ITERATIVE ALGORITHM FOR KINETIC TYPE FOKKER-PLANCK EQUATION

Nizomjon Jumaniyazov
Mamun University, *nizomjon_jumaniyazov@yahoo.com*
Go'zal Boltayeva
RANCH University of Technology, *boltayevagozal1334@mail.ru*
Dilnoza Qodirova
School #17 in Khiva city, *kodirovadilnozaa@gmail.com*

https://doi.org/10.5281/zenodo.13926365

Abstract. *So far, the standard implementation of finite difference schemes for forward-backward partial differential equations has included in employing an iterative method. This paper proposes an iterative algorithm for Fokker-Planck type kinetic equation, which can be compared with the direct algorithm. An effective way of choosing the seed for the iterative method naturally arises.*

Keywords: *forward-backward partial differential equation, two-way partial differential equation, finite difference method, Fokker-Planck equation, direct method, iterative method.*

ИТЕРАЦИОННЫЙ АЛГОРИТМ ДЛЯ КИНЕТИЧЕСКОГО УРАВНЕНИЯ ФОККЕРА-ПЛАНКА

Аннотация. *До сих пор стандартная реализация конечно-разностных схем для дифференциальных уравнений в частных производных вперед-назад включала использование итерационного метода. В этой статье предлагается итерационный алгоритм для кинетического уравнения типа Фоккера-Планка, который можно сравнить с прямым алгоритмом. Эффективный способ выбора затравки для итерационного метода возникает естественным образом.*

Ключевые слова: *дифференциальное уравнение в частных производных вперед-назад, двухстороннее дифференциальное уравнение в частных производных, метод конечных разностей, уравнение Фоккера-Планка, прямой метод, итерационный метод.*

1 Introduction

The current paper has been motivated by reference [11], which proposes several finite difference schemes for solving certain problem of interest to the nuclear engineering community. To specify, the problem is the one given on $Q = [-1;1]\times\left[Z_{ini};\ Z_{fin}\right]$ by the Fokker-Planck equation

$$\mu\frac{\partial\psi}{\partial z}+\alpha\psi-\sigma\frac{\partial}{\partial\mu}\left[D(\mu)\frac{\partial\psi}{\partial\mu}\right]=W,\quad \text{for } (\mu, z)\in Q \tag{1}$$

being $D(\mu)=1-\mu^2$ and the incoming flux boundary conditions

$$\psi(\mu, z_0)=f(\mu) \text{ for } \mu\in(0;1], \tag{2}$$

$$\psi(\mu, z_1)=g(\mu) \text{ for } \mu\in[-1;0). \tag{3}$$

In equation (1), $\alpha\geq 0, \sigma>0$ and W are given functions of (μ, z), while f and g in equations (2) and (3) are given functions of μ..

From a general perspective, this equation belongs to the category of the forward-backward or two-way partial differential equations (PDEs). It is remarkable the fact that, before [11], the existing literature on the numerical resolution of this kind of problems by means of finite difference methods has given prominence to propose and analyze iterative algorithms (see [7], [8], [14], [15], [16]). For this reason, it is natural to investigate whether the iterative approach results in a faster method or not; the major contribution of this paper is to provide a body of numerical evidence that in fact it is slower in most cases. With the aim of illustrating the matter in question, let us consider the simple forward-backward heat equation

$$x\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}=f(x,t),\ (x, t)\in[-1, 1]\times[t_0, t_1]. \tag{4}$$

The PDE (4) is parabolic when $x>0$ and backward parabolic when $x<0$. Consequently, leaving apart the necessity of imposing some type of boundary conditions at $|x|=1$, it demands to split the spatial domain $[-1;1]$ into two parts, $(0;1]$ and $[-1;0)$, in order to impose an initial condition on $(0;1]$ and the final condition on $[-1;0)$ (see [16]). From a numerical “finite differences" standpoint, it is clear that this problem cannot be solved with a marching method by means of one single sweep,

as it is the case for the heat equation $\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}=f(x,t)$; indeed, for the equation (4) it is not possible to advance starting at time $t=t_0$ because the initial condition only holds on $(0;1]$, and it is not possible either to go back starting at time $t=t_1$ because the final condition only holds on $[-1;0)$. As said above, the standard treatment is of iterative type, and consists in:

