

Numerical simulation of explosion phenomena in industrial environments

T. Huld*, G. Peter, H. Städtke

*Joint Research Centre, Inst. for Safety Technology, Process Engineering Division, Ispra Establishment,
I-21020 Ispra (Varese), Italy*

Received 28 April 1994; accepted 27 January 1995

Abstract

The code REACFLOW developed at the JRC Ispra combines advanced numerical techniques for the simulation of transient, multi-dimensional, multi-component gas flows undergoing chemical reactions to a unique tool. It uses a true 2-D discretisation with an unstructured triangular grid to ensure a maximum of flexibility for the representation of complex geometries. The numerical discretisation uses a finite volume scheme based on an approximate Riemann solver. Explicit, implicit and semi-implicit methods cover the whole range of time scales. Compressible and incompressible flow is treated with an arbitrary number of components. Chemical reactions are calculated fully implicitly. Diffusion processes are also modelled using a finite volume equivalence to the finite element Galerkin method. A $k-\varepsilon$ turbulence model is currently being implemented. A system for dynamic grid adaptation automatically detects locations of refinement and coarsening based on local gradients of flow variables. The code capability will be demonstrated by various applications, including a hydrogen/air explosion in a containment and a 'tulip' flame calculation.

Keywords: Combustion; Explosion; Fluid dynamics; Reactive flows; Detonation

1. Introduction

Understanding, predicting and modelling the heat and mass transfer processes in a postulated accident is an important issue to prove the safety of an industrial plant. It helps to predict the thermal and mechanical loads on structures due to the combustion of vapour clouds, gas explosions, and flame front and detonation/blow wave propagation. In this context it may be important to investigate the conditions under which a flammable mixture may form as a consequence of a fuel release and the

* Corresponding author. Tel.: +39-332-785273, +39-332-789216, +39-332-789898. Fax: +39-332-785869.

redistribution of gaseous components by free convection and species diffusion for a given geometry. The efficiency of safety devices such as explosion vents, flame proof enclosures, flame arresters, etc. may be analysed taking into account the nature of the fuel (e.g. hydrogen hydrocarbon mixtures), the heterogeneity of the explosive mixture and in particular the geometry of the containment (obstacles, multi-compartment, etc.). For this purpose the code REACFLOW has been developed at the JRC Ispra, which combines advanced numerical simulation techniques for the prediction of transient, multi-dimensional, multi-component gas flows undergoing chemical reactions.

2. Structure of the numerical solver

The basic equations to be solved are the equations imposing the conservation of mass, momentum and total energy (see Refs. [1, 2] and practically any fluid dynamics textbook). These are time-varying non-linear differential equations. The terms in the equations can be divided into four categories:

- the transient term giving the rate of change of the conserved quantity,
- convective terms describing the flow due to pressure differences and velocity,
- diffusive terms, covering heat conduction, molecular diffusion and viscosity,
- source terms, in this case chemical source terms and gravity.

In an integral formulation a conservation equation will have the following general form where q is the conserved quantity and the integration is over the volume Ω with the boundary $\partial\Omega$

$$\frac{\partial}{\partial t} \int_{\Omega} q \, dV + \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{F}_{\text{conv}} \, dS + \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{F}_{\text{diff}} \, dS + \int_{\Omega} P \, dV = 0.$$

Here \mathbf{n} is the outward normal to the volume. \mathbf{F}_{conv} and \mathbf{F}_{diff} are convective and diffusive fluxes, respectively, and P is the source term. The advantage of this formulation is that once the fluxes at each boundary segment are calculated they can be subtracted directly from the amount of the conserved variable in the given control volume and added to the neighbouring volumes. Thus, in the absence of source terms the total amount of the conserved variable is automatically conserved.

