MECHANIKA TEORETYCZNA
I STOSOWANA
Journal of Theoretical
and Applied Mechanics
3, 35, 1997

3D h-ADAPTIVE FINITE ELEMENT SIMULATIONS OF INVISCID AND VISCOUS FLOWS

KRZYSZTOF BANAŚ

Section of Applied Mathematics, Cracow University of Technology

e-mail: Krzysztof.Banas@pk.edu.pl

The paper presents an adaptive finite element method for solving compressible fluid flow problems with a special emphasis on techniques of mesh refinements. A specially designed for 3D viscous flows strategy of applying h-refinements, both anisotropic in boundary layers and isotropic elsewhere, together with underlying technical tools and assumptions, is presented. Examples of application to 3D viscous and inviscid supersonic flow problems are included.

Key words: adaptive finite element method, compressible fluid flow

1. Introduction

When solving typical transonic and supersonic flow problems it is common that the regions with rapid changes in state variables occupy only a small part of the whole computational domain while in a major part of it the flow is either slowly changing or uniform. In such cases good resolution of a numerical scheme is necessary only in these parts of the computational domain where changes occur while for the rest of the domain the scheme can be less accurate still keeping the overall error on the same level. Such selective accuracy can be obtained by means of adaptivity built into a computer code. Usually its implementation comprise two independent parts: the first devoted to selection of places where increased accuracy is necessary (here a reliable error indicator is of primary importance) and the second providing technical tools for improving approximation.

The technique of h-adaptivity consists in reducing the size of a mesh (the distance between the neighbouring grid points) locally by introduction of new grid points (nodes). Ideally, this procedure should lead to the optimal approximation where some assumed error level is achieved with the least possible number of nodes in the mesh. This however would require the use of local *a posteriori* error estimation quantitatively expressing how by locally decreasing h one actually decreases the error of computations. No such error estimation exists till now for the compressible Navier-Stokes equations. However, for practically useful adaptation we often require only some error indicator – a value computed at each point of a flow field, which allows one to distinguish between the regions with small and big errors, respectively. Still, when based on such approximate tools for error indication, h-adaptivity helps to reduce dramatically the computational resources necessary to perform simulations. This becomes especially vital in the case of three dimensional simulations where time and memory requirements are usually important.

The present contribution presents some techniques and strategies for implementing h-adaptivity into a 3D finite element code for solving inviscid and viscous flow problems.

2. Finite element formulation for simulations of compressible flows

As a mathematical model for numerical approximations of compressible flows we use the regularized Navier-Stokes equations

$$\boldsymbol{U}(\boldsymbol{x},t)_{,t} + \boldsymbol{f}_i^E(\boldsymbol{U})_{,i} = \left[\left(\mathsf{K}_{ij}^{AV}(\boldsymbol{U},\nabla\boldsymbol{U}) + \mathsf{K}_{ij}^{\mu}(\boldsymbol{U})\right)\boldsymbol{U}_{,j}\right]_{,i} \tag{2.1}$$

where $\boldsymbol{U}$ is the vector of conservation variables, $\boldsymbol{f}_i^E$ are the Eulerian fluxes and K_{ij}^{AV}, K_{ij}^{μ} matrices corresponding to artificial viscosity and natural viscosity and heat conduction, respectively (for the complete listing of notation see Appendix A). Artificial viscosity is included into the model already at this early stage to emphasize the fact that our numerical approximation will consist of two parts: regularization of the problem by artificial viscosity which makes the problem solvable on practically used meshes, and then application of some numerical algorithm to the regularized problem. A particular form of artificial viscosity usually depends on the approximation method (e.g. for ensuring that with the increased accuracy of computations it convergences to zero). In our computations we have used the model originally developed for the Streamline Upwind Petrov-Galerkin method (cf Hansbo and Johnson, 1991; Shakib et al.,

1991) and adapted for the Taylor-Galerkin time approximation algorithm (cf Banaś and Demkowicz, 1996)

$$\mathsf{K}_{ij}^{AV} = Ch \frac{(f_{k,k}^{E})^{\top} \eta_{,UU} f_{k,k}^{E}}{U_{,l}^{\top} \eta_{,UU} U_{,l}} \delta_{ij} \mathsf{I}$$

where $-\eta_{,UU}$ is the symmetric and positive definite Hessian of the nondimensional entropy density η with respect to the conservation variables, h is a representative local size of a finite element mesh and C is a parameter. This artificial viscosity model, based on the residual of the steady-state Euler equations $f_{k,k}^{E}$, played an important role in the first proof of convergence of a finite element method for nonlinear hyperbolic conservation laws (cf Johnson et al., 1990). Apart from being theoretically justified it behaves well in practice.

