Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Journal of Computational Physics 193 (2003) 349–355
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Short Note

A partial differential equation approach to multidimensional extrapolation

Tariq D. Aslam *

Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received 24 April 2003; received in revised form 5 August 2003; accepted 5 August 2003

Abstract

In this short note, a general methodology for multidimensional extrapolation is presented. The approach assumes a level set function exists which separates the region of known values from the region to be extrapolated. It is shown that arbitrary orders of polynomial extrapolation can be formulated by simply solving a series of linear partial differential equations (PDEs). Examples of constant, linear and quadratic extrapolation are given.

Keywords: Extrapolation; Level set; Partial differential equation

1. Introduction

In many fields of computational physics, it is often required to extrapolate a function from a region where it is known to a region where it is unknown. Examples include the Ghost Fluid Method (GFM) [2–4], where one needs to extrapolate data from a "real" region to a "ghost" region, image processing [10] and level set methods [5–7]. Here, a general methodology for multidimensional extrapolation is presented. The basic method stems from the constant extrapolation method presented in [2] and earlier in [1].

2. Mathematical formulation

2.1. Constant extrapolation

Here, we have a function, u, which is defined only in a portion of space, and we would like to extrapolate it into the remaining areas of space. We assume there exists a level set function, ψ, such that $\psi \leqslant 0$ defines the region where u is known, and $\psi > 0$ is the region where u needs to be extrapolated. Typically ψ will be

* Tel.: 1-505-667-1367; fax: 1-505-667-6372.
E-mail address: aslam@lanl.gov (T.D. Aslam).

0021-9991/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.jcp.2003.08.001

the signed distance function from the interface, Γ, of known/unknown regions of u. Here, we wish to extrapolate values of u defined at $\psi = 0$ into the region $\psi > 0$. In this section, the extrapolation of the function u is done as a constant along a normal, $\hat{n}$, which is perpendicular to the interface, Γ. This normal is defined everywhere in space by

$$\hat{n} = \frac{\vec{\nabla}\psi}{|\vec{\nabla}\psi|}. \tag{1}$$

The PDE used to achieve constant extrapolation [1,2] is

$$\frac{\partial u}{\partial t} + H(\psi)\hat{n} \cdot \vec{\nabla} u = 0, \tag{2}$$

where $H(\psi)$ is the unit Heaviside function

$$H(\psi) = \begin{cases} 1 & \text{if } \psi > 0, \\ 0 & \text{if } \psi \leqslant 0. \end{cases} \tag{3}$$

Note that the Heaviside function is used simply to not disturb the known values of u in the region $\psi \leqslant 0$. In the region $\psi > 0$, Eq. (2) has the following properties: (1) It is a linear hyperbolic PDE in u. (2) It has characteristics that are along the direction of $\hat{n}$. (3) The PDE will become steady a distance x away from Γ at time equal to that distance (because the characteristic wave speed is unity). (4) Once the PDE is steady, then we have $\hat{n} \cdot \vec{\nabla} u = 0$, which yields that u will be constant along the characteristic direction, $\hat{n}$.

So, to achieve constant extrapolation (in the normal direction), one can simply solve Eq. (2) to steady state. Furthermore, many applications, such as GFM, only require a narrow band of points near Γ to be populated, so Eq. (2) would only need to be solved for a few time steps.

2.2. Linear extrapolation

Here, we wish to use linear extrapolation in the normal direction. This is done in a series of steps. First, define the directional derivative of u in the normal direction as

$$u_n = \hat{n} \cdot \vec{\nabla} u. \tag{4}$$

Note that this is only done in the region $\psi \leqslant 0$, where u is defined. Now, this scalar function can be extrapolated in a constant manner into the region $\psi > 0$ via the PDE:

$$\frac{\partial u_n}{\partial t} + H(\psi)\hat{n} \cdot \vec{\nabla} u_n = 0. \tag{5}$$

Note that structurally, this PDE is the same as Eq. (2). Again, this is solved until steady state. Once we have the directional derivative everywhere in the region $\psi > 0$, then we can solve for the function u itself

$$\frac{\partial u}{\partial t} + H(\psi)(\hat{n} \cdot \vec{\nabla} u - u_n) = 0. \tag{6}$$

Note that this PDE also has same characteristic properties as Eq. (2), but now instead of u going to a constant along the normal direction, it will tend to have a directional derivative equal to u_n which had previously, through Eq. (5), been extrapolated itself from Γ. Note that u_n appears as a source term, similar in nature to the "reinitialization" PDE used in [9].

