Proceedings of Czech–Japanese Seminar in Applied Mathematics 2004
August 4-7, 2004, Czech Technical University in Prague
http://geraldine.fjfi.cvut.cz
pp. 96–105

ARBITRARY LAGRANGIAN-EULERIAN (ALE) CODE FOR PLASMA SIMULATIONS

MILAN KUCHAŘÍK AND RICHARD LISKA[1]

Abstract. Lagrangian codes with mesh moving with the fluid suffer in many cases by mesh distortion leading to tangled mesh which in principle stops the computation as basic assumptions of the numerical method as e.g. positive volume of cells are not valid any more. Arbitrary Lagrangian-Eulerian (ALE) method overcomes this difficulty by untangling or smoothing the mesh and conservative remapping of conserved quantities from the old mesh to the new one. After this Lagrangian computation can continue. We will report on the development of ALE code for simulation of problems from plasma physics.

Key words. ALE methods, conservative interpolations, quantity remapping, laser plasma simulations.

AMS subject classifications. 35L60, 35L65, 76N1

1. Introduction. In this paper we describe our newly developed 2D ALE code for fluid and plasma simulations on logically orthogonal grids. Our task is to perform laser plasma simulations, where considerable changes of computational domain volume and shape with moving boundaries appear. The pure Eulerian approach is not suitable for such simulations, as it is unable to treat moving boundary conditions. In the Eulerian approach, the computational grid if fixed in time, it does not move at all. There exists a mass flux between computational cells, representing the fluid movement through the grid. In the Lagrangian approach, the computational grid moves with the fluid, so changes of the computational domain make no problems in the simulation. On the other hand, pure Lagrangian computation can lead to severe grid distortions, e.g. by a shear in velocity, so the mathematical assumptions of the numerical method are violated.

A way, how to deal with these problems is using the Arbitrary Lagrangian-Eulerian (ALE) method. In the ALE method, several steps of pure Lagrangian computation are performed. Then, some mesh rezoning technique is used to keep the computational grid smooth and convex during the whole computation. Finally, the last part of the ALE method comes – quantity remapping which recomputes the conservative quantities from the original Lagrangian grid, to the new, smoothed one. The complete ALE method combines positives of both approaches – the computational grid moves with the fluid as in the Lagrangian approach, but the smoothing-remapping parts corresponding to the Eulerian approach, as it allows the mass flux between cells, keeps the computational mesh smooth.

The ALE method was first proposed in 1974 [9], and in recent years it is becoming more and more popular, see e.g. [10, 1].

[1] Department of Physical Electronics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Trojanova 13, 120 00 Prague, Czech Republic.

2. Lagrangian Step. Hydrodynamical Euler equations, describing the conservation of mass, momentum and energy, in Lagrangian coordinates are

$$\frac{1}{\rho}\frac{d\rho}{dt} = -\operatorname{div}\,\vec{v} \tag{2.1}$$

$$\rho\frac{d\vec{v}}{dt} = -\operatorname{grad}\,p \tag{2.2}$$

$$\rho\frac{de}{dt} = -p\operatorname{div}\,\vec{v}, \tag{2.3}$$

where ρ is density, p is pressure, e is specific internal density, and $\vec{v} = d\vec{x}/dt$ is velocity vector. Note that the differential operator d/dt in (2.1)-(2.3) applied to quantities depending on space and time is the operator of total derivative, e.g. $d\rho/dt = \partial\rho/\partial t + \vec{v}\cdot\operatorname{grad}\rho$, advective term $\vec{v}\cdot\operatorname{grad}\rho$ is included in the Lagrangian moving coordinates. This system is closed by the equation of state giving pressure as a function of internal energy and density $p(e,\rho)$. The ideal gas equation of state is used for testing fluid dynamics problems and quotidian equation of state (QEOS) [14] for plasma problems. Compatible differencing proposed in [6] and [5] is employed to treat this system numerically in 2D on quadrilateral, logically rectangular computational mesh and is explained in this section. Staggered discretization, typically used in Lagrangian simulations [1], places scalar quantities ρ, e, p into the mesh cells and vector quantities $\vec{v}, \vec{x}$ into the mesh nodes. Each quadrilateral cell, zone, is divided into four subzones by connecting the cell center with the edge centers as shown in Fig. 2.1.

