Volume preserving integrators for solenoidal fields on a grid

John M. Finn, and Luis Chacón

Citation: Physics of Plasmas **12**, 054503 (2005); doi: 10.1063/1.1889156
View online: https://doi.org/10.1063/1.1889156
View Table of Contents: http://aip.scitation.org/toc/php/12/5
Published by the American Institute of Physics

Volume preserving integrators for solenoidal fields on a grid

John M. Finn and Luis Chacón
T-15, Plasma Theory, Los Alamos National Laboratory, Los Alamos, New Mexico 87545

(Received 10 January 2005; accepted 15 February 2005; published online 18 April 2005)

A novel method of integrating solenoidal flows given on a grid is developed. In three dimensions, the method involves splitting the flow into two or possibly three flows that are two dimensional and area preserving, and can therefore be integrated by Crank–Nicolson or a modified form of leapfrog, both of which exactly conserve area. The method involves an interpolation scheme which ensures the solenoidal nature of the field throughout the domain (on and off the grid) by means of tricubic splines. It is shown that the method is easily generalized to arbitrary curvilinear coordinates and also generalizes to higher dimensions. A method of integrating compressional three-dimensional flows given on a grid is outlined. © *2005 American Institute of Physics*. [DOI: 10.1063/1.1889156]

In simulation codes for incompressible fluid dynamics there is a need for exactly volume-preserving integrators for the Lagrangian trajectories of fluid elements,

$$d\mathbf{x}/dt = \mathbf{v}(\mathbf{x},t) \tag{1}$$

from a solenoidal velocity field $\mathbf{v}(\mathbf{x},t)$ given on a grid of points. Similarly, in magnetohydrodynamics (MHD) there is a need for a volume-preserving integrator for magnetic field lines $d\mathbf{x}/d\tau = \mathbf{B}(\mathbf{x})$, for a magnetic field line given on a grid. In the latter instance, the "time" τ is not the physical time. Often, the variation of $\mathbf{B}$ in time t can be ignored. An exactly volume-preserving integrator for such systems is analogous to a symplectic integrator for Hamiltonian systems. In both cases the volume-preserving (or the symplectic) property can be crucially important if the orbits are integrated for very long time intervals, over which nonvolume preserving integrators can allow orbits that converge to a fixed point and other such phenomena that are impossible for solenoidal fields.

There are two aspects that must be dealt with to obtain a volume-preserving integrator for flows given on a grid. First, the ordinary differential equation (ODE) integrator must conserve volume. Second, the interpolation scheme to obtain the flow at points off the grid must exactly (to machine precision) preserve the solenoidal nature of the flow. Often magnetic field line tracing codes in three dimensions (3D) have been constructed using time integration schemes that do not preserve volume and an interpolation scheme that does not preserve the solenoidal nature of the flow; accuracy is obtained by using very high-order time stepping and high-order interpolation. These methods can be adequate for relatively short integrations, but methods that preserve volume exactly allow arbitrarily long integrations. The methods outlined here have lower order accuracy in time (second-order accurate), but conserve volume exactly. For chaotic field lines (or chaotic Lagrangian trajectories in the fluid context) accuracy is lost rapidly with any integrator, and indeed even by roundoff.[1] However, integrable field lines are integrated accurately, and chaotic field lines and integrable field lines are distinguished well. Further, even though a chaotic field line may not be integrated accurately for long time, quantitative measures such as the Lyapunov exponent can be computed accurately.

Crank–Nicolson (CN) time stepping is area preserving in 2D, but we show that it is not volume preserving in 3D. The method of time integration we use[2] involves splitting the flow or magnetic field into separate two-dimensional flows determined by individual components of the vector potential, each of which is a stream function for its 2D flow. These 2D flows can be integrated in an area preserving method (CN or modified leapfrog), and therefore the split-step scheme is exactly volume preserving in 3D.

The method we introduce for interpolation involves first obtaining a vector potential $\mathbf{A}$ from the flow or magnetic field on the grid. This vector potential is found by a method involving line integrals and is much more efficient than methods that require solving a vector Poisson equation. Then $\mathbf{A}$ is interpolated by means of tricubic splines, and $\mathbf{v}$ or $\mathbf{B}$ is found by taking the analytic curl of the spline formulas for $\mathbf{A}$, thus assuring that the field is solenoidal to machine precision in the whole domain.