2.3. Quadratic extrapolation

Here, we wish to use quadratic extrapolation in the normal direction. This is also done in a series of steps. First, define the second directional derivative of u in the normal direction as

$$u_{nn} = \hat{n} \cdot \vec{\nabla}(\hat{n} \cdot \vec{\nabla} u). \tag{7}$$

Again, this is only done in the region $\psi \leqslant 0$, where u is defined. Now, this scalar function can be extrapolated in a constant manner into the region $\psi > 0$ via the PDE

$$\frac{\partial u_{nn}}{\partial t} + H(\psi)\hat{n} \cdot \vec{\nabla} u_{nn} = 0. \tag{8}$$

Note the same structure as Eq. (2). Again, this is solved until steady state. Once we have the second directional derivative everywhere in the region $\psi > 0$, then we can solve for the first directional derivative u_n via the PDE

$$\frac{\partial u_n}{\partial t} + H(\psi)(\hat{n} \cdot \vec{\nabla} u_n - u_{nn}) = 0. \tag{9}$$

Note that this PDE also has same characteristic properties as Eq. (6), but now instead of u_n going to a constant along the normal direction, it will tend to have a directional derivative equal to u_{nn} which had previously, through Eq. (8), been extrapolated itself from Γ. Once again, now that we have u_n, we can now solve

$$\frac{\partial u}{\partial t} + H(\psi)(\hat{n} \cdot \vec{\nabla} u - u_n) = 0 \tag{10}$$

to obtain the function u. Note that Eqs. (10) and (6) are identical, only the source term u_n is different.

2.4. Higher-order extrapolation

The pattern is clear to extend this method to higher-order polynomial extrapolation methods. First, one would compute the Nth-order directional derivative of u in the region $\psi \leqslant 0$. Then this would be extrapolated in a constant fashion. Then, each successive lower-order directional derivative would be integrated until u is calculated. Next, the numerical implementation is discussed.

3. Numerical implementation

Here, we will outline the numerical methods needed to perform the various extrapolation methods of the previous section. In particular, a uniform Cartesian grid is used to discretize the domain $x \in (x_{\min}, x_{\max})$, with $N_x + 1$ and $y \in (y_{\min}, y_{\max})$, with $N_y + 1$ equally spaced nodes (for simplicity it is further assumed that the grid spacing in each dimension are equal, i.e., $\Delta x = \Delta y$). For each of the PDEs being solved, a method of lines approach will be taken. In particular for the examples given in the following sections, a second-order Runge–Kutta (R–K) time integration will be used in conjunction with a second-order upwind spatial discretization. See [8] for implementation details. Some specific details for each extrapolation method are given next.

3.1. Constant extrapolation

For constant extrapolation in the normal direction, one only needs to solve Eq. (2) with u and ψ given. The normal components, n_x and n_y, are computed with standard second-order accurate central differences.

Once these are defined, then Eq. (2) is simply solved with second-order upwind (MinMod) spatial discretization and second-order R–K time discretization.

3.2. Linear extrapolation

Here, there are two PDEs to be solved. The first, Eq. (5), needs u_n as initial data. Note that since u may only be given in the region $\psi \leqslant 0$, u_n can only be computed in the region $\psi \leqslant -\Delta x$. Eq. (4), in two dimensional Cartesian coordinates becomes

$$u_n = \hat{n} \cdot \vec{\nabla} u = n_x \frac{\partial u}{\partial x} + n_x \frac{\partial u}{\partial y}, \tag{11}$$

where all derivatives are computed by second-order central differences. Now we have initial conditions for u_n in the region $\psi \leqslant -\Delta x$. Now Eq. (5) is slightly modified numerically to account for the fact that there is no well defined u_n in the region $\psi > -\Delta x$

$$\frac{\partial u_n}{\partial t} + H(\psi + \Delta x)\hat{n} \cdot \vec{\nabla} u_n = 0. \tag{12}$$

Once this PDE is solved to steady state (again using the same second-order upwind method), we then solve Eq. (6) to steady state. Note that we use the results of the previous, steady-state solution of u_n in this PDE. Also note that the Heaviside function need not be modified in this PDE, since u_n will be well set everywhere, and u is defined right up to $\psi = 0$.

