Monografías Matemáticas *García de Galdeano* **43**, 85–95 (2024)

EFFICIENT NUMERICAL ALGORITHMS FOR SEMILINEAR SINGULARLY PERTURBED CONVECTION-DIFFUSION SYSTEMS

Carmelo Clavero & Juan Carlos Jorge

Abstract. In this work we introduce a technique to develop robust and efficient numerical methods for solving semilinear parabolic singularly perturbed systems which are coupled by suitable nonlinear reaction terms. We pay special attention to systems where the small diffusion parameters have different orders of magnitude; this feature provokes that overlapping boundary layers appear in their solutions close to the outflow boundary. Our proposal combines a linearized version of the fractional implicit Euler method together with a splitting by components, to discretize in time, and the upwind finite difference scheme defined on appropriate piecewise uniform meshes of Shishkin type, to discretize in space. It is proven and checked that the proposed numerical algorithms are uniformly convergent.

Keywords: singular perturbation problems, uniform convergence, splitting methods, Shishkin meshes, computational cost.

AMS classification: 65M06, 65M12, 65M50.

§1. Introduction

In this paper we deal with the numerical solving of one dimensional semilinear parabolic singularly perturbed convection-diffusion systems, coupled by nonlinear reaction terms, given by

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t}(x,t) + \mathcal{L}_{x,\varepsilon}(t)\mathbf{u}(x,t) + \mathcal{A}(x,t,\mathbf{u}) = \mathbf{0}, \ (x,t) \in Q, \\ \mathbf{u}(0,t) = \mathbf{g}_0(t), \\ \mathbf{u}(1,t) = \mathbf{g}_1(t), \ t \in [0,T], \\ \mathbf{u}(x,0) = \boldsymbol{\varphi}(x), \ x \in (0,1), \end{cases} \tag{1}$$

where $Q \equiv (0,1) \times (0,T]$, the spatial differential operator $\mathcal{L}_{x,\varepsilon}(t)$ is defined as

$$\mathcal{L}_{x,\varepsilon}(t)\mathbf{u} \equiv -\mathcal{D}_\varepsilon \frac{\partial^2 \mathbf{u}}{\partial x^2} + \mathcal{B}(x)\frac{\partial \mathbf{u}}{\partial x}, \tag{2}$$

with $\mathbf{u} = (u_1, u_2)^T$, the diffusion matrix is $\mathcal{D}_\varepsilon = \mathrm{diag}(\varepsilon_1, \varepsilon_2)$, the convection matrix is $\mathcal{B}(x) = \mathrm{diag}(b_{11}(x), b_{22}(x))$, the nonlinear reaction terms are $\mathcal{A}(x,t,\mathbf{u}) = (a_1(x,t,\mathbf{u}), a_2(x,t,\mathbf{u}))^T$, the boundary conditions are $\mathbf{g}_0(t) = (g_{1,0}(t), g_{2,0}(t))^T$, $\mathbf{g}_1(t) = (g_{1,1}(t), g_{2,1}/t))^T$ and the initial condition is $\boldsymbol{\varphi}(x) = (\varphi_1(x), \varphi_2(x))^T$.

We assume that the diffusion parameters $\varepsilon_k,\ k = 1, 2$, can be very small and, in general, they can have different order of magnitude. We will suppose that the equations of (1) have been ordered in such a way that $0 < \varepsilon_1 \leq \varepsilon_2 \leq 1$; as well, we assume that the coefficients of the convection matrix satisfy $b_{kk}(x) \geq \beta > 0,\ k = 1, 2$, and that the components of the reaction term $\mathcal{A}$ satisfy

$$\begin{aligned} &L \geq \frac{\partial a_k}{\partial u_k}(x,t,\mathbf{v}) \geq 0,\ -L \leq \frac{\partial a_k}{\partial u_j}(x,t,\mathbf{v}) \leq 0,\ k \neq j,\ k,j = 1,2,\ \forall\, \mathbf{v} \in \mathbf{R}^2 \text{ and} \\ &\min_{\mathbf{v}\in\mathbb{R}^2} \sum_{j=1}^{n} \frac{\partial a_k}{\partial u_j}(x,t,\mathbf{v}) \geq 0,\ k = 1,2. \end{aligned} \tag{3}$$

Singularly perturbed convection-diffusion coupled systems model many physical phenomena like, for instance, convection-diffusion enzyme models, tubular models in chemical reactor theory or neutron transport problems with diffusion coefficients (see [6]). The same type of problems as (1) were considered in some previous papers (see for instance [2, 7, 5]); in those papers a numerical uniformly convergent method was constructed. In this work, we develop a numerical method following similar ideas to those ones in [2], where the main improvement is focused on reducing the computational cost of the algorithm, which is even more remarkable when nonlinear problems are considered.

Henceforth, C denotes a generic positive constant independent of the diffusion parameters $\varepsilon_k,\ k = 1, 2$ and the discretization parameters N and M.

