ISBN 978-981-07-8391-4
Proceedings of 2013 3rd International Conference on Computer Engineering and Bioinformatics (ICCEB 2013)
Bangkok, Thailand, November 23-24, 2013, pp. 37-43

Automatic Detection and Segmentation of Brain Tumors using Binary Morphological Level Sets with Bounding Box

Jayalaxmi S. Gonal[1+] , Vinayadatt V. Kohir [2]

[1] BLDEA's Engineering College, Bijapur, Karnataka, India

[2] PDA Engineering College, Gulbarga, Karnataka State, India

Abstract. In this paper, a method for automatic detection and accurate segmentation of brain tumor from magnetic resonance imagery (MRI), with no user intervention has been developed. It first detects the brain tumor location using bounding box algorithm; this performs crude segmentation of tumors in brain images. To refine this initial segmentation, we propose Binary Morphological Level Sets coupled with Bounding Box algorithm. Binary Morphological Level Sets evolves level set functions using simple binary morphological operations. This morphological level set method is capable of bidirectional evolution of level set functions, i.e., the interface of a level set function can either expand or shrink toward the object boundary. Our detection and segmentation algorithm can play a useful role in indexing and storage of MRI data and as an initial step toward accurate tumor boundary delineation.

Keywords: Bounding box algorithm, Level set method, Morphological image segmentation.

1. Introduction

Automated tumor segmentation from brain MR (magnetic resonance) images can play a significant role in medical research and clinical practice. Radiologists currently segment patients' tumors by hand on MR images before applying a treatment such as radiation therapy. This manual segmentation process is laborious and expensive. Unless these images are segmented, however, it is difficult to retrieve the relevant images, for analysis – e.g., allowing a clinician to use tumor location to retrieve historical cases relevant to the diagnosis and treatment of cancers in new patients.

Automatically segmenting tumors in brain MRI is an extremely challenging task. For an account of this topic from image analysis and machine learning perspective, see [7]. There are many challenges here, many corresponding to the challenge of incorporating domain knowledge. On the computational front, pixel labeling algorithms, such as support vector classifiers, which learn local features (computed within a window around a pixel), are not adequate to segment brain abnormality (tumor, edema, etc.). On the other hand, incorporation of global region-based features is non-trivial and computationally intensive [2]. Attempts have also been made to incorporate pixel classification and region similarity (see [2], [5]).

In this paper we propose Binary Morphological Level Sets Bounding Box algorithm, a fast method for locating a bounding box around the region of abnormality i.e. tumor and then performing its accurate segmentation. The proposed algorithm is described in two phases:First-phase is detection of the brain tumor location using Bounding Box algorithm; this performs crude segmentation of tumors in brain images. Bounding Box method exploits the facts that a normal brain structure is symmetric – the left part and the right part can be divided by an axis of symmetry, and abnormalities (tumors, edema) typically disturb this symmetry. Second-phase is refining of this initial segmentation using Binary Morphological Level Sets algorithm. Binary Morphological Level Sets evolves level set functions using simple binary morphological operations. This morphological level set method is capable of bidirectional evolution of level set functions,

i.e., the interface of a level set function can either expand or shrink towards the object boundary. We initialize the level set function within the bounding box that circumscribe the tumor, and allows the level set

Corresponding author. Tel.: + 919880061474; fax: +(p.
E-mail address: (please specify).

function to evolve till it sits on the boundary of the tumor.

The paper is organized as follows: the concept of Bounding Box algorithm is explained in section 2.The background of Level sets is dealt in the section 3. Morphological Level Sets are explained in section 4. Experiment results and conclusion are seen in section 5 correspondingly.

2. Concept of Bounding Box algorithm

The basic principle behind Bounding Box (BB) is a change detection principle, where a region of change (D) is detected on a test image (I), when compared with a reference image (R) as shown in Fig. 1(a) after finding the axis of symmetry on an axial MR slice, the left (or the right) half serves as the test image I, and the right (or the left) half supplies as the reference image *R*. The region of change D here is restricted to be an axis-parallel rectangle, which essentially aims to circumscribe the abnormality.