3.3. Quadratic extrapolation

Here, we need u_{nn} as initial data, which in two dimensions becomes

$$u_{nn} = n_x^2 \frac{\partial^2 u}{\partial x^2} + n_x \frac{\partial u}{\partial x}\frac{\partial n_x}{\partial x} + 2 n_x n_y \frac{\partial^2 u}{\partial x \partial y} + \left(n_x \frac{\partial u}{\partial y} + n_y \frac{\partial u}{\partial x} \right) \frac{\partial n_x}{\partial y} + n_y^2 \frac{\partial^2 u}{\partial y^2} + n_y \frac{\partial u}{\partial y}\frac{\partial n_y}{\partial y}. \tag{13}$$

Note that the above equation also uses second-order central differences. In particular the cross-derivative $\partial^2 u/\partial x \partial y$ is needed. Again, if u is only given in $\psi \leqslant 0$, then we can only compute u_{nn} in the region $\psi \leqslant -\sqrt{2}\Delta x$. So, we modify Eq. (8) to

$$\frac{\partial u_{nn}}{\partial t} + H(\psi + \sqrt{2}\Delta x)\hat{n} \cdot \vec{\nabla} u_{nn} = 0 \tag{14}$$

and solve to steady state. At this point, we then compute u_n in the same fashion as in the previous section, and solve Eq. (12) to steady state for u_n, and finally solve Eq. (6) to steady state for u.

4. Numerical examples

Here, examples are given from the constant, linear and quadratic extrapolation cases. The computational domain is a square: $(-\pi, \pi) \times (-\pi, \pi)$, with a level set function $\psi = \sqrt{x^2 + y^2} - 2$. The function to be extrapolated is given initially by

$$u = \begin{cases} 0 & \text{if } \psi > 0, \\ \cos(x)\sin(y) & \text{if } \psi \leqslant 0. \end{cases} \tag{15}$$

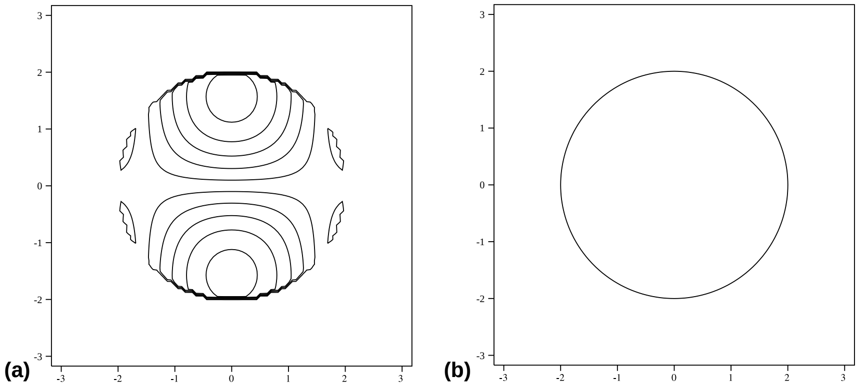

Fig. 1. (a) Initial conditions for u (0.2 incremental contours) and (b) $\psi = 0$ contour.

