Fast Explicit Operator Splitting Method for Convection-Diffusion Equations

Alina Chertock[†], Alexander Kurganov[‡] and Guergana Petrova[§]

Abstract

Systems of convection-diffusion equations model a variety of physical phenomena which often occur in real life. Computing the solutions of these systems, especially in the convection dominated case, is an important and challenging problem that requires development of fast, reliable and accurate numerical methods. In this paper, we propose a second-order fast explicit operator splitting (FEOS) method based on the Strang splitting. The main idea of the method is to solve the parabolic problem via a discretization of the formula for the exact solution of the heat equation, which is realized using a conservative and accurate quadrature formula. The hyperbolic problem is solved by a second-order finite-volume Godunov-type scheme.

We provide a theoretical estimate for the convergence rate in the case of one-dimensional systems of linear convection-diffusion equations with smooth initial data. Numerical convergence studies are performed for one-dimensional nonlinear problems as well as for linear convection-diffusion equations with both smooth and nonsmooth initial data. We finally apply the FEOS method to the one- and two-dimensional systems of convection-diffusion equations which model the polymer flooding process in enhanced oil recovery. Our results show that the FEOS method is capable to achieve a remarkable resolution and accuracy in a very efficient manner, that is, when only few splitting steps are performed.

1 Introduction

We present a fast explicit operator splitting (FEOS) method for the initial value problem (IVP) for the systems of convection-diffusion equations:

$$\mathbf{u}_t + \nabla_{\mathbf{x}} \cdot \mathbf{f}(\mathbf{u}) = D\Delta\mathbf{u}, \qquad \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}). \tag{1.1}$$

Here, $\mathbf{u}(\mathbf{x},t) = (u_1(\mathbf{x},t),\dots,u_l(\mathbf{x},t))^T$ is an unknown l-vector, $\mathbf{f}$ is a nonlinear convection flux, and $D = \mathrm{diag}(\varepsilon_1,\dots,\varepsilon_l)$ is a constant diagonal matrix with positive entries. In the general multidimensional case, $\mathbf{u}$ is a vector function of a time variable t and d-spatial variable $\mathbf{x} = (x_1,\dots,x_d)$ with corresponding fluxes $\mathbf{f} = (f^1,\dots,f^d)$.

[†]Department of Mathematics, North Carolina State University, Raleigh, NC 27695; chertock@math.ncsu.edu
[‡]Mathematics Department, Tulane University, New Orleans, LA 70118; kurganov@math.tulane.edu
[§]Department of Mathematics, Texas A&M University, College Station, TX 77843; gpetrova@math.tamu.edu

Systems of convection-diffusion equations arise in a variety of applications and represent mathematical models for a number of (physical) processes in fluid mechanics, astrophysics, meteorology, multiphase flow in oil reservoirs, polymer flow, financial modeling, and many other areas. The convection dominated case, which is of a particular importance, is the most challenging one from numerical perspective: although the solution of (1.1) is typically smooth for $t > 0$, its gradients may be very large and a full resolution of viscous shock layers may be out of practical reach, especially in the multidimensional case. Therefore, one may have to apply shock capturing methods, originally developed for hyperbolic systems of conservation laws. At the same time, even though the impact of diffusion is not too significant, its presence typically reduces the efficiency of an explicit numerical scheme for (1.1). One of the ways to overcome this difficulty is to use an operator splitting method, which can be briefly described as follows.

We denote by $\mathcal{S}_{\mathcal{H}}$ the *exact* solution operator associated with the corresponding hyperbolic system:

$$\mathbf{u}_t + \nabla_{\mathbf{x}} \cdot \mathbf{f}(\mathbf{u}) = 0, \tag{1.2}$$

and by $\mathcal{S}_{\mathcal{P}}$ the *exact* solution operator associated with the (linear) parabolic system

$$\mathbf{u}_t = D\Delta\mathbf{u}. \tag{1.3}$$

Then, introducing a (small) time step Δt, the solution of the original convection-diffusion system (which is assumed to be available at time t) is evolved in time in two substeps. First, the hyperbolic system (1.2) is solved on the time interval $(t, t + \Delta t]$:

$$\mathbf{u}^*(\mathbf{x}) = \mathcal{S}_{\mathcal{H}}(\Delta t)\mathbf{u}(\mathbf{x}, t), \tag{1.4}$$

and then the parabolic solution operator is applied to $\mathbf{u}^*$, which results in the following approximate solution at time $t + \Delta t$:

$$\mathbf{u}(\mathbf{x}, t + \Delta t) = \mathcal{S}_{\mathcal{P}}(\Delta t)\mathbf{u}^*(\mathbf{x}) = \mathcal{S}_{\mathcal{P}}(\Delta t)\mathcal{S}_{\mathcal{H}}(\Delta t)\mathbf{u}(\mathbf{x}, t). \tag{1.5}$$

In general, if all solutions involved in the two-step splitting algorithm (1.4)–(1.5) are smooth, the operator splitting method is second-order accurate at each time step and *first-order* accurate when it is applied for advancing the solution of the IVP (1.1) from $t = 0$ to the final time $T = N\Delta t$, $N \in \mathbb{N}$ (see, e.g., [13]).

Higher-order operator splitting algorithms can be derived by considering further substeps. For instance, one time step of the *second-order* Strang splitting method [13] consists of three substeps:

$$\mathbf{u}(\mathbf{x}, t + \Delta t) = \mathcal{S}_{\mathcal{H}}(\Delta t/2)\mathcal{S}_{\mathcal{P}}(\Delta t)\mathcal{S}_{\mathcal{H}}(\Delta t/2)\mathbf{u}(\mathbf{x}, t). \tag{1.6}$$

We also refer the reader to [17], where similar splitting methods were derived by simply choosing certain longer time increment sequences, while again alternating $\mathcal{S}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{P}}$. The time increments of methods of orders 4, 6 and 8 can be found, for example, in [9]. Although they are, in general, not unique, it can be shown that splitting methods of orders higher than two will require some negative time increments [14], which may cause numerical instability when time irreversible (dissipative) systems are solved.

In applications, the exact solution operators $\mathcal{S}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{P}}$ are replaced by their numerical approximations. Note that the main advantage of the operator splitting technique is the fact

that the hyperbolic, (1.2), and the parabolic, (1.3), subproblems, which are of different nature, can be solved numerically by different methods.

Since hyperbolic (systems of) conservation laws usually develop discontinuities, the system (1.2) should be solved by a (high resolution) shock capturing scheme. The "hyperbplic" substep in our FEOS method is based on finite-volume schemes. Our particular choice is the second-order semi-discrete Godunov-type central schemes, originally introduced in [8], and then further improved in [4, 5, 6], where the so-called *central-upwind* schemes have been developed. We note, however, that the "hyperbolic" substep of our FEOS method is not tied up to a specific choice of a finite-volume scheme and can be implemented with one's favorite projection-evolution method, in which a piecewise polynomial interpolant of the solution is first evolved to a new time level and then projected back onto the space of piecewise polynomial functions.

The outcome of every "hyperbolic" substep is a global approximation of $\mathbf{u}^*$, realized in terms of polynomial pieces over spatial cells. The main idea of the proposed FEOS method is to perform the "parabolic" substep using the exact solution operator for the heat equation. The solution is in the form of a convolution integral, which is approximated using an appropriate conservative and sufficiently accurate quadrature. Such a quadrature is studied in §2.

In the case of a linear flux $\mathbf{f}$, we provide a rigorous error estimate for the introduced FEOS method (see §2). The theoretical results are also supported by numerical experiments, reported in §3.1. The case of a nonlinear flux $\mathbf{f}$ is much more complicated since the solution may develop jump discontinuities in the process of "hyperbolic evolution". In this case, only numerical convergence study is performed (see §3.2). Finally, we apply the FEOS method to the one- (1-D) and two-dimensional (2-D) systems that model polymer flooding processes in enhanced oil recovery [2, 12, 16]. The proposed FEOS method seems to outperform the existing alternative approaches.

2 Fast Explicit Operator Splitting (FEOS) Method

In this section, we describe the FEOS method. For simplicity, we present only its 2-D version and provide a convergence rate analysis in the 1-D case.

The FEOS method is based on the Strang operator splitting (1.6), in which the exact solution operators $\mathcal{S}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{P}}$ are replaced by their numerical approximations. The "hyperbolic" substep is carried out using a Godunov-type finite-volume scheme, in which a piecewise polynomial interpolant is first reconstructed from cell averages (computed at the previous time step), and then evolved to the next time level according to the integral form of the hyperbolic system (1.2). More precisely, the projection-evolution Godunov-type approach may be described as follows.

We introduce a uniform grid (the extension to nonuniform Cartesian grids is pretty straightforward) with $x_\alpha = \alpha\Delta x$ and $y_\beta = \beta\Delta y$. Assume that we have computed the cell averages of the solution at some time level t:

$$\bar{\mathbf{u}}_{j,k}(t) \approx \bar{\mathbf{u}}(x_j, y_k, t) := \frac{1}{\Delta x\Delta y}\int\limits_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}}\int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{u}(x, y, t)\,dx\,dy.$$

We then reconstruct a piecewise polynomial interpolant for each component of the vector $\mathbf{u} = (u_1, \ldots, u_l)^T$. The (formal) order of accuracy of these reconstructions usually determines the (formal) spatial order of the finite-volume scheme. In this paper, we will focus on second-order

schemes that require conservative piecewise linear (in x and y) reconstructions of the following form:

$$\widetilde{\mathbf{u}}(x,y;t)=\bar{\mathbf{u}}_{j,k}(t)+\mathbf{s}^x_{j,k}(x-x_j)+\mathbf{s}^y_{j,k}(y-y_k),\quad \text{for } (x,y)\in(x_{j-\frac12},x_{j+\frac12})\times(y_{k-\frac12},y_{k+\frac12}),\quad (2.1)$$

where the slopes $\mathbf{s}^x_{j,k}$ and $\mathbf{s}^y_{j,k}$ have to be (at least) first-order approximations of the partial derivatives $\mathbf{u}_x(x_j,y_k,t)$ and $\mathbf{u}_y(x_j,y_k,t)$, respectively. In order to ensure a non-oscillatory behavior of the reconstruction, which is a necessary condition for the overall scheme to be non-oscillatory, the slopes should be computed with the help of a nonlinear limiter. In our numerical experiments, we have used the one-parameter family of *minmod* limiters (see, e.g., [10, 11, 15]):

$$\begin{aligned}\mathbf{s}^x_{j,k}&=\text{minmod}\left(\theta\frac{\bar{\mathbf{u}}_{j+1,k}(t)-\bar{\mathbf{u}}_{j,k}(t)}{\Delta x},\ \frac{\bar{\mathbf{u}}_{j+1,k}(t)-\bar{\mathbf{u}}_{j-1,k}(t)}{2\Delta x},\ \theta\frac{\bar{\mathbf{u}}_{j,k}(t)-\bar{\mathbf{u}}_{j-1,k}(t)}{\Delta x}\right),\\ \mathbf{s}^y_{j,k}&=\text{minmod}\left(\theta\frac{\bar{\mathbf{u}}_{j,k+1}(t)-\bar{\mathbf{u}}_{j,k}(t)}{\Delta y},\ \frac{\bar{\mathbf{u}}_{j,k+1}(t)-\bar{\mathbf{u}}_{j,k-1}(t)}{2\Delta y},\ \theta\frac{\bar{\mathbf{u}}_{j,k}(t)-\bar{\mathbf{u}}_{j,k-1}(t)}{\Delta y}\right),\end{aligned}\qquad (2.2)$$

where $\theta\in[1,2]$, and the multivariate minmod function is defined by

$$\text{minmod}(z_1,z_2,...):=\begin{cases}\min_j\{z_j\}, & \text{if } \ z_j>0\ \ \forall j,\\ \max_j\{z_j\}, & \text{if } \ z_j<0\ \ \forall j,\\ 0, & \text{otherwise.}\end{cases}$$

Notice that larger θ's correspond to less dissipative but, in general, more oscillatory limiters.

