*Bull. Nov. Comp. Center*, , (2013), 1–10


# A semi-Lagrangian scheme for convection equations sing the finite element method

V.I. Kuzin, A.Yu. Chursina, V.V. Kravtchenko

**Abstract.** This paper considers a semi-Lagrangian scheme (SLS) for solving convection equations. The transport equations, as written in the Lagrangian form at each time step, are approximated on the basis of a weak form using a finite element method representation with various coordinate functions: delta functions, piecewise-linear functions, and various interpolation methods: those based on piecewise-linear functions and third order B-splines. For the two-dimensional case, the calculation of two-dimensional B-spline reduces to solving a set of the algebraic equations of the dimension less than initial system allows one to reduce the calculation time. For the schemes considered, test calculations have been conducted, the results for the one- and two-dimensional cases are presented.

## Introduction

The semi-Lagrangian methods [1, 2] as applied to the advection equation for a certain substance are algorithms with allowance for the fact that the transport equation is a total time derivative in the direction of a flow. If such a substance is preserved, it does not change along trajectories of the flow. When calculating the advection transport with the semi-Lagrangian methods, a discrete set of particles, which results in a regular set of the nodal points, is traced backward to the departure points in the time interval. The value of the function at these points is obtained with the help of certain interpolation procedures. This method gives us numerical schemes that are unconditionally stable for linear problems and have inessential phase errors for the waves, whose length exceeds two grid intervals [3, 4]. In [5], it was shown that in the projective statement, the semi-Lagrangian method of particles-in-cell can be reduced to the finite element method for different coordinate functions and interpolation. This paper is aimed at the comparison of two methods of interpolation of the original points and two versions of representing a solution. To this end, we will use the linear interpolation and interpolation with B-splines combined with presenting a desired solution as a linear combination of the Dirac delta-functions or piecewise linear polynomials. This paper reveals that the interpolation with B-splines in combination with a piecewise linear presentation is the most accurate method for the problem in question. For the two-dimensional case, the calculation of two-dimensional B-spline reduces to the solution of a set of $I$ linear algebraic equations of $J \times J$ dimension and a set of $J$ linear algebraic equations of $I \times I$ dimension, which require less computer costs than for solving the initial system. This allows us to essentially reduce the calculation time.

## 1. Description of the method

Let us consider an open rectangle $\Omega = (a, b) \times (c, d)$. Without loss of generality let us consider it to be a part of a larger domain as to avoid consideration of boundary conditions and to solve in it the advection equation with the velocity $\vec{u}$ and the scalar function $\varphi$

$$\begin{gathered} \frac{\partial \varphi}{\partial t} + \vec{u} \cdot \nabla \varphi = 0, \quad \vec{x} \in \Omega, \quad t \geq 0, \\ \varphi(\vec{x}, 0) = \varphi_0(\vec{x}). \end{gathered} \tag{1}$$

The velocity vector $\vec{u}(\vec{x}, t)$ is assumed to be non-divergent, continuous and with a limited first derivative. In order to solve system (1), consider the characteristic trajectories $\vec{X}(\vec{x}, s; t)$ for (1), satisfying the equation

$$\frac{d\vec{X}(\vec{x}, s; t)}{dt} = \vec{u}(\vec{X}(\vec{x}, s; t), t), \qquad \vec{X}(\vec{x}, s; s) = \vec{x}, \tag{2}$$

where $\vec{X}(\vec{x}, s; t)$ denotes the original position at the instant $t$ of a liquid particle being at the point $\vec{x}$ at the instant $s$. The equation (2) can be replaced by

$$\vec{X}(\vec{x}, s; t) - \vec{x} = \int_s^t \vec{u}(\vec{X}(\vec{x}, s; \tau), \tau)\, d\tau.$$

Solution to (1) is as follows:

$$\varphi(\vec{x}, s) = \varphi\big(\vec{X}(\vec{x}, s; 0), 0\big).$$

Then for any two subsequent temporal levels $t$ and $s = t + \tau$, the following will be valid:

$$\varphi(\vec{x}, s) = \varphi\big(\vec{X}(\vec{x}, s; t), t\big). \tag{3}$$

In the projection form, formula (3) corresponds to a weak solution, given by the integral relation

$$\begin{aligned} \int_\Omega \varphi(\vec{x}, s)\Psi(\vec{x})\, d\vec{x} &= \int_\Omega \varphi\big(\vec{X}(\vec{x}, s; t), t\big)\Psi(\vec{x})\, d\vec{x} \\ &= \int_\Omega \varphi(\vec{y}, t)\Psi(\vec{X}(\vec{y}, t; s))\, d\vec{y}, \end{aligned} \tag{4}$$

since the main linear part of the Jacobian determinant $\left|\frac{d\vec{X}(\vec{x}, s; t)}{d\vec{x}}\right|$ is equal to 1 because $\nabla \cdot \vec{u} = 0$. Here $\vec{y} = \vec{X}(\vec{x}, s; t)$ and $\Psi(\vec{x})$ is a test function.