§2. Asymptotic behavior of the exact solution

In this section we describe briefly the behavior of the solution of (1) and its derivatives, paying special attention to the overlapping boundary layers which appear in the outflow boundary $x = 1$. This knowledge is essential to construct appropriate special meshes which are capable of capturing a reliable approximation of such exact solution, even in the areas of rapid variation for it. The key to do that consists of rewriting the continuous problem as a linear convection diffusion problem of the form

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t}(x,t) - \mathcal{D}_\varepsilon \dfrac{\partial^2 \mathbf{u}}{\partial x^2}(x,t) + \mathcal{B}(x)\dfrac{\partial \mathbf{u}}{\partial x} + \hat{\mathcal{A}}(x,t)\mathbf{u} = -\mathcal{A}(x,t,\mathbf{0}),\ (x,t) \in Q, \\ \mathbf{u}(0,t) = \mathbf{g}_0(t),\ t \in (0,T], \\ \mathbf{u}(1,t) = \mathbf{g}_1(t),\ t \in (0,T], \\ \mathbf{u}(x,0) = \boldsymbol{\varphi}(x),\ x \in [0,1], \end{cases} \tag{4}$$

where $\hat{\mathcal{A}}(x,t)$ is an adequate reaction matrix. Using this fact, we can reproduce the analysis made in [2] to know in detail the behavior of the exact solution as well as those of its derivatives which are involved in the analysis of the uniform convergence of the numerical method. Concretely, we obtain the following estimates for the partial derivatives of the two

components of the exact solution

$$\begin{aligned}
&\left|\frac{\partial^k u_j}{\partial t^k}\right| \le C,\ k=0,1,2,\ j=1,2,\\
&\left|\frac{\partial^k u_1}{\partial x^k}\right| \le C\left(1+\varepsilon_1^{-k}B_{\varepsilon_1}(x)+\varepsilon_2^{-k+1}B_{\varepsilon_2}(x)\right),\ k=1,2,\\
&\left|\frac{\partial^3 u_1}{\partial x^3}\right| \le C\left(\varepsilon_1^{-1}+\varepsilon_1^{-3}B_{\varepsilon_1}(x)+\varepsilon_2^{-2}B_{\varepsilon_2}(x)\right),\\
&\left|\frac{\partial^k u_2}{\partial x^k}\right| \le C\left(1+\varepsilon_2^{-k}B_{\varepsilon_2}(x)\right),\ k=1,2,\\
&\left|\frac{\partial^3 u_2}{\partial x^3}\right| \le C\varepsilon_2^{-1}\left(1+\varepsilon_1^{-1}B_{\varepsilon_1}(x)+\varepsilon_2^{-2}B_{\varepsilon_2}(x)\right),
\end{aligned} \tag{5}$$

being $B_\gamma(x)=e^{-\frac{\beta(1-x)}{\gamma}}$.

§3. The fully discrete scheme: uniform convergence

To discretize in space problem (1), we use the classical upwind finite difference scheme on an adequate piecewise uniform mesh of Shishkin type, $\overline{\Omega}^N \equiv \{0=x_0<x_1<\ldots<x_N=1\}$, which is defined as follows (see [4]). Let N be a positive integer multiple of 3. From previous section we know that the exact solution has overlapping regular boundary layers at $x=1$; then, we define the transition parameters

$$\sigma_{\varepsilon_2}=\min\{2/3,\sigma_0\varepsilon_2\ln N\},\quad \sigma_{\varepsilon_1}=\min\{\sigma_{\varepsilon_2}/2,\sigma_0\varepsilon_1\ln N\}, \tag{6}$$

which separate the coarse and the fine mesh, where σ_0 is a constant to be fixed later, and the grid points are given by

$$x_i=\begin{cases} iH, & i=0,\ldots,N/3,\\ x_{N/3}+(i-N/3)h_{\varepsilon_2}, & i=N/3+1,\ldots,2N/3,\\ x_{2N/3}+(i-2N/3)h_{\varepsilon_1}, & i=2N/3+1,\ldots,N,\end{cases} \tag{7}$$

with $H=3(1-\sigma_{\varepsilon_2})/N$, $h_{\varepsilon_2}=3(\sigma_{\varepsilon_2}-\sigma_{\varepsilon_1})/N$, $h_{\varepsilon_1}=3\sigma_{\varepsilon_1}/N$. We denote by $h_i=x_i-x_{i-1}$, $i=1,\ldots,N$, and $\overline{h}_i=(h_i+h_{i+1})/2$, $i=1,\ldots,N-1$.