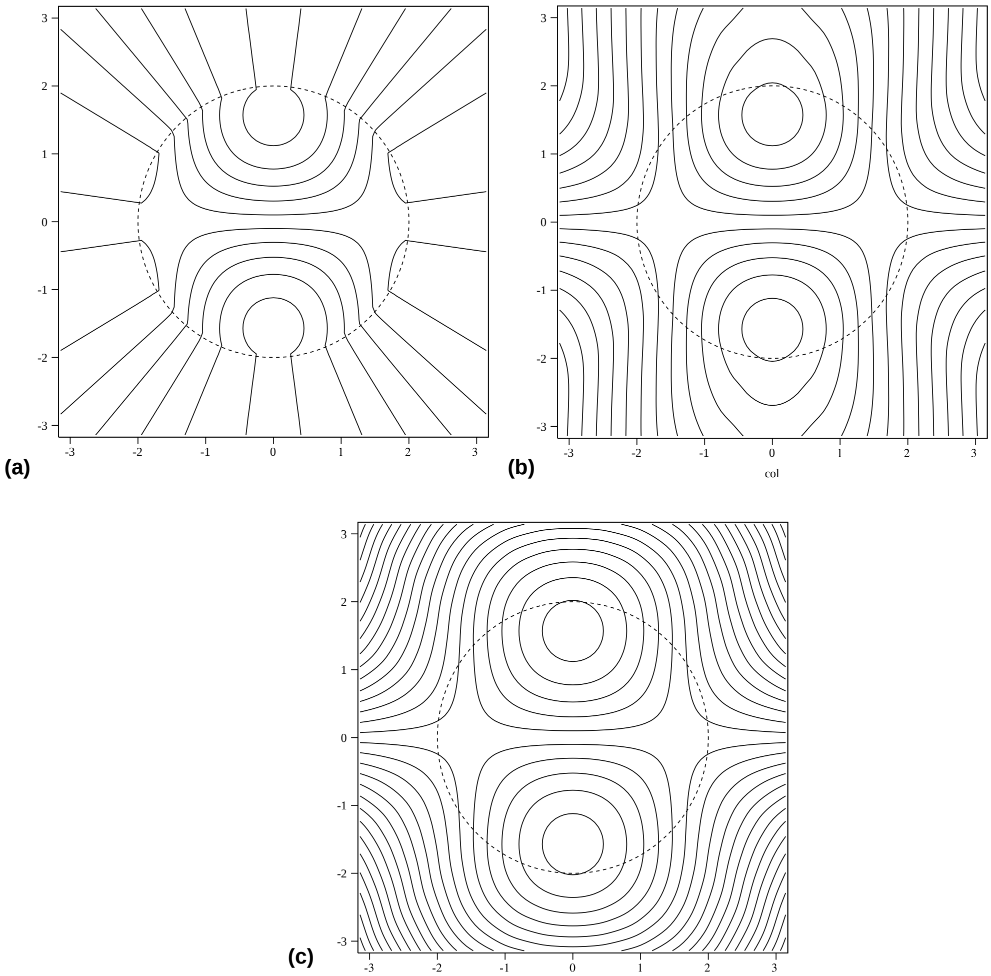

Fig. 2. (a) Constant, (b) linear and (c) quadratic extrapolation results (0.2 incremental contours); dashed contour is $\psi = 0$.

Table 1
Numerical accuracy for constant extrapolation

$\Delta x = \Delta y$	E_1	R_1	E_∞	R_∞	$E_{\text{avg-band}}$	$R_{\text{avg-band}}$	$E_{\infty\text{-band}}$	$R_{\infty\text{-band}}$
$\pi/100$	3.78×10^{-1}		3.13×10^{-2}		1.41×10^{-2}		3.13×10^{-2}	
$\pi/200$	1.75×10^{-1}	1.12	1.52×10^{-2}	1.04	6.53×10^{-3}	1.11	1.52×10^{-2}	1.04
$\pi/400$	8.94×10^{-2}	0.96	7.57×10^{-3}	1.01	3.51×10^{-3}	0.89	7.57×10^{-3}	1.01
$\pi/800$	4.50×10^{-2}	0.99	4.24×10^{-3}	0.84	1.75×10^{-3}	1.01	4.24×10^{-3}	0.84

Table 2
Numerical accuracy for linear extrapolation

$\Delta x = \Delta y$	E_1	R_1	E_∞	R_∞	$E_{\text{avg-band}}$	$R_{\text{avg-band}}$	$E_{\infty\text{-band}}$	$R_{\infty\text{-band}}$
$\pi/100$	1.00×10^{0}		1.27×10^{-1}		4.85×10^{-3}		1.90×10^{-2}	
$\pi/200$	4.37×10^{-1}	1.20	5.57×10^{-2}	1.19	1.08×10^{-3}	2.17	4.33×10^{-3}	2.13
$\pi/400$	2.09×10^{-1}	1.06	2.82×10^{-2}	0.98	2.65×10^{-4}	2.03	1.13×10^{-3}	1.94
$\pi/800$	1.04×10^{-1}	1.01	1.39×10^{-2}	1.02	6.72×10^{-5}	1.98	2.93×10^{-4}	1.95

