



Solving procedure for a 25-diagonal coefficient matrix: Direct numerical solutions of the three-dimensional linear Fokker–Planck equation

Maximiliano Ujevic *, Patricio S. Letelier

*Departamento de Matemática Aplicada, Instituto de Matemática, Estatística e Computação Científica,
Universidade Estadual de Campinas, Barão Geraldo, 13081-970, Campinas, SP, Brazil*

Received 11 May 2005; received in revised form 3 November 2005; accepted 4 November 2005
Available online 19 December 2005

Abstract

We describe an implicit procedure for solving linear equation systems resulting from the discretization of the three-dimensional (seven variables) linear Fokker–Planck equation. The discretization of the Fokker–Planck equation is performed using a 25-point molecule that leads to a coefficient matrix with equal number of diagonals. The method is an extension of Stone's implicit procedure, includes a vast class of collision terms and can be applied to stationary or non stationary problems with different discretizations in time. Test calculations and comparisons with other methods are presented in two stationary examples, including an astrophysical application for the Miyamoto–Nagai disk potential for a typical galaxy.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Numerical method; Three-dimensional Fokker–Planck equation

1. Introduction

In dealing with solutions of partial differential equations we often encounter a set of linear equations that has to be solved. This set of linear equations depends on the method used for discretization. In general, when dealing with three-dimensional systems the number of linear equations increases and the numerical solution of these equations uses most of the computing time. An extreme case is the Fokker–Planck equation. The Fokker–Planck equation is also known as the Fokker–Planck approximation because truncates the BBGKY (N.N. Bogoliubov, M. Born, H.S. Green, J.G. Kirkwood, and J. Yvon) hierarchy of kinetic equations at its lowest order by assuming that correlation between particles only plays a role as a sequence of uncorrelated two-body encounters [1,2]. Note that the only “approximation” made in the Fokker–Planck equation comes from the model adopted for collisions and, in fact, the Fokker–Planck equation can be derived from first

* Corresponding author.

E-mail addresses: mujevic@ime.unicamp.br (M. Ujevic), letelier@ime.unicamp.br (P.S. Letelier).

principles and no ad hoc suppositions are needed. The solution of the Fokker–Planck equation is not an easy task because in the three dimensional case it has seven variables: three space coordinates (\mathbf{x}), three velocity coordinates (\mathbf{v}) and time (t). In the two-dimensional case there is a simplification because the total number of variables are five. In either case, the large number of grid nodes needed for the computation of the solution becomes a storage data problem. In a three-dimensional problem, for a stationary or non-stationary equation, the number of linear equations corresponds to the number of nodes in the phase-space grid (\mathbf{x}, \mathbf{v}). If we divide each of the phase-space variables' interval of the distribution function in nine parts (10 nodes), we will have a grid with 10^6 nodes. In a simple numerical method we have to store and solve a matrix with 10^{12} elements. For the two-dimensional case, the main matrix will have 10^8 elements. With 10 grid nodes per variable only very simple geometries can be described. The large number of matrix elements brings us another computational problem, the slowness of the codes. In the discretization process of the Fokker–Planck equation, a system of linear equations is obtained and arranged into a matrix form (coefficient matrix). For the case of a finite difference scheme discretization in three dimensions with a 25-point molecule, we see that approximately less than 0.003% of the elements are different from zero. This incentives us to search for alternative and faster methods, usually iterative, to solved the linear system using only the non-null data. Note that, in general, the coefficient matrix is not symmetric. So, powerful methods like the Conjugate Gradient [3,4] and Cholesky [4] decomposition can not be used. Our main goal is to obtain a code that allows us to obtain a fast and effective numerical solutions on high resolution schemes of the three-dimensional linear Fokker–Planck equation in a direct way. The importance and difficulties of having three-dimensional solutions of the Fokker–Planck equation can be summarized in the words of Binney and Tremaine [5, p. 245], here in relation with galactic dynamics: *Finding the particular function of three variables that describes any given galaxy is no simple matter. In fact, this task has proved so daunting that only in the last few years, three-quarters of a century after Jeans's [6] paper posed the problem, has the serious quest for the distribution function of even our own Galaxy got underway.*

We mean by direct numerical calculations of the Fokker–Planck equation a method that is neither statistical nor mean field approximation [5]. Numerical solutions can be performed using statistical approximate methods like the method of moment equations [7–9] and Monte Carlo methods [10,11]. Also, solutions have been found using the orbit-averaged Fokker–Planck equation with action-angle variables [5], this last method reduces the equation involving six phase-space coordinates plus time to one involving only three actions plus time. Another method that have been used for solving the Vlasov equation with good results is the operator splitting technique [12,13]. Two-dimensional numerical integration of the Vlasov equations can be found in [14]. The problem associated with this method is that, even in the two-dimensional case, the time spent in making the interpolation is very expensive. Direct three-dimensional numerical calculations of the Fokker–Planck equations, to the best of our knowledge, has not been done. As we said before, the main difficulty to find the solution of the three-dimensional Fokker–Planck equation is the large number of linear equations that is translated in a high computational cost involve in the process. In this article we present a variations of Stone's [15] method that leads us to solve with low computational cost the three dimensional linear Fokker–Planck equation. Variations of Stone's method has been applied to other situations in two [16] and three [17] space dimensions when dealing with fluid flow, heat transfer and Laplace equation.

In this article we study how to solve the system of equations that arises from the discretization process using a finite difference scheme, in particular we used a 25-point molecule, but we have to keep in mind that other discretization procedures gives practically the same form for the coefficient matrix and this method can also be applied, i.e. sparse matrices with the same number of diagonals. For example, in two dimensions, the diffusion equation can be discretized in an uniform rectangular grid by using the finite difference scheme, the finite volume method (which is a discretization of the equation in integral form) or the finite element method, see [18]. All of these methods gives five diagonals, in the finite element method the number of diagonals depends on the type of interpolation considered, and the coefficient matrix can be solved by the same numerical method. So, in this article, we shall not discuss what discretization method is better for the problem considered, we rather present a numerical method that allows us to find the solution of a coefficient matrix with 25 diagonals that can be obtained with different discretization methods.

The article is organized as follows. In Section 2 we present the linear Fokker–Planck equation to be solved. We consider a linear collision term that includes a vast class of collisions. In Section 3, we describe the algo-

algorithm of the modified Stone method to obtain the incomplete LU decomposition. The discretization of the Fokker–Planck equation is performed using a central difference approximation for the phase-space variables (\mathbf{x}, \mathbf{v}) that is described with a 25-point molecule. This point molecule also allows to describe different discretizations in time, as the implicit Euler or Crank–Nicolson discretization. The derivation of the LU decomposition is made without assuming any particular discretization in time, maintaining its derivation as general as possible. The notation used for the diagonals in the coefficient matrix is also shown. In Section 4, we test our algorithm with an astrophysical example by solving the Fokker–Planck equation for the widely used Miyamoto–Nagai disk potential [19] in three dimensions. The parameters of the model are chosen to represent the Newtonian potential of a typical galaxy. Also, we used a Fokker–Planck test equation to compare our algorithm with the Generalized Minimal Residual Method (GMRES) that solves large sparse matrices. In Section 5, we show how to modified the code to implement curved boundary conditions. Finally, in Section 6, we summarized our results.

2. The general problem

The linear Fokker–Planck equation can be written as

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \dot{\mathbf{v}} \cdot \nabla_v f = \Gamma[f], \quad (1)$$

where \mathbf{v} represent the velocity of the particles, ∇ is the usual gradient, ∇_v is the velocity gradient (derivations are done with respect to the velocities), $\dot{\mathbf{v}}$ is the acceleration and the symbol $\Gamma[f]$ denotes the rate of change of f due to encounters (collision term). We consider as the collision term the expression

$$\Gamma[f] = A(\mathbf{x}, \mathbf{v}) \nabla^2 f + B(\mathbf{x}, \mathbf{v}) \nabla_v^2 f + C(\mathbf{x}, \mathbf{v}) \nabla f + D(\mathbf{x}, \mathbf{v}) \nabla_v f + \sum_{i \neq j=1}^3 E_{ij}(\mathbf{x}, \mathbf{v}) \frac{\partial^2 f}{\partial v_i \partial v_j}, \quad (2)$$

where A , B , C , D and E_{ij} are arbitrary functions of the phase-space variables \mathbf{x} and \mathbf{v} . The equation above describes a vast family of collisions. In particular, with the mixed velocity derivative term present in (2) we can take into account the important collision term found by Rosenbluth et al. [20] used in gravitating systems and plasma physics, see for example [21–24] and reference therein. If we need mixed space derivatives instead of mixed velocity derivatives, we can use the same code presented in this article to solve the problem. If a particular problem requires the inclusion of mixed velocity derivatives as well as mixed space derivatives, it is possible to develop a similar numerical procedure following the steps of this article, but it complicates the incomplete LU decomposition used for the method, i.e. we need 12 extra diagonals on the coefficient matrix (37 instead of 25 diagonals).

In Section 4 we solve numerically the stationary linear Fokker–Planck equation (1). In general, Eq. (1) is non-linear because $\dot{\mathbf{v}} = \mathbf{F}(f)/m$, where \mathbf{F} is the force and m is the mass of the particle. Another kind of non-linearity may arise from the collision term considered. A way to deal with this kind of problem is to start with a given distribution function at time t from which we can calculate the force and collision term at this time. Then, this force and collision term are replaced into the Fokker–Planck equation from which we obtain the distribution function for a later time $t + \Delta t$. With the recently calculated distribution function we can calculate again the force and collision term at time $t + \Delta t$ and the process is repeated. An application for a stationary non-linear problem can be found in [25], in which we found the distribution function that satisfies both Fokker–Planck and Poisson equation in two dimensions for a Kuzmin–Toomre thin disk.

3. Description of the algorithm

The system of equations obtained from the discretization of the Fokker–Planck equations (1) and (2) using the central finite difference approximation for the phase-space and a temporal discretization in time (implicit Euler, Crank–Nicolson, etc.) can be cast (for each time step) into the simple form,

$$A\Psi = Q, \quad (3)$$

in which A is a square coefficient matrix $N_{\text{node}} \times N_{\text{node}}$ (N_{node} the number of nodes in the discretization grid), Ψ is the vector matrix of the nodal variable values, and Q is the source vector. The position of the grid nodes in the phase-space (x, y, z, v_x, v_y, v_z) is performed by six indexes (i, j, k, l, m, n) , where i represents the index for the variable x , j represents the index for the variable y , etc. The ordering of nodes in this six-dimensional space is made as follows. The surface $n = \text{constant}$ are stacked one above another. Within the fifth dimensional space (for each n) the hyper-surfaces $m = \text{constant}$ are stacked one above another. Within the fourth-dimensional space (for each n and m) the hyper-surfaces $l = \text{constant}$ are stacked one above another. Within the three-dimensional space (for each n, m and l) the surfaces $k = \text{constant}$ are stacked one above another. Within the two dimensional space (for each n, m, l and k) the index j increases first (y -direction) than the index i (x -direction). The one-dimensional storage index p of the vector matrix Ψ is calculated from the six-dimensional grid indexes (i, j, k, l, m, n) , i.e.

$$p = (n - 1)n_{ijklm} + (m - 1)n_{ijkl} + (l - 1)n_{ijk} + (k - 1)n_{ij} + (i - 1)n_j + j, \tag{4}$$

with

$$\begin{aligned} i &= 1 \cdots n_i; & j &= 1 \cdots n_j; & k &= 1 \cdots n_k; & l &= 1 \cdots n_l; & m &= 1 \cdots n_m; & n &= 1 \cdots n_n; \\ n_{ijklm} &= n_i n_j n_k n_l n_m; & n_{ijkl} &= n_i n_j n_k n_l; & n_{ijk} &= n_i n_j n_k; & n_{ij} &= n_i n_j, \end{aligned} \tag{5}$$

where $n_i, n_j, n_k, n_l, n_m, n_n$ denote the number of grid points for each variable. Therefore $N_{\text{node}} = n_i n_j n_k n_l n_m n_n$. With the help of the storage index p we can switch each point of the 25-point molecule from the matrix form to the one dimensional position representation. This is done in Table 1 by making the equivalence $f(x, y, z, v_x, v_y, v_z) \equiv \Psi(i, j, k, l, m, n)$. Until now, we considered only the non-stationary case, but the discretization in Table 1 can be used in the non-stationary as well as the stationary Fokker–Planck equation. In both cases, the final system of equations is of the form (3), with A being a sparse matrix with elements different from zero

Table 1

Nomenclature and relations between the matrix form and the one-dimensional storage index at node p of the 25 terms used in the discretization of the Fokker–Planck equation