To approximate the solutions to (2.1) we use a sequence of solutions $\boldsymbol{U}^n(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x}, t^n)$ of the one step finite element problem given below:

- Find $\boldsymbol{U}^{n+1}$ belonging to the suitable finite element approximation space, satisfying the Dirichlet boundary conditions and such that for every test function $\boldsymbol{W}$ the following holds

$$\int_{\Omega_C} \boldsymbol{W}^{\top} \boldsymbol{U}^{n+1} \, dV + \Delta t \int_{\Omega_C} \boldsymbol{W}_{,i}^{\top} (\mathsf{K}_{ij})^n \boldsymbol{U}_{,j}^{n+1} \, dV =$$

$$= \int_{\Omega_C} \boldsymbol{W}^{\top} \boldsymbol{U}^n \, dV + \Delta t \int_{\Omega_C} \boldsymbol{W}_{,i}^{\top} (\boldsymbol{f}_i^E)^n \, dV - \Delta t \int_{\partial \Omega_C} \boldsymbol{W}^{\top} [(\boldsymbol{f}_i^E)^n - (\boldsymbol{f}_i^{\mu})^n] n_i \, dS \tag{2.2}$$

where Ω_C is the computational domain with n_i the outward unit vector, normal to its boundary, $\Delta t = t^{n+1} - t^n$ is the time step length and $\boldsymbol{f}_i^{\mu} = \mathsf{K}_{ij}^{\mu} \boldsymbol{U}_{,j}$.

The algorithm (2.2) is a version of the Taylor-Galerkin method for solving flow problems (the Taylor-Galerkin method is the finite element counterpart of the Lax-Wendroff finite difference method). In this formulation the matrices K_{ij} are sums of three elements: $\mathsf{K}_{ij} = \mathsf{K}_{ij}^{TG} + \mathsf{K}_{ij}^{AV} + \mathsf{K}_{ij}^{\mu}$ with the Taylor-Galerkin method matrices $\mathsf{K}_{ij}^{TG} = \frac{\Delta t}{2} (\boldsymbol{f}_i)_{,U} (\boldsymbol{f}_j)_{,U}$ coming from the second order accurate approximation in time.

For steady state problems the algorithm (2.2) is additionally modified by the introduction into K_{ij}^{TG}, instead of the real time step Δt, of the so called critical time step $\Delta t_c = h/(c + |\boldsymbol{u}|)$, where c is the speed of sound and $|\boldsymbol{u}|$ is the modulus of gas velocity. This corresponds to the assumption of the uniform local Courant-Friedrichs-Levy (CFL) number approximately equal to one for all elements in the mesh. The reason for modification is on one hand to make the converged solution independent of the time step length (Δt included in $\mathbf{K}_{ij}^{TG}$ affects the solution while in other places it can be treated just as a formal parameter on the path to convergence) and on the other hand to provide the optimal amount of artificial diffusion to the scheme in order to compensate for the negative artificial dissipation inherent to the central differencing schemes (typical for the Galerkin finite element method) when it is applied to convection dominated problems (cf Brooks and Hughes, 1982).

A proper application of the boundary conditions require that the whole boundary of the computational domain Ω_C is *a priori* divided into parts of different kinds and that this partition does not change in time. This fact is especially important for the so called open boundary, i.e. the part of the boundary arbitrarily imposed across the flowfield, where the actual solution is not known in advance. For the purpose of, presented in this paper, supersonic calculations we identify the following types of boundary and corresponing boundary conditions:

- inflow $(u_i n_i < 0)$

$$\boldsymbol{U}(x,t) = \boldsymbol{U}_\infty(x)$$

U_∞ given data

- outflow $(u_i n_i > 0)$

$$\boldsymbol{U}(x,t)_{,i} n_i = 0$$

- solid wall with no heat exchange

$$\boldsymbol{u} = 0 \qquad \wedge \qquad q_i n_i = 0$$

- solid wall with a prescribed temperature

$$\boldsymbol{u} = 0 \qquad \wedge \qquad T = T_{wall}$$

T_{wall} specified

- symmetry hypersurface

$$u_i n_i = 0 \qquad \wedge \qquad q_i n_i = 0 \qquad \wedge \qquad u^{t_1}_{,i} n_i = u^{t_2}_{,i} n_i = 0$$

$u^{t_{1,2}}$ are the components of velocity tangent to the boundary $\partial\Omega_C$ and the condition means that normal derivatives of tangential components of velocity vanish (cf Demkowicz et. al., 1991)

When simulating inviscid flows the no-slip condition on walls is replaced with enforcement of zero normal velocity relative to the wall and all the terms corresponding to viscosity and heat flux are neglected.