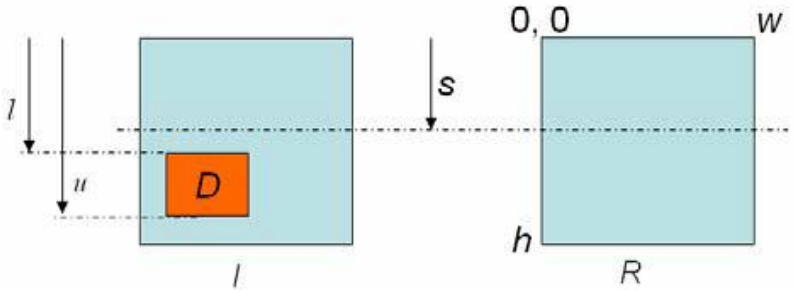


Fig. 1(a): Finding *D* from image *I*, using a reference image *R*,

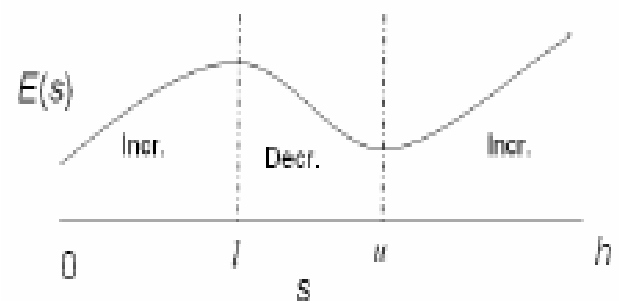


Fig. 1(b): A typical score function plot.

Consider Fig. 1(a), with a horizontal dotted line drawn at a distance s from the top of the images. Now consider the regions: A(s) = [0, w] × [0, s], and B(s) = [0, w] × [s, h], where w and h are respectively the width and the height of both the images I and R. Thus A(s) and B(s) are the portions of image domain respectively above and below the aforementioned horizontal line. Let E(s) denote the following score function:

$$E(s) = \langle \sqrt{P_I^{A(s)}}, \sqrt{P_R^{A(s)}} \rangle - \langle \sqrt{P_I^{B(s)}}, \sqrt{P_R^{B(s)}} \rangle$$

where, P's denote normalized intensity histograms (probability mass functions of image intensities), the subscripts indicate whether this histogram is of image I or of template R, and the superscripts denote whether this histogram is computed within the region A(s) or within the region B(s). For example, $P_I^{A(s)}$ denotes the normalized intensity histogram of image I within A(s). $\langle X, Y \rangle$ denotes the inner product (i.e., sum of element-wise product) between two vectors X and Y. The inner product between square roots of two normalized histograms is known as Bhattacharya coefficient (BC), [4] which is a real number between 0 and 1 that measures the correlation between two histograms. When two normalized histograms are identical, their BC value is 1; whereas when the histograms are completely different, their BC value is 0.

Note that the score function E(s) measures the difference of correlations between the upper histograms and the lower histograms. We therefore expect a high score when the upper histograms match very well, while the lower histograms have high mismatch. On the other hand, a low value of E(s) denotes a low correlation between upper histograms, and a high correlation between lower histograms. Based on these observations, we note that a plot of E(s) v/s s should look like the plot shown in Fig. 1(b). The important

observations in Fig. 1(b) are that the plot has three distinct regions – increasing, decreasing, then increasing – where the decreasing segment begins at l and ends at u, where l and u respectively denotes the lower and the upper bound for the rectangular region D, measuring from top of the image. In fact we can prove these statements rigorously with some mild assumptions about the data, i.e., about the image I and the template R. Essentially, we require that the correlation between the image histogram outside D and the template histogram is much larger than that between the image histogram inside D and the template histogram:

$$\text{(i)} \quad \langle \sqrt{P_I^{A(s)\cap D}}, \sqrt{P_R^{A(s)}} \rangle \ll \langle \sqrt{P_I^{A(s)\setminus D}}, \sqrt{P_R^{A(s)}} \rangle;$$

$$\text{(ii)} \quad \langle \sqrt{P_I^{B(s)\cap D}}, \sqrt{P_R^{B(s)}} \rangle \ll \langle \sqrt{P_I^{B(s)\setminus D}}, \sqrt{P_R^{B(s)}} \rangle.$$

