University of Colorado at Denver and Health Sciences Center

Construction of a Level Function for Fireline Data Assimilation

Jan Mandel and Vaibhav V. Kulkarni

June 2006

UCDHSC/CCM Report No. 234

CENTER FOR COMPUTATIONAL MATHEMATICS REPORTS

Constructing a Level Function for Fireline Data Assimilation

Jan Mandel [a,*,1] and Vaibhav V. Kulkarni [b,2]

[a] *Center for Computational Mathematics, University of Colorado at Denver and Health Sciences Center, Denver, CO 80217-3364, and National Center for Atmospheric Research, Boulder, CO*

[b] *Department of Mathematical Sciences, University of Colorado at Denver and Health Sciences Center, Denver, CO 80217-3364.*

Abstract

A new method is described for determining the values of the level function from a given interface that is to be represented exactly as the zero level set. The level function is linear on boundary segments of the grid cells, which precludes the definition of the level function on the grid nodes as the signed distance from the interface. Instead, the values of the level function on a strip along the interface are found by solving a constrained least squares problem. The level function is then continued by a nearest neighbor algorithm, or a fast marching method can be used. The method was applied to representation of the burning region and of the fireline from a wildfire simulation code. Unlike other descriptions of interfaces, level set representations can be easily combined by forming linear combinations. This is useful in data assimilation methods, which modify the state of running simulation to match the data.

Key words: Level sets, least squares, ensemble Kalman filters, data assimilation, wildfire modelling, combustion, reaction front tracking, submesh approximation.

1 Introduction

Computational modelling of problems with internal interfaces, such as reaction fronts, is an important problem. If standard discretization methods are

* Campus Box 170, UCDHSC, Denver CO 80217-3364, USA

1 Supported by the NSF Grant CNS-0325314 and by NCAR Faculty Fellowship.
2 Supported by the NSF Grant CNS-0325314.

used, the requirements for accuracy dictate very fine meshes at least in the neighborhood of the interface. Therefore, it is advantageous to use submesh schemes, such as dividing a single cell into two regions with and without the reaction. The dividing line can be specified by points on the interface, called tracers. Another approach is to represent the reaction region R as the level set

$$R = \{x : f(x, y) > 0\}$$

of some function f, called a level function. The interface is then represented as the zero level set

$$C = \{x : f(x, y) = 0\}.$$

Often the level function is chosen so that it equals to the signed distance from the interface C and an equation for evolution of the interface is added to the modelled system. The fast marching method is a well known algorithm for computing the distance from the interface C on grid points. However, the fast marching method does not specify how are the values of f to be chosen on the grid nodes adjacent to the interface C so that the interface is represented as the zero level set exactly. Also, a method faster and simpler that the fast marching method, even if it may be less accurate, is of interest. These two issues are studied in this paper.

Level functions are useful also in a number of other applications, such as multi-phase flows and image processing. The monograph [1] is a comprehensive source of information on level set methods and fast marching algorithms.

Data assimilation [2] is the problem of sequential statistical estimation of a system state from given initial and boundary condition, and from data that is arriving while the simulation is running. Data assimilation methods need to modify the model state to match the data. To provide useful predictions, a data assimilation scheme needs to run faster than real time.

This work has been motivated by the problem of assimilating data into a coupled wildfire and weather model [3], which uses tracers to represent the fireline [4]. Ensemble Kalman filters [5], used in [3], run a collection of simulations, called an ensemble, and when the data arrives, a new ensemble is created by forming linear combinations of ensemble states. However, it is not possible to create meaningful linear combinations of tracers. Therefore, one possibility to modify the system state is to translate the tracers into a level function, form linear combinations of the level functions, and then translate the resulting level functions back into tracers. The translation of tracers to level function is considered in this paper; the inverse translation is basically just finding the zeros of the level function on the grid lines.