Finally, the solution at the new time level $t+\Delta t_{\text{HYP}}$ is obtained by (approximately) solving the integral form of the system (1.2), subject to the piecewise linear initial data (2.1)–(2.2), prescribed at time t. This may be done using either the central or the upwind approach. In this paper, we evolved the solution using the semi-discrete central-upwind scheme from [5].

Remark. Note that due to the CFL condition, Δt_{HYP} may be smaller than $\Delta t/2$, where Δt is the size of the splitting step. In this case, the approximate "hyperbolic" substep of the splitting algorithm would consist of several smaller "finite-volume subsubsteps" of size Δt_{HYP}. This is a typical situation, for example, in applications to polymer flows (see §4), where one is interested in developing a reliable operator splitting method that is capable to produce a high quality approximate solution whith a small number of splitting steps, that is, while keeping Δt relatively large (see, e.g., [3] and the references therein).

Once the solution of the first "hyperbolic" substep in (1.6) is performed, the cell averages

$$\bar{\mathbf{u}}^*_{j,k}\approx\frac{1}{\Delta x\Delta y}\int\limits_{y_{k-\frac12}}^{y_{k+\frac12}}\int\limits_{x_{j-\frac12}}^{x_{j+\frac12}}\mathcal{S}_{\mathcal{H}}(\Delta t/2)\mathbf{u}(x,y,t)\,dx\,dy$$

are available, and we reconstruct another piecewise linear interpolant $\widetilde{\mathbf{u}}^*(x,y)$ following (2.1)–(2.2). This piecewise linear function is then used as an initial condition for the parabolic IVP:

$$\mathbf{u}_t=D\Delta\mathbf{u},\qquad \mathbf{u}(x,y,t)=\widetilde{\mathbf{u}}^*(x,y),\qquad (2.3)$$

which is now, according to the Strang splitting algorithm (1.6), to be solved on the time interval $(t, t+\Delta t]$.

Note that since D is a diagonal matrix, the parabolic system in (2.3) is actually a set of l uncoupled heat equations for each component of $\mathbf{u}$:

$$(u_i)_t = \varepsilon_i \Delta u_i, \qquad u_i(x,y,t) = \widetilde{u}_i^*(x,y), \quad i = 1,\dots,l. \tag{2.4}$$

From now on, we will simplify our notation by using v instead of any of the u_i's and ε instead of any of the ε_i's.

Next, we recall that in the 2-D case, the exact solution of (2.4) at time $t+\Delta t$ may be expressed in the following integral form:

$$v^{**}(x,y) := v(x,y,t+\Delta t) = \int\limits_{\mathbb{R}^2} G(x-\xi, y-\eta, \varepsilon\Delta t)\widetilde{v}^*(\xi,\eta)\, d\xi\, d\eta, \tag{2.5}$$

where G is the "heat" kernel:

$$G(x,y,t) = \frac{1}{4\pi t} e^{-\frac{x^2+y^2}{4t}}. \tag{2.6}$$

Now, using the fact that

$$\int\limits_{\mathbb{R}^2} G(x-\xi, y-\eta, \varepsilon\Delta t)\, d\xi\, d\eta = 1,$$

we write (2.5) in the equivalent form:

$$v^{**}(x,y) = \widetilde{v}^*(x,y) + \int\limits_{\mathbb{R}^2} G(x-\xi, y-\eta, \varepsilon\Delta t)\left(\widetilde{v}^*(\xi,\eta) - \widetilde{v}^*(x,y)\right)\, d\xi\, d\eta. \tag{2.7}$$

Formula (2.7) is more suitable for our purposes because it is symmetric and allows us to discretize the spatial integral while preserving the conservation of v, that is, ensuring that the equality,

$$\int\limits_{\mathbb{R}^2} v^{**}(x,y)\, dx\, dy = \int\limits_{\mathbb{R}^2} \widetilde{v}^*(x,y)\, dx\, dy,$$

is satisfied on a discrete level as well.

Since for the next "hyperbolic" substep only the cell averages of $v^{**}(x,y)$ are needed, we average (2.7) over the corresponding computational cells to obtain:

$$\bar{v}^{**}_{j,k} = \bar{v}^*_{j,k} + \frac{1}{\Delta x \Delta y} \int\limits_{y_{k-\frac12}}^{y_{k+\frac12}} \int\limits_{x_{j-\frac12}}^{x_{j+\frac12}} \left[\int\limits_{\mathbb{R}^2} G(x-\xi, y-\eta, \varepsilon\Delta t)\left(\widetilde{v}^*(\xi,\eta) - \widetilde{v}^*(x,y)\right) d\xi\, d\eta \right] dx\, dy$$

$$= \bar{v}^*_{j,k} + \frac{1}{\Delta x \Delta y} \sum_{i,\ell\in\mathbb{Z}} \int\limits_{y_{k-\frac12}}^{y_{k+\frac12}} \int\limits_{x_{j-\frac12}}^{x_{j+\frac12}} \int\limits_{y_{\ell-\frac12}}^{y_{\ell+\frac12}} \int\limits_{x_{i-\frac12}}^{x_{i+\frac12}} G(x-\xi, y-\eta, \varepsilon\Delta t)\left(\widetilde{v}^*(\xi,\eta) - \widetilde{v}^*(x,y)\right) d\xi\, d\eta\, dx\, dy. \tag{2.8}$$

Next, the integrals on the right-hand side (RHS) of (2.8) are discretized using the midpoint quadrature, that is

$$\bar{v}^{**}_{j,k}=\bar{v}^{*}_{j,k}+\Delta x\Delta y\sum_{i,\ell\in\mathbb{Z}}G(x_j-x_i,y_k-y_\ell,\varepsilon\Delta t)(\bar{v}^{*}_{i,\ell}-\bar{v}^{*}_{j,k}). \tag{2.9}$$

Remarks.

1. In practice, the computational domain is finite and the infinite sum on the RHS of (2.9) reduces to the sum over all computational cells (we obviously need to assume that the solution is "exponentially flat" near the artificially imposed boundaries).

2. The midpoint quadrature (2.9) is conservative:

$$\sum_{j,k\in\mathbb{Z}}\bar{v}^{**}_{j,k}=\sum_{j,k\in\mathbb{Z}}\bar{v}^{*}_{j,k}+\Delta x\Delta y\sum_{j,k,i,\ell\in\mathbb{Z}}G(x_j-x_i,y_k-y_\ell,\varepsilon\Delta t)(\bar{v}^{*}_{i,\ell}-\bar{v}^{*}_{j,k})=\sum_{j,k\in\mathbb{Z}}\bar{v}^{*}_{j,k},$$

since $G(\pm(x_j-x_i),\pm(y_k-y_\ell),\varepsilon\Delta t)=G(x_j-x_i,y_k-y_\ell,\varepsilon\Delta t)$ due to the symmetry of the "heat" kernel (2.6). We also note that one may use a higher-order quadrature for discretizing the integrals in (2.8), which will be conservative as long as it is symmetric.

The third and last substep of the FEOS method is again "hyperbolic". We start with the cell averages $\bar{\mathbf{u}}^{**}_{j,k}$, computed at the "parabolic" substep, reconstruct a piecewise linear interpolant $\hat{\mathbf{u}}$ (following (2.1)–(2.2)), and then evolve it using the same finite-volume method as in the first "hyperbolic" substep to obtain the cell averages of the solution of (1.1) at the new time level $t+\Delta t$,

$$\bar{\mathbf{u}}_{j,k}(t+\Delta t)\approx\frac{1}{\Delta x\Delta y}\int\limits_{y_{k-\frac12}}^{y_{k+\frac12}}\int\limits_{x_{j-\frac12}}^{x_{j+\frac12}}\mathcal{S}_{\mathcal{H}}(\Delta t/2)\hat{\mathbf{u}}(x,y)\,dx\,dy.$$

This completes the description of one time step of the FEOS method.

2.1 Error Analysis

Here, we study the convergence rate of the FEOS method. At each time step ($t\to t+\Delta t$), the L^∞-error of our method, $E(\Delta t)$, is a sum of three errors: the operator splitting error $E_{\mathcal{S}}(\Delta t)$, the error $E_{\mathcal{H}}(\Delta t)$ of the "hyperbolic" substep and the error $E_{\mathcal{P}}(\Delta t)$ of the "parabolic" substep. The first two errors are hard (if not impossible) to estimate in the nonlinear case since then the solution of the hyperbolic subproblem may develop discontinuities. Our error analysis will be thus limited to the linear case only. However, we will be able to bound the quadrature error $E_{\mathcal{P}}(\Delta t)$ in both smooth and nonsmooth cases and the resulting orders will depend on the smoothness of the solution.

We begin with the simplest case: the 1-D system with a linear flux $\mathbf{f}$ and smooth initial data $\mathbf{u}_0$. Then both solutions of the IVP (1.1) and of the corresponding hyperbolic system (1.2) are globally smooth and, as it has been mentioned in §1, the error of one step of the Strang splitting is $E_{\mathcal{S}}(\Delta t)\le C(\varepsilon\Delta t)^3$, where $\varepsilon:=\max(\varepsilon_1,\ldots,\varepsilon_l)$.[1] To simplify the notation, we assume from now

[1]Throughout this section, we denote by $C,C_1,C_2,\ldots$ the absolute constants, which are independent of the small scales of both the PDE system: $\varepsilon_1,\ldots,\varepsilon_l$, and the numerical method: $\Delta x,\Delta t$. Particular values of these constants are of no (practical) importance, and thus we do not keep track of them.

on that $\varepsilon_1 = \ldots = \varepsilon_l = \varepsilon$. We also assume that the hyperbolic subproblem is numerically solved using a convergent, uniformly second-order method. In our numerical experiments, reported in §3.1.1, we have used the "no limiter" version of the second-order central-upwind scheme, in which the slopes in (2.1) are computed using central differences. Then the size of $E_{\mathcal{H}}(\Delta t)$ can be bounded by $C\Delta t(\Delta x)^2$ (note that here, Δt is the total time period for which the linear hyperbolic problem is solved in substeps 1 and 3 of the FEOS method). We will show in this section that the error of the "parabolic" substep is:

$$E_{\mathcal{P}}(\Delta t) \le C_1(\Delta x)^2 + C_2\frac{(\Delta x)^4}{(\varepsilon\Delta t)^2} + C_3\frac{(\Delta x)^4}{(\varepsilon\Delta t)^{3/2}}.$$

Therefore, adding the three components of the error at each time step and taking into account that the total number of splitting steps is proportional to $1/\Delta t$, we obtain the following estimate for the the global L^∞-error of the FEOS method:

$$E(\Delta t) \le C\left[\varepsilon^3(\Delta t)^2 + (\Delta x)^2 + \frac{(\Delta x)^2}{\Delta t} + \frac{(\Delta x)^4}{\Delta t(\varepsilon\Delta t)^2} + \frac{(\Delta x)^4}{\Delta t(\varepsilon\Delta t)^{3/2}}\right]. \tag{2.10}$$

Notice that the presence of the ε^3 factor in the first term on the RHS of (2.10), ensures that even when the size of the splitting step (Δt) is large, the overall error may be small in the convection-dominated case, that is, when ε is small. This is the reason why we have been able to obtain a high quality results for the polymer flow (see §4), where it is essential (from an application perspective) to keep the number of splitting steps as small as possible.

