EasyChair Preprint

№ 6306

# A High-Order Differential Equation Based Unsteady Wall Distance Solver

Hemanth Chandra Vamsi Kakumani and
Nagabhushana Rao Vadlamani



August 17, 2021

**Proceedings of the 8th International and 47th National Conference on Fluid Mechanics and Fluid Power (FMFP)**
**December 9-11, 2020, IIT Guwahati, Assam, India**

**FMFP2020-058**

# A high-order differential equation based unsteady wall distance solver

**Hemanth Chandra Vamsi K[1], Nagabushana Rao Vadlamani[1]**

[1]Department of Aerospace Engineering, Indian Institute of Technology Madras, Chennai 600036, India.

**ABSTRACT** Wall-distance, defined as the distance from the closest surface, is used in the formulation of several turbulence modelling strategies like Detached Eddy Simulations or in turbulence models such as Spalart Allmaras and k-$\omega$ SST. In the current work, a solver is developed to compute wall distances based on the differential equations: Eikonal, Hamilton-Jacobi (H-J) and Poisson. The baseline solver, which employs first-order up-wind scheme for the spatial discretization of advection term, has been validated against several test cases. Subsequently, the upwind schemes are replaced with high order Explicit/Compact schemes up-to 6th order accuracy. High-order schemes outperformed the accuracy of first order up-wind scheme when solving the H-J equation. Dispersion errors due to the hyperbolic nature of the Eikonal equation, as expected, has affected the wall-distance accuracy of high order schemes despite using a 10th order filter. The solver is further extended to compute unsteady wall distances, which is applicable to the flow simulations in which the walls/bodies are moving inside the computational domain. Results of an oscillating cube and a bouncing cube are demonstrated to be in agreement with the exact solution. Wall distance evolution inside the combustion chamber of a solid rocket motor due to burnback of star grain propellant is also presented.

**Keywords:** Wall distance; Hamilton-Jacobi; Eikonal; Poisson; Compact scheme; Unsteady wall distance

## I. INTRODUCTION

Wall distance computations play a vital role in several turbulence modelling strategies such as Detached Eddy Simulations (DES), Reynolds Averaged Navier Stokes (RANS) and also in peripheral flow applications incorporating additional solution physics (such as multiphase flow and electrostatic particle force modeling). There are various search-based algorithms available in the literature to estimate the wall distance. However, search algorithms are computationally expensive and suffer from weak scalability specifically on large meshes and with increase in the number of wall faces present in the domain. On the other hand, the differential equation based wall-distance algorithms (Ref. [1], [2], [3]) are both accurate and scalable. Such strategies can also be used to alter the dissipation term in the turbulence models or introduce surface roughness effects by locally altering the length scales of flow using wall-distance [7], [8]. The algorithm proposed by Tucker et. al. [1], employs first-order accurate schemes to drive the wall-distance equations to a steady state solution. In the current work, we examine the suitability of high-order methods to estimate wall distance by solving different differential equations: Eikonal, Hamilton-Jacobi (H-J) and Poisson. In addition, we extend the steady solver to compute wall-distance in unsteady geometrical environments with moving bodies. The pros and cons of the numerical algorithms are brought out in the discussion.

## II. GOVERNING EQUATIONS

In the literature, three differential equation based approaches are available to estimate the wall distance: Eikonal, Hamilton-Jacobi (H-J) and Poisson. Of these, the Eikonal equation based approach (Eqn.1) is the exact governing differential equation for wall distance.

**Eikonal equation:**

$$|\nabla\phi| = 1 \;\Rightarrow\; (\nabla\phi)^2 = 1 \;\Rightarrow\; \mathbf{U}\cdot\nabla\phi = 1 \tag{1}$$

**Poisson equation:**

$$\nabla^2\phi' = -1 \tag{2}$$

$$\phi = \pm\sqrt{\sum_{j=1,3}\left(\frac{\partial\phi'}{\partial x_j}\right)^2} + \sqrt{\sum_{j=1,3}\left(\frac{\partial\phi'}{\partial x_j}\right)^2 + 2\phi'} \tag{3}$$

**Hamilton-Jacobi (H-J) equation:**

$$\mathbf{U}\cdot\nabla\phi = 1 + (\nu\phi)\nabla^2\phi \tag{4}$$

The other two approaches, Poisson (Eqn.2, 3) and Hamilton-Jacobi (Eqn.4) equations, are numerically stable than the Eikonal equation due to the Laplacian operator. Wall-distances estimated using these two approaches are accurate close to the walls. Although the Laplacian operator enhances stability, the accuracy deteriorates as the distance from the wall increases particularly for the Poisson equation. Nevertheless, for most of the turbulence modelling strategies, it is sufficient to have accurate wall-distances close to the wall where the viscous effects are dominant. However, accurate wall distance away from the wall are required for certain applications like Computer Aided Design (CAD), computing minimal surfaces, automated meshing, etc.

