ADVECT: ADVECTION OF A SCALAR FUNCTION

C. BUI [1], C. DAPOGNY [2] AND P. FREY [3]

[1] *CEA Saclay, DEN-DM2S-SFME, Gif sur Yvette.*
[2] *Laboratoire Jean Kuntzmann, CNRS, Université Joseph Fourier, Grenoble INP, Université Pierre Mendès France, BP 53, 38041 Grenoble Cedex 9, France.*
[3] *UPMC Univ Paris 06, UMR 7598, Laboratoire J.-L. Lions, F-75005 Paris, France.*

This manual presents the main features of the code `advect` for realizing the advection of a scalar function $\phi_0 : D \to \mathbb{R}$ on a computational set D, according to a velocity field $V : D \to \mathbb{R}^d$. More precisely, the following equation is solved:

$$\left\{ \begin{array}{ll} \frac{\partial \phi}{\partial t}(t,x) + V(x) \cdot \nabla \phi(t,x) = 0 & \text{on } (0,T) \times D, \\ \phi(0,x) = \phi_0(x) & \text{for } x \in D, \end{array} \right. \tag{1}$$

in any of the following situations:

- D is a 2d domain, equipped with a triangular mesh $\mathcal{T}$;
- D is a 3d domain, equipped with a tetrahedral mesh $\mathcal{T}$;
- D is a 3d surface, equipped with a surface triangulation $\mathcal{T}$, and $V : D \to \mathbb{R}^3$ is a vector field.

This operation is useful, for instance, in the practice of the level set method on a simplicial computational mesh [2].

The resolution relies on the method of characteristics. Briefly, for any given point $x \in D$, we introduce the characteristic curve $t \mapsto \chi(0,t,x)$ of V emerging from x at $t = 0$, as the solution to the ordinary differential equation

$$\left\{ \begin{array}{ll} \frac{\mathrm{d}\chi}{\mathrm{d}t}(0,t,x) = V(\chi(0,t,x)) & \text{for } t \in \mathbb{R}, \\ \chi(0,0,x) = x; \end{array} \right. \tag{2}$$

characterizing the location $\chi(0,t,x)$ at time t of the particle transported along the velocity field V which is initially placed in x. The exact solution $\phi(t,x)$ to (1) is completely determined by these curves:

$$\phi(t,x) = \phi_0(\chi(t,0,x)), \tag{3}$$

i.e. the value $\phi(t,x)$ of the considered quantity at any time t and point x equals the value of the initial function ϕ_0 at the point $\chi(t,0,x)$ where the particle located in x at time t was lying at time 0. In order to solve (1) for $\phi(T,x)$, the program `advect`

(1) Calculates, at every vertex $x \in \mathcal{T}$, the characteristic curve $t \mapsto \chi(t,T,x)$ emerging from x at time T by solving (2) with a Runge-Kutta 4 scheme,
(2) Evaluates directly the formula (3).

This code and the examples illustrating this document can be downloaded from the `github` repository

`https://github.com/ISCDtoolbox/Advection`

The underlying theory is described in the journal article [1].

Contents

---

1. Files structures

The program `advect` expects the input of three data files: one `.mesh` file for the mesh $\mathcal{T}$ of the computational domain D, and two `.sol` files describing (the values at the vertices of $\mathcal{T}$ of) the scalar function ϕ_0 to be transported and the velocity field $V : D \to \mathbb{R}^d$ driving the advection.

- A `.mesh` file contains information about a simplicial mesh (i.e. composed of triangles in 2d, tetrahedra in 3d). This format is used, in particular, by `inria` programs, the finite element software `FreeFem`[1] or the libraries `ISCDToolbox`[2] and `mmg`[3]. Such a file is organized as follows:

```
/* Header */
MeshVersionFormatted 1

Dimension
2

/* List of the vertices of the mesh: two floats in 2d (three in 3d) for the
coordinates, and an integer for a possible reference */
Vertices

3030     // Number of vertices

1 1 2
1 0.975 0
0.975 1 2
0.983333333333 0.966666666154 0
1 0.95 0
....

/* List of the elements of the mesh: three integers in 2d (four in 3d) for the
indices of the vertices, and one additional integer for a possible reference */
Triangles    // Tetrahedra in 3d

5898
900 833 899 0
834 828 770 0
769 834 770 0
900 893 834 0
...

/* Ending keyword */
End
```

Listing 1. Organization of a `.mesh` file

- A `.sol` file is attached to a `.mesh` file: `advect` uses this format to read and print information about either scalar, or vector functions defined at the vertices of a corresponding mesh. A typical `.sol` file describing a scalar function on the mesh defined in Listing 3 is organized as follows:

```
/* Header */
MeshVersionFormatted 1

Dimension
2

/* Number of vertices for supporting solution */
SolAtVertices
```