In order to achieve (theoretically) the fastest convergence rate of the FEOS method, one needs to balance the errors in (2.10), that is, one needs to select α in $\Delta t \sim \varepsilon^{-1}(\Delta x)^\alpha$ so that to minimize the overall error $E(\Delta t)$. The optimal choice is $\alpha = 2/3$, and then the best theoretical error bound is:

$$E(\Delta t) \le C\max\{(\Delta x)^2, \varepsilon(\Delta x)^{4/3}\}.$$

Here, for small ε the dominating factor in the error is $\mathcal{O}((\Delta x)^2)$. This explains why in our numerical examples (see §3.1.1) we have observed a second order of convergence even for non-optimal values of α.

We now turn to the proof of the error estimate for $E_{\mathcal{P}}(\Delta t)$. Note that if we start with a smooth initial condition $\mathbf{u}_0$, the exact solution is globally smooth, and thus the function $S_{\mathcal{H}}(\Delta t/2)\mathbf{u}(x,t)$ will be at least as smooth as $\mathbf{u}_0$. Let us denote by $v(x)$ the i-th component of this function. Let

$$\widetilde{v}^*(x) = \sum_{j=-\infty}^{\infty}[\bar{v}_j^* + s_j(x-x_j)]\chi_{[x_{j-\frac12},x_{j+\frac12}]}(x), \quad \chi_{[x_{j-\frac12},x_{j+\frac12}]}(x) := \begin{cases} 1, & \text{if } x_{j-\frac12} < x < x_{j+\frac12}, \\ 0, & \text{otherwise,} \end{cases}$$

be a second-order piecewise linear approximation of v, that is $\bar{v}_j^* = v(x_j) + \mathcal{O}((\Delta x)^2)$ and $s_j = v'(x_j) + \mathcal{O}(\Delta x)$. Then the formula for the cell averages of the solution obtained at the end of the "parabolic" substep, (2.8), reduces to

$$\bar{v}_j^{**} = \bar{v}_j^* + \frac{1}{\Delta x}\int\limits_{x_{j-\frac12}}^{x_{j+\frac12}}\int\limits_{-\infty}^{\infty} G(x-\xi,\varepsilon\Delta t)(\widetilde{v}^*(\xi) - \widetilde{v}^*(x))\,d\xi\,dx, \tag{2.11}$$

where G is the 1-D "heat" kernel:

$$G(z,\varepsilon\Delta t)=\frac{1}{2\sqrt{\pi\varepsilon\Delta t}}e^{-\frac{z^2}{4\varepsilon\Delta t}}.$$

Next, we compute the cell averages $\bar{v}_j^{**}$ by applying the midpoint quadrature to the integral on the RHS of (2.11):

$$\bar{v}_j^{**}=\bar{v}_j^{*}+\Delta x\sum_{i=-\infty}^{\infty}G(x_j-x_i,\varepsilon\Delta t)(\bar{v}_i^{*}-\bar{v}_j^{*}),$$

incurring an error

$$\left|\frac{1}{\Delta x}\int\limits_{x_{j-\frac12}}^{x_{j+\frac12}}\int\limits_{-\infty}^{\infty}G(x-\xi,\varepsilon\Delta t)(\widetilde{v}^*(\xi)-\widetilde{v}^*(x))\,d\xi\,dx-\Delta x\sum_{i=-\infty}^{\infty}G(x_j-x_i,\varepsilon\Delta t)(\bar{v}_i^{*}-\bar{v}_j^{*})\right|,$$

which is the error of the "parabolic" substep of our method. Then, the following theorem holds.

Theorem 2.1 *Let us assume that $\widetilde{v}^*$ is a second-order piecewise linear approximation of the function $v\in C^2(\mathbb{R})$, whose second derivative is uniformly bounded in $\mathbb{R}$. Then the error $E_{\mathcal{P}}(\Delta t)$ of the "parabolic" substep in the FEOS method is:*

$$E_{\mathcal{P}}(\Delta t)\le C_1(\Delta x)^2+C_2\frac{(\Delta x)^4}{(\varepsilon\Delta t)^2}+C_3\frac{(\Delta x)^4}{(\varepsilon\Delta t)^{3/2}},\tag{2.12}$$

where C_1,C_2, and C_3 are absolute constants that are independent of $\Delta t,\Delta x$, and ε.

Proof: We first rewrite $E_{\mathcal{P}}(\Delta t)$ as follows:

$$\begin{aligned}E_{\mathcal{P}}(\Delta t)\;=\;&\frac{1}{\Delta x}\max_{j\in\mathbb{Z}}\left|\sum_{i=-\infty}^{\infty}\Big[\int\limits_{x_{j-\frac12}}^{x_{j+\frac12}}\int\limits_{x_{i-\frac12}}^{x_{i+\frac12}}G(x-\xi,\varepsilon\Delta t)(\widetilde{v}^*(\xi)-\widetilde{v}^*(x))\,d\xi\,dx\right.\\&\left.-(\Delta x)^2G(x_j-x_i,\varepsilon\Delta t)(\bar{v}_i^{*}-\bar{v}_j^{*})\Big]\right|.\end{aligned}$$

Then, we fix the point (x_j,x_i), denote by $\mathcal{I}_{j,i}:=[x_{j-\frac12},x_{j+\frac12}]\times[x_{i-\frac12},x_{i+\frac12}]$ the corresponding square centered at this point, and consider the Taylor expansion up to order four of a sufficiently smooth function $h(x,\xi)$ about this point. Integrating the Taylor expansion of h over $\mathcal{I}_{j,i}$, and using the fact that

$$\int\limits_{x_{j-\frac12}}^{x_{j+\frac12}}(x-x_j)^{2k+1}\,dx=0,\quad\int\limits_{x_{i-\frac12}}^{x_{i+\frac12}}(\xi-x_i)^{2k+1}\,d\xi=0,\quad k=0,1,\ldots,$$

we obtain the local midpoint quadrature rule for the cell $\mathcal{I}_{j,i}$:

$$\int\limits_{\mathcal{I}_{j,i}}h(x,\xi)\,d\xi\,dx=h(x_j,x_i)(\Delta x)^2+E_{j,i}+\widetilde{E}_{j,i}.$$

Here, the two components of the error are:

$$E_{j,i} = K_1(\Delta x)^4 \left(\frac{\partial^2 h}{\partial x^2}(x_j, x_i) + \frac{\partial^2 h}{\partial \xi^2}(x_j, x_i) \right), \tag{2.13}$$

and

$$|\widetilde{E}_{j,i}| \le K_2(\Delta x)^6 \max_{(x,\xi)\in\mathcal{I}_{j,i}} \left\{ \left| \frac{\partial^4 h}{\partial x^\alpha \partial \xi^\beta}(x,y) \right|,\ \alpha+\beta \le 4,\ \alpha \ge 0,\ \beta \ge 0 \right\}, \tag{2.14}$$

where the constants K_1 and K_2 are independent of both Δx and h. Then we have

$$E_{\mathcal{P}}(\Delta t) \le \frac{1}{\Delta x} \max_j \left| \sum_{i=-\infty}^{\infty} E_{j,i} + \sum_{i=-\infty}^{\infty} \widetilde{E}_{j,i} \right| \le \max_j E_j + \max_j \widetilde{E}_j, \tag{2.15}$$

where the following notation has been used:

$$E_j := \frac{1}{\Delta x} \left| \sum_{i=-\infty}^{\infty} E_{j,i} \right|, \qquad \widetilde{E}_j := \frac{1}{\Delta x} \sum_{i=-\infty}^{\infty} \left| \widetilde{E}_{j,i} \right|. \tag{2.16}$$

We now take $h(x,\xi) = G(x-\xi, \varepsilon\Delta t)(\widetilde{v}^*(\xi) - \widetilde{v}^*(x))$, which is a C^∞ function on the domain $(x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}) \times (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$, and estimate the errors E_j and $\widetilde{E}_j$. To this end, we first compute the second derivatives of h:

$$\frac{\partial^2 h}{\partial x^2}(x_j, x_i) = G_{zz}(x_j - x_i, \varepsilon\Delta t)(\bar{v}_i^* - \bar{v}_j^*) - 2G_z(x_j - x_i, \varepsilon\Delta t)s_j,$$

$$\frac{\partial^2 h}{\partial \xi^2}(x_j, x_i) = G_{zz}(x_j - x_i, \varepsilon\Delta t)(\bar{v}_i^* - \bar{v}_j^*) - 2G_z(x_j - x_i, \varepsilon\Delta t)s_i,$$

and after plugging them into (2.13) and (2.16), we obtain

$$E_j = 2K_1(\Delta x)^3 \left| \sum_{i=-\infty}^{\infty} G_{zz}(x_j - x_i, \varepsilon\Delta t)(\bar{v}_i^* - \bar{v}_j^*) - \sum_{i=-\infty}^{\infty} G_z(x_j - x_i, \varepsilon\Delta t)(s_i + s_j) \right|.$$

Using the fact that $G_z(z, \varepsilon\Delta t)$ is an odd function of z, $G_{zz}(z, \varepsilon\Delta t)$ is an even function of z, and that $G_z(0, \varepsilon\Delta t) = 0$, and applying the triangle inequality, we bound E_j as follows:

$$E_j \le 2K_1(\Delta x)^3 \sum_{i=1}^{\infty} \left\{ |G_{zz}(i\Delta x, \varepsilon\Delta t)| \left| \frac{\bar{v}_{j+i}^* + \bar{v}_{j-i}^*}{2} - \bar{v}_j^* \right| + |G_z(i\Delta x, \varepsilon\Delta t)||s_{j-i} - s_{j+i}| \right\}. \tag{2.17}$$