In the above equations, $\phi$ represents the wall distance, U represents the Eikonal front propagation velocity and $\phi'$ represents an arbitrary quantity from which the Poisson wall distance is estimated using Eqn.3.

## III. NUMERICAL METHOD

The solver employs finite difference based schemes to discretize and evaluate the derivative terms of the differential equations in general (curvilinear) coordinates. The baseline solver uses 1st order up-wind scheme for spatial discretization of advection analogues terms. The formulation of derivative in x-direction is given in equations 5, 6 and 7 which is based on the formulation given in [1], [4]. The entire Poisson equation and the spatial derivatives of Laplacian operator in H-J equation are discretized using second order central difference scheme. In the enhanced solver, higher order spatial discretization schemes namely explicit second order (E2), explicit 4th order (E4), compact 4th order (C4) and compact 6th order (C6) schemes are used to discretize all the spatial derivative terms to solve all the terms in the differential equations. These higher order schemes are based on the formulation discussed in the Ref. [6]. As the Eikonal and H-J equations are hyperbolic and have non-linear advection analogues terms, certain amount of artificial dissipation (or filtering) is required to suppress the dispersion errors. A 10th order filter is used for this purpose [6].

$$\frac{\partial \phi}{\partial x} \approx n_{i-1}F + n_{i+1}B \tag{5}$$

$$n_{i-1} = 0.25(1 + \text{SIGN}(1, F+B)\,(1 + \text{SIGN}\,(1, B)) \tag{6}$$

$$n_{i+1} = 0.25(1 - \text{SIGN}(1, F+B)\,(1 - \text{SIGN}\,(1, F)) \tag{7}$$

where,

$$F = \frac{\phi_{i+1} - \phi_i}{\Delta x_{i+1}} \quad and \quad B = \frac{\phi_i - \phi_{i-1}}{\Delta x_{i-1}}$$

All the three differential equations are marched in pseudo time ($t^*$) with Runge-Kutta 4th order time integration by solving Eqn.8 & 9. Local-time stepping is used to accelerate the solution to a steady state. For unsteady cases, where the body(s) or the wall(s) move w.r.t time, the new mesh and boundary conditions are provided to the solver at every physical time step. Pseudo iterations with local time-stepping are used to drive the solution to a steady state between two physical time steps. All the equations were non-dimensionalized using the reference length scale $L_{ref} = 1$ m and reference time scale $\tau = 1$ sec, before solving them.

$$\frac{\partial \phi}{\partial t^*} + \mathbf{U} \cdot \nabla \phi = 1 \tag{8}$$

$$\frac{\partial \phi}{\partial t^*} + \mathbf{U} \cdot \nabla \phi = 1 + (\nu\phi)\nabla^2\phi \tag{9}$$

## IV. BOUNDARY CONDITIONS, GRID AND CASE SET-UPS

Dirichlet boundary condition is imposed on all the grid points on walls, where the value of wall distance is set to zero. Wall distance ($\phi$) is also set to zero at all the grid points lying within the volume of the solid body/bodies. Following [6], the filtering order is gradually reduced to 2nd order at the boundaries of computational domain.

$$\phi_{wall} = 0 \tag{10}$$

Neumann boundary condition is imposed at the far-field boundaries where the gradient of wall distance is set to zero.

$$\frac{\partial \phi}{\partial n} = 0 \tag{11}$$

Here, $n$ is the boundary-normal direction. The wall distance ($\phi$) values are initialized to zero inside the domain before the starting of every simulation.

