A Comparison of a Modified Curvilinear Approach for Compressible Problems in Spherical Geometry and a Truly Spherical High-Order Method

M. El Rafei, L. Heidt and B. Thornber

Aerospace, Mechanical and Mechatronic Engineering
University of Sydney, NSW 2006, Australia

Abstract

The dynamics of a variety of compressible physical phenomena such as implosions and explosions occur in spherical geometries. An example application is inertial confinement fusion, where a small capsule containing nuclear material (Deuterium/Tritium mixture) is imploded at high velocities such as undertaken at the National Ignition Facility. In many of those applications, the physics of the flow is greatly impacted by asymmetries generated in the initial stages. Thus, reconstruction methods for these problems must be able to resolve accurately the growth of fundamental instabilities triggered by, for example, non-axisymmetric material properties or initial geometry. These initial asymmetries are so small that numerical generated asymmetries may swamp the physical perturbations.

Here a modified curvilinear approach is presented which reduces asymmetry in spherical meshes by more than an order of magnitude when compared to the standard approach, even when using Cartesian momenta. This is then compared to a true spherical solver written in spherical co-ordinates and which computes spherical momenta. High-order reconstructions were achieved using a modified spherical fifth-order MUSCL scheme that can be used for both uniform or non-uniform grids. Special attention was carried out in integrating the geometric source terms to avoid situations where a constant pressure and a zero velocity case is not recognized by the solver as a static solution. The performance of the adapted spherical reconstruction was investigated against the modified curvilinear approach and showed that the current approach avoids spurious oscillations at symmetry axis and yields enhanced results where initial physical instabilities are triggered by perturbations smaller than a specified cut off.

Introduction

Finite volume (FV) numerical methods have proved their ability to accurately model flows with strong discontinuities that could be encountered in many compressible physical phenomena in a conservative and oscillation-free way. Compressible solvers based on FV methods rely on computing the fluxes at the cell interfaces from the solution of a Riemann problem which represents a discontinuity in the solution between the left and right states of a specified interface. Those states are reconstructed from the volume-averaged quantities using several reconstruction schemes that are vastly documented in the literature and the reader is referred to [1, 2, 3] for more details regarding FV methods, Riemann solvers and reconstruction techniques. Originally, reconstruction methods were derived for Cartesian domains that are based on uniform grids and little attention was carried out for curvilinear geometries [5]. As a result, Cartesian reconstruction techniques were applied to arbitrary curvilinear grids. One of the ways to achieve that is by using a Jacobian to map the curvilinear grid to a uniform Cartesian numerical grid [4]. However, this approach may not be able to represent accurately some physical phenomena present in the physical grid. Hence, it is more advantageous to use the reconstruction schemes in their original coordinates rather than per-
cality of dissipation at low Mach number [15]. The Harten, Lax and Leer Contact (HLLC) approximate Riemann solver [16] is used to solve the Riemann problem and the second-order total variation diminishing Runge-Kutta scheme [17] is adopted for time integration.

The true spherical solver implemented in FLAMENCO uses spherical velocity components \((v_r, v_\theta, v_\phi)\), where \(r, \theta\) and \(\phi\) are the radial, polar and azimuthal directions respectively. This numerical algorithm is devised in such a way that exact cell volumes and face areas are introduced within the governing equations to preserve the required symmetry in a conservative framework. It should be noted that the vector basis in which the vector components are expressed in spherical coordinates is a local vector basis, which means that the latter has no fixed orientation in space. Hence, additional source terms appear in the momentum equations and these terms should be integrated in the conservative approach. The \(r, \theta\) and \(\phi\) source terms added to the momentum equations could be written as:

\[
S_r = \frac{2P}{r} + \frac{\rho (v_\theta^2 + v_\phi^2)}{r} \tag{1}
\]

\[
S_\theta = -\left[ \frac{\rho v_\theta v_r}{r} - \frac{\cos \theta \rho v_\phi^2 + P}{\sin \theta} \right] \tag{2}
\]

\[
S_\phi = -\left[ \frac{\rho v_\phi v_r}{r} + \frac{\cos \theta \rho v_\theta v_\phi}{\sin \theta} \right] \tag{3}
\]

Furthermore, the fifth-order MUSCL scheme used for variables reconstruction is modified so that it uses spherical coordinates rather than Cartesian transformed coordinates. This modification allows the use of the fifth-order MUSCL scheme on arbitrary uniform and non-uniform grids since the space increment in all spatial directions are taken into account. Hence, the use of a Jacobian to map the curvilinear grid into a uniform Cartesian grid in order to use the original Cartesian formulation of the reconstruction scheme is no more necessary.