The Dirichlet boundary conditions are applied using the penalty method. The enforcement of inflow values or no-slip condition on walls is realised in a standard way by scaling diagonal and corresponing right hand side values. Vanishing of the normal velocity is achieved by adding to the variational formulation (2.2) the term

$$\int_{\partial\Omega_{wall}} \frac{1}{\epsilon}\Big(W_2 n_1 + W_3 n_2 + W_4 n_3\Big)\Big(U_2^{n+1} n_1 + U_3^{n+1} n_2 + U_4^{n+1} n_3\Big)\, dS$$

where ϵ is a small penalty parameter. Temperature boundary condition is applied using the relation $T = \frac{e}{c_V} = \frac{1}{c_V}\cdot\frac{u_5}{u_1}$ valid on the wall on which $\boldsymbol{u} = \mathbf{0}$. Addition of the term

$$\int_{\partial\Omega_{wall}} \frac{1}{\epsilon} W_5 \Big(-c_V T_{wall} U_1^{n+1} + U_5^{n+1}\Big)\, dS$$

to Eq (2.2) enforces a desired constant value of temperature on the corresponing wall boundary (cf Tworzydlo et al., 1992).

Boundary conditions that involves normal derivatives of solution are directly incorporated into the finite element formulation by means of modifications to the boundary terms in Eq (2.2). For example, vanishing of heat flux is simply reflected by neglecting all terms containing $\boldsymbol{q}$, while the enforcement of zero tangential derivatives of velocity requires transformation of fluxes to the coordinate system with one axis normal to the boundary and modification of fluxes in this system.

3. Space discretization of the one step problem and h adaptivity

The computational domain is discretized using prismatic elements shown in Fig.1. In each element the 3D shape functions are constructed as tensor products of the standard 2D linear shape functions on triangles (bases) and the standard bilinear 2D shape functions on quadrilaterals (sides), cf Banaś and Demkowicz (1993).

In our time discretization algorithm each one step computations constitute a separate problem which can be solved on a new finite element mesh taking the interpolated values of the previous one step calculations as an initial condition. This allows us to introduce h adaptations of the mesh at chosen instants during simulations.

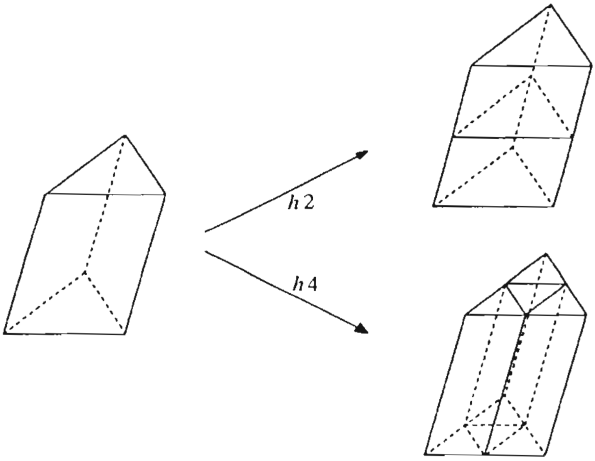

Fig. 1. Two kinds of the element division

Among many error indicators developed so far most are designed for linear elliptic second order problems. Some of them are based on *a priori* error estimation for the finite element method and estimation of interpolation error. Although straightforward they can give good results also for flow problems. We use the error indicator of this kind to adapt the mesh in boundary layers. It is the error indicator developed by Rachowicz (1997) and based on estimation of the errors in heat flux and tangential components of viscous stresses.