Without these assumptions the abnormal region D would look like rest of the image I, and it would be difficult (if not impossible) to detect D then. Our task is to find the maximum and the minimum points of the plot corresponding to the lower and the upper bounds of region D Fig. 1(b). Thus, algorithmically, a vertical sweep of the image I and R finds the upper and lower bounds of D. Similarly, a horizontal sweep and a similar plot of the score function help find out the left and the right bound for D. The plots from this procedure appear in Fig. 2. The upper left picture shows a brain MRI with an abnormality on the left of the image. In general, we assume the input image is an axial slice, which is close to being vertical. We first detect the skull boundary by active contour algorithm [8]. Next an ellipse is fitted to the skull boundary, from which we extract the line of symmetry (shown on the upper right picture of Fig. 2. Then we treat I as the image formed by the portion to the left of the line of symmetry and R as the portion of the image to right of the line of symmetry after taking a reflection. Next, a vertical sweep and a horizontal sweep produce the score plots respectively shown on the lower left and lower right pictures in Fig. 2. The maxima and the minima are detected from these two plots, and the corresponding bounding box is overlaid on the upper left picture of Fig. 2.

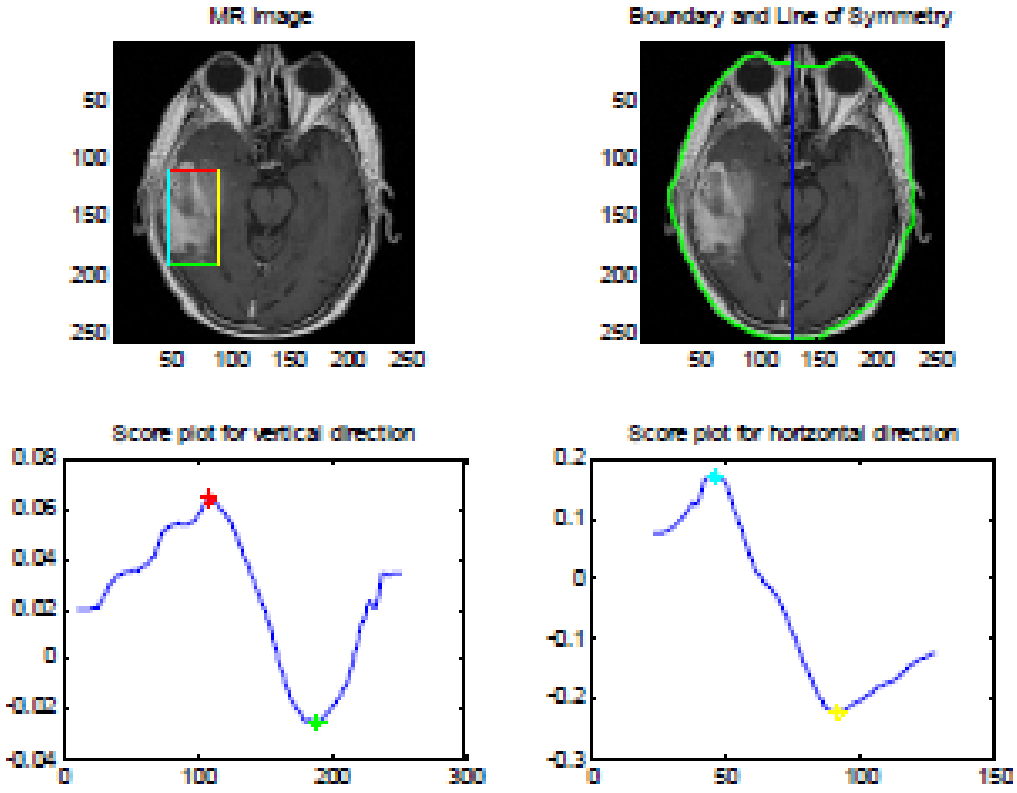


Fig. 2: Finding a bounding box on brain MRI.

3. Background of Level Sets

Level Sets: The main idea behind the level set formulation is to represent an interface Γ bounding a

possibly multiply connected region $\mathbf{R}^n$ in by a Lipschitz continuous function ϕ, changing sign at the interface. This is normally done by requiring to be the signed distance function ,which is defined as

$$\begin{cases} \phi(x) > d(\Gamma, x), & if\ x\ is\ inside\ \Gamma \\ \phi(x) = 0, & if\ x\ is\ at\ \Gamma \\ \phi(x) < d(\Gamma, x), & if\ x\ is\ outside\ \Gamma \end{cases} \quad (1)$$

where, d(Γ, x) denotes Euclidian distance between x and Γ . We emphasize that requiring equation (1) is a technicality to prevent instabilities in numerical implementations. Having defined the level set function as in equation (1), there is an one to one correspondence between the curve and the function.