Matrix form	Abbreviation	Name	Position from node p
<i>Basic diagonals</i>			
$\Psi(i, j, k, l, m, n)$	P	Point	p
$\Psi(i, j + 1, k, l, m, n)$	N	North	$p + 1$
$\Psi(i, j - 1, k, l, m, n)$	S	South	$p - 1$
$\Psi(i + 1, j, k, l, m, n)$	E	East	$p + n_j$
$\Psi(i - 1, j, k, l, m, n)$	W	West	$p - n_j$
$\Psi(i, j, k + 1, l, m, n)$	T	Top	$p + n_{ij}$
$\Psi(i, j, k - 1, l, m, n)$	B	Bottom	$p - n_{ij}$
$\Psi(i, j, k, l + 1, m, n)$	$U1$	Up1	$p + n_{ijk}$
$\Psi(i, j, k, l - 1, m, n)$	$D1$	Down1	$p - n_{ijk}$
$\Psi(i, j, k, l, m + 1, n)$	$U2$	Up2	$p + n_{ijkl}$
$\Psi(i, j, k, l, m - 1, n)$	$D2$	Down2	$p - n_{ijkl}$
$\Psi(i, j, k, l, m, n + 1)$	$U3$	Up3	$p + n_{ijklm}$
$\Psi(i, j, k, l, m, n - 1)$	$D3$	Down3	$p - n_{ijklm}$
<i>Mixed diagonals</i>			
$\Psi(i, j, k, l + 1, m + 1, n)$	$UT2$	Uptop2	$p + (n_{ijkl} + n_{ijk})$
$\Psi(i, j, k, l - 1, m + 1, n)$	$UB2$	Upbottom2	$p + (n_{ijkl} - n_{ijk})$
$\Psi(i, j, k, l + 1, m - 1, n)$	$DT2$	Downtop2	$p - (n_{ijkl} - n_{ijk})$
$\Psi(i, j, k, l - 1, m - 1, n)$	$DB2$	Downbottom2	$p - (n_{ijkl} + n_{ijk})$
$\Psi(i, j, k, l + 1, m, n + 1)$	$UT3$	Uptop3	$p + (n_{ijklm} + n_{ijk})$
$\Psi(i, j, k, l - 1, m, n + 1)$	$UB3$	Upbottom3	$p + (n_{ijklm} - n_{ijk})$
$\Psi(i, j, k, l + 1, m, n - 1)$	$DT3$	Downtop3	$p - (n_{ijklm} - n_{ijk})$
$\Psi(i, j, k, l - 1, m, n - 1)$	$DB3$	Downbottom3	$p - (n_{ijklm} + n_{ijk})$
$\Psi(i, j, k, l, m + 1, n + 1)$	$UU3$	Upup3	$p + (n_{ijklm} + n_{ijkl})$
$\Psi(i, j, k, l, m - 1, n + 1)$	$UD3$	Updown3	$p + (n_{ijklm} - n_{ijkl})$
$\Psi(i, j, k, l, m + 1, n - 1)$	$DU3$	Downup3	$p - (n_{ijklm} - n_{ijkl})$
$\Psi(i, j, k, l, m - 1, n - 1)$	$DD3$	Downdown3	$p - (n_{ijklm} + n_{ijkl})$

in only 25 diagonals, see Fig. 1. Now, we want to develop an iteration method in order to solve the system of Eq. (3). After h iterations of such method, the approximate solution Ψ^h do not satisfies (3) exactly, their is a non-zero residual R such as

$$A\Psi^h = Q - R^h. \tag{6}$$

The purpose of the iteration procedure is to drive the residual term to zero after some number of iteration (actually we stop the iteration when the residual term attained some imposed small value condition). Let us consider an iterative scheme for a linear system in the form

$$M\Psi^{h+1} = O\Psi^h + B, \tag{7}$$

when convergence is achieved we must have that $A = M - O$ and $B = Q$. An alternative version of this procedure can be obtained by subtracting $M\Psi^h$ from both sides of (7) to have

$$M\Delta^{h+1} = R^h, \tag{8}$$

where $\Delta^{h+1} = \Psi^{h+1} - \Psi^h$ and $R^h = B - (M - O)\Psi^h = Q - A\Psi^h$. Any effective iterative method to solve (7) must be cheap and converge rapidly. For faster convergence, the matrix M have to be a good approximation of the coefficient matrix A , i.e. we must have $O\Psi^h$ small. The original idea of Stone is to use for the iteration matrix M an incomplete LU decomposition of the matrix A . The reason for this choice is that LU decomposition is an excellent linear system solver. The matrices L and U have elements different from zero only in the diagonals in which A have also elements different from zero. The product of the matrices, L and U , provide a

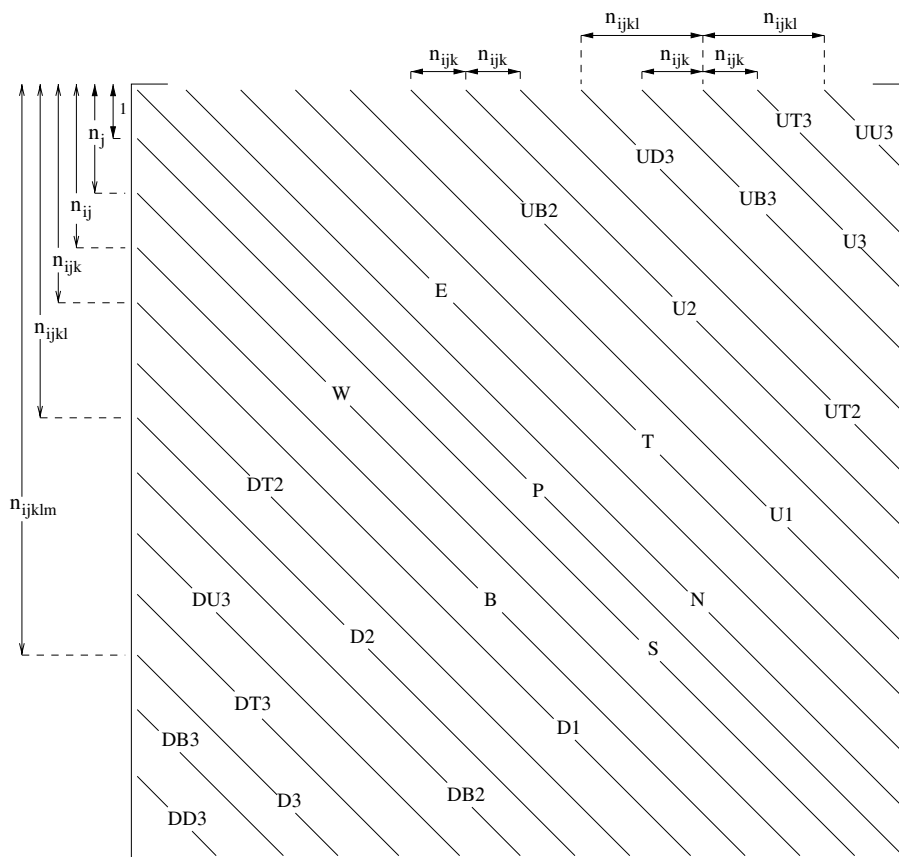


Fig. 1. Form of the coefficient matrix A obtained from the discretization of the partial difference equation of our problem. The separation between the main diagonal P and the other diagonals are indicated. Also, the distance between the diagonals of the mixed derivatives to their nearest diagonals are shown. Note that the figure is not in scaled.

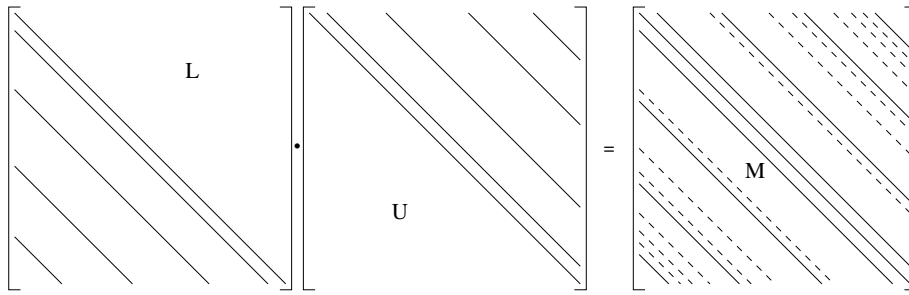


Fig. 2. Schematic representation of the matrices L , U and their product M . The multiplication of the matrices L and U leads to extra diagonals (dotted lines) not present in the coefficient matrix A . The diagonals of L , from the left-bottom corner to the main diagonal, are: $DD3, DB3, D3, DT3, DU3, DB2, D2, DT2, D1, W, S, P$. The diagonals of U , from the main diagonal to the right-up diagonal, are: $1, N, E, T, U1, UB2, U2, UT2, UD3, UB3, U3, UT3, UU3$. The diagonals in M are products of these two sets of diagonals but we have to be careful because more than one product can be at the same diagonal.

matrix M with a larger number of diagonals with elements different from zero, see for instance Fig. 2. To make the decomposition LU unique, we set the elements on the principal diagonal of U equal to 1. In doing the LU multiplication, we have to pay extra attention because sometimes more than one diagonal product are at the same distance from the main diagonal P , e.g. the product $L_{D3} \cdot U_1$ and $L_{DD3} \cdot U_{U2}$ are at the same diagonal in M . The multiplication rules for matrices furnish the elements of $M = LU$ at node p (see Appendix A), where for example, $M_{W|UU3}$ represents the diagonal in M that is obtain from the multiplication between the elements of the diagonal S in L with the elements of the diagonal $UU3$ in U .

Now, we choose L and U in such a way that $M (= A + O)$ is the best possible approximation to A . The standard method for decomposition is to let O to have elements different from zero in the diagonals of M that corresponds to the diagonals not present in A and to force the other diagonals of M to be equal to the corresponding diagonals in A . But this method converges slowly because the resulting matrix O is not small. Stone recognized that the convergence of the method could be faster if we allow O to have elements different from zero also in the diagonals present in A . The key idea is that the contribution of $M\Psi$ of the diagonals not present in A partially canceled the contribution of $O\Psi$ of the diagonals present in A , in such a way that

$$O\Psi \approx 0. \tag{9}$$

Note that in (A1) they are diagonals in M that present more than one term. In general, the principal diagonals have more than one element, as for example M_{DT3} , but beside these principal diagonals there are other no-principal diagonals that have more than one element, like $M_{D3|UB2}$. Now, relation (9) can be written for one grid node in several ways. The usual way is to consider the elements of these no-principle diagonals as part of the same diagonal. Other way is to consider these elements as they were from different diagonals, thus in this case, following the above example, the no-principal diagonal $M_{D3|UB2}$ is split into two diagonals $M_{D3|UB2}$ and $M_{DB3|U2}$. We obtained the final relations for the LU decomposition in both ways and we found that the LU decomposition considering the elements as they were from different diagonals is faster by a factor of 2. Hereafter we consider this case. The explicit form of Eq. (9) is given in Appendix B.

The problem now is to defined the elements of O to satisfy Eq. (B1) without introducing additional unknowns. If we expect the solution of the partial differential equation to be smooth, we can approximate the values of $\Psi_{B|N}$, $\Psi_{N|W}$, etc, in terms of the values of Ψ at nodes corresponding to the diagonals of A . Stone proposed the following approximations (other approximations are possible),

$$\begin{aligned} \Psi_{B|N} &\approx \alpha(\Psi_B + \Psi_N - \Psi_P), \\ \Psi_{W|N} &\approx \alpha(\Psi_W + \Psi_N - \Psi_P), \quad \text{etc.,} \end{aligned} \tag{10}$$

where α is a constant. Stability analysis made by Stone requires that α must be between $0 < \alpha < 1$. Replacing the above approximations into (B1) we obtain the elements of O as a linear combination of the elements of M , see for instance Appendix C. Now, using the relation $M = A + O$ together with expressions (A1) and (C1), we find that the elements of the matrices L and U are given by

$$\begin{aligned}
 L_{[X]}^p &= \frac{A_{[X]}^p - C_{[X]}}{1 + \alpha K_{[X]}}, \quad [X] = DD3, DB3, D3, DT3, DU3, DB2, D2, DT2, D1, B, W, S, \\
 L_p^p &= A_p^p - L_S^p U_N^{p-1} - L_W^p U_E^{p-n_j} - L_B^p U_T^{p-n_{ij}} - L_{D1}^p U_{U1}^{p-n_{ijk}} - L_{DT2}^p U_{UB2}^{p-(n_{ijkl}-n_{ijk})} - L_{D2}^p U_{U2}^{p-n_{ijkl}} \\
 &\quad - L_{DB2}^p U_{UT2}^{p-(n_{ijkl}+n_{ijk})} - L_{DU3}^p U_{UD3}^{p-(n_{ijklm}-n_{ijkl})} - L_{DT3}^p U_{UB3}^{p-(n_{ijklm}-n_{ijk})} - L_{D3}^p U_{U3}^{p-n_{ijklm}} \\
 &\quad - L_{DB3}^p U_{UT3}^{p-(n_{ijklm}+n_{ijk})} - L_{DD3}^p U_{UU3}^{p-(n_{ijklm}+n_{ijkl})} + \alpha(K_N + K_E + K_T + K_{U1} + K_{UB2} + K_{U2} + K_{UT2} \\
 &\quad + K_{UD3} + K_{UB3} + K_{U3} + K_{UT3} + K_{UU3}), \\
 U_{[X]}^p &= \frac{A_{[X]}^p - \alpha K_{[X]} - C_{[X]}}{L_p^p}, \quad [X] = N, E, T, U1, UB2, U2, UT2, UD3, UB3, U3, UT3, UU3,
 \end{aligned} \tag{11}$$

