# Conservative Semi-Lagrangian Numerical Algorithm with Decomposition of Integration Domain into Tetrahedrons for Three-Dimensional Advection Problem

Alexander Vyatkin[1] and Elena Kuchunova[2]

[1] Institute of Computational Modelling of SB RAS,
Akademgorodok, 660036, Krasnoyarsk, Russia,
`vyatkin@icm.krasn.ru`,
[2] Siberian Federal University, 79 Svobodny pr., 660041, Krasnoyarsk, Russia,
`hkuchunova@sfu-kras.ru`

**Abstract.** A conservative semi-Lagrangian method is developed in order to solve three-dimensional linear advection equation. It based on balance equation in integral form. Main feature of proposed method consists in way of computation of integral at lower time level. To compute integral, we decompose a domain of integration into several tetrahedrons and approximate integrand by trilinear function.

**Keywords:** Semi-Lagrangian method, advection equation, decomposition of integration domain, local conservation low.

## 1 Introduction

Semi-Lagrangian methods was developed from idea of applying a characteristic trajectories in problems for weather prediction [1]. Elaboration of this approach led to a new wide family of semi-Lagrangian methods for solving advection dominated problems [2–7]. Semi-Lagrangian algorithms do not involve the time step restriction [8] which traditionally ensues from the Courant-Friedrichs-Lewy condition for Eulerian schemes. Most of the semi-Lagrangian methods [9–11] are based on the integral equality for unknown function between two neighboring time levels. This integral equality express conservation low. In general case algorithm consists of three main steps: calculation of the integral of an approximate solution at the upper time level; computation of trajectories backward in time to the lower time level; calculation of the integral at the lower time level. There are many ways to implement each step. The improvement of properties of a numerical method usually leads to computational complications. This is especially valid for three-dimensional problems [12, 13]. Despite of that there is a wide variety of semi-Lagrangian algorithms, only some of them can be conveniently employed for three-dimensional problem.

We present the semi-Lagrangian algorithm for three-dimensional problem. It is based on the conservation low. The focus of described investigation is on

constructing conservative algorithm without weight coefficients. Another aim of research is to create method algorithmically accessible for three dimensional problem. For with purpose we make several simplifications. Firstly, we use only uniform cubic space grid. Algorithm can be generalized into space grid with different grid steps only for different space axes. So, we can use only one fixed grid step for one space direction. Therefore, space grid should be uniform. Secondly, we construct method of first order of convergence. There are many semi-Lagrangian methods of convergence order higher than one. All these methods use weight coefficients to make valid the conservation low for numerical solution. Such approach distorts numerical solution. To avoid this weakness, we construct method of first order of convergence. Thirdly, we use time step which is twice smaller than time step corresponding to Courant-Friedrichs-Lewy condition for Eulerian schemes. This restriction reduces the algorithm of integration at lower time level substantially. The algorithm became more convenient for three-dimensional problem. We find numerical solution of problem as a grid function. To make algorithm explicit, we set domain of integration at the upper time level as cube neighborhood of a grid node and approximate this integral by the midpoint cubature rule. Domain of integration at the lower time level is defined by trajectories which are issued out from the domain of integration at the upper time level. To compute integral, we approximate integrand by trilinear function. Due to restriction for time step, domain of integration at the lower time level is a curved cuboid. To approximate it, we compute coordinates of its vertices by one step of Euler method. We decompose this cuboid by planes into several parts. In general case the number of parts is eight. Each part we decompose into several tetrahedrons. To compute integral over a tetrahedron, we use Gauss quadrature rule which is accurate for trilinear function. Different numerical experiments confirm the first-order convergence and validity of conservation low.

## 2    The Formulation of the Problem

We use computational region $D = [0,1] \times [0,1] \times [0,1]$ and its boundary $\partial D$. In the cylinder $[0,T] \times D$ we consider three-dimensional advection equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial(u\rho)}{\partial x} + \frac{\partial(v\rho)}{\partial y} + \frac{\partial(w\rho)}{\partial z} = f. \tag{1}$$

