

## Internal Ballistic Code for Solid Rocket Motors using Minimum Distance Function for Grain Burnback

Afroz Javed<sup>1</sup>, Iyer Arvind Sundaram<sup>#</sup>, and Debasis Chakraborty<sup>1,\*</sup>

<sup>1</sup>Defence Research and Development Laboratory, Hyderabad, India

<sup>#</sup>Indian Institute of Science, Bengaluru, India

\*E-mail: debasis\_cfd@drdl.drdo.in

### ABSTRACT

A computer code has been developed for internal ballistic performance evaluation of solid rocket motors, using minimum distance function (MDF) approach for prediction of geometry evolution. This method can handle any complex geometry without the need to define different geometrical shapes and their evolution as used in several existing analytical geometry evolution-based methodologies. The code is validated with both experimental results published in literature, as well as for solid rocket motors of tactical and strategic missiles and a very good match is obtained with static test results. The output of the code gives pressure-time (p-t) curve as well as the detailed parameters of the flow along the axial direction, and geometries in the form of mesh file, which can be further used as input to codes for CFD analysis.

**Keywords:** Internal ballistic code, grain burnback, minimum distance function, solid rocket motors

### 1. INTRODUCTION

In the design and development of solid propellant rocket motors, an accurate and reliable internal ballistics model provides a physical understanding of the main internal ballistic driving phenomena, besides being used for design optimisation through parametric analysis of the available design options<sup>1-4</sup>. A good internal ballistics model is also required for the characterisation of the solid rocket motor performance and mission capabilities for both nominal and non-nominal behaviour. Both the flow field in the flow cavity of the motor and the grain evolution with time needs to be predicted by the internal ballistics model. The problems of solving the flow field and the grain-burning surface evolution are completely coupled with each other, through the geometry evolution of the chamber and the local burning rate values. However, it is possible to assume that, during small time intervals, the flow-field conditions within the chamber are not influenced in any relevant manner by the grain geometrical evolution. This assumption introduces only negligible approximation in the internal ballistics solution, considering that the burning surface evolution occurs at a significantly slower rate (the grain-burning rate, of the order of mm/s) than the flow-field development (the flow-field velocity, of the order of m/s, or the acoustic signals velocity, of the order of hundred-to-thousand of m/s). Therefore, an offline coupling between the grain burnback and the flow-field model is considered without much loss of accuracy. The flow in a solid rocket motor generally remains low, subsonic, and accurate predictions can be made with inviscid flow assumptions.

The grain geometry evolution for simple grain configurations like star, wagon wheel, dendrite, etc. are

generally carried out using analytical methods<sup>1,5</sup>. The grain geometry evolution of a complex initial geometry grains like conocyl or finocyl require use of generalised three-dimensional computer programs using different algorithms and grain burnback evaluation methodology<sup>5</sup>. Level set method<sup>6,7</sup> is one of such methods, and has been reported in literature<sup>8,9</sup> for evaluation of the burnback of solid propellant grain. Zero level set of a signed distance function is set at the grain surface to track its motion in this method. The motion of the interface is represented by the evolution of this zero level set function. The initial value partial differential equation for the evolution of the level set function resembles a Hamilton–Jacobi equation. The level set method allows easy evaluation of the curvatures and normals, and natural evolution of topology. However, there are numerical issues which are amplified by the vast density difference across the interface (the density ratio being around 150–200), any small computational error becomes magnified due to this density ratio and poses problems in convergence and the accuracy of the solution.

The shortcomings of level set method have been addressed by Wilcox<sup>10,11</sup>, *et al.* using a method of interface capturing based on the use of minimum distance function (MDF). The minimum distance to the initial surface can be used to represent the detailed shape of the burning surface of the 3-D grain as it evolves in time by manipulating a signed MDF to describe the motion of the solid propellant interface. In contrast to the level set method, there is no hyperbolic partial differential equation to be solved. Though computationally intense, the geometry initialisation needs to be run only once for each grain design. In the methodology proposed by Wilcox<sup>10,11</sup>, *et al.* the grain surface normal having a component in axial direction may

introduce errors with a combination of situations, while in the present work, this drawback has been solved by calculating the burn surface areas directly. Also the present methodology exports the geometries at different burn instants which can be directly used with commercial CFD softwares for quasi-steady state simulations.

## 2. FLOW SOLVER MODULE

The flow solver module predicts the flow field, for a given grain geometry, nozzle throat diameter and the propellant properties. To solve the flow field, the flow domain surrounded by the propellant grain, is discretized along the axial direction. A schematic representation of this discretized domain is shown in Fig. 1. One end is the head-end of the grain and the other is the nozzle-end. At the head-end, a pressure value is assumed initially, and using the conservation of mass, momentum and energy, the flow parameters are calculated at successive nodes till the nozzle-end is reached. The total pressure at the nozzle-end and the nozzle diameter determine the mass flux through the nozzle, which must match the mass flux at the nozzle-end of the grain. The absolute difference in these two mass fluxes is the error, which is sought to be minimised, by changing the head-end pressure.