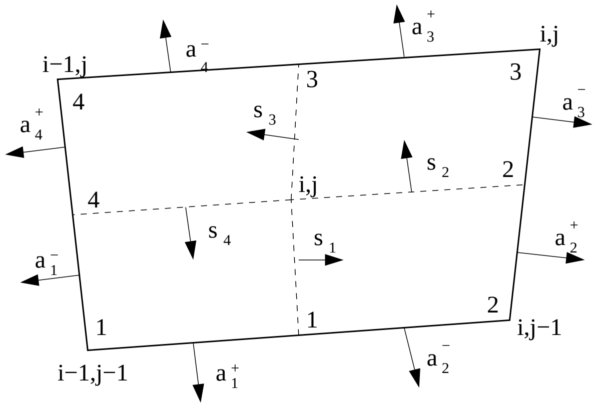


Fig. 2.1. *Quadrilateral cell and its four subzones. The vectors $\vec{s}_k$ are perpendicular to the line connecting the cell center and the appropriate edge center and have the length of this segment. The vectors $\vec{a}_k^+$ and $\vec{a}^- + k$ are normals to the edge with length of the half of the edge.*

The movement of the Lagrangian computational mesh is obtained by solving the ordinary differential equation

$$\frac{d\vec{x}_n}{dt} = \vec{v}_n \tag{2.4}$$

for each node n. The density is obtained by the standard way from mesh movement and Lagrangian assumption that mass does not flow through the mesh edges, so that the masses of cells remain constant. The discrete form of the momentum equation

(2.2) at node n can be written as

$$m_n \frac{d\vec{v}_n}{dt} = \vec{F}_n = \sum_{c \in C(n)} \vec{f}_c^n . \tag{2.5}$$

where $\vec{f}_c^n$ is the force from the cell c to the node n, $\vec{F}_n$ is the total force from four cells (from the node neighborhood $C(n)$) neighboring the node n. Each force $\vec{f}_c^n$ has three components – zonal pressure force $f_c^{\mathrm{p}\ n}$, subzonal pressure force $f_c^{\mathrm{dp}\ n}$, and artificial viscosity force $f_c^{\mathrm{q}\ n}$

$$f_c^n = f_c^{\mathrm{p}\ n} + f_c^{\mathrm{dp}\ n} + f_c^{\mathrm{q}\ n}.$$

and is responsible for changes in the velocity. The forces $\vec{f}_c^n$ are also used in the discretization of (2.3)

$$m_c \frac{de_c}{dt} = - \sum_{n \in N(c)} \vec{f}_n^c \, \vec{v}_n . \tag{2.6}$$

which guarantees conservation of the total energy [6, 5]. Here $N(c)$ is the set of four nodes of the cell c.

The zonal pressure forces

$$\vec{f}_c^{\mathrm{p}\ n} = p_c \left(\vec{s}_{n+1}^c - \vec{s}_n^c \right),$$

express the force from zonal pressure in cell c to the node n. The vectors $\vec{s}_k$ are defined in Fig. 2.1. The subzonal pressure force was developed to control the artificial grid distortion and prevent the hourglass motion of the grid cells [7]. The subzones are assumed to be also Lagrangian, they have a constant mass, the subzonal density can be calculated and can differ in different subzones of the same cell. The subzonal pressure forces are calculated proportionally to the differences between the subzonal and zonal densities. For each subzone, the pressure difference is

$$\delta P_n^c = (\rho_n^c - \rho_c) \frac{s_c^2}{\gamma},$$

where s_c is the speed of sound in the cell c and γ is the gas constant. Here we assume, that the specific internal energy is the same in the subzone as in the cell. The corner force due to this pressure difference is than calculated by integrating around the boundary of the subzone which gives

$$\vec{f}_{\mathrm{dp}\ n}^c = \delta P_n^c \, (a_n^+ + a_n^-) + \frac{1}{2} \left((\delta P_n^c - \delta P_c^{n+1}) \vec{s}_{n+1}^c + (\delta P_{n-1}^c - \delta P_c^n) \vec{s}_n^c \right),$$

where $a_k^\pm$ vectors are defined in the Fig. 2.1 again. For more details see [5].