where the explicit form of the functions $C_{[X]}$ and $K_{[X]}$ are given in Appendix D. The elements of the LU decomposition have to be calculated in the order specified in (11). In doing this, we must take into account that a certain element is considered equal to zero if its storage index is less or equal zero, e.g. if $p = 3$ and $n_j = 5$ then the elements with index p and $p - 1$ are different from zero, and the elements with index $p - n_j$, $p - n_{ij}$, etc are equal to zero. When mixed derivatives are not present we must set all the elements with index $DD3, DB3, DT3, DU3, DB2, DT2, UB2, UT2, UD3, UB3, UT3, UU3$ equal to zero. Once obtained the LU decomposition, the system of equation is solved combining $M = LU$ with (8) to obtain

$$LU\Delta^{h+1} = R^h, \tag{12}$$

and here we set

$$\begin{aligned}
 L\mathcal{T}^h &= R^h, \\
 U\Delta^{h+1} &= \mathcal{T}^h,
 \end{aligned} \tag{13}$$

from which we obtain the solution of our problem by solving two triangular systems. In this iterative method, the matrix elements of L and U are calculated only once before the first iteration. In other iterations, only the residual R , and Δ are calculated using the two triangular system mentioned above, i.e.,

$$\begin{aligned}
 \mathcal{T}^p &= (R^p - L_{DD3}^p \mathcal{T}^{p-(n_{ijklm}+n_{ijkl})} - L_{DB3}^p \mathcal{T}^{p-(n_{ijklm}+n_{ijk})} - L_{D3}^p \mathcal{T}^{p-n_{ijklm}} - L_{DT3}^p \mathcal{T}^{p-(n_{ijklm}-n_{ijk})} \\
 &\quad - L_{DU3}^p \mathcal{T}^{p-(n_{ijklm}-n_{ijkl})} - L_{DB2}^p \mathcal{T}^{p-(n_{ijkl}+n_{ijk})} - L_{D2}^p \mathcal{T}^{p-n_{ijkl}} - L_{DT2}^p \mathcal{T}^{p-(n_{ijkl}-n_{ijk})} - L_{D1}^p \mathcal{T}^{p-n_{ijk}} \\
 &\quad - L_B^p \mathcal{T}^{p-n_{ij}} - L_W^p \mathcal{T}^{p-n_j} - L_S^p \mathcal{T}^{p-1}) / L_p^p, \\
 \Delta^p &= \mathcal{T}^p - U_{UU3}^p \Delta^{p+(n_{ijklm}+n_{ijkl})} - U_{UT3}^p \Delta^{p+(n_{ijklm}+n_{ijk})} - U_{U3}^p \Delta^{p+n_{ijklm}} - U_{UB3}^p \Delta^{p+(n_{ijklm}-n_{ijk})} \\
 &\quad - U_{UD3}^p \Delta^{p+(n_{ijklm}-n_{ijkl})} - U_{UT2}^p \Delta^{p+(n_{ijkl}+n_{ijk})} - U_{U2}^p \Delta^{p+n_{ijkl}} - U_{UB2}^p \Delta^{p+(n_{ijkl}-n_{ijk})} - U_{U1}^p \Delta^{p+n_{ijk}} \\
 &\quad - U_T^p \Delta^{p+n_{ij}} - U_E^p \Delta^{p+n_j} - U_N^p \Delta^{p+1},
 \end{aligned} \tag{14}$$

where

$$\begin{aligned}
 R^p &= Q^p - A_p^p \Psi^p - A_N^p \Psi^{p+1} - A_S^p \Psi^{p-1} - A_E^p \Psi^{p+n_j} - A_W^p \Psi^{p-n_j} - A_T^p \Psi^{p+n_{ij}} - A_B^p \Psi^{p-n_{ij}} - A_{U1}^p \Psi^{p+n_{ijk}} \\
 &\quad - A_{D1}^p \Psi^{p-n_{ijk}} - A_{UB2}^p \Psi^{p+(n_{ijkl}-n_{ijk})} - A_{DT2}^p \Psi^{p-(n_{ijkl}-n_{ijk})} - A_{U2}^p \Psi^{p+n_{ijkl}} - A_{D2}^p \Psi^{p-n_{ijkl}} - A_{UT2}^p \Psi^{p+(n_{ijkl}+n_{ijk})} \\
 &\quad - A_{DB2}^p \Psi^{p-(n_{ijkl}+n_{ijk})} - A_{UD3}^p \Psi^{p+(n_{ijklm}-n_{ijkl})} - A_{DU3}^p \Psi^{p-(n_{ijklm}-n_{ijkl})} - A_{UB3}^p \Psi^{p+(n_{ijklm}-n_{ijk})} - A_{DT3}^p \Psi^{p-(n_{ijklm}-n_{ijk})} \\
 &\quad - A_{U3}^p \Psi^{p+n_{ijklm}} - A_{D3}^p \Psi^{p-n_{ijklm}} - A_{UT3}^p \Psi^{p+(n_{ijklm}+n_{ijk})} - A_{DB3}^p \Psi^{p-(n_{ijklm}+n_{ijk})} - A_{UU3}^p \Psi^{p+(n_{ijklm}+n_{ijkl})} \\
 &\quad - A_{DD3}^p \Psi^{p-(n_{ijklm}+n_{ijkl})},
 \end{aligned}$$

in which, for simplicity, we have omitted the iterative index h .

4. Test results and comparisons with other methods

As was mentioned in the Introduction, the huge number of nodes needed to solve the Fokker–Planck equation leads to large amount of data that has to be stored in a matrix, this fact reduces the possible codes for

testing the results. Furthermore, in general, the Fokker–Planck equation (1) is not symmetric, and for that reason other methods can not be used. In this case, the Generalized Minimal Residual method (GMRES) [26] is the most appropriate choice. The GMRES method belongs to the class of Krylov based iterative methods [27–29] and was proposed in order to solve large, sparse and non-Hermitian linear systems. This section is divided into two parts. In the first part, we apply the method presented in this article to a physical example. In particular, we choose the astrophysical problem of finding the distribution function of the Miyamoto–Nagai disk in three dimensions. In the second part, we used a simplified version of the physical example to make numerical comparisons. We compare our method with the best available method in solving huge sparse matrices, i.e. GMRES. We prefer to make the numerical comparisons with the simplified version of the physical example because the large amount of parameters present in the physical example make the analysis cumbersome. This analysis is important because we show some advantages and limitations of our method.

4.1. Physical example

To test of our code we begin with an important astrophysical application, i.e. solve the Fokker–Planck equation to find the distribution function of the Miyamoto–Nagai disk [19] in three dimensions in a stationary regime. The Fokker–Planck equation to be solved is [5]

$$\mathbf{v} \cdot \nabla f + \dot{\mathbf{v}} \cdot \nabla_v f = - \sum_{i=1}^3 \frac{\partial}{\partial v_i} [f(\mathbf{x}, \mathbf{v}) D(\Delta v_i)] + \frac{1}{2} \sum_{i,j=1}^3 \frac{\partial^2}{\partial v_i \partial v_j} [f(\mathbf{x}, \mathbf{y}) D(\Delta v_i \Delta v_j)], \quad (15)$$

where $\dot{\mathbf{v}} = -\nabla \Phi$, $\Phi = -\frac{GM}{\sqrt{x^2+y^2+(a+\sqrt{z^2+b^2})^2}}$, G is the gravitational constant, M is the total mass of the system, a and b are parameters that depending on the choice the potential can represent anything from an infinitesimal thin disk to a spherical system, and the functions $D(\Delta v_i)$ and $D(\Delta v_i \Delta v_j)$ are known as the diffusion coefficients. These diffusion coefficients were calculated by Rosenbluth et al. [20] considering a test star of mass m moving through an infinite homogeneous sea of field stars of mass m_a who has mean velocity equal to zero. Moreover, the interaction between the particles are ruled by an inverse square force, and also, each stellar encounter involve only a single pair of stars and are independent of all others. These diffusion coefficients are simplified if the field stars distribution function is a Maxwellian distribution. The explicit form of these coefficients are [5]

$$D(\Delta v_i) = \frac{v_i}{v} D(\Delta v_{\parallel}), \quad (16)$$

$$D(\Delta v_i \Delta v_j) = \frac{v_i v_j}{v^2} \left[D(\Delta v_{\parallel}^2) - \frac{1}{2} D(\Delta v_{\perp}^2) \right] + \frac{1}{2} \delta_{ij} D(\Delta v_{\perp}^2), \quad (17)$$

where $D(\Delta v_{\parallel})$, $D(\Delta v_{\parallel}^2)$ and $D(\Delta v_{\perp}^2)$ are given by

$$D(\Delta v_{\parallel}) = -\frac{4\pi G^2 \rho (m + m_a) \ln A G(X)}{\sigma^2}, \quad (18)$$

$$D(\Delta v_{\parallel}^2) = \frac{4\sqrt{2}\pi G^2 \rho m_a \ln A}{\sigma} \frac{G(X)}{X}, \quad (19)$$

$$D(\Delta v_{\perp}^2) = \frac{4\sqrt{2}\pi G^2 \rho m_a \ln A}{\sigma} \left[\frac{\operatorname{erf}(X) - G(X)}{X} \right], \quad (20)$$

where ρ and σ are the density and velocity dispersion of the field stars respectively, $X \equiv v/(\sqrt{2}\sigma)$, $\operatorname{erf}(X)$ is the error function, $G(X) = \frac{1}{2X^2} \left[\operatorname{erf}(X) - \frac{2X}{\sqrt{\pi}} e^{-X^2} \right]$, $A = \frac{b_{\max} v_{\text{typ}}^2}{G(m+m_a)}$, b_{\max} is the maximum possible impact parameter (usually set of order the radius of the system), and v_{typ} is a typical velocity of stars in the system. We shall find the solution of Eq. (15) in a six-dimensional ‘box’ using 10 grid nodes for variable that leads to a problem involving 10^6 unknowns. If we want to solve this set of equations using a conventional solver we need to storage 10^{12} elements in the coefficient matrix. We considered that the ranges of the velocities v_x , v_y , and v_z are between $[-350 \text{ km s}^{-1}, 350 \text{ km s}^{-1}]$, the ranges of the coordinates x and y are between $[-40 \text{ kpc}, 40 \text{ kpc}]$, and the coordinate z between $[-1 \text{ kpc}, 1 \text{ kpc}]$. At the borders of the ‘box’ we used a Dirichlet boundary con-

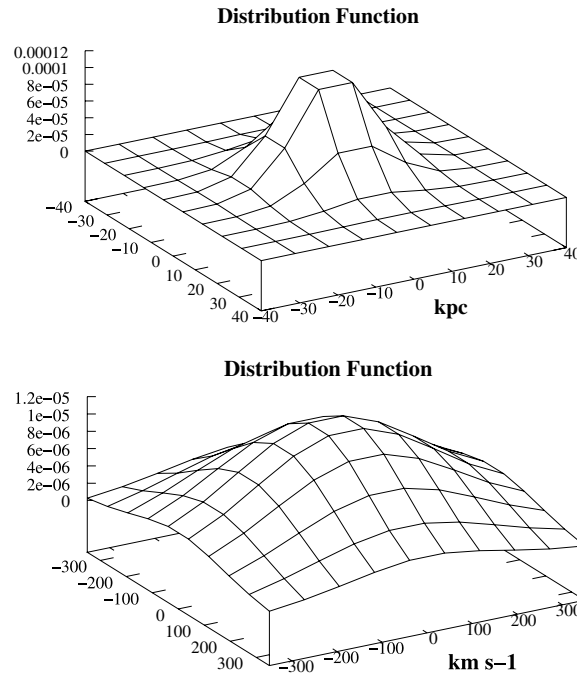


Fig. 3. Numerical solutions of the distribution function of Eq. (15). Top: surface distribution for the case with $v_x \approx -117 \text{ km s}^{-1}$, $v_y \approx 39 \text{ km s}^{-1}$, $v_z \approx -272 \text{ km s}^{-1}$ and $z \approx 0.55 \text{ kpc}$. Bottom: velocity distribution for the case with $x \approx -22.23 \text{ kpc}$, $y \approx 13.32 \text{ kpc}$, $z \approx -0.11 \text{ kpc}$ and $v_x \approx 117 \text{ km s}^{-1}$.

dition of the form $f = \exp[-(x^2 + y^2 + z^2)/2\sigma_s^2] \exp[-(v_x^2 + v_y^2 + v_z^2)/2\sigma_v^2]$. We set the parameters $a = 4 \text{ kpc}$, $b = 1 \text{ kpc}$, $m_a = m = M_\odot$, $\sigma_s = 10 \text{ kpc}$, $\sigma_v = 100 \text{ km s}^{-1}$, $v_{\text{typ}} = 200 \text{ km s}^{-1}$, $b_{\text{max}} = 40 \text{ kpc}$ and the total mass $M = 10^{12} M_\odot$. The values set for the parameters correspond to typical galaxy data like the Milky Way. Also, as a first approximation we spread the total mass uniformly along the disk and set the density $\rho = \text{constant}$. In Fig. 3, we present two graphs of the distribution function that represent the numerical solution of (15). One is the surface distribution function on the plane $x - y$ and the other is the velocity distribution function on the plane $v_y - v_z$; they are plotted at different points of the grid. We note that these figures have Maxwellian form as it should.