We suppose that components $u(t,\mathbf{x})$, $v(t,\mathbf{x})$, $w(t,\mathbf{x})$, $\mathbf{x} = (x,y,z)$, of the velocity vector $\mathbf{u} = (u,v,w)$ and function $f(t,\mathbf{x})$ are known and sufficiently smooth in $[0,T] \times D$. Function $\rho(t,\mathbf{x})$ is unknown. We denote the inflow boundary by $\Gamma_{\text{in}} = \partial D\big|_{x=0}$, the outflow boundary by $\Gamma_{\text{out}} = \partial D\big|_{x=1}$, and the rigid boundary by $\Gamma_{\text{rig}} = \partial D \setminus (\Gamma_{\text{in}} \cup \Gamma_{\text{out}})$. For the velocity functions we use the conditions

$$\mathbf{u}(t,\mathbf{x})\big|_{\Gamma_{\text{rig}}} = \mathbf{0}, \quad u(t,\mathbf{x})\big|_{\Gamma_{\text{in}}} \geq 0, \quad u(t,\mathbf{x})\big|_{\Gamma_{\text{out}}} \geq 0 \qquad \forall\, t \in [0,T]. \tag{2}$$

Also we set the following boundary and initial conditions for function $\rho(t,\mathbf{x})$:

$$\rho(t,\mathbf{x})\big|_{\Gamma_{\text{in}}} = \rho_{\text{in}}(t,y,z) \quad \forall\, t \in [0,T],\ y \in [0,1],\ z \in [0,1], \tag{3}$$

$$\rho(0,\mathbf{x}) = \rho_{\text{init}}(\mathbf{x}) \quad \forall\, \mathbf{x} \in D. \tag{4}$$

Functions $\rho_{\text{in}}(t, y, z)$ and $\rho_{\text{init}}(\mathbf{x})$ are known and sufficiently smooth.

To solve problem (1)–(4), we construct grid $D^h$ with mesh-size $h = 1/N$

$$D^h = \{(x_i, y_j, z_k) : x_i = ih, y_j = jh, z_k = kh; \ i, j, k = 0, 1, \ldots, N\}.$$

We denote $\mathbf{x}_{i,j,k} = (x_i, y_j, z_k), i, j, k = 0, \ldots, N$, and define neighborhood

$$\Omega_{i,j,k} = ([x_i - h/2, x_i + h/2] \times [y_j - h/2, y_j + h/2] \times [z_k - h/2, z_k + h/2]) \cap D$$

and grid cell

$$D_{i,j,k} = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}].$$

We introduce time grid $T^\tau = \{t_m : t_m = m\tau, m = 0, \ldots, M\}$ with time step $\tau = T/M$. We find solution of problem (1)–(4) as grid function $\rho^h$ defined on $T^h \times D^h$.

## 3   The Numerical Algorithm

We introduce notations $g_{i,j,k}^m = g(t_m, \mathbf{x}_{i,j,k})$ and $g_{i,j,k}^{h,m} = g^h(t_m, \mathbf{x}_{i,j,k})$ for a function $g$ and a grid function $g^h$ defined in $T \times D$ and $T^\tau \times D^h$, respectively. We suppose that $\rho^h$ is already defined at time level $t_{m-1}$ and we need to compute $\rho^h$ at next time level $t_m$. To construct numerical algorithm, we suppose that

$$\tau \leq \frac{h}{2 \max |\mathbf{u}(t, \mathbf{x})|}, \qquad (5)$$

where $|\mathbf{u}(t, \mathbf{x})| = \sqrt{u^2(t, \mathbf{x}) + v^2(t, \mathbf{x}) + w^2(t, \mathbf{x})}$.

### 3.1   The Local Integral Equality

To calculate $\rho_{i,j,k}^{h,m}$, $i = 1, ..., N; j, k = 0, ..., N$, we consider $\Omega_{i,j,k}$ and its boundary $\partial \Omega_{i,j,k}$. For a point $\boldsymbol{B} = (B_x, B_y, B_z) \in \partial \Omega_{i,j,k}$ at the time level $t_m$ we construct trajectory to the previous time level $t_{m-1}$ backward in time. This trajectory is solution of Cauchy problem for the system of ordinary differential equations

$$\hat{x}'(t) = u(t, \hat{x}, \hat{y}, \hat{z}), \ \hat{y}'(t) = v(t, \hat{x}, \hat{y}, \hat{z}), \ \hat{z}'(t) = w(t, \hat{x}, \hat{y}, \hat{z}) \qquad (6)$$

with the initial condition

$$\hat{x}(t_m) = B_x, \quad \hat{y}(t_m) = B_y, \quad \hat{z}(t_m) = B_z \qquad (7)$$

and $t \in [t_{m-1}, t_m]$. Denote solution of problem (6)–(7) by $\bar{\mathbf{x}}(t; \boldsymbol{B})$. Set of all points $\boldsymbol{B} \in \partial \Omega_{i,j,k}$ forms boundary $\partial \mathrm{V}_{i,j,k}^m(t) = \{\bar{\mathbf{x}}(t; \boldsymbol{B}) : \boldsymbol{B} \in \partial \Omega_{i,j,k}\}$ of a region $\mathrm{V}_{i,j,k}^m(t)$. Thus, $V_{i,j,k}^m(t_m) = \Omega_{i,j,k}$. The following statement is available.