To construct the finite element approximations of relation (4), let us introduce into $\Omega$ a regular grid with a mesh size $h$. For solving the problem, let us represent the solution $\varphi$ as a linear combination of some coordinate functions. For such functions either the Dirac delta-functions, centered at the nodal points of the domain with number $k$, or the functions $\Psi_k$ (piecewise linear basis functions of a finite element space) will be selected. As a test function, the functions $\Psi_k$ will be chosen. According to the particle-in-cell method, we place one particle into each grid node, trace its position backwards during the time interval $\tau$ along the characteristic line and assume that a substance is transferred by a particle, so that the value $\varphi$ at the nodal point $\vec{x}_k$ at the level $s$ coincides with the value $\varphi$ at the departure points $\vec{X}(\vec{x}_k, s; t)$. In this case, the value of the function at these points is to be found by two different versions of interpolation, i.e., by linear and by B-splines of third order.

## 2. The concept of solution by delta-functions with linear interpolation

For a vivid presentation let us consider a one-dimensional case. The solution $\varphi$ at the level $s = t + \tau$ will be presented as a linear combination of the Dirac delta-functions centered at the nodal point $x_k$, $k = 1, \ldots, I$,

$$\varphi(x, s) = \sum_{k=1}^{I} \varphi(x_k, s)\delta(x - x_k). \tag{5}$$

The function $\varphi$ at the time level $t$ is presented in a similar way:

$$\varphi\big(X(x, s; t), t\big) = \sum_{k=1}^{I} \varphi\big(X(x_k, s; t), t\big)\ \delta\big(x - X(x_k, s; t)\big).$$

Here $\varphi(x_k, s)$ is the value of the function at the nodal point at the time level $s$, $\varphi(X(x_k, s; t), t)$ is the value of the function at the departure point at a previous time level. Consider the projective form of the problem. As test functions in (4), let us consider the piecewise-linear polynomials of the form $\Psi_i(x)$

$$\Psi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h}, & x \in [x_{i-1}, x_i], \\ \dfrac{x_{i+1} - x}{h}, & x \in [x_i, x_{i+1}], \\ 0, & \text{otherwise.} \end{cases}$$

As a result we come to

$$\sum_{k=1}^{I} \varphi(x_k, s) \int_\Omega \delta(x - x_k)\ \Psi_l(x)\ dx$$
$$= \sum_{k=1}^{I} \varphi\big(X(x_k, s; t), t\big) \int_\Omega \delta(x - X(x_k, s; t))\ \Psi_l(x)\ dx, \quad l = 1, \dots, I.$$

With allowance for properties of delta-functions we obtain

$$\int_\Omega \delta(x - x_k)\ \Psi_l(x)\ dx = \Psi_l(x_k),$$
$$\int_\Omega \delta(x - X(x_k, s; t))\ \Psi_l(x)\ dx = \Psi_l\big(X(x_k, s; t)\big).$$

In addition, from the definition of the function $\Psi_k$ it follows that $\Psi_l(x_k) = 1$, $k = l$, and $\Psi_l(x_k) = 0$, $k \neq l$. For approximation of the functions $\varphi(X(x_k, s; t), t)$, we will use the linear interpolation

$$\varphi\big(X(x_k, s; t), t\big) \approx \varphi(x_{k-1}, t) + \big(\varphi(x_k, t) - \varphi(x_{k-1}, t)\big) \frac{X(x_k, s; t) - x_{k-1}}{h}.$$

The conducted transformations result in a system of equations, written down in the matrix form $\{\varphi(s)\} = \{b\}$, where $\varphi(s)$ is a vector column of unknowns of $I$-dimension, and entries of the right-hand side are given by the relation

$$b_l = \sum_{k=1}^{I} \Psi_l\big(X(x_k, s; t)\big) \times$$
$$\Big(\varphi(x_{k-1}, t) + (\varphi(x_k, t) - \varphi\big(x_{k-1}, t)\big) \frac{X(x_k, s; t) - x_{k-1}}{h}\Big).$$

As for the left-hand side, there is only a vector-column of unknowns, the system is resolved explicitly.