4.2. Numerical comparison

In the next example we shall compare our code based in the LU decomposition (11) with the GMRES algorithm. The problem of this method is that the storage of the orthonormal basis may become prohibitive for some matrices, this storage depends on the value of the restarting parameter. The restarting parameter of the GMRES algorithm determine the number of the orthonormal vectors for the Krylov subspace that the code stores in order to calculate the updated solution and residual at each time step. At each time step the code stores one vector. After a number of steps equal to the restarting parameter, the code construct the most recent update and use it as a first guess to restart the next set of iterations. The convergence of the method is guaranteed for large numbers of the restarting parameter, but this means that more vectors have to be stored and computer memory problems may appear. We can use lower values of the restarting parameter but this increases the time spent in finding the solution. In particular, we used the *gmres* routine of Matlab because it can handle sparse matrices. We also used a public GMRES software [30], this software allows us to choose between different kinds of preconditioners and orthogonalization procedures but its drawback is that can not handle sparse matrices. We choose as a test equation a stationary Fokker–Planck equation similar to the physical example of the previous section. Note that in the non-stationary Fokker–Planck equation we can perform a Crank–Nicolson discretization in time, which is also an implicit proce-

ture, and the method presented in this article can be applied. The stationary Fokker–Planck equation considered for the test is

$$\mathbf{v} \cdot \nabla f + \dot{\mathbf{v}} \cdot \nabla_v f = -\nabla_v f + \frac{1}{2} \sum_{i \neq j=1}^3 \frac{\partial^2 f}{\partial v_i \partial v_j} + \beta \sum_{i=1}^3 \frac{\partial^2 f}{\partial v_i^2}, \quad (21)$$

where $\dot{\mathbf{v}} = -\nabla \Phi$, $\Phi = 1/\sqrt{x^2 + y^2 + z^2 + 1^2}$, and β is a constant. The form of Eq. (21) is similar to the equation found when Rosenbluth potentials [20] are used in a gravitational potential Φ . Note that the collision term has mixed velocity derivatives and that the resulting coefficient matrix from the discretization is non-symmetric. We used central difference and a 25-point molecule to perform the discretization of Eq. (21), see also Table 1. Here, we find the solution of the above equation in a six-dimensional ‘box’ of length 1.22 units. At the borders we used a Dirichlet boundary condition of the form $f = \exp(-x^2 - y^2 - z^2) \exp(-v_x^2 - v_y^2 - v_z^2)$.

We first started with coarse grid of four points per variable that leads to a matrix of 4096^2 elements (in this number we are not considering the border grid points given by the boundary condition) and $\beta = 1$. We found for this case that our method spent approximately 0.2 seconds to find the solution (all the calculations were performed with a Pentium IV of 1.8 GHz, the Fortran code was compiled with the Linux free compiler). We stop the iteration when $\sum_{i=1}^n (\Delta_i^{t+1}) < 10^{-10}$. The *gmres* routine and the public GMRES code [30] spent approximately 0.35 and 3 seconds respectively to solve the system of equations with the same stop criteria, but it could be more if we choose wrong the restarting parameter. Here we are considering only the time spent to solve the coefficient matrix and not the time due to create the coefficient matrix and upload it into the code. In our code, we only upload 25 vectors of approximately N_{node} .

For a grid of five points per variable some code compilers can not allow the storage of the coefficient matrix because is too large 5^{12} , approximately 244 millions of elements. This was the case of the public code because it can not manage sparse matrices. For this number of grid points the modified Stone method was almost 7 times faster than the *gmres* routine. In Fig. 4 we present the efficiency of our method compared to the *gmres* routine. Note that when we increase the number of grid points the efficiency also increases. For a grid of nine points per variable the *gmres* routine can not manage to find the solution in one iteration because we have computer storage problems. To handle this, we lower the restarting parameter to the maximum possible value that avoids this problem. For a grid of 10 points per variable the number of elements in the coefficient matrix increases to 10^{12} . Our code can manage this huge amount of data in a faster and efficient way. The time spent for this case was approximately 85 seconds. This should be the time spent for each time step in a non-stationary problem, which is a great result considering the number of nodes in the grid and the precision attained with the stop criteria. The variation of α between the accepted limits $0 < \alpha < 1$ slightly alter the time spent in finding the solution. In Fig. 5, we present two graphs of the distribution function from the numerical solu-

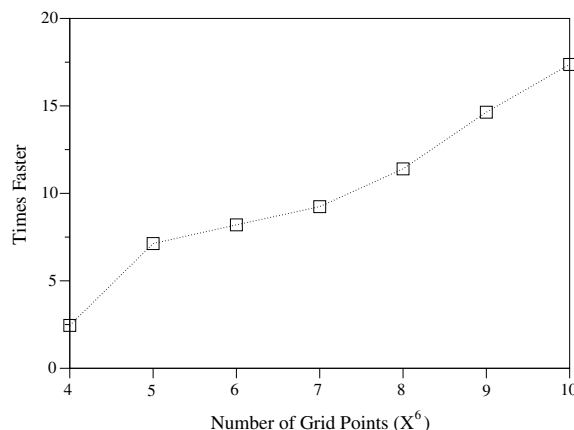


Fig. 4. Efficiency of our modified Stone method when compared to the *gmres* routine of Matlab that handles sparse matrices. When we increase the number of grid points our method becomes more efficient. To handle the 9^6 and 10^6 grid in our computer, we lower the value of the restarting parameter of the GMRES routine to the maximum possible value that avoids computer memory problems.

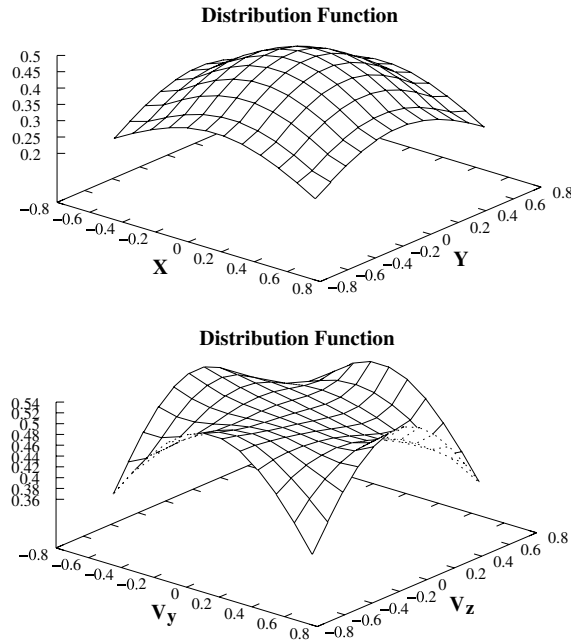


Fig. 5. Numerical solutions of the distribution function of Eq. (21). Top: surface distribution for the case with $v_x \approx 0.38$, $v_y \approx -0.5$, $v_z \approx -0.5$ and $z \approx 0.16$. Bottom: velocity distribution for the case with $x \approx -0.39$, $y \approx 0.06$, $z \approx -0.17$ and $v_x \approx 0.28$.

tion of Eq. (21) for a grid of 10 grid per variable. One is the surface distribution function on the plane $x - y$ and the other is the velocity distribution function on the plane $v_y - v_z$; they are plotted at different points of the grid. Note that these figures have not Maxwellian form because Eq. (21) is not in the collisional regime for the parameters considered in the collision term. This example was chosen only for didactic reasons.

Note that when we used central difference for discretization of Eq. (21), the only contribution to the main diagonal comes from the last right hand term, i.e. the term with β . For some values of $\beta < 1$ our code diverge because our resulting coefficient matrix is not diagonal dominant. For the same value of β , the *gmres* routine converges. The break in convergence at these values of β coincide with the appearance of negatives values on the distribution function solution found by *gmres*. We know that one of condition to attained a physical solution of the Fokker–Planck equation is that the distribution function has to be always positive. It is remarkable that for Eq. (21) our code diverge when physical solutions are not possible, this could be an indication that we are applying a wrong scheme for discretization or that we are not describing well the physical phenomena considered. As we said in the Introduction, other discretization scheme may be applied to avoid this problem. Thus, in our method the convergence is conditioned to the form of the functions present in (1) and (2), i.e. physical considerations; and to the difference scheme applied for discretization, i.e. numerical implementation. For a non-stationary scheme the Crank–Nicolson decomposition in time is suggested because it has implicit character and it is usually more stable than other methods, but strictly speaking, the stability of the system has to be studied for each particular case considered.

5. Domains with curved boundaries

The *LU* code was tested in the previous section with a structure-orthogonal grid but this does not mean that it can not be applied to more general geometries. To handle a non-square domain we proceed as follows. First we make a square structured grid with $N_{\text{node}} = n_x n_y n_z n_t$ nodes, then we label them according to the index p defined in (4). Later, the nodes that laid outside the boundary regions are not considered for the calculations. Now, lets see this procedure in an example. For didactic purpose we used only the one-dimensional case of the Fokker–Planck equation in which we have two variables (x, v_x). In Fig. 6 is depicted in the two dimensional

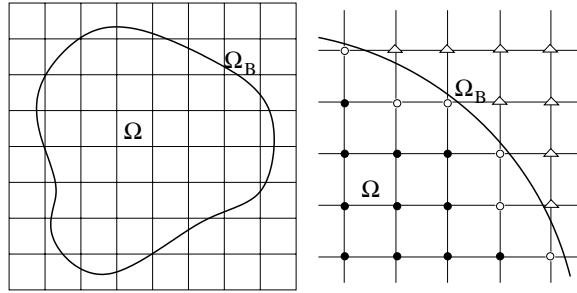


Fig. 6. Application of the incomplete LU decomposition to a general domain Ω with curved boundaries Ω_B . The square structured grid leads three classes of grid points: interior points (full circles), boundary points (empty circles) and exterior points (triangles).

plane (x, v_x) a domain region Ω with boundary Ω_B . The domain Ω is filled with a square structured grid that leads three classes of grid points: the interior points in which the normal discretization procedure can be done; the boundary points in which special care have to be taken when Dirichlet, Neumann or mixed boundary conditions are applied; and the exterior points that have to be neglected for the calculation, see Fig. 6.

The incorporation of the Dirichlet boundary conditions using central differences for the boundary points of region Ω can be done using a Taylor expansion around the nearest nodes. For example, in Fig. 7, we take the nearest boundary points for point 1: the internal node (node 2) and the point at the boundary Ω_B (point E); and make two Taylor expansions around point 1. These expansions give us (our conventions are: partial derivative with respect to the coordinate x denoted by $(,x)$; Δx is the discretization interval in the x direction),

$$\begin{aligned} \Psi_E &= \Psi_1 + b\Delta x\Psi_{,x} + \frac{(b\Delta x)^2}{2!}\Psi_{,xx} + O[\Delta x^3], \\ \Psi_2 &= \Psi_1 - \Delta x\Psi_{,x} + \frac{(\Delta x)^2}{2!}\Psi_{,xx} + O[\Delta x^3], \end{aligned} \tag{22}$$

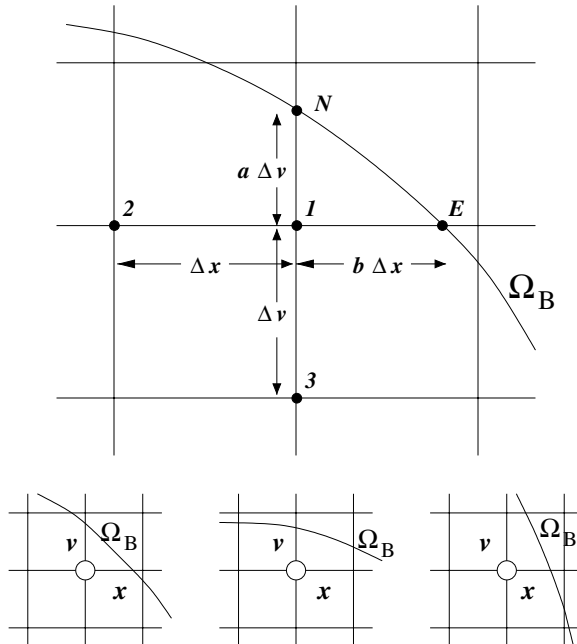


Fig. 7. Schematic representation of a five-point molecule for the finite difference approximation of the derivatives in the plane (x, v_x) for a boundary point. Below, we see different boundary situations that can occur in the numerical computation depending on the discretization grid and boundary S .

eliminating the second order derivative between these equations we can express the partial derivative at a boundary point as

$$\Psi_x = \frac{1}{\Delta x} \left[\frac{1}{b(1+b)} \Psi_E - \frac{b}{1+b} \Psi_2 - \frac{1-b}{b} \Psi_1 \right] + O[\Delta x^2]. \quad (23)$$

When $b = 1$ we recover the usual expression for the central derivative. The same procedure can be applied to the northern boundary, between node 3 and point A , and to the different types of boundary nodes in Fig. 7. Also, partial derivative of higher orders and mixed derivatives can be found in a similar way. Furthermore, it is also possible to implement Neumann boundary conditions in an irregular boundary using finite difference, see for instance [31].

The exterior points of Fig. 6 are strictly necessary to maintain the ordering of the nodes in a domain with curved boundaries. This ordering is needed by the code to operate normally but they do not enter into the calculations. To implement this condition, we have to set the values of the elements $L_{[x]}^p$, L_p^p , $U_{[x]}^p$, R^p , Δ^p and Δ^p of the external grid point (each exterior point has a position p from the one-dimensional storage index) equal to zero.