1. Providing a seed, that is to say, a guess of the solution on the interface $\{0\}\times[t_0, t_1]$,

2. Solving on $(0;1]\times[t_0,\ t_1]$, advancing in time
3. Solving on $[-1;0)\times[t_0,\ t_1]$, going back in time
4. Updating the value on the interface, and
5. Iterating steps 2, 3 and 4 until convergence is achieved.

It is not being claimed that all kind of forward-backward PDEs can be solved in this way. In fact, there are more complex examples that must viewed from some other perspective (see [1]), but the problem (1)-(3) admits a similar treatment.

As an alternative to the method above, one has what it is called in this paper the direct approach, consisting in solving the problem simultaneously on the whole time-space domain, treating time t as another spatial variable, as if one were solving, in the elliptic manner, the Dirichlet problem for the Poisson equation in two dimensions.

The principle of the iterative approach is that it is much faster to march in time than to solve on the whole time-space domain by means of the direct approach. In other words, it is much faster to solve N linear square systems of order I than one single linear square system of order $I\times N$. However, since in this case marching in time implies performing iterations, the following question arises: is it possible that the number of iterations caused the iterative method to be slower than the direct method? This is a point of interest that, to our knowledge, has not been explored yet.

As said above, this paper focuses on showing that the direct approach is generally advantageous over the iterative approach when solving the problem (1)-(3) by means of finite differences. It is important to say that the scope of this assertion goes beyond the example studied, in the sense that the same statement is true for other forward-backward diffusion problems, such as the boundary value problem studied in [7], the forward-backward heat equation (4) or, more generally, as it stands in [16] when $a=a(x)$. To the authors knowledge there is no reference, apart from [11], where the direct method, as it is being understood here, be used.

The paper is organized as follows: Section 2 establishes some basic hypotheses and notations. Sections 3 and 4 include, respectively, descriptions of the direct and iterative algorithms. Section 5 collects the numerical results and their analysis. Some further explanations are given in Section 6, and finally Section 7 contains the conclusions.

2 Preliminaries

It is clear that in Problem (1)-(3) z is the time-like variable and is the space-like one. Then, Equations (2) and (3) can be interpreted, respectively, as an initial and a final condition.

In respect of hypotheses on the data functions, we shall assume that

α, σ and W are continuous on Q, $\alpha \geq 0, \sigma > 0$ (5)

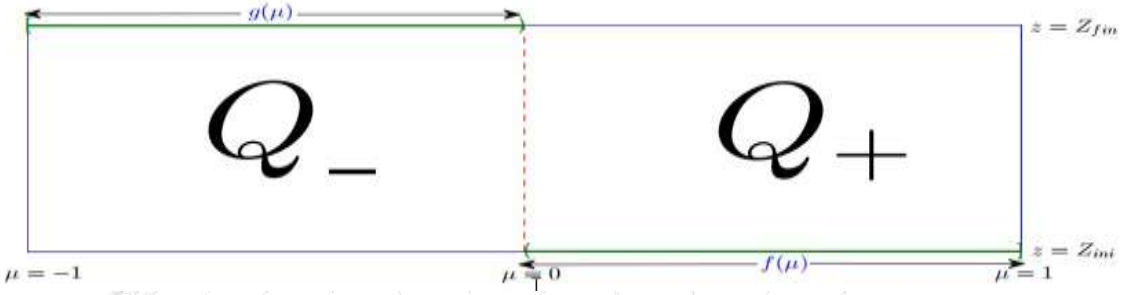
f is continuous on $(0; 1]$; g is continuous on $[-1; 0)$: (6)

Recall that $Q = [-1; 1] \times [Z_{ini}; Z_{fin}]$. For the brevity, from now on we write $a = Z_{ini}$ and $b = Z_{fin}$. Let $\{(\mu_i, z_n) : i \in \{1,...,I\}, n \in \{1,...,N\}\}$ be a mesh of Q obtained as the Cartesian product of two uniform meshes of $[-1; 1]$ [1; 1] and $[a; b]$, and think of using a finite difference scheme over this mesh. Let $h = \frac{2}{I-1}$ be the distance between μ-nodes and let $k = \frac{b-a}{N-1}$ be the distance between z-nodes.

