{1,\nu} \mathbf{I} - \mathbf{A}_1) \mathbf{X}^{\nu+1/2} + \mathbf{B}, \tag{2.18b}$$

where  $\mathbf{I}$  are identity matrices. As explained in [17], if  $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A} \in \mathcal{M}_{(N-2) \times (N-2)}$ , then (2.17) can be solved exactly after  $N - 1$  stages, by taking  $\omega_{1,\nu} = \omega_{2,\nu} = \lambda_\nu$ ,  $\nu = 0, 1, \dots, N - 2$ , where  $\lambda_\nu$  are chosen to be the eigenvalues of  $\mathbf{A}$ . On the other hand, the number of iterations can be further reduced to  $N/2$ , by using the even-odd uncoupling of the coefficients of  $\mathbf{X}$ , to factor (2.17) into four separate equations. In any case, since each stage requires  $\mathcal{O}(N^2)$  operations, we need again  $\mathcal{O}(N^3)$  operations to solve (2.17), i.e., the same order of magnitude needed to solve it through a diagonal factorization of  $\mathbf{A}_1$  and  $\mathbf{A}_2$ . Therefore, in our opinion, the ADI approach should be preferred rather for larger Systems.

### 3 Numerical experiments in two dimensions

#### 3.1 Numerical tests for problems with known exact solutions

It is a well known fact [5] that Hermite and sinc functions work best for problems with exponential decay, while rational Chebyshev polynomials are an optimal choice (not only) for problems with polynomial decay. Hence, we consider two modifications of  $u_t = \Delta u + u^3$ :

$$\begin{cases} u_t = \Delta u - 2 \exp(4t) [\sinh^2(x) \cosh^2(y) + \cosh^2(x) \sinh^2(y)] u^3, \\ u_0(x, y) = \operatorname{sech} x \operatorname{sech} y, \end{cases} \tag{3.1}$$

whose known exact solution,  $u(x, y, t) = \exp(-2t) \operatorname{sech} x \operatorname{sech} y$ , has exponential decay; and

$$\begin{cases} u_t = \Delta u - 2 \exp(4t) [(x^2 + y^2 + 2)^2 - 5] u^3, \\ u_0(x, y, z) = \frac{1}{1 + x^2 + y^2}, \end{cases} \tag{3.2}$$

whose known exact solution,  $u(x,y,t) = \exp(-2t)/(1+x^2+y^2)$ , has polynomial decay. We test both equations with the three classes of functions considered in this paper, as well as with the method developed in [9], which combines the domain truncation technique with a fourth-order exponential-time differencing (ETDRK4) scheme [22] to advance in time. This is a very exigent test, because the method in [9] is extremely well suited for the type of equations we are considering, provided that the solutions decay fast enough.

Before proceeding with the experiments, some important remarks are necessary. If we look at (2.3), we will see that there are two positive constants  $L_x$  and  $L_y$ . Indeed, if a family of functions  $\{\phi_i(x)\phi_j(y)\}$  forms an orthogonal basis of  $\mathbb{R}^2$ , so does  $\{\phi_i(L_x x)\phi_j(L_y y)\}$ ,  $\forall L_x, L_y > 0$ . Therefore, in order to specify a pseudo-spectral method over an unbounded domain, both the number of grid points and the scaling factors are required, i.e., two parameters. This happens in the domain truncation technique, too, where two parameters are also required: the dimensions of the domain, and the number of points.

In Appendix A, we describe the differentiation matrices associated to the families of functions we are dealing with. While our definition of the sinc differentiation matrix is taken directly from [35], we have constructed a new version of the Hermite differentiation matrix that is numerically identical (up to the machine accuracy) to that in [35], yet it is stable for a much larger number of points. We have also developed our own versions of the rational Chebyshev differentiation matrices, both in space and in the Fourier side. For the Hermite and rational Chebyshev cases, we give directly the corresponding rescaling factor  $L$ ; while it is customary to give the separation between grid points  $h$  in the sinc case. Remark that in the definition of the Hermite differentiation matrix in [35], their parameter  $b$  is the inverse of the rescaling factor, i.e.,  $b = 1/L$ .

In practice, the correct choice of the scaling is a matter of concern. Even if some theoretical results do exist [3], the optimal values depend on the number of points, the class of functions, and the type of problem. The optimal scaling can even change during time [25]. A good working rule of thumb seems to be that the absolute value of the function at the extremal grid points is smaller than an accuracy threshold  $\varepsilon$ .

Coming back to our experiments, we have computed the numerical solution for (3.1) and (3.2) at  $t = 1$ , for  $N = N_x = N_y = 129$ , and several  $\Delta t$ . It is convenient to take an odd number of nodes in the spatial discretization, so that the origin of coordinates, where the maximum is located, is included. Some extra care may be also needed with (3.1) if extremely large domains are considered, because the multiplier of the non-linear part grows as  $\mathcal{O}(\exp(2|x|+2|y|))$ , while  $u^3$  decreases as  $\mathcal{O}(\exp(-3|x|-3|y|))$ . Hence, indeterminations of the form  $\infty \cdot 0$  may arise when evaluating  $[\sinh^2(x)\cosh^2(y) + \cosh^2(x)\sinh^2(y)]u^3$  at the grid points  $(x_i, y_j)$  with very large values  $|x_i| + |y_j|$ , obtaining eventually NaN values. In those cases, we have to give the computer the correct value of the indetermination, i.e.,  $\infty \cdot 0 \equiv 0$ .

In order to make the results easier to compare, we have chosen the scalings in such a way that the grid points satisfy  $(x_i, y_j) \in [-20, 20]^2$  in the truncation, Hermite and sinc cases, which seems to be an adequate domain for  $N = 129$ , because  $\text{sech}(20) = 4.1223 \cdots \times 10^{-9}$ . More carefully chosen discretization domains give slightly better results. On the

other hand, we have found that a much larger domain is needed to obtain good results in the rational Chebyshev case; more precisely, we have taken a scaling such that the grid points satisfy  $(x_i, y_j) \in [-500, 500]^2$ . Nevertheless, although the discretization domain for the rational Chebyshev case is  $25^2$  times the size of the discretization domain for the other cases, the corresponding grid points are much more spread out ( $x_0/x_1 = y_0/y_1 \approx 3$ ,  $x_0/x_2 = y_0/y_2 \approx 5$ , etc.), so two thirds of them satisfy  $(x_i, y_j) \in [-20, 20]^2$ , too.

Tables 1 and 2 show the respective errors in  $\mathcal{L}^\infty$ -norm at  $t = 1$ . Notice that  $\exp(-2) = 0.1353\dots$ . In the exponentially decaying solutions, the four cases yield good results. The best ones are given by the rational Chebyshev polynomials, followed by the Hermite functions, while the sinc functions give an accuracy similar to the truncation method; even if it is true that the truncation method is slightly more accurate than the other three for larger  $\Delta t$ , due to its powerful ETDRK4 discretization in time. The fourth order of the SBDF4 method [1] that we are using is obvious from the results. Remark that the error of the rational Chebyshev polynomials further decays to  $9.5141 \cdot 10^{-13}$ , for  $\Delta t = 1/800$ .

Table 1: Maximum error in  $\mathcal{L}^\infty$ -norm at  $t = 1$ ,  $\max_{ij} |U_{ij} - u(x_i, y_j, 1)|$ , of the numerical solutions with exponential decay, for  $N_x = N_y = 129$ , and different  $\Delta t$ . The scalings are: truncation,  $(x_i, y_j) \in [-20, 20]^2$ ; Hermite,  $(x_i, y_j) \in [-20, 20]^2$ ,  $L = 1.3025031128994204$ ; sinc:  $(x_i, y_j) \in [-20, 20]^2$ ,  $h = 0.3125$ ; rational Chebyshev:  $(x_i, y_j) \in [-500\dots, 500\dots]^2$ ,  $L = 6.088658785462852$ .