The total pressure ( $p_0$ ) and the static pressure ( $p$ ) are the same at the head-end. The total temperature ( $T_0$ ) of the gas is the adiabatic flame temperature of the propellant, which is generally evaluated through the NASA CEA 600<sup>12,13</sup> program using propellant composition, initial temperature and a nominal pressure. The adiabatic flame temperature is not a very strong function of pressure and can be taken as constant for the kind of variations observed in the nominal pressures generally occurring in the solid rocket motors. At the head-end the static temperature ( $T$ ) is the same as the total temperature. The density of the gas ( $\rho$ ) at head-end is evaluated using ideal gas equation

$$\rho = \frac{p}{RT}$$

The velocity of the gas and consequently the Mach number at the head-end are zero. Thus for the first cell, all the head-end side properties are known. The head-end side properties are represented by the subscript  $l$  (for left) and the nozzle-end side properties be represented by  $r$  (for right). For any cell, the mass flux balance gives,

$$\dot{m}_l + \dot{m}_b = \dot{m}_r$$

Here  $\dot{m}_b$  is the rate of mass addition in the cell due to burning of the propellant from the surface, between the left and the right faces. The area of the burning surface is obtained from the geometry evolution module, and the rate of burning is the

average rate of burning on the left cell and the right cell. The rate of burning is given as

$$\dot{r} = Ap^n$$

However, towards the nozzle-end, due to significant values of axial velocity erosive burning can occur altering the burn rate from that evaluated using above equation. The modified burn rate is given as<sup>14</sup>

$$\dot{r}_{erosive} = \dot{r} \left( 1 + 0.023 (g^{0.8} - g_{th}^{0.8}) \mathcal{H}(g - g_{th}) \right)$$

Here  $g$  is the ratio of mass flux through the port to the mass efflux from the surface modified for size effects as  $(G/\rho_p \dot{r})(\rho_p \dot{r} d_0 / \mu)^{0.125}$ , with  $G$  as the mass flux through the port,  $\rho_p \dot{r}$  is the non-erosive mass efflux from the propellant burning surface,  $g_{th}$  is the threshold value obtained from the plot of the various available experimental data as 35, and  $\mathcal{H}$  is Heaviside step function that is introduced to cater for a critical flux below which there is no erosive burning. The dynamic viscosity of the gaseous products of combustion is represented by  $\mu$ . The value of  $d_0$  is taken as  $P/\pi$  where  $P$  is the perimeter of the grain port<sup>15</sup>.

The pressure initially is assumed on the right face to be the same as that on the left face and the same is utilised to evaluate the local burn rate. This assumption is later discarded after evaluation of the correct pressure at the downstream face. To evaluate the pressure at downstream face, conservation laws for mass, momentum, and energy are used as follows:

The momentum conservation equation can be written as

$$\dot{m}_l u_l + A_l p_l + \left( \frac{p_l + p_r}{2} \right) (A_r - A_l) = \dot{m}_r u_r + A_r p_r$$

This equation can be re-ordered to define a new parameter  $\dot{M}$  as

$$\dot{M} \equiv \dot{m}_l u_l + p_l \frac{(A_r + A_l)}{2} = \dot{m}_r u_r + p_r \frac{(A_r + A_l)}{2}$$

This term is the same on both planes of the cell. From the upstream (left) state, the value of  $\dot{M}$  can be evaluated. Writing the average port area  $\frac{(A_r + A_l)}{2}$  as  $A_p$ , using  $\dot{m} = \rho A u$ , and using the ideal gas equation, to represent  $p_r$ , one gets the following:

$$\dot{M} = \frac{\dot{m}_r^2}{\rho_r A_r} + \rho_r R T_r A_p$$

The above relation contains the static temperature, density, and mass flow rate at the right face as unknown quantities. Finally, the total enthalpy remains a constant.

$$h_0 = h + \frac{u^2}{2}$$

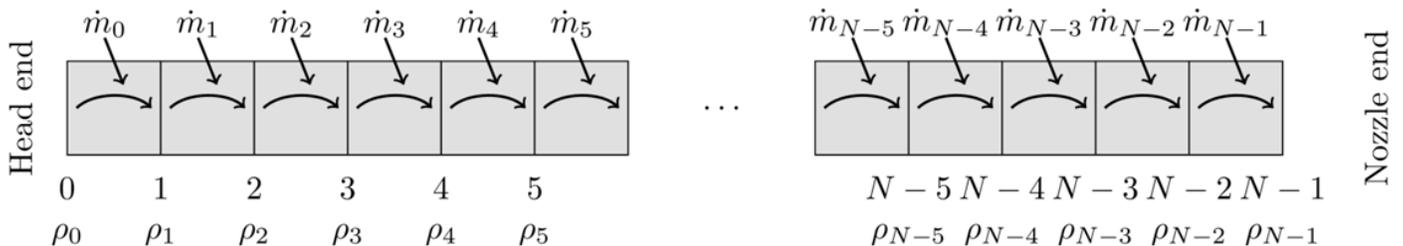


Figure 1. Schematic representation of the discretized solid rocket motor flow domain.

This gives the relation between the temperature and the velocity as

$$T_r = \frac{h_0}{C_p} - \frac{u_r^2}{2C_p}$$

And from mass conservation ( $\dot{m} = \rho Au$ ), one gets

$$T_r = \frac{h_0}{C_p} - \frac{\dot{m}_r^2}{2\rho_r^2 A_r^2 C_p}$$

Substituting this in the momentum equation, one gets

$$\dot{M} = \frac{\dot{m}_r^2}{\rho_r A_r} + \rho_r RA_p \left( \frac{h_0}{C_p} - \frac{\dot{m}_r^2}{2\rho_r^2 A_r^2 C_p} \right)$$

Reorganizing this yields

$$\frac{h_0 RA_p}{C_p} \rho_r^2 - \dot{M} \rho_r + \frac{\dot{m}_r^2}{A_r} - \frac{\dot{m}_r^2 RA_p}{2A_r^2 C_p} = 0$$

Which is solved for  $\rho_r$ , and the mass conservation equation is used to get the velocity on the downstream surface. Using energy conservation equation, the temperature on the downstream surface is obtained, and then finally, the pressure is obtained using ideal gas equation. With this new value as the fresh estimate, the calculations are repeated till the pressure on the downstream surface converges, and all flow parameters become consistent within the required tolerance.