## 3. A piecewise linear presentation. Interpolation with cubic splines

In this section, for the interpolation of values at departure points through values at nodal points, we will not use a linear interpolation as in the previous section, but interpolation with B-splines in one-dimensional case. The function $\varphi$ will be presented as a linear combination of piecewise linear basis functions $\Psi_i$ of a finite element space, centered at the nodal points $x_i$. A natural extension to a two-dimensional case will be given in Section 4. This brings about the presentation

$$\varphi(x,s) = \sum_{k=1}^{I} \varphi(x_k,s)\Psi_k(x). \tag{6}$$

The corresponding representation for the function $\varphi$ at the previous level yields

$$\varphi(y,t) = \sum_{k=1}^{I} \varphi(x_k,t)\Psi_k(y). \tag{7}$$

Substituting formulas (6) and (7) into relation (4) for two temporal layers, obtain

$$\sum_{k=1}^{I} \varphi(x_k,s) \int_\Omega \Psi_k(x)\Psi_l(x)\,dx = \sum_{k=1}^{I} \varphi(x_k,t) \int_\Omega \Psi_k(y)\Psi_l(X(y,t;s))\,dy$$
$$= \sum_{k=1}^{I} \varphi(x_k,t) \int_\Omega \Psi_k(y)\Psi_l(y-(X(x_l,s;t)-x_l))\,dy, \quad l=1,\dots,I, \tag{8}$$

resulting in the system of linear algebraic equations of the form

$$A\{\varphi(s)\} = \{b\}. \tag{9}$$

Here $A$ is a symmetric positive-definite matrix of masses with the components $a_{kl}$ given by the equalities

$$a_{kl} = \int_\Omega \Psi_k(x)\,\Psi_l(x)\,dx.$$

Elements of the vector-column $b$ are calculated as

$$b_l = \sum_{k=1}^{I} \varphi(x_k,t) \int_\Omega \Psi_k(x)\Psi_l(x-(X(x_l,s;t)-x_l))\,dx. \tag{10}$$

To transform formulas in the right-hand side of the matrix equation or, which is the same, of the components $b_l$, consider the function $K_k(X_l)$, $X_l = X(x_l,s;t)$, presented as

$$K_k(X_l) = \int_\Omega \Psi_k(x)\Psi_l(x-(X_l-x_l))\,dx, \tag{11}$$

where

$$\Psi_l(x-(X_l-x_l)) = \begin{cases} -\dfrac{X_l-(x+h)}{h}, & x\in[X_l-h,X_l], \\ \dfrac{X_l-(x-h)}{h}, & x\in[X_l,X_l+h], \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Substituting formula (12) into (11), we obtain the functions $K_k(X_l)$, possessing the following properties [1]:

1. $K_k(X_l)$ is the cubic polynomial of $X_l$.
2. $\frac{dK_k(X_l)}{dX_l}$ is continuous.
3. $\frac{d^2K_k(X_l)}{dX_l^2}$ is a function that is continuous in the domain.

Conditions 1–3 define $K_k(X_l)$ as a cubic spline with inner points at the nodal points of the grid under consideration. So, the components $b_l$ are linear combinations of cubic splines, i.e.,

$$b_l = \sum_{k=1}^{I} \varphi(x_k, t) K_k(X_l). \tag{13}$$

Since a linear combination of cubic splines is also a cubic spline, then from (13) it follows that the components $b_l$ are interpolation values at the points $X_l$ of the new bicubic spline $S(X_l)$. To calculate such a spline, we set its values at the nodal grid points. Denote by $\upsilon_l$ the values of the spline $S(x)$ at the nodal points. Further, assuming $X_l = x_l$, from (10) obtain

$$A\big\{\varphi(t)\big\} = \{\upsilon\}.$$

Since cubic B-splines form the local basis of a linear space of cubic splines, then, according to [6], we can obtain

$$S(x) = \sum_{i=1}^{I} c_i B_{4,i}(x),$$

where $B_{4,i}(x)$ denotes the $i$-th cubic B-spline. The factors $c_i$ depend on time, therefore in order to calculate them, we take the nodal points and arrive at:

$$\big\{S(x_l)\big\} = \{\upsilon\} = A\{\varphi(t)\}. \tag{14}$$

For finding the factors $c_i$ at the level $t$, it is required to solve system

$$B\{c\} = A\big\{\varphi(t)\big\}. \tag{15}$$

Here the matrix $B = \{b_{ij}\} = \{B_{4,j}(x_i)\}$ is positive definite and has a symmetric tridiagonal structure at inner nodes of the domain.