$\Delta t^{-1}$	Truncation	Hermite	Sinc	R. Cheb.
25	$5.3126 \cdot 10^{-7}$	$1.2066 \cdot 10^{-6}$	$1.2171 \cdot 10^{-6}$	$1.2080 \cdot 10^{-6}$
50	$3.0468 \cdot 10^{-8}$	$7.1401 \cdot 10^{-8}$	$8.6348 \cdot 10^{-8}$	$7.1257 \cdot 10^{-8}$
100	$1.7258 \cdot 10^{-8}$	$4.5719 \cdot 10^{-9}$	$2.1372 \cdot 10^{-8}$	$4.3203 \cdot 10^{-9}$
200	$1.7109 \cdot 10^{-8}$	$5.9570 \cdot 10^{-10}$	$1.7422 \cdot 10^{-8}$	$2.6578 \cdot 10^{-10}$
400	$1.7100 \cdot 10^{-8}$	$3.5248 \cdot 10^{-10}$	$1.718 \cdot 10^{-8}$	$1.6411 \cdot 10^{-11}$

Table 2: Maximum error in  $\mathcal{L}^\infty$ -norm at  $t = 1$ ,  $\max_{ij} |U_{ij} - u(x_i, y_j, 1)|$ , of the numerical solutions with polynomial decay, for  $N_x = N_y = 129$ , and different  $\Delta t$ . The scalings are: truncation,  $(x_i, y_j) \in [-20, 20]^2$ ; Hermite,  $(x_i, y_j) \in [-20, 20]^2$ ,  $L = 1.3025031128994204$ ; sinc:  $(x_i, y_j) \in [-20, 20]^2$ ,  $h = 0.3125$ ; rational Chebyshev:  $(x_i, y_j) \in [-500\dots, 500\dots]^2$ ,  $L = 6.088658785462852$ .

$\Delta t^{-1}$	Truncation	Hermite	Sinc	R. Cheb.
25	$1.5772 \cdot 10^{-5}$	$8.6635 \cdot 10^{-5}$	$2.3734 \cdot 10^{-4}$	$1.0270 \cdot 10^{-6}$
50	$1.5770 \cdot 10^{-5}$	$8.6672 \cdot 10^{-5}$	$2.3737 \cdot 10^{-4}$	$6.0346 \cdot 10^{-8}$
100	$1.5770 \cdot 10^{-5}$	$8.6674 \cdot 10^{-5}$	$2.3737 \cdot 10^{-4}$	$3.6591 \cdot 10^{-9}$
200	$1.5770 \cdot 10^{-5}$	$8.6674 \cdot 10^{-5}$	$2.3737 \cdot 10^{-4}$	$4.1954 \cdot 10^{-10}$
400	$1.5770 \cdot 10^{-5}$	$8.6674 \cdot 10^{-5}$	$2.3737 \cdot 10^{-4}$	$4.1956 \cdot 10^{-10}$

The situation changes completely for the polynomially decaying solutions, where the rational Chebyshev polynomials are the only alternative. Indeed, the large discretization domain  $[-500, 500]^2$  is very convenient for those solutions. On the other hand, the domains for the other three cases,  $(x_i, y_j) \in [-20, 20]^2$ , are obviously too small, but increas-

ing them does not significantly improve the results; rather the opposite, because we lose resolution.

In view of the results, we can summarize by saying that the method developed in this paper is a serious competitive alternative to the domain truncation technique for exponentially decaying solutions, and that it really excels in the case of polynomially decaying solutions.

While the excellent behavior of the rational Chebyshev polynomials for problems involving solutions with polynomial decay makes them the choice *par excellence* for that kind of situations, they fit extremely well also with solutions with exponential decay. To better illustrate this, in Fig. 1 we plot the rational Chebyshev spectrum of  $\text{sech}(x)$  (in blue), and also of  $\exp(-x^2)$  (in red), for  $x \in [-500, 500]$ ,  $L = 6.088658785462852$ ,  $N = 129$ . Since these functions are even, their corresponding odd modes are equal to zero, so we only show the even modes, which, according to Fig. 1, decay exponentially. A similar analysis of the spectrum could be done for other types of functions, like  $\exp(-x^{2n})$ ,  $n > 1$ , etc., confirming the rational Chebyshev polynomials as an adequate tool for studying numerically a wide range of problems.

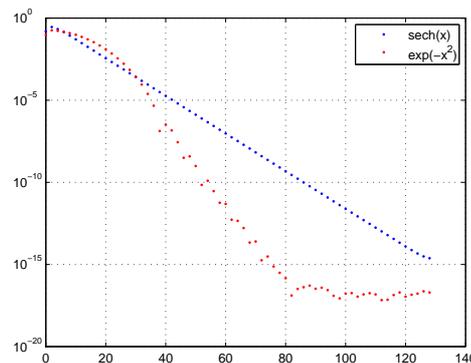


Figure 1: Spectra of  $\text{sech}(x)$  (blue) and  $\exp(-x^2)$  (red), for  $x \in [-500, 500]$ ,  $L = 6.088658785462852$ ,  $N = 129$ . We plot the even modes 0:2:128. The ordinate axis is in logarithmic scale.

### 3.2 Some blow-up problems in two dimensions

In this section, we consider two examples, for which no exact solution is known. In our first example, we chose an initial condition of (2.1) with three peaks and Gaussian decay:

$$u_0(x, y) = 11.2 \exp[-10((x+2)^2 + (y+4)^2)] + 10 \exp[-10((x-2)^2 + (y-2)^2)] \\ + 10 \exp[-15((x+2)^2 + (y-4)^2)].$$

We take  $p = 3/2$ , so, according to Theorem 1.1, the solution will blow up at some finite time; in this case, at  $t \approx 6.13$ .

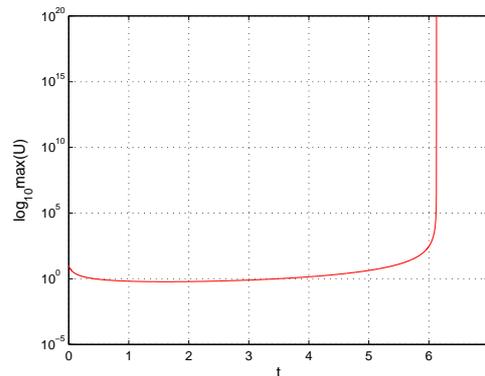


Figure 2: Maximum of  $U$  in logarithmic scale.

The maximum of  $U$  in function of  $t$  is plotted in Fig. 2. Initially, all the peaks decrease until  $t \approx 1.623$ , when the maximum is approximately 0.6. During the evolution, the peak located at  $t=0$  at  $(x,y) = (2,2)$  absorbs the peak located at  $t=0$  at  $(x,y) = (-2,4)$ . The two remaining peaks blow up at  $t \approx 1.63$ . In Fig. 3, we have plotted the evolution at  $t = 0.3$ , where the three peaks are clearly visible; at  $t = 1.2$ , where the smaller peak has been almost completely absorbed; and at  $t = 6.1$ , where the blow-up has almost happened. There is strong numerical evidence that the Gaussian decay is preserved throughout the evolution.

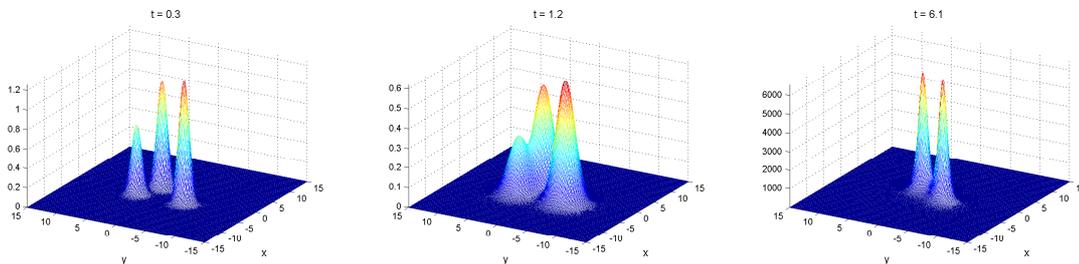


Figure 3: Evolution of the three peaks at  $t=0.3, 1.2, 6.1$ .

We have compared the four numerical methods at  $t = 6.12$ , where the solution is very close to the blow-up. The two remaining peaks, i.e., those initially located at  $(x,y) = (2,2)$  and  $(x,y) = (-2,-4)$  are of almost the same height, but the later is slightly higher.

Because the position of the peaks changes with time, and the grid points chosen in each method are different, each method will locate the summits of the peaks at slightly different positions, so the results will be necessarily different. In Table 3, together with the elapsed times, we show the heights of the summits obtained with the four methods; it is remarkable that these values only change in the second decimal place. Moreover, to compare them graphically, we have intersected the three dimensional graphics  $(x,y,u(x,y))$

Table 3: Height of the peaks at  $t=6.12$ , for  $N=243$ ,  $\Delta t=10^{-3}$ . The scalings are: truncation,  $(x_i, y_j) \in [-20, 20]^2$ ; Hermite,  $(x_i, y_j) \in [-20, 20]^2$ ,  $L = 0.93438423130303103$ ; sinc:  $(x_i, y_j) \in [-20, 20]^2$ ,  $h = 0.16528925619834711$ ; rational Chebyshev:  $(x_i, y_j) \in [-309.39\dots, 309.39\dots]^2$ ,  $L=2$ .