Two-dimensional flows. In two dimensions it is known[3] that CN discretization is exactly area preserving, which in 2D is equivalent to being symplectic. We give a simplified derivation in order to understand why CN is not volume preserving in 3D. For $\nabla\cdot\mathbf{v}=0$, CN takes the form

$$\mathbf{x}' = \mathbf{x} + h v[(\mathbf{x}+\mathbf{x}')/2, t + h/2] \tag{2}$$

for time step h, with $\mathbf{x}=\mathbf{x}(t)$ and $\mathbf{x}'=\mathbf{x}(t+h)$. It is second-order accurate in time, i.e., the error in $\mathbf{x}'$ is of order h^3. In practice, the implicit nature of CN requires Picard or Newton iterations. Because these ODEs are not expected to be stiff, h is a small parameter compared to the time scale for vatiation of $\mathbf{x}(t)$; typically, for values of h adequate for overall accuracy, five to seven Picard iterations two to three Newton iterations are required. The Jacobian matrix J, with $J_{ij} = \partial x_i'/\partial x_j$ satisfies

$$J_{ij} = \delta_{ij} + hF_{ij}/2 + h\sum_k F_{ik}J_{kj}/2,$$

where $F_{ij}=\partial v_i/\partial x_j$, or $\mathsf{J}=\mathsf{I}+(h/2)(\mathsf{F}+\mathsf{FJ})$. Thus

$$\mathsf{J} = (\mathsf{I} - h\mathsf{F}/2)^{-1}(\mathsf{I} + h\mathsf{F}/2). \tag{3}$$

In two dimensions we have

$$\det \mathsf{J} = \frac{1 + h\ \mathrm{Tr}(\mathsf{F})/2 + h^2 \det(\mathsf{F})/4}{1 - h\ \mathrm{Tr}(\mathsf{F})/2 + h^2 \det(\mathsf{F})/4}, \tag{4}$$

where $\mathrm{Tr}(\mathsf{F})$ is the trace of F and $\det(\mathsf{F})$ is its determinant. The condition $\nabla \cdot \mathbf{v} = 0$ implies $\mathrm{Tr}(\mathsf{F}) = 0$, showing that $\det \mathsf{J} = 1$; therefore the 2D map (2) is area preserving.

Another scheme which is area preserving in 2D is *leapfrog*, which can be used if $v_x = u(y)$ and $v_y = v(x)$. Leapfrog over one time step takes the form of a map $\mathbf{x}' = M(\mathbf{x})$:

$$x' = x + hu(y), \quad y' = y + hv(x'). \tag{5}$$

Note the Gauss–Seidel nature of this scheme: if the y integration step follows the x step, x' is used in the y step as soon as it is obtained. This can also be written in the form $M_h = P_h \circ Q_h$,

$$Q_h: \begin{cases} x_1 = x + hu(y) \\ y_1 = y, \end{cases} \qquad P_h: \begin{cases} x' = x_1 \\ y' = y_1 + hv(x_1), \end{cases}$$

so that it is a composition of two maps that are trivially area preserving. Leapfrog can be symmetrized $\tilde{M}_h = Q_{h/2} \circ P_h \circ Q_{h/2}$ to make it second-order accurate in h.

For general 2D flows we introduce a modified form of leapfrog, which can be written as $M_h = P_h^e \circ Q_h^i$

$$Q_h^i: \begin{cases} x_1 = x + hu(x_1, y) \\ y_1 = y, \end{cases} \qquad P_h^e: \begin{cases} x' = x_1 \\ y' = y_1 + hv(x_1, y_1) \end{cases}. \tag{6}$$