For computing the artificial viscosity force, just a simple "bulk viscosity" can be used. The Kurapatenko [12] scalar artificial viscosity in each cell is

$$q_c = \begin{cases} \rho_c \left(c_2 \frac{\gamma-1}{4} |dV_c| + \sqrt{2\, c_2^2 \frac{\gamma-1}{4} dV_c^2 + c_1 \, s_c} \right) |dV_c| & \text{for } \operatorname{div}(\vec{v}_c) < 0 \\ 0 & \text{else} \end{cases},$$

where $\operatorname{div}(\vec{v}_c)$ is the numerical divergence of the velocity in the cell c, dimensionless constants $c_1 = 1$ and $c_2 = 1$ multiply the linear and nonlinear viscosity terms, and

$dV_c = -\operatorname{div}(\vec{v})\, l_c$, where the characteristic length of the cell l_c is set to the square root of the cell volume $l_c = \sqrt{V(c)}$. The appropriate corner viscosity force is than computed by

$$f_c^{q\ n} = q_c \left(\vec{s}^{\,c}_{n+1} - \vec{s}^{\,c}_n \right), \tag{2.7}$$

which finishes the description of spatial discretization.

The system of ordinary differential equations (2.4) for nodes position, (2.5) for nodes velocity and (2.6) for cells internal energy is numerically treated by second order Runge-Kutta method for all nodes and all cells.

3. Mesh Rezoning. Mesh rezoning part of the ALE algorithm performs untangling and smoothing techniques to repair possible distorted Lagrangian grids. For minimization of the numerical error of the following remapping algorithm, we want to move just the vertexes, that are necessary to be moved, and as little as possible. There exist many untangling and smoothing techniques for mesh rezoning. The combination of the feasible set method and global optimization [20] is a very strong tool for untangling of severely distorted grids. In real computation, we prefer to keep the grid smooth during the whole computation. For smoothing easier and faster methods are available.

A classical method is the Winslow's iterative smoothing [21] technique. Starting with the original Lagrangian mesh one smoothing iteration applied to nodes positions $\vec{x}_{i,j} = (x_{i,j}, y_{i,j})$ is given by

$$\vec{x}^{\,k+1}_{i,j} = \frac{1}{2\,(\alpha^k + \gamma^k)} \Big(\alpha^k \, (\vec{x}^{\,k}_{i,j+1} + \vec{x}^{\,k}_{i,j-1}) + \gamma^k \, (\vec{x}^{\,k}_{i+1,j} + \vec{x}^{\,k}_{i-1,j}) \\ - \frac{1}{2}\, \beta^k \, (\vec{x}^{\,k}_{i+1,j+1} - \vec{x}^{\,k}_{i-1,j+1} + \vec{x}^{\,k}_{i-1,j-1} - \vec{x}^{\,k}_{i+1,j-1}) \Big),$$

where the coefficients $\alpha^k = x_\xi^2 + y_\xi^2$, $\beta^k = x_\xi\, x_\eta + y_\xi\, y_\eta$, $\gamma^k = x_\eta^2 + y_\eta^2$, and (ξ, η) are logical coordinates $\xi_i = i/M$, $\eta_j = j/N$ for $i = 0, \ldots, M$ and $j = 0, \ldots, N$. The derivatives x_ξ, x_η are approximated by the central differences

$$(x_\xi)_{i,j} \approx \frac{x_{i+1,j} - x_{i-1,j}}{2\,\Delta\xi}, \qquad (x_\eta)_{i,j} \approx \frac{x_{i,j+1} - x_{i,j-1}}{2\,\Delta\eta} \tag{3.1}$$

and similarly for y. The Winslow's smoothing technique is very fast, easy to implement and it can be used in more iterations during one rezoning process to obtain smoother grids for the next Lagrangian step.

4. Quantity Remapping. To complete our ALE method we need a remapping algorithm for recomputing the conservative quantities from the original, Lagrangian grid to the new, smoother one. We have two different, but similar grids: the Lagrangian grid $\{c\}$, and rezoned smoother one $\{\tilde{c}\}$ created by small movements of nodes of the original one. Suppose, there exists some underlying function g defined on the original grid expressing some conservative quantity, e.g. $g = \rho$, $g = \rho\, u$, $g = \rho\,(e + |\vec{v}|^2/2)$. We know just the mean value g_c of this function in each cell c and its total value, or mass $m_c = \int_c g dV = g_c V(c)$. Our task now is to find the new mean values in the rezoned cells $\tilde{g}_{\tilde{c}}$, and we have several conditions which these new means must satisfy:

- Accuracy: $\tilde{m}_{\tilde{c}} \approx \int_{\tilde{c}} g(\mathbf{r})\, dV$ – to be as close to the exact means, as possible.