Method	First peak	Second peak	Elapsed time
Truncation	$1.1376 \cdot 10^5$	$1.0617 \cdot 10^5$	138 s.
Hermite	$1.1095 \cdot 10^5$	$1.0491 \cdot 10^5$	51 s.
Sinc	$1.1336 \cdot 10^5$	$1.0510 \cdot 10^5$	51 s.
Rational Chebyshev	$1.1450 \cdot 10^5$	$1.0552 \cdot 10^5$	140 s.

with the plane parallel to the  $OZ$  axis and that contains both summits, obtaining the curves show in Fig. 4, which are practically coincident. It is also interesting to note that the Hermite and sinc methods, which do not use FFT, are much faster. This is no surprise, because schemes like (2.12) are as efficient as an explicit method.

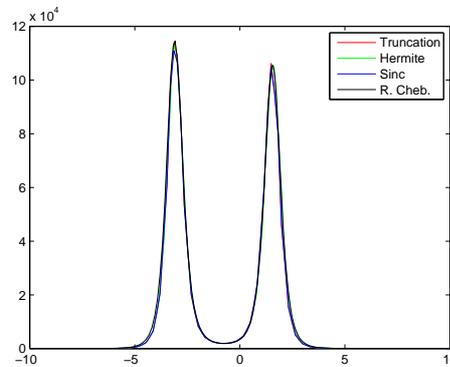


Figure 4: Transversal cuts. The four curves are practically coincident.

In our second example, we simulate the evolution of (2.1) for different  $p$ , but the same initial data,

$$u_0(x, y) = \exp(-x^2 - y^2),$$

using Hermite functions, together with  $N = 199$ ,  $L = 2$ ,  $(x_i, y_j) \in [-38.58\dots, 38.58\dots]$ ,  $\Delta t = 10^{-2}$ . In order to estimate the blow-up time in function of  $p$ , we have executed the code for  $p = 1 : 0.01 : 1.99$ , and stored the  $t$  when  $\max_{ij}(U_{ij}) > 10^{10}$ ,  $\max_{ij}(U_{ij}) > 10^{15}$ , and  $\max_{ij}(U_{ij}) > 10^{20}$ , which are plotted in Fig. 5. For  $p=1$ , there is obviously no blow-up, but an exponential growing of the maximum of  $u$ . For  $1 < p < 2$ , there is indeed explosion, but two situations have to be distinguished. Until  $p \approx 1.365$ , the blow-up time decreases, as  $p$  increases, while from  $p \approx 1.365$ , the blow-up time increases with  $p$ , tending to infinity as we approach the critical exponent  $p_c = 2$ .

In Fig. 6, we have depicted in logarithmic scale the evolution of the maximum of  $\mathbf{U}$  in function of time. The left hand-side corresponds to  $p = 1 : 0.01 : 1.36$ . The case  $p = 1$  yields obviously a straight line in logarithmic scale, because of the exponential growing.

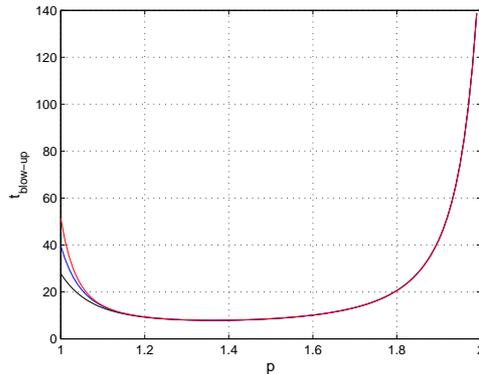


Figure 5: Instants when  $\max(\mathbf{U}) > 10^{10}$  (black),  $\max(\mathbf{U}) > 10^{15}$  (blue), and  $\max(\mathbf{U}) > 10^{20}$  (red), in function of  $p$ .

Then, the curve immediately to its left corresponds to  $p = 1.01$  and so on, until the left-most curve, which corresponds to  $p = 1.36$ . Observe that for  $p$  close to  $p = 1$ , the blow-up time can be calculated only very roughly, which is also evident from Fig. 5. Indeed, Fig. 5 suggests that the actual blow-up time would tend to infinity also as  $p \rightarrow 1^+$ .

The right hand-side of Fig. 6 corresponds to  $p = 1 : 39 : 0.05 : 1.99$ . Unlike the previous case, the left-most curve corresponds to  $p = 1 : 39$ ; the next one to its right, to  $p = 1.44$ ; the next one, to  $p = 1.49$ , and so on. In all of them, the explosion in finite time happens very sharply and, hence, the blow-up time can be estimated accurately. Observe also that, as we approach  $p = 2$ , the distance between two curves becomes bigger and bigger.

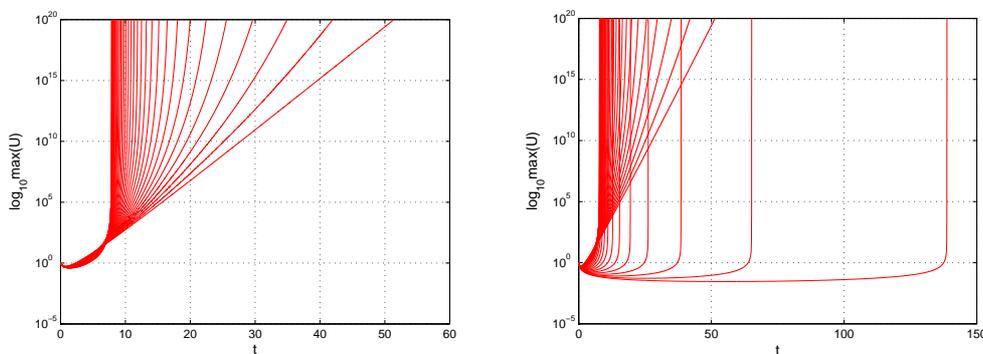


Figure 6: Maximum of  $\mathbf{U}$  in logarithmic scale, in function of  $t$ , for  $p=1:0.01:1.36$  (left), and for  $p=1.39:0.05:1.99$  (right).

We have also repeated the experiments with the method from [9], obtaining very similar results. It seems that the blow-up time can be approximated reasonably well, even with rather small  $N$  and large  $\Delta t$ . Obviously, should we want to obtain very accurate estimates of the blow-up time, larger  $N$  and smaller  $\Delta t$  would be required, especially for  $p$  very close to  $p_c$ .

## 4 Description of the numerical method in three spatial-dimensions

An important feature of the scheme that we have developed is that it can be extended to higher dimensions. More precisely, in this section, we apply it to the three-dimensional semi-linear parabolic equation

$$u_t = u_{xx} + u_{yy} + u_{zz} + u^p, \quad p > 1, \quad (x, y, z, t) \in \mathbb{R}^3 \times \mathbb{R}^+, \quad (4.1)$$

with initial data

$$u(x, y, z, 0) = u_0(x, y, z), \quad (x, y, z) \in \mathbb{R}^3. \quad (4.2)$$

Remember that, in three dimensions, the critical exponent is  $p_c = 5/3$ . To discretize (4.1), we need to approximate  $u_{xx}$ ,  $u_{yy}$  and  $u_{zz}$ . Again, using the second-order differentiation matrices  $\mathbf{D}_x^{(2)}$ ,  $\mathbf{D}_y^{(2)}$  and  $\mathbf{D}_z^{(2)}$ ,

$$\begin{aligned} u_{xx}(x_i, y_j, z_k) &\approx \mathbf{D}_x^{(2)} \square_1 \mathbf{U} = \sum_{\beta} \left[ \mathbf{D}_x^{(2)} \right]_{i\beta} U_{\beta j k}, \\ u_{yy}(x_i, y_j, z_k) &\approx \mathbf{D}_y^{(2)} \square_2 \mathbf{U} = \sum_{\beta} \left[ \mathbf{D}_y^{(2)} \right]_{j\beta} U_{i \beta k}, \\ u_{zz}(x_i, y_j, z_k) &\approx \mathbf{D}_z^{(2)} \square_3 \mathbf{U} = \sum_{\beta} \left[ \mathbf{D}_z^{(2)} \right]_{k\beta} U_{i j \beta}, \end{aligned}$$

where  $\mathbf{U} \equiv [U_{ijk}]$ ,  $U_{ijk}(t) = u(x_i, y_j, z_k, t)$  is a three-dimensional array, and  $\square_{\alpha}$ ,  $\alpha = 1, 2, 3$ , denotes the matrix-array product along the  $\alpha$  dimension of  $\mathbf{U}$  [24]. Using this approximations, the discretized version of (4.1) is

$$\mathbf{U}_t = \mathbf{D}_x^{(2)} \square_1 \mathbf{U} + \mathbf{D}_y^{(2)} \square_2 \mathbf{U} + \mathbf{D}_z^{(2)} \square_3 \mathbf{U} + \mathbf{U}^p. \quad (4.3)$$