3. Discretisation of the conservation equations

The transient term is discretised by the Euler step method. The time is divided into discrete steps of varying size, Δt^n , and the partial derivative with time becomes

$$\frac{\partial q}{\partial t} \cong \frac{q^{n+1} - q^n}{\Delta t^n}.$$

The plane is discretised by dividing it into triangular areas as shown in Fig. 1. A *control volume* is defined around each vertex (node) bounded by the medians of the

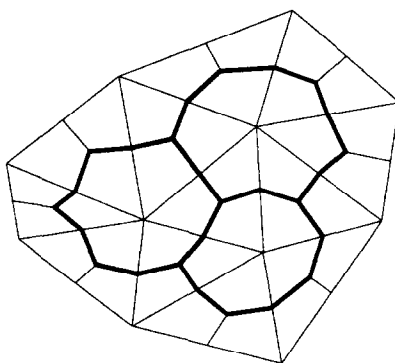


Fig. 1. Triangular grid and the corresponding control volumes.

adjacent triangles (Fig. 1). The variables to calculate at each time step are then the averages over each control volume of the conserved variables.

The time scales of the processes described in the problem vary widely, with some of the elementary chemical reactions having time scales up to 10 orders of magnitude smaller than the diffusive processes. Therefore different methods will be used to deal with the convective, diffusive and source terms. They are solved separately by an *operator splitting procedure*: starting with the state q^n at a given time t^n a problem which consists only of the transient term and convective term resulting in an intermediate state q^* is solved, then from this new state the problem of the transient + diffusive terms is solved and finally the transient + source terms problem is solved to reach the state at the end of the n 'th time step. The convective and diffusive steps may be solved simultaneously. Each of these steps will be described in more detail below.

4. Solution of the convective problem

In each control volume only the average of each conserved variable is known. In order to find the values of the flux at the edge of the control volume the conserved variables are either assumed to be constant (first-order calculation) or varying linearly (second order) over the control volume. This implies that the fields of conserved variables in general will be discontinuous across the boundaries. The problem of a discontinuity dividing two regions with constant values is termed a *Riemann problem*. Various methods exist for solving such a problem numerically for a system of conservation laws [3–5]. These will not be described in detail here. The fluxes may be evaluated either at the present time step (explicit method) or at the next time step (implicit method). In the first case the fluxes may be calculated directly and the state updated accordingly. In the implicit case the fluxes depend on the unknown state at the next time step. The result is a set of $N \times (T + 2 + 1)$ coupled algebraic equations to be solved. N is the total number of control volumes and T is the number of different chemical species. The 2 comes from the equations for the two components of the

momentum and the 1 from the single equation for the energy. These equations are non-linear but may be linearised by Taylor expansion to first order. The resulting set of linear equations may be solved using an iterative procedure, for example, a Gauss–Seidel method. The implicit method is much more time-consuming but has greater stability and allows larger time steps. Generally, the explicit solver is suited for fast transonic flows whereas the implicit solver is better adapted for subsonic flows.

5. Diffusive terms

The diffusive part describes internal heat conduction, heat conduction from walls, viscosity and molecular diffusion. The diffusive terms depend on the gradients of various quantities that are functions of the conserved variables. For instance the heat conduction is proportional to the gradient of the temperature. In the discontinuous representation of the convective solver these gradients are not defined at the boundaries of the control volumes. Therefore a different approach is needed. For the diffusive solver the conserved variables are assumed to be linear functions on each triangle. The values at the vertices can be set equal to the average values in the control volume surrounding the vertex. This is known as *mass lumping* and means that the unknowns are the same as in the convective problem. Then the two problems may be solved together. Alternatively the vertex values may be calculated in such a way that the integral of each conserved variable remains constant on the control volume. This is in general a more accurate approach, but the unknowns are no longer the same as the unknowns in the convective problem, and the two problems must be solved separately [6].

Here too, both explicit and implicit solvers have been developed. Again, the explicit solver is the fastest but least stable. If the time step size exceeds the Fourier limit the system will become linearly unstable and the solution will blow up in a few time steps.