An application of the LU method in a two-dimensional Fokker–Planck equation (x, y, v_x, v_y) with curved boundaries, as well as the incomplete LU decomposition for this case can be seen in [25]. Here, the distribution function of a stationary gravitational thin disk is calculated. Note that in order to obtain the elements of the LU decomposition for the two-dimensional case, we have to start the calculations from the beginning, i.e. we can not use the elements found in the three-dimensional case.

6. Concluding remarks

We have developed a variation of the incomplete LU decomposition proposed by Stone that solves the three-dimensional linear Fokker–Planck equation. The method presented can manage the large set of linear equations that appears from the discretization procedure. The convergence of the iterative process is done in a fast and effective way. Also, this method can be easily adapted to support irregular boundaries with Dirichlet, Neumann or mixed boundary conditions and can be used to follow the evolution of a distribution function for non-stationary equations. In this case a Crank–Nicolson discretization in time is recommended because it has implicit character and is more stable than other methods, but strictly speaking, the stability of the system has to be studied in each particular case considered. The good properties that our method shares and the lack of methods that handle the large amount of data (given by the Fokker–Planck equation with six independent variables) make the method presented in this article worthy and advantageous. In general, the algorithm presented in the article can solve the system of equations that arises from the discretization of the Eq. (1) or similar equations (with or without the mixed derivatives), but we have to keep in mind that its convergence is conditioned to the form of the functions \dot{v} , $A(\mathbf{x}, v)$, $B(\mathbf{x}, v)$, $C(\mathbf{x}, v)$, $D(\mathbf{x}, v)$ and $E_{ij}(\mathbf{x}, v)$ that appear in the collision term (2).

Appendix A. Elements of $M = LU$ at node p

In this section we present how the elements of the matrix M are related to the elements of the upper U and lower L matrices:

$$\begin{aligned} M_{DD3}^p &= L_{DD3}^p, \\ M_{DD3|N}^p &= L_{DD3}^p U_N^{p-(n_{ijklm}+n_{ijkl})}, \\ M_{DD3|E}^p &= L_{DD3}^p U_E^{p-(n_{ijklm}+n_{ijkl})}, \\ M_{DD3|T}^p &= L_{DD3}^p U_T^{p-(n_{ijklm}+n_{ijkl})}, \\ M_{DD3|U1}^p &= L_{DD3}^p U_{U1}^{p-(n_{ijklm}+n_{ijkl})}, \\ M_{DB3}^p &= L_{DB3}^p + L_{DD3}^p U_{UB2}^{p-(n_{ijklm}+n_{ijkl})}, \end{aligned}$$

$$\begin{aligned}
M_{DB3|N}^p &= L_{DB3}^p U_N^{p-(n_{ijklm}+n_{ijk})}, \\
M_{DB3|E}^p &= L_{DB3}^p U_E^{p-(n_{ijklm}+n_{ijk})}, \\
M_{DB3|T}^p &= L_{DB3}^p U_T^{p-(n_{ijklm}+n_{ijk})}, \\
M_{D3}^p &= L_{D3}^p + L_{DB3}^p U_{U1}^{p-(n_{ijklm}+n_{ijk})} + L_{DD3}^p U_{U2}^{p-(n_{ijklm}+n_{ijk})}, \\
M_{D3|N}^p &= L_{D3}^p U_N^{p-n_{ijklm}}, \\
M_{D3|E}^p &= L_{D3}^p U_E^{p-n_{ijklm}}, \\
M_{D3|T}^p &= L_{D3}^p U_T^{p-n_{ijklm}}, \\
M_{DT3}^p &= L_{DT3}^p + L_{D3}^p U_{U1}^{p-n_{ijklm}} + L_{DD3}^p U_{UT2}^{p-(n_{ijklm}+n_{ijk})}, \\
M_{DT3|N}^p &= L_{DT3}^p U_N^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DT3|E}^p &= L_{DT3}^p U_E^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DT3|T}^p &= L_{DT3}^p U_T^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DT3|U1}^p &= L_{DT3}^p U_{U1}^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DB3|UB2}^p &= L_{DB3}^p U_{UB2}^{p-(n_{ijklm}+n_{ijk})}, \\
M_{D3|UB2}^p &= L_{D3}^p U_{UB2}^{p-n_{ijklm}} + L_{DB3}^p U_{U2}^{p-(n_{ijklm}+n_{ijk})}, \\
M_{DU3}^p &= L_{DU3}^p + L_{DT3}^p U_{UB2}^{p-(n_{ijklm}-n_{ijk})} + L_{D3}^p U_{U2}^{p-n_{ijklm}} + L_{DB3}^p U_{UT2}^{p-(n_{ijklm}+n_{ijk})}, \\
M_{DU3|N}^p &= L_{DU3}^p U_N^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DU3|E}^p &= L_{DU3}^p U_E^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DU3|T}^p &= L_{DU3}^p U_T^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DU3|U1}^p &= L_{DU3}^p U_{U1}^{p-(n_{ijklm}-n_{ijk})} + L_{DT3}^p U_{U2}^{p-(n_{ijklm}-n_{ijk})} + L_{D3}^p U_{UT2}^{p-n_{ijklm}}, \\
M_{DT3|UT2}^p &= L_{DT3}^p U_{UT2}^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DU3|UB2}^p &= L_{DU3}^p U_{UB2}^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DU3|U2}^p &= L_{DU3}^p U_{U2}^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DU3|UT2}^p &= L_{DU3}^p U_{UT2}^{p-(n_{ijklm}-n_{ijk})}, \\
M_{DD3|UD3}^p &= L_{DD3}^p U_{UD3}^{p-(n_{ijklm}+n_{ijk})}, \\
M_{DB2}^p &= L_{DB2}^p + L_{DB3}^p U_{UD3}^{p-(n_{ijklm}+n_{ijk})} + L_{DD3}^p U_{UB3}^{p-(n_{ijklm}+n_{ijk})}, \\
M_{DB2|N}^p &= L_{DB2}^p U_N^{p-(n_{ijkl}+n_{ijk})}, \\
M_{DB2|E}^p &= L_{DB2}^p U_E^{p-(n_{ijkl}+n_{ijk})}, \\
M_{DB2|T}^p &= L_{DB2}^p U_T^{p-(n_{ijkl}+n_{ijk})}, \\
M_{D2}^p &= L_{D2}^p + L_{DB2}^p U_{U1}^{p-(n_{ijkl}+n_{ijk})} + L_{D3}^p U_{UD3}^{p-n_{ijklm}} + L_{DD3}^p U_{U3}^{p-(n_{ijklm}+n_{ijk})}, \\
M_{D2|N}^p &= L_{D2}^p U_N^{p-n_{ijkl}}, \\
M_{D2|E}^p &= L_{D2}^p U_E^{p-n_{ijkl}}, \\
M_{D2|T}^p &= L_{D2}^p U_T^{p-n_{ijkl}}, \\
M_{DT2}^p &= L_{DT2}^p + L_{D2}^p U_{U1}^{p-n_{ijkl}} + L_{DT3}^p U_{UD3}^{p-(n_{ijklm}-n_{ijk})} + L_{DD3}^p U_{UT3}^{p-(n_{ijklm}+n_{ijk})},
\end{aligned}$$

$$\begin{aligned}
 M_{DT2|N}^p &= L_{DT2}^p U_N^{p-(n_{ijkl}-n_{ijk})}, \\
 M_{DT2|E}^p &= L_{DT2}^p U_E^{p-(n_{ijkl}-n_{ijk})}, \\
 M_{DT2|T}^p &= L_{DT2}^p U_T^{p-(n_{ijkl}-n_{ijk})}, \\
 M_{DT2|U1}^p &= L_{DT2}^p U_{U1}^{p-(n_{ijkl}-n_{ijk})}, \\
 M_{DB2|UB2}^p &= L_{DB2}^p U_{UB2}^{p-(n_{ijkl}+n_{ijk})} + L_{DB3}^p U_{UB3}^{p-(n_{ijklm}+n_{ijk})}, \\
 M_{D1}^p &= L_{D1}^p + L_{D2}^p U_{UB2}^{p-n_{ijkl}} + L_{DB2}^p U_{U2}^{p-(n_{ijkl}+n_{ijk})} + L_{D3}^p U_{UB3}^{p-n_{ijklm}} + L_{DB3}^p U_{U3}^{p-(n_{ijklm}+n_{ijk})}, \\
 M_{D1|N}^p &= L_{D1}^p U_N^{p-n_{ijk}}, \\
 M_{D1|E}^p &= L_{D1}^p U_E^{p-n_{ijk}}, \\
 M_{D1|T}^p &= L_{D1}^p U_T^{p-n_{ijk}}, \\
 M_B^p &= L_B^p, \\
 M_{B|N}^p &= L_B^p U_N^{p-n_{ij}}, \\
 M_{B|E}^p &= L_B^p U_E^{p-n_{ij}}, \\
 M_W^p &= L_W^p, \\
 M_{W|N}^p &= L_W^p U_N^{p-n_j}, \\
 M_S^p &= L_S^p, \\
 M_P^p &= L_P^p + L_S^p U_N^{p-1} + L_W^p U_E^{p-n_j} + L_B^p U_T^{p-n_{ij}} + L_{D1}^p U_{U1}^{p-n_{ijk}} + L_{DT2}^p U_{UB2}^{p-(n_{ijkl}-n_{ijk})} + L_{D2}^p U_{U2}^{p-n_{ijkl}} \\
 &\quad + L_{DB2}^p U_{UT2}^{p-(n_{ijkl}+n_{ijk})} + L_{DU3}^p U_{UD3}^{p-(n_{ijklm}-n_{ijkl})} + L_{DT3}^p U_{UB3}^{p-(n_{ijklm}-n_{ijk})} + L_{D3}^p U_{U3}^{p-n_{ijklm}} \\
 &\quad + L_{DB3}^p U_{UT3}^{p-(n_{ijklm}+n_{ijk})} + L_{DD3}^p U_{UU3}^{p-(n_{ijklm}+n_{ijkl})}, \\
 M_N^p &= L_P^p U_N^p, \\
 M_{S|E}^p &= L_S^p U_E^{p-1}, \\
 M_E^p &= L_P^p U_E^p, \\
 M_{W|T}^p &= L_W^p U_T^{p-n_j}, \\
 M_{S|T}^p &= L_S^p U_T^{p-1}, \\
 M_T^p &= L_P^p U_T^p, \\
 M_{B|U1}^p &= L_B^p U_{U1}^{p-n_{ij}}, \\
 M_{W|U1}^p &= L_W^p U_{U1}^{p-n_j}, \\
 M_{S|U1}^p &= L_S^p U_{U1}^{p-1}, \\
 M_{U1}^p &= L_P^p U_{U1}^p + L_{DT2}^p U_{U2}^{p-(n_{ijkl}-n_{ijk})} + L_{D2}^p U_{UT2}^{p-n_{ijkl}} + L_{DT3}^p U_{U3}^{p-(n_{ijklm}-n_{ijk})} + L_{D3}^p U_{UT3}^{p-n_{ijklm}}, \\
 M_{DT2|UT2}^p &= L_{DT2}^p U_{UT2}^{p-(n_{ijkl}-n_{ijk})} + L_{DT3}^p U_{UT3}^{p-(n_{ijklm}-n_{ijk})}, \\
 M_{D1|UB2}^p &= L_{D1}^p U_{UB2}^{p-n_{ijk}}, \\
 M_{B|UB2}^p &= L_B^p U_{UB2}^{p-n_{ij}}, \\
 M_{W|UB2}^p &= L_W^p U_{UB2}^{p-n_j}, \\
 M_{S|UB2}^p &= L_S^p U_{UB2}^{p-1}, \\
 M_{UB2}^p &= L_P^p U_{UB2}^p + L_{D1}^p U_{U2}^{p-n_{ijk}} + L_{DU3}^p U_{UB3}^{p-(n_{ijklm}-n_{ijkl})} + L_{DB3}^p U_{UU3}^{p-(n_{ijklm}+n_{ijk})}, \\
 M_{B|U2}^p &= L_B^p U_{U2}^{p-n_{ij}}, \\
 M_{W|U2}^p &= L_W^p U_{U2}^{p-n_j},
 \end{aligned}$$