The above process is continued from the head-end to the nozzle-end. On reaching the nozzle-end, the total pressure is calculated using the Mach number, and the static pressure. The total pressure at the nozzle-end ( $p_c$ ) determines the mass flux through the nozzle.

$$\dot{m} = \frac{p_c A_t}{c^*}$$

Here  $c^*$  is determined from the propellant properties as:

$$c^* = \frac{\sqrt{RT_0}}{\Gamma}$$

where  $R$  is the gas constant for the products of combustion,  $T_0$  is the adiabatic flame temperature, and  $\Gamma$  is a function of ratio of specific heats given as

$$\Gamma = \sqrt{\gamma} \left( \frac{2}{\gamma+1} \right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

This mass flux should match the mass flux at the rightmost surface of the domain of computation. The head-end pressure is suitably estimated from the difference between these two mass fluxes, and the entire process is repeated till there is a balance in the mass fluxes.

Occasionally, the assumption of the pressure of the head-end is such that during iterations, the solution of the quadratic equation yields complex results. This indicates the choking of the flow path, and the required mass flux cannot pass through the port at the assumed head-end pressure. In this situation, the head-end pressure is increased slightly, and the calculations are restarted from the head-end.

### 3. GEOMETRY EVOLUTION USING MINIMUM DISTANCE FUNCTION

Instantaneous burning areas, port areas, and the burning

perimeters are calculated at different axial locations by the geometry evolution module for being used as instantaneous geometry description by the flow-solver module.

#### 3.1 Grain Geometry and Grid

The geometry evolution model is based on the ‘minimum distance function’ technique based on the work of Wilcox<sup>10,11</sup>, *et al.* The input geometry is provided in the form of a triangulated surface file. The input format of the file is converted to a native format which also contains the required connectivity information. The computations are performed on a grid of points which is a Cartesian distribution of  $n_x \times n_y \times n_z$  points separated by distances of  $d_x, d_y, d_z$ . The grid is so chosen that it completely encompasses the grain. Usually, the axial point at the head-end is the origin of the grid. Often, to reduce the computation, symmetries are made use of, and the grid is present in only one (quarter symmetry) or two (half symmetry) quadrants.

#### 3.2 The Minimum Distance Function

The first step, which is performed only once for the entire calculation, is the calculation of the ‘minimum distance’ scalar ( $\phi(x, y, z)$ ). This is a signed scalar, whose value is equal to the minimum distance of this point from any triangular facet of the mesh. The sign is positive when the point is outside the flow domain and inside the grain, negative when the point is in the flow domain, and zero on the burning surface. This involves finding the distance of the grid point with each facet of the mesh. Depending on where the point is located relative to a given facet, the minimum distance of the point to the facet may be the distance to the face, to an edge or to a vertex of the facet.

#### 3.3 Capturing the Burning Surface

Once  $\phi(x, y, z)$  is obtained, the burning surface area, the port area, and the port perimeter can be evaluated. To calculate the burning surface area, in the current implementation (which differs from the method<sup>10,11</sup>, which uses the perimeter to estimate the burning area) the constant surface of  $\phi = 0$  is tracked. This is done using the marching cubes technique. The result is a set of triangles which represent the current burning surface. Furthermore, the triangles were aligned to the grid, so that between any two locations along the axial direction which are aligned with the grid points, the triangles representing the burning surface are always whole, can be listed, and the sum of the areas of these triangle represents the total burning area between these two locations. The current technique results in more accurate estimation for the burning surface area, especially in the cases where there is a significant axial component of the grain surface normal. This also automatically handles the cases with end-burning grains. To estimate the port area and the port perimeter, the technique of marching squares was implemented at each axial plane of the grid. This gives the contours of  $\phi = 0$  along each axial location, which could be easily analysed for the port area as well as the perimeter. The perimeter is required to evaluate the equivalent diameter for erosive burning calculations.

### 3.4 Evolution of the Grain Geometry

The grain evolution occurs due to burning in a direction normal to the surface. Burning of the surface by a distance  $d$  causes the value of  $\phi$  to decrease by  $d$ . When the burn distance changes along the length of the axis, the increment in  $\phi$  also is chosen at the same axial location. Once the new values of  $\phi$  are obtained at all the grid points; the process of capturing the burning surface and the burning contours is repeated, which gives the burning area, the port area, and the port perimeter at different axial locations. The evolution of a sample grain is shown in Fig. 2. It can be seen that the evolution of the grain geometry is captured quite well by the present technique. It may be noticed especially that the sharp convex features of the grain gets blunted out, and the sharp concave feature remains sharp. This property, known as the entropy solution, is captured well with the present technique, as can be seen in Fig. 2, highlighted by the dark lines on the surfaces.

### 3.5 Handling of the Casing

Special consideration needs to be given when the grain approaches the casing. Once the surface reaches the casing, it is no longer burning and does not contribute to mass addition but it is required for calculation of the port area and perimeter. Hence, in the present work, a separate parameter is used, which is associated with each facet on the burning surface. This parameter takes a value of 1 when the surface is burning, and 0 when it is not, and fractional, when a portion of the triangle is burning.