6. Chemical solver

The chemical solver can handle an arbitrary number of elementary chemical reactions, provided only that they obey van't Hoff's law and that the reaction kinetics can be described by Arrhenius' law. These laws are strongly non-linear, so here a solution must be found to the full non-linear algebraic equations. The chemical source terms depend only on the state at every point, not on the gradients and therefore, the problem becomes a set of ordinary differential equations. This can be solved separately in each control volume. Here it is assumed that the state is constant within each control volume. In each control volume one equation must be solved for each active chemical species (i.e. those involved in and changed by the chemical reactions) plus one for the internal energy. The number of equations can then be reduced by noting that the total number of atoms of any given species must remain constant (nuclear reactions are not considered). The total number of equations is then

$\Gamma - \Gamma^A + 1$ where Γ^A is the number of active atomic species. The system of equations is solved by an iterative method similar to the classical Newton–Raphson method [7].

If necessary, special modules can be developed for specific sets of chemical reactions. This may be done for various purposes. A dedicated module can be made to run considerably faster than the general solver, in the eight-reaction hydrogen–air scheme shown below, a factor 2 was attained. In addition a special module may allow deviations from the strict implementation of van't Hoff and Arrhenius' laws. The drawback is of course the time needed to develop a dedicated module. At present only two modules have been developed: one for an eight-reaction hydrogen burning scheme [8] and one for a two-reaction hydrogen burning scheme of Rogers and Chinitz [9]. This last one uses a modification of Arrhenius' law to compensate for the severe simplification of using only two chemical reactions to describe hydrogen combustion.

7. Grid adaptation

The grid adaptation module SANGRIA (System for Automatic Non-structured Grid Adaptation) consists of four steps: detection, refinement, coarsening, interpolation. It has been developed for two-dimensional, unstructured, triangular grids which are used by REACFLOW. The key idea behind SANGRIA is to keep as much as possible the original grid structure during refining and coarsening and to find back exactly to the original grid structure once all refined nodes have been taken away [10].

The detection of the locations where additional nodes have to be added or where nodes can be taken away is completely automatic. It is based on local gradients of a variable, that can be any variable which the code provides or even a combination of variables. Typically the pressure, the density, the Mach number or the temperature is chosen. From this variable the gradients from every node to its neighbour nodes are calculated. If the gradient exceeds a critical value the grid refinement is triggered. If the gradient, however, is flat and has kept this flatness for a certain amount of time then the coarsening is triggered.

The grid refinement algorithm in SANGRIA aims to put a new node exactly in the middle between the two nodes, between which the local gradient has exceeded the critical value. This means dividing the section which connects these two nodes and to divide also the two triangles (T_1 , T_2) which have this section in common (Fig. 2). If the section which has to be divided is the longest section of both triangles (T_1 and T_2) the operation results in less flat triangles. In all other cases the triangles become flatter, which is not desirable. Therefore an algorithm which tries to avoid this effect has been implemented: If the section which should be divided is not the longest section of T_1 and T_2 , then it is attempted to divide the longest section of T_1 and T_2 , but only if this does not result in a flatter triangle. If this is not possible then the flattening of the triangles is accepted and the division of T_1 and T_2 is performed.

To make the whole operation reversible (see grid coarsening), two pieces of information are stored for every refining process: The node history (h_1) and the information on which nodes belong to the section which is being divided (h_2 , h_3). The

node history is equal to the node number for the original grid and is incremented by 1 for every refining operation. The higher is the number the more recent is the node. The history information is illustrated in Fig. 3, where a simple four-node grid is successively refined to a 13-node grid. The history information after the refining steps is shown in Table 1. The node number does not correspond necessarily to the history information h_1 , especially after a certain number of refining and coarsening operations!