Combining, we have

$$x' = x + hu(x', y), \quad y' = y + hv(x', y). \tag{7}$$

The map Q_h^i is implicit and must be done by means of Picard or Newton iterations. The second map P_h^e is explicit. The map $P_h^e \circ Q_h^i$ is area preserving by the following argument: the determinant of the Jacobians of Q_h^i and P_h^e are, respectively,

$$D(Q_h^i) = \partial x_1 / \partial x = \frac{1}{1 - hu_1(x_1, y)},$$
$$D(P_h^i) = \partial y' / dy = 1 + hv_2(x_1, y_1), \tag{8}$$

where $u_1(x,y) = \partial u(x,y)/\partial x$, $v_2 = \partial v(x,y)/\partial y$. The solenoidal nature of $\mathbf{v}$ implies $u_1 + v_2 = 0$, and the fact that $y_1 = y$ leads to $D = D(P_h^e) D(Q_h^i) = [1 + hv_2(x', y)][1 - hu_1(x', y)]^{-1} = 1$. Notice that if we had used the explicit form of Q_h and the implicit form of P_h we would obtain $D = [1 - hu_1(x', y')]^{-1}[1 + hv_2(x, y)] \neq 1$.

This modified leapfrog (ML) scheme (7) can be expressed in terms of a generating function[4] $F_3(x', y) = -x'y + h\phi(x', y)$. Here, ϕ is the stream function, with $\mathbf{v} = \nabla \phi \times \hat{e}_3$, and the map (7) is obtained by the usual relations $x = -\partial F_3 / \partial y$, $y' = -\partial F_3 / \partial x'$.

Finally, ML can be symmetrized by composing $Q_{h/2}^e \circ P_{h/2}^i \circ P_{h/2}^e \circ Q_{h/2}^i$. The order implicit-followed-by-explicit in the first pair of maps and in the second pair retains the area preserving nature; the alternation of the steps in x and y leads to second-order accuracy in h, a property that can be checked directly.

Three-dimensional flows. In MHD it is often assumed that a component of the magnetic field, e.g., B_z never vanishes, and the 3D flow along field lines is equivalent to a nonautonomous one degree of freedom (1.5 degree of freedom) Hamiltonian system,[5–8] e.g., $[dx/dz = B_x/B_z,\ dy/dz = B_y/B_z]$, where z is treated like time. However, in a field with nulls, this reduction is not possible and the nonlinear dynamics of the field lines can be qualitatively different from that of a Hamiltonian system.[9] So we must find a volume-preserving method of integrating 3D flows. Let us first consider CN in three dimensions. Following the line of reasoning (2)–(4) we find, with Eq. (3) and the divergence-free condition $\mathrm{Tr}(\mathsf{F}) = 0$,

$$\det(\mathsf{J}) = \frac{1 + h^2 Q(\mathsf{F})/4 - h^3 \det(\mathsf{F})/8}{1 + h^2 Q(\mathsf{F})/4 + h^3 \det(\mathsf{F})/8} = 1 + O(h^3),$$

where $Q(\mathsf{F}) = F_{11}F_{22} + F_{22}F_{33} + F_{33}F_{11} - F_{12}F_{21} - F_{23}F_{32} - F_{31}F_{13}$. Thus, the error in the volume in CN per time step is $O(h^3)$, the same as the overall order of accuracy of x'. Thus, even though CN is area preserving in 2D, it is *not* a volume-preserving integrator in 3D.

A generalization of ML, i.e., a generating function approach, is not completely straightforward because the reasoning leading to Eq. (8) involves an implicit step in one variable and an explicit step in the other. There is, however, a generalization of the generating function approach (leading above to ML) to 3D, described in Ref. 10. The volume-preserving integrators of Ref. 11 are special cases of those of Ref. 10.

A conceptually simpler scheme in 3D is based on Ref. 2. Writing the solenoidal field as $\mathbf{v} = \nabla \times \mathbf{A}$, we split the flow into three parts

$$\mathbf{v}_1 = \nabla A_x \times \hat{e}_x, \quad \mathbf{v}_2 = \nabla A_y \times \hat{e}_y, \quad \mathbf{y}_3 = \nabla A_z \times \hat{e}_z.$$

We then integrate separately in split steps, i.e.,

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}_1, \quad \frac{d\mathbf{x}}{dt} = \mathbf{v}_2, \quad \frac{d\mathbf{x}}{dt} = \mathbf{v}_3. \tag{9}$$

Notice that the three separate flows are two-dimensional, e.g., $\mathbf{v}_1$ is in the y-z plane and A_x is the stream function for this 2D flow. Therefore we can use CN or ML to integrate the three separate 2D divergence-free flows. Thus we write $N = N_3(h) \circ N_2(h) \circ N_1(h)$, where map N_i consists of CN or ML with flow $\mathbf{v}_i$, and each map N_i is volume preserving.