In regards to the classical approach where FLAMENCO uses Cartesian momenta on curvilinear grids, two fixes are implemented to reduce the asymmetry that is introduced due purely to numerics. Before going into the details of the new fixes, it should be noted that the curvilinear metrics being used to transform the Cartesian momenta into Spherical momenta are second-order accurate and not exact. Hence, the induced discretization errors could affect the symmetry of the conversion between Cartesian and spherical components. Considering for example the radial direction, the radial cell-centred velocity \(u_r\) is made up from the Cartesian components \(u, v, w\) in three-dimensions, which represent the velocity components in \(x, y\) and \(z\) directions respectively. This transformed radial velocity component is analytically perfectly radial but it is not perfectly aligned with the numerical face normal which creates a kind of misalignment and hence could be considered as a source of asymmetry. Moreover, the radial fluxes computed at the cell interfaces are based on face-centred conversions and thus, one could notice that two different methods are used to compute the transformed curvilinear components, one is face-centred and the other is cell-centred. Accordingly, a slight angle difference is introduced which affects the symmetry of the problem as shown in figure 1 where \(i, j\) represent the radial and azimuthal directions respectively and \(F\) is the Flux at the interface. One fix of this issue is to enforce numerically symmetric velocities in the direction where the flux is computed. In other words, the velocity is enforced to be aligned with the face normal numerically and not analytically. This is made by ensuring that the fluxes in the directions where the flux is not computed are kept to zero values.

Another issue that should be addressed is that when the solution is evolved in time a division by the cell volume is applied. Considering a uniform grid, the ratio between the volume and face area between different cells is ideally identical. However, using coordinates transformation, a second-order estimation of the cell-volume is considered which induces discretization errors and the symmetry of the numerical solution is slightly affected. A second fix of this problem is to ensure that the ratio of the volume to face area is constant in the polar direction where slight changes were observed. This is achieved by taking the mean of the maximum and minimum values of this ratio and fixing this value in the computational cells. This fix introduced a small modification of 0.012% to the ratio of volume to face area. These modifications improve significantly the performance of the curvilinear algorithm and preserve the symmetry of the problem.

In the following, a brief description of the setup used to perform ideal spherical implosions is introduced. The physical model is similar to the one in [18, 19, 20], where the computational domain is split into three regions: the gas region that represents the inner section of the capsule, the shell region which represents the outer shell of the capsule and the source region. The source region is used to incorporate the effect of the shock waves and time-varying physical conditions are prescribed to drive the implosion inwards in time. This is done by applying a global velocity gradient written as \(-v_0x\), where \(v_0\) is a constant and \(x\) is the radial position of a computational cell. Low convergence cases with \(v_0 = 0.6/\text{ns}\) are adopted for the current simulations. More details about the initial conditions used for the spherical implosion setup are represented in table 1.

<table>
<thead>
<tr>
<th>(\rho [\text{g/cm}^3])</th>
<th>Source</th>
<th>Shell</th>
<th>Gas</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>1.00</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>(p [\text{Mbar}])</td>
<td>1000</td>
<td>10.00</td>
<td>10.00</td>
</tr>
<tr>
<td>(v_r [\text{mm/ns}])</td>
<td>(-v_0r)</td>
<td>(-v_0r)</td>
<td>(-v_0r)</td>
</tr>
<tr>
<td>(r [\text{mm}])</td>
<td>1.50</td>
<td>1.20</td>
<td>1.00</td>
</tr>
<tr>
<td>(\gamma [-])</td>
<td>5/3</td>
<td>5/3</td>
<td>5/3</td>
</tr>
</tbody>
</table>