A re-distance procedure is usually needed to keep the level set function to be a signed distance function during its evolution. However, the re-distance procedure is an expensive operation, which prevents the level set methods from being used in real time applications.

Binary Level Sets: Recently, in order to improve the computational efficiency of the level set methods, the signed distance function is replaced by the following binary level set function,

$$\begin{cases} \phi(\mathrm{x}) = 1, & \text{if x is inside } \Gamma \\ \phi(\mathrm{x}) = -1, & \text{if x is outside } \Gamma \end{cases} \quad (2)$$

Therefore, the re-distance procedure is no longer needed by the binary level set function, and the expensive computational cost can be saved. For two-phase image segmentation problem, the binary level set function formulated by equation (2) can use its interface Γ to partition the image domain Ω into two sub-domains, which are inside and outside the interface, respectively. Given a gray-value image I: Ω→R+, we assume that image I can be approximated by a binary function,

$$u(\phi, c_1, c_2) = \frac{c_1}{2}(1+\phi) + \frac{c_2}{2}(1-\phi) \quad (3)$$

where, c_1 and c_2 are two constants. As pointed out in the papers [6, 7], the problem of two-phase image segmentation can be modeled to minimize the energy functional,

$$E(\phi, c1, c2) = \frac{1}{2}\int_\Omega |u(\phi, c_1, c_2) - I|^2 dx + \beta \int_\Omega |\nabla \phi| dx, \quad (4)$$

subject to

$$\phi^2 = 1 \quad (5)$$

where, $\beta\ is$ non-negative parameter. The first term in equation (4) measures how well the function u approximates I and the second term measures the length of the interface Γ. It is difficult to directly solve the constrained minimization problem given by equation (4) and (5).Tai suggested an efficient level set method to obtain approximate solution for optimal problem. The method is an iterative procedure, in which each iteration contains three steps:

First, evolve the function ø by

$$\frac{d\phi}{dt} = \beta \nabla \frac{\nabla \phi}{|\nabla \phi|} - [u(\phi, c_1, c_2) - I] \frac{\partial u}{\partial \phi} \quad (6)$$

which is obtained by minimizing the energy functional E in equation (5) with respect to ø without considering the constraint equation (6).

Then, update the parameters $c1$ and $c2$, respectively, by

$$c1 = \int_\Omega \frac{I(1+\phi)\,dx}{(1+\phi)\,dx}, \quad c1 = \int_\Omega \frac{I(1-\phi)\,dx}{(1-\phi)\,dx} \quad (7)$$

Finally, compulsively set as a binary image, i.e., let $\phi = 1$ if $\phi \geq 1$ and $\phi = -1$ if $\phi < 1$.

Unfortunately, their level set method loses the curve evolution property, i.e., the sign change of the binary level set function value is no longer restricted to the neighborhood of the interface when the binary level set function evolves.

The curve evolution property is preserved by Zhang's level set method [9], which evolves the binary level set function by the basic morphological operations. However, in Zhang's level set method, the interface of the level set function can only evolve inward or outward, which requires the initial interface to be set completely inside or outside the object boundary. To overcome this problem, we tried a novel morphological level set method in which the interface can evolve bi-directionally.

4. Morphological Level Sets

The main principle underlying our method is that an annular region along the interface of the binary level set function is built by the dilation and erosion operations, and then the sign of the level set function value changes only in the annular region to form a new interface for the level set function. In accordance with this principle, the interface iteratively evolves until it has converged.

Note that the iterative procedure proceeds until the evolution of the level set function has converged or the number of iterations has reached. The implementation detail of the proposed algorithm is explained in the following steps:

Implementation steps of the algorithm:

Step 1: Initialize a binary level set function ϕ, and let $A = (\phi + 1)/2$.

Step 2: Build an annular region along the interface by $E = [(A \oplus B) (A \ominus B)]$.