There is some simplification if a gauge in which, for example, $A_x = 0$ is used. In this case there are only two 2D volume-preserving flows, specified by stream functions A_y and A_z.

Second-order accuracy can be assured while retaining the volume-preserving nature by integrating Eq. (9) for a half time step $h/2$ followed by reversing the order. This leads to

$$N_1(h/2) \circ N_2(h/2) \circ N_3(h) \circ N_2(h/2) \circ N_1(h/2). \tag{10}$$

Using data on a grid. Let us first treat the special case in which the flow $\mathbf{v}(\mathbf{x})$ is known on a uniformly spaced rectangular grid $x_i = x_{\min} + (i-1)\Delta_x$, $y_j = y_{\min} + (j-1)\Delta_y$, $z_k = z_{\min} + (k-1)\Delta_z$, with $\Delta_x \sim \Delta_y \sim \Delta_z \sim \Delta$. We will assume that the flow is solenoidal on the grid, i.e., that the grid divergence is zero, although this assumption is not essential. We compute the

vector potential $\mathbf{A}$ for which $\mathbf{v}=\nabla\times\mathbf{A}$, in a gauge with $A_x=0$, on the grid by

$$A_y(x,y,z)=\int_{x_{\min}}^{x} v_z(x',y,z)dx', \tag{11}$$

$$A_z(x,y,z)=\int_{y_{\min}}^{y} v_x(x_{\min},y',z)dy'-\int_{x_{\min}}^{x} v_y(x',y,z)dx', \tag{12}$$

using approximations to the line integrals which are accurate to order Δ^2. The first term in Eq. (13) is $A_z(x_{\min},y,z)$ and we have chosen, without loss of generality $A_z(x_{\min},y_{\min},z)=0$.

In the next step we find the tricubic spline interpolation for $A_y(x,y,z)$ and $A_z(x,y,z)$. This interpolation has $\mathbf{A}$ and its first and second derivatives continuous across the cells, accurate to order Δ^3. We then differentiate *analytically* the spline interpolation formula to obtain a flow $\mathbf{v}(\mathbf{x})$ which is exactly divergence-free throughout the domain, and has $\mathbf{v}$ and its first derivatives continuous across the cells, accurate to order Δ^2. The flow $\nabla\times\mathbf{A}$ evaluated on the grid will not be exactly equal to the specified values of $\mathbf{v}$ on the grid. On the other hand, if the original flow is not divergence-free on the grid due to numerical errors, this equality is not expected; in this case the above procedure yields a divergence cleaned field.

Example in 3D. We have constructed a model, consisting of an axisymmetric magnetic field $\mathbf{B}=\nabla\psi(r,z)\times\nabla\phi+f(\psi)\nabla\phi$, where (r,ϕ,z) are cylindrical coordinates, $\psi(r,z)=r^2(1-r^2)(1-z^2)$ is the poloidal flux function, and $f(\psi)=rB_\phi=\alpha\psi^2$. To this we add a nonaxisymmetric vacuum field perturbation given by $\tilde{\mathbf{B}}=\nabla\tilde{\varphi}_m$ with