- Local-bound-preservation: $g_c^{\max} \geq \tilde{g}_{\tilde{c}} \geq g_c^{\min}$ – the new mean value must be between the local extremes of the mean values over the neighborhood of the original cell.
- Conservation: $\sum_{\forall\, c} \tilde{m}_{\tilde{c}} = \sum_{\forall\, c} m_c$ – total mass must be the same.
- Linearity-preservation: $\tilde{m}_{\tilde{c}} = \int_{\tilde{c}} g(\mathbf{r})\, dV$ if $g(\mathbf{r}) = a + b\, x + c\, y$ – exact for linear function.

The remapping method is based on piecewise-linear reconstruction of the underlying function in each original cell. Just piecewise-constant reconstruction would violate our demand for linearity-preservation. Then, the "exact integration" of the reconstructed function over overlapped areas would be most natural. Unfortunately, this would require finding the intersections of the new and the original cells, which is very time consuming, so the algorithm would be very slow. Our approach is based on approximate integration, it does not require finding these intersections and thus it is much more efficient. On the other hand, there are some problems with local-bound-preservation, so the third stage is needed – repair. The repair is a conservative redistribution of conservative quantities constructed to enforce satisfaction of this condition.

4.1. Stage 1 – Piecewise-linear Reconstruction. Piecewise-linear reconstruction is the first part of the remapping algorithm. We can use any method for slope computation, using or not using limiters, the only condition for us is the bound-preservation. We present here a method based on the computation of the unlimited slopes, followed by the limiting.The formulas here are shown in 1D case, generalization to multidimensional case is straightforward, just the computation of the slopes in all directions is needed. At first, we compute the unlimited slope from the mean values of the neighboring cells. For example one can use the simple central difference

$$s_{i+\frac{1}{2}}^{\text{unlim}} = \left(\frac{\delta g}{\delta x}\right)_{i+\frac{1}{2}}^{\text{unlim}} = \frac{g_{i+\frac{3}{2}} - g_{i-\frac{1}{2}}}{x_{i+\frac{3}{2}} - x_{i-\frac{1}{2}}}.$$

When using this pure unlimited formula the local bounds might not be preserved. To preserve them we are looking for the final limited slope in the form

$$s_{i+\frac{1}{2}}^{\text{BJ}} = \left(\frac{\delta g}{\delta x}\right)_{i+\frac{1}{2}}^{\text{BJ}} = \Phi_{i+\frac{1}{2}} \left(\frac{\delta g}{\delta x}\right)_{i+\frac{1}{2}}^{\text{unlim}}, \quad \Phi_{i+\frac{1}{2}} = \min(\Phi_i^{i+\frac{1}{2}}, \Phi_{i+1}^{i+\frac{1}{2}}), \tag{4.1}$$

where the Barth-Jasperson limiter [2] $\Phi^{i+\frac{1}{2}}$ is employed to satisfy the bounds.

4.2. Stage 2 – Numerical Quadrature. As said above, now it is the easiest to integrate the reconstructed function over overlapped areas of the original and the new cells. This "exact integration" requires finding of all intersections of both grids, the complete algorithm is then very slow and inefficient. Our approach [11] is based on the approximate integration over swept regions. By the swept region we mean the quadrilateral area defined by smooth movement of the cell edge from initial to its new position. Just a small note, the exact integration method is very easy in 1D, it is very inefficient, but possible to be done in 2D, and almost impossible in 3D.

The main idea of the swept-area integration is the fact that the integral over the new cell can be decomposed as a sum of integrals over swept regions. Then, the volume integral of the polygonal (linear) function over the arbitrary polyhedron (polygon) can be reduced to the boundary integral which can be evaluated exactly.

The mass of the new cell can be computed as the original mass plus signed masses of all swept regions

$$\tilde{m}_{i+\frac{1}{2},j+\frac{1}{2}} = m_{i+\frac{1}{2},j+\frac{1}{2}} + F_{i+1,j+\frac{1}{2}} - F_{i,j+\frac{1}{2}} + F_{i+\frac{1}{2},j+1} - F_{i+\frac{1}{2},j}.$$

Here the mass of the swept region $\delta_{i,j+\frac{1}{2}}$ is computed as the signed integral of the reconstructed function over this region

$$F_{i,j+\frac{1}{2}} = \int\limits_{\delta_{i,j+\frac{1}{2}}} g_{i,j+\frac{1}{2}}(x,y)\, dV \tag{4.2}$$

and this linear reconstruction is taken from the neighboring cell, in which more of the swept region lies.