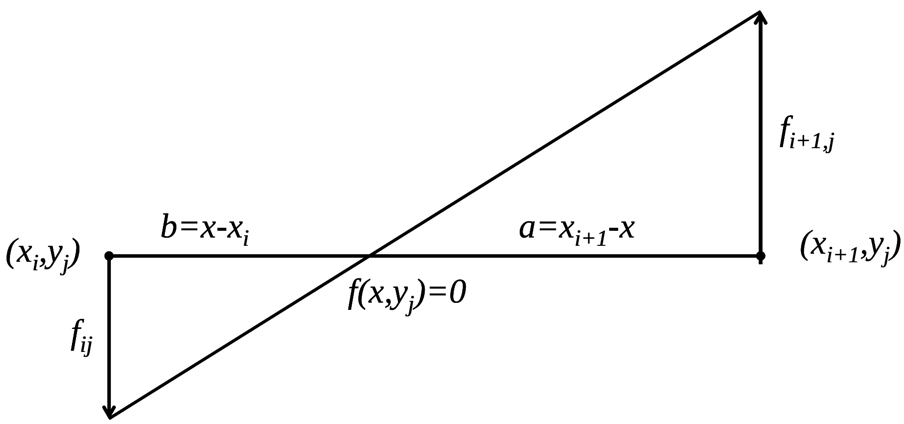

Fig. 1. Level function f on one cell boundary segment

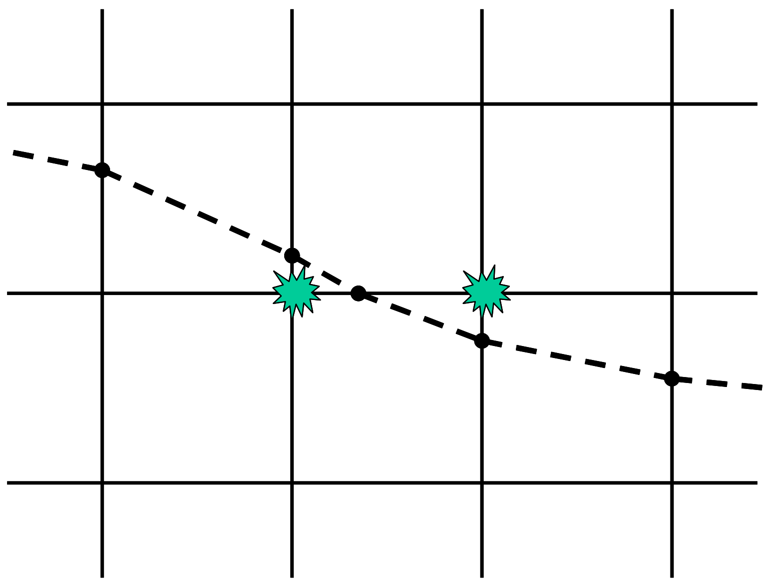

Fig. 2. The level function f cannot be defined at marked nodes consistently either as Euclidean distance or as distance along the gridlines.

2 Construction of the Level Set Function near the Interface

Suppose we are given a uniform rectangular mesh with spacing (h, k) and nodes

$$(x_i, y_j) = (hi, kj)\,, \quad i = 1, \dots, m, \quad j = 1, \dots, m.$$

Grid cell (i, j) is defined by $\{(hu, kv) : i \le u \le i+1,\ j \le v \le j+1\}$. The boundaries of grid cell consist of the boundary segments (hi, kv), $j \le v \le j+1$ and (hu, kj), $i \le u \le i+1$. Let C be a simple closed curve such that the intersection of C with each grid cell is a straight segment(Fig. 1). We wish to construct a function f defined by its values $f_{ij} = f\,(x_i, y_j)$ and piecewise linear on the grid lines, such that

(1) f is linear on each grid cell boundary segment,
(2) for all (x, y) on grid cell boundary, $f(x, y) = 0$ if and only if $(x, y) \in C$,
(3) $f_{ij} > 0$ if (x_i, y_j) is inside and $f_{ij} < 0$ if (x_i, y_j) is outside the region defined by C,
(4) $f_{ij} \approx \pm \operatorname{dist}((x_i, y_j), C)$,

where

$$\operatorname{dist}((x_i, y_j), C) = \min \{\|(x, y) - (x_i, y_j)\| : (x, y) \in C\}$$

is the Euclidean distance from the curve C, with the $+$ sign if (x_i, y_j) is inside the region encircled by C and $-$ otherwise.