Among all possible finite difference schemes, we choose the odd scheme described in [11], which is the best option out of two schemes considered therein. Consequently, I (the number of μ-nodes) is assumed to be odd, and $\mu_{i^*} = 0$ if $i^* = \frac{I+1}{2}$.

The following notations will also be taken: $\overline{D}_i = D(\mu_i), \overline{D}_{i\pm\frac{1}{2}} = D(\mu_i \pm \frac{h}{2})$; $\overline{\alpha}_i^n = \alpha(\mu_i, z_n), \overline{\sigma}_i^n = \sigma(\mu_i, z_n), \overline{W}_i^n = W(\mu_i, z_n)$; $\overline{f}_i = f(\mu_i), \overline{g}_i = g(\mu_i); \psi_n^i \approx \psi(\mu_i, z_n)$, understanding that ψ is the exact solution of the problem (1)-(3).

3 Description of the iterative algorithm



Consider the following the subsets of Q: $Q_- = [-1, 0) \times [a, b]$, $Q_+ = (0, 1) \times [a, b]$ and $Q_0 = \{0\} \times [a, b]$ (see Figure 1).

Figure 1: Domain $Q = Q_- \cup Q_0 \cup Q_+$, with the indication of the incoming flux boundary conditions. The vertical dotted line represents the subset Q_0.

The aim of the iterative algorithm is to compute the solution of the same linear system (7)-(12), with the caveat of taking $I \geq 7$, observing that it is possible to use a marching strategy in the variable z by working separately on each of the subdomains Q_+ and Q_-.

Clearly, one must advance when solving on Q_+ and must go back when solving on Q_- (see Figure 1).

In order to start this procedure, an initial guess of the solution on the points of the mesh belonging to Q_0 is required. The set of these values is updated after performing one forward march on Q_+ and one backward march on Q_- until convergence is achieved or a given number of iterations has been carried out.

In application to particular cases of our problem, and also to other forward-backward PDEs, proofs of convergence for iterative algorithms based on the same guidelines can be found (see [7], [8], [15] and [16]).

The detailed description reads as follows (every time we say "on Q_0" we actually mean "on the relative interior of Q_0", that is, on $Q_0 = \{0\} \times [a,b]$):

STEP 0. The seed:

Provide the algorithm with an initial guess of the values of the solution on Q_0:

$$(\psi_{i^*}^n)^{[0]} \text{ for } n \in \{2, \ldots, N-1\} \qquad (13)$$

This set of values is usually called the **seed** (initial guess) of the iterative algorithm.

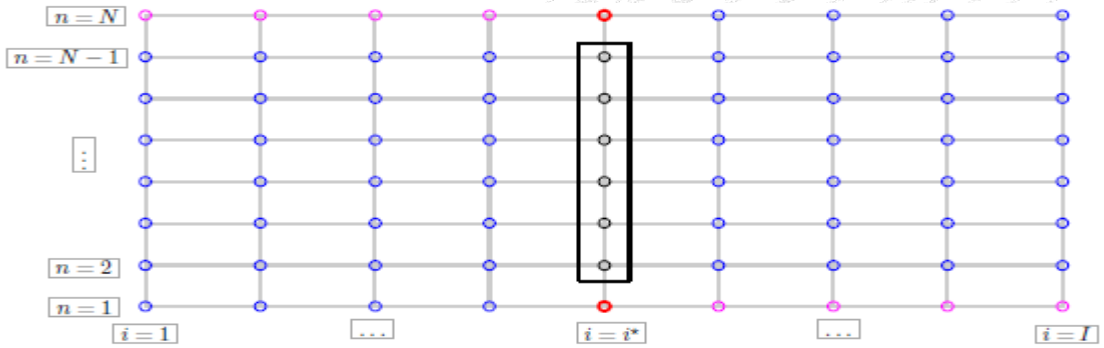


Figure 2 shows the points of the mesh where the seed is given.