Let us denote by Ω^N the subgrid of $\overline{\Omega}^N$ composed only by the interior points of it, i.e., by $\overline{\Omega}^N\bigcap\Omega$, by $[\mathbf{v}]_{\Omega^N}$ (analogously $[v]_{\Omega^N}$ for scalar functions) the restriction operators, applied to vector functions defined on Ω, to the mesh Ω^N. For all $x_i\in\Omega^N$, we introduce the semidiscretization approach $\mathbf{U}^N(t)\equiv(\mathbf{U}_i^N(t))$, $i=1,\ldots,N-1$, with $\mathbf{U}_i^N(t)\equiv(U_{1,i},U_{2,i})^T\approx\mathbf{u}(x_i,t)$, as the solution of the following Initial Value Problem

$$\begin{cases} \dfrac{d\mathbf{U}^N}{dt}(t)+\mathcal{L}_\varepsilon^N\overline{\mathbf{U}}^N(t)+\mathcal{A}^N(t,\mathbf{U}^N(t))=\mathbf{0}^N,\ \text{in }\Omega^N\times(0,T],\\ \mathbf{U}_0^N(t)=\mathbf{g}_0(t),\text{in }[0,T],\\ \mathbf{U}_N^N(t)=\mathbf{g}_1(t),\text{in }[0,T],\\ \mathbf{U}^N(0)=[\boldsymbol{\varphi}(x)]_{\Omega^N},\end{cases} \tag{8}$$

where $\overline{\mathbf{U}}^N(t)$ is the natural extension to $\overline{\Omega}^N \times [0,T]$ of the semidiscrete functions $\mathbf{U}^N(t)$, defined on $\Omega^N \times [0,T]$, by adding the corresponding boundary data. As well, $\mathcal{L}_\varepsilon^N(t)$ is the discretization of the operator $\mathcal{L}_{x,\varepsilon}(t)$ using the upwind scheme, i. e.,

$$(\mathcal{L}_\varepsilon^N \overline{\mathbf{U}}^N)_{k,i} = r_{k,i}^- U_{k,i-1}^N + r_{k,i}^+ U_{k,i+1}^N + r_{k,i}^c U_{k,i}^N, \tag{9}$$

with

$$r_{k,i}^- = \frac{-\varepsilon_k}{h_i \overline{h}_i} - \frac{b_{kk}(x_i)}{h_i},\ r_{k,i}^+ = \frac{-\varepsilon_k}{h_{i+1}\overline{h}_i},\ r_{k,i}^c = -(r_{k,i}^- + r_{k,i}^+), \tag{10}$$

and

$$(\mathcal{A}^N(t, \mathbf{U}^N))_{k,i} = a_k(x_i, t, \mathbf{U}_i^N), \tag{11}$$

for $k = 1, 2$ and $i = 1, \ldots, N-1$.

To complete the definition of our numerical algorithm, we apply an appropriate time integrator to the semidiscrete problems (8). Let us consider, for simplicity, a constant time step $\tau = T/M$; let us denote $\mathbf{U}^{N,m} \equiv (\mathbf{U}_i^{N,m}), i = 1, \ldots, N-1$, where $\mathbf{U}_i^{N,m} \equiv (U_{1,i}^{N,m}, U_{2,i}^{N\cdot m})^T$ are the numerical approaches of $\mathbf{u}(x_i, t_m), i = 1, \ldots, N-1$ at $t_m = m\tau$, for $m = 0, 1, \ldots, M$ and let us denote $\overline{\mathbf{U}}^{N,m} \equiv (\mathbf{U}_i^{N,m}), i = 0, \ldots, N$. Next, we describe the fully discrete scheme:

$$\begin{array}{l}
\text{Initialize} \\
\overline{\mathbf{U}}^{N,0} = [\boldsymbol{\varphi}]_{\overline{\Omega}^N}, \\
\left\{ \begin{array}{l}
\text{For } m = 0, 1, \ldots, M-1, \\
\text{First fract. step} \\
\mathbf{U}_0^{N,m+1/3} = \mathbf{U}_0^{N,m}, \\
\dfrac{\mathbf{U}^{N,m+1/3} - \mathbf{U}^{N,m}}{\tau} + \mathcal{A}^N(t_m, \mathbf{U}^{N,m}) = \mathbf{0}^N, \quad i = 1, \ldots, N-1, \\
\mathbf{U}_N^{N,m+1/3} = \mathbf{U}_N^{N,m} \\
\text{Second fract. step} \\
U_{2,i}^{N,m+2/3} = U_{2,i}^{N,m+1/3},\ i = 0, \ldots, N, \\
\left\{ \begin{array}{l}
U_{1,0}^{N,m+2/3} = g_{1,0}(t_{m+1}), \\
\dfrac{U_{1,i}^{N,m+2/3} - U_{1,i}^{N,m+1/3}}{\tau} + (\mathcal{L}_\varepsilon^N \overline{\mathbf{U}}^{N,m+2/3})_{1,i} = 0, \quad i = 1, \ldots, N-1, \\
U_{1,N}^{N,m+2/3} = g_{1,1}(t_{m+1}),
\end{array} \right. \\
\text{Third fract. step} \\
U_{1,i}^{N,m+1} = U_{1,i}^{N,m+2/3},\ i = 0, \ldots, N, \\
\left\{ \begin{array}{l}
U_{2,0}^{N,m+1} = g_{2,0}(t_{m+1}), \\
\dfrac{U_{2,i}^{N,m+1} - U_{2,i}^{N,m+2/3}}{\tau} + (\mathcal{L}_\varepsilon^N \overline{\mathbf{U}}^{N,m+1})_{2,i} = 0, \quad i = 1, \ldots, N-1, \\
U_{2,n}^{N,m+1} = g_{2,1}(t_{m+1}).
\end{array} \right.
\end{array} \right.
\end{array} \tag{12}$$

Observe that the advance in time with (12) only requires the resolution of tridiagonal linear systems. Consequently, we obtain a remarkable reduction of the computational cost when

we compare our proposal with classical implicit methods. In [1], we perform in detail the numerical analysis of our method. Next, we include the main result of uniform convergence for it.