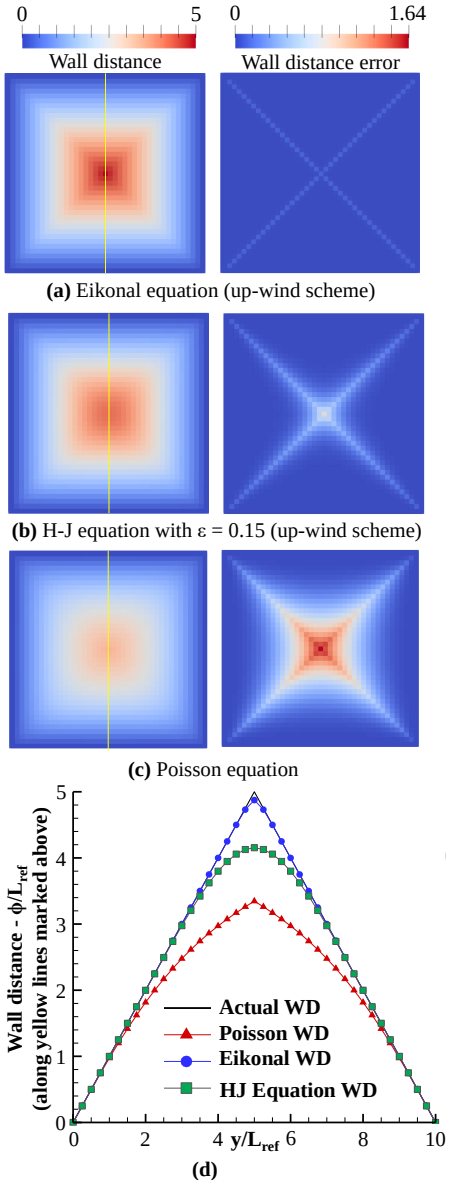
The solver has been tested and validated on four different test cases. Uniform rectilinear grid is generated for all the cases. Details of the test cases are listed below:

- **Case-1** : A 3-Dimensional (3-D) cube of length $10\times 10 \times 10$, with all of its faces treated as walls. Grid resolution is taken as $41 \times 41 \times 41$.
- **Case-2** : A $10\times10$ 2-D square domain with multiple geometric shapes engraved into it. Grid resolution chosen for this case is $101 \times 101$.
- **Case-3** : An 2D unsteady case in which a square body executes simple harmonic motion in vertical direction about the origin. The frequency and amplitude of oscillating cube are specified to be $\frac{1.5}{\pi} Hz$ and 2 units respectively. The domain size is taken as $10 \times 10$ with a grid resolution of $41 \times 41$. While the top and bottom boundaries are treated as walls, far-field Neumann boundary condition is imposed on the left and right boundaries.
- **Case-4** : An unsteady $2D$ bouncing cube simulation. In this case a square shaped object is dropped in the presence of gravity. The cube bounces off from the bottom surface with a momentum loss of $10\%$ every time after each bounce. The domain size is taken as $10 \times 10$ with a grid resolution of $41 \times 41$. Acceleration due to gravity was taken as 9.81 $m/s^2$. Boundary conditions for this test case are the same as those of Case-3.

While cases 1 and 2 are used to validate the baseline solver, cases 3 and 4 are used to validate the unsteady solver. Case-1 is also used to demonstrate the results obtained from the high order schemes incorporated into the solver. Case-2, chosen from Ref.[3], is used to demonstrate the capability of the wall-distance solver in the presence of complex geometries in the computational domain. In addition to these test cases, wall distance evolution inside a solid rocket motor with star shaped propellant grain is also simulated based on the 2-D test case details presented in Ref.[9].

## V. VALIDATION OF BASELINE SOLVER

The baseline solver, using upwind scheme, is validated for case-1, as discussed in section IV, by comparing the results against the exact solution (calculated manually as the test cases chosen here are simple). Fig 1(a,b,c) shows the contours of wall-distance solutions for case-1 at $Z = 5$ plane. Predictions from Eikonal, H-J and Poisson solvers are compared. Contours of the absolute error (difference between exact and computed solution), after deducting the