**Statement.** For each $i = 1, ..., N; j, k = 0, ..., N$ is valid

$$\int_{\Omega_{i,j,k}} \rho(t_m, \mathbf{x}) \, d\Omega = \int_{V_{i,j,k}^m(t_{m-1})} \rho(t_{m-1}, \mathbf{x}) \, dV + \int_{t_{m-1}}^{t_m} \int_{V_{i,j,k}^m(t)} f(t, \mathbf{x}) \, dV \, dt. \quad (8)$$

To construct numerical method, we approximate each term of this equality.

### 3.2    The Integration at the Upper Time Level

We compute integral over $\Omega_{i,j,k}$ in (8) approximately in the following way:

$$\int\limits_{\Omega_{i,j,k}} \rho(t_m,\mathbf{x})\,d\Omega \approx \rho_{i,j,k}^{h,m}\,\mathrm{meas}\left(\Omega_{i,j,k}\right), \tag{9}$$

where $\mathrm{meas}\left(\Omega_{i,j,k}\right)$ is volume of $\Omega_{i,j,k}$. To approximate the integral for integrand $f(t,\mathbf{x})$ in (8), we employ the rectangle rule in $t$-direction. After that, we get

$$\int\limits_{t_{m-1}}^{t_m} \int\limits_{V_{i,j,k}^m(t)} f\left(t,\mathbf{x}\right)\,dV\,dt \approx \tau \int\limits_{V_{i,j,k}^m(t_m)} f\left(t_m,\mathbf{x}\right)\,dV \approx \tau f_{i,j,k}^m\,\mathrm{meas}\left(\Omega_{i,j,k}\right). \tag{10}$$

### 3.3    The Integration at the Lower Time Level

To compute integral over $V_{i,j,k}^m(t_{m-1})$ in (8), we consider eight vertices $\boldsymbol{A}_n = (A_{n,x}, A_{n,y}, A_{n,z})$ of $\Omega_{i,j,k}$. We consistently use $\boldsymbol{A}_n$ as initial condition for system (6). Thus we get eight Cauchy problems. To find numerical solution of Cauchy problem, we use Euler method. We denote by $\boldsymbol{W}_n^h = \left(W_{n,x}^h, W_{n,y}^h, W_{n,z}^h\right)$ value of numerical solution at the time level $t_{m-1}$. Therefore

$$W_{n,x}^h = A_{n,x}^h - \tau\,u\left(t_m, \boldsymbol{A}_n\right),\ W_{n,y}^h = A_{n,y}^h - \tau\,v\left(t_m, \boldsymbol{A}_n\right),$$
$$W_{n,z}^h = A_{n,z}^h - \tau\,w\left(t_m, \boldsymbol{A}_n\right).$$

Nodes $W_n^h, n = 1,\ldots,8$, form an octagon $V_{i,j,k}^{h,m}(t_{m-1})$. We decompose $V_{i,j,k}^{h,m}(t_{m-1})$ by planes $x = x_i$, $y = y_j$, and $z = z_k$ into several octagons $R_n$. If $\mathbf{x}_{i,j,k} \notin \partial D$, then due to (5) the number of octagons is eight If $\mathbf{x}_{i,j,k} \in \partial D$, then the number of octagons is four or two. Here and after we consider case with eight octagons (see Fig. 1). For other cases the reasoning is similar. Due to (5), any $R_n$ lies in
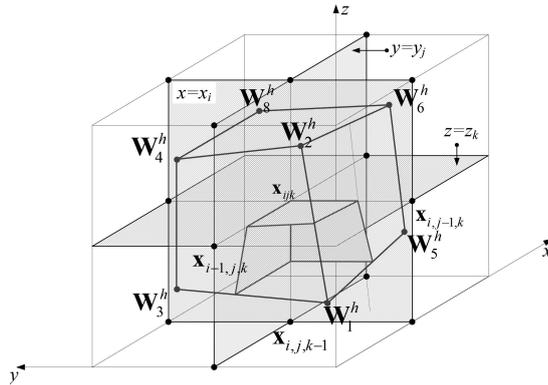