The numerical method is exactly the same as in the two-dimensional case. The first-order semi-implicit Euler discretization in time for (4.3) is

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = \mathbf{D}_x^{(2)} \square_1 \mathbf{U}^{n+1} + \mathbf{D}_y^{(2)} \square_2 \mathbf{U}^{n+1} + \mathbf{D}_z^{(2)} \square_3 \mathbf{U}^{n+1} + (\mathbf{U}^n)^p, \quad (4.4)$$

which can be rewritten as

$$\begin{aligned} \left( \frac{1}{3} \mathbf{I}_x - \Delta t \mathbf{D}_x^{(2)} \right) \square_1 \mathbf{U}^{n+1} + \left( \frac{1}{3} \mathbf{I}_y - \Delta t \mathbf{D}_y^{(2)} \right) \square_2 \mathbf{U}^{n+1} \\ + \left( \frac{1}{3} \mathbf{I}_z - \Delta t \mathbf{D}_z^{(2)} \right) \square_3 \mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t (\mathbf{U}^n)^p, \end{aligned} \quad (4.5)$$

i.e., we get a three-dimensional Sylvester equation of the form

$$\mathbf{A}_1 \square_1 \mathbf{X} + \mathbf{A}_2 \square_2 \mathbf{X} + \mathbf{A}_3 \square_3 \mathbf{X} = \mathbf{B}. \quad (4.6)$$

Of course, if  $N_x=N_y=N_z$ , then  $\mathbf{A}=\mathbf{A}_1=\mathbf{A}_2=\mathbf{A}_3$ . In a recent paper by Li et al. [24, p. 1199], they claim to have solved this kind of equations **for the first time**. In fact, we have taken the useful notation  $\square_\alpha$  from them. On the other hand, they solve Sylvester equations of the form

$$\mathbf{A}_1 \square_1 \mathbf{X} + \mathbf{X} \square_2 \mathbf{A}_2 + \mathbf{X} \square_3 \mathbf{A}_3 = \mathbf{B}, \tag{4.7}$$

losing the symmetry of (4.6). Indeed, writing all the addends in the form of  $\mathbf{A} \square_\alpha \mathbf{X}$ ,  $\alpha = 1, \dots, n$ , allows us to extend naturally our algorithms to an arbitrary number of dimensions, reducing the difficulty of dealing with higher dimensions to just a notational problem and, hence, avoiding the curse of dimensionality.

Following the reasoning of the two-dimensional case, the eigenvalues of  $\mathbf{A}_1 = \frac{1}{3}\mathbf{I}_x - \Delta t \mathbf{D}_x^{(2)}$ ,  $\mathbf{A}_2 = \frac{1}{3}\mathbf{I}_y - \Delta t \mathbf{D}_y^{(2)}$  and  $\mathbf{A}_3 = \frac{1}{3}\mathbf{I}_z - \Delta t \mathbf{D}_z^{(2)}$  are respectively  $\lambda_i = \frac{1}{3} - \Delta t \mu_i > 0$ ,  $\lambda_j = \frac{1}{3} - \Delta t \mu_j > 0$  and  $\lambda_k = \frac{1}{3} - \Delta t \mu_k > 0$ ,  $\forall t > 0$ , which guarantees that a unique solution for (4.5) exists.

To solve (4.5), the most robust option is to compute the Schur decomposition of  $\mathbf{A}_1$ ,  $\mathbf{A}_2$  and  $\mathbf{A}_3$ , combined with a three-dimensional algorithm inspired on the Bartels-Steward algorithm [2], which is offered, for the sake of completeness, in Appendix B. This algorithm is much shorter and clearer than the one in [24] and can be easily generalized to  $N$ -dimensional problems [8].

On the other hand, as we pointed out in the two-dimensional case, if the matrices  $\mathbf{A}_\alpha$  are diagonalizable,  $\mathbf{A}_\alpha = \mathbf{P}_\alpha \cdot \mathbf{D}_\alpha \cdot \mathbf{P}_\alpha^{-1}$ , with their eigenvalue matrices  $\mathbf{P}_\alpha$  well conditioned, then it is possible to trivialize (4.6), obtaining a three-dimensional version of (2.12):

$$\mathbf{U}^{n+1} = \mathbf{P}_3 \square_3 \left[ \mathbf{P}_2 \square_2 \left[ \mathbf{P}_1 \square_1 \left[ \left[ \mathbf{P}_3^{-1} \square_3 \left[ \mathbf{P}_2^{-1} \square_2 \left[ \mathbf{P}_1^{-1} \square_1 \left[ \mathbf{U}^n + \Delta t (\mathbf{U}^n)^p \right] \right] \right] \right] \right] \right] \circ \Lambda^{-1} \right], \tag{4.8}$$

where  $\Lambda^{-1} \equiv [1/(\lambda_i + \lambda_j + \lambda_k)]$ . The computational cost is of just  $2N_x N_y N_z (N_x + N_y + N_z)$  multiplications and  $N_x N_y N_z$  divisions per time-step.

It is also immediate to adapt the fourth-order scheme (2.14) to the three-dimensional case; for the sake of brevity, we omit the details, which can be easily completed by the reader. The ideas in Section 2.1 are also valid here, when dealing with rational Chebyshev polynomials.

## 5 Numerical experiments in three dimensions

### 5.1 Numerical tests for problems with known exact solutions

As in the two dimensional case, we have considered two modified equations of  $u_t = \Delta u + u^3$ :

$$\begin{cases} u_t = \Delta u - 2 \exp(6t) [\sinh^2(x) \cosh^2(y) \cosh^2(z) \\ \quad + \cosh^2(x) \sinh^2(y) \cosh^2(z) + \cosh^2(x) \cosh^2(y) \sinh^2(z)] u^3, \\ u_0(x, y, z) = \operatorname{sech} x \operatorname{sech} y \operatorname{sech} z, \end{cases} \tag{5.1}$$

and

$$\begin{cases} u_t = \Delta u - 3\exp(6t) \left[ \left(x^2 + y^2 + z^2 + \frac{4}{3}\right)^2 - \frac{25}{9} \right] u^3, \\ u_0(x, y, z) = \frac{1}{1 + x^2 + y^2 + z^2}, \end{cases} \tag{5.2}$$

whose known exact solutions are respectively  $u(x, y, z, t) = \exp(-3t)\operatorname{sech}x\operatorname{sech}y\operatorname{sech}z$ , with exponential decay, and  $u(x, y, z, t) = \exp(-3t)/(1 + x^2 + y^2 + z^2)$ , with polynomial decay. As predicted in the theory, Hermite and sinc functions are very well suited for solutions with exponential decay; while rational Chebyshev polynomials work well in both cases, but are the obligatory choice for solutions with polynomial decay.

We have computed the numerical solution at  $t = 1$ , for  $N = N_x = N_y = N_z = 129$ . Tables 4 and 5 show the respective errors in discrete  $\mathcal{L}^\infty$ -norm at  $t = 1$ , for several  $\Delta t$ . All and each one of the comments in Section 3.1 fully apply here.

Table 4: Maximum error in  $\mathcal{L}^\infty$ -norm at  $t = 1$ ,  $\max_{ijk} |\mathbf{U}_{ijk} - u(x_i, y_j, z_k, 1)|$ , of the numerical solutions with exponential decay, for  $N_x = N_y = N_z = 129$ , and different  $\Delta t$ . The scalings are those used in Table 1.

$\Delta t^{-1}$	Hermite	Sinc	R. Cheb.
25	$2.5985 \cdot 10^{-6}$	$2.6001 \cdot 10^{-6}$	$2.5987 \cdot 10^{-6}$
50	$1.4491 \cdot 10^{-7}$	$1.5379 \cdot 10^{-7}$	$1.4496 \cdot 10^{-7}$
100	$8.6909 \cdot 10^{-9}$	$1.7938 \cdot 10^{-8}$	$8.5380 \cdot 10^{-9}$
200	$7.0123 \cdot 10^{-10}$	$9.9484 \cdot 10^{-9}$	$5.1775 \cdot 10^{-10}$
400	$2.1718 \cdot 10^{-10}$	$9.4644 \cdot 10^{-9}$	$3.1850 \cdot 10^{-11}$

Table 5: Maximum error in  $\mathcal{L}^\infty$ -norm at  $t = 1$ ,  $\max_{ijk} |\mathbf{U}_{ijk} - u(x_i, y_j, z_k, 1)|$ , of the numerical solutions with polynomial decay, for  $N_x = N_y = N_z = 129$ , and different  $\Delta t$ . The scalings are those used in Table 2.