**(a)** Eikonal equation (up-wind scheme)

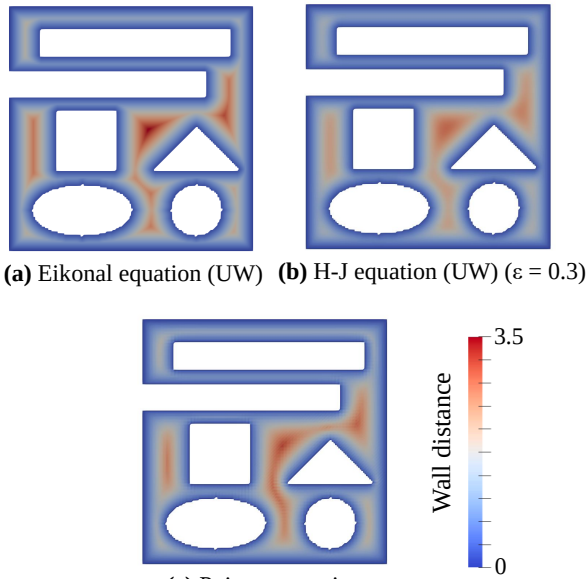
**(b)** H-J equation with ε = 0.15 (up-wind scheme)

**(c)** Poisson equation

**(d)**

**Figure** 1: **Comparison of the wall distance field and error magnitude obtained through: (a) Eikonal, (b) H-J and (c) Poisson equations for case-1 with UW scheme. (d) Comparison of wall distance along vertical line from various equations. The results are shown at $Z = 5$ plane.**

exact solution from the predictions, is also shown. Figure 1(d) compares the wall-distance estimated on a vertical line passing through the domain's geometric center. It is evident that the error close to the walls is negligible for all the approaches. However, error increases with increasing distance from the wall particularly using the Poisson approach. On the other hand, the predictions of both Eikonal and H-J are highly accurate in most of the domain. Predictions of the Eikonal equation are superior to that of H-J due to the lack of Laplacian operator in it.




**(a)** Eikonal equation (UW) **(b)** H-J equation (UW) (ε = 0.3)

**(c)** Poisson equation

**Figure** 2: **Wall distance field obtained using various equations for case-2 using the base-line solver.**

Fig 2 shows the computed wall distance contours for case-2 using all the three differential equations. The contours obtained are in agreement with the results reported in Ref. [1]. This also demonstrates the robustness and performance of the present solver in treating geometries with complex shapes. Having validated the baseline solver using upwind schemes, we have attempted to solve the same differential equations with the higher order schemes.

### A. **Effect of higher order schemes on H-J equation solution**

Wall distance solutions are computed using H-J equation for case-1 with the higher order schemes: Explicit 2nd order (E2), Explicit 4th order (E4), Compact 4th order (C4) and Compact 6th order (C6) [6]. Fig. 3(a,b) compares the wall distance values and absolute error along the vertical line passing through the origin using different schemes. The superiority of the high-order schemes can be clearly appreciated from the figure. The RMS values of error along the mid vertical line and the filtering coefficients ($\alpha_f$) used to carry out the simulations are listed in the Table 1. The RMS error using high-order schemes is clearly an order of magnitude smaller than the first order up-wind scheme. The CPU time consumed per 1000 iterations to solve H-J equation for case-1 using upwind, explicit and compact schemes are 5.78, 1.68 and 2.82 minutes respectively.

### B. **Effect of higher order schemes on Eikonal equation solution**

A similar analysis is also carried out for the solution of Eikonal equation. Performance of various schemes is tested using case-1 and the results are summarised in Table 2. The wall distance and the error plots are shown in the Fig 4. When compared to the H-J equation, the Eikonal equation

**Table** 1: **Net RMS errors with H-J equation along mid vertical line excluding the part between $y/L_{ref}$ = 3 to 7, using various schemes for case-1.**

| Scheme | $\alpha_f$ | RMS error |
|---|---|---|
| E2 | 0.495 | $8.74 \times 10^{-4}$ |
| E4 | 0.495 | $8.36 \times 10^{-4}$ |
| C4 | 0.495 | $8.33 \times 10^{-4}$ |
| C6 | 0.49 | $8.60 \times 10^{-4}$ |
| UW | - | $2.48 \times 10^{-3}$ |