$$\begin{aligned}
M_{S|U_2}^p &= L_S^p U_{U_2}^{p-1}, \\
M_{U_2}^p &= L_P^p U_{U_2}^p + L_{D_1}^p U_{UT_2}^{p-n_{ijk}} + L_{DU_3}^p U_{U_3}^{p-(n_{ijklm}-n_{ijkl})} + L_{D_3}^p U_{UU_3}^{p-n_{ijklm}}, \\
M_{B|UT_2}^p &= L_B^p U_{UT_2}^{p-n_{ij}}, \\
M_{W|UT_2}^p &= L_W^p U_{UT_2}^{p-n_j}, \\
M_{S|UT_2}^p &= L_S^p U_{UT_2}^{p-1}, \\
M_{UT_2}^p &= L_P^p U_{UT_2}^p + L_{DU_3}^p U_{UT_3}^{p-(n_{ijklm}-n_{ijkl})} + L_{DT_3}^p U_{UU_3}^{p-(n_{ijklm}-n_{ijkl})}, \\
M_{DU_3|UU_3}^p &= L_{DU_3}^p U_{UU_3}^{p-(n_{ijklm}-n_{ijkl})}, \\
M_{DB_2|UD_3}^p &= L_{DB_2}^p U_{UD_3}^{p-(n_{ijkl}+n_{ijk})}, \\
M_{D_2|UD_3}^p &= L_{D_2}^p U_{UD_3}^{p-n_{ijkl}}, \\
M_{DT_2|UD_3}^p &= L_{DT_2}^p U_{UD_3}^{p-(n_{ijkl}-n_{ijk})}, \\
M_{DB_2|UB_3}^p &= L_{DB_2}^p U_{UB_3}^{p-(n_{ijkl}+n_{ijk})}, \\
M_{D_1|UD_3}^p &= L_{D_1}^p U_{UD_3}^{p-n_{ijk}} + L_{D_2}^p U_{UB_3}^{p-n_{ijkl}} + L_{DB_2}^p U_{U_3}^{p-(n_{ijkl}+n_{ijk})}, \\
M_{B|UD_3}^p &= L_B^p U_{UD_3}^{p-n_{ij}}, \\
M_{W|UD_3}^p &= L_W^p U_{UD_3}^{p-n_j}, \\
M_{S|UD_3}^p &= L_S^p U_{UD_3}^{p-1}, \\
M_{UD_3}^p &= L_P^p U_{UD_3}^p + L_{DT_2}^p U_{UB_3}^{p-(n_{ijkl}-n_{ijk})} + L_{D_2}^p U_{U_3}^{p-n_{ijkl}} + L_{DB_2}^p U_{UT_3}^{p-(n_{ijkl}+n_{ijk})}, \\
M_{DT_2|U_3}^p &= L_{DT_2}^p U_{U_3}^{p-(n_{ijkl}-n_{ijk})} + L_{D_2}^p U_{UT_3}^{p-n_{ijkl}}, \\
M_{DT_2|UT_3}^p &= L_{DT_2}^p U_{UT_3}^{p-(n_{ijkl}-n_{ijk})}, \\
M_{D_1|UB_3}^p &= L_{D_1}^p U_{UB_3}^{p-n_{ijk}}, \\
M_{B|UB_3}^p &= L_B^p U_{UB_3}^{p-n_{ij}}, \\
M_{W|UB_3}^p &= L_W^p U_{UB_3}^{p-n_j}, \\
M_{S|UB_3}^p &= L_S^p U_{UB_3}^{p-1}, \\
M_{UB_3}^p &= L_P^p U_{UB_3}^p + L_{D_1}^p U_{U_3}^{p-n_{ijk}} + L_{DB_2}^p U_{UU_3}^{p-(n_{ijkl}+n_{ijk})}, \\
M_{B|U_3}^p &= L_B^p U_{U_3}^{p-n_{ij}}, \\
M_{W|U_3}^p &= L_W^p U_{U_3}^{p-n_j}, \\
M_{S|U_3}^p &= L_S^p U_{U_3}^{p-1}, \\
M_{U_3}^p &= L_P^p U_{U_3}^p + L_{D_1}^p U_{UT_3}^{p-n_{ijk}} + L_{D_2}^p U_{UU_3}^{p-n_{ijkl}}, \\
M_{B|UT_3}^p &= L_B^p U_{UT_3}^{p-n_{ij}}, \\
M_{W|UT_3}^p &= L_W^p U_{UT_3}^{p-n_j}, \\
M_{S|UT_3}^p &= L_S^p U_{UT_3}^{p-1}, \\
M_{UT_3}^p &= L_P^p U_{UT_3}^p + L_{DT_2}^p U_{UU_3}^{p-(n_{ijkl}-n_{ijk})}, \\
M_{D_1|UU_3}^p &= L_{D_1}^p U_{UU_3}^{p-n_{ijk}}, \\
M_{B|UU_3}^p &= L_B^p U_{UU_3}^{p-n_{ij}}, \\
M_{W|UU_3}^p &= L_W^p U_{UU_3}^{p-n_j}, \\
M_{S|UU_3}^p &= L_S^p U_{UU_3}^{p-1}, \\
M_{UU_3}^p &= L_P^p U_{UU_3}^p.
\end{aligned} \tag{A1}$$

Appendix B. Explicit form of Eq. (9)

In this section we present the explicit form of $O\Psi \approx 0$ for each grid node:

$$\begin{aligned}
& O_{DD3}\Psi_{DD3} + O_{DB3}\Psi_{DB3} + O_{D3}\Psi_{D3} + O_{DT3}\Psi_{DT3} + O_{DU3}\Psi_{DU3} + O_{DB2}\Psi_{DB2} + O_{D2}\Psi_{D2} + O_{DT2}\Psi_{DT2} \\
& + O_{D1}\Psi_{D1} + O_B\Psi_B + O_W\Psi_W + O_S\Psi_S + O_P\Psi_P + O_N\Psi_N + O_E\Psi_E + O_T\Psi_T + O_{U1}\Psi_{U1} \\
& + O_{UB2}\Psi_{UB2} + O_{U2}\Psi_{U2} + O_{UT2}\Psi_{UT2} + O_{UD3}\Psi_{UD3} + O_{UB3}\Psi_{UB3} + O_{U3}\Psi_{U3} + O_{UT3}\Psi_{UT3} \\
& + O_{UU3}\Psi_{UU3} + M_{DD3|N}\Psi_{DD3|N} + M_{DD3|E}\Psi_{DD3|E} + M_{DD3|T}\Psi_{DD3|T} + M_{DD3|U1}\Psi_{DD3|U1} \\
& + M_{DB3|N}\Psi_{DB3|N} + M_{DB3|E}\Psi_{DB3|E} + M_{DB3|T}\Psi_{DB3|T} + M_{D3|N}\Psi_{D3|N} + M_{D3|E}\Psi_{D3|E} \\
& + M_{D3|T}\Psi_{D3|T} + M_{DT3|N}\Psi_{DT3|N} + M_{DT3|E}\Psi_{DT3|E} + M_{DT3|T}\Psi_{DT3|T} + M_{DT3|U1}\Psi_{DT3|U1} \\
& + M_{DB3|UB2}\Psi_{DB3|UB2} + M_{DB3|U2}\Psi_{DB3|U2} + M_{D3|UB2}\Psi_{D3|UB2} + M_{DU3|N}\Psi_{DU3|N} + M_{DU3|E}\Psi_{DU3|E} \\
& + M_{DU3|T}\Psi_{DU3|T} + M_{D3|UT2}\Psi_{D3|UT2} + M_{DT3|U2}\Psi_{DT3|U2} + M_{DU3|U1}\Psi_{DU3|U1} + M_{DT3|UT2}\Psi_{DT3|UT2} \\
& + M_{DU3|UB2}\Psi_{DU3|UB2} + M_{DU3|U2}\Psi_{DU3|U2} + M_{DU3|UT2}\Psi_{DU3|UT2} + M_{DD3|UD3}\Psi_{DD3|UD3} \\
& + M_{DB2|N}\Psi_{DB2|N} + M_{DB2|E}\Psi_{DB2|E} + M_{DB2|T}\Psi_{DB2|T} + M_{D2|N}\Psi_{D2|N} + M_{D2|E}\Psi_{D2|E} \\
& + M_{D2|T}\Psi_{D2|T} + M_{DT2|N}\Psi_{DT2|N} + M_{DT2|E}\Psi_{DT2|E} + M_{DT2|T}\Psi_{DT2|T} + M_{DT2|U1}\Psi_{DT2|U1} \\
& + M_{DB3|UB3}\Psi_{DB3|UB3} + M_{DB2|UB2}\Psi_{DB2|UB2} + M_{D1|N}\Psi_{D1|N} + M_{D1|E}\Psi_{D1|E} + M_{D1|T}\Psi_{D1|T} \\
& + M_{B|N}\Psi_{B|N} + M_{B|E}\Psi_{B|E} + M_{W|N}\Psi_{W|N} + M_{S|E}\Psi_{S|E} + M_{W|T}\Psi_{W|T} + M_{S|T}\Psi_{S|T} + M_{B|U1}\Psi_{B|U1} \\
& + M_{W|U1}\Psi_{W|U1} + M_{S|U1}\Psi_{S|U1} + M_{DT2|UT2}\Psi_{DT2|UT2} + M_{DT3|UT3}\Psi_{DT3|UT3} + M_{D1|UB2}\Psi_{D1|UB2} \\
& + M_{B|UB2}\Psi_{B|UB2} + M_{W|UB2}\Psi_{W|UB2} + M_{S|UB2}\Psi_{S|UB2} + M_{B|U2}\Psi_{B|U2} + M_{W|U2}\Psi_{W|U2} \\
& + M_{S|U2}\Psi_{S|U2} + M_{B|UT2}\Psi_{B|UT2} + M_{W|UT2}\Psi_{W|UT2} + M_{S|UT2}\Psi_{S|UT2} + M_{DU3|UU3}\Psi_{DU3|UU3} \\
& + M_{DB2|UD3}\Psi_{DB2|UD3} + M_{D2|UD3}\Psi_{D2|UD3} + M_{DT2|UD3}\Psi_{DT2|UD3} + M_{DB2|UB3}\Psi_{DB2|UB3} \\
& + M_{D1|UD3}\Psi_{D1|UD3} + M_{D2|UB3}\Psi_{D2|UB3} + M_{DB2|U3}\Psi_{DB2|U3} + M_{B|UD3}\Psi_{B|UD3} \\
& + M_{W|UD3}\Psi_{W|UD3} + M_{S|UD3}\Psi_{S|UD3} + M_{DT2|U3}\Psi_{DT2|U3} + M_{D2|UT3}\Psi_{D2|UT3} + M_{DT2|UT3}\Psi_{DT2|UT3} \\
& + M_{D1|UB3}\Psi_{D1|UB3} + M_{B|UB3}\Psi_{B|UB3} + M_{W|UB3}\Psi_{W|UB3} + M_{S|UB3}\Psi_{S|UB3} \\
& + M_{B|U3}\Psi_{B|U3} + M_{W|U3}\Psi_{W|U3} + M_{S|U3}\Psi_{S|U3} + M_{B|UT3}\Psi_{B|UT3} + M_{W|UT3}\Psi_{W|UT3} + M_{S|UT3}\Psi_{S|UT3} \\
& + M_{D1|UU3}\Psi_{D1|UU3} + M_{B|UU3}\Psi_{B|UU3} + M_{W|UU3}\Psi_{W|UU3} + M_{S|UU3}\Psi_{S|UU3} \approx 0.
\end{aligned} \tag{B1}$$

Appendix C. O as a linear combination of M

In this section we present the elements of the matrix O as a linear combination of the elements of matrix M using the relations (10):

$$\begin{aligned}
O_{DD3} &= -\alpha(M_{DD3|N} + M_{DD3|E} + M_{DD3|T} + M_{DD3|U1} + M_{DD3|UD3}), \\
O_{DB3} &= -\alpha(M_{DB3|N} + M_{DB3|E} + M_{DB3|T} + M_{DB3|UB2} + M_{DB3|U2} + M_{DB3|UB3}), \\
O_{D3} &= -\alpha(M_{D3|N} + M_{D3|E} + M_{D3|T} + M_{D3|UB2} + M_{D3|UT2}), \\
O_{DT3} &= -\alpha(M_{DT3|N} + M_{DT3|E} + M_{DT3|T} + M_{DT3|U1} + M_{DT3|U2} + M_{DT3|UT2} + M_{DT3|UT3}), \\
O_{DU3} &= -\alpha(M_{DU3|N} + M_{DU3|E} + M_{DU3|T} + M_{DU3|U1} + M_{DU3|UB2} + M_{DU3|U2} + M_{DU3|UT2} + M_{DU3|UU3}), \\
O_{DB2} &= -\alpha(M_{DB2|N} + M_{DB2|E} + M_{DB2|T} + M_{DB2|UB2} + M_{DB2|UD3} + M_{DB2|UB3} + M_{DB2|U3}), \\
O_{D2} &= -\alpha(M_{D2|N} + M_{D2|E} + M_{D2|T} + M_{D2|UD3} + M_{D2|UB3} + M_{D2|UT3}), \\
O_{DT2} &= -\alpha(M_{DT2|N} + M_{DT2|E} + M_{DT2|T} + M_{DT2|U1} + M_{DT2|UT2} + M_{DT2|UD3} + M_{DT2|U3} + M_{DT2|UT3}), \\
O_{D1} &= -\alpha(M_{D1|N} + M_{D1|E} + M_{D1|T} + M_{D1|UB2} + M_{D1|UD3} + M_{D1|UB3} + M_{D1|UU3}), \\
O_B &= -\alpha(M_{B|N} + M_{B|E} + M_{B|U1} + M_{B|UB2} + M_{B|U2} + M_{B|UT2} + M_{B|UD3} + M_{B|UB3} + M_{B|U3} + M_{B|UT3} + M_{B|UU3}), \\
O_W &= -\alpha(M_{W|N} + M_{W|T} + M_{W|U1} + M_{W|UB2} + M_{W|U2} + M_{W|UT2} + M_{W|UD3} + M_{W|UB3} \\
&\quad + M_{W|U3} + M_{W|UT3} + M_{W|UU3}),
\end{aligned}$$