Step 3: Compute the parameters c1 and c2 according to equation (7).

Step 4: Evolve the function ϕ, according to equation (6).

Step 5: Update the speed image by V = 1 if $\phi \geq 0$, V = 0 if $\phi < 0$.

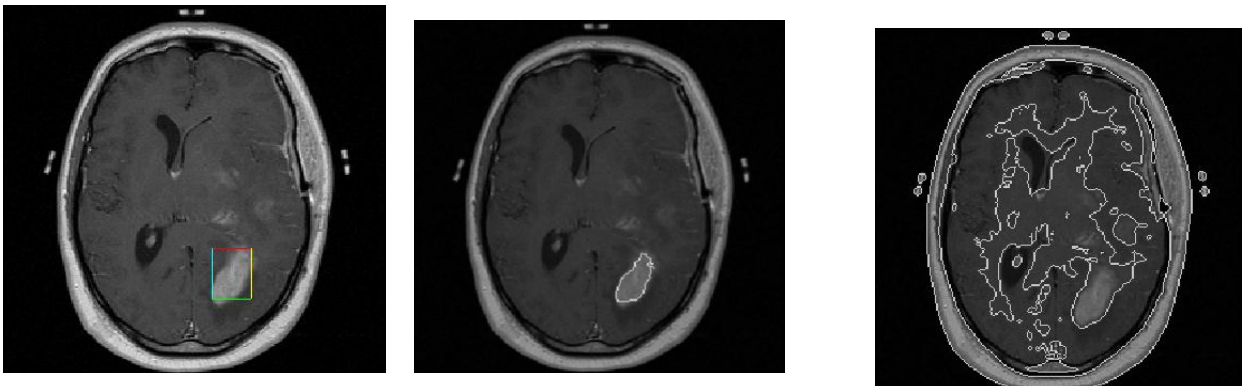
Step 6: Perform D = E AND V , and then let A= $\left(A \oplus B\right) \cup D$ *and* $\phi = 2A - 1$.

5. Experimental Results

In this section we discuss the outcomes of the algorithm. We have run the algorithm on T1-C MRIs for segmentation of tumors from MR images. Implementation was done in MATLAB 7 according to the steps as shown in block diagram below. All the experiments were run on a 1.5GHz PC with 2GB Memory.

5.1. Comparison of tumor segmentation results of BMLS-BB with BMLS

As mentioned before, the proposed bounding box algorithm only provides a rough estimate of the abnormal region. However, after finding the bounding box, we can fine tune the segmentation boundary as shown in fig. 4. While in Fig. 4(b) the segmentation boundary is confined to the correct region of abnormality, in Fig. 4(c) spurious segmentation boundaries are created.



(a) (b) (c)

Fig. 4: Brain MRI with different algorithms. (a) Bounding box, (b) BMLS algorithm within Bounding Box and (c) BMLS on the entire image.

5.2. Comparison of Tumor Segmentation Results of BMLS-BB algorithm with Chan–Vese and Graph– cut methods within bounding box

The output of this work is 2D contour fitted accurately on the boundary of tumor and also necrosis within the tumor, as shown in Fig. 5(b). This method performs tumor segmentation more precisely than many recent popular methods like, Chan-Vese algorithm, Graph-cut methods within bounding box, as seen shown in Fig. 5(b), 5(c) and 5(d) in results and discussion.