$\Delta t^{-1}$	Hermite	Sinc	R. Cheb.
25	$2.5756 \cdot 10^{-5}$	$8.1165 \cdot 10^{-5}$	$1.5839 \cdot 10^{-6}$
50	$2.5778 \cdot 10^{-5}$	$8.1181 \cdot 10^{-5}$	$8.7469 \cdot 10^{-8}$
100	$2.5779 \cdot 10^{-5}$	$8.1183 \cdot 10^{-5}$	$5.1298 \cdot 10^{-9}$
200	$2.5779 \cdot 10^{-5}$	$8.1183 \cdot 10^{-5}$	$3.1624 \cdot 10^{-10}$
400	$2.5779 \cdot 10^{-5}$	$8.1183 \cdot 10^{-5}$	$2.3078 \cdot 10^{-10}$

### 5.2 A blow-up problem in three dimensions

We have simulated the evolution of

$$\begin{cases} u_t = u_{xx} + u_{yy} + u_{zz} + u^p, & (x, y, z, t) \in \mathbb{R}^3 \times \mathbb{R}^+, \\ u_0(x, y, z) = \exp(-x^2 - y^2 - z^2), \end{cases}$$

using Hermite functions, for  $N = 99$ ,  $L = 3$ ,  $(x_i, y_j, z_k) \in [-40.00 \dots, 40.00 \dots]^3$ ,  $\Delta t = 10^{-2}$ . As in the last example of Subsection 3.2, we have considered different  $p$ , and all the

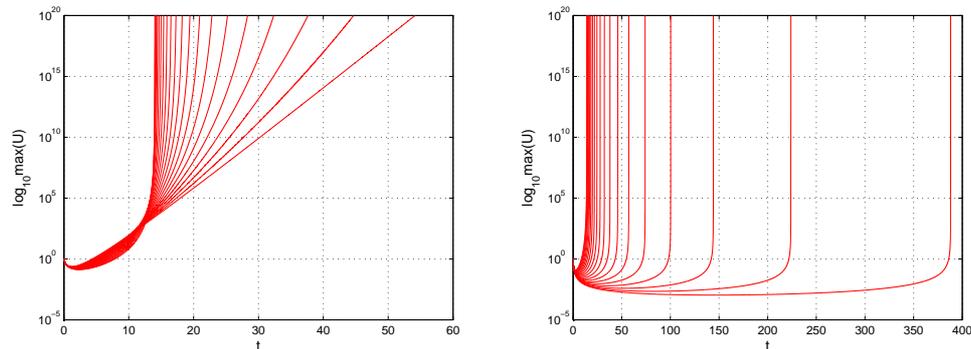


Figure 7: Maximum of  $U$  in logarithmic scale, in function of  $t$ , for  $p=1:0.01:1.2$  (left), and for  $p=1.22:0.02:1.6$  (right).

conclusions of the two-dimensional case are valid here. For  $p=1$ , there is no blow-up, but an exponential growing. Then, the blow-up time decreases as  $p$  increases, until  $p \approx 1.2$  (left hand-side of Fig. 7). On the other hand, from  $p \approx 1.2$ , the blow-up time quickly increases with  $p$ , tending to infinity as  $p$  tends to the critical exponent  $p_c = 5/3$  (right hand-side of Fig. 7). As in the two-dimensional case, for  $p$  close to 1, the blow-up time can be estimated only very roughly; while the blow-up happens very sharply for larger  $p$ .

## 6 Conclusions

In this paper, we have developed a new method to simulate numerically  $u_t = \Delta u + u^p$  in two and three spatial-dimensions, using Hermite functions, sinc functions and rational Chebyshev polynomials. On the one hand, it is theoretically interesting, because it relates several disconnected areas of applied mathematics, such as numerical spectral methods and Sylvester matrix equations, and can be implemented elegantly in an arbitrary number of dimensions; of particular interest is the pretty careful treatment of the third-dimensional case that we do, which involves the resolution of third-dimensional Sylvester equations. Indeed, solving higher-dimensional Sylvester equations is a new, yet unexplored area that might be extremely powerful in many mathematical domains.

On the other hand, the method reveals itself as a practical alternative for solutions with polynomial or exponential decay. To better justify this point, we have compared the results of the two-dimensional case with those obtained by using the method from [9], which combines a domain truncation technique with an ETDRK4 scheme; let us mention that ETDRK4 schemes can be regarded as an optimal choice for the kind of equations that we are studying. While our new method, with any of the aforementioned families of functions, competes well when exponentially fast decaying solutions are considered, it really excels for polynomially fast decaying solutions, being the rational Chebyshev polynomials the only possible choice in that case.

At this point, two important comments are needed. First, some preliminary tests, and ideally some sort of theorem, may be needed to check that the decay of the initial solution of  $u_t = \Delta u + u^p$ , or a modified version of it, does not change with time, or that the support does not expand. This is vital to choose an adequate class of functions and/or an adequate numerical method. Second, the implicit-explicit scheme (SBDF4) that we have chosen decreases the stability restrictions on  $\Delta t$  by dumping the higher frequency modes. This may be of concern when studying blow-up problems, because the high frequency modes become extremely relevant. In those cases, a very small  $\Delta t$  might be needed to track the infinitesimally fast changing dynamics, so an explicit scheme which is stable for that small  $\Delta t$  and which does not dump the highest nodes might be preferable, especially if an extremely refined grid were required. Obviously, it is always possible to use a numerical scheme for a small time-interval close to the blow-up, and another scheme for the rest of the simulation. Therefore, even in those cases when a tremendous resolution is required, our method can still be very useful.

To finish this paper, let us mention that the method can be applied, with slight modifications, to other types of nonlinear parabolic equations like, for instant, the exponential reaction model

$$u_t = \Delta u + \lambda e^u, \quad \lambda > 0,$$

which is important in combustion theory and also in other areas (see [14] and its references).

## Acknowledgments

This work was supported by MEC (Spain), with the project MTM2011-24054, and by the Basque Government, with the project IT-305-07. The authors would like to thank the referee for the thorough, constructive and helpful comments and suggestions on the manuscript.

## Appendices

### A Pseudo-spectral methods over unbounded domains

#### A.1 Hermite functions

The Hermite polynomials  $H_n(x)$  [5,29] are a very important class of orthogonal functions. They are defined by the three-term recurrence relation:

$$\begin{cases} H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), & n \geq 1, \\ H_0(x) = 1, \\ H_1(x) = 2x, \end{cases} \quad (\text{A.1})$$

and are orthogonal with respect to the weight  $\omega(x) = \exp(-x^2)$ :

$$\int_{-\infty}^{+\infty} H_m(x)H_n(x)\omega(x)dx = \delta_{mn}\sqrt{\pi}2^n n!, \tag{A.2}$$

where  $\delta_{mn}$  is the Dirac delta. Moreover, they satisfy

$$H'_n(x) = 2nH_{n-1}. \tag{A.3}$$

However, in practice, they are not very useful, due to their wild behavior at infinite. Instead, we consider the normalized Hermite functions, which are defined by

$$\psi_n(x) = \frac{1}{\pi^{1/4}\sqrt{2^n n!}}e^{-x^2/2}H_n(x); \tag{A.4}$$

then

$$\int_{-\infty}^{+\infty} \psi_m(x)\psi_n(x)dx = \delta_{mn}. \tag{A.5}$$

Moreover, from (A.1), we obtain the recurrence relation

$$\begin{cases} \psi_{n+1}(x) = \sqrt{\frac{2}{n+1}}x\psi_n(x) - \sqrt{\frac{n}{n+1}}\psi_{n-1}(x), & n \geq 1, \\ \psi_0(x) = \pi^{-1/4}e^{-x^2/2}, \\ \psi_1(x) = \sqrt{2}x\psi_0(x). \end{cases} \tag{A.6}$$

Furthermore, from this last expression, together with (A.3), we obtain recurrence relations also for  $\psi'(x)$  and  $\psi''(x)$ :

$$\begin{cases} \psi'_{n+1}(x) = -x\psi_{n+1}(x) + \sqrt{2(n+1)}\psi_n(x), & n \geq 0, \\ \psi'_0(x) = -x\psi_0(x), \end{cases} \tag{A.7}$$

and

$$\begin{cases} \psi''_{n+1}(x) = (x^2-1)\psi_{n+1}(x) - 2\sqrt{2(n+1)}x\psi_n(x) + 2\sqrt{n(n+1)}\psi_{n-1}(x), & n \geq 1, \\ \psi''_0(x) = (x^2-1)\psi_0(x), \\ \psi''_1(x) = (x^2-1)\psi_1(x) - 2\sqrt{2}x\psi_0(x). \end{cases} \tag{A.8}$$