$$\begin{aligned}
O_S &= -\alpha(M_{S|E} + M_{S|T} + M_{S|U1} + M_{S|UB2} + M_{S|U2} + M_{S|UT2} + M_{S|UD3} + M_{S|UB3} + M_{S|U3} + M_{S|UT3} + M_{S|UU3}), \\
O_P &= \alpha(M_{DD3|N} + M_{DD3|E} + M_{DD3|T} + M_{DD3|U1} + M_{DB3|N} + M_{DB3|E} + M_{DB3|T} + M_{D3|N} + M_{D3|E} + M_{D3|T} \\
&\quad + M_{DT3|N} + M_{DT3|E} + M_{DT3|T} + M_{DT3|U1} + M_{DB3|UB2} + M_{DB3|U2} + M_{D3|UB2} + M_{DU3|N} + M_{DU3|E} \\
&\quad + M_{DU3|T} + M_{D3|UT2} + M_{DT3|U2} + M_{DU3|U1} + M_{DT3|UT2} + M_{DU3|UB2} + M_{DU3|U2} + M_{DU3|UT2} \\
&\quad + M_{DD3|UD3} + M_{DB2|N} + M_{DB2|E} + M_{DB2|T} + M_{D2|N} + M_{D2|E} + M_{D2|T} + M_{DT2|N} + M_{DT2|E} + M_{DT2|T} \\
&\quad + M_{DT2|U1} + M_{DB3|UB3} + M_{DB2|UB2} + M_{D1|N} + M_{D1|E} + M_{D1|T} + M_{B|N} + M_{B|E} + M_{W|N} + M_{S|E} \\
&\quad + M_{W|T} + M_{S|T} + M_{B|U1} + M_{W|U1} + M_{S|U1} + M_{DT2|UT2} + M_{DT3|UT3} + M_{D1|UB2} + M_{B|UB2} + M_{W|UB2} \\
&\quad + M_{S|UB2} + M_{B|U2} + M_{W|U2} + M_{S|U2} + M_{B|UT2} + M_{W|UT2} + M_{S|UT2} + M_{DU3|UU3} + M_{DB2|UD3} \\
&\quad + M_{D2|UD3} + M_{DT2|UD3} + M_{DB2|UB3} + M_{D1|UD3} + M_{D2|UB3} + M_{DB2|U3} + M_{B|UD3} + M_{W|UD3} + M_{S|UD3} \\
&\quad + M_{DT2|U3} + M_{D2|UT3} + M_{DT2|UT3} + M_{D1|UB3} + M_{B|UB3} + M_{W|UB3} + M_{S|UB3} + M_{B|U3} + M_{W|U3} \\
&\quad + M_{S|U3} + M_{B|UT3} + M_{W|UT3} + M_{S|UT3} + M_{D1|UU3} + M_{B|UU3} + M_{W|UU3} + M_{S|UU3}), \\
O_N &= -\alpha(M_{DD3|N} + M_{DB3|N} + M_{D3|N} + M_{DT3|N} + M_{DU3|N} + M_{DB2|N} + M_{D2|N} + M_{DT2|N} \\
&\quad + M_{D1|N} + M_{B|N} + M_{W|N}), \\
O_E &= -\alpha(M_{DD3|E} + M_{DB3|E} + M_{D3|E} + M_{DT3|E} + M_{DU3|E} + M_{DB2|E} + M_{D2|E} + M_{DT2|E} \\
&\quad + M_{D1|E} + M_{B|E} + M_{S|E}), \\
O_T &= -\alpha(M_{DD3|T} + M_{DB3|T} + M_{D3|T} + M_{DT3|T} + M_{DU3|T} + M_{DB2|T} + M_{D2|T} + M_{DT2|T} \\
&\quad + M_{D1|T} + M_{W|T} + M_{S|T}), \\
O_{U1} &= -\alpha(M_{DD3|U1} + M_{DT3|U1} + M_{DU3|U1} + M_{DT2|U1} + M_{B|U1} + M_{W|U1} + M_{S|U1}), \\
O_{UB2} &= -\alpha(M_{DB3|UB2} + M_{D3|UB2} + M_{DU3|UB2} + M_{DB2|UB2} + M_{D1|UB2} + M_{B|UB2} + M_{W|UB2} + M_{S|UB2}), \\
O_{U2} &= -\alpha(M_{DB3|U2} + M_{DT3|U2} + M_{DU3|U2} + M_{B|U2} + M_{W|U2} + M_{S|U2}), \\
O_{UT2} &= -\alpha(M_{D3|UT2} + M_{DT3|UT2} + M_{DU3|UT2} + M_{DT2|UT2} + M_{B|UT2} + M_{W|UT2} + M_{S|UT2}), \\
O_{UD3} &= -\alpha(M_{DD3|UD3} + M_{DB2|UD3} + M_{D2|UD3} + M_{DT2|UD3} + M_{D1|UD3} + M_{B|UD3} + M_{W|UD3} + M_{S|UD3}), \\
O_{UB3} &= -\alpha(M_{DB3|UB3} + M_{DB2|UB3} + M_{D2|UB3} + M_{D1|UB3} + M_{B|UB3} + M_{W|UB3} + M_{S|UB3}), \\
O_{U3} &= -\alpha(M_{DB2|U3} + M_{DT2|U3} + M_{B|U3} + M_{W|U3} + M_{S|U3}), \\
O_{UT3} &= -\alpha(M_{DT3|UT3} + M_{D2|UT3} + M_{DT2|UT3} + M_{B|UT3} + M_{W|UT3} + M_{S|UT3}), \\
O_{UU3} &= -\alpha(M_{DU3|UU3} + M_{D1|UU3} + M_{B|UU3} + M_{W|UU3} + M_{S|UU3}). \tag{C1}
\end{aligned}$$

Appendix D. Explicit form of the functions $C_{[X]}$ and $K_{[X]}$

In this section we show the functions $C_{[X]}$ and $K_{[X]}$ that appear in the final expression (11) of the LU decomposition:

$$\begin{aligned}
C_{DB3} &= L_{DD3}^p U_{UB2}^{p-(n_{ijklm}+n_{ijk})}, \\
C_{D3} &= L_{DD3}^p U_{U2}^{p-(n_{ijklm}+n_{ijk})} + L_{DB3}^p U_{U1}^{p-(n_{ijklm}+n_{ijk})}, \\
C_{DT3} &= L_{DD3}^p U_{UT2}^{p-(n_{ijklm}+n_{ijk})} + L_{D3}^p U_{U1}^{p-n_{ijklm}}, \\
C_{DU3} &= L_{DB3}^p U_{UT2}^{p-(n_{ijklm}+n_{ijk})} + L_{D3}^p U_{U2}^{p-n_{ijklm}} + L_{DT3}^p U_{UB2}^{p-(n_{ijklm}-n_{ijk})}, \\
C_{DB2} &= L_{DD3}^p U_{UB3}^{p-(n_{ijklm}+n_{ijk})} + L_{DB3}^p U_{UD3}^{p-(n_{ijklm}+n_{ijk})}, \\
C_{D2} &= L_{DD3}^p U_{U3}^{p-(n_{ijklm}+n_{ijk})} + L_{D3}^p U_{UD3}^{p-n_{ijklm}} + L_{DB2}^p U_{U1}^{p-(n_{ijkl}+n_{ijk})}, \\
C_{DT2} &= L_{DD3}^p U_{UT3}^{p-(n_{ijklm}+n_{ijk})} + L_{DT3}^p U_{UD3}^{p-(n_{ijklm}-n_{ijk})} + L_{D2}^p U_{U1}^{p-n_{ijkl}}, \\
C_{D1} &= L_{DB3}^p U_{U3}^{p-(n_{ijklm}+n_{ijk})} + L_{D3}^p U_{UB3}^{p-n_{ijklm}} + L_{DB2}^p U_{U2}^{p-(n_{ijkl}+n_{ijk})} + L_{D2}^p U_{UB2}^{p-n_{ijkl}}, \\
C_{U1} &= L_{DT2}^p U_{U2}^{p-(n_{ijkl}-n_{ijk})} + L_{D2}^p U_{UT2}^{p-n_{ijkl}} + L_{DT3}^p U_{U3}^{p-(n_{ijklm}-n_{ijk})} + L_{D3}^p U_{UT3}^{p-n_{ijklm}},
\end{aligned}$$

$$\begin{aligned}
 C_{UB2} &= L_{D1}^p U_{U2}^{p-n_{ijk}} + L_{DU3}^p U_{UB3}^{p-(n_{ijklm}-n_{ijkl})} + L_{DB3}^p U_{UU3}^{p-(n_{ijklm}+n_{ijk})}, \\
 C_{U2} &= L_{D1}^p U_{UT2}^{p-n_{ijk}} + L_{DU3}^p U_{U3}^{p-(n_{ijklm}-n_{ijkl})} + L_{D3}^p U_{UU3}^{p-n_{ijklm}}, \\
 C_{UT2} &= L_{DU3}^p U_{UT3}^{p-(n_{ijklm}-n_{ijkl})} + L_{DT3}^p U_{UU3}^{p-(n_{ijklm}-n_{ijk})}, \\
 C_{UD3} &= L_{DT2}^p U_{UB3}^{p-(n_{ijkl}-n_{ijk})} + L_{D2}^p U_{U3}^{p-n_{ijkl}} + L_{DB2}^p U_{UT3}^{p-(n_{ijkl}+n_{ijk})}, \\
 C_{UB3} &= L_{D1}^p U_{U3}^{p-n_{ijk}} + L_{DB2}^p U_{UU3}^{p-(n_{ijkl}+n_{ijk})}, \\
 C_{U3} &= L_{D1}^p U_{UT3}^{p-n_{ijk}} + L_{D2}^p U_{UU3}^{p-n_{ijkl}}, \\
 C_{UT3} &= L_{DT2}^p U_{UU3}^{p-(n_{ijkl}-n_{ijk})}, \\
 C_{DD3} &= C_B = C_W = C_S = C_N = C_E = C_T = C_{UU3} = 0, \\
 K_{DD3} &= U_N^{p-(n_{ijklm}+n_{ijkl})} + U_E^{p-(n_{ijklm}+n_{ijkl})} + U_T^{p-(n_{ijklm}+n_{ijkl})} + U_{U1}^{p-(n_{ijklm}+n_{ijkl})} + U_{UD3}^{p-(n_{ijklm}+n_{ijkl})}, \\
 K_{DB3} &= U_N^{p-(n_{ijklm}+n_{ijk})} + U_E^{p-(n_{ijklm}+n_{ijk})} + U_T^{p-(n_{ijklm}+n_{ijk})} + U_{UB2}^{p-(n_{ijklm}+n_{ijk})} + U_{U2}^{p-(n_{ijklm}+n_{ijk})} + U_{UB3}^{p-(n_{ijklm}+n_{ijk})}, \\
 K_{D3} &= U_N^{p-n_{ijklm}} + U_E^{p-n_{ijklm}} + U_T^{p-n_{ijklm}} + U_{UB2}^{p-n_{ijklm}} + U_{UT2}^{p-n_{ijklm}}, \\
 K_{DT3} &= U_N^{p-(n_{ijklm}-n_{ijk})} + U_E^{p-(n_{ijklm}-n_{ijk})} + U_T^{p-(n_{ijklm}-n_{ijk})} + U_{U1}^{p-(n_{ijklm}-n_{ijk})} + U_{U2}^{p-(n_{ijklm}-n_{ijk})} \\
 &\quad + U_{UT2}^{p-(n_{ijklm}-n_{ijk})} + U_{UT3}^{p-(n_{ijklm}-n_{ijk})}, \\
 K_{DU3} &= U_N^{p-(n_{ijklm}-n_{ijkl})} + U_E^{p-(n_{ijklm}-n_{ijkl})} + U_T^{p-(n_{ijklm}-n_{ijkl})} + U_{U1}^{p-(n_{ijklm}-n_{ijkl})} + U_{UB2}^{p-(n_{ijklm}-n_{ijkl})} \\
 &\quad + U_{U2}^{p-(n_{ijklm}-n_{ijkl})} + U_{UT2}^{p-(n_{ijklm}-n_{ijkl})} + U_{UU3}^{p-(n_{ijklm}-n_{ijkl})}, \\
 K_{DB2} &= U_N^{p-(n_{ijkl}+n_{ijk})} + U_E^{p-(n_{ijkl}+n_{ijk})} + U_T^{p-(n_{ijkl}+n_{ijk})} + U_{UB2}^{p-(n_{ijkl}+n_{ijk})} + U_{UD3}^{p-(n_{ijkl}+n_{ijk})} + U_{UB3}^{p-(n_{ijkl}+n_{ijk})} \\
 &\quad + U_{U3}^{p-(n_{ijkl}+n_{ijk})}, \\
 K_{D2} &= U_N^{p-n_{ijkl}} + U_E^{p-n_{ijkl}} + U_T^{p-n_{ijkl}} + U_{UD3}^{p-n_{ijkl}} + U_{UB3}^{p-n_{ijkl}} + U_{UT3}^{p-n_{ijkl}}, \\
 K_{DT2} &= U_N^{p-(n_{ijkl}-n_{ijk})} + U_E^{p-(n_{ijkl}-n_{ijk})} + U_T^{p-(n_{ijkl}-n_{ijk})} + U_{U1}^{p-(n_{ijkl}-n_{ijk})} + U_{UT2}^{p-(n_{ijkl}-n_{ijk})} + U_{UD3}^{p-(n_{ijkl}-n_{ijk})} \\
 &\quad + U_{U3}^{p-(n_{ijkl}-n_{ijk})} + U_{UT3}^{p-(n_{ijkl}-n_{ijk})}, \\
 K_{D1} &= U_N^{p-n_{ijk}} + U_E^{p-n_{ijk}} + U_T^{p-n_{ijk}} + U_{UB2}^{p-n_{ijk}} + U_{UD3}^{p-n_{ijk}} + U_{UB3}^{p-n_{ijk}} + U_{UU3}^{p-n_{ijk}}, \\
 K_B &= U_N^{p-n_{ij}} + U_E^{p-n_{ij}} + U_{U1}^{p-n_{ij}} + U_{UB2}^{p-n_{ij}} + U_{U2}^{p-n_{ij}} + U_{UT2}^{p-n_{ij}} + U_{UD3}^{p-n_{ij}} + U_{UB3}^{p-n_{ij}} + U_{U3}^{p-n_{ij}} + U_{UT3}^{p-n_{ij}} + U_{UU3}^{p-n_{ij}}, \\
 K_W &= U_N^{p-n_j} + U_T^{p-n_j} + U_{U1}^{p-n_j} + U_{UB2}^{p-n_j} + U_{U2}^{p-n_j} + U_{UT2}^{p-n_j} + U_{UD3}^{p-n_j} + U_{UB3}^{p-n_j} + U_{U3}^{p-n_j} + U_{UT3}^{p-n_j} + U_{UU3}^{p-n_j}, \\
 K_S &= U_E^{p-1} + U_T^{p-1} + U_{U1}^{p-1} + U_{UB2}^{p-1} + U_{U2}^{p-1} + U_{UT2}^{p-1} + U_{UD3}^{p-1} + U_{UB3}^{p-1} + U_{U3}^{p-1} + U_{UT3}^{p-1} + U_{UU3}^{p-1}, \\
 K_N &= L_{DD3}^p U_N^{p-(n_{ijklm}+n_{ijkl})} + L_{DB3}^p U_N^{p-(n_{ijklm}+n_{ijkl})} + L_{D3}^p U_N^{p-n_{ijklm}} + L_{DT3}^p U_N^{p-(n_{ijklm}-n_{ijk})} \\
 &\quad + L_{DU3}^p U_N^{p-(n_{ijklm}-n_{ijkl})} + L_{DB2}^p U_N^{p-(n_{ijkl}+n_{ijk})} + L_{D2}^p U_N^{p-n_{ijkl}} + L_{DT2}^p U_N^{p-(n_{ijkl}-n_{ijk})} + L_{D1}^p U_N^{p-n_{ijk}} \\
 &\quad + L_B^p U_N^{p-n_{ij}} + L_W^p U_N^{p-n_j}, \\
 K_E &= L_{DD3}^p U_E^{p-(n_{ijklm}+n_{ijkl})} + L_{DB3}^p U_E^{p-(n_{ijklm}+n_{ijkl})} + L_{D3}^p U_E^{p-n_{ijklm}} + L_{DT3}^p U_E^{p-(n_{ijklm}-n_{ijk})} \\
 &\quad + L_{DU3}^p U_E^{p-(n_{ijklm}-n_{ijkl})} + L_{DB2}^p U_E^{p-(n_{ijkl}+n_{ijk})} + L_{D2}^p U_E^{p-n_{ijkl}} + L_{DT2}^p U_E^{p-(n_{ijkl}-n_{ijk})} + L_{D1}^p U_E^{p-n_{ijk}} \\
 &\quad + L_B^p U_E^{p-n_{ij}} + L_S^p U_E^{p-1}, \\
 K_T &= L_{DD3}^p U_T^{p-(n_{ijklm}+n_{ijkl})} + L_{DB3}^p U_T^{p-(n_{ijklm}+n_{ijkl})} + L_{D3}^p U_T^{p-n_{ijklm}} + L_{DT3}^p U_T^{p-(n_{ijklm}-n_{ijk})} \\
 &\quad + L_{DU3}^p U_T^{p-(n_{ijklm}-n_{ijkl})} + L_{DB2}^p U_T^{p-(n_{ijkl}+n_{ijk})} + L_{D2}^p U_T^{p-n_{ijkl}} + L_{DT2}^p U_T^{p-(n_{ijkl}-n_{ijk})} + L_{D1}^p U_T^{p-n_{ijk}} \\
 &\quad + L_W^p U_T^{p-n_j} + L_S^p U_T^{p-1}, \\
 K_{U1} &= L_{DD3}^p U_{U1}^{p-(n_{ijklm}+n_{ijkl})} + L_{DT3}^p U_{U1}^{p-(n_{ijklm}-n_{ijk})} + L_{DU3}^p U_{U1}^{p-(n_{ijklm}-n_{ijkl})} + L_{DT2}^p U_{U1}^{p-(n_{ijkl}-n_{ijk})} \\
 &\quad + L_B^p U_{U1}^{p-n_{ij}} + L_W^p U_{U1}^{p-n_j} + L_S^p U_{U1}^{p-1},
 \end{aligned}$$