Theorem 1. *(Uniform convergence). Assuming that* $\mathbf{u} \in C^{4,2}(\overline{Q})$*, the global error associated to the numerical method defined by (12) on the Shishkin mesh defined by (6) and (7), satisfies*

$$\max_{0\le m\le M} \|\mathbf{U}^{N,m} - [\mathbf{u}(x,t_m)]_{\overline{\Omega}^N}\|_{\overline{\Omega}^N} \le C\left(N^{-1}\ln N + M^{-1}\right), \tag{13}$$

where the constant C is independent of the diffusion parameters and the discretization parameters N and M.

§4. Numerical tests

In this section we show the numerical results obtained with the algorithm for some test problems of type (1). All the experiments have been performed in a PC with an Intel(R) Core(TM) i5-3470 running at @ 3.20 GHz processor with four cores. Although the algorithm admits an easy parallel programming, the computations have been performed using only one core using GNU Fortran with optimisation -O2. The tridiagonal linear systems involved in our method are solved by using our own implementation of the Thomas algorithm.

The data for the first test problem are given by

$$\begin{array}{l} T=1,\ \ b_{11}(x)=4-3\sin(x),\quad b_{22}(x)=2+xe^{x},\\ a_1(x,t,\mathbf{u})=2u_1+t^2(\cos(u_2)-u_2)+x+t,\ \ a_2(x,t,\mathbf{u})=t^2e^{-u_1}+3u_2+\sin(xt),\\ \boldsymbol{\varphi}(x)=(0,0)^T,\ \ \mathbf{g_0}(t)=(t^3,t\sin(t))^T,\ \ \mathbf{g_1}(t)=(t(1-e^{-t}),t(1-\cos(t)))^T. \end{array} \tag{14}$$

Figure 1 displays the numerical approximation for both components, showing the overlapping boundary layers at $x = 1$. As the exact solution is unknown, to approximate the maximum errors for each component $u_k, k = 1, 2$, defined as

$$\max_{0\le m\le M}\max_{0\le i\le N} |U^m_{N,i,k} - u_k(x_i,t_m)|, k = 1,2,$$

we use the double-mesh principle (see [3] for instance), which calculates

$$d^{N,M}_{\boldsymbol{\varepsilon},k} = \max_{0\le m\le M}\max_{0\le i\le N} |U^m_{N,i,k} - \widehat{U}^{2m}_{2N,2i,k}|,\ d^{N,M}_k = \max_{\varepsilon} d^{N,M}_{\boldsymbol{\varepsilon},k}, k = 1,2, \tag{15}$$

to approximate the errors, where $\{\widehat{\mathbf{U}}^m_{2N,j}\}$ is the numerical solution on a finer mesh $\{(\hat{x}_j,\hat{t}_m)\}$ which has the mesh points of the coarse mesh and their midpoints. From the double-mesh approximated errors we obtain the corresponding numerical orders of convergence in a usual way by

$$p_k = \log(d^{N,M}_{\boldsymbol{\varepsilon},k}/d^{2N,2M}_{\boldsymbol{\varepsilon},k})/\log 2,\ \ p^{uni}_k = \log(d^{N,M}_k/d^{2N,2M}_k)/\log 2, k = 1,2. \tag{16}$$

Table 1 shows the errors and their corresponding orders of convergence for some values of diffusion parameter ε_2, when ε_1 belong to the set

$$R = \{\varepsilon_1; \varepsilon_1 = \varepsilon_2, 2^{-2}\varepsilon_2, \ldots, 2^{-32}\},$$

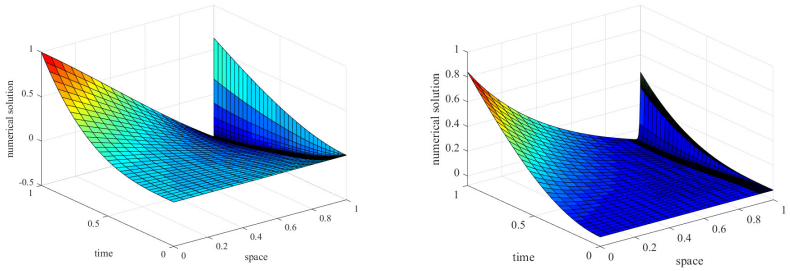

Figure 1: Components 1 (left) and 2 (right) of problem (14) for $\varepsilon_1 = 10^{-4}$, $\varepsilon_2 = 10^{-2}$ with $N = 48, M = 32$.