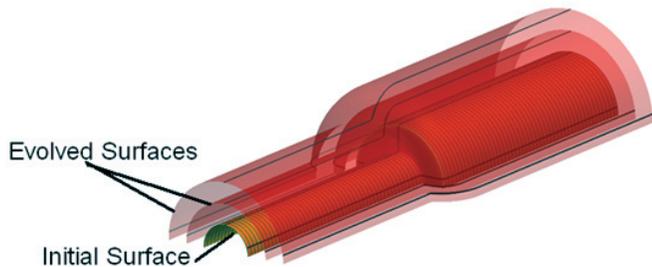


Figure 2. Evolution of the burning surface.

To be able to handle cases with non-cylindrical and non-axi-symmetric casings, a separate MDF is calculated which goes to zero on the casing surface. The value of  $\phi$  is constrained so as not to go below the value of this minimum value. A value of  $\phi$  equal to this value represents that the surface is not burning, and the value of the live parameter is set to 0 for such triangles. This technique provides the correct port area, and the perimeter as well as the correct burning surface area. This can be seen in Fig. 3, where the red colour represents the burning surface and the blue represents the surface which has reached the casing. Furthermore, to have a better approximation, the triangles, which are partially cut by the casing, have been assigned a fractional live factor.

## 4. DETAILS OF SOFTWARE

The bulk of the ballistic code software is developed in the Python<sup>16</sup> programming language. The major parts of the Python programme are as follows;

- The calculation of MDF

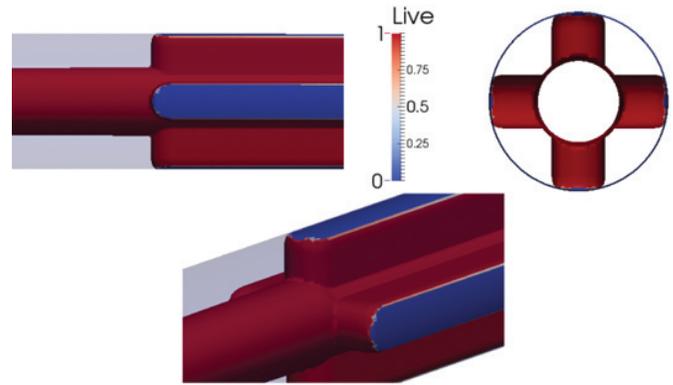


Figure 3. Evolution of a grain and the factor contributing to burning.

- Fluid flow solver
- Generation of burning surface and the contours of  $\phi = 0$

The complete solution in Python, would become prohibitively time-consuming. In view of this, several steps were taken to speed up the processing. The complete geometry evolution module is coded in pure C++. The MDF calculations were very much amenable to parallelization. This fact was made use of and the module was parallelised with open MP (shared memory architecture). Wrappers for this C++ library were then written to make it accessible from Python. The fluid flow solver was optimised using Cython. This usage makes it much faster than the pure Python version.

A schematic flow chart is shown in Fig. 4 to represent the process of the transfer of information within different modules of the internal ballistics code. The initial geometry in suitable format is imported to the program, which then generates MDF. This MDF is used for generating burning surface information and contours of the port geometry. The instantaneous grain geometries can be exported at this stage. The flow solver module uses the information of burning surface area, port area, and perimeter along with burn rate estimation module to generate instantaneous flow parameters. These parameters contain axial velocities, static and total pressures, densities and static temperatures along the length of the grain at each instant. For the succeeding time instant, MDF is modified

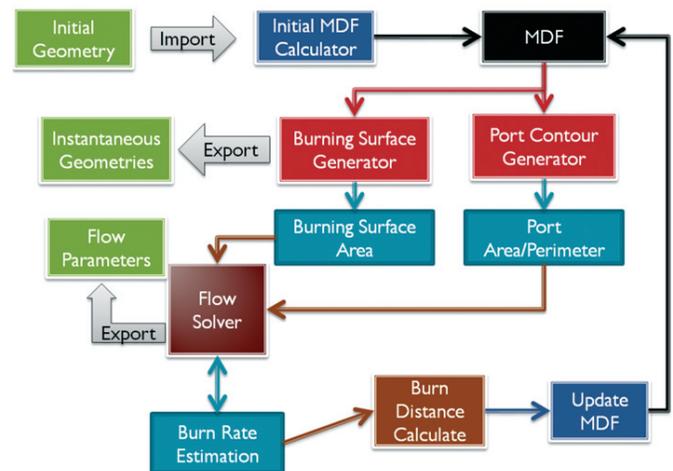


Figure 4. Flow chart of the information transfer within different modules of the code.

using burn distance calculating module. This updated MDF is used for carrying out the calculations for the next time step. The procedure is continued till all the propellant is consumed.