Conditions 1 and 2 imply an equation of the form

$$af_{ij} - bf_{i,j+1} = 0 \tag{1}$$

for each horizontal grid cell boundary segment intersected by C, and similarly for vertical segments. Thus, the location of the intersection of the curve C with a grid cell boundary segment dictates the ratio of the values of the function f at the endpoints of the segment. Clearly, setting f_{ij} to the Euclidean distance does not in general satisfy the fixed proportions (1), and setting f_{ij} to the distance from the zero of f measured along the grid boundary segments cannot satisfy (1) either because of inconsistent requirements when the curve C cuts across the corner of a grid cell (Fig. 2).

Let us now construct f_{ij} on a strip of nodes nodes adjacent to C. Let N be the set of all corners of grid cells that are intersected by C and f_a the vector $(f_{ij})_{(i,j)\in N}$ for some ordering of N, and B be the matrix with one row for each grid cell boundary segment that intersects C. Therefore, $Bf_a = 0$ is the collection of all constraints of the form (1). Let d be the vector of signed distances from C,

$$d = \pm \left(\operatorname{dist}((x_i, y_j), C)\right)_{(i,j)\in N},$$

with the same sign as in the condition 4 above. Then, on the strip, the condition 4 can be written as $f_a \approx d$, and we construct f on the strip by satisfying this condition in the least square sense.

Algorithm 1 *(1) When C passes through a node (x_i, y_j), put $f_{ij} = 0$.*
(2) Find f_a by solving the constrained least squares problem

$$\|f_a - d\|^2 \to \min, \quad \text{subject to } Bf_a = 0, \tag{2}$$

or, equivalently, from the linear system

$$\begin{bmatrix} I & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} f_a \\ \lambda \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix} \tag{3}$$

with Lagrange multipliers λ.

Because the norm is strictly convex, the minimization problem (2) has a unique solution f_a and thus (3) determines f_a uniquely. The matrix B is very sparse, with at most 2 nonzeros per row. Each column of B corresponds to one value f_{ij}, $(i, j) \in N$, and it has one nonzero when the node (x_i, y_j) is the endpoint of only one grid cell boundary segment that intersect C, and two nonzeros when the node (x_i, y_j) is the endpoint of two such segments. Clearly, if there exists at least one node such that the former case occurs, the matrix B has linearly independent rows, the matrix BB^T is regular, and one gets

$$Bf_a + BB^T\lambda = Bd,$$
$$\lambda = \left(BB^T\right)^{-1} Bd,$$

so

$$f_a = d - B^T\lambda = d - B^T \left(BB^T\right)^{-1} Bd. \tag{4}$$

3 Continuing the Level Set Function away from the Interface

Once the values of the level function f on the nodes adjacent to the interface C are known, the level function can be built recursively. Perhaps the simplest and least expensive is the followig simple recursive construction. The node (x_i, y_j) is called a neghbor of the node $(x_{i'}, y_{j'})$ if $(i, j) \neq (i', j')$, $|i - i'| \leq 1$, and $|j - j'| \leq 1$. The following algorithm continues building the level function inside the region given by the curve C by setting f at all neighbors of the nodes adjacent to C, then on all second neighbors, e.t.c.

Algorithm 2 *(1) Set L_0 to be the set of all nodes (x_i, y_j) such that $f_{ij} \geq 0$ was set by the method in Sec. 2. Set $k = 1$.*

(2) Set L_k to be the set of all neighbors of nodes in L_{k-1} that are on the same side of C as L_{k-1} and such that the value of f was not set yet.