Table 1: Maximum errors and orders of convergence for problem (14)

ε_2	N=24	N=48	N=96	N=192	N=384	N=768
2^{-6}	7.5693E-2	5.7289E-2	3.4214E-2	1.8453E-2	9.5637E-3	4.8730E-3
	0.4019	0.7437	0.8907	0.9482	0.9728	
	2.0611E-1	1.5558E-1	1.0291E-1	6.9919E-2	4.3333E-2	2.5761E-2
	0.4058	0.5962	0.5576	0.6902	0.7503	
2^{-8}	8.4013E-2	6.3795E-2	3.7467E-2	1.9933E-2	1.0229E-2	5.1703E-3
	0.3972	0.7678	0.9104	0.9625	0.9843	
	2.0783E-1	1.5723E-1	1.0388E-1	7.0511E-2	4.3722E-2	2.5930E-2
	0.4025	0.5980	0.5590	0.6895	0.7537	
2^{-10}	8.5952E-2	6.5867E-2	3.8609E-2	2.0519E-2	1.0530E-2	5.3268E-3
	0.3840	0.7706	0.9120	0.9625	0.9832	
	2.0826E-1	1.5766E-1	1.0413E-1	7.0672E-2	4.3758E-2	2.5911E-2
	0.4015	0.5984	0.5592	0.6916	0.7560	
2^{-12}	8.6393E-2	6.6412E-2	3.8922E-2	2.0681E-2	1.0615E-2	5.3722E-3
	0.3795	0.7709	0.9123	0.9622	0.9825	
	2.0837E-1	1.5777E-1	1.0419E-1	7.0709E-2	4.3835E-2	2.6190E-2
	0.4013	0.5986	0.5593	0.6898	0.7430	
...	...	...	...	...	...	...
...	...	...	...	...	...	...
2^{-24}	8.6535E-2	6.6596E-2	3.9034E-2	2.0737E-2	1.0644E-2	5.3878E-3
	0.3778	0.7707	0.9125	0.9621	0.9823	
	2.0600E-1	1.5515E-1	1.0326E-1	6.8339E-2	4.2211E-2	2.5037E-2
	0.4090	0.5874	0.5955	0.6951	0.7535	
$d_1^{N,M}$	8.6535E-2	6.6596E-2	3.9034E-2	2.0737E-2	1.0644E-2	5.3878E-3
p_1^{uni}	0.3778	0.7707	0.9125	0.9621	0.9823	
$d_2^{N,M}$	2.0839E-1	1.5779E-1	1.0421E-1	7.0721E-2	4.3848E-2	2.6210E-2
p_2^{uni}	0.4012	0.5985	0.5593	0.6896	0.7424	

and for different values of the discretization parameters N and $M = N/3$, taking $\sigma_0 = 1$ in (6). For each value of ε_2 the first two rows show the errors and the numerical orders associated with the first component u_1 and the following two rows these ones associated to the second component u_2. From it, we observe numerical orders of convergence tending to one according with the theoretical results.

An important aspect of the constructed algorithm is related with the reduction of the

computational cost; to see that, we compare the CPU times when solving (14) with a classical method and our algorithm for some values of $N, M = N/3$, and fixed values of the diffusion parameters. The classical method combines the implicit Euler method to discretize, on a uniform mesh, in time, which is well known does not decouple the components of the system, with the upwind finite difference scheme defined on the same piecewise uniform Shishkin mesh as before. In this case, to obtain the numerical approximation of both components a nonlinear system must be solved at each time step; for that, we use the Newton's method with the following stopping criterion

$$\|\mathbf{U}_N^{m,k+1} - \mathbf{U}_N^{m,k}\| \leq 10^{-1} \min\{M^{-2}, M^{-1}N^{-1}\ln N\}, \tag{17}$$

being $\mathbf{U}_N^{m,k}$ the approximation of $\mathbf{U}_N^m$ given by the iteration k of the Newton's method. In all cases, the initial iteration of the Newton's method is taken as $\mathbf{U}_N^{m,0} = \mathbf{U}_N^{m-1}$.

Table 2 shows the maximum errors and their corresponding orders of convergence, obtained for the same values of the diffusion parameters as before, but now using the classical method. From Tables 1 and 2 it is observed that the maximum errors and the orders of convergence are similar; nevertheless there are remarkable differences respect to their computational costs. Table 3 shows the required CPU times in seconds using the algorithm (12) and the classical method described previously, and also the speed-ups that we define as the quotients between the CPU times of the classical method and the corresponding ones to our method. From them, we exhibit that our algorithm is considerably faster than the classical one, as it was expected.

To show that the technique can be extended to systems with more equations, we have chosen a system with three equations for a second test; such system, whose exact solution is again unknown, is given by

$$\begin{cases} \dfrac{\partial u_1}{\partial t} - \varepsilon_1 \dfrac{\partial^2 u_1}{\partial x^2} + (8 + 3\cos(x))\dfrac{\partial u_1}{\partial x} + 3u_1 + t^2(\cos(u_2) - u_2)te^{-u_3} + \sin(x+t) = 0, \\ \dfrac{\partial u_2}{\partial t} - \varepsilon_2 \dfrac{\partial^2 u_2}{\partial x^2} + (2 + e^x)\dfrac{\partial u_2}{\partial x} + t^2 e^{-u_1} + 5u_2 - \sin(u_3) - \cos(xt) = 0, \\ \dfrac{\partial u_3}{\partial t} - \varepsilon_3 \dfrac{\partial^2 u_3}{\partial x^2} + (5 + x\sin(x))\dfrac{\partial u_3}{\partial x} + t\cos(u_1) + e^{-u_2} + 10u_3 + x + t^2 = 0, \\ \mathbf{u}(0,t) = \mathbf{u}(1,t) = (10t\sin(t), 10\cos(t)(1-e^{-t}), 10\sin^2(t))^T,\ t \in [0,1], \\ \mathbf{u}(x,0) = (\sin(\pi x), x(1-x), x\cos(\pi x/2))^T,\ x \in [0,1]. \end{cases} \tag{18}$$