[1] https://freefem.org/
[2] https://github.com/ISCDtoolbox
[3] http://www.mmgtools.org/

```
3030

/* 1 = 1 field , 1 = scalar field */
1 1

/* List of solutions associated to the previous mesh */
0.92393
0.000270181
0.886448
0.000515695
...

/* Ending keyword */
End
```

Listing 2. Organization of a `.sol` file for scalar fields

The organization of a `.sol` file associated to a vector field defined on the mesh is almost identical:

```
/* Header */
MeshVersionFormatted 1

Dimension
2

/* Number of vertices for supporting solution */
SolAtVertices
5493

/* 1 = 1 field , 2 = vector field , even in 3d */
1 2

/* List of solutions associated to the previous mesh:
                x and y components of the field */
-2.15281916388442 4.12478858900704
-2.84889376017758 3.5971214208725
0.091380924825527 0.438646338840986
3.16916162757214 13.0128696134889
-20.8104030795421 6.61738931657041
1.54955630535398 7.43865170825501
...

/* Ending keyword */
End
```

Listing 3. Organization of a `.sol` file for a vector field

2. Description of the main commands

The execution of `advect` carries out the advection of a scalar function $\phi_0 : D \to \mathbb{R}$, according to a velocity field $V : D \to \mathbb{R}^d$, over a period of time $(0, T)$, on the computational mesh $\mathcal{T}$ of the domain D: the evolution equation (1) is solved.

The command line associated to this resolution is:

```
advect box.mesh -c phi0.sol -s vit.sol -o out.sol -dt 0.5 -nocfl
```

Here,

- The computational mesh $\mathcal{T}$ of D is supplied as the file `box.mesh`;
- The transported scalar function ϕ_0 is given as a $\mathbb{P}_1$ function on $\mathcal{T}$, by the file `phi0.sol`;
- The $\mathbb{P}_1$ vector field V is supplied by the file `vit.sol`;

- The result of the operation is stored as a $\mathbb{P}_1$ function on $\mathcal{T}$, in the `out.sol` file;
- The parameter `dt` is the final time T of advection (here equal to 0.5);
- see Section 4 for an explanation of the option `-nocfl`.

Remark 1. Depending on the computational domain D and the velocity field V supplied by the user, the equation (1) may turn out to be ill-posed! This happens for instance in the very common situation where $V(x)$ is pointing inside D at some point $x \in \partial D$: physically speaking, this means that some points in the domain are positions at time T of particles whose initial locations are “outside” D. In other words, the characteristic curve $t \mapsto \chi(t, T, x)$ travels outside D for some $t < t_0$ and no value of ϕ_0 or V is supplied in there. When such a situation is detected, `advect` extrapolates the values of the initial data ϕ_0 outside D to give a consistent value to $\phi(T, x)$.

Example 1. We advect the scalar function `carre.phi.sol` defined on `carre.mesh`, which is depicted on Fig. 1 (the latter is a level set function for a disk), according to the radial velocity field `carre.vit.sol`: the command

```
advect carre.mesh -c carre.phi.sol -s carre.vit.sol -o carre.out.sol -dt 0.5 -nocfl
```

yields the result displayed in Fig. 1.