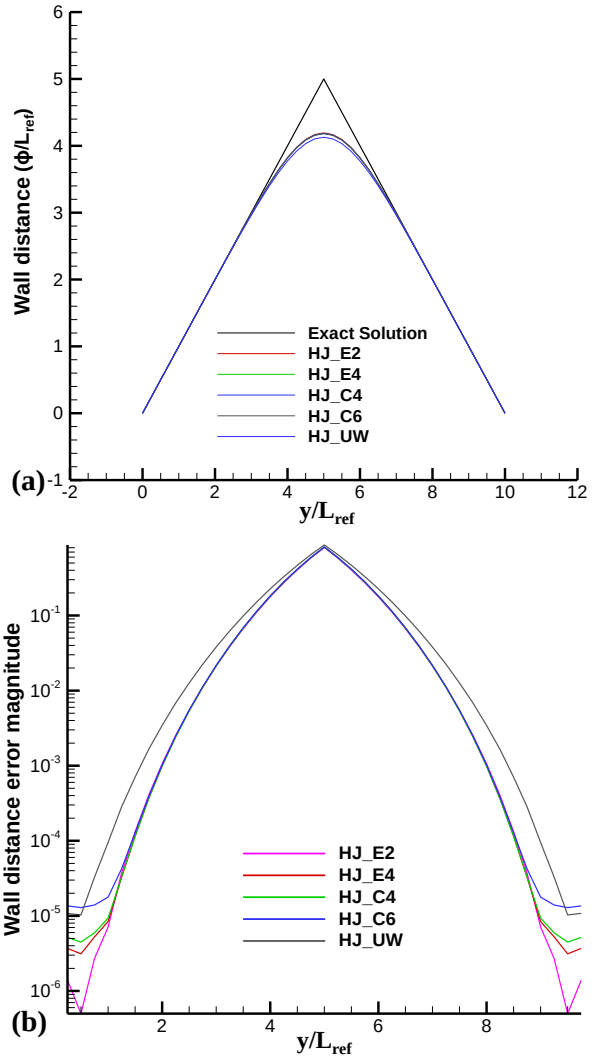



**Figure** 3: **Comparison of (a) Wall distances and (b) Wall distance error, along mid vertical line using H-J equation with schemes, E2, E4, C4, C6 and UW for case-1.**

is hyperbolic and high-order schemes are susceptible to stability issues due to the lack of Laplacian term. The effect of dispersion error is clearly seen in the solutions from

**Table** 2: **Net RMS errors with Eikonal equation along mid vertical line using various schemes for case-1.**

| Scheme | $\alpha_f$ | RMS error |
|---|---|---|
| E2 | 0.48 | $2.14 \times 10^{-2}$ |
| E4 | 0.48 | $1.40 \times 10^{-2}$ |
| C4 | 0.48 | $1.09 \times 10^{-2}$ |
| C6 | 0.47 | $8.65 \times 10^{-3}$ |
| UW | - | $2.42 \times 10^{-5}$ |

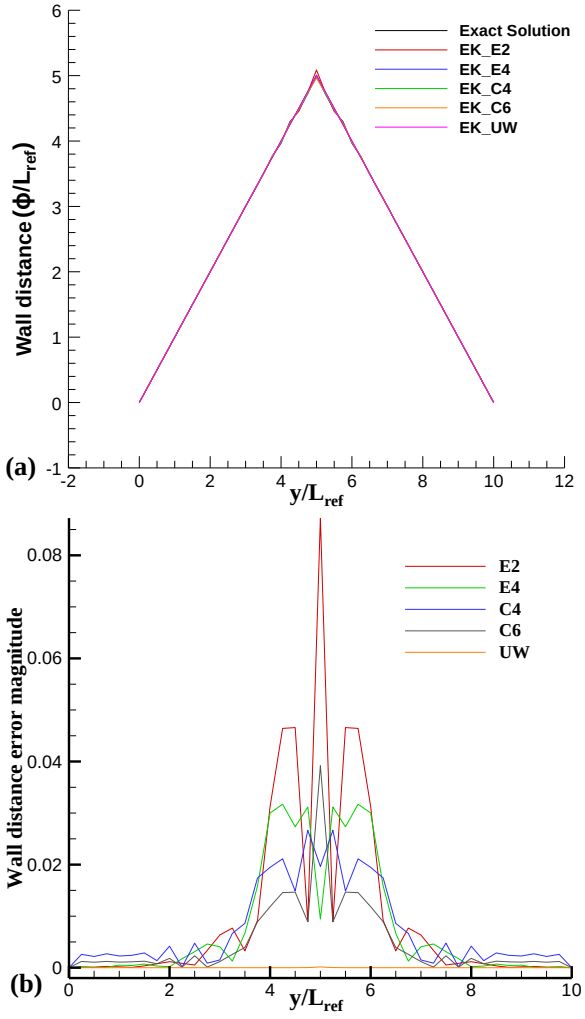



**Figure** 4: **Comparison of (a) Wall distances and (b) Wall distance error, along mid-vertical using Eikonal equation with schemes, E2, E4, C4, C6 and UW for case-1.**