4.3. Stage 3 – Repair. The swept-region integration algorithm satisfies all our conditions, with only one exception, it does not have to preserve local bounds. To enforce it, we apply the last stage – repair, the conservative redistribution of the conserved quantity. At first, we define the bound-determining neighborhood $\mathcal{C}(c)$, the segment of the original grid fully covering the new cell. For example, it can be the original cell and 8 its nearest neighbors. Then we compute the local minimum and maximum over this bound-determining neighborhood

$$g_c^{\min} = \min_{k\in\mathcal{C}(c)} g_k\,, \quad g_c^{\max} = \max_{k\in\mathcal{C}(c)} g_k\,. \tag{4.3}$$

We check, whether the new mean value $\tilde{g}_{\tilde{c}}$ from the numerical quadrature integration lies within this range. If yes, we do not do anything, the new mean value satisfies all our conditions. We emphasize, that this is always true for the global linear function, so the repair stage does not affect the linearity-preservation of the integration stage. If the local bound is violated, we have to perform the repair.

Suppose that the lower bound is violated $\tilde{g}_{\tilde{c}} < g_c^{\min}$. We search for mass, which can safely be taken from the neighboring cells without violating their lower bound. If we find enough mass, we perform the repair – we bring the violated value to the local minimum and take the mass from the neighboring cells proportionally to the mass available. If there is not enough mass available in the neighborhood, we extend the neighborhood and search again. We have proved, that this conservative redistribution of mass (repair) process is always successful in a finite number of steps [11]. If forces the remapping algorithm to be local-bound-preserving. The approach outlined here is applicable to bounds in conserved quantities. Physically however more natural is to use local bounds in density, internal energy and velocity which prevent some nonphysical situations as negative internal energy. The repair of conserved quantities with these bounds is developed in [18].

Other complication is introduced by the staggered discretization used in the Lagrangian step described in the section 2. We perform remapping of all quantities in subzones, however one must be very careful during the reconstruction of velocities in nodes and internal energies in cells to keep total energy conservation [13].

The complete remapping algorithm is conservative, linearity and local-bound preserving and stable. It is very efficient due to the fact, that it is face based, and all computed quantities (for example, swept masses) can be used for both adjacent cells. The algorithm is applicable to general unstructured meshes in 2D [11] and 3D [8].

5. Heat Conductivity. The heat conductivity is essential for many problems in laser plasma physics. The heat conductivity term $\operatorname{div} \kappa \operatorname{grad} T$ (with T being the temperature and κ the heat conductivity) is added to the right hand side of the energy equation (2.3). The full system is split into hyperbolic part (2.1)-(2.3) and the parabolic part. The parabolic part reduces into the heat equation

$$T_t - \operatorname{div} K \operatorname{grad} T = f \tag{5.1}$$

which is treated on the region V after each Lagrangian step (before possible rezone-remap step) by the mimetic discretization [17] using the support operators method [16]. The integral properties of differential operators div and grad are required to hold also for their discrete approximations, so that the discrete operators mimic the properties of the continuum ones.

First the generalized gradient $\mathbf{w} = \mathbf{G}u = -K \operatorname{grad} u$ and extended divergence

$$\mathbf{D}\,\mathbf{w} = \begin{cases} \operatorname{div} \mathbf{w} & \text{on} \quad V \\ -(\mathbf{w}, \mathbf{n}) & \text{on} \quad \partial V \end{cases}$$

operators are defined. The integral property of extended divergence is given by the divergence Green formula

$$\int_V \operatorname{div} \mathbf{w}\, d\, V - \oint_{\partial V} (\mathbf{w}, \mathbf{n})\, d\, S = 0$$

which can be expressed as $(\mathbf{D}\ \mathbf{w}, 1)_H = 0$ where $(u, v)_H = \int_V u\ v\ d\ V + \oint_{\partial V} u\ v\ d\ S$ is the inner product on the space H of smooth scalar functions on the region V. The Gauss theorem

$$\int_V u \operatorname{div} \mathbf{w}\ d\ V - \oint u(\mathbf{w}, \mathbf{n})\ d\ S + \int_V (\mathbf{w}, K^{-1} K \operatorname{grad} u) d\ V = 0$$

is the integral property relating the operators divergence and gradient. It can be rewritten as $(\mathbf{D}\mathbf{w}, u)_H = (\mathbf{w}, \mathbf{G}u)_{\mathbf{H}}$ where $(\mathbf{A}, \mathbf{B})_{\mathbf{H}} = \int_V (K^{-1}\mathbf{A}, \mathbf{B}) d\ V$ is the inner product on the space $\mathbf{H}$ of smooth vector functions on the region V. The generalized gradient operates from scalar to vector space $\mathbf{G} : H \to \mathbf{H}$ and extended divergence from vector to scalar space $\mathbf{D} : \mathbf{H} \to H$. The Gauss theorem expresses the fact that $\mathbf{G}$ is adjoint operator of $\mathbf{D}$ $\mathbf{G} = \mathbf{D}^*$ in the sense of these inner products. One can show that the operator $\mathbf{DG}$ needed to treat (5.1) is self-adjoint and positive definite.