$$\begin{aligned}
K_{UB2} &= L_{DB3}^p U_{UB2}^{p-(n_{ijklm}+n_{ijk})} + L_{D3}^p U_{UB2}^{p-n_{ijklm}} + L_{DU3}^p U_{UB2}^{p-(n_{ijklm}-n_{ijkl})} + L_{DB2}^p U_{UB2}^{p-(n_{ijk}+n_{ijk})} + L_{D1}^p U_{UB2}^{p-n_{ijk}} \\
&\quad + L_B^p U_{UB2}^{p-n_{ij}} + L_W^p U_{UB2}^{p-n_j} + L_S^p U_{UB2}^{p-1}, \\
K_{U2} &= L_{DB3}^p U_{U2}^{p-(n_{ijklm}+n_{ijk})} + L_{DT3}^p U_{U2}^{p-(n_{ijklm}-n_{ijk})} + L_{DU3}^p U_{U2}^{p-(n_{ijklm}-n_{ijkl})} + L_B^p U_{U2}^{p-n_{ij}} + L_W^p U_{U2}^{p-n_j} + L_S^p U_{U2}^{p-1}, \\
K_{UT2} &= L_{D3}^p U_{UT2}^{p-n_{ijklm}} + L_{DT3}^p U_{UT2}^{p-(n_{ijklm}-n_{ijk})} + L_{DU3}^p U_{UT2}^{p-(n_{ijklm}-n_{ijkl})} + L_{DT2}^p U_{UT2}^{p-(n_{ijk}-n_{ijk})} + L_B^p U_{UT2}^{p-n_{ij}} \\
&\quad + L_W^p U_{UT2}^{p-n_j} + L_S^p U_{UT2}^{p-1}, \\
K_{UD3} &= L_{DB3}^p U_{UD3}^{p-(n_{ijklm}+n_{ijk})} + L_{DB2}^p U_{UD3}^{p-(n_{ijkl}+n_{ijk})} + L_{D2}^p U_{UD3}^{p-n_{ijkl}} + L_{DT2}^p U_{UD3}^{p-(n_{ijkl}-n_{ijk})} + L_{D1}^p U_{UD3}^{p-n_{ijk}} \\
&\quad + L_B^p U_{UD3}^{p-n_{ij}} + L_W^p U_{UD3}^{p-n_j} + L_S^p U_{UD3}^{p-1}, \\
K_{UB3} &= L_{DB3}^p U_{UB3}^{p-(n_{ijklm}+n_{ijk})} + L_{DB2}^p U_{UB3}^{p-(n_{ijkl}+n_{ijk})} + L_{D2}^p U_{UB3}^{p-n_{ijkl}} + L_{D1}^p U_{UB3}^{p-n_{ijk}} + L_B^p U_{UB3}^{p-n_{ij}} + L_W^p U_{UB3}^{p-n_j} \\
&\quad + L_S^p U_{UB3}^{p-1}, \\
K_{U3} &= L_{DB2}^p U_{U3}^{p-(n_{ijkl}+n_{ijk})} + L_{DT2}^p U_{U3}^{p-(n_{ijkl}-n_{ijk})} + L_B^p U_{U3}^{p-n_{ij}} + L_W^p U_{U3}^{p-n_j} + L_S^p U_{U3}^{p-1}, \\
K_{UT3} &= L_{DT3}^p U_{UT3}^{p-(n_{ijklm}-n_{ijk})} + L_{D2}^p U_{UT3}^{p-n_{ijkl}} + L_{DT2}^p U_{UT3}^{p-(n_{ijkl}-n_{ijk})} + L_B^p U_{UT3}^{p-n_{ij}} + L_W^p U_{UT3}^{p-n_j} + L_S^p U_{UT3}^{p-1}, \\
K_{UU3} &= L_{DU3}^p U_{UU3}^{p-(n_{ijklm}-n_{ijkl})} + L_{D1}^p U_{UU3}^{p-n_{ijk}} + L_B^p U_{UU3}^{p-n_{ij}} + L_W^p U_{UU3}^{p-n_j} + L_S^p U_{UU3}^{p-1}, \tag{D1}
\end{aligned}$$

References

- [1] L.E. Reichl, A Modern Course in Statistical Physics, University of Texas Press, 1980.
- [2] K. Huang, Statistical Mechanics, second ed., John Wiley, 1987.
- [3] M. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Stand. 49 (1952) 409–436.
- [4] C.F. Gerald, P.O. Wheatley, Applied Numerical Analysis, fifth ed., Addison-Wesley, Reading, MA, 1994.
- [5] J. Binney, S. Tremaine, Galactic Dynamics, Princeton University Press, Princeton, 1987.
- [6] J.H. Jeans, On the theory of star-streaming and the structure of the universe, MNRAS 76 (1915) 70–84.
- [7] R.B. Larson, A method for computing the evolution of star clusters, MNRAS 147 (1970) 323–337.
- [8] R.B. Larson, The evolution of star clusters, MNRAS 150 (1970) 93–110.
- [9] D. Lynden-Bell, P.P. Eggleton, On the consequences of the gravothermal catastrophe, MNRAS 191 (1980) 483–498.
- [10] M. Hénon, Monte Carlo models of star clusters, Astrophys. Space Sci. 13 (1971) 284–299.
- [11] M. Hénon, The Monte Carlo method, Astrophys. Space Sci. 14 (1971) 151–167.
- [12] C.Z. Cheng, G. Knorr, Integration of Vlasov equation in configuration space, J. Comput. Phys. 22 (1976) 330–351.
- [13] R.R.J. Gagne, M.M. Shoucri, Splitting scheme for numerical-solution of a one-dimensional Vlasov equation, J. Comput. Phys. 24 (1977) 445–449.
- [14] M. Shoucri, H. Gerhauser, K.-H. Finken, Study of the generation of a charge separation and electric field at a plasma edge using Eulerian Vlasov codes in cylindrical geometry, Comput. Phys. Comm. 164 (2004) 138–149.
- [15] H.L. Stone, Iterative solution of implicit approximations of multidimensional partial differential equation, SIAM J. Numer. Anal. 5 (1968) 530–558.
- [16] G.E. Schneider, M. Zedan, A modified strongly implicit procedure for the numerical solution of field problems, Numer. Heat Trans. 4 (1981) 1–19.
- [17] H.-J. Leister, M. Perić, Vectorized strongly implicit solving procedure for a seven-diagonal coefficient matrix, Int. J. Numer. Methods Heat Fluid Flow 4 (1994) 159–172.
- [18] C.A.J. Fletcher, Computational Techniques for Fluid Dynamics 1, Springer-Verlag, Berlin, 1991.
- [19] M. Miyamoto, R. Nagai, 3-Dimensional models for distribution of mass in galaxies, PASJ 27 (1975) 533–543.
- [20] M.N. Rosenbluth, W.M. MacDonald, D.L. Judd, Fokker–Planck equation for an inverse-square force, Phys. Rev. 107 (1957) 1–6.
- [21] C. Einsel, R. Spurzem, Dynamical evolution of rotating stellar systems – I. Pre-collapse, equal-mass system, MNRAS 302 (1999) 81–95.
- [22] R.K. Galloway, A.L. MacKinnon, E.P. Kontar, P. Helander, Fast electron slowing-down and diffusion in a high temperature coronal X-ray source, A&A 438 (2005) 1107–1114.
- [23] K. Nanbu, Probability theory of electron–molecule, ion–molecule, molecule–molecule, and Coulomb collisions for particle modeling of materials processing plasmas and gases, IEEE Trans. Plasma Sci. (2000) 971–990.
- [24] L.S. Brown, D.L. Preston, R.L. Singleton, Charged particle motion in a highly ionized plasma, Phys. Rep. 410 (2005) 237–333.
- [25] M. Ujevic, P.S. Letelier, Numerical self-consistent stellar models of thin disks, A&A 442 (2005) 785–793.
- [26] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, J. Comput. Phys. 7 (1986) 856–869.
- [27] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Stand. 45 (1950) 255–282.

- [28] W. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, *Quart. Appl. Math.* 9 (1951) 17–29.
- [29] C. Lanczos, Solution of systems of linear equations by minimized iteration, *J. Res. Bur. Stand.* 49 (1952) 33–53.
- [30] V. Frayss, L. Giraud, S. Gratton, J. Langou, A set of GMRES routines for real and complex arithmetics on high performance computers, CERFACS Technical Report TR/PA/03/3, public domain software available on <<http://www.cerfacs.fr/algor/Softs>>, 2003.
- [31] See for example: G.E. Forsythe, W.R. Wasow, *Finite Difference Methods for Partial Differential Equations*, John Wiley, 1960.