As a result, from (9), (13) and (14) we obtain

$$A\{\varphi(s)\} = \left\{ \sum_{i=1}^{I} c_i B_{4,i}(X_l) \right\} = \{b\}. \tag{16}$$

The value $\varphi(s)$ at any time instant can be found by the following algorithm:

1. Calculate $c_i$ from (15).
2. Find $B_{4,i}(X_l)$.
3. Construct $b_l = \sum_i c_i B_{4,i}(X_l)$.
4. Solve system (9).

## 4. A two-dimensional case

In this section, let us consider the problem in the 2D case with a piecewise coordinate function and spline interpolation. Based on the previous arguments, we come to the equation of the form of (9)

$$A\{\varphi(s)\} = \{b\}, \tag{17}$$

where matrix $A$ represents a tensor product of the tridiagonal matrices $A_x$ and $A_y$ having the structure of the above-considered matrix $A$ and acting in the directions $x$ and $y$, respectively. Thus, we come to the system of equations

$$(A_x \otimes A_y)\{\varphi(s)\} = \left\{ \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} B_{4,j}(X_p) B_{4,j}(Y_q) \right\} = \{b\}. \tag{18}$$

Here $b = \{b_{pq}\}$ is a vector column of the right-hand side. Let us rewrite it in the vector-matrix representation

$$(A_x \otimes A_y)\{\varphi(s)\} = (D_x \otimes D_y)\{c\}, \tag{19}$$

where $D_x = \{B_{4,i}(X_p)\}$, $D_y = \{B_{4,j}(Y_q)\}$ are square $I \times I$ and $J \times J$ matrices, and $c = \{c_{ij}\}$ is a vector column of $IJ$ dimension.

The vector $\{c\}$ is the solution to a system of equations similar to (15):

$$(B_x \otimes B_y)\{c\} = (A_x \otimes A_y)\{\varphi(t)\}. \tag{20}$$

Thus, having found the factors $c_{ij}$ from system (20), we can find the solution from (19). It should be noted that the matrices $A_x \otimes A_y$, $B_x \otimes B_y$ are nine-diagonal $IJ \times IJ$ matrices, therefore computer costs of their conversion essentially increase.

An alternative approach proposed here is in representing a tensor product through a usual product of matrices. Let us explain it, as an example, on equation (19). The left-hand side of the equation is represented as

$$(A_x \otimes A_y)\{\varphi(s)\} \approx A_x(A_y \Phi^T)^T, \tag{21}$$

where $\Phi$ is a matrix, whose $j$-th column is a solution at the points $x_i$ ($i = 1, \ldots, I$) at the level $y_j$ ($j = 1, \ldots, J$).

Thus, the original problem reduces to a linear algebraic problem with a matrix, decomposing to the product of matrices. Denoting the matrix $(A_y\Phi^T)^T$ through $\Psi$, we have arrived at solving $I$ systems of linear algebraic equations with $J \times J$ tridiagonal matrices and $J$ systems of linear algebraic equations with $I \times I$ tridiagonal matrices.

In the same way it is possible to reduce problem (20) to solving a set of the algebraic equations of the dimension less than initial system.

## 5. A test example

As a test example, consider a "step" function

$$\varphi(x) = \begin{cases} 1, & 0 \le x \le 0.25, \\ 0, & \text{otherwise}, \end{cases}$$

which moves with a velocity $u = 0.15$. The calculation results at the instant $t = 0.5$ are displayed in Figures 1, 2.

It is clear that the linear interpolation, as was expected, strongly smoothes the solution. Therefore the semi-Lagrangian method employing the finite element method with interpolation by B-splines can be considered to be the best version of the solution. At the same time, from Figure 2 it is also clear that the interpolation with B-splines will give weak "splashes" of the solution in the areas with a jump. Test calculations with a 2D "step" function moving on the plane have the same results as in the 1D case (Figure 3). These results also indicate to the fact that representation (21) essentially decreases the volume of calculations.

*Acknowledgements.* The authors would like to express their gratitude to Dr. A.I. Rozhenko for his constructive suggestions.