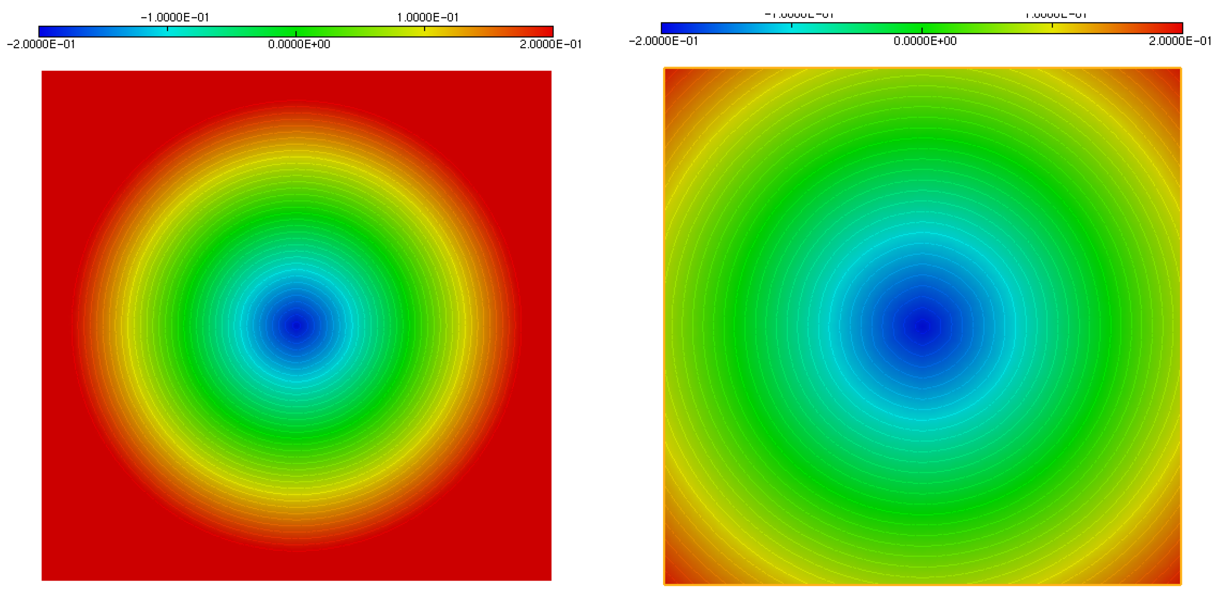

FIGURE 1. *(Right) Isovalues of the result $\phi(T, \cdot)$ of the advection of the level set function ϕ_0 contained in `carre.phi.sol` (left).*

Example 2. In three space dimensions, we advect the level set function `cube.phi.sol` for a sphere, defined on `cube.mesh`, see Fig. 2, with respect to the radial velocity field `cube.vit.sol`: the command

```
advect cube.mesh -c cube.phi.sol -s cube.vit.sol -o cube.out.sol -dt 0.3 -nocfl
```

yields the result displayed in Fig. 2.

3. Advection on surfaces

This option is still under experiment, and may contain bugs!

The program `advect` can also be used to advect a function $\phi_0 : D \to \mathbb{R}$ defined on a surface D of $\mathbb{R}^3$. The command line reads:

```
advect box.mesh -surf -c phi0.sol -s vit.sol -o out.sol -dt 0.5 -nocfl
```

This same command line allows to deal with two different situations:

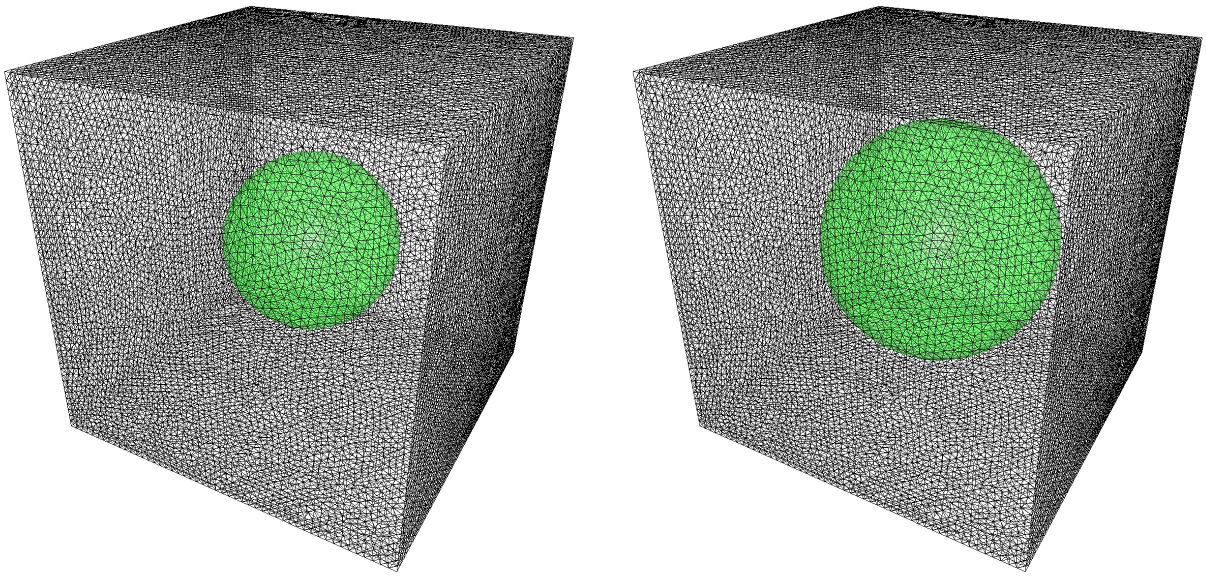

FIGURE 2. *(Right) 0 level set of the result $\phi(T,\cdot)$ of the advection of the level set function ϕ_0 contained in `cube.phi.sol` (left).*

- Either `box.mesh` is a surface triangulation, and the scalar and vector fields `phi0.sol` and `vit.sol` are defined on this surface mesh;
- Or `box.mesh` is a three-dimensional, tetrahedral mesh and the scalar and vector fields `phi0.sol` and `vit.sol` are defined on this full mesh. Then, only the surface part of `box.mesh` is considered, and only the values of `phi0.sol` and `vit.sol` at the vertices of this surface part are retained; advection is then carried out only on the surface part. The output file `out.sol` is a scalar field defined on the full tetrahedral mesh, but it is understood that only its values at surface vertices are relevant.