Figure 2 displays the numerical approximation for the three components, showing the overlapping boundary layers at $x = 1$.

Now, to construct the piecewise uniform Shishkin mesh, we use the transition parameters

$$\sigma_{\varepsilon_3} = \min\left\{\frac{1}{2}, \sigma_0\varepsilon_3 \ln N\right\},\ \sigma_{\varepsilon_2} = \min\left\{\frac{2\sigma_{\varepsilon_3}}{3}, \sigma_0\varepsilon_2 \ln N\right\},\ \sigma_{\varepsilon_1} = \min\left\{\frac{\sigma_{\varepsilon_2}}{2}, \sigma_0\varepsilon_1 \ln N\right\}, \tag{19}$$

Table 2: Maximum errors and orders of convergence for problem (14) using the classical method

ε_2	N=24	N=48	N=96	N=192	N=384	N=768
2^{-6}	8.5241E-2	6.1352E-2	4.0172E-2	2.4831E-2	1.4705E-2	8.4205E-3
	0.4744	0.6109	0.6941	0.7558	0.8044	
	7.1611E-2	5.9692E-2	4.4385E-2	3.0394E-2	1.9768E-2	1.2047E-2
	0.2627	0.4275	0.5463	0.6206	0.7146	
2^{-8}	8.5221E-2	6.1374E-2	4.0210E-2	2.4851E-2	1.4711E-2	8.4188E-3
	0.4736	0.6101	0.6942	0.7564	0.8052	
	7.3975E-2	6.1253E-2	4.5310E-2	3.1052E-2	2.0164E-2	1.2278E-2
	0.2723	0.4349	0.5451	0.6229	0.7157	
2^{-10}	8.5266E-2	6.1443E-2	4.0270E-2	2.4893E-2	1.4736E-2	8.4330E-3
	0.4727	0.6095	0.6939	0.7564	0.8052	
	7.4597E-2	6.1666E-2	4.5556E-2	3.1229E-2	2.0271E-2	1.2341E-2
	0.2746	0.4368	0.5448	0.6235	0.7159	
2^{-12}	8.5281E-2	6.1465E-2	4.0289E-2	2.4907E-2	1.4744E-2	8.4378E-3
	0.4725	0.6094	0.6938	0.7564	0.8052	
	7.4753E-2	6.1770E-2	4.5618E-2	3.1273E-2	2.0298E-2	1.2357E-2
	0.2752	0.4373	0.5447	0.6236	0.7160	
...	...	...	...	...	...	...
...	...	...	...	...	...	...
2^{-24}	8.5278E-2	6.1471E-2	4.0296E-2	2.4912E-2	1.4747E-2	8.4395E-3
	0.4723	0.6093	0.6938	0.7564	0.8052	
	7.2245E-2	5.8948E-2	4.3842E-2	2.9347E-2	1.9279E-2	1.1851E-2
	0.2934	0.4271	0.5791	0.6062	0.7020	
$d_1^{N,M}$	8.5286E-2	6.1473E-2	4.0296E-2	2.4912E-2	1.4747E-2	8.4395E-3
p_1^{uni}	0.4724	0.6093	0.6938	0.7564	0.8052	
$d_2^{N,M}$	7.4793E-2	6.1795E-2	4.5632E-2	3.1283E-2	2.0304E-2	1.2361E-2
p_2^{uni}	0.2754	0.4374	0.5447	0.6236	0.7160	

Table 3: CPU times and speed-ups for problem (14) with $\varepsilon_2 = 2^{-16}$ and $\varepsilon_1 = 2^{-20}$

	N=192	N=384	N=768	N=1536	N=3072
splitting	0.01562	0.07312	0.31250	1.26562	5.06250
classical	0.40625	1.60937	6.43750	25.73437	103.03125
speed-ups	26.00832	22.00998	20.60000	20.33409	20.351851

and, taking N a positive integer multiple of 3, the grid points are given by

$$x_j = \begin{cases} jH, & j = 0, \ldots, N/2, \\ x_{N/2} + (j - N/2)h_{\varepsilon_3}, & j = N/2 + 1, \ldots, 2N/3, \\ x_{2N/3} + (j - 2N/3)h_{\varepsilon_2}, & j = 2N/3 + 1, \ldots, 5N/6, \\ x_{5N/6} + (j - 5N/6)h_{\varepsilon_1}, & j = 5N/6 + 1, \ldots, N, \end{cases}$$

where $h_{\varepsilon_1} = 6\sigma_{\varepsilon_1}/N,\ h_{\varepsilon_2} = 6(\sigma_{\varepsilon_2} - \sigma_{\varepsilon_1})/N, h_{\varepsilon_3} = 6(\sigma_{\varepsilon_3} - \sigma_{\varepsilon_2})/N,\ H = 2(1 - \sigma_{\varepsilon_3})/N.$