**5. TEST CASES FROM LITERATURE**

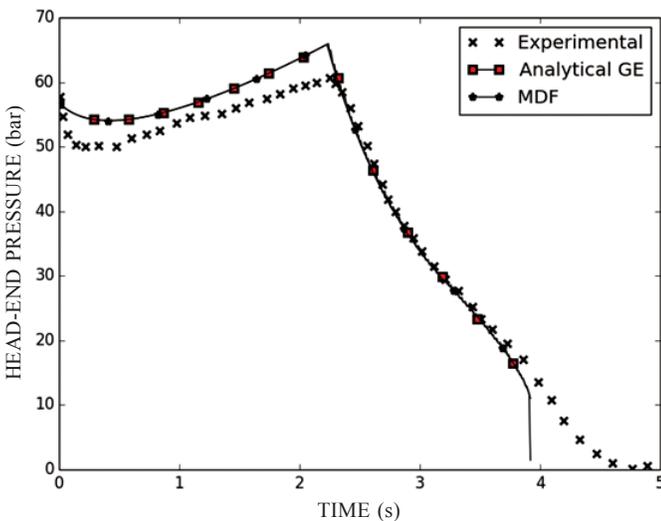
To test the combination of the flow solver and the geometry evolver, test cases for cylindrical motors from Hasegawa<sup>17</sup>, *et al.* were simulated. These motors were cast using a composite propellant with the composition of 69 per cent AP, 17 per cent HTPB, and 14 per cent Aluminium. The density of the propellant was 1700 kg/m<sup>3</sup>, linear burning rate was 4.9 mm/s at 49 bar pressure and 20 °C temperature, pressure exponent value was 0.3, the adiabatic flame temperature was reported to be 3041 K at 50 bar with frozen equilibrium assumption, mean molecular mass as 25.4 g/mol, and ratio of specific heats was 1.19. Three different motors namely type A, B, and C with outer shell diameters of 80 mm were tested with this propellant. The geometrical properties of these motors are given in Table 1.

**Table 1. Geometrical properties of cylindrical motors**

	Type A	Type B	Type C
Nozzle throat dia (mm)	34	23	34
Initial port dia (mm)	40	40	60
Grain length (mm)	1680	840	1260

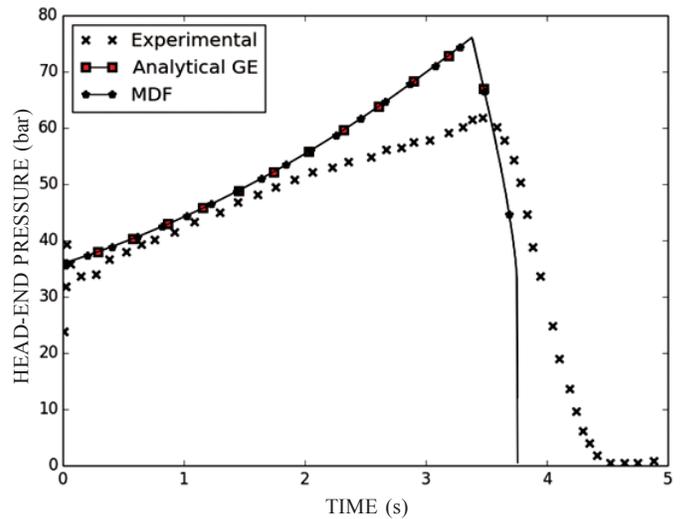
Present software is used to evaluate the pressure-time (*p-t*) curves of these motors. The *p-t* curve of type ‘A’ motor is shown in Fig. 5. The experimental curve is also shown for comparison. It can be seen that the initial pressure and location of pressure peak show very good match with the experimental result. The values predicted are nearly 5 per cent higher than the experimental values in the initial region, while the tail-off region shows exact match. For comparison, taking the advantage of simplicity of the geometry, simulation using an analytical geometry evolver is also shown in the same figure. The analytical geometry evolver and MDF methodology match very well with each other.

Similarly the *p-t* curve for Type ‘B’ and Type ‘C’ motors are shown in Figs 6 and 7, respectively. Type ‘B’ motor shows

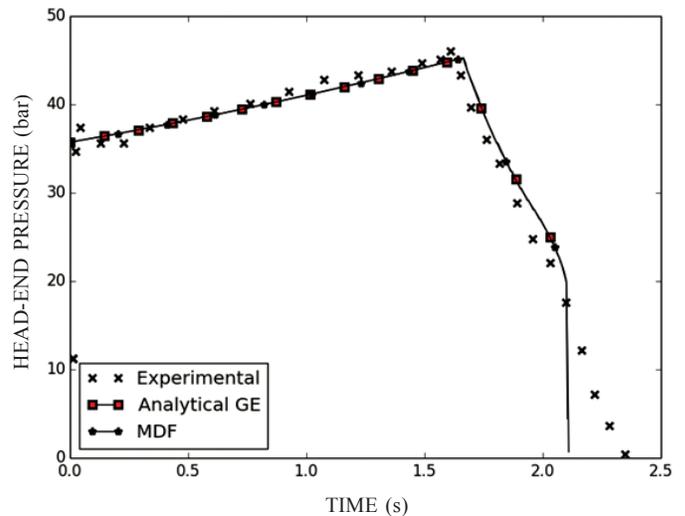


**Figure 5. p-t curve for type A motor.**

very good match initially, but after around 2 s of burning, the pressure starts departing from the experimental pressure values. This departure could be attributed to the nozzle throat erosion, resulting from higher burn time. For Type ‘C’ motor as shown in Fig. 7, the burn time was comparatively smaller, and an exact match of the predicted and experimental results is observed. For all these predicted curves a difference in the tail-off region may be observed with the experimental data. One of the reasons for the tail-off region in the experimental curve is the predicted chamber pressure being on the higher side, resulting in higher burn rates and earlier consumption of the propellant, as can be observed for type ‘A’ and ‘B’ motors. In addition, the tail-off region occurs due to the fact that the internal ballistic programme stops the calculations once all the propellant is consumed, but the motor chamber is still filled with high pressure and temperature gases. It takes some time for these gases to exhaust out of the nozzle; also the high temperature may cause pyrolysis of the liner material adding to the mass of these residual gases. The tail-off region in the type ‘C’ motor could be occurring due to this reason, as the predicted pressures are nearly equal to the experimental



**Figure 6. p-t curve for type ‘B’ motor.**



**Figure 7. p-t curve for type ‘C’ motor.**

pressures for most of the burn time. The sharp change in the computed  $p$ - $t$  profile before burnout is due to the propellant boundary reaching the motor casing near the nozzle-end, and consequent decrease of propellant burning area. The same is observed in experimental curve in Figs 5-7 also.