Next, the Taylor formula applied to v and v' results in

$$\begin{aligned} \left| \frac{\bar{v}_{j+i}^* + \bar{v}_{j-i}^*}{2} - \bar{v}_j^* \right| &= \left| \frac{v(x_{j+i}) + v(x_{j-i})}{2} - v(x_j) + \mathcal{O}((\Delta x)^2) \right| \\ &= \left| \frac{(i\Delta x)^2}{2} v''(\zeta_i) + \mathcal{O}((\Delta x)^2) \right| \le C_1(i\Delta x)^2, \end{aligned} \tag{2.18}$$

and

$$|s_{j-i} - s_{j+i}| = |v'(x_{j-i}) - v'(x_{j+i}) + \mathcal{O}(\Delta x)| = |2(i\Delta x)v''(\kappa_i) + \mathcal{O}(\Delta x)| \le C_2(i\Delta x), \tag{2.19}$$

since v'' is uniformly bounded in $\mathbb{R}$. Then, substituting (2.18) and (2.19) into (2.17), and bounding the resulting series by the corresponding improper integral, we obtain

$$\begin{aligned} E_j &\le C(\Delta x)^3\sum_{i=1}^{\infty}\Big\{(i\Delta x)^2|G_{zz}(i\Delta x,\varepsilon\Delta t)|+(i\Delta x)|G_z(i\Delta x,\varepsilon\Delta t)|\Big\}\\ &\le C(\Delta x)^2\int\limits_0^{\infty}\Big\{z^2|G_{zz}(z,\varepsilon\Delta t)|+z|G_z(z,\varepsilon\Delta t)|\Big\}\,dz. \end{aligned} \tag{2.20}$$

Next, we compute the required z-derivatives of the "heat" kernel:

$$G_z(z,\varepsilon\Delta t)=K_1\frac{z}{(\varepsilon\Delta t)^{3/2}}e^{-\frac{z^2}{4\varepsilon\Delta t}},\quad G_{zz}(z,\varepsilon\Delta t)=\left(K_2\frac{1}{(\varepsilon\Delta t)^{3/2}}+K_3\frac{z^2}{(\varepsilon\Delta t)^{5/2}}\right)e^{-\frac{z^2}{4\varepsilon\Delta t}}, \tag{2.21}$$

and therefore, using the triangle inequality and making the substitution $s=\frac{z}{2\sqrt{\varepsilon\Delta t}}$, we obtain a uniform upper bound on the integral on the RHS of (2.20):

$$\int\limits_0^{\infty}\Big\{z^2|G_{zz}(z,\varepsilon\Delta t)|+z|G_z(z,\varepsilon\Delta t)|\Big\}\,dz\le C_1\int\limits_0^{\infty}(s^4+s^2)e^{-s^2}\,ds=C_2.$$

Hence, the first component of the error $E_{\mathcal{P}}(\Delta t)$ in (2.15) is bounded by

$$E_j\le C_1(\Delta x)^2. \tag{2.22}$$

We proceed now with estimating $\widetilde{E}_j$. First, we note that

$$\begin{aligned} \max_{(x,\xi)\in\mathcal{I}_{j,i}}\left|\frac{\partial^4 h}{\partial x^\alpha\partial\xi^\beta}(x,\xi)\right| &\le \max_{(x,\xi)\in\mathcal{I}_{j,i}}|G_{zzzz}(x-\xi,\varepsilon\Delta t)|\cdot\max_{x\in\mathbb{R}}|\widetilde{v}^*(x)|\\ &\quad+\max_{(x,\xi)\in\mathcal{I}_{j,i}}|G_{zzz}(x-\xi,\varepsilon\Delta t)|\cdot\max_{m\in\mathbb{Z}}|s_m|\\ &= \max_{z\in((j-i-1)\Delta x,(j-i+1)\Delta x)}|G_{zzzz}(z,\varepsilon\Delta t)|\cdot\max_{x\in\mathbb{R}}|\widetilde{v}^*(x)|\\ &\quad+\max_{z\in((j-i-1)\Delta x,(j-i+1)\Delta x)}|G_{zzz}(z,\varepsilon\Delta t)|\cdot\max_{m\in\mathbb{Z}}|s_m|. \end{aligned}$$

Then, we use this estimate in (2.14) and (2.16) to obtain

$$\begin{aligned} \widetilde{E}_j &\le C(\Delta x)^5\Bigg[\max_{x\in\mathbb{R}}|\widetilde{v}^*(x)|\sum_{i=-\infty}^{\infty}\max_{z\in((j-i-1)\Delta x,(j-i+1)\Delta x)}|G_{zzzz}(z,\varepsilon\Delta t)|\\ &\quad+\max_{m\in\mathbb{Z}}|s_m|\sum_{i=-\infty}^{\infty}\max_{z\in((j-i-1)\Delta x,(j-i+1)\Delta x)}|G_{zzz}(z,\varepsilon\Delta t)|\Bigg]\\ &\le C(\Delta x)^5\Bigg[\max_{x\in\mathbb{R}}|v^*(x)|\sum_{\ell=-\infty}^{\infty}\max_{z\in(\ell\Delta x,(\ell+2)\Delta x)}|G_{zzzz}(z,\varepsilon\Delta t)|\\ &\quad+\max_{m\in\mathbb{Z}}|s_m|\sum_{\ell=-\infty}^{\infty}\max_{z\in(\ell\Delta x,(\ell+2)\Delta x)}|G_{zzz}(z,\varepsilon\Delta t)|\Bigg]. \end{aligned} \tag{2.23}$$

Next, we calculate the third and the fourth z-derivatives of G and bound them as follows:

$$|G_{zzz}(z,\varepsilon\Delta t)| \le \left(C_1\frac{|z|^3}{(\varepsilon\Delta t)^{7/2}} + C_2\frac{|z|}{(\varepsilon\Delta t)^{5/2}}\right)e^{-\frac{z^2}{4\varepsilon\Delta t}},$$

$$|G_{zzzz}(z,\varepsilon\Delta t)| \le \left(C_3\frac{z^4}{(\varepsilon\Delta t)^{9/2}} + C_4\frac{z^2}{(\varepsilon\Delta t)^{7/2}} + C_5\frac{1}{(\varepsilon\Delta t)^{5/2}}\right)e^{-\frac{z^2}{4\varepsilon\Delta t}}.$$

These bounds are then used to estimate the sums on the RHS of (2.23):

$$\begin{aligned}\sum_{\ell=-\infty}^{\infty}\max_{z\in(\ell\Delta x,(\ell+2)\Delta x)}|G_{zzzz}(z,\varepsilon\Delta t)| &\le C_6\sum_{\ell=0}^{\infty}\left\{\frac{((\ell+2)\Delta x)^4}{(\varepsilon\Delta t)^{9/2}} + \frac{((\ell+2)\Delta x)^2}{(\varepsilon\Delta t)^{7/2}} + \frac{1}{(\varepsilon\Delta t)^{5/2}}\right\}e^{-\frac{(\ell\Delta x)^2}{4\varepsilon\Delta t}}\\ &\le \frac{C_7}{\Delta x(\varepsilon\Delta t)^2}\int_0^{\infty}(s^4+s^2+1)e^{-s^2}\,ds = \frac{C_8}{\Delta x(\varepsilon\Delta t)^2}, \qquad (2.24)\end{aligned}$$

and similarly,

$$\begin{aligned}\sum_{\ell=-\infty}^{\infty}\max_{z\in(\ell\Delta x,(\ell+2)\Delta x)}|G_{zzz}(z,\varepsilon\Delta t)| &\le C_9\sum_{\ell=0}^{\infty}\left\{\frac{((\ell+2)\Delta x)^3}{(\varepsilon\Delta t)^{7/2}} + \frac{(\ell+2)\Delta x}{(\varepsilon\Delta t)^{5/2}}\right\}e^{-\frac{(\ell\Delta x)^2}{4\varepsilon\Delta t}}\\ &\le \frac{C_{10}}{\Delta x(\varepsilon\Delta t)^{3/2}}\int_0^{\infty}(s^3+s)e^{-s^2}\,ds = \frac{C_{11}}{\Delta x(\varepsilon\Delta t)^{3/2}}. \qquad (2.25)\end{aligned}$$

Substituting (2.24) and (2.25) into (2.23) results in

$$\widetilde{E}_j \le C_2\frac{(\Delta x)^4}{(\varepsilon\Delta t)^2}\max_{x\in\mathbb{R}}|\widetilde{v}^*(x)| + C_3\frac{(\Delta x)^4}{(\varepsilon\Delta t)^{3/2}}\max_{m\in\mathbb{Z}}|s_m|. \qquad (2.26)$$

Finally, we note that by the assumptions of the theorem both the interpolant $\widetilde{v}^*(x)$ and the slopes s_m are uniformly bounded. Thus, combining the estimates (2.22) and (2.26), we obtain the desired error bound (2.12), and the proof is completed. ■

Next, we establish a bound on the error of the "parabolic" substep in the case of *nonsmooth initial data* and/or a *nonlinear flux* $\mathbf{f}$ in (1.1). In this case, the piecewise linear approximation $\widetilde{v}^*$ is still uniformly bounded in $\mathbb{R}$, but its slopes may be large (proportional to $1/\Delta x$) due to either the initial layer or the breakdown of the solution at the "hyperbolic" substep. Therefore, the accuracy of the midpoint quadrature used in the "parabolic" substep reduces as quantified in the following theorem.

Theorem 2.2 *Let us assume that $\widetilde{v}^*$ is a (formally) second-order piecewise linear approximation of the piecewise smooth function v, and $|\widetilde{v}^*(x)| \le C$ for all $x \in \mathbb{R}$ and $|s_j| \le C(\Delta x)^{-1}$ for all j. Then the error $E_{\mathcal{P}}(\Delta t)$ of the "parabolic" substep in the FEOS method is:*

$$E_{\mathcal{P}}(\Delta t) \le C\left(\frac{\Delta x}{\sqrt{\varepsilon\Delta t}}\right), \quad \text{provided} \quad \frac{\Delta x}{\sqrt{\varepsilon\Delta t}} \le 1, \qquad (2.27)$$

where C is an absolute constant that is independent of Δt, Δx and ε.