Table 1: Initial conditions of the spherical implosion case.
In the low convergence case, the shell and gas regions have a zero velocity gradient \( -v_0 r = 0 \). However, the source region still has a velocity gradient of 0.2\(n_s^{-1} \) applied to it. The perturbation consists of a single spherical harmonic mode that could be described as:

\[
\delta R(\theta) = a_0 Y_0(\theta)
\]

where \( a_0 \) is the amplitude of the harmonic and \( l \) is the mode number. It should be noted that \( a_0 \) is assumed to be considerably small compared to the wavelength of the harmonic. The computational domain extends from 0.05mm to 1.5mm in the radial direction and from \([0, \pi]\) in the polar or \( \theta \) direction. A small cut-out is introduced in the region where the origin is located to reduce the computational cost.

Results and Discussions

In this section, qualitative and quantitative analyses of the results given by each algorithm used in FLAMENCO are presented. Qualitatively, a study of a zero-perturbation case is performed which helps in understanding if the interface between the gas and shell regions remains symmetric and free of small-scale numerical perturbations at late time in the simulation. Quantitatively, the perturbation amplitude that is defined as the half of the difference between the maximum and minimum interface radius is presented. A low mode number \( l = 5 \) is considered along with a perturbation amplitude of \( a_0 = 0.001 \). The choice of this amplitude value is based on the fact that small-scale numerical perturbations are more discernable using low mode perturbations. A low convergence case and a high convergence case characterised by a velocity gradient \( v_0 = 0.6 \, n_s^{-1} \) are considered for the computation of the perturbation amplitudes. For more validation, the FLAMENCO results are compared as well to those obtained using PLUTO astrophysical code [21] to understand the performance of our spherical algorithm compared to another well-documented and validated spherical solver. In regards to PLUTO setup, the third-order WENO scheme is used for variable reconstruction and hence a finer grid resolution is adopted in PLUTO \((3072 \times 3072)\) for more consistency in the comparisons. The second-order Runge-Kutta scheme is used for time integration. Details about the grid resolutions adopted in FLAMENCO and the domain sizes are presented in table 2.

<table>
<thead>
<tr>
<th>Grid Geometry</th>
<th>Spherical Solver</th>
<th>Curvilinear Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Size</td>
<td>([0.05, 1.5] \times [0, \pi])</td>
<td>([0.05, 1.5] \times [0, \pi])</td>
</tr>
<tr>
<td>Resolution</td>
<td>1536 (\times) 1536</td>
<td>1536 (\times) 1536</td>
</tr>
</tbody>
</table>

Table 2: Grid geometry, resolution and domain size used for each numerical algorithm implemented in FLAMENCO code.

In terms of computational time, it is important to mention that the curvilinear solver was 1.2 times slower than the spherical solver which induce a considerable reduction in the computational cost. Density contours of a high convergence, zero perturbation case \((v_0 = 0.6 \, n_s^{-1})\) given by the different codes are shown in figure 2. The results given by all solvers are similar in terms of density distribution and shock position. However, the morphology of the small-scale instabilities is significantly different. One could obviously notice that the spherical solver of FLAMENCO remains qualitatively perfectly symmetric with no obvious numerical noise or secondary small-scale instabilities, similarly to the density distribution of PLUTO. In contrast, the interface between the gas and shell regions is overwhelmed with small instabilities that are purely due to numerics using FLAMENCO’s curvilinear solver. Quantitatively and based on the maximum and minimum perturbation amplitudes, we estimate approximately a 99.8% reduction of small-scale numerical instabilities using FLAMENCO’s spherical solver at very late times. Accordingly, a curvilinear solver is not accurate enough for very low amplitudes as can be seen in figure 2. It is interesting to note that a quantitative comparison between the perturbation heights given by PLUTO and FLAMENCO spherical solver in the zero perturbation case showed that the amplitude of the perturbation was \(1/120\) the size of the cell which means that numerically seeded instabilities are negligible. Figure 3 represents a comparison of the perturbation amplitude growth between the different algorithms used in FLAMENCO and PLUTO spherical algorithm. A mode number \( l = 5 \) and an initial perturbation amplitude \( a_0 = 0.001 \) are considered for low and high-convergence setups \((v_0 = 0 \, n_s^{-1}, v_0 = 0.6 \, n_s^{-1})\).