Out of the boundary layers we are mostly interested in the error in the solution itself not in the gradients of solution, and we use the residual error indicator (cf Banaś and Demkowicz, 1994) based on the error estimates developed by Johnson et al. for linearized convection-diffusion equations (cf Eriksson and Johnson, 1993). In each element as an error indicator we compute the approximate integral of the residual of the steady state Euler equations

$$I_{el} = h^2 \boldsymbol{f}^i_{,i} \eta_{,UU} \boldsymbol{f}^j_{,j}$$

Although developed for linear problems the error indicator behaves well also in nonlinear flow simulations.

3.1. Techniques of element division

We have used in our flow simulations two kinds of element division, see Fig.1. One, called the vertical or $h2$ division, consist in breaking an element into two "sons" sharing a common base while the other, horizontal or $h4$, consist in breaking an element into four sons sharing sides. Both divisions introduce anizotropy into a mesh by changing the element aspect ratio, which in our 3D mesh of prizmatic elements is measured as the ratio of the height of an element to some characteristic linear parameter corresponding to the size of bases.

The straightforward application of the adaptivity concept would consist of computing error indicators (of both kinds) and then dividing elements with biggest errors to improve the quality of approximation. However due to requirements imposed on meshes and approximate solutions one cannot divide an element without considering all its neighbours. At this stage a certain strategy of dividing elements in the neighbourhood of an indicated element has to be applied. The aim of introducing particular, sometimes complicated, strategies is, on the one hand, to lead to the most efficient reduction of error and, on the other, to maintain the consistency of the data structure and to allow for further element divisions. In order to preserve the continuity of finite element approximation after the insertion of new nodes the technique of constrained approximation is introduced. This means that at the nodes which lie in the middle of any edge of bigger, neghbouring elements solution values are simply interpolated. The constrained approximation allows two elements of different sizes to lie adjacent to each other and thus makes possible to confine mesh modificiations only to the regions where they are necessary without introducing new kinds of elements or temporary data structure constructs. However, it further complicates the data structure and introduces new requirements imposed on the strategy of adaptation.

The strategy of adaptation implemented in our code is based on several assumptions. First, we require that anisotropic divisions are limited only to the boundary layers where the solution changes substantially only in the direction normal to the boundary, so only in this direction the refinements are necessary. Since the solid wall boundaries in the mesh generation algorithm which we used are covered with the triangular mesh this implies that in boundary layers we use exclusively $h2$ divisions. On the other hand, within shocks we always try to maintain isotropy of the mesh which allows the mesh to adapt more easily to moving shocks. Hence, in the places indicated by the residual error indicator we always apply to each element a sequence of two divisions: $h2$

followed by $h4$.

Several further requirements are of more technical nature and are introduced to make changes in elements sizes smooth and to maintain compatibility between different generations of elements so as to further divisions are possible. These requirements include:

- Each node of a divided element must be a real, not constrained, node – if it is not the case, the neighbours of the element are divided as long as this condition is satisfied
- The number of $h2$ refinements must be greater than the number of $h4$ refinements for each element of the initial mesh
- After $h2$ refinements all nodes of two new elements must be real (not constrained)
- Two elements sharing a common base should not differ by more than one $h2$ generation level
- Some further conditions for avoiding inconsistencies of the data structure when two different refinement zones meet at some element.

4. Numerical examples

We show two numerical examples of steady state problems, one inviscid and one viscous Both present basic capabilities of the h adaptive code and the strategy implemented, although the refinements are of limited depth and spread, due to limited computer time and memory resources.

4.1. Flow over a blunt body

The first example is the inviscid flow over the half-sphere, which, due to symmetry of the problem, is computed as the flow over one eights of a sphere. In Fig.2 the geometry of the problem is presented together with the solution on the final adapted mesh. The inner sphere is a solid wall boundary, the outer sphere is an inflow boundary, with the inflow Mach number of the flow equal to 3. The boundary conditions on two planes (z = const and y = const) enforce the symmetry of the problem (the condition for inviscid flows is the same as on solid wall boundaries) and the x = const plane is an