the high-order methods. Although the Gibbs phenomenon is partially suppressed due to filtering, thereby enhancing the stability, the RMS error due to high-order schemes is larger when compared to the upwind scheme. Decreasing the filter order or filtering coefficient can minimize the dispersion error and improve the results. Simulations in these lines will

be addressed in the final version of the paper. The CPU time consumed per 1000 iterations to solve Eikonal equation for case-1 using upwind, explicit and compact schemes are 3.33, 1.65 and 2.78 minutes respectively.

## VI. RESULTS FOR UNSTEADY CASES

As discussed in section I, the solver has also been enhanced to compute unsteady wall distances. In this section, we demonstrate the results obtained on 2D test cases namely, oscillating cube (case-3), bouncing cube (case-4) and star grain burnback in solid rocket motor (case-5) using H-J approach with E4 scheme. The computational set-up, grid and boundary conditions for these cases are described in section IV. Fig 5(a), Fig 6(a), and Fig 7 shows the temporal variation of the wall-distance field for the 2D oscillating cube, bouncing cube and star grain solid propellant burn-back test cases respectively. It has been observed that the simulation time for these unsteady cases reduced drastically after first physical time-step, since the solution at the old physical time step provides the initial condition for the new physical time step. A probe, marked in Fig 5(a) and Fig 6(a), is placed in the domain to record the temporal variation of the wall-distance field. The corresponding temporal variation

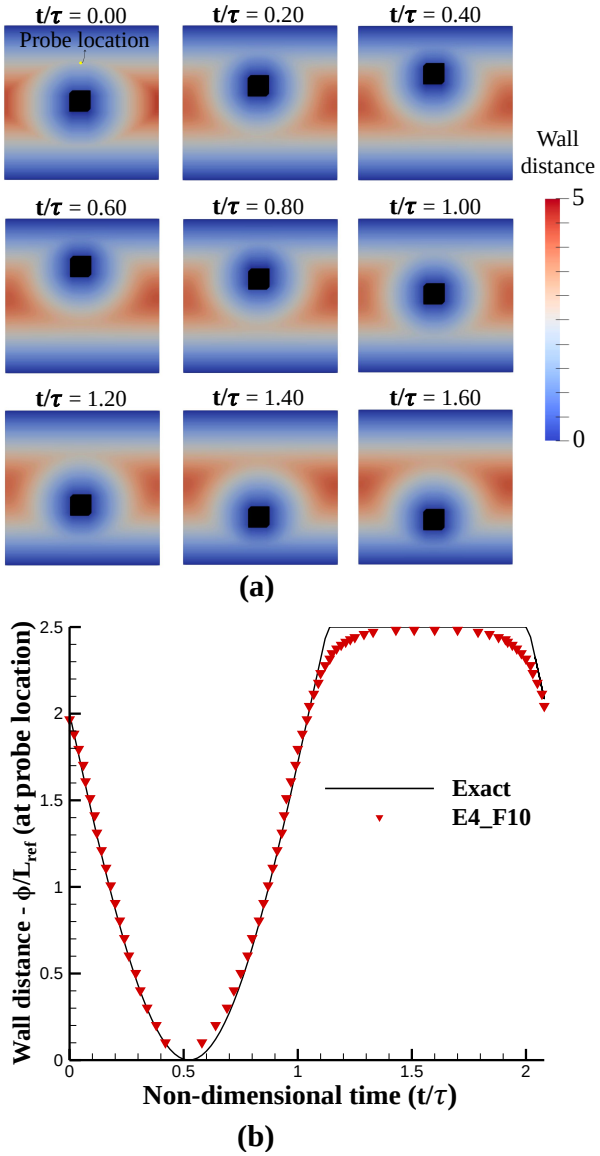



**Figure 5: (a) Temporal variation of wall distance field due to oscillating cube present inside an enclosed space using H-J equation (with E4 scheme, 10th order filtering and $\alpha_f$ = 0.495), at probe marked in the first snapshot. (b) Temporal variation of wall distance at the probe location.**