The basic constraint for SANGRIA's coarsening algorithm is to preserve the original grid and to make the refining process fully reversible. When the local gradient of the sensitivity variable around one node is flat for a certain period, the node is flagged as 'possible to delete'. The coarsening algorithm checks with the node-wise

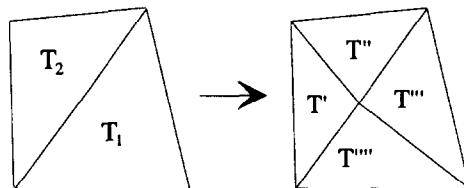


Fig. 2. Grid refinement in SANGRIA.

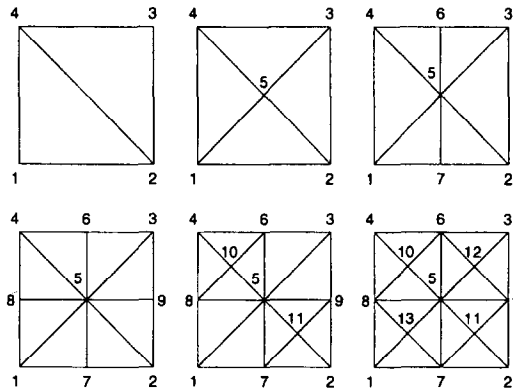


Fig. 3. Refining example with history information.

Table 1
History information after refining example

Node No.	1	2	3	4	5	6	7	8	9	10	11	12	13
h_1	1	2	3	4	5	6	7	8	9	10	11	12	13
h_2	0	0	0	0	2	4	1	1	2	4	2	3	1
h_3	0	0	0	0	4	3	2	4	3	5	5	5	5

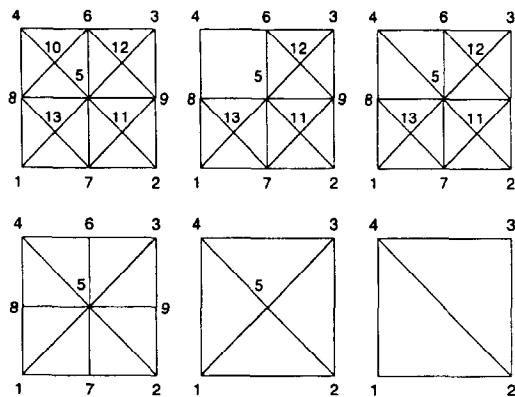


Fig. 4. Coarsening example with history information.

history information, if the nodes around the flagged node are older than itself. If this is the case, it can be deleted without conflicting with the reversibility constraint. With the second history information (which nodes have to be connected if this node is taken away) it is possible to restore the situation before the refining step was performed. In Fig. 4 the refined grid of Fig. 3 is coarsened by one node. Only node number 10, 11, 12 or 13 can be taken away, because the h_1 information shows, that only these nodes are surrounded only by nodes which are 'older' than themselves. Node number 10 is taken away and with it all four triangles which have this node in common. The empty space is filled by a connection between two nodes. h_2 and h_3 provide node number 4 and 5. Analogously also node 11, 12 and 13 can be taken away to arrive to the first picture in the second row. From there node numbers 6–9 can be taken away to arrive to the next picture. Finally node 5 can be taken away to leave a blank square. From the history information of node number 5 we retrieve the information that node numbers 2 and 4 have to be connected. And we are back to the origin.

Adding nodes or taking them away always require local interpolation of data, because the fluid properties are distributed on single elements, whose number has been changed. Since the changes are only local, it is sufficient to correct the local values of the surrounding nodes. This has to be done conserving mass, momentum and energy by balancing the properties before and after refining or coarsening processes.