$$\tilde{\varphi}_m(x,y,z)=\epsilon\sin(\pi x)\sin(\pi y)\sinh(\lambda z)$$

and $\lambda=\lambda\sqrt{2}$, so that $\nabla^2\tilde{\varphi}_m=0$, i.e., $\nabla\times\tilde{\mathbf{B}}=0$, $\nabla\cdot\tilde{\mathbf{B}}=0$. On $z=0$ near the origin $\tilde{\varphi}_m\propto xy=r^2\sin(2\phi)/2$ [toroidal mode number $n=2$], and $\tilde{B}_z\sim\cosh(\lambda z)$ does not change sign in z [poloidal mode number $m=1$]. The winding number for toroidal motion relative to poloidal motion $q(\psi)$ decreases as distance from the magnetic axis increases, and the value at the magnetic axis [maximum of $\psi(r,z)$] is $q(\psi_0)=\alpha/16$. We have chosen $q(\psi_0)=0.525$, so that there is a $q=1/2$ surface in the plasma, where the $m/n=1/2$ nature of $\tilde{\mathbf{B}}$ will be resonant. The computational domain is $-1<x<1$, $-1<y<1$, $-1<z<1$, and 64 grid points are used in each of the three directions. Some field lines will exit the domain because $\tilde{B}_z\neq0$ at top and bottom, but individual field lines will not exit if they are bounded by a Kolmogorov–Arnold–Moser (flux) surface. In Fig. 1 we show the surfaces of section $y=0$ ($x-z$ or $r-z$) and $x=0$ ($y-z$ or $r-z$) for $\epsilon=2\times10^{-3}$, showing a $m=1$, $n=2$ island. The time step was $\Delta\tau=0.02$ and the period in z was about 2, giving about 100 steps per period. The maximum time was $\tau=3000$, about 1500 periods in z. In principle there is a small amount of field line chaos near the apparent hyperbolic line [at $x=0.71$, $z=-0.15$ in Fig. 1(a)], but it occupies a very small area. Fig. 1(c) shows an expansion of the scale of the volume-preserving results of Fig. 1(b), showing how well the nonchaotic field lines lie on a magnetic surface. We have also shown results using second-order Runge–Kutta (RK) for the same time step. It is clear that RK allows unphysical drift from the flux surface.

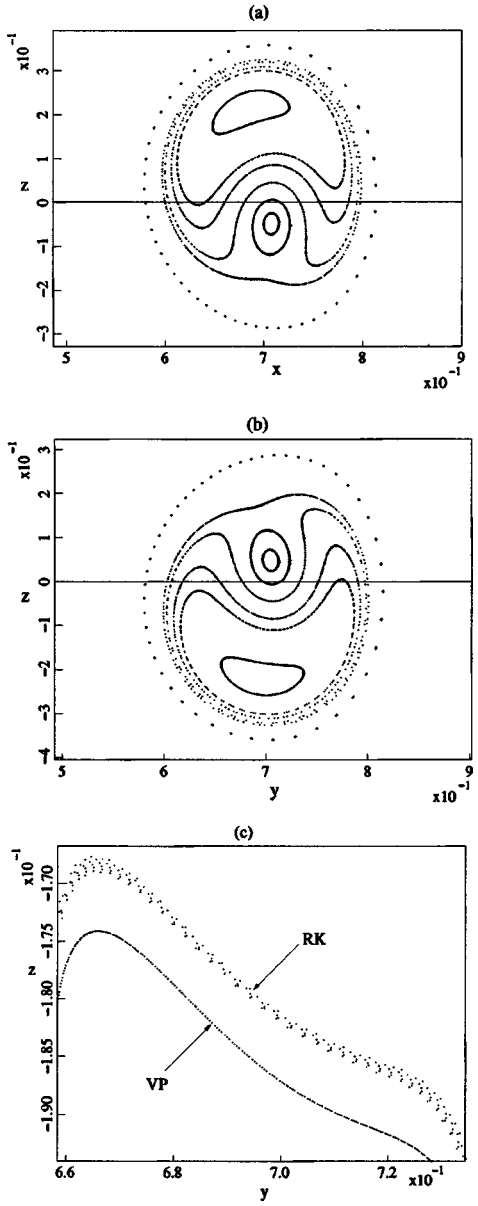

FIG. 1. Surfaces of section with the volume-preserving integrator (a) $y=0$ ($\phi=0$) and (b) $x=0$ ($\phi=\pi/2$). The plots in (a), (b) reveal a poloidal mode number $m=1$ resonance, and the apparent rotation between (a) and (b) indicates that $n=2$ is the dominant toroidal mode number. (c) is a blowup of (b) near $y=0.7$, $z=-0.18$, for the volume-preserving (VP) integrator along with results with second-order Runge–Kutta (RK), the latter offset for visibility by $\Delta z=0.005$.

For a 64^3 grid, with $\Delta t=0.08$ (about 80 time steps per oscillation period), a run to $\tau=1000$ (12 500 time steps) takes 10 CPU seconds (on a single 600 MHZ LINUX workstation processor) and about 2 s with second-order RK. This amount of time (1 ms per time step) is negligible compared to the time required to run a time step of a 3D simulation code with the same grid. For integrating flows for which $\mathbf{v}(\mathbf{x},t)$ is updated frequently, the spline setup time (0.8 s for 64^3) must be included when necessary. However, this setup time is still

very small compared with the typical CPU times for a 3D simulation code. Since the amount of CPU time used is so small in all cases, the advantage of the scheme we have outlined over those of Refs. 10 and 11 are mainly its simplicity. Also, in these references there was no mention of treating fields given on a grid.