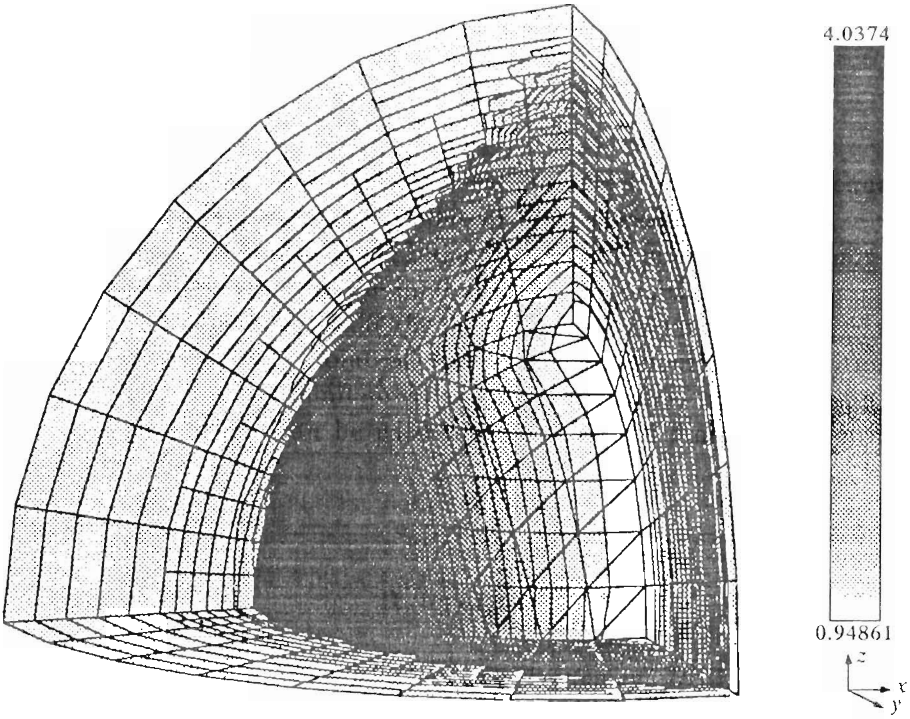

Fig. 2 The mesh and density contours for the flow over a sphere problem

outflow boundary. The solution of the problem is represented by 20 density contours ranging from 0.9486 to 4.037. The bow shock which forms in front of the sphere is clearly visible as well as refinement zone which improves shock resolution.

4.2. Flat plate problem

The second example is the viscous flow over a flat plate with the Reynolds number equal to 10 000 and the Mach number equal to 3. In Fig.3 the bottom boundary consist of the left-hand part with symmetry condition and the rest, starting at $x = 0.1$ being a solid wall with specified temperature. The left-hand and upper sides of the computational domain are inflow boundaries and the right-hand side, $(x = \text{const})$ boundary is an outflow. The rest of the boundary $(z = \text{const})$ enforces the symmetry of the problem, which essentially is two dimensional. The solution, once again illustrated by the

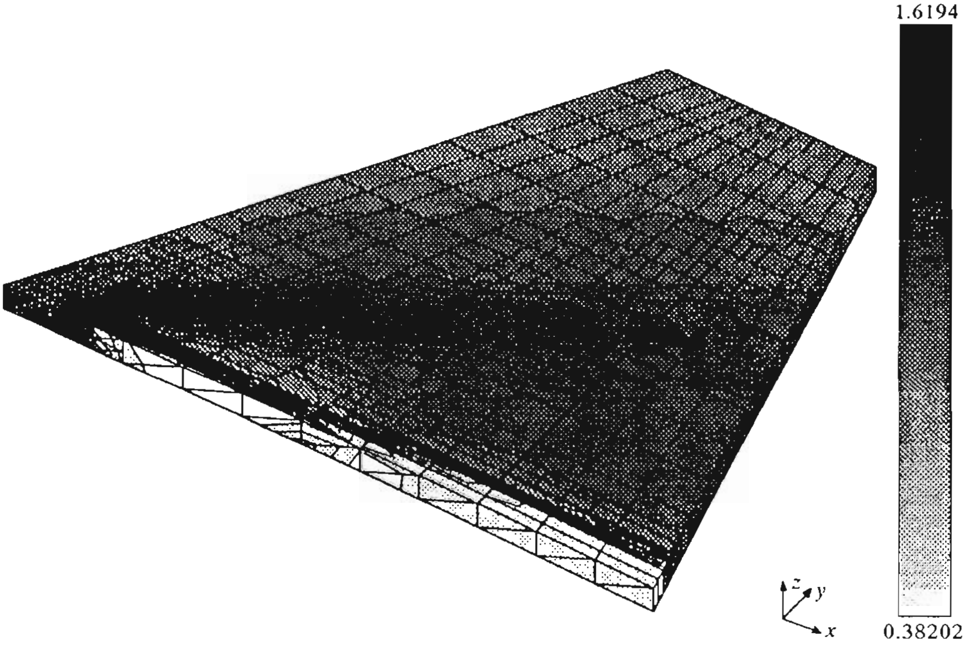

Fig. 3. The mesh and density contours for the flow over a flat plate problem

density contours, depicts creation of a boundary layer, and a shock which results from separation of the boundary layer at the tip of plate. Refinements of both kinds, isotropic in shock regions and anisotropic in the boundary layer, are visible.