The optimal pseudo-spectral points are the roots of the  $N+1$ -th Hermite function  $H_{N+1}$ , which are also the eigenvalues of the following symmetric tridiagonal matrix:

$$A_{N+1} = \begin{pmatrix} 0 & \sqrt{\frac{1}{2}} & & & & & \\ \sqrt{\frac{1}{2}} & 0 & \sqrt{\frac{2}{2}} & & & & \\ & \sqrt{\frac{2}{2}} & 0 & \sqrt{\frac{3}{2}} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \sqrt{\frac{N-1}{2}} & 0 & \sqrt{\frac{N}{2}} & \\ & & & & \sqrt{\frac{N}{2}} & 0 & \end{pmatrix}. \tag{A.9}$$

The grid spacing varies slightly, but is almost uniform over the interval. Moreover,

$$\max_j |x_j| \sim \sqrt{2N}, \quad \min_j |x_j - x_{j-1}| \sim N^{-1/2}. \tag{A.10}$$

The first and second-order Hermite differentiation matrices  $\mathbf{D}^{(1)}$ ,  $\mathbf{D}^{(2)}$  can be calculated through polynomial interpolation, as in [35]. Although they do have very nice properties, for instance their spectral radii are respectively  $\rho(\mathbf{D}^{(1)}) = \mathcal{O}(\sqrt{N})$  and  $\rho(\mathbf{D}^{(2)}) = \mathcal{O}(N)$ , their numerical computation seems to be by no means trivial. Indeed, the codes associated to [35] seems not to work with more than  $246+1$  grid points. On the other hand, we have been able to obtain correct Hermite differentiation matrices for as many as  $764+1$  grid points using a different approach: Given  $N+1$  points  $x_0, \dots, x_N$ , find the only matrices  $\mathbf{D}^{(1)}$  and  $\mathbf{D}^{(2)}$  that satisfy:

$$\begin{pmatrix} \psi'_0(x_0) & \cdots & \psi'_N(x_0) \\ \vdots & \ddots & \vdots \\ \psi'_0(x_N) & \cdots & \psi'_N(x_N) \end{pmatrix} = \mathbf{D}^{(1)} \cdot \begin{pmatrix} \psi_0(x_0) & \cdots & \psi_N(x_0) \\ \vdots & \ddots & \vdots \\ \psi_0(x_N) & \cdots & \psi_N(x_N) \end{pmatrix}, \tag{A.11a}$$

$$\begin{pmatrix} \psi''_0(x_0) & \cdots & \psi''_N(x_0) \\ \vdots & \ddots & \vdots \\ \psi''_0(x_N) & \cdots & \psi''_N(x_N) \end{pmatrix} = \mathbf{D}^{(2)} \cdot \begin{pmatrix} \psi_0(x_0) & \cdots & \psi_N(x_0) \\ \vdots & \ddots & \vdots \\ \psi_0(x_N) & \cdots & \psi_N(x_N) \end{pmatrix}, \tag{A.11b}$$

respectively, i.e., those matrices that differentiate correctly all the Hermite functions  $\psi_k(x)$  at the grid points,  $\forall k \leq N$ . Of course, if we scale  $x$  to  $Lx$ , then the matrices are scaled accordingly to  $\mathbf{D}^{(1)}/L$  and  $\mathbf{D}^{(2)}/L^2$ . Finally, let us mention that larger differentiation matrices could be obtained without problems by using a multiple-precision library like MPFR [12].

### A.2 Sinc functions

A function  $f(x)$ , analytic  $\forall |x| < \infty$  and decaying exponentially along the real axis as  $|x| \rightarrow \infty$ , can be conveniently approximated as sums of the Whittaker cardinal function, better known as the sinc function [4, 30, 31]:

$$f(x) \approx \sum_{j=-N/2}^{N/2} f(x_j) C_j(x), \tag{A.12}$$

where

$$C_j(x;h) \equiv \text{sinc}\left(\frac{x-jh}{h}\right), \quad \text{sinc}(x) \equiv \frac{\sin(\pi x)}{\pi x}. \tag{A.13}$$

As in all the pseudo-spectral methods over unbounded domains, there are two parameters; in this case, the number of grid points  $N$ , and the grid spacing  $h$ , which is the

equivalent to the scaling  $L$  appearing in the Hermite and rational Chebyshev polynomials. It can be shown [5, p. 344] that, for most situations, best results are obtained by choosing  $h$  proportional to  $\sqrt{N}$ .

The sinc expansion is the easiest pseudo-spectral method to program and understand, and has an enormous range of applications. Its corresponding differential matrices have a Toeplitz structure and, unlike their Hermite equivalents, can be constructed numerically with no issues  $\forall N > 0$ . In our numerical experiments, we have taken the implementation from [35].

### A.3 Rational Chebyshev polynomials

By using a transformation that maps an infinite interval into a finite domain, it is possible to generate a great variety of new basis sets for the infinite interval that are the images under the change-of-coordinate of Chebyshev polynomials or Fourier series [5, p. 355]. Although an infinite variety of maps is possible, we will concentrate on an extremely important one, the so call algebraic map (see for instance [4, 16]):

$$x = \frac{Lp}{\sqrt{1-p^2}} \iff p = \frac{x}{\sqrt{L+x^2}}, \tag{A.14}$$

which maps  $p \in [-1,1]$  into the whole real line  $x \in \mathbb{R}$ ; in what follows, we will assume  $L=1$ . Moreover, given the Chebyshev polynomials

$$T_k(p) = \cos(n \arccos(p)), \tag{A.15}$$

it allows to define the so-called rational Chebyshev Polynomials

$$TB_k(x) = T_k\left(\frac{x}{\sqrt{1+x^2}}\right), \quad x \in \mathbb{R}, \tag{A.16}$$

which form an orthogonal basis in  $\mathbb{R}$ :

$$\int_{-\infty}^{+\infty} \frac{TB_m(x)TB_n(x)}{1+x^2} dx = \begin{cases} \pi/2, & m=n > 0, \\ \pi, & m=n=0, \\ 0, & m \neq n. \end{cases} \tag{A.17}$$

However, the easiest way to program them is to use a trigonometric representation, i.e., through the change of variable

$$p = \cos(s) \iff x = \cot(s), \quad s \in [0, \pi]. \tag{A.18}$$

Then, it is not difficult to check that

$$u_x = (1-p^2)^{3/2} u_p, \quad u_{xx} = -3p(1-p^2)^2 u_p + (1-p^2)^3 u_{pp}, \tag{A.19a}$$

$$u_x = -\sin^2(s) u_s, \quad u_{xx} = 2\sin^3(s) \cos(s) u_s + \sin^4(s) u_{ss}. \tag{A.19b}$$

The first formula allows us to obtain the differentiation matrices associated to the rational Chebyshev polynomials. Indeed, if  $\mathbf{D}_{Cheb}^{(1)}$  and  $\mathbf{D}_{Cheb}^{(2)}$  denote respectively the first and second-order Chebyshev differentiation matrices, their rational Chebyshev counterparts  $\mathbf{D}^{(1)}$  and  $\mathbf{D}^{(2)}$  are given by

$$\mathbf{D}^{(1)} = \text{diag}((1-p^2)^{3/2}) \cdot \mathbf{D}_{Cheb}^{(1)}, \tag{A.20a}$$

$$\mathbf{D}^{(2)} = -3\text{diag}(p(1-p^2)^2) \cdot \mathbf{D}_{Cheb}^{(1)} + \text{diag}((1-p^2)^3) \cdot \mathbf{D}_{Cheb}^{(2)}, \tag{A.20b}$$

where, for example,  $\text{diag}((1-p^2)^{3/2})$  is the diagonal matrix whose diagonal elements are  $(1-p_j^2)^{3/2}$ , being  $p_j \in [-1,1]$  the collocation points, etc.