With these predictions by the internal ballistics programme for geometrically simple motors, the performance of flow solver and geometry evolver using MDF methodology is validated. As further testing of the software, few solid rocket motors with non-cylindrical 3-D grain geometries are studied and the results are discussed in the following section.

**6. SIMULATIONS OF EXISTING MOTORS**

The motors considered for analysis with their salient features are listed in Table 2. All the motors are having finocyl grain structure. The grain diameters shown in Table 2 represent diameter of the cylindrical part of the grain. The ballistic code is run for these geometries using the surface mesh files. With the progress of burn time, grain burnback takes place according to the burn rate law specified. The typical grain geometry evolution with time for Motor-4 is shown in Fig. 8 at different time instants. The red colour surfaces show grain boundaries while the blue colour surfaces show the exposed combustion chamber casing in due course of burning. These grain geometries are in suitable format to be directly exported to CFD softwares for more detailed quasi-steady state flow analysis required for combustion instability and other related phenomena.

The pressures are normalised by the maximum pressures occurring for the particular cases. These maximum pressures

used for normalisation are also shown in Table 2. The variations of the normalised head-end pressure with time are plotted for all the motors in Fig. 9. In Fig. 9(a), the analytical predictions are also shown along with the predictions carried out using MDF approach, while only MDF predictions are shown in Figs 9(b)-9(d).

Figure 9(a) depicts the  $p$ - $t$  curve for Motor-1. The MDF geometrical evolution methodology shows a maximum of around 4 per cent difference with the experimental values at around 0.5 s of burn time. The location of pressure peak also shows a good match. For this geometry a considerable amount of effort was made to make an analytical geometric evolver by defining all the different geometrical shapes. The results from the analytical geometric evolver are also shown in the same figure. In the beginning, both analytical and MDF-based burnback methodologies predictions match well, but with the progress of burning, these start departing from each other. A larger departure from the experimental  $p$ - $t$  curve is noticed for analytical geometry evolution methodology. This difference can be attributed to the inability of the analytical geometric evolver in accurate prediction of the shapes of intersecting curves, their movement, change in topology, vanishing surfaces, etc.

The Motor-2  $p$ - $t$  curve is shown in Fig. 9(b). The predicted pressure is initially around 3 per cent higher than the experimental pressure values but this difference goes on reducing with burn time. The ballistic code is able to exactly predict the location of pressure peak in time. The  $p$ - $t$  curve for Motor-3 is shown in Fig. 9(c) as evaluated from the ballistic software. It can be observed that the experimental head-end pressure matches well with the predicted head-end pressure for

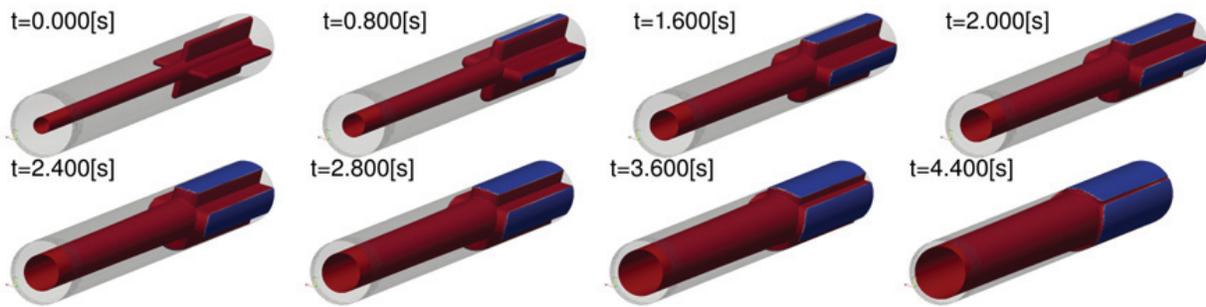


Figure 8. Grain geometry evolution of Motor-4.

Table 2. Salient features of the motors considered for internal ballistic analysis

Motor	1	2	3	4
Type	Finocyl four fin	Finocyl four fin	Finocyl eight fin	Finocyl four fin
Grain length (m)	2.1	3.8	4.8	1.727
Grain diameter (m)	0.17	0.41	0.31 - 0.49	0.168-0.165
Throat diameter (mm)	45	116	176	40
Burn rate at 7 MPa (mm/s)	6.5	9.6	19	8.2
Pressure index	0.25	0.4-0.48	0.38	0.3
Adiabatic flame temperature (K)	2980	3386	3008	2750
Molecular weight	25.1	19.2	27.19	24.8
Ratio of specific heats	1.214	1.266	1.15	1.19
Maximum pressure (bar)	120	104	110	124