Proof: We proceed as in the proof of Theorem 2.1 and obtain the estimate (2.17). However, since the function v is only piecewise smooth now, the bounds (2.18) and (2.19) cannot be established. We instead use the assumptions on $\widetilde{v}^*$ and its slopes, namely, $|\widetilde{v}^*(x)| \le C\,\forall x \in \mathbb{R}$ and $|s_j| \le C(\Delta x)^{-1}\,\forall j$, the formulae for G_z and G_{zz} in (2.21), and apply the substitution $s = \frac{z}{2\sqrt{\varepsilon\Delta t}}$ to obtain the following bound on E_j (compare with (2.20)):

$$\begin{aligned}
E_j &\le C_1(\Delta x)^3 \sum_{i=1}^{\infty} |G_{zz}(i\Delta x, \varepsilon\Delta t)| + C_2(\Delta x)^2 \sum_{i=1}^{\infty} |G_z(i\Delta x, \varepsilon\Delta t)| \\
&\le C_1(\Delta x)^2 \int_0^{\infty} |G_{zz}(z, \varepsilon\Delta t)|\,dz + C_2(\Delta x) \int_0^{\infty} |G_z(z, \varepsilon\Delta t)|\,dz \\
&\le \frac{C_1(\Delta x)^2}{\varepsilon\Delta t} \int_0^{\infty} (s^2+1)e^{-s^2}\,ds + \frac{C_2(\Delta x)}{\sqrt{\varepsilon\Delta t}} \int_0^{\infty} s e^{-s^2}\,ds \le C\left(\left(\frac{\Delta x}{\sqrt{\varepsilon\Delta t}}\right)^2 + \frac{\Delta x}{\sqrt{\varepsilon\Delta t}}\right).
\end{aligned}$$

Finally, combining this estimate with (2.26) and assuming that $\Delta x \le \sqrt{\varepsilon\Delta t}$, we obtain the error bound (2.27). This completes the proof of the theorem. ■

3 Experimental Convergence Study

In this section, we present a numerical study of the convergence rates of the FEOS method. We consider the 1-D scalar linear convection-diffusion equation subject to both smooth and nonsmooth initial data and the nonlinear Burgers' equation. We observe a second-order convergence rate for the linear equation with smooth initial data, which correlates reasonably well with our theoretical results in §2.1. In the case of nonsmooth initial data, the order of convergence reduces to the first one. In the nonlinear case, we surprisingly recover the same second order (in the L^1-norm) as in the linear case. We also observe that the convergence rates are not too sensitive to the change in the size of the splitting steps (Δt).

3.1 1-D Linear Convection-Diffusion Equation

We apply the FEOS method to the 1-D linear convection-diffusion equation:

$$u_t + u_x = \varepsilon u_{xx}, \qquad \varepsilon = 0.01, \tag{3.1}$$

subject to the compactly supported initial data:

$$u(x,0) = u_0(x). \tag{3.2}$$

We take the computational domain to be $[-1.5, 1.5]$ and the final time to be $t = 0.2$.

The exact solution of the IVP (3.1)–(3.2) is available and is given by

$$u(x,t) := \frac{1}{2\sqrt{\pi\varepsilon t}} \int_{-\infty}^{\infty} e^{-\frac{(x-t-\xi)^2}{4\varepsilon t}} u_0(\xi)\,d\xi.$$

A numerical convergence study is performed by calculating the L^∞-error (the required values of the exact solution are computed using the fourth-order Simpson quadrature over the uniform grid with 12800 cells). In the next two subsections, we present the experimental convergence rates in the case of smooth and nonsmooth initial data.

3.1.1 Smooth Initial Data

We solve equation (3.1) subject to the smooth initial data:

$$u(x,0)=\begin{cases}\cos^4\left(\frac{\pi x}{2}\right), & \text{if } |x|\le 1,\\ 0, & \text{otherwise.}\end{cases} \tag{3.3}$$

Since the solution of the IVP (3.1),(3.3) as well as the "intermediate solutions" obtained at the "hyperbolic" substeps are globally smooth, the nonlinear limiting mechanism in the hyperbolic solver (the central-upwind scheme) has been switched off.

The size of the splitting steps is taken $\Delta t = C\varepsilon^{-1}(\Delta x)^\alpha$. The corresponding convergence rates for different values of α are shown in Table 3.1, where the second order of convergence can be clearly observed for both the (theoretically) optimal value of $\alpha = 2/3$ and the non-optimal ones, $\alpha = 1/3$ and $\alpha = 1$.

Number of	$\alpha=\frac{1}{3}$		$\alpha=\frac{2}{3}$		$\alpha=1$	
grid cells	L^∞-error	Rate	L^∞-error	Rate	L^∞-error	Rate
100	2.41e-02	–	3.17e-02	–	2.46e-02	–
200	5.75e-03	2.07	5.71e-03	2.47	5.80e-03	2.08
400	1.42e-03	2.02	1.42e-03	2.00	1.42e-03	2.03
800	3.51e-04	2.01	3.51e-04	2.02	3.51e-04	2.02
1600	8.75e-05	2.00	8.75e-05	2.00	8.75e-05	2.00
3200	2.19e-05	2.00	2.19e-05	2.00	2.19e-05	2.00

Table 3.1: Linear convection-diffusion equation with smooth initial data — the L^∞-errors.

3.1.2 Nonsmooth Initial Data

Next, we numerically solve equation (3.1) subject to the following nonsmooth initial condition:

$$u(x,0)=\begin{cases}1, & \text{if } -1\le x<0,\\ -1, & \text{if } 0\le x<1,\\ 0, & \text{otherwise,}\end{cases}$$

using the FEOS method. Since initially the solution contains very large gradients, we apply the (nonlinear) minmod limiter (2.2) with $\theta = 2$ in the beginning, but after $t = 5\Delta t_{\text{HYP}}$ the limiters

are safely switched off since the equation (3.1) is linear and thus does not have any built-in breakdown mechanisms.

In Table 3.2 and 3.3, we present the computed L^∞- and L^1-errors and the corresponding numerical convergence rates. We note that since in the case of nonsmooth initial data our error analysis is not complete, it is not clear what the optimal value of α should be. If we assume that the accuracy of one Strang splitting step is the same as in the smooth case, then balancing the error bound in (2.27) with the splitting error of size $(\varepsilon\Delta t)^3$ results in the optimal $\alpha = 2/7$. However, it is natural to expect that in this case the order of the splitting algorithm would be smaller and then balancing $\Delta x/\sqrt{\varepsilon\Delta t}$ with $(\varepsilon\Delta t)^2$ would give us another optimal $\alpha = 2/5$. One can observe the convergence rate of about $0.8 \div 0.9$ in the L^∞-norm and of 1 in the L^1-norm. As in the case of smooth initial data (§3.1.1), these rates are higher than expected, but the effect of boundary layer can be clearly seen via comparison of the data in Tables 3.2 and 3.3 with the results reported in Table 3.1.

Number of	$\alpha = \frac{1}{7}$		$\alpha = \frac{2}{7}$		$\alpha = \frac{2}{5}$		$\alpha = \frac{4}{5}$	
grid cells	L^∞-error	Rate	L^∞-error	Rate	L^∞-error	Rate	L^∞-error	Rate
100	7.07e-01	–	5.51e-01	–	4.32e-01	–	5.75e-01	–
200	2.97e-01	1.25	2.99e-01	0.88	3.12e-01	0.47	3.24e-01	0.83
400	1.72e-01	0.79	1.70e-01	0.82	1.71e-01	0.87	1.86e-01	0.80
800	9.85e-02	0.81	9.71e-02	0.81	9.69e-02	0.82	9.33e-02	1.00
1600	6.22e-02	0.67	5.05e-02	0.94	5.92e-02	0.71	5.33e-02	0.81
3200	3.73e-02	0.74	2.89e-02	0.81	3.11e-02	0.93	3.13e-02	0.76

Table 3.2: Linear convection-diffusion equation with nonsmooth initial data — the L^∞-errors.

Number of	$\alpha = \frac{1}{7}$		$\alpha = \frac{2}{7}$		$\alpha = \frac{2}{5}$		$\alpha = \frac{4}{5}$	
grid cells	L^1-error	Rate	L^1-error	Rate	L^1-error	Rate	L^1-error	Rate
100	2.21e-01	–	1.68e-01	–	1.39e-01	–	1.70e-01	–
200	6.10e-02	1.85	6.58e-02	1.35	6.46e-02	1.11	6.35e-02	1.43
400	3.34e-02	0.87	3.34e-02	0.98	3.31e-02	0.96	3.35e-02	0.92
800	1.73e-02	0.95	1.72e-02	0.96	1.72e-02	0.95	1.73e-02	0.96
1600	8.84e-03	0.97	8.79e-03	0.97	8.79e-03	0.97	8.75e-03	0.98
3200	4.49e-03	0.98	4.44e-03	0.98	4.43e-03	0.99	4.40e-03	0.99

Table 3.3: Linear convection-diffusion equation with nonsmooth initial data — the L^1-errors.

3.2 Burgers' Equation

In this section, we use the FEOS method to compute the solution of the nonlinear 1-D Burgers' equation:

$$u_t + \left(\frac{u^2}{2}\right)_x = \varepsilon u_{xx}, \qquad \varepsilon = 0.01, \tag{3.4}$$

subject to the smooth compactly supported initial data (3.3). The computational domain is $[-1.5, 1.5]$, and the final time is $t = 1$, which is a post-shock time for the corresponding inviscid Burgers' equation. Unlike the case of a linear convection-diffusion equation, the splitting solution may now break down at the "hyperbolic" substep. Therefore, the hyperbolic solver employs the minmod limiter (2.2) with $\theta = 2$ at each time step.

The exact solution of the IVP (3.4),(3.3) is given by the Hopf formula:

$$u(x,t) = \frac{\int_{-\infty}^{\infty}(x-\xi)e^{-\frac{\lambda(x,\xi,t)}{2\varepsilon}}\,d\xi}{\int_{-\infty}^{\infty} te^{-\frac{\lambda(x,\xi,t)}{2\varepsilon}}\,d\xi}, \qquad \lambda(x,\xi,t) := \frac{(x-\xi)^2}{2t} + \int_0^{\xi} u_0(\eta)\,d\eta.$$

Its point values, needed to calculate the errors, are obtained using the fourth-order Simpson quadrature over a uniform grid with 12800 cells. The computed L^∞- and L^1-errors are shown in Tables 3.4 and 3.5. Once again, the optimal value of α in $\Delta t = C\varepsilon^{-1}(\Delta x)^\alpha$ is not known, but we observe that the convergence rates are now larger for larger α (and thus, for smaller Δt). Even though the formal proof is unavailable, this phenomenon is quite expected since the solution of the inviscid Burgers' equation, solved at the "hyperbolic" substeps, typically breaks down after a certain time, and when Δt is relatively small, the "hyperbolic" solution remains smooth. This pushes the orders of convergence all the way up to second order. However, when α is small, that is, when the number of splitting steps is small, the accuracy reduces to first order, which has also been observed in the linear case with nonsmooth initial data. At the same time, we would like to point out that in the latter case, the error is still very small. This suggests that our FEOS method may be efficiently applied to more interesting problems (see §4), for which the ability of the method to achieve a reasonably high accuracy while performing only few splitting steps is a very important feature.

4 Application to the Polymer System

In this section, we apply our FEOS method to 1-D and 2-D models describing flow in porous media. We numerically solve the systems of convection-diffusion equations (4.1) and (4.6), see §4.1 and §4.2 below. These systems model polymer flooding processes in enhanced oil recovery (see [2, 12, 16] and the references therein). The initial data in our examples are taken form [3] and [1], and thus a comparison of our method with some existing alternative methods can be made.