**Fig. 1.** Decomposition of octagon $V_{i,j,k}^{h,m}(t_{m-1})$

one corresponding cell $D_{p,q,r}$. We define trilinear function $\rho_I^h(t_{m-1}, \mathbf{x})$ on $D_{p,q,r}$

$$\rho_I^h(t_{m-1}, \mathbf{x}) = \sum_{a=p}^{p+1} \sum_{b=q}^{q+1} \sum_{c=r}^{r+1} \varphi_a(x)\psi_b(y)\chi_c(z)\rho_{a,b,c}^{h,m-1}, \tag{11}$$

where

$$\varphi_a(x) = \begin{cases} \frac{(x-x_{a-1})}{h}, & \text{if } x \in [x_{a-1}, x_a), \\ \frac{(x_{a+1}-x)}{h}, & \text{if } x \in [x_a, x_{a+1}], \\ 0, & \text{otherwise,} \end{cases} \quad \psi_b(y) = \begin{cases} \frac{(y-y_{b-1})}{h}, & \text{if } y \in [y_{b-1}, y_b), \\ \frac{(y_{b+1}-y)}{h}, & \text{if } y \in [y_b, y_{b+1}], \\ 0, & \text{otherwise,} \end{cases}$$

$$\chi_c(z) = \begin{cases} \frac{(z-z_{c-1})}{h}, & \text{if } z \in [z_{c-1}, z_c), \\ \frac{(z_{c+1}-z)}{h}, & \text{if } z \in [z_c, z_{c+1}], \\ 0, & \text{otherwise.} \end{cases}$$

Thus we construct the following approximation:

$$\int_{V_{i,j,k}^m(t_{m-1})} \rho(t_{m-1}, \mathbf{x})\, dV \approx \sum_{n=1}^{8} \int_{R_n} \rho_I^h(t_{m-1}, \mathbf{x})\, dR. \tag{12}$$

To compute integral over $R_n$, we unscramble $R_n$ into two prisms. Each prism we decompose into three tetrahedrons (see Fig. 2). Therefore
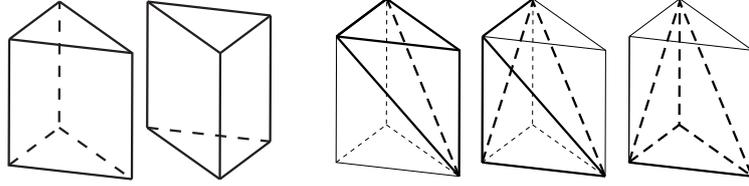


**Fig. 2.** Decomposition of $R_n$ into six tetrahedrons

$$\int_{R_n} \rho_I^h(t_{m-1}, \mathbf{x})\, dR = \sum_{s=1}^{6} \int_{R_n^s} \rho_I^h(t_{m-1}, \mathbf{x})\, dR. \tag{13}$$

We denote vertices of a tetrahedron $R_n^s$ by $P_l = (P_l^x, P_l^y, P_l^z,)$, $l = 0, 1, 2, 3$, and vectors $\mathbf{e}_l = P_l - P_0$, $l = 1, 2, 3$. We consider unit tetrahedron $E$ with vertices in points $(0,0,0), (1,0,0), (0,1,0), (0,0,1)$. To compute integral over $R_n^s$, we change variables $x, y, z$ to $\xi, \eta, \zeta$ in the following way:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} P_0^x \\ P_0^y \\ P_0^z \end{pmatrix} + \begin{pmatrix} e_1^x & e_2^x & e_3^x \\ e_1^y & e_2^y & e_3^y \\ e_1^z & e_2^z & e_3^z \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}. \tag{14}$$