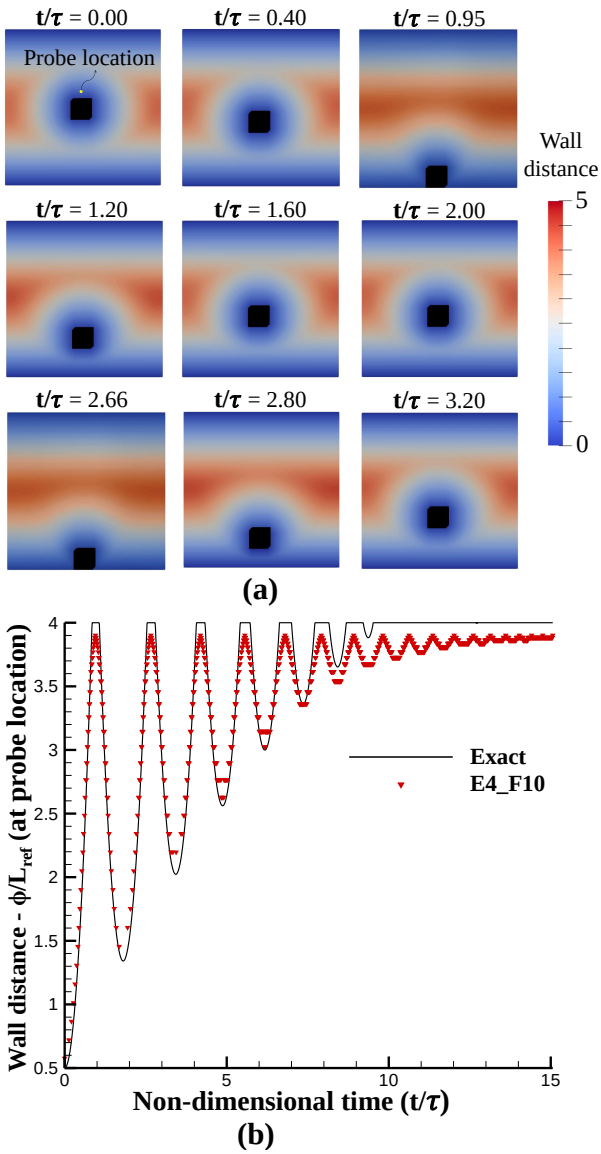



**Figure 6: (a) Temporal variation of wall distance field for case-5 (bouncing cube) using H-J equation (with E4 scheme, 10th order filtering and $\alpha_f$ = 0.495) , probe location marked in the first snapshot. (b) Temporal variation of wall distance at the probe location.**

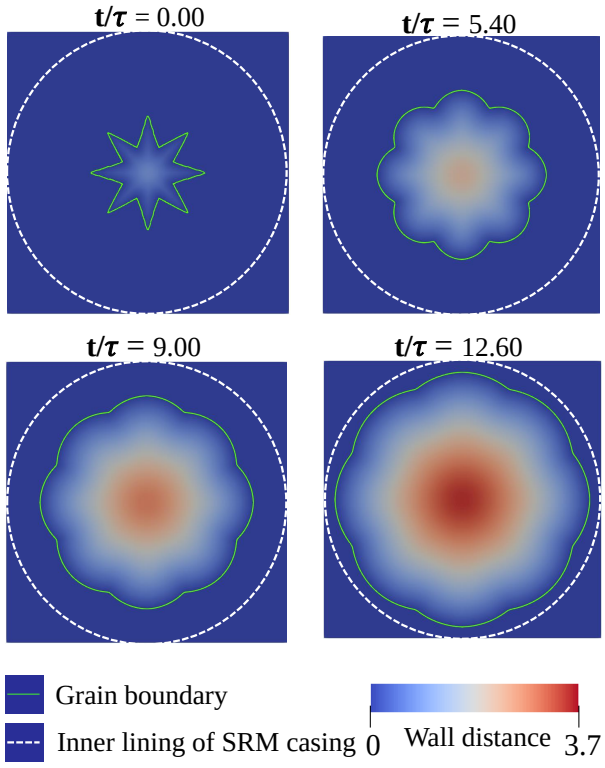



**Figure** 7: **Temporal variation of wall distance field using H-J equation inside the expanding combustion chamber of a solid rocket motor with 2-D eight legged star like propellant grain in it.**