Number of	$\alpha=\frac{1}{7}$		$\alpha=\frac{2}{7}$		$\alpha=\frac{2}{5}$		$\alpha=\frac{4}{5}$	
grid cells	L^∞-error	Rate	L^∞-error	Rate	L^∞-error	Rate	L^∞-error	Rate
100	1.99e-02	–	1.98e-02	–	2.29e-02	–	2.31e-02	–
200	6.11e-03	1.71	6.10e-03	1.70	6.06e-03	1.92	5.78e-03	2.00
400	1.56e-03	1.97	1.55e-03	1.97	1.55e-03	1.97	1.48e-03	1.96
800	3.91e-04	2.00	3.90e-04	2.00	3.90e-04	1.99	3.91e-04	1.92
1600	1.50e-04	1.37	1.20e-04	1.70	1.18e-04	1.72	1.07e-04	1.87
3200	8.60e-04	0.78	5.37e-05	1.16	4.68e-05	1.33	1.93e-05	1.87

Table 3.4: Burgers' equation — the L^∞-errors.

Number of	$\alpha=\frac{1}{7}$		$\alpha=\frac{2}{7}$		$\alpha=\frac{2}{5}$		$\alpha=\frac{4}{5}$	
grid cells	L^∞-error	Rate	L^∞-error	Rate	L^∞-error	Rate	L^∞-error	Rate
100	2.16e-03	–	2.06e-03	–	2.50e-03	–	2.57e-03	–
200	6.76e-04	1.68	6.74e-04	1.61	6.23e-04	1.90	6.62e-04	1.96
400	1.75e-04	1.94	1.76e-04	1.94	1.75e-04	1.94	1.76e-04	1.91
800	4.59e-05	1.93	4.57e-05	1.94	4.57e-05	1.94	4.59e-05	1.94
1600	1.30e-05	1.82	1.23e-05	1.90	1.22e-05	1.90	1.19e-05	1.95
3200	6.69e-06	0.96	4.27e-06	1.52	3.80e-06	1.69	3.07e-06	1.96

Table 3.5: Burgers' equation — the L^1-errors.

4.1 One-Dimensional Examples

We first consider the 1-D system of two convection-diffusion equations:

$$\begin{cases} s_t + f(s,c)_x = \varepsilon s_{xx} \\ b_t + (cf(s,c))_x = \varepsilon b_{xx}, \end{cases} \tag{4.1}$$

with $b = b(s,c) = sc + a(c)$, subject to the initial data $s(x,0) = s_0(x), c(x,0) = c_0(x)$. Here, $(s,c)^T$ is the unknown state vector, $\varepsilon > 0$ is a small scaling parameter, and

$$f := f(s,c) = \frac{s^2}{s^2 + \mu(1+\nu c)(1-s)^2}, \qquad a := a(c) = \frac{c}{5(1+c)}. \tag{4.2}$$

In all the numerical experiments, presented in this section, we take $\mu = 1/2$ and $\nu = 2$.

We will compare the numerical solution computed by the FEOS method with a reference solution obtained without any operator splitting by combining the second-order central-upwind scheme with the explicit second-order central-difference approximation for the diffusion term in (4.1). In all our numerical experiments, we will take the value of the minmod (see (2.2)) parameter $\theta = 1$, since the flux here is nonconvex and, as it has been demonstrated in [7], the

use of compressive minmod limiter with $\theta > 1$ may lead to a convergence to a "wrong" solution that does not satisfy the entropy condition for all entropies. Obviously, this will not be a problem when ε is large, but we are focusing on the convection-dominated regime, in which a large error at the "hyperbolic" substep cannot be "fixed" by a small diffusion acting at the "parabolic" substep.

Example 1. In the first example, we consider the polymer system (4.1) subject to the following discontinues initial data:

$$(s_0, c_0)(x) = \begin{cases} (1.0, 0.5), & x \le 0.25, \\ (0.1, 0.1), & x > 0.25. \end{cases} \tag{4.3}$$

In the inviscid case, these initial data correspond to a Riemann problem, whose solution consists of an s-shock, followed by a c-shock and an s-rarefaction wave.

In Figure 4.1, we plot the approximate solutions of (4.1),(4.3) (dotted line) at time $t = 1$ for $\varepsilon = 0.001$ and $\varepsilon = 0.01$, computed by the FEOS method with two splitting steps and 500 uniform grid cells. The solid line represents a reference solution computed with 10000 cells. As one can clearly see, for $\varepsilon = 0.001$ the computed solution agrees well with the reference one, while for $\varepsilon = 0.01$ the s-component of the solution is smeared. Therefore, one needs to perform more than two splitting steps. In Figure 4.2, the solutions computed for $\varepsilon = 0.01$ by the FEOS method with 8 and 32 splitting steps are shown. Now the resolution of both s and c fronts is very high.

We also study the convergence rate of the FEOS method with respect to the number of splitting steps. In order to do this, we fix the spatial mesh to 1000 uniform cells and increase the number of splitting steps by powers of two. We then compute relative errors in the L^∞ and L^1 norms for each component of the solution according to the following formulae:

$$E_\infty = \frac{||u - u^{\text{ref}}||_\infty}{||u^{\text{ref}}||_\infty}, \qquad E_1 = \frac{||u - u^{\text{ref}}||_1}{||u^{\text{ref}}||_1},$$

where u denotes one of the components of the splitting solution and u^{ref} denotes the corresponding component of the reference solution. Tables 4.1 and 4.2 show the errors (computed at time $t = 1$ in both norms for $\varepsilon = 0.001$ and $\varepsilon = 0.01$) and the convergence rates for the s- and c-components of the solution, respectively. As one can see, in all cases, the convergence rate start decreasing after a certain number of splitting steps since, as it follows from the error estimates (2.12) and (2.27), the splitting step in the FEOS method cannot be taken too small. A similar convergence study has been performed in [3] and we would like to point out that the relative errors obtained in the FEOS method are, on the average, ten times smaller than those obtained in [3] (we refer the reader to that work for comparison).

Example 2. Next, we consider system (4.1) subject to the following nonmonotone Riemann initial data:

$$(s_0, c_0)(x) = \begin{cases} (0.45, 0), & x \le 0.25, \\ (0.20, 1), & x > 0.25. \end{cases} \tag{4.4}$$

In the inviscid case, the solution of this IVP consists of five intermediate states, separated by simple waves (see, e.g., [2]). Figure 4.3 shows both components of the computed solution (dotted lines) at time $t = 1$ of the viscous case with $\varepsilon = 0.0025$, obtained using the FEOS method with two splitting steps only. The computed solutions are compared with the reference solutions (solid lines). As in Example 1, the approximate solutions are computed with 500 uniform grid cells

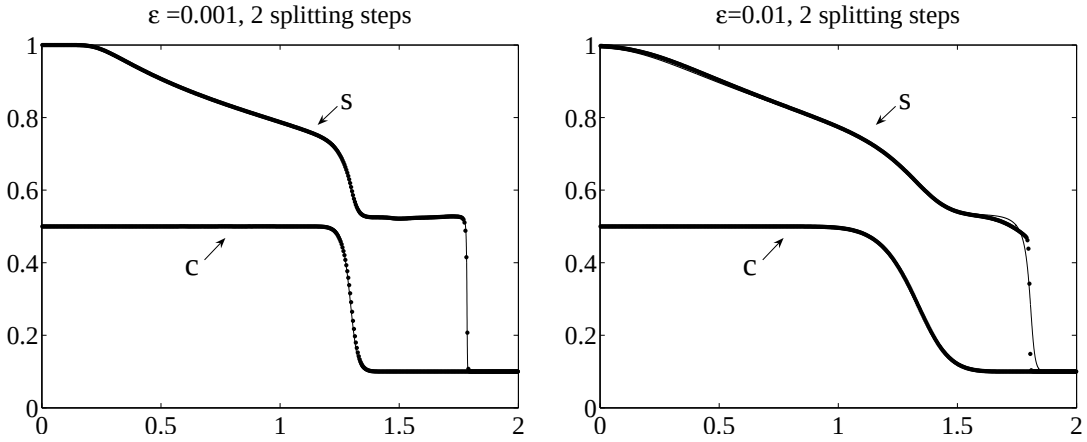

Figure 4.1: Solution of (4.1),(4.3) with $\varepsilon = 0.001$ and $\varepsilon = 0.01$ computed by the FEOS method with 2 splitting steps (dotted line). The solid line represents the reference solution.

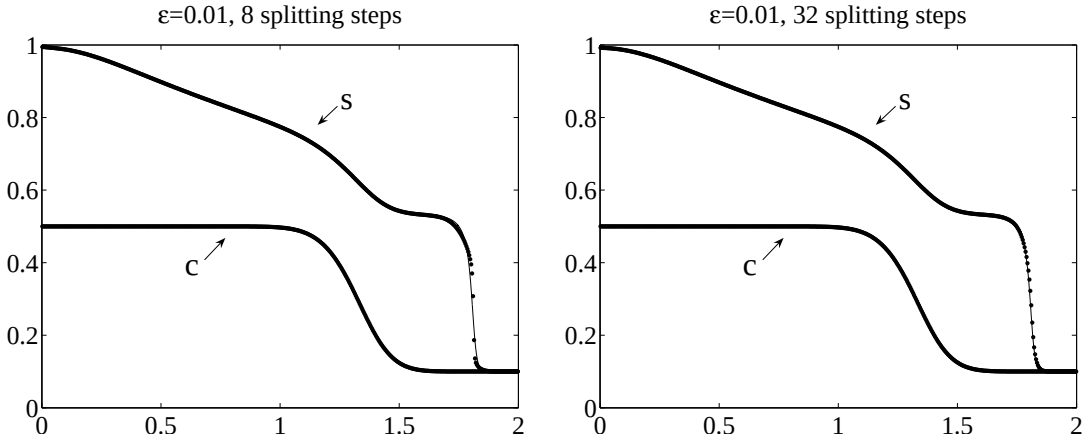

Figure 4.2: Solution of (4.1),(4.3) with $\varepsilon = 0.01$ computed by the FEOS method with 8 and 32 splitting steps (dotted line). The solid line represents the reference solution.

and the reference solution is computed with 10000 cells, using the second-order central-upwind scheme without splitting. As one can see, the implementation of the FEOS method results in a fairly accurate approximation of the c-wave while the s-component is again smeared. In order to demonstrate the convergence, we increase a number of splitting steps and show, in Figure 4.4, the s-component of the solution, computed by the FEOS method with 8 and 32 splitting steps.

Example 3. This example is a Riemann problem corresponding to a compressive shock in the inviscid case, in which both the s- and the c-characteristics go into the shock and contribute to its self-sharpening. The initial condition is given by

$$(s_0, c_0)(x) = \begin{cases} (0.75, 0.8), & x \le 0.25, \\ (0.839619, 0.4), & x > 0.25. \end{cases} \tag{4.5}$$

If this Riemann problem is slightly perturbed, the solution changes from a single shock to a composition of waves moving with almost the same speed (see, e.g., [3]). There are two possible

Number	$\varepsilon = 0.001$				$\varepsilon = 0.01$			
of steps	L^∞-error	Rate	L^1-error	Rate	L^∞-error	Rate	L^1-error	Rate
2	2.46e-01	–	2.53e-03	–	2.05e-01	–	8.80e-03	–
4	1.16e-01	1.10	1.66e-03	0.61	1.46e-01	0.50	5.94e-03	0.57
8	8.40e-02	0.46	1.09e-03	0.61	8.50e-02	0.78	3.47e-03	0.78
16	3.29e-02	1.35	7.74e-04	0.49	3.42e-02	1.31	1.57e-03	1.14
32	1.93e-02	0.77	6.14e-04	0.33	1.27e-02	1.43	5.96e-04	1.40
64	4.82e-02	-1.32	5.56e-04	0.14	4.87e-03	1.38	2.07e-04	1.52
128	5.60e-02	-0.21	5.20e-04	0.09	2.35e-03	1.04	9.05e-05	1.19
256	5.69e-02	-0.03	5.14e-04	0.01	1.70e-03	0.48	7.38e-05	0.30

Table 4.1: Example 1. Estimated errors and convergence rates for the s-component of the solution.