5. Conclusions

Combining isotropic and anisotropic h refinements in one finite element code creates a challenging and technically difficult problem. To help solving this problem not only sophisticated strategies, as the one presented in the article, could be used but also hybrid data structures with different kinds of elements and robust mesh generation procedures compatible with particular data structures. This constitutes the next steps in the development of the 3D code for flow simulations presented in this article.

References

1. Banaś K., Demkowicz L., 1993, 3D $h-p$ Adaptive Package, Report 4/1993, Section of Applied Mathematics UCK, Cracow University of Technology

2. Banaś K., Demkowicz L., 1994, Entropy Stable Gas Dynamics Simulations by Adaptive Finite Elements, *Proceedings of the Second European Fluid Dynamics Conference ECCOMAS 94*, Stuttgart, edit. S.Wagner, E.H.Hirchel, J.Periaux, R.Piva, vol. II, 97-104, Chichester, Wiley

3. Banaś K., Demkowicz L., 1996, Entropy Controlled Adaptive Finite Element Simulations for Compressible Gas Flow, *Journal of Computational Physics,* **126**, 181-201

4. Brooks A.N., Hughes T.J.R., 1982, Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with the Particular Emphasis on the Incompressible Navier-Stokes Equations, *Computer Methods in Applied Mechanics and Engineering,* **32**, 199-259

5. Demkowicz L., Oden J.T., Rachowicz W., Hardy O., 1991, An h-p Taylor-Galerkin Finite Element Method for Compressible Euler Equations, *Computer Methods in Applied Mechanics and Engineering,* **88**, 363-396

6. Eriksson K., Johnson C., 1993, Adaptive Streamline Diffusion Finite Element Methods for Stationary Convection Diffusion Problems, *Mathematics of Computation,* **60**, 167-188

7. Hansbo P., Johnson C., 1991, Adaptive Streamline Diffusion Method for Compressible Flow Using Conservation Variables, *Computer Methods in Applied Mechanics and Engineering,* **87**, 267-280

8. Johnson C., Szepessy A., Hansbo P., 1996, On the Convergence of Shock-Capturing Streamline Diffusion Finite Element Methods for Hyperbolic Conservation Laws, *Mathematics of Computation,* **54**, 107-129

9. Rachowicz W., An Anisotropic h-Adaptive Finite Element Method for Compressible Navier-Stokes Equations, *Computer Methods in Applied Mechanics and Engineering,* to appear in 1997

10. Shakib F., Hughes T.J.R., Johan Z., 1991, A New Finite Element Formulation for Computational Fluid Dynamics: X. The Compressible Euler and Navier-Stokes Equations, *Computer Methods in Applied Mechanics and Engineering,* **89**, 141-219

11. Tworzydło W.W., Oden J.T., Thornton E.A., 1992, Adaptive Implicit/Explicit Finite Element Method for Compressible Viscous Flow, *Computer Methods in Applied Mechanics and Engineering,* **95**, 397-440

Appendix A. Notation

$\boldsymbol{x}$	–	point inside Ω_C
t	–	moment of time
Ω_C	–	computational domain, $\Omega_C \subset \mathbb{R}^n$, $n = 1,2,3$
n_i	–	outward unit vector, normal to the boundary $\partial\Omega_C$
ρ	–	density
u_i	–	ith component of velocity
e	–	total specific energy, $e = e_K + e_I$
e_K	–	specific kinetic energy, $e_K = \frac{1}{2}u_l u_l$
e_I	–	specific internal energy, $e_I = \frac{p}{(\gamma-1)\rho}$
p	–	pressure, $p = (\gamma-1)(\rho e - \frac{1}{2}\rho u_i u_i)$
γ	–	ratio of specific heats, $\gamma = c_V/c_p$, c_V at a constant volume, c_p at a constant pressure
c	–	speed of sound, $c = \sqrt{\gamma p/\rho}$
U	–	vector of conservation variables, $\boldsymbol{U} = [\rho, \rho u_k, \rho e]^{\mathsf{T}}$
μ, λ	–	coefficients of viscosity, ($\lambda = -\frac{2}{3}\mu$ by the Stokes hypothesis)
q_i	–	components of the heat flux vector q, $q_i = -\kappa T_{,i}$, and κ is the coefficient of thermal conductivity
T	–	temperature, $T = \frac{1}{c_V}e_I$
$\eta(U)$	–	nondimensional entropy density, $\eta(U) = \rho\ln(p\rho^{-\gamma})$
$\eta_{,UU}$	–	hessian of nondimensional entropy with respect to conservation variables ($-\eta_{,UU}$ is symmetric and positive definite)