To complete the numerical algorithm, we use again a linearly implicit method of type "splitting-by-components", which now has four fractionary steps, being the first one explicit. We use again the double mesh principle to estimate the maximum errors. Table 4 shows the maximum errors and their corresponding numerical orders of convergence for the three

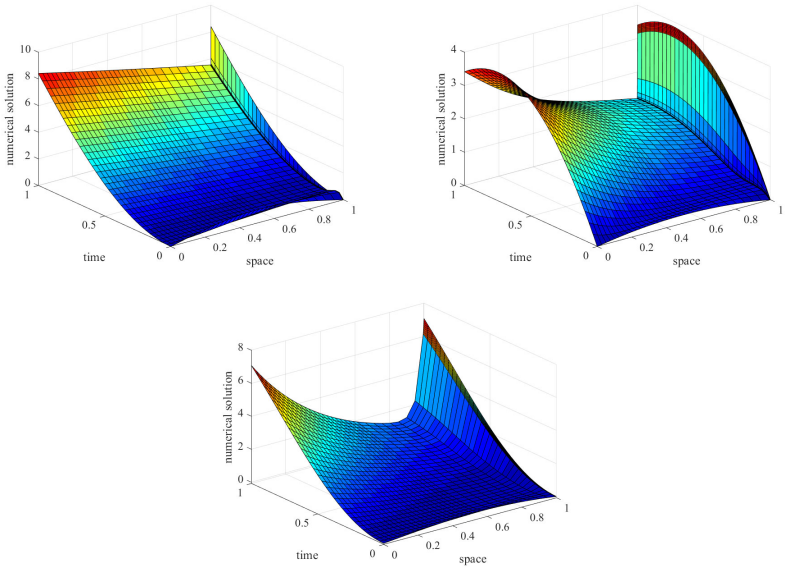

Figure 2: Components of problem (18) for $\varepsilon_1 = 10^{-5}$, $\varepsilon_2 = 10^{-3}$, $\varepsilon_3 = 10^{-1}$, with $N = 48, M = 32$ (left up u_1, right up u_2, bottom u_3.)

components, choosing some values of ε_3, of the discretization parameters N and $M = N/2$ and $\sigma_0 = 1$ in (19). Besides ε_2 covers the set $R_2 = \{\varepsilon_2; \varepsilon_2 = \varepsilon_3, 2^{-2}\varepsilon_3, \ldots, 2^{-22}\}$ and ε_1 the set $R_1 = \{\varepsilon_1; \varepsilon_1 = \varepsilon_2, 2^{-2}\varepsilon_2, \ldots, 2^{-26}\}$. For each value of ε_3, the first and second rows correspond to errors and orders for the first component, the third and the fourth ones to the second component and the fifth and sixth ones to the third component. From it, we observe uniform convergence of almost first order.

We compare again the CPU times of our method and the same classical method as in the first example, when solving the problem (18) for some values of $N, M = N/2$, and fixed values for the diffusion parameters ε_i, $i = 1, 2, 3$. We have used the same stopping criterion given in (17) for the Newton's method. Table 5 shows the required CPU time in seconds using our algorithm and the classical method. From it, we see a higher speed up for our algorithm respect to the classical one, due to the increase of the number of equations, as it was expected.

Acknowledgements

This research was partially supported by the Aragón Government, European Social Fund (group E24-17R), Universidad Pública de Navarra and the research project TED2021-130884B-I00.

Table 4: Maximum errors and orders of convergence for problem (18)