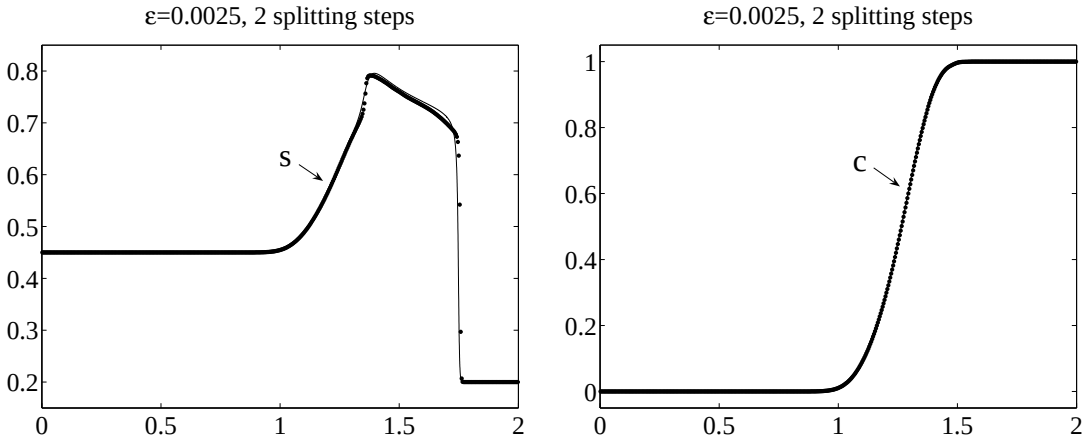

Figure 4.3: Solution of (4.1),(4.4) with $\varepsilon = 0.0025$ computed by the FEOS method with 2 splitting steps (dotted line). The solid line represents the reference solution.

results of the perturbation: either a monotone or a nonmonotone solution. In the viscous case, the problem will be perturbed instantly, which results in a truly nonlinear phenomenon: monotone initial data evolve into nonmonotone solutions. In Figure 4.5, we plot the approximate solutions of (4.1),(4.5) with $\varepsilon = 0.005$ (dotted lines) at time $t = 1$, computed by the FEOS with two and four splittings steps. As in the previous two Examples, we compare these solutions computed with 500 uniform grid cells with the corresponding reference solutions computed with 10000 cells. The exact (reference) solution has a dip due to the presence of the diffusion term. As one can observe, the dip in the s-component of the solution is not resolved well when only two or four splitting steps are performed. Therefore, in Figure 4.6, we show the results obtained with 8 and 32 splittings steps, where a very high resolution is achieved. We would like to point out that the alternative operator splitting methods, described in [3], fail to resolve the dip in the s-component of the solution (see Figures 10 and 11 in [3]).

Number	$\varepsilon = 0.001$				$\varepsilon = 0.01$			
of steps	L^∞-error	Rate	L^1-error	Rate	L^∞-error	Rate	L^1-error	Rate
2	1.34e-02	–	7.06e-04	–	1.31e-02	–	3.28e-03	–
4	9.63e-03	0.48	5.33e-04	0.41	7.93e-03	0.72	1.87e-03	0.81
8	7.27e-03	0.40	4.11e-04	0.38	4.05e-03	0.97	9.25e-04	1.02
16	5.70e-03	0.35	3.34e-04	0.30	1.74e-03	1.21	3.96e-04	1.23
32	4.66e-03	0.29	3.28e-04	0.02	6.04e-04	1.53	1.45e-04	1.44
64	5.17e-03	-0.15	3.54e-04	-0.11	1.43e-04	2.08	4.31e-05	1.75
128	5.96e-03	-0.21	3.77e-04	-0.09	8.97e-05	0.67	2.01e-05	1.10
256	6.42e-03	-0.10	3.92e-04	-0.05	9.38e-05	-0.06	2.58e-05	-0.36

Table 4.2: Example 1. Estimated errors and convergence rates for the c-component of the solution.

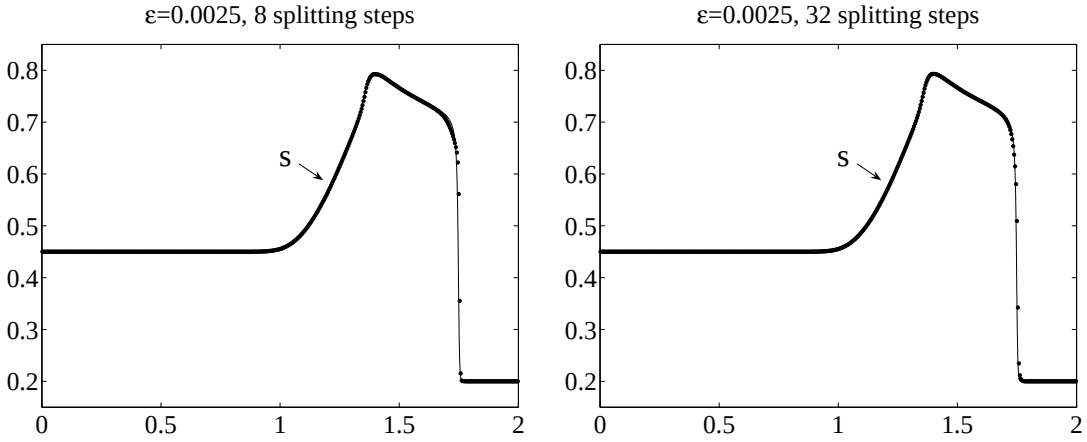

Figure 4.4: s-component of the solution of (4.1),(4.4) with $\varepsilon = 0.0025$ computed by the FEOS method with 8 and 32 splitting steps (dotted line). The solid line represents the reference solution.

4.2 Two-Dimensional Example

Finally, we consider the 2-D polymer system:

$$\begin{cases} s_t + f(s,c)_x + f(s,c)_y = \varepsilon(s_{xx} + s_{yy}) \\ b_t + (cf(s,c))_x + (cf(s,c))_y = \varepsilon(b_{xx} + b_{yy}), \end{cases} \tag{4.6}$$

where, as in the 1-D case, $b = b(s,c) = sc + a(c)$ and f and a are given by formula (4.2). We now take $\mu = \nu = 1$ and consider the 2-D Riemann problem with the following initial data:

$$(s,c)(x,y,0) = \begin{cases} (1.0, 0.0), & x < 0,\ y < 0, \\ (1.0, 0.1), & x > 0,\ y > 0, \\ (0.0, 0.0), & \text{otherwise.} \end{cases} \tag{4.7}$$

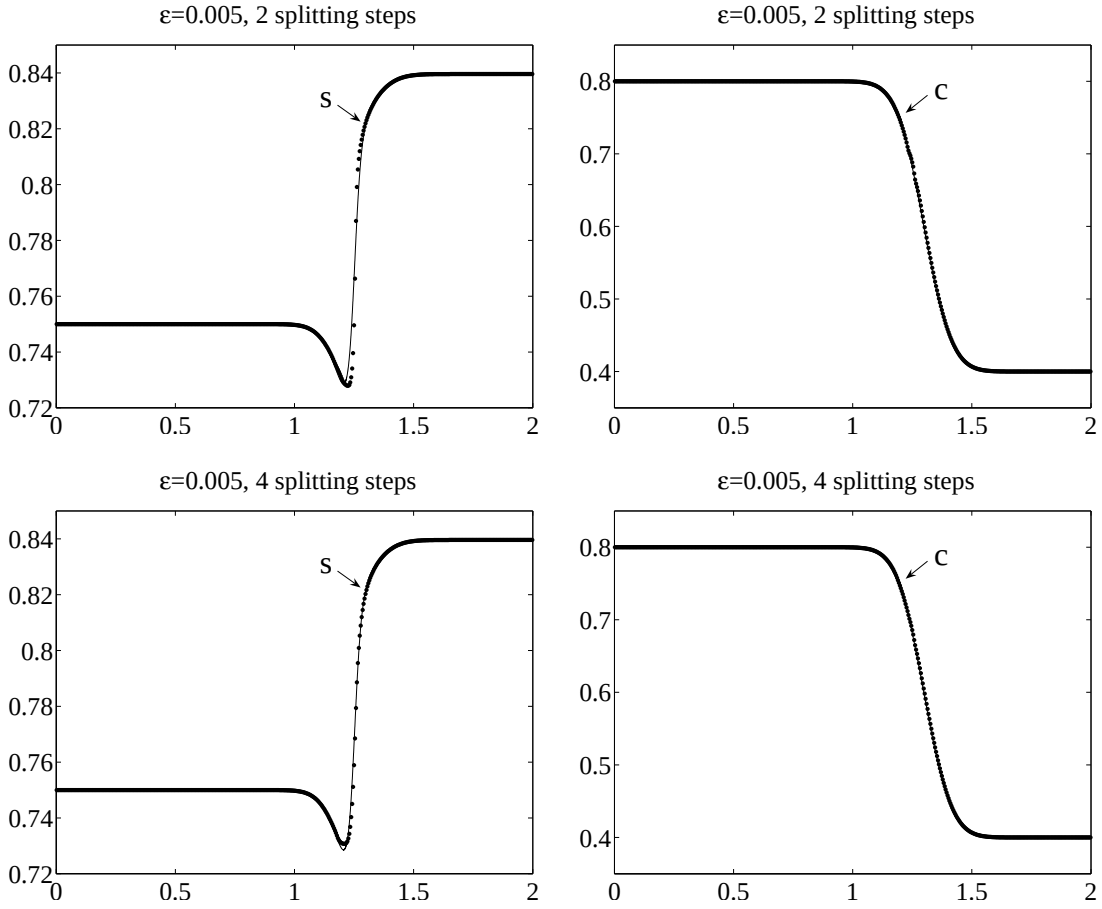

Figure 4.5: Solution of (4.1),(4.5) with $\varepsilon = 0.005$ computed by the FEOS method with 2 and 4 splitting steps (dotted line). The solid line represents the reference solution.

The example is taken from [1], where the corresponding inviscid system was numerically solved by a front tracking method. Here, we consider the viscous case with $\varepsilon = 0.01$. The solution, computed at time $t = 0.4$ by the FEOS method with 200×200 grid cells and four splitting steps, is plotted in Figure 4.7. In order to demonstrate convergence of the method, we also show, in Figure 4.8, the same solution but computed on 400×400 grid with two and four splittings steps. As one can see, all major waves are already accurately captured with four splittings steps for the s-component of the solution and with only two steps for the c-component of the solution.