$$\eta_{,UU} = -\frac{1}{\rho e_I^2}\begin{bmatrix} \gamma e_I^2 + e_K^2 & -u_j e_K & e_K - e_I \\ -u_i e_K & e_I\delta_{ij} + u_i u_j & -u_i \\ e_K - e_I & -u_j & 1 \end{bmatrix}$$

$\boldsymbol{f}_i^E$	–	Eulerian fluxes, $\boldsymbol{f}_i^E = [\rho u_i, \rho u_i u_j + p\delta_{ij}, (\rho e + p)u_i]^{\mathsf{T}}$
$\mathsf{K}_{ij}^{\mu}(U)$	–	natural viscosity and heat flux matrices

$$\mathbf{K}^{\mu}_{ij} = \frac{1}{\rho}\begin{bmatrix} 0 & 0 & 0 \\ -\mu u_l\delta_{ij} - \mu u_i\delta_{jl} - \lambda u_j\delta_{il} & \mu\delta_{ij}\delta_{kl} + \mu\delta_{ik}\delta_{jl} + \lambda\delta_{jk}\delta_{il} & 0 \\ -(\lambda+\mu)u_iu_j - 2\mu e_K\delta_{ij} & \mu u_k\delta_{ij} + \mu u_j\delta_{ik} + \lambda u_i\delta_{jk} & 0 \end{bmatrix} +$$

$$+ \quad \delta_{ij}\frac{\kappa}{c_V\rho}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ e_K - e_I & u_k & 1 \end{bmatrix}$$

$\mathbf{K}^{AV}_{ij}(U, \nabla U)$ – matrix functions corresponding to a particular model of artificial viscosity

$\mathbf{K}^{TG}_{ij}$ – matrix functions of the second order Taylor-Galerkin (Lax-Wendroff) approximation method, $\mathbf{K}^{TG}_{ij} = \frac{\Delta t}{2}(f_i)_{,U}(f_j)_{,U}$

Δt – time step length, $\Delta t = t^{n+1} - t^n$

$\boldsymbol{W}$ – test function

h – characteristic linear local mesh (element) size

C – constant

$\mathbf{I}$ – identity matrix

δ_{ij} – Kronecker delta

$(\cdot)_{i,j,k,l}$ – subscripts corresponding to space dimensions, $i, j, k = 1, ..., n$

$(\cdot)_\infty$ – subscript for free stream values

$(\cdot)_n$ – subscript denoting the normal component of a vector

$(\cdot)^n$ – superscript for values at instant t^n

$(\cdot)_{,t}, (\cdot)_{,i}$ – partial derivatives, $\partial/\partial t$, $\partial/\partial x_i$

in all vectors and matrices the middle rows and columns should be expanded according to the number of space dimensions n

h-adaptacyjne symulacje trójwymiarowych lepkich i nielepkich przepływów gazu metodą elementów skończonych

Streszczenie

Przedstawiono sformułowanie algorytmu metody elementów skończonych do aproksymacji ściśliwych równań Navier-Stokesa. Uwzględniono zagadnienia postawienia warunków brzegowych i oszacowania błędu aproksymacji. Opisano implementację algorytmu w programie h-adaptacyjnej metody elementów skończonych do symulacji zadań trójwymiarowych. Nacisk położono na specjalnie dostosowane do symulacji przepływów, zwłaszcza lepkich, techniki i strategie adaptacji siatki elementów skończonych. Zaprezentowano przykłady obliczeń dla zagadnień trójwymiarowych naddźwiękowych przepływów lepkich i nielepkich.

Manuscript received October 4, 1996; accepted for print January 8, 1997