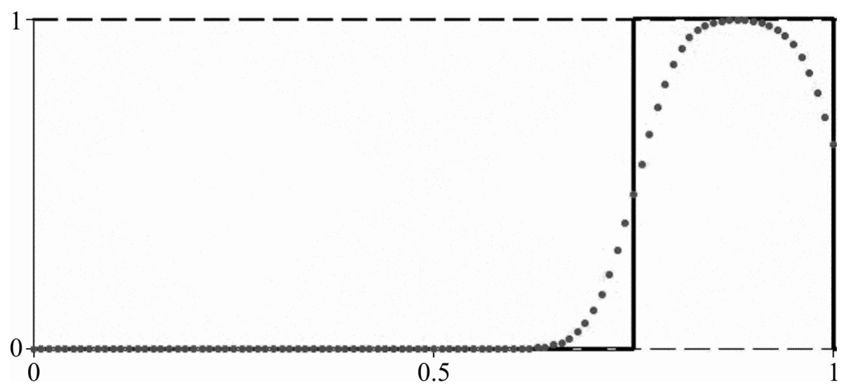


**Figure 1.** Representation of the numerical solution with linear interpolation

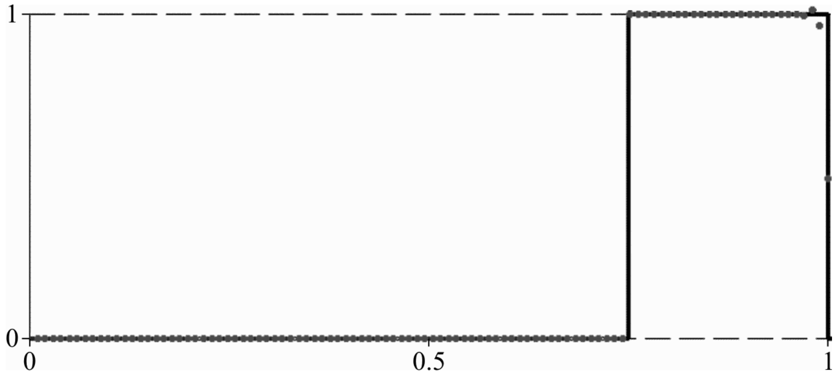


**Figure 2.** Representation of the numerical solution with interpolation by cubic B-splines

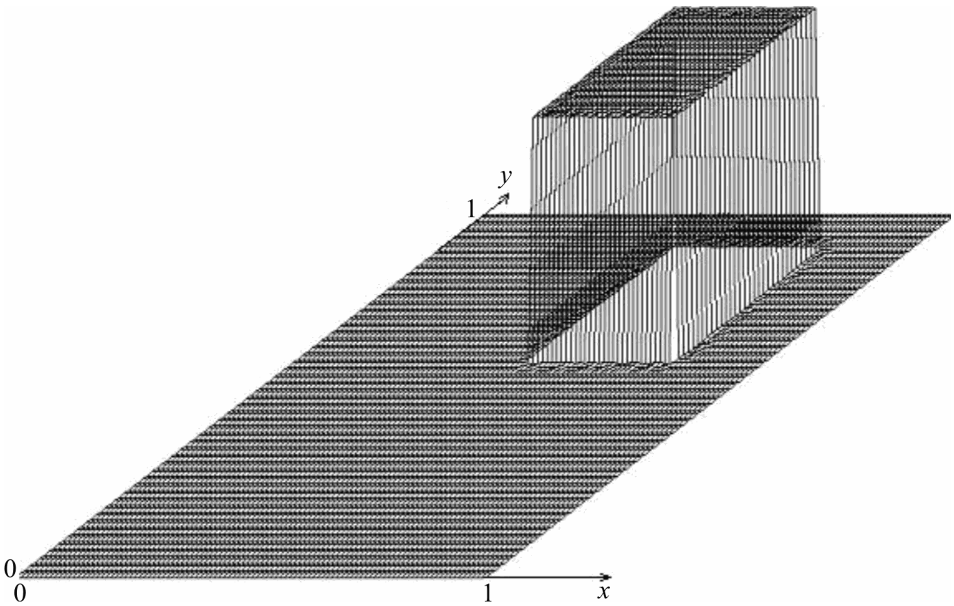


**Figure 3.** The results of calculations with interpolation by bicubic B-splines

## References

[1] Bermejo R. On the equivalence of semi-Lagrangian schemes and particle-in-cell finite element methods // Month Weather Rev. — 1990. — Vol. 118, No. 4. — P. 979–987.

[2] McDonald A. Accuracy of multiply-upstream, advective schemes // Month Weather Rev. — 1984. — Vol. 112. — P. 1267–1275.

[3] Pudykiewcz J., Staniforth A. Some properties and comparative performance of the semi-Lagrangian method of Robert in the solution of the advection-diffusion equation // Atmosphere-Ocean. — 1984. — Vol. 22. — P. 283–308.

[4] Purnell D.K. Solution of the advective equation by upstream interpolation with cubic splines // Month Weather Rev. — 1976. — Vol. 104. — P. 42–48.

[5] Robert A.A. A stable numerical integration scheme for the primitive meteorological equations // Atmosphere-Ocean. — 1981. — Vol. 19. — P. 35–46.

[6] Alberg J., Nilson E., Walsh J. Theory of Splines and its Applications. — Moscow: Mir, 1972.