This formula signifies transformation of the unit tetrahedron $E$ to tetrahedron $R_n^s$. Therefore we get

$$\int\limits_{R_n^s} \rho_I^h \left(t_{m-1}, \mathbf{x}\right) dR = \int\limits_{E} \rho_I^h \left(t_{m-1}, \mathbf{x}\left(\xi, \eta, \zeta\right)\right) J \, dE, \tag{15}$$

where $J$ is Jacobian of transformation. To compute the integral over $E$, we use the following Gauss quadrature rule

$$\int\limits_{E} g\left(\xi, \eta, \zeta\right) dE \approx \tfrac{1}{24} g\left(\tfrac{5+3\sqrt{5}}{20}, \tfrac{5-\sqrt{5}}{20}, \tfrac{5-\sqrt{5}}{20}\right) + \tfrac{1}{24} g\left(\tfrac{5-\sqrt{5}}{20}, \tfrac{5+3\sqrt{5}}{20}, \tfrac{5-\sqrt{5}}{20}\right) +$$

$$\tfrac{1}{24} g\left(\tfrac{5-\sqrt{5}}{20}, \tfrac{5-\sqrt{5}}{20}, \tfrac{5+3\sqrt{5}}{20}\right) + \tfrac{1}{24} g\left(\tfrac{5-\sqrt{5}}{20}, \tfrac{5-\sqrt{5}}{20}, \tfrac{5-\sqrt{5}}{20}\right).$$

This quadrature rule is accurate for a trilinear function $g$.

## 4  A Numerical Experiment

To test algorithm, several numerical experiments were carried out. The main purpose was to study order of convergence. We take the components of velocity

$$u = 11y(1-y)z(1-z)\left[\pi/2 - \mathrm{arctg}(x)\right],$$
$$v = w = \mathrm{arctg}\left(x(1-x)y(1-y)z(1-z)(1+t)/10\right)$$

and consider the function $\rho = 1.1 + \sin(t\,x\,y\,z)$, $t \in [0,1]$. We substituted these functions into (1) and get the function $f$. To study the convergence of proposed scheme, we use the discrete analogue of the $L_1$-norm

$$\left\|\rho^{h,m}\right\|_{L_1^h} = \sum_{i,j,k=0}^{N} \left|\rho_{i,j,k}^{h,m}\right| \mathrm{meas}\left(\Omega_{i,j,k}\right).$$

We compute numerical solution $\rho^h$ on the set of space grids with different grid steps. Denote by $\rho_n^h$ a numerical solution $\rho^{h,M}$ computed on the grid $D^h$ with parameter $N = 10 \cdot 2^n$, $n = 0, \ldots, 5$. Since order of convergence depends on $h$ and $\tau$, we put $\tau = c\,h$. To make inequality (5) valid, we set $c = 0.5$. We define

$$\sigma(n) = \log_2 \left(\frac{\left\|\rho - \rho_{n-1}^h\right\|_{L_1^h}}{\left\|\rho - \rho_n^h\right\|_{L_1^h}}\right)$$

to evaluate the order of convergence. As shown in Fig. 3, the scheme has the first order of convergence.

In next numerical test we check validity of conservation low for numerical solution. For this purpose we consider level set equation for the Enright test [14]

$$\rho_{\mathrm{init}}(\mathbf{x}) = \begin{cases} 1, & \text{if } |\mathbf{x} - \mathbf{x}_c| \leq R^2, \\ 0, & \text{otherwise,} \end{cases} \qquad f = 0 \tag{16}$$
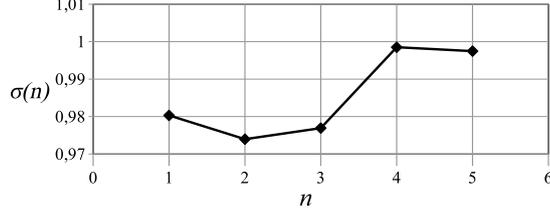
**Fig. 3.** The order of convergence

where $R = 0.1$, $\mathbf{x}_c = (0.35, 0.35, 0.35)$, $t \in [0, 1]$. The velocity functions we define in the following way:

$$
\begin{aligned}
u &= 2\cos(\pi t)\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z), \\
v &= -\cos(\pi t)\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z), \\
w &= -\cos(\pi t)\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z).
\end{aligned} \tag{17}
$$

Computations are performed under condition $\tau = h/5$. To check the validity of conservation low, we consider two integrals

$$
I_0 = \int_D \rho_I^h(0, \mathbf{x})\, d\mathbf{x}, \quad I_N = \int_D \rho_I^h(T, \mathbf{x})\, d\mathbf{x}. \tag{18}
$$

Since velocity functions $u, v, w$ are equal to zero on boundary $\partial D$, then values $I_0$ and $I_N$ should be the same for conservative method. It is easy to understand that if $\rho_{i,j,k}^{h,m} \geq 0$, then

$$
\int_D \rho_I^h(t_m, \mathbf{x})\, d\mathbf{x} = \sum_{i,j,k=0}^{N} \left| \rho_{i,j,k}^{h,m} \right| \operatorname{meas}(\Omega_{i,j,k}) = \left\| \rho^{h,m} \right\|_{L_1^h}.
$$