(3) If $L_k = \emptyset$, stop.

(4) For all nodes $(x_i, y_j) \in L_k$, set

$$f_{ij} = \min_{\substack{(x_{i'}, y_{j'}) \in L_{k-1} \\ |i-i'| \leq 1 \; |j-j'| \leq 1}} f_{i'j'} + \|(x_{i'}, y_{j'}) - (x_i, y_j)\| \tag{5}$$

(5) Set $k = k + 1$ and go to 2.

Continuing the level function outside of the region enclosed by C is done in the same way starting from nodes such that $f_{ij} \leq 0$ and replacing (5) by

$$f_{ij} = - \min_{\substack{(x_{i'}, y_{j'}) \in L_{k-1} \\ |i-i'| \leq 1 \; |j-j'| \leq 1}} -f_{i'j'} + \|(x_{i'}, y_{j'}) - (x_i, y_j)\|.$$

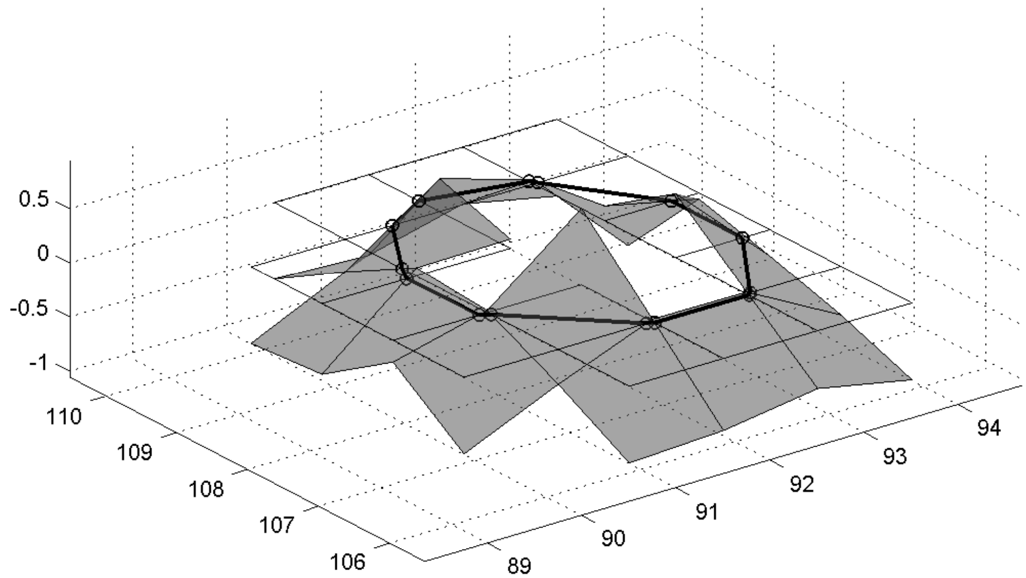

Fig. 3. The level function on a strip along the fireline.

Algorithm 2 solves approximately the least distance problem

$$f(x,y) = \pm \operatorname{dist}\left((x,y), C\right), \tag{6}$$

by measuring the distance along the grid lines and the diagonals of the grid cells only. This is significantly less accurate but much faster and simpler than converting the problem (6) to the equivalent hyperbolic eikonal equation

$$\|\nabla f\| = 1, \quad f = 0 \text{ on } C$$

and solving it by a fast marching method [1].

4 Numerical Results and the Wildfires Application

Sparse matrix operations in (4) were implemented by calls to the Sparse Matrix Multiplication Package (SMMP) [6] and the Yale Sparse Matrix Package (YSMP) [7]. Algorithm 2 was implemented by a modification of the routine ROOTLS from SPARSPAK [8].