is compared against the exact solution in Fig 5(b) and Fig 6(b). The predictions are in good agreement with the analytical data. For the bouncing cube case, every time the cube bounces off from ground it gradually looses energy and settles down on the bottom surface. Fig 6(b). shows the effect of bouncing on the evolution of exact and computed wall distance (raise, fall and asymptotic convergence of wall distance with increasing time) recorded at the probe. Under this steady state, the probe is at a location which is farthest from all the walls. It has been demonstrated in Fig. 1 that H-J equation is less accurate away from the walls. Hence, the error between the analytical and predicted wall distance at $t/\tau$ = 10-15 is notable due to Laplacian operator in H-J equation. From the snapshots of simulation presented in Fig 7 it can be seen that, as the propellant burns inside the solid rocket motor (SRM), the grain boundary moves progressively away from the center, increasing the wall distance at the central portion with time as expected.

## VII. CONCLUSIONS

A differential equation based wall distance solver is developed. Three different approaches of solving Eikonal, Hamilton-Jacobi (H-J) and Poisson equations have been demonstrated. The baseline version of the solver, using first order up-wind scheme for the spatial discretization, has been validated on simple and complex geometries. The use of high-order schemes for wall-distance solvers has been demonstrated. While solving the Hamilton-Jacobi equation, the high-order schemes out-performed the up-winding schemes by reducing the error by nearly an order of magnitude. On the other hand, the high-order schemes suffered with dispersion errors while solving the nonlinear hyperbolic Eikonal equation.

An unsteady wall distance solver has also been developed using H-J approach with an Explicit 4th order scheme. The temporal variation of wall distance field for the 2D oscillating and bouncing cube test-cases has been validated against the analytical solutions. Also, the wall distance field evolution inside the combustion chamber of a solid rocket motor with star grain structure has been presented.

## NOMENCLATURE

| | |
|---|---|
| $\phi$ | Wall distance |
| $\phi'$ | A scalar field from which Poisson equation based wall distance can be estimated |
| $U$ | Front propagation velocity implied in Eikonal equation |
| $t$ | Physical time |
| $t^*$ | Pseudo time |
| $L_{ref}$ | Length scale for non-dimensionalization |
| $\tau$ | Time scale for non-dimensionalization |
| $\alpha_f$ | Filtering coefficient |
| E2 | Explicit second order scheme |
| E4 | Explicit fourth order scheme |
| C4 | Compact fourth order scheme |
| C6 | Compact sixth order scheme |
| UW | First order up-wind scheme |

## REFERENCES


[1] Tucker, P. G., Rumsey, C. L., Spalart, P. R., Bartels, R. E., and Biedron, R. T., *Computations of wall distances based on differential equations*, AIAA journal, 43(3), 539-549. (2005)

[2] Richard J. Jefferson-Loveday, V. Nagabhushana Rao, James C. Tyacke and Paul G. Tucker, *High-order detached eddy simulation, zonal LES and URANS of cavity and labyrinth seal flows*, Int. J. Numer. Meth. Fluids. (2013)

[3] Tucker, P. G., *Hybrid Hamilton–Jacobi–Poisson wall distance function model*, Computers & fluids, 44(1), 130-142. (2011)

[4] Jing-lei XU, Chao YAN, Jing-jing FAN. *Computations of wall distances by solving a transport equation*, Appl. Math. Mech. -Engl. Ed., 32(2), 141–150. (2011)

[5] Anthony Bouchard, *Wall distance evaluation via Eikonal solver for RANS application*, Masters Thisis, UNIVERSITÉ DE MONTRÉAL. (2017)

[6] Miguel R. Visbal and Datta V. Gaitonde, *High-Order Schemes for Navier-Stokes Equations: Algorithm and Implementation Into FDL3DI*, AFRL. (1998)

[7] Jefferson-Loveday, R. J., Tucker, P. G., Northall, J. D., Nagabhushana Rao, V. *Differential equation specification of integral turbulence length scales*, Journal of turbomachinery, 135(3). (2013)

[8] Vadlamani, N. R., *Numerical investigation of separated flows in low pressure turbines*, Doctoral dissertation, University of Cambridge. (2014)

[9] Arnau Pons Lorente, *Study of grain burnback and performance of solid rocket motors.*, Project report, Universitat Politècnica de Catalunya. (2013)