Therefore to evaluate the validity of conservation low, we consider two items

$$
I_0 = \left\| \rho^{h,0} \right\|_{L_1^h}, \quad \delta I = \frac{\left| \left\| \rho^{h,0} \right\|_{L_1^h} - \left\| \rho^{h,0} \right\|_{L_1^h} \right|}{\left\| \rho^{h,0} \right\|_{L_1^h}}.
$$

This values presented in the table below. The items of $\delta I$ are equals to zero

**Table 1.** The numerical result for second test problem

| $n$ | $N$ | $I_0$ | $\delta I$ |
|---|---|---|---|
| 0 | 10 | $8.00E-03$ | $3.89E-09$ |
| 1 | 20 | $3.37E-03$ | $8.50E-10$ |
| 2 | 40 | $3.92E-03$ | $1.88E-10$ |
| 3 | 80 | $4.11E-03$ | $3.02E-11$ |
| 4 | 160 | $4.17E-03$ | $4.26E-12$ |

with precision of computational error.

## 5   Conclusion

We develop new algorithm from the family of semi-Lagrangian methods for the three-dimensional advection equation. The main feature of the scheme consists in the technique of the computation of an integral at the previous time level. The described approach allows to avoid algorithmic complexity and to decrease resource-intensive computations. The numerical experiments show the first-order convergence and validity of conservation low for numerical solution.

## References

1. Wiin-Nielson, A.: On the application of trajectory methods in numerical forecasting. Tellus 11, 180–186 (1959)
2. Robert, A.: A stable numerical integration scheme for the primitive meteorological equations. Atmosphere-Ocean 19, 3546 (1981)
3. Morton, K.: Numerical Solution of Convection-Diffusion Problems. Chapman and Hall, London (1996)
4. Russell, T., Celia, M.: An over view of research on Eulerian-Lagrangian localized adjoint methods. Adv. Water Resources 25, 1215–1231 (2002)
5. Andreeva, E., Vyatkin, A., Shaidurov, V.: The semi-Lagrangian approximation in the finite element method for Navier-Stokes equations for a viscous incompressible fluid. AIP Conference Proceedings 1611, (2014); doi: 10.1063/1.4893794
6. Celledoni, E., Kometa, B., Verdier, O.: High Order Semi-Lagrangian Methods for the Incompressible Navier-Stokes Equations. J. Sci. Comput. 66(1), 91-115 (2016)
7. Cameron, A., Raynaud, R., Dormy, E.: Multi-stage high order semi-Lagrangian schemes for incompressible flows in Cartesian geometries. Int. J. Numer. Meth. Fluids 82, 879-892 (2016); doi: 10.1002/fld.4245
8. Lentine, M., Gretarsson, J., Fedkiw, R.: An unconditionally stable fully conservative semi-Lagrangian method. J. Comput. Phys. 230, 2857–2879 (2011)
9. Arbogast, T., Wang, W.: Convergence of a Fully Conservative Volume Corrected Characteristic Method for Transport Problems. SIAM J. Numer. Anal. 48(3), 797–823 (2010).
10. Efremov, A., Karepova, E., Shaydurov, V., Vyatkin, A.: A Computational Realization of a Semi-Lagrangian Method for Solving the Advection Equation. Journal of Applied Mathematics 2014, ID 610398 (2014); doi: 10.1155/2014/610398
11. Phillips, T., Williams, A.: Conservative Semi-Lagrangian Finite Volume Schemes. Numer. Meth. Part. Diff. Eq. 17, 403-425 (2001).
12. Behrens, J., Mentrup, L.: A conservative scheme for 2D and 3D adaptive semi-Lagrangian advection. Contemporary Mathematics 383, 175–19 (2005)
13. Vyatkin, A.: A semi-Lagrangian algorithm based on the integral transformation for the threedimensional advection problem. AIP Conference Proceedings 1684, 090012 (2015); doi: 10.1063/1.4934337
14. Terekhov K.M., Nikitin K.D., Olshanskii M.A. and Vassilevski Yu.V. A semi-Lagrangian method on dynamically adapted octree meshes Russ. J. Numer. Anal. Math. Modelling 30(6), 363-380 (2015).