A good choice of the seed, which amounts to say choosing values close to the exact solution of the discrete problem, will make the iterative process converge in fewer iterations. Maybe the simplest reasonable possibility lies in the idea of joining the pairs $(a, f(0))$ and $(b, g(0))$ by means of a straight line:

$$(\psi_{i^*}^n)^{[0]} = f(0) + [g(0) - f(0)]\frac{(n-1)}{(N-1)} \text{ for } n \in \{2, \ldots, N-1\} \qquad (14)$$

However, a better seed can be obtained as follows: firstly, solve the problem with the direct method using a coarse grid (in order to spend only a negligible amount of time), and then take the

restriction to Q_0 of this numerical solution as the basis to calculate the seed in Equation (13) by means of linear interpolation. Neither spline interpolation nor piecewise cubic Hermite interpolation, the other two possibilities offered by our Matlab® version, has outperformed the linear one in any of our numerical experiments.

STEP 1. Computation of values on Q_+ at iteration $q+1$:

For a fixed number $q \in \mathbb{N} \cup \{0\}$, assume that

$$(\psi_{i^*}^n)^{[q]} \text{ for } n \in \{2, \ldots, N-1\} \tag{15}$$

are known and obtain all values on Q_+ at iteration $q+1$, i. e.,

$$(\psi_i^n)^{[q+1]} \text{ for } (i,n) \in \{i^*+1,...,I-1\} \times \{2,...,N-1\} \tag{16}$$

by solving Equations (8) for $i \in \{i^*+1,...,I-1\}$, (10) and (11) with the help of the boundary values (15).

This is done by means of a one-step forward marching procedure starting from the known values at $z=a$ given by Equation (11).

STEP 2. Computation of values on Q_- at iteration $q+1$:

Obtain all values on Q_- at the same iteration $q+1$, i. e.,

$$(\psi_i^n)^{[q+1]} \text{ for } (i,n) \in \{1,...,i^*-1\} \times \{2,...,N-1\} \tag{17}$$

by solving Equations (7), (8 for $i \in \{2,...,i^*-1\}$), and (12) with the help of the same boundary values (15).

This is done by means of a one-step forward marching procedure starting from the known values at $z=b$ given by Equation (12).

STEP 3. Update (computation of values on Q_0 at iteration $q+1$):

Update of the values on Q_0 can be done by using Equation (9). That is to say, the updated values

$$(\psi_{i^*}^n)^{[q+1]} \text{ for } n \in \{2, \ldots, N-1\} \tag{18}$$

can follow from

$$\left(-\frac{\bar{\sigma}_{i^*}^n}{h^2}\right)(\psi_{i^*-1}^n)^{[q+1]}+\left(\bar{\alpha}_{i^*}^n+\frac{2\bar{\sigma}_{i^*}^n}{h^2}\right)(\psi_{i^*}^n)^{[q+1]}+\left(-\frac{\bar{\sigma}_{i^*}^n}{h^2}\right)(\psi_{i^*+1}^n)^{[q+1]}=\bar{W}_{i^*}^n. \qquad (19)$$