![Figure 2: Density contours at \(t = 1.2 \, n_s\) using the spherical and curvilinear FLAMENCO solver.](image)

![Figure 3: Variation of the perturbation amplitude in time for the low and high convergence cases.](image)
The spherical algorithm.

ter the fluxes with the face normals in each direction. Ideal
nifications to the estimation of the cell volume and to the align-
try of the spherical problem is preserved by introducing modi-
coordinates into uniform Cartesian coordinates and the symme-
the curvilinear algorithm computes Cartesian momenta by mapping the curvilinear
that is adapted for uniform grids. The second curvilinear algo-
ment fusion cases. The spherical algorithm of FLAMENCO
solves the spherical Euler equations and computes spherical
equations due to the local vector basis used in spherical
algorithms. A zero perturbation case showed that the interface
between the gas and shell regions remained perfectly symmet-
with no discernable mixing or secondary instabilities when
computed using the spherical solver of FLAMENCO. However,
significant numerical instabilities were present at the interface
using the curvilinear solver. It is interesting to mention that the
perturbation amplitude was reduced by 99.8% in the zero perturbation case with FLAMENCO spherical solver. Quantita-
tively, the perturbation amplitude of our spherical solver had an important agreement with PLUTO [21] spherical solver until re-shock where FLAMENCO had higher amplitudes, whereas the curvilinear solver had significantly larger amplitudes com-
pared to PLUTO. This indicates that the spherical algorithm
was able to prevent imminent numerical instabilities that could
swamp the physical perturbations. Another important result of
this study is that a curvilinear solver is not accurate for low per-
turbation cases. Finally, the curvilinear solver of FLAMENCO
was 1.2 times slower than FLAMENCO spherical solver which
presents an important reduction in the computational time for the spherical algorithm.

Conclusions

This paper presented two different numerical algorithms used to compute the dynamics of compressible physical phenomena that occur in spherical geometries such as in inertial confinement fusion cases. The spherical algorithm of FLAMENCO solves the spherical Euler equations and computes spherical momenta that are aligned with the direction of the flux in each spatial direction. Additional Source terms appear in the momentum equations due to the local vector basis used in spherical coordinates. The fifth-order MUSCL scheme was modified in such a way it uses the true spherical coordinates. This is a novelty compared to the classical formulation of this scheme that is adapted for uniform grids. The second curvilinear algorithm computes Cartesian momenta by mapping the curvilinear coordinates into uniform Cartesian coordinates and the symmetry of the spherical problem is preserved by introducing modifi-
cations to the estimation of the cell volume and to the align-
ment of the fluxes with the face normals in each direction. Ideal spherical implosions with a mode number \( l = 5 \) were considered within this study. The morphology of the small-scale numerical secondary instabilities was quite different between the two algorithms. A zero perturbation case showed that the interface between the gas and shell regions remained perfectly symmetric with no discernable mixing or secondary instabilities when computed using the spherical solver of FLAMENCO. However, significant numerical instabilities were present at the interface using the curvilinear solver. It is interesting to mention that the perturbation amplitude was reduced by 99.8% in the zero perturbation case with FLAMENCO spherical solver. Quantitatively, the perturbation amplitude of our spherical solver had an important agreement with PLUTO [21] spherical solver until re-shock where FLAMENCO had higher amplitudes, whereas the curvilinear solver had significantly larger amplitudes compared to PLUTO. This indicates that the spherical algorithm was able to prevent imminent numerical instabilities that could swamp the physical perturbations. Another important result of this study is that a curvilinear solver is not accurate for low perturbation cases. Finally, the curvilinear solver of FLAMENCO was 1.2 times slower than FLAMENCO spherical solver which presents an important reduction in the computational time for the spherical algorithm.

References

[5] Ziegler, U., A semi-discrete central scheme for magneto-
[10] Larrouturou, B., How to preserve the mass fractions positivity when computing compressible multi-component flows, INRIA Rapports de Recherche No. 1080.
[17] Spiteri, R.J and Ruuth, S.J., A new class of optimal high-
order strong-stability preserving time discretization meth-