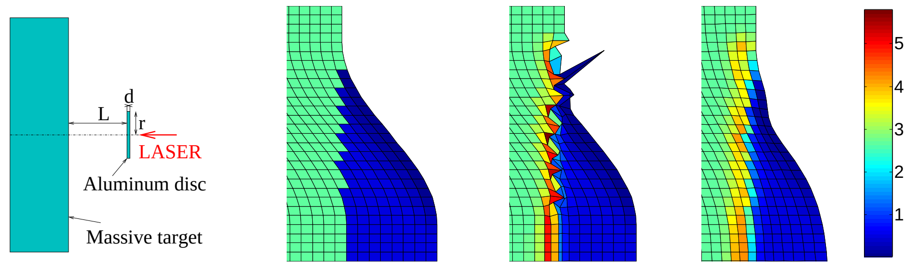
Mimetic discretization defines the discrete function spaces $HC, \mathbf{HL}$ and the discrete operators (div,grad) $D : \mathbf{HL} \to HC, G : HC \to \mathbf{HL}$. The discrete Green formula can be written as $(D\mathbf{W}, 1)_{HC} = 0$ (with appropriate definition of discrete inner product) and defines the discrete operator D. The discrete Gauss theorem $G = D^*$ states that G is adjoint of D, i.e. $(D\mathbf{W}, U)_{HC} = (\mathbf{W}, GU)_{\mathbf{HL}}$, and it is used to compute the discrete operator G [17]. Boundary condition are naturally included through the discrete inner products. Having both discrete operators D and G a fully implicit scheme in time

$$\frac{(T^{n+1} - T^n)}{\Delta t} + DGT^{n+1} = F^{n+1} \tag{5.2}$$

is used to approximate (5.1). The matrix of global system $(I + \Delta t DG)U^{n+1} = U^n + \Delta t F^{n+1}$ remains symmetric and positive definite so that fast converging conjugate gradient method can be used to solve the implicit scheme (5.2). The described method is exact on piecewise constant or piecewise linear solutions, otherwise it is second order accurate in space. It works very well on bad quality meshes appearing in Lagrangian simulations. The diffusion coefficient K can be discontinuous.

For plasma we use either the classical Spitzer-Harm [19] or Rozmus-Offenberger [15] heat conductivity coefficient.

6. Disc Target Impact. As the numerical example we present here the simulation from laser plasma physics, laser accelerated disc impact problem, the initial conditions of which we have taken from a part of the experiment performed on the PALS (Prague Asterix Laser System) laser facility [3]. Fig. 6.1(a) shows the problem setup. A small Aluminum disc ($r = 150\,\mu m$, $d = 11\,\mu m$) is irradiated by an intense laser beam, and it is ablatively accelerated up to 82 km/s (for 390 J laser beam). The fast moving disc (flayer) strikes into a massive Aluminum target. Our simulation starts at the moment of the impact. Fig. 6.1(b) shows the initial density colormap in the critical area. The left green higher density region is the massive Aluminum target and the right blue lower density region is the impacting Aluminum disc. The density, width and temperature of the impacting disc has been obtained from a 1D Lagrangian hydrodynamical simulation. Fig. 6.1(c) presents the pure Lagrangian results in the same area after at a very short time $0.2\,ns$. The Lagrangian computational grid is already very distorted and in fact, the Lagrangian simulation fails due to the tangled mesh soon after this snapshot. On the other hand the complete ALE method shown in Fig. 6.1(d) keeps the computational mesh smooth and can continue the computation without troubles.