8. Graphics

An Interactive System for On-Line Data Animation (ISOLDA) provides the possibility to control the ongoing calculation. ISOLDA can produce spectral plots, contour plots and vector plots. It is based on a device independent graphic library DIGLIB [11]. The post-processing of the data is done with a graphics package called TURCOM (TURbulent COMbustion) [12]. It produces spectral graphs, iso-contour-graphs, vector graphs, both with and without overlaid mesh at different time steps as well as x/y graphs of variables over time or variables over a cut in the domain. Various

colour tables can be chosen. Zooming in time as well as the automatic sequence of plots over time is possible. Macros can be created interactively by the user using a macrorecorder.

9. Hydrogen/air explosion

For demonstrating the code capabilities a two-dimensional calculation of a hydrogen/air explosion is presented. A closed rectangular area of 1 m × 0.5 m is filled with

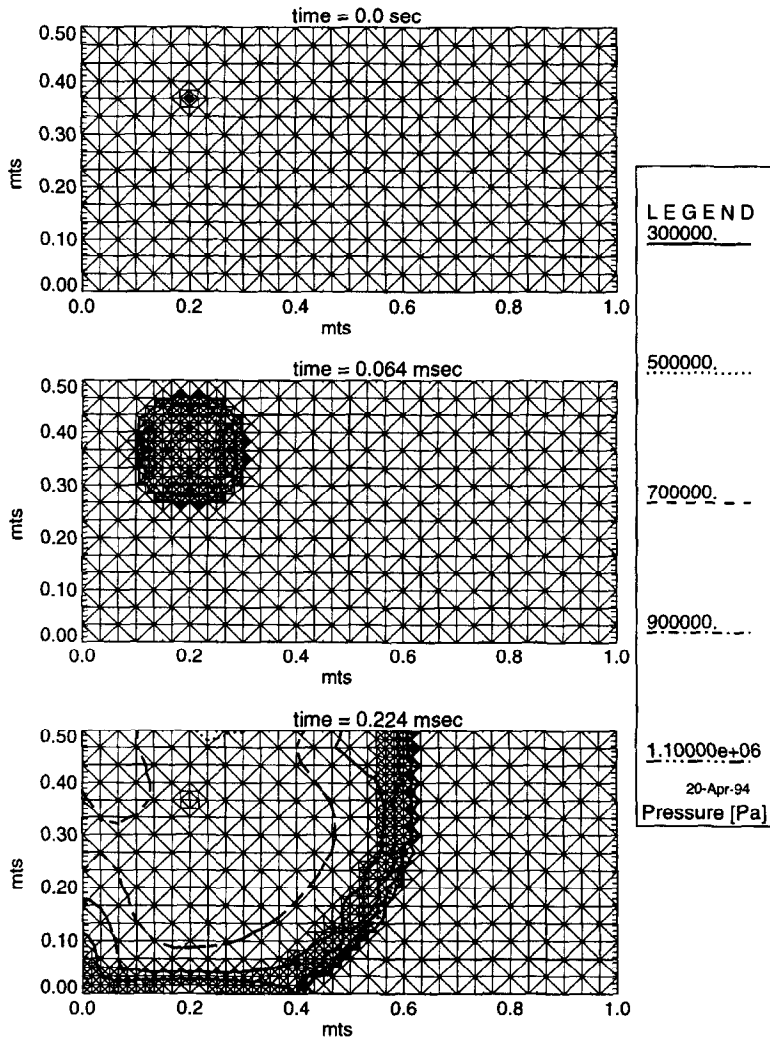


Fig. 5. Time sequence of pressure distribution of a hydrogen explosion in a container.

a mixture of dry air and hydrogen so that the hydrogen is 0.9 times stoichiometric with respect to the oxygen. The chemical scheme uses 8 chemical reactions involving the six components (O_2 , H_2O , H_2 , OH , O and H) [7, 8]. N_2 does not take part in the chemical reactions but influences them as a third body. The initial state is: pressure 1 bar, temperature 293 K everywhere, except for the ignition region where the pressure is 5 bar and the temperature 1500 K. The hydrogen burning in this region causes a detonation wave to propagate through the flammable mixture. A dynamically adaptive grid is used based on a t_0 grid of 504 nodes. The number of nodes is automatically blown up to about 2000 nodes using the local pressure gradient as criterion for refining and coarsening. The numerical scheme is time and space second-order explicit van Leer flux vector splitting. Diffusive processes are not taken into account. Fig. 5 shows the iso-contour lines for the pressure at three different time steps. After the ignition the pressure wave expands radially symmetric until it hits the wall where it is reflected. The adaptive grid follows the pressure wave, refining near the pressure front and restoring the original grid where the pressure front has passed.