The curve C was obtained as the fireline from the Coupled Atmosphere-Wildland Fire-Environment model (CAWFE) [9,4]. The fireline is approximated as a straight line in each grid cell and it is encoded by the so-called tracers [4], which are 4 points located in each grid cell. The tracers

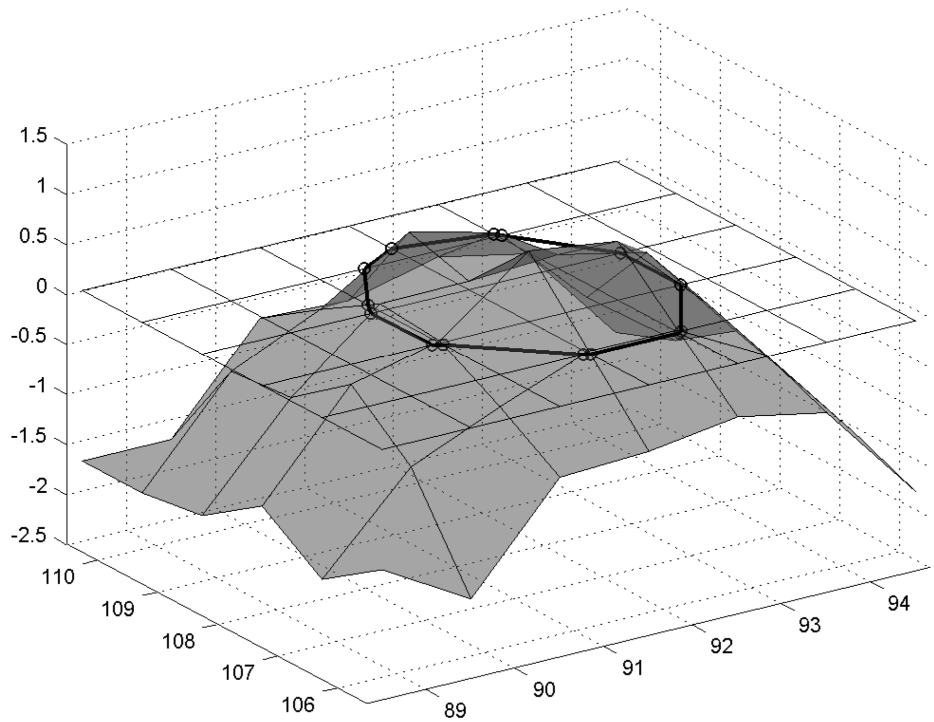

Fig. 4. The level function continued away from the fireline.

are meant to define a quarilateral burning region within the cell, with some exceptions, such as when the fireline crosses two adjacent grid cell boundary segments so that it cuts off a corner; the tracer scheme is quite complicated with some special cases that can be understood only by perusing the source code. So, the first step was to convert the tracers to a data structure that consists of the coordinates of the two ends of the fireline segment within the cell and a flag to indicate if the cell is burning or not. We have then applied the method described in Sec. 2 and 3 to obtain the level function.

The level function on nodes adjacent to the fireline C is shown in Fig. 3. The continuation away from the strip along the fireline is in Fig. 4. Note the directional preference of the level function, which is due to the fact that the distance was measured only along the grid lines and the grid cell diagonals.

5 Acknowledgements

The authors would like to thank Janice Coen for providing the CAWFE code, and Craig Douglas for providing interfaces to the CAWFE, SMMP, and YSMP codes, and for setting up a collaborative software development environment.

References

[1] J. A. Sethian, Level set methods and fast marching methods, 2nd Edition, Vol. 3 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 1999.

[2] E. Kalnay, Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, 2003.

[3] J. Mandel, L. S. Bennethum, M. Chen, J. L. Coen, C. C. Douglas, L. P. Franca, C. J. Johns, M. Kim, A. V. Knyazev, R. Kremens, V. Kulkarni, G. Qin, A. Vodacek, J. Wu, W. Zhao, A. Zornes, Towards a dynamic data driven application system for wildfire simulation, in: V. S. Sunderam, G. D. van Albada, P. M. A. Sloot, J. J. Dongarra (Eds.), Computational Science - ICCS 2005, Vol. 3515 of Lecture Notes in Computer Science, Springer, 2005, pp. 632–639.