(a) Experiment setup

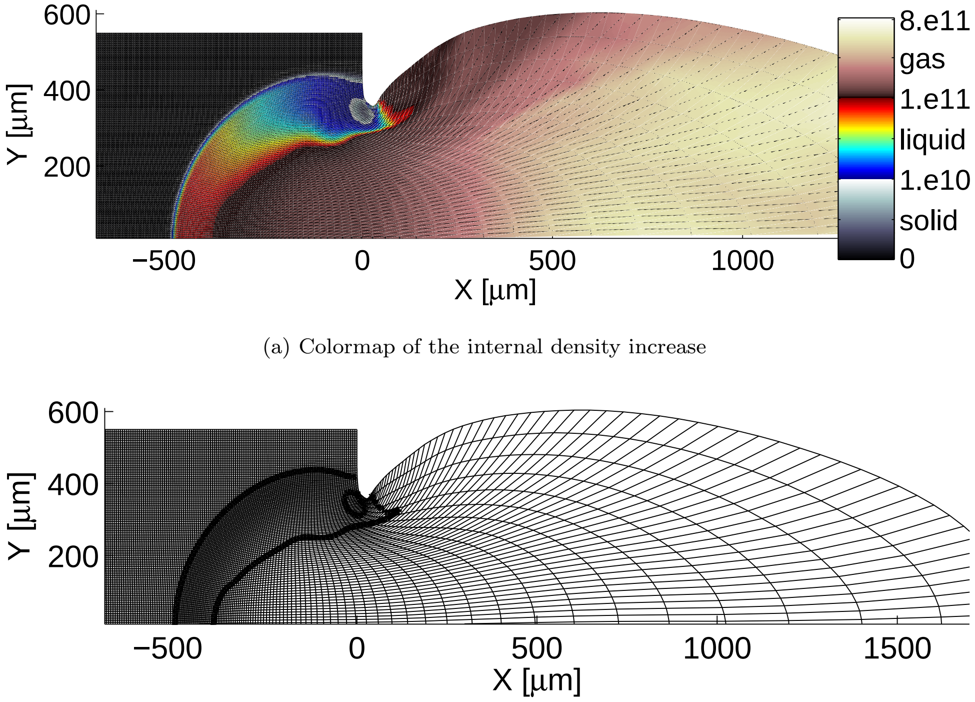
(b) Initial conditions

(c) Lagrangian simulation at time $0.2\,ns$

(d) ALE simulation at time $0.2\,ns$

Fig. 6.1. *Experimental setup, initial density of the disc impact simulation, and density of the Lagrangian and ALE simulation after* $0.2\,ns$.

After the impact, both the massive target and the disc start to raise their temperature, a shock wave is formed that propagates into the target, and causes heating, melting, and evaporation of the target material. Fig. 6.2(a) presents the increase of the internal energy at time $30ns$. The right pink colormap is used in the area, where the material has evaporated, the reflected gas is moving right. The central colored colormap shows the melted liquid part of the material, and the left black and white colormap shows material remaining in solid state. Approximately circular shock wave propagates left and is positioned at the solid-liquid interface. From the figure, we can also see the evolved crater shape. By the crater, we mean the interface between



(a) Colormap of the internal density increase

(b) Computational grid

Fig. 6.2. *Internal energy increase and final grid in time* 30 ns *after the impact. Isolines of melting and evaporation energy included.*

the gas and the liquid state, because the gas flows away, but the liquid Aluminum becomes solid again after temperature decrease. At later times crater does not propagate further left into the massive target, while the shock wave continues to move inside the target. The numerical carter shape and volume corresponds reasonably well to the experimental data [4]. Fig. 6.2(b) presents the computational grid at time 30 ns after the impact. The grid remains smooth and the computation can continue. One can distinguish the position of the circular shock wave in the mesh.

7. Conclusion. We have developed an universal linearity and local-bound preserving interpolation algorithm for conservative remapping of conservative quantities between similar grids. This remapping algorithm is robust enough to be applicable in real ALE codes. We have developed such ALE code on logically orthogonal 2D grids, which can be used for performing of laser plasma simulations. In this paper we presented the ALE simulation of the accelerated disc impact, the initial conditions of which are inspired with the real experiment. The numerical results are comparable with the experimental data.

Acknowledgment. Partial support of the project No. 202/03/H162 of the Grant Agency of the Czech Republic, No. FRVS 2087/2004 of the Ministry of Education of the Czech Republic, No. CTU 0410914 of the Czech Technical University in Prague as well as the project MSM 6840770010 of the Ministry of Education, Youth and Sport of the Czech Republic is acknowledged. The authors thank R. Garimella,

R. Loubère, M. Shashkov, P. Váchal and B. Wendroff for fruitful discussions and constructive comments.

REFERENCES

[1] R. W. Anderson, N. S. Elliott, and R. B. Pember, *An arbitrary lagrangian-eulerian method with adaptive mesh refinement for the solution of the euler equations*, J. Comp. Phys., 199 (2004), pp. 598–617.