10. Flame propagation in a tube

A second demonstration of the capabilities of REACFLOW is the calculation of a so-called 'tulip' flame in a closed vessel. The vessel has dimensions $0.16 \text{ m} \times 0.04 \text{ m}$ and initially is filled with a mixture of hydrogen and dry air at ambient pressure and temperature. The stoichiometric ratio of hydrogen to oxygen is 0.5 ($[\text{H}_2] = [\text{O}_2]$). The chemical scheme is the eight-reaction scheme mentioned above [8]. As a very crude modelling of turbulence the heat conduction coefficient is set to $\lambda = 2.0 \text{ W K}^{-1} \text{ m}^{-1}$ for all chemical species. There is no wall friction. The ignition is done by initially setting two control volumes near the middle of the bottom wall to 1100 K, 1 atm. The initial grid has $16 \times 62 = 992$ nodes. By dynamic grid adaptation this is increased to about 1400 nodes before the flame reaches the side walls, then it decreases with the size of the flame front. The numerical scheme used is the fully implicit second-order Roe's approximate Riemann solver.

Initially the flame expands almost as a semicircle (see Fig. 6). Then when it reaches the walls it elongates. As the flame burns out at the walls it accelerates in the near-wall region, creating a tulip-like shape. This is thought to be due to the creation of vortices at the walls which convect the flame forward at the walls [13]. The flame front may be further distorted by a Rayleigh–Taylor instability due to the large difference in density between the burned and the unburned gas.

11. Future developments

The 2-D version of REACFLOW will be completed by the end of 1994. The verification against experiments and other codes will be continued. The pre- and post-processor capabilities will be further improved according to needs of future users.