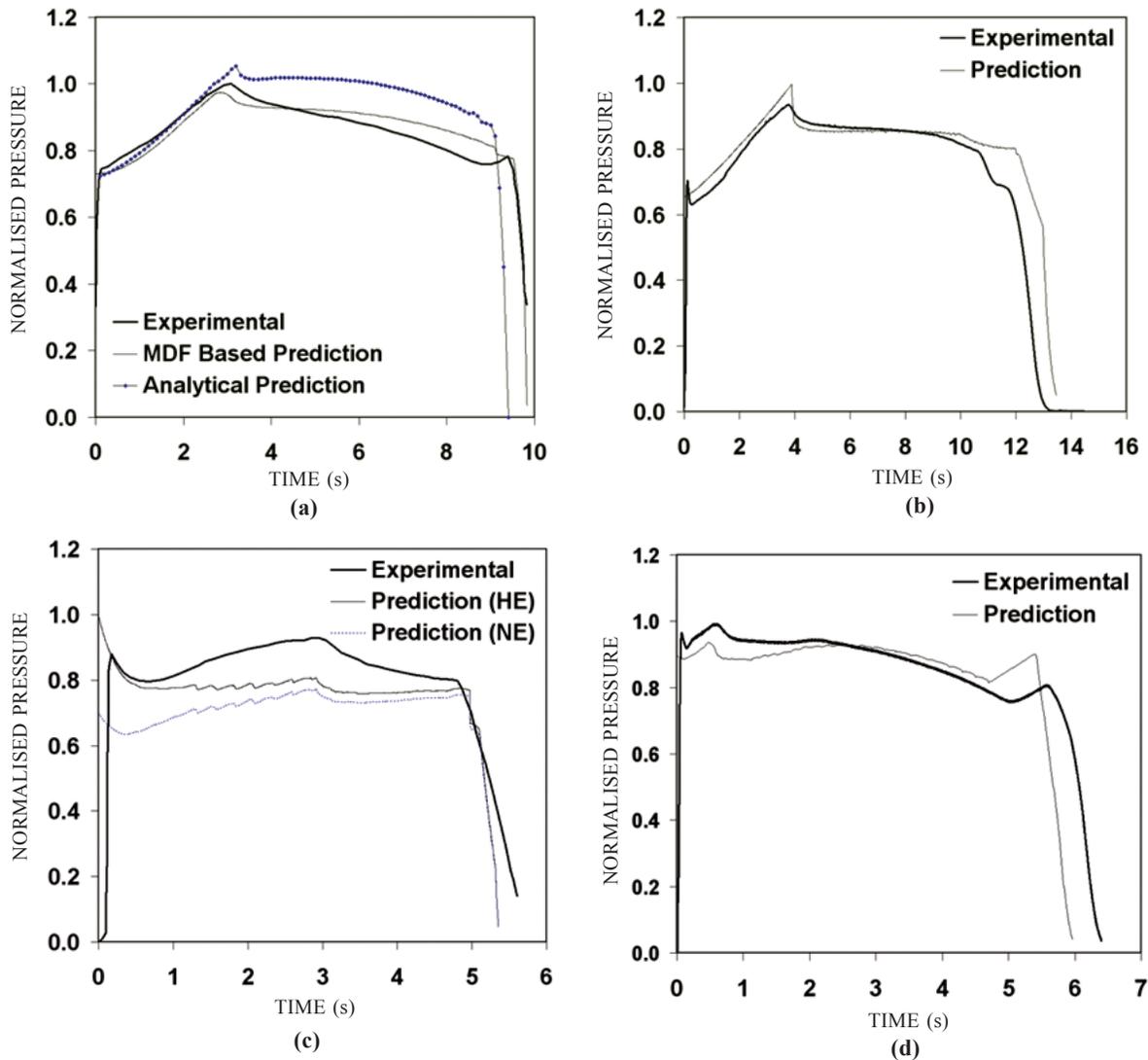


Figure 9. Pressure-time curves of motors considered for analysis (a) Motor-1 (b) Motor-2 (c) Motor-3 (d) Motor-4.

first one second of the motor operation. However, the predicted pressure becomes lesser than that observed experimentally, and a pressure hump is observed with a maximum of around 14 per cent difference. Some other independently written codes using analytical method also have failed in capturing this hump. The reason of this behaviour may be the ‘hump effect’ or ‘mid-web anomaly’ discussed in literature associated with grain casting process and AP particles orientation in composite propellants. Hasegawa<sup>18</sup>, *et al.* have presented a review and analysis of this effect in detail with experimental studies on different casting processes and parameters. However, no numerical modelling has been suggested to predict this anomaly quantitatively. Nozzle-end pressure is also drawn for this case in Fig. 9(c). A large difference between the head-end and nozzle-end pressures is indicative of significant erosive burning. It can be noticed that the erosive burning model used in the present internal ballistic software is able to correctly predict this behaviour as evident from the initial matching of the predicted head-end pressure with the experimental pressure values. The  $p-t$  curve for Motor-4 is shown in Fig. 9(d). A slightly lower (5 per cent) value of the head-end pressures are observed initially from

the predicted results when compared with the experimental head-end pressures. The time instant ( $\approx 0.5$  s) of occurrence of pressure peaks for both the experimental and predicted results match well with each other. After the initial stage ( $\approx 1.0$  s), the difference goes on decreasing and both the experimental and predicted pressures match well till around 5 s of burn time. Near the end of the burning, the predicted pressure shows a higher and slightly earlier pressure peak than that observed experimentally. This difference could be due to throat erosion and associated decrease in chamber pressure.