However, unlike with Hermite polynomials and sinc functions, it is much compact and elegant to consider rational Chebyshev differentiation matrices on the Fourier space, instead of the real space. To obtain them, we consider a function expressed as a linear combination of rational Chebyshev polynomials,

$$u(x) = \sum_{k=0}^N a_k TB_k(x) = \sum_{k=0}^N a_k \cos(ks), \tag{A.21}$$

and, using (A.19b), express the coefficients  $b_k$  of the development of  $u_{xx}(x)$  as a linear transform of the coefficients  $a_k$  of  $u(x)$ :

$$\begin{aligned} u_{xx} &= 2\sin^3(s)\cos(s)u_s + \sin^4(s)u_{ss} \\ &= -\sum_{k=0}^N a_k \left[ k^2 \sin^4(s)\cos(ks) + 2k \sin^3(s)\cos(s)\sin(ks) \right]. \end{aligned} \tag{A.22}$$

Bearing in mind the identities

$$\begin{aligned} \sin^4(s)\cos(ks) &\equiv \frac{1}{16}\cos((k-4)s) - \frac{1}{4}\cos((k-2)s) + \frac{3}{8}\cos(ks) \\ &\quad - \frac{1}{4}\cos((k+2)s) + \frac{1}{16}\cos((k+4)s), \\ \sin^3(s)\cos(s)\sin(ks) &\equiv -\frac{1}{16}\cos((k-4)s) + \frac{1}{8}\cos((k-2)s) \\ &\quad - \frac{1}{8}\cos((k+2)s) + \frac{1}{16}\cos((k+4)s), \end{aligned}$$

we finally get

$$\begin{aligned} u_{xx} &= \sum_{k=0}^N a_k \left[ \frac{-k^2+2k}{16}\cos((k-4)s) + \frac{k^2-k}{4}\cos((k-2)s) - \frac{3k^2}{8}\cos(ks) \right. \\ &\quad \left. + \frac{k^2+k}{4}\cos((k+2)s) - \frac{k^2+2k}{16}\cos((k+4)s) \right] \\ &= \sum_{k=0}^{N+4} b_k \cos(ks) = \sum_{k=0}^{N+4} b_k TB_k(x), \end{aligned} \tag{A.23}$$

where, obviously,  $\cos(-2s) = \cos(2s)$  and  $\cos(-4s) = \cos(4s)$ . Observe that  $b_0 = 0$  always, and that  $u_{xx}$  requires four more Chebyshev rational polynomials to be exactly expressed, namely  $T_{B_{N+1}}(x)$ ,  $T_{B_{N+2}}(x)$ ,  $T_{B_{N+3}}(x)$  and  $T_{B_{N+4}}(x)$ . Since they are orthogonal to  $T_{B_k}(x)$ ,  $0 \leq k \leq N$ , an easy (and numerically safe) way is just to ignore them, setting  $b_{N+1} = b_{N+2} = N+3 = N+4 = 0$ . Thus, we obtain a sparse  $(N+1) \times (N+1)$ -matrix  $\mathbf{D}^{(2)}$  that maps  $\{a_0, \dots, a_N\}$  into  $\{b_0, \dots, b_N\}$ ; furthermore, doing an even-odd uncoupling,  $\mathbf{D}^{(2)}$  can be decomposed into two pentadiagonal matrices. On the other hand, the collocation differential matrix (A.20b) is full and with no structure. Finally, for both types of matrices, if we scale  $x$  to  $Lx$ , they are scaled accordingly to  $\mathbf{D}^{(1)}/L$  and  $\mathbf{D}^{(2)}/L^2$ .

Given a function  $u(x)$ , to obtain their rational Chebyshev coefficients  $a_k$ ,  $0 \leq k \leq N$ , as in (A.21), we need to evaluate it on  $x_j = \cot(s_j)$ , where  $s_j = \frac{\pi}{2(N+1)} + \frac{j\pi}{N+1}$ , for  $0 \leq j \leq N$ ; observe that the smallest distance between two adjacent  $x_j$  is  $\mathcal{O}(1/N)$ . It is straightforward to transform  $\{u(x_j)\}$  into  $\{a_k\}$  and vice-versa by means of a shifted discrete cosine transform.

## B Solving a three-dimensional Sylvester equation

We may think of different approaches to solve the three-dimensional Sylvester equation

$$\mathbf{A}_1 \square_1 \mathbf{X} + \mathbf{A}_2 \square_2 \mathbf{X} + \mathbf{A}_3 \square_3 \mathbf{X} = \mathbf{B}, \tag{B.1}$$

where  $\mathbf{A}_\alpha \in \mathcal{M}_{N_\alpha \times N_\alpha}(\mathbb{C})$ ,  $\alpha = 1, 2, 3$ , are squared matrices, and  $\mathbf{B}, \mathbf{X} \in \mathcal{M}_{N_1 \times N_2 \times N_3}(\mathbb{C})$  are multi-dimensional arrays. On the one hand, the simplest (but completely unfeasible) approach is to apply Kronecker tensor sums, together with the vec operator, to (B.1), getting a huge linear system  $(\mathbf{A}_1 \oplus \mathbf{A}_2 \oplus \mathbf{A}_3) \cdot \text{vec}(\mathbf{X}) = \text{vec}(\mathbf{B})$ , where  $(\mathbf{A}_1 \oplus \mathbf{A}_2 \oplus \mathbf{A}_3)$  is a matrix with  $\mathcal{O}(N_1^2 N_2^2 N_3^2)$  elements! On the other hand, we may generalize Krylov-subspace methods in two dimensions [21], to obtain iterative methods that would be computationally extremely expensive in three dimensions. Therefore, the best option seems to use some sort of decomposition for  $\mathbf{A}_\alpha$ . If, for instance,  $\mathbf{A}_\alpha = \mathbf{P}_\alpha \cdot \mathbf{D}_\alpha \cdot \mathbf{P}_\alpha^{-1}$  are diagonalizable matrices, with  $\mathbf{P}_\alpha$  well conditioned, then (B.1) is reduced to solving an equation of the type

$$\mathbf{D}_1 \square_1 \mathbf{Y} + \mathbf{D}_2 \square_2 \mathbf{Y} + \mathbf{D}_3 \square_3 \mathbf{Y} = \mathbf{C}, \tag{B.2}$$

whose solution can be computed trivially. Since all the matrices  $\mathbf{A}_\alpha$  that appear in the numerical experiments in this paper are diagonalizable with well-conditioned eigenvalue matrices, we have been able to implement safely this idea, to build extremely efficient schemes like (4.8). On the other hand, in [10], we have applied a related idea for implementing an efficient scheme for the  $N$ -dimensional sine-Gordon equation.

Another approach is to obtain the Schur decompositions of  $\mathbf{A}_\alpha = \mathbf{U}_\alpha \cdot \mathbf{T}_\alpha \cdot \mathbf{U}_\alpha^*$ , where  $\mathbf{T}_\alpha$  are upper triangular, and  $\mathbf{U}_\alpha^* \cdot \mathbf{U}_\alpha = \mathbf{I}$  are unitary matrices i.e., their inverse coincides with their conjugate transpose. Although this is computationally more expensive, it is also much more robust, because we can consider also non-diagonalizable matrices  $\mathbf{A}_\alpha$ ; and

$U_\alpha$  are well conditioned by construction. Indeed, all the two-dimensional algorithms are based on the Bartels-Steward algorithm [2], which also employs Schur decompositions. Hence, unless extra information on  $A_\alpha$  is provided, this approach can be labelled as standard.

In what follows, we use complex Schur decompositions, which can be applied both to real and complex matrices  $A_\alpha$ . Introducing the Schur decompositions of  $A_\alpha$  into (B.1), our problem is reduced to solving an equation of the form

$$T_1 \square_1 Y + T_2 \square_2 Y + T_3 \square_3 Y = C. \tag{B.3}$$

If we expand (B.3), bearing in mind that  $T_\alpha$  are upper triangular, we get

$$\begin{aligned} C_{i_1 i_2 i_3} &= \sum_{k_1=1}^{N_1} T_{1,i_1 k_1} Y_{k_1 i_2 i_3} + \sum_{k_2=1}^{N_2} T_{2,i_2 k_2} Y_{i_1 k_2 i_3} + \sum_{k_3=1}^{N_3} T_{3,i_3 k_3} Y_{i_1 i_2 k_3} \\ &= \sum_{k_1=i_1}^{N_1} T_{1,i_1 k_1} Y_{k_1 i_2 i_3} + \sum_{k_2=i_2}^{N_2} T_{2,i_2 k_2} Y_{i_1 k_2 i_3} + \sum_{k_3=i_3}^{N_3} T_{3,i_3 k_3} Y_{i_1 i_2 k_3}. \end{aligned}$$

The central idea when solving (B.3) is to write separately  $Y_{i_1 i_2 i_3}$ :

$$\begin{aligned} C_{i_1 i_2 i_3} &= (T_{1,i_1 i_1} + T_{2,i_2 i_2} + T_{3,i_3 i_3}) Y_{i_1 i_2 i_3} + \sum_{k_1=i_1+1}^{N_1} T_{1,i_1 k_1} Y_{k_1 i_2 i_3} \\ &+ \sum_{k_2=i_2+1}^{N_2} T_{2,i_2 k_2} Y_{i_1 k_2 i_3} + \sum_{k_3=i_3+1}^{N_3} T_{3,i_3 k_3} Y_{i_1 i_2 k_3}. \end{aligned} \tag{B.4}$$