Another issue is that of integrating fields that are very localized in space, e.g., near current sheets in magnetic reconnection. With adaptive gridding, it should be possible to resolve these scales and the field integration should proceed well. (In cases in which the localized behavior is marginally resolved, it might be advantageous to interpolate $\mathbf{A}(\mathbf{x})$ by a smoothing interpolator, e.g., least-squares tricubic splines, even though such interpolation schemes are less efficient than the tricubic splines we have used.)

Extension to curvilinear coordinates. To integrate Eq. (1) in arbitrary curvilinear coordinates ξ^i, we use contravariant components v^i such that $\mathbf{v}=v^1\nabla\xi^2\times\nabla\xi^3+v^2\nabla\xi^3\times\nabla\xi^1+v^3\nabla\xi^1\times\nabla\xi^2$. (See Ref. 12, in which the relation with the more conventional form of differential geometry is discussed.) Then the components v^i are given by $v^i=J\mathbf{v}\cdot\nabla\xi^i$, where the Jacobian $J=(\nabla\xi^1\cdot\nabla\xi^2\times\nabla\xi^3)^{-1}$. The divergence is given by $\nabla\cdot\mathbf{v}=1/J\Sigma_i\partial v^i/\partial\xi^i$, so that the requirement $\nabla\cdot\mathbf{v}=0$ implies that the formal or logical divergence $\partial v^i/\partial\xi^i$ is zero. Then Eq. (1) takes the form $d\xi^i/dt=\mathbf{v}\cdot\nabla\xi^i=v^i/J$ with v^i formally divergence-free. If the vector potential is written in covariant components $\mathbf{A}=\Sigma_i A_i\nabla\xi^i$, we have $\nabla\times\mathbf{A}=\Sigma_i\nabla A_i\times\nabla\xi^i$, which gives $\mathbf{v}=\nabla\times\mathbf{A}$ in contravariant form with $v_i=\epsilon^{ijk}(\partial A_k/\partial\xi^j)$. Thus the curl formula looks identical to that in rectangular coordinates, so that the method of obtaining A_i by line integrals (for one component $A_1=0$) is unchanged, and evaluation of the curl of the spline formulas is also unchanged. Therefore, the contravariant components ξ^i can be advanced by $d\xi^i/dt=v^i/J$ where the formal divergence of v^i is zero and the components v^i are obtained as a formal curl using the covariant components of $\mathbf{A}$. That is, the whole procedure outlined for rectangular coordinates still holds, with the exception of the $1/J$ factor in $d\xi^i/dt=v^i/J$. This factor can be absorbed by introducing the integration variable λ such that $dt/d\lambda=J$, giving $d\xi^i/d\lambda=v^i$. Different orbits integrated to the same value of λ will not end at the same time, but we keep track of time as an independent variable. A similar situation arises in compressible flows, as outlined below.

Higher dimensions. In Ref. 2 the extension to n dimensions was discussed. This approach begins with the fact that a divergence-free field in n dimensions (rectangular coordinates), with $\Sigma_i\partial v_i/\partial x_i=0$, satisfies

$$v^i=\sum_j \partial A_{ij}/\partial x_j$$

for an antisymmetric second rank tensor $\mathrm{A}(\mathbf{x})$. Then a 2D volume-preserving flow $\mathbf{v}^{(i,j)}$ is obtained for each pair (i,j), $i\neq j$ by

$$v_i^{(i,j)}=\partial A_{ij}/\partial x_j,\quad v_j^{(i,j)}=\partial A_{ji}/\partial x_i=-\partial A_{ij}/\partial x_i,$$

so that A_{ij} is a stream function for 2D flow in the x_i-x_j plane. Then a split-step scheme can be constructed by integrating separately with each flow $\mathbf{v}^{(i,j)}$ for each of the $n(n-1)/2$ independent components A_{ij}. Since these flows are area preserving in 2D (thus volume-preserving in n dimensions), either CN or ML can be used, and symmetrizing as in Eq. (10) will provide second-order accuracy. Further, a gauge transformation on A_{ij} can again reduce the number of components.