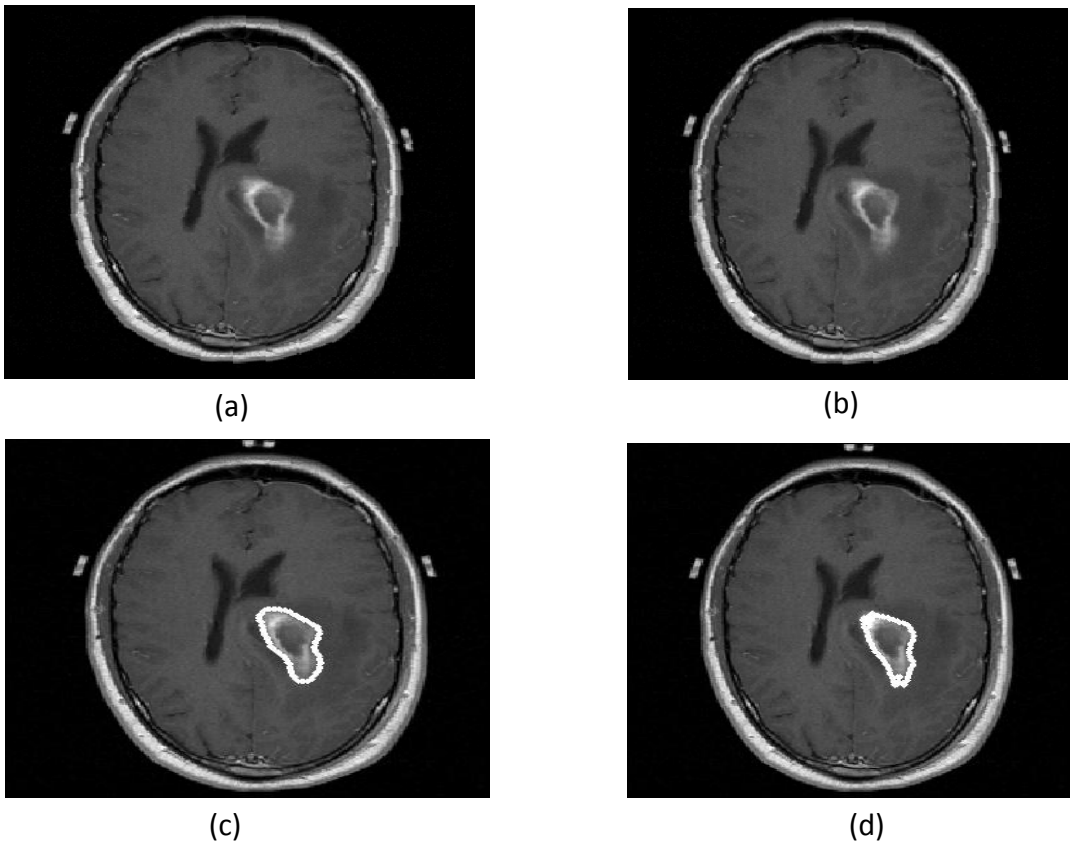


Fig.5: Comparision of Segmentation results. (a) Original MRI, (b) Our algorithm, (c) Chan – Vese algorithm and (d) Normalized Graph-cut

6. Conclusion And Future Work

This work provides automatic detection and accurate segmentation of brain tumor from magnetic resonance imagery (MRI) with no user intervention. The principal advantages of the proposed algorithm are that it (a) exploits approximate left-right symmetry of brain, (b) uses only a single MR image, so there is no effect of variability in image intensity across MR images, (c) needs no training data, (d) requires no image registration, and (e) can be implemented in real-time. We also plan to extend this to 3-dimensions.

7. References

[1] T.F. Chan, and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, vol.10, no.2, pp.266-277, 2001.

[2] D. Cobzas, N. Birkbeck, M. Schmidt, M. Jägersand, A. Murtha. 3D variational brain tumor segmentation using a high dimensional feature set. In *Mathematical Methods in Biomedical Image Analysis*, a workshop in conjunction with International Conference on Computer Vision (ICCV 2007), Rio de Janeiro, Brazil, October 2007.

[3] L.R. Dice. Measures of the amount of ecologic association between species. *Ecology*, vol. 26, pp.297-302, 1945.

[4] K. Fukunaga. *Introduction to statistical pattern recognition*, Academic Press, 2nd ed., 1990.

[5] C-H. Lee, S. Wang, F. Jiao, R. Greiner, D. Schuurmans. Learning to model spatial dependency: semi-supervised discriminative random fields. Neural Information Processing Systems. Vancouver, BC. December 2006.

[6] N. Ray, B. Saha, M. Brown. Locating brain tumor from MR imagery using symmetry, accepted at *Asilomar conf. on signals, systems, and computers*, Pacific Grove, California, USA, 2007.

[7] M. Schmidt. *Automatic brain tumor segmentation*, M.Sc. Thesis, University of Alberta, 2005.

[8] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image.*

[9] Caselles,V., Catte,F., Coll,T., and Dibos,F.(1993). A geometric model for active contours. *Numeric Mathematik*, 66:1–31.

[10] Osher, S. and Sethian, J. (1988). Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49.