Table 3
Numerical accuracy for quadratic extrapolation

$\Delta x = \Delta y$	E_1	R_1	E_∞	R_∞	$E_{\text{avg-band}}$	$R_{\text{avg-band}}$	$E_{\infty\text{-band}}$	$R_{\infty\text{-band}}$
$\pi/100$	2.09×10^{0}		4.27×10^{-1}		1.13×10^{-3}		4.72×10^{-3}	
$\pi/200$	9.95×10^{-1}	1.07	2.09×10^{-1}	1.03	1.44×10^{-4}	2.97	6.04×10^{-4}	2.96
$\pi/400$	4.87×10^{-1}	1.03	1.01×10^{-1}	1.05	1.84×10^{-5}	2.97	7.74×10^{-5}	2.97
$\pi/800$	2.39×10^{-1}	1.03	5.19×10^{-2}	0.96	2.28×10^{-6}	3.01	9.78×10^{-6}	2.98

See Fig. 1 for contour plots. Note the discontinuity in u in the initial conditions. Fig. 2 shows the results from each of the methods as computed on a 101×101 computational domain. Notice, in Fig. 2(a), that u is now continuous across $\psi = 0$, although there is a discontinuity in derivative at $\psi = 0$. In Fig. 2(b) the function and its derivative are continuous across $\psi = 0$. Fig. 2(c) shows the quadratic result which is also has continuity in its second derivative across $\psi = 0$. Further resolutions were conducted up to 801×801 grids, and first-order convergence, in the L_1 norm, to the exact solutions was observed (the exact solution to constant, linear and quadratic extrapolation in the normal direction can be carried out for this case due to the relatively simple $\psi = 0$ surface). The first-order nature of the convergence is intimately linked to the discontinuous behavior of $H(\psi)$. Both lower order (first-order) and higher-order (fifth-order weighted ENO) were also tested with similar errors in the L_1 norm. Even though globally only first-order results are achieved, it is usually more important, for example in GFM, to get smoothness near $\psi = 0$. Examining the error within $3\Delta x$ of $\psi = 0$, yielded higher rates of convergence. For example, the MinMod method achieves third-order rates of convergence to the exact solution in the quadratic extrapolation case, second order for the linear case and first order for the constant case. See Tables 1–3 for the global L_1 and L_∞ errors using the MinMod method, as well as average, $L_{\text{avg-band}}$ and $L_{\infty\text{-band}}$ errors in the $3\Delta x$ band of points near $\psi = 0$. The results indicate that in the narrow band near $\psi = 0$ the rates of convergence are proportional to the order of extrapolation, which is the desired goal.

As a note, if one is only interested in extrapolating into a narrow band of N_x points then solving the PDEs for $10 \times N_x$ will typically ensure that the solution is steady in the narrow band. But one may want to do checks on steadiness in the narrow band to ensure proper rates of convergence are achieved.

References

[1] T.D. Aslam, Investigations on detonation shock dynamics, PhD Thesis, University of Illinois at Urbana-Champaign, 1996.
[2] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[3] R.P. Fedkiw, T. Aslam, S. Xu, The Ghost Fluid Method for deflagration and detonation discontinuities, J. Comput. Phys. 154 (2) (1999) 393–427.
[4] F. Gibou, R. Fedkiw, High order Eulerian discretizations of the Stefan problem, J. Comput. Phys. (2003), to be submitted.
[5] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, Berlin, 2002.
[6] J.A. Sethian, Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Sciences, Cambridge University Press, Cambridge, 1996.
[7] J.A. Sethian, Fast Marching Methods and Level Set Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Sciences, Cambridge University Press, Cambridge, 1999.
[8] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[9] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[10] L.A. Vese, T.F. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, Int. J. Comput. Vis. 50 (3) (2002) 271–293.