Note that, from the mathematical point of view, the velocity field $V : D \to \mathbb{R}^d$ should be tangential to the surface (i.e. $V \cdot n = 0$ on D, where n is any normal vector field to D) for the equation (1) to make sense in this context. However, the program `advect` can use any vector field $V : D \to \mathbb{R}^d$, being understood that only its tangential component will be retained.

Example 3. We advect the scalar function `sphere.phi.sol` defined on `sphere.mesh`, which is depicted on Fig. 3 (the latter is a level set function for a surface disk), according to the "radial" velocity field `sphere.vit.sol`: the command

```
advect -surf surf.mesh -c surf.phi.sol -s surf.vit.sol -out surf.out.sol -dt 10 -nocfl
```

yields the result displayed in Fig. 3 (right).

4. Additional options

- By default, the final time `dt` of the advection process is truncated, in order to take into account a CFL condition depending on the maximum value of the norm $|V(x)|$ over $x \in D$. This operation is relevant in certain contexts (e.g. in fluid mechanics), but it may be undesirable in other situations. It can be de-actived by using the parameter

  `-nocfl`

  on the command line.
- By default, when the velocity field $V(x)$ is entering the domain D at some points $x \in \partial D$, the values of ϕ_0 are extrapolated outside D to give a consistent value of the foot of the characteristic curve, and to the value of ϕ_0 in there, see Remark 1. This feature can be de-actived (i.e. the integration of the characteristic curve is stopped when ∂D is hit) by entering the parameter

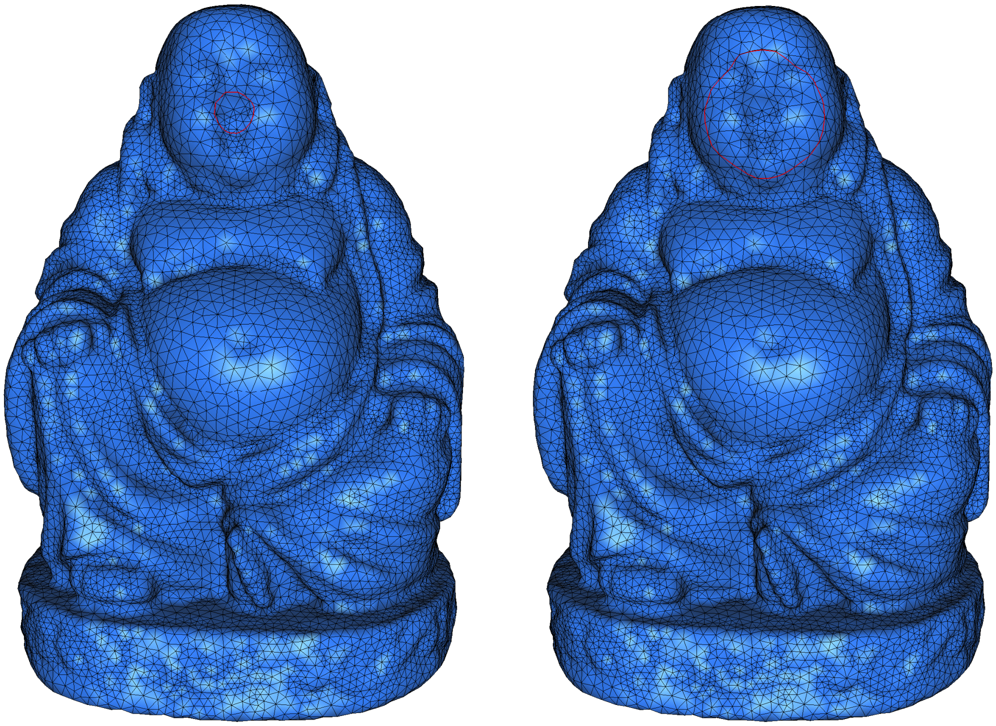

FIGURE 3. *(Right)* 0 *isovalue of the result* $\phi(T,\cdot)$ *of the advection of the level set function* ϕ_0 *contained in* ***surf.phi.sol*** *(left).*

`-noex`

on the command line.

References

[1] C. Bui, C. Dapogny and P. Frey, *An accurate anisotropic adaptation method for solving the level set advection equation*, Int. J. Numer. Methods in Fluids, Volume 70, Issue 7, pp. 899–922 (2012).

[2] J.A. Sethian, *Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry,Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, (1999).