## 7. CONCLUSIONS

An internal ballistic code has been developed consisting of an MDF-based geometry evolution code, and a fluid solver code. The input geometry of the grain burnback module is a suitably discretised surface geometry file, and hence, can be used for generic geometries. The code is validated with both experimental results published in literature, as well as motors for tactical and strategic missiles. A very good match of  $p-t$  curves is obtained from the program and experimental results for cylindrical motors reported in literature, indicating an

acceptable performance of the flow solver module. Several motors with complex grain geometries are also analysed using this program. The geometrical evolution does not show any physical surface or singularity. The  $p$ - $t$  curves when compared with the experimental results show good match except for one case where 'hump effect' could be a reason.

In addition to  $p$ - $t$  curve, the output of the code gives the detailed axial flow field parameters as well as the geometries in the form of mesh file, which can be further used as input to codes for CFD analysis.

## REFERENCES

- Hartfield, R.; Jenkins, R. & Burkhalter, J. A review of analytical methods for solid rocket motor grain analysis. 2003, AIAA Paper No. 2003-4506. doi: 10.2514/6.2003-4506
- Anderson, M.; Burkhalter, J. & Jenkins, R. Design of a guided missile interceptor using a genetic algorithm. *J. Space. Rock.*, 2001, **38**(1), 28-35. doi: 10.2514/2.3668
- Anderson, M.; Burkhalter, J. & Jenkins, R. Missile shape optimization using genetic algorithms. *J. Space. Rock.*, 2000, **37**(5), 663-669. doi: 10.2514/2.3615
- Anderson, M.B.; Burkhalter, J.E. & Jenkins, R.M. intelligent systems approach to designing an interceptor to defeat highly maneuverable targets. 2001, AIAA Paper No. 2001-1123,
- Solid propellant grain design and internal ballistics. NASA SP 8076.
- Osher, S. & Sethian, J.A. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comp. Phy.*, 1988, **79**, 12-49. doi: 10.1016/0021-9991(88)90002-2
- Sethian, J.A. Level set methods and fast marching methods. Cambridge University Press, Cambridge, England, 1999
- Yildirim, C. & Aksel, M. H. Numerical simulation of the grain burnback in solid propellant rocket motor. AIAA Paper No. 2005-4160, 2005.
- Atilgan, T.K.; Tugrul, T.H. & Haluk A.M. Three-dimensional internal ballistic analysis by fast marching method applied to propellant grain burn-back. AIAA Paper No. 2005-4492, 2005.
- Willcox, M.A.; Brewster, M.Q.; Tang K.C. & Stewart D.S. Solid propellant grain design and burnback simulation using a minimum distance function. 2005, AIAA Paper No.2005-4350.
- Willcox, M.A.; Brewster, M.Q.; Tang K.C. & Stewart D.S. Solid propellant grain design and burnback simulation using a minimum distance function. *J. Propulsion Power*, 2007, **23**(2), 465-475.
- Gordon, S. & McBride, B.J. Computer program for calculation of complex chemical equilibrium compositions and applications – I. Analysis, 1994, NASA-RP-1311.
- Gordon, S. & McBride, B.J. Computer program for calculation of complex chemical equilibrium compositions and applications – II. Users Manuel and Program Description, 1996, NASA RP-1311.
- Mukunda, H.S. & Paul, P.J. Universal behavior in erosive burning of solid propellants. *Combustion Flame*, 1997, **109**, 224-236. doi: 10.4236/wet.2010.12009
- Mukunda, H.S.; Paul, P.J.; Javed, A. & Chakraborty, D. Extension of the universal erosive burning law to partly symmetric propellant grain geometries. *Acta Astronautica*, 2014, **93**, 176-181. doi: 10.1016/j.actaastro.2013.07.017
- Python. www.python.org (Accessed on 28 August 2014).
- Hasegawa, H.; Hanzawa, M.; Tokudome, S. & Kohno, M. Erosive burning of aluminized composite propellants: X-ray absorption measurement, correlation, and application. *J. Propul. Power*, 2006, **22**(5), 975-983. doi: 10.2514/1.7950
- Hasegawa H.; Fukunaga M.; Kitagawa, K. & Shimada, T. Burning rate anomaly of composite propellant grains. *Comb., Explos., Shock Waves*, 2013, **49**(5), 583–592. doi: 10.1134/S0010508213050109

## ACKNOWLEDGEMENTS

Authors wish to express their gratitude and acknowledgment of the initiative taken by Late Prof. PJ Paul, IISc. Bangalore, for the development of a universal internal ballistic code for solid rocket motors. Without his initiative this work could not have been a reality. The authors are also thankful to Mr M. Pandu Ranga Sarma, Sc 'F', Mr Sampath Kumar, Sc 'F', and Dr K.K. Rajesh, Sc 'E', from DOP, DRDL, Mr M. Rathnam Sc 'E' and Mr Guru Sudhakar, Sc 'C', from SPSC, ASL for providing the motor geometries and test data.

## CONTRIBUTORS

**Dr Afroz Javed** has done problem formulation, arrangement of geometries and test data from literature and rocket motor designers, and preparation of draft manuscript.

**Dr Arvind Sundaram Iyer** did coding of the program, generation of burn back geometries of different rocket motors at various instants and generation of pressure-time curves for different rocket motors.

**Dr Debasis Chakraborty** has done Overall planning and guidance of the work, review of the simulation results and suggest modification, and preparation of final manuscript.