[2] T. J. Barth, *Numerical methods for gasdynamic systems on unstructured meshes*, in An introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws, C. R. D. Kroner, M. Ohlberger, ed., Berlin, 1997, Lecture Notes in Computational Science and Engineering, Springer, pp. 195–284.

[3] S. Borodziuk, A. Kasperczuk, T. Pisarczyk, M. Kalal, J. Ullschmied, J. Limpouch, K. Rohlena, N. N. Demchenko, S. Y. Gus'kov, V. B. Rozanov, V. N. Kondrashov, and P. Pisarczyk, *Application of the three frame interferometric and crater replica methods for investigation of a laser accelerated macroparticles - massive target interaction in the PALS experiment*, Optica Applicata, 34 (2004). In Press.

[4] S. Borodziuk, A. Kasperczuk, T. Pisarczyk, K. Rohlena, J. Ullschmied, M. Kalal, J. Limpouch, and P. Pisarczyk, *Application of laser simulation method for the analysis of crater formation experiment on PALS laser*, Cz. J. Phys., 53 (2003), pp. 799–810.

[5] J. Campbell and M. Shashkov, *The compatible Lagrangian hydrodynamics algorithm on unstructured grids*, Tech. Rep. LA-UR-00-3231, Los Alamos National Laboratory, 2000. Submitted to Journal of Computational Physics.

[6] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen, *The construction of compatible hydrodynamics algorithms utilizing conservation of total energy*, Journal of Computational Physics, 146 (1998), pp. 227–262.

[7] E. J. Caramana and M. J. Shashkov, *Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures*, Journal of Computational Physics, 142 (1998), pp. 521–561.

[8] R. Garimella, M. Kuchařík, and M. Shashkov, *Efficient algorithm for local-bound-preserving remapping in ALE methods*, in Numerical Mathematics and Advanced Applications, M. Feistauer, V. Dolejsi, P. Knobloch, and K. Najzar, eds., Springer-Verlag Berlin Heidelberg New York, 2004, pp. 358–367.

[9] C. W. Hirt, A. A. Amsden, and J. L. Cook, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, Journal of Computational Physics, 14 (1974), pp. 227–253.

[10] P. Knupp, L. Margolin, and M. Shashkov, *Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods*, J. Comp. Phys., 176 (2002), pp. 93–128.

[11] M. Kuchařík, M. Shashkov, and B. Wendroff, *An efficient linearity-and-bound-preserving remapping method*, Journal of Computational Physics, 188 (2003), pp. 462–471.

[12] V. F. Kuropatenko, in Difference Methods for Solutions of Problems of Mathematical Physics, N. N. Janenko, ed., Amer. Math. Soc., Providence, 1967, p. 116.

[13] R. Loubere and M. Shashkov, *A subcell remapping method on staggered polygonal grids for ALE methods*, Tech. Rep. LA-UR-04-6692, Los Alamos National Laboratory, 2004.

[14] R. M. More, K. Warren, D. Young, and G. Zimmerman, *A new quotidian equation of state (qeos) for hot dense matter*, Phys. Fluids, 31 (1988), pp. 3059–3078.

[15] W. Rozmus and A. A. Offenberger, *Thermal conductivity for dense, laser compressed plasmas*, Phys. Rev. A, 31 (1985), pp. 1177–1179.

[16] M. Shashkov, *Conservative Finite-Difference Methods on General Grids*, CRC Press, Boca Raton, Florida, 1996.

[17] M. Shashkov and S. Steinberg, *Solving diffusion equation with rough coefficients in rough grids*, J. Comp. Phys., 129 (1996), pp. 383–405.

[18] M. Shashkov and B. Wendroff, *The repair paradigm and application to conservation laws*, J. Comp. Phys., 198 (2004), pp. 265–277.

[19] L. Spitzer and R. Harm, *Transport phenomena in a completely ionized gas*, Phys. Rev., 89 (1953), pp. 977–981.

[20] P. Váchal, R. Garimella, and M. Shashkov, *Untangling of 2d meshes in ale simulations*, J. Comp. Phys., 196 (2004), pp. 627–644.

[21] A. M. Winslow, *Equipotential zoning of two-dimensional meshes*, Tech. Rep. UCRL-7312, Lawrence Livermore National Laboratory, 1963.