Observe that  $T_{\alpha,i_\alpha i_\alpha}$  are precisely the eigenvalues of  $A_\alpha$ , so that the last formula proves that (B.1) has a solution if and only the sum of any eigenvalue of  $A_1$  plus any eigenvalue of  $A_2$  plus any eigenvalue of  $A_3$  is always non-zero. From the last formula, it is also straightforward to compute  $Y_{N_1 N_2 N_3}$ :

$$Y_{N_1 N_2 N_3} = \frac{C_{i_1 i_2 i_3}}{T_{1,i_1 i_1} + T_{2,i_2 i_2} + T_{3,i_3 i_3}}. \tag{B.5}$$

Bearing in mind that to obtain one  $Y_{i_1 i_2 i_3}$ , only the elements  $Y_{i_1+1:N_1, i_2 i_3}$ ,  $Y_{i_1, i_2+1:N_2, i_3}$  and  $Y_{i_1 i_2, i_3+1:N_3}$  are needed, we can compute all the  $Y_{i_1 i_2 i_3}$  recursively:

$$\begin{aligned} Y_{i_1 i_2 i_3} &= \frac{1}{T_{1,i_1 i_1} + T_{2,i_2 i_2} + T_{3,i_3 i_3}} \left[ C_{i_1 i_2 i_3} - \sum_{k_1=i_1+1}^{N_1} T_{1,i_1 k_1} Y_{k_1 i_2 i_3} \right. \\ &\quad \left. - \sum_{k_2=i_2+1}^{N_2} T_{2,i_2 k_2} Y_{i_1 k_2 i_3} - \sum_{k_3=i_3+1}^{N_3} T_{3,i_3 k_3} Y_{i_1 i_2 k_3} \right]. \end{aligned} \tag{B.6}$$

Of course, a convenient ordering of the indices, as well as vectorizing the code as much as possible, is vital when implementing (B.6) on a computer in an efficient way. Moreover,

if we are not interested in  $\mathbf{C}$ , it can be progressively overridden with the values of the solution  $\mathbf{Y}$ , so no additional storage is required. We refer the reader to [8] for all the details.

It is not difficult to check that (B.6) requires  $N_1 N_2 N_3 (N_1 + N_2 + N_3 - 3)/2$  multiplications and  $N_1 N_2 N_3$  divisions, compared with no multiplications and just  $N_1 N_2 N_3$  divisions required to solve (B.2), which, additionally, can be done in parallel, further increasing the speed. Besides,  $N_1 N_2 N_3 (N_1 + N_2 + N_3)$  multiplications are required to transform (B.1) into (B.3), and another  $N_1 N_2 N_3 (N_1 + N_2 + N_3)$  multiplications are required to recover the solution  $\mathbf{X}$  from  $\mathbf{Y}$ . Altogether, the algorithm needs  $N_1 N_2 N_3 (5N_1 + 5N_2 + 5N_3 - 3)/2$  multiplications and  $N_1 N_2 N_3$  divisions, compared with the  $2N_1 N_2 N_3 (N_1 + N_2 + N_3)$  operations and  $N_1 N_2 N_3$  divisions required when working with (B.2). Obviously, if  $\mathbf{A}_\alpha$  are time-independent, their Schur decomposition needs to be done just once.

## References

- [1] U.M. Ascher, S.J. Ruuth, and B.T.R. Wetton, Implicit-Explicit Methods for Time-Dependent Partial Differential Equations, *SIAM Journal on Numerical Analysis* 32 (1995), no. 3, 797–823.
- [2] R.H. Bartels and G.W. Stewart, Solution of the Matrix Equation  $AX + XB = C$ , *Communications of the ACM* 15 (1972), no. 9, 820–826.
- [3] J.P. Boyd, The Optimization of Convergence for Chebyshev Polynomial Methods in an Unbounded Domain, *J. Comput. Phys.* 45 (1982), no. 1, 43–79.
- [4] J.P. Boyd, Spectral Methods Using Rational Basic Functions on an Infinite Interval, *J. Comput. Phys.* 69 (1987), no. 1, 112–142.
- [5] J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, second (revised) ed., Dover, 2001.
- [6] C. Canuto, M.Y. Hussain, A. Quarteroni, and T.A. Zang, *Spectral Methods in Fluid Dynamics*, Springer, 1988.
- [7] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang, *Spectral Methods: Fundamentals in Single Domains*, Springer, 2007.
- [8] F. de la Hoz, A Schur-based algorithm for solving  $N$ -dimensional matrix equations, preprint.
- [9] F. de la Hoz and F. Vadillo, A Numerical Simulation for the Blow-up of Semi-linear Diffusion Equations, *Int. J. Comput. Math.* 86 (2009), no. 3, 493–502.
- [10] F. de la Hoz and F. Vadillo, Numerical simulation of the  $N$ -dimensional sine-Gordon equation via operational matrices, *Computer Physics Communications* 183 (2012), no. 4, 864–879.
- [11] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, 1998.
- [12] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélicier, and P. Zimmermann, Mpf: A multiple-precision binary floating-point library with correct rounding, *ACM Transactions on Mathematical Software (TOMS)* 33 (2007), no. 2, Article no. 13.
- [13] H. Fujita, On the blowing up of solutions of the Cauchy problem for  $u_t = \Delta u + u^{1+\alpha}$ , *J Fac Sci Univ Tokyo Sect I* 13 (1966), 109–124.
- [14] V.A. Galaktionov and J.L. Vázquez, The problem of blow-up in nonlinear parabolic equations, *Discrete and Continuous Dynamical Systems* 8 (2002), no. 2, 339–433.
- [15] D. Gottlieb and S.A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, 1977.

- [16] C.E. Grosch and S.T. Orszag, Numerical Solution of Problems in Unbounded Regions: Coordinate Transforms, *J. Comput. Phys.* 25 (1977), no. 3, 273–295.
- [17] D.B. Haidvogel and T. Zang, The Accurate Solution of Poisson's Equation by Expansion in Chebyshev Polynomials, *J. Comput. Phys.* 30 (1979), no. 2, 167–180.
- [18] M.A. Herrero and J.J.L. Velázquez, Blow-up behavior of one-dimensional semilinear parabolic equations, *Ann. Inst. Henri Poincaré* 10 (1993), no. 2, 131–189.
- [19] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, 1996.
- [20] N.J. Higham, *Functions of Matrices. Theory and Computation*, SIAM, 2008.
- [21] D.Y. Hu and L. Reichel, Krylov-Subspace Methods for the Sylvester Equation, *Linear Algebra and its Applications* 172 (1992), no. 15, 283–313.
- [22] A.-K. Kassam and L.N. Trefethen, Fourth-Order Time-Stepping for Stiff PDEs, *SIAM J. Sci. Comput* 26 (2005), no. 4, 1214–1233.
- [23] H.A. Levine, The Role of Critical Exponents in Blowup Theorems, *SIAM Review* 32 (1990), no. 2, 262–288.
- [24] Ben-Wen Li, Shuai Tian, Ya-Song Sun, and Zhang-Mao Hu, Schur-decomposition for 3D matrix equations and its application in solving radiative discrete ordinates equations discretized by Chebyshev collocation spectral method, *Journal of Computational Physics* 229 (2010), no. 4, 1198–1212.
- [25] Heping Ma, Weiwei Sun, and Tao Tang, Hermite Spectral Methods with a Time-Dependent Scaling for Parabolic Equations in Unbounded Domains, *SIAM Journal on Numerical Analysis* 43 (2006), no. 1, 58–75.
- [26] C.D.M. Martin and C.F. Van Loan, Solving real lineal systems with the complex Schur decomposition, *SIAM J. of Matrix Analy. and Appl.* 29 (2006), no. 1, 177–183.
- [27] D.W. Peaceman and H.H. Rachford, Jr., The Numerical Solution of Parabolic and Elliptic Differential Equations, *Journal of the Society for Industrial and Applied Mathematics* 3 (1955), no. 1, 28–41.
- [28] L.F. Richardson, The deferred approach to the limit, I-single lattice, *Trans. of the Roy. Soc. of London, Series A* 226 (1927), 299–349.
- [29] Jie Shen, Tao Tang, and Li-Lian Wang, *Spectral Methods. Algorithms, Analysis and Applications*, Springer, 2011.
- [30] F. Stenger, *Numerical Methods Based on Sinc and Analytic Functions*, Springer, 1993.
- [31] F. Stenger, *Handbook of Sinc Numerical Methods*, Chapman & Hall/CRC, 2011.
- [32] L.N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.
- [33] J.J.L. Velázquez, Classification of singularities for blowing up solutions in higher dimensions, *Trans. Amer. Math. Soc.* 338 (1993), no. 1, 441–464.
- [34] J.A.C. Weideman, The eigenvalues of Hermite and rational spectral differentiation matrices, *Numer. Math.* 61 (1992).
- [35] J.A.C. Weideman and S.C. Reddy, A MATLAB Differentiation Matrix Suite, *ACM Transactions On Mathematical Software* 26 (2000), no. 4, 465–519.