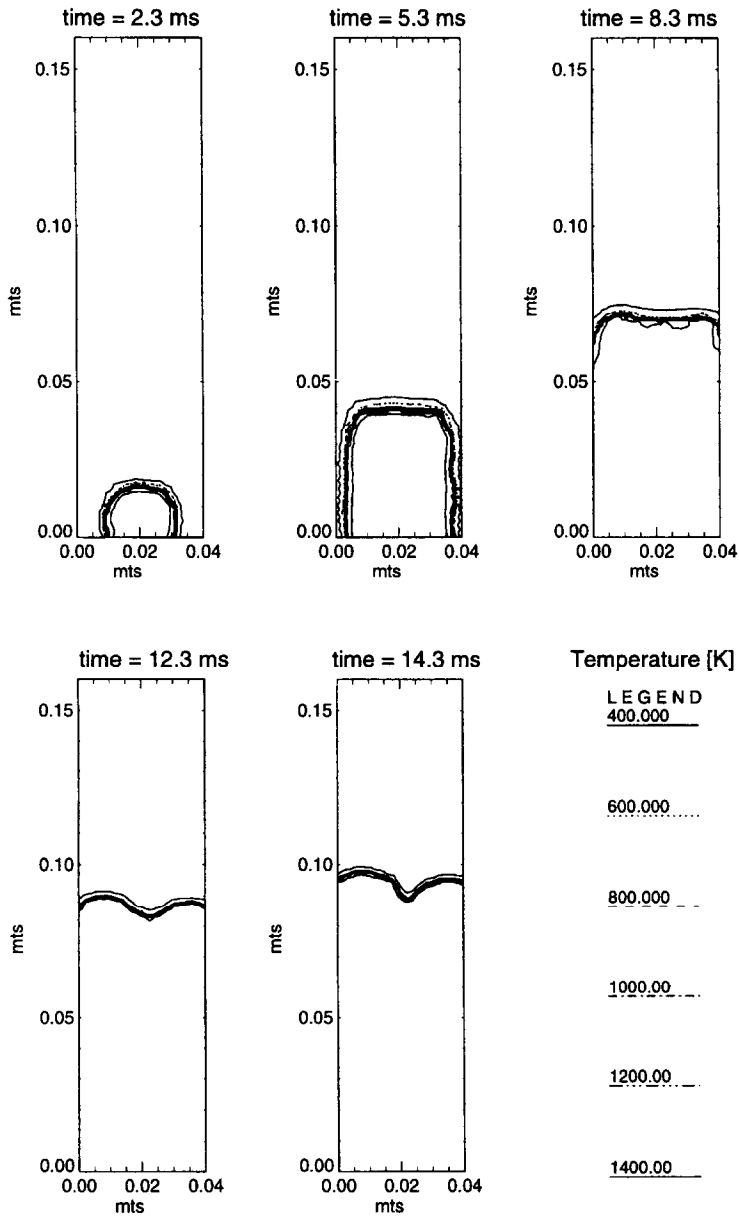


Fig. 6. Time sequence of temperature distribution of a tulip-flame in a closed tube

An extension to particle–gas mixtures like dust and sprays seems to be very interesting. Also a 3-D version could be envisaged.

12. Conclusion

A computer program for calculating the behaviour of compressible and incompressible gas flows in a two-dimensional area has been presented. An arbitrary number of components can react following an arbitrary chemical reaction scheme. Implicit, explicit and semi-implicit numerical solvers cover a wide range of time scales. Diffusion processes are calculated fully implicitly. A automatic and fully reversible grid adaptation system automatically adapts the grid to the flow behaviour. Calculations of a hydrogen/air explosion and of a ‘tulip’ flame have shown the capabilities of the code.

References

- [1] L.D. Landau and E.M. Lifshitz, *Fluid Mechanics*, Pergamon, Oxford, 1959.
- [2] R. Courant and K.O. Friedrichs, *Supersonic Flow and Shock Waves*, Interscience Publishers, New York, 1948.
- [3] P.L. Roe, *J. Comput. Phys.*, 43 (1981) 357.
- [4] Y. Liu and M. Vinokur, *J. Comput. Phys.*, 83 (1989) 373.
- [5] J.L. Steger and R.F. Warming, *J. Comput. Phys.*, 63 (1981) 263.
- [6] F. Ruel, *Numerical Simulation of Reacting Gas Flows: Basic Models and Algorithms. Presentation of a Preliminary Code Version*, JRC Technical Note I.91.109, September 1991.
- [7] T. Huld, *An Implicit Chemical Kinetics Solver for Multidimensional Reactive Flow Simulations. Application to Hydrogen Combustion Schemes*, JRC Technical Note I.91.155, December 1991.
- [8] J.S. Evans and C.J. Schexnayder, *A.I.A.A. J.*, 18 (1980) 188.
- [9] R.C. Rogers and W. Chinitz, *A.I.A.A. J.*, 21 (1982) 586.
- [10] G. Peter, *SANGRIA: A System for automatic, dynamic, fully reversible grid adaptation (in preparation)*.
- [11] G. Peter, *DIGLIB: A Device Independent Graphics Library*, JRC Technical Note I.93.102, July 1993.
- [12] D. Roebelen, C. Best, *Extension of TURCOM Visualization of Reacting Gas Flows on Non-Structured Grids*, JRC Technical Note I.93.134, October 1993.
- [13] D. Dunn-Rankin, *Interaction between a Laminar Flame and its Self-generated Flow*, Lawrence Berkeley Lab. LBL-19582, 1985.