ε_3	N=36	N=72	N=144	N=288	N=576	N=1152
	3.3023E-1	3.0252E-1	2.6316E-1	1.8947E-1	1.3010E-1	8.1556E-2
	0.1264	0.2011	0.4739	0.5423	0.6738	
	3.0569E-1	2.4313E-1	1.6402E-1	1.1439E-1	7.1461E-2	4.2542E-2
2^{-6}	0.3303	0.5678	0.5199	0.6788	0.7483	
	6.8311E-1	5.8837E-1	4.3311E-1	2.9840E-1	1.9115E-1	1.1521E-1
	0.2154	0.4420	0.5375	0.6425	0.7305	
	3.3046E-1	3.0254E-1	2.6324E-1	1.8952E-1	1.3013E-1	8.1574E-2
	0.1273	0.2007	0.4740	0.5423	0.6738	
	3.0660E-1	2.4372E-1	1.6433E-1	1.1459E-1	7.1580E-2	4.2602E-2
2^{-8}	0.3311	0.5687	0.5200	0.6789	0.7486	
	6.9180E-1	5.9395E-1	4.3608E-1	3.0065E-1	1.9247E-1	1.1597E-1
	0.2200	0.4458	0.5365	0.6435	0.7308	
	3.3052E-1	3.0255E-1	2.6327E-1	1.8953E-1	1.3014E-1	8.1579E-2
	0.1276	0.2006	0.4741	0.5423	0.6738	
	3.0666E-1	2.4389E-1	1.6442E-1	1.1466E-1	7.1625E-2	4.2627E-2
2^{-10}	0.3304	0.5689	0.5200	0.6788	0.7487	
	6.9312E-1	5.9446E-1	4.3643E-1	3.0079E-1	1.9261E-1	1.1610E-1
	0.2215	0.4458	0.5370	0.6430	0.7304	
	3.3053E-1	3.0255E-1	2.6327E-1	1.8954E-1	1.3015E-1	8.1580E-2
	0.1276	0.2006	0.4741	0.5424	0.6738	
	3.0647E-1	2.4370E-1	1.6442E-1	1.1462E-1	7.1564E-2	4.2621E-2
2^{-12}	0.3306	0.5677	0.5206	0.6795	0.7477	
	6.8962E-1	5.9079E-1	4.3474E-1	2.9893E-1	1.9176E-1	1.1572E-1
	0.2232	0.4425	0.5403	0.6405	0.7287	
...	...	...	...	...	...	...
...	...	...	...	...	...	...
	3.3053E-1	3.0255E-1	2.6328E-1	1.8954E-1	1.3015E-1	8.1581E-2
	0.1276	0.2006	0.4741	0.5424	0.6738	
	2.8003E-1	2.1899E-1	1.5344E-1	1.0148E-1	6.2995E-2	3.7185E-2
2^{-18}	0.3548	0.5131	0.5966	0.6878	0.7605	
	6.4356E-1	4.8540E-1	3.2835E-1	2.1755E-1	1.3564E-1	7.9586E-2
	0.4069	0.5639	0.5939	0.6816	0.7692	
$d_1^{N,M}$	3.3053E-1	3.0255E-1	2.6328E-1	1.8954E-1	1.3015E-1	8.1581E-2
p_1^{uni}	0.1276	0.2006	0.4741	0.5424	0.6738	
$d_2^{N,M}$	3.0666E-1	2.4389E-1	1.6442E-1	1.1466E-1	7.1625E-2	4.2627E-2
p_2^{uni}	0.3304	0.5688	0.5200	0.6788	0.7487	
$d_3^{N,M}$	6.9312E-1	5.9446E-1	4.3643E-1	3.0079E-1	1.9261E-1	1.1610E-1
p_3^{uni}	0.2215	0.4458	0.5370	0.6430	0.7304	

Table 5: CPU times and speed-ups for problem (18) with $\varepsilon_3 = 2^{-12}, \varepsilon_2 = 2^{-16}$ and $\varepsilon_1 = 2^{-20}$

	N=144	N=288	N=576	N=1152	N=2304
splitting	0.04680	0.17160	0.62400	2.51161	9.90606
classical	1.20121	4.75803	18.90732	75.20808	301.28473
speed-ups	25.66688	27.72744	30.30019	29.94417	30.41418

References

[1] Clavero, C., and Jorge, J. C. Efficient numerical methods for semilinear one dimensional parabolic singularly perturbed convection-diffusion systems. *in preparation* (2023).

[2] Clavero, C., and Jorge, J. C. A splitting uniformly convergent method for one-dimensional parabolic singularly perturbed convection-diffusion systems. *Appl. Numer. Math. 183* (2023), 317–332. Available from: https://doi.org/10.1016/j.apnum.2022.09.012.

[3] Farrell, P. A., Hegarty, A. F., Miller, J. J. H., O'Riordan, E., and Shishkin, G. I. *Robust computational techniques for boundary layers*, vol. 16 of *Applied Mathematics (Boca Raton)*. Chapman & Hall/CRC, Boca Raton, FL, 2000.

[4] Linss, T., and Stynes, M. Numerical solution of systems of singularly perturbed differential equations. *Comput. Methods Appl. Math. 9*, 2 (2009), 165–191. Available from: https://doi.org/10.2478/cmam-2009-0010.

[5] Liu, L.-B., Long, G., and Zhang, Y. Parameter uniform numerical method for a system of two coupled singularly perturbed parabolic convection-diffusion equations. *Adv. Difference Equ.* (2018), Paper No. 450, 19. Available from: https://doi.org/10.1186/s13662-018-1907-1.

[6] Pao, C. V. *Nonlinear parabolic and elliptic equations*. Plenum Press, New York, 1992.

[7] Singh, M. K., and Natesan, S. Numerical analysis of singularly perturbed system of parabolic convection-diffusion problem with regular boundary layers. *Differ. Equ. Dyn. Syst. 30*, 3 (2022), 695–717. Available from: https://doi.org/10.1007/s12591-019-00462-2.

Carmelo Clavero
IUMA & Departamento de Matemática Aplicada,
Universidad de Zaragoza
clavero@unizar.es

Juan Carlos Jorge
ISC & Departamento de Estadística, Informática y Matemáticas,
Universidad Pública de Navarra
jcjorge@unavarra.es