However, it is possible to accelerate convergence by over-relaxing the update, in the way we explain now: firstly, one computes

$$(\tilde{\psi}_{i^*}^n)^{[q+1]} \text{ for } n\in\{2,\ldots,N-1\} \qquad (20)$$

from

$$\left(-\frac{\bar{\sigma}_{i^*}^n}{h^2}\right)(\psi_{i^*-1}^n)^{[q+1]}+\left(\bar{\alpha}_{i^*}^n+\frac{2\bar{\sigma}_{i^*}^n}{h^2}\right)(\tilde{\psi}_{i^*}^n)^{[q+1]}+\left(-\frac{\bar{\sigma}_{i^*}^n}{h^2}\right)(\psi_{i^*+1}^n)^{[q+1]}=\bar{W}_{i^*}^n. \qquad (21)$$

and, finally, the updated values are computed from

$$(\psi_{i^*}^n)^{[q+1]}=\omega\left(\tilde{\psi}_{i^*}^n\right)^{[q+1]}+(1-\omega)(\psi_{i^*}^n)^{[q]} \qquad (22)$$

for $n\in\{2,\ldots,N-1\}$, where $\omega\in\mathbb{R}$ is a given relaxation parameter.

Notice that $\omega=1$ means that no relaxation is being deployed. The value of ω cannot be 0, otherwise no update is taking place, but being different from 0 is not guarantee of convergence. Among those values of ω offering convergence, the optimal one should be employed, but as of today the issue of finding a closed expression for this optimum (or for an estimation of it) has not been investigated; accordingly, the values of ω used in Section 5 when reporting the numerical results have been found by performing several trials, looking for values that reduce the number of iterations in a significant way with respect to the case $\omega=1$. The idea of relaxing the update in forward-backward diffusion problems has been already used in [15] and [16].

STEP 4. **Checking convergence**:

Let us define the vectors $u^{[q]},u^{[q+1]}\in\mathbb{R}^{N-2}$ as follows:

$$u_j^{[q]}=\left(\psi_{i^*}^{j+1}\right)^{[q]},\ u_j^{[q+1]}=\left(\psi_{i^*}^{j+1}\right)^{[q+1]} \qquad (23)$$

for $j\in\{1,\ldots,N-2\}$.

Given a small real number $\varepsilon>0$, we say that the algorithm has converged with ε-tolerance if

$$\frac{\left\|u^{[q+1]}-u^{[q]}\right\|}{1+\left\|u^{[q+1]}\right\|}\le\varepsilon, \qquad (24)$$

where $\|.\|$ stands for the Euclidean norm in $\mathbb{R}^{N-2}$. The quotient in Equation (24) will be referred to as the **residual**.

In case that condition (24) is not satisfied, one sets $q=q+1$ and goes back to STEP 1. If on the contrary condition (24) holds, then the process finishes and the last computed values are taken as the ultimate numerical solution:

$$\psi_i^n=(\psi_i^n)^{[q+1]} \text{ for } (i,n)\in\{1,...,I\}\times\{2,...,N\}. \qquad (25)$$

In the sequel, an *iteration* will be the process that consists of performing the steps 1, 2, 3 and 4.

4 Conclusion

In this paper it is proposed the iterative algorithm, which majority of researchers and scientists regard as reliable method to solve most of PDEs numerically. After employing codes for the solving the problem (1)-(3), one can compare direct method (see **maqola**) with the iterative method, which is based on the iterative algorithm developed in this work and conclude that which one works well for the forward-backward PDEs.

REFERENCES

1. R. Beals. On an equation of mixed type from electron scattering theory, Journal of Mathematical Analysis and Applications 58 (1977) 32–45.
2. T. A. Davis (2006) Direct Methods for Sparse Linear Systems. SIAM, Philadelphia, PA.
3. A. D. Kim and P. Tranquilli. Numerical solution of the Fokker-Planck equation with variable coefficients, Journal of Quantitative Spectroscopy & Radiative Transfer 109, no. 5 (2008) 727–740.
4. O. Lopez Pouso and N. Jumaniyazov. Numerical experiments with the Fokker-Planck equation in 1D slab geometry, Journal of Computational and Theoretical Transport 45, no. 3 (2016) 184–201.
5. V. Vanaja. Numerical solution of a simple Fokker-Planck equation, Applied Numerical Mathematics 9, no. 6 (1992) 533–540. [16] V. Vanaja and R. B. Kellogg. Iterative methods for a forward-backward heat equation, SIAM Journal on Numerical Analysis 27, no. 3 (1990) 622–635.