Compressible flows. For application to steady compressible flows in 3D, we wish to integrate $d\mathbf{x}/dt=\mathbf{v}(\mathbf{x})$ but with $\nabla\cdot\mathbf{v}\neq0$. Suppose that the steady flow $\mathbf{v}(\mathbf{x})$ has a density $\rho(\mathbf{x})$ which is finite, i.e., bounded away from zero and infinity $0<\rho_0<\rho<\rho_1<\infty$, and satisfies $\nabla\cdot(\rho\mathbf{v})=0$. One might integrate $d\mathbf{x}/dt=\rho(\mathbf{x})\mathbf{v}(\mathbf{x})$; the trajectory would follow the flow but be incorrectly labeled in time. However, we introduce $x_0=t$ and integrate

$$dx_0/d\lambda=\rho(\mathbf{x}),\quad dx_i/d\lambda=\rho(\mathbf{x})v_i(\mathbf{x}) \tag{13}$$

with respect to a new independent variable λ. The 4D flow $(\rho,\rho\mathbf{v})$ is also incompressible. It can therefore be integrated by the above scheme in 4D. To integrate to a fixed value of time $t=T$, one must integrate in λ until $x_0=T$, but x_0 does not affect the dynamics because it does not enter into ρ or $\mathbf{v}$. These orbits follow the flow lines correctly labeled in time because $dx_i/dt=dx_i/dx_0=v_i$. Note that to integrate a collection of orbits to $t=T$, each orbit will need to be integrated to a different value of λ. Because $(\rho,\rho\mathbf{v})$ is a steady field, the integration of Eq. (13) can be performed in a postprocessing step.

To generalize further, for a time dependent compressible flow $\mathbf{v}(\mathbf{x},t)$ with a finite conserved density $\rho(\mathbf{x},t)$ satisfying $\partial\rho/dt+\nabla\cdot(\rho\mathbf{v})=0$, $0<\rho_0<\rho<\rho_1<\infty$, we see that $(\rho,\rho\mathbf{v})$ is again solenoidal in 4D. We integrate similarly

$$dx_0/d\lambda=\rho(\mathbf{x},x_0),\quad dx_i/d\lambda=\rho(\mathbf{x},x_0)v_i(\mathbf{x},x_0).$$

In this case the variable x_0 is not passive. That is, it affects ρ and $\mathbf{v}$ and therefore the evolution of the system. In the same way as before, one can integrate each orbit to $t=T$ by integrating in λ until $x_0=T$, and then the orbits satisfy $dx_i/dt=v_i$ up to $t=T$. In this case the flow must be intergrated along with the simulation code. The numerical method outlined above is efficient enough that it typically should not affect the efficiency of the simulation code.

[1] E. Ott, *Chaos in Dynamical Systems* (Cambridge University Press, Cambridge, UK, 2002).
[2] F. Kang and W. Dao-Liu, in *Contemporary Mathematics*, edited by Z.-C. Shi and C.-C. Yang (American Mathematical Society, Providence, RI, 1994), Vol. 163, pp. 1–32.
[3] J. M. Finn and D. del Castillo-Negrete, Chaos **11**, 816 (2001).
[4] H. Goldstein, *Classical Mechanics* (Addison-Wesley, Reading, MA, 1950), p. 242.
[5] D. W. Kerst, J. Nucl. Energy, Part C **4**, 253 (1962).
[6] M. N. Rosenbluth, R. Z. Sagdeev, J. B. Taylor, and G. M. Zaslavski, Nucl. Fusion **6**, 297 (1966).
[7] J. M. Finn, Nucl. Fusion **15**, 845 (1975).
[8] A. Boozer, Phys. Fluids **26**, 1288 (1983).
[9] Y.-T. Lau and J. M. Finn, Physica D **57**, 283 (1992).
[10] G. R. W. Quispel, Phys. Lett. A **206**, 26 (1995).
[11] A. Thyagaraga and F. A. Haas, Phys. Fluids **28**, 1005 (1985).
[12] L. Chacon, Comput. Phys. Commun. **163**, 143 (2004).