It should be pointed out that a fast and efficient implementation of the FEOS method in two (and more) dimensions can only be achieved by taking into account the special form of the heat kernel given by (2.6). The presence of the exponents of the type $e^{-\frac{(x_j-x_i)^2+(y_k-y_\ell)^2}{4\varepsilon\Delta t}}$ on the RHS of the midpoint rule (2.9), used in the "parabolic" substep of the FEOS method, allows one to perform the summation only in a (relatively) small neighborhood of each cell, where the exponents do not practically vanish. This significantly reduces CPU times and thus makes the FEOS method very efficient.

Acknowledgment: The work of A. Chertock was supported in part by the NSF Grant # DMS-0410023. The work of A. Kurganov was supported in part by the NSF Grant # DMS-0310585.

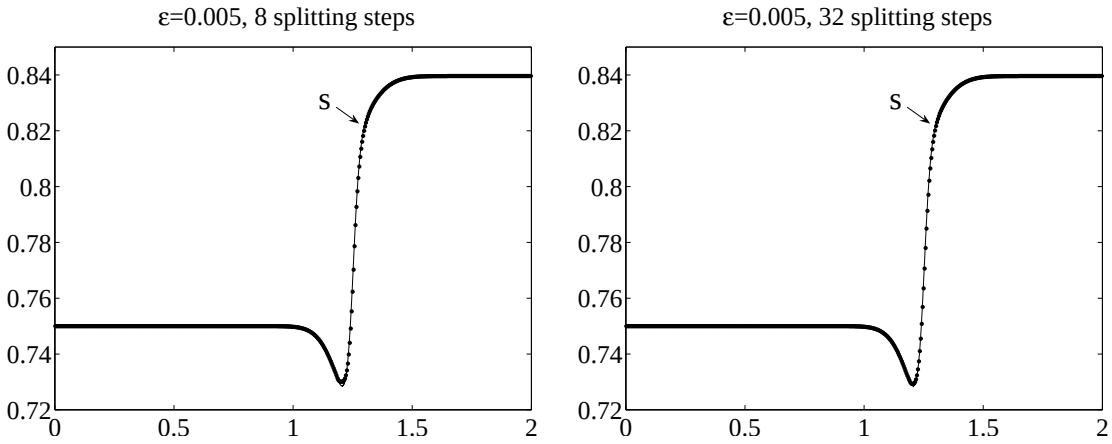

Figure 4.6: s-component of the solution of (4.1),(4.5) with $\varepsilon = 0.005$ computed by the FEOS method with 8 and 32 splitting steps (dotted line). The solid line represents the reference solution.

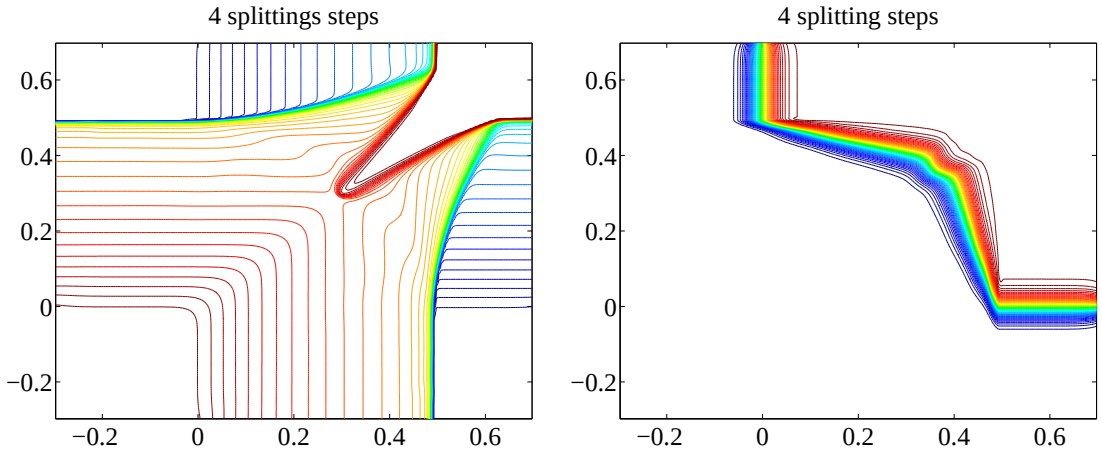

Figure 4.7: s (left) and c (right) components of the solution of (4.6)–(4.7) with $\varepsilon = 0.01$ computed by the FEOS method on a 200×200 grid with 4 splitting steps.

The work of G. Petrova was supported in part by the NSF Grant # DMS-0296020.

References

[1] V. HAUGSE, K.H. KARLSEN, K.-A. LIE AND J.R. NATVIG, *Numerical solution of the polymer system by front tracking*, Transp. Porous Media, 44 (2001), pp. 63–83.

[2] T. JOHANSEN AND R. WINTHER, *The solution of the Riemann problem for a hyperbolic system of conservation laws modeling polymer flooding*, SIAM J. Math. Anal., 19 (1988), pp. 541–566.

[3] K. H. KARLSEN, K.-A. LIE, J. R. NATVIG, H. F. NORDHAUG AND H. K. DAHLE, *Operator splitting methods for systems of convection-diffusion equations: nonlinear error mechanisms and correction strategies*, J. Comput. Phys., 173 (2001), pp. 636–663.

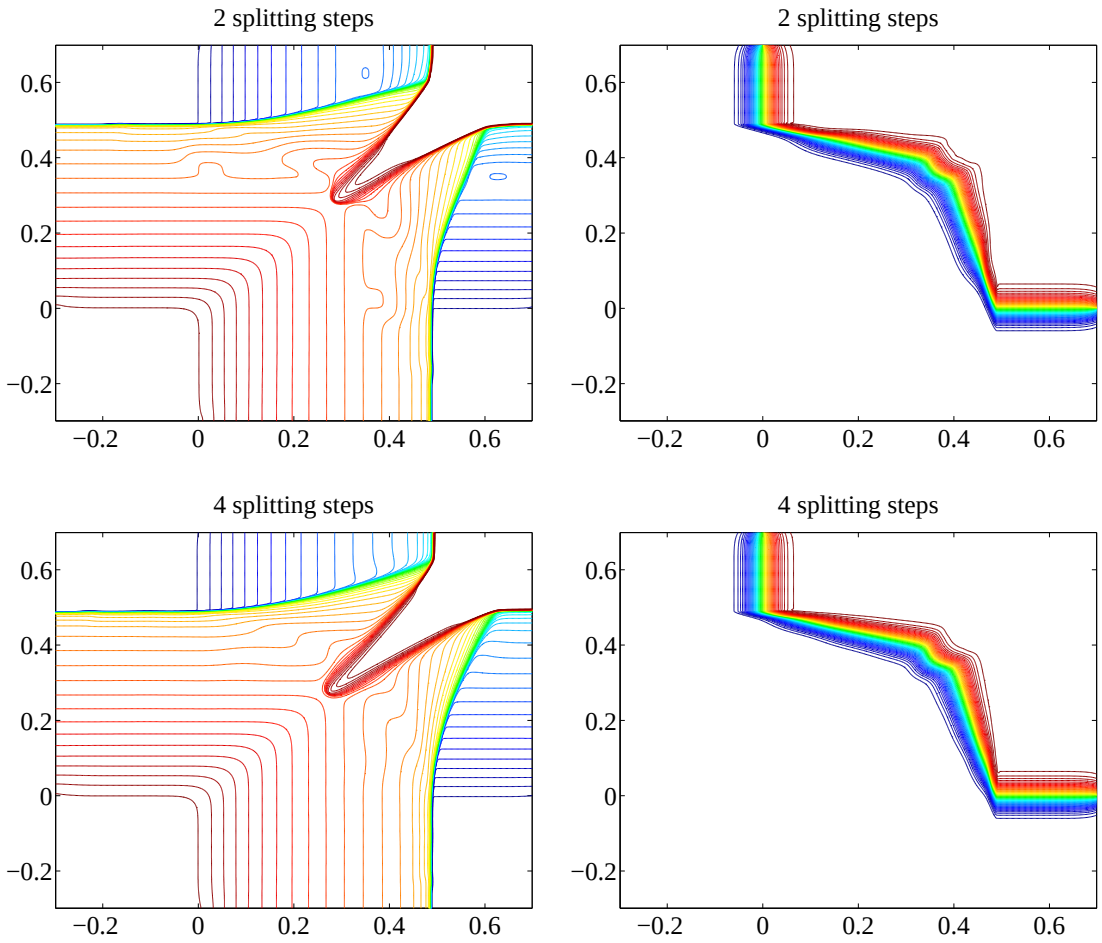

Figure 4.8: s (left) and c (right) components of the solution of (4.6)–(4.7) with $\varepsilon = 0.01$ computed by the FEOS method on a 400×400 grid with 2 and 4 splitting steps.

[4] A. KURGANOV AND C.-T. LIN, *On the reduction of numerical diss ipation in central-upwind schemes*, in preparation.

[5] A. KURGANOV, S. NOELLE AND G. PETROVA, *Semi-discrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 23 (2001), pp. 707–740.

[6] A. KURGANOV AND G. PETROVA, *Central schemes and contact discontinuities*, M2AN Math. Model. Numer. Anal., 34 (2000), pp. 1259–1275.

[7] A. KURGANOV, G. PETROVA AND B. POPOV, *Adaptive semi-discrete central-upwind schemes for nonconvex hyperbolic conservation laws*, submitted to SIAM J. Sci. Comput.

[8] A. KURGANOV AND E. TADMOR, *New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations*, J. Comput. Phys., 160 (2000), pp. 214–282.

[9] J. LEE AND B. FORNBERG, *A split step approach for the 3-D Maxwell's equations*, J. Comput. Appl. Math., 158 (2003), pp. 485–505.

[10] K.-A. Lie and S.Noelle, *On the artificial compression method for second-order nonoscillatory central difference schemes for systems of conservation laws*, SIAM J. Sci. Comput., 24 (2003), pp. 1157–1174.

[11] H. Nessyahu and E. Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys., 87 (1990), pp. 408–463.

[12] N. H. Risebro and A. Tveito, *Front tracking applied to a non-strictly hyperbolic system of conservation laws*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 1401–1419.

[13] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal., 5 (1968), pp. 506–517.

[14] M. Suzuki, *General theory of fractal path integrals with applications to many-body theories and statistical physics*, J. Math. Phys., 32 (1991), pp.400-407.

[15] P. K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Numer. Anal., 21 (1984), pp. 995–1011.

[16] A. Tveito, *Convergence and stability of the Lax-Friedrichs scheme for a nonlinear parabolic polymer flooding problem*, Adv. in Appl. Math., 11 (1990), pp. 220–246.

[17] H. Yoshida, *Construction of higher order symplectic integrators*, Phys. Lett. A, 150 (1990), pp. 262–268.