[4] T. L. Clark, J. Coen, D. Latham, Description of a coupled atmosphere-fire model, Intl. J. Wildland Fire 13 (2004) 49–64.

[5] G. Evensen, The ensemble Kalman filter: Theoretical formulation and practical implementation, Ocean Dynamics 53 (2003) 343–367.

[6] R. E. Bank, C. C. Douglas, SMMP: Sparse matrix multiplication package, Advances in Computational Mathematics 1 (1993) 127–137.

[7] S. C. Eisenstat, M. C. Gursky, M. H. Schultz, A. H. Sherman, Yale sparse matrix package I: The symmetric codes, Int. J. Numer. Methods Engrg. 18 (1982) 1145–1151.

[8] A. George, J. W. H. Liu, Computer solution of large sparse positive definite systems, Prentice-Hall Inc., Englewood Cliffs, N.J., 1981.

[9] J. L. Coen, Simulation of the Big Elk Fire using using coupled atmosphere-fire modeling, International J. of Wildland Fire 14 (1) (2005) 49–59.

CENTER FOR COMPUTATIONAL MATHEMATICS REPORTS

University of Colorado at Denver and Health Sciences Center
P.O. Box 173364, Campus Box 170
Denver, CO 80217-3364

Fax: (303) 556-8550
Phone: (303) 556-8442
http://www-math.cudenver.edu/ccm

222 I. Lashuk, M. E. Argentati, E. Ovchinnikov and A. V. Knyazev, "Preconditioned Eigensolver LOBPCG in hypre and PETSc." September 2005.

223 A. V. Knyazev and M. E. Argentati, "Majorization for Changes in Angles Between Subspaces, Ritz values, and graph Laplacian spectra." September 2005.

224 D. W. Dean, "A Semimartingale Approach For Modeling Multiphase Flow In Heterogeneous Porous Media." November 2005.

225 D. W. Dean, T.H. Illangasekare and T.F. Russell, "A Stochastic Differential Equation Approach For Modeling Of NAPL Flow In Heterogeneous Porous Media." November 2005.

226 J. Mandel and B. Sousedík, "Adaptive Coarse Space Selection in the BDDC and the FETI-DP Iterative Substructuring Methods: Optimal Face Degrees of Freedom." November 2005.

227 J. Mandel and B. Sousedík, "Adaptive Selection of Face Coarse Degrees of Freedom in the BDDC and the FETI-DP Iterative Substructuring Methods." November 2005.

228 Craig C. Douglas, Jonathan D. Beezley, Janice Coen, Deng Li, Wei Li, Alan K. Mandel, Jan Mandel, Guan Qin, and Anthony Vodacek, "Demonstrating the Validity of a Wildfire DDDAS." February 2006.

229 Lynn Schreyer Bennethum, "Flow and Deformation: Understanding the Assumptions and Thermodynamics, Updated." February 2006.

230 Lynn Schreyer Bennethum, "Compressibility Moduli for Porous Materials Incorporating Volume Fraction with Detailed Computations." February 2006.

231 Jan Mandel, "Efficient Implementation of the Ensemble Kalman Filter." May 2006.

232 Jan Mandel and Jonathan D. Beezley, "Predictor-Corrector Ensemble Filters for the Assimilation of Sparse Data into High Dimensional Nonlinear Systems." June 2006.

233 Jan Mandel, Lynn S. Bennethum, Jonathan Beezley, Janice L. Coen, Craig C. Douglas, Leopoldo P. Franca, Minjeong Kim, and Anthony Vodacek, "A Wildfire Model with Data Assimilation." June 2006.

234 Jan Mandel and Vaibhav Kulkarni, "Construction of a Level Function for Fireline Data Assimilation." June 2006.