Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR TP 96431

An explicit multi time stepping algorithm for aerodynamic flows

H. van der Ven, B.E. Niemann-Tuitman, A.E.P. Veldman

DOCUMENT CONTROL SHEET

	ORIGINATOR'S REF. NLR TP 96431 U		**SECURITY CLASS.** Unclassified

ORIGINATOR
National Aerospace Laboratory NLR, Amsterdam, The Netherlands

TITLE
An explicit multi time stepping algorithm for aerodynamic flows

PRESENTED AT
The International Congress on Computational and Applied Mathematics, July 21-26, 1996, Leuven, Belgium

AUTHORS	**DATE**	**pp**	**ref**
H. van der Ven, B.E. Niemann-Tuitman, A.E.P. Veldman	960708	15	5

DESCRIPTORS

Aerodynamics coefficients	Finite volume method	Step functions
Airfoils	Multiblock grids	Time dependence
Algorithms	Numerical stability	Turbulent flow
Boundary conditions	Run time (computers)	
Computational grids	Runge-Kutta method	

ABSTRACT
An explicit multi time stepping algorithm with applications to aerodynamic flows is presented. In the algorithm in different parts of the computational domain different time steps are taken, and the flow is synchronized at so-called synchronization levels. The algorithm is validated for aerodynamic turbulent flows. For two dimensional flows speedups in the order of five with respect to single time stepping are obtained.

Summary

An explicit multi time stepping algorithm with applications to aerodynamic flows is presented. In the algorithm in different parts of the computational domain different time steps are taken, and the flow is synchronized at so-called synchronization levels. The algorithm is validated for aerodynamic turbulent flows. For two dimensional flows speedups in the order of five with respect to single time stepping are obtained.

Contents

1 Table
5 Figures

(15 pages in total)

1 Introduction

For time accurate simulations stability restrictions on the time step in finely resolved parts of the computational domain decrease the performance of explicit methods applying one uniform time step in the entire domain. Multi time stepping methods, where in different parts of the domain different time steps are taken, are more efficient.

Crucial to multi time stepping methods is the information exchange between the different parts of the domain. For an overview of previous work in the field of linear structural dynamics we refer to Belytschko and Lu, Ref. 1. For convection dominated flow problems Maurits et al. Ref. 4 considered the one-dimensional convection-diffusion equation as a model problem. Maurits et al. exchanged information only at synchronization levels, that is, at the largest time step. Time accurate simulations were performed with a ratio of 400 between the different time steps. Kleb et al. Ref. 3 considered point-wise multi time stepping: each grid point is advanced using its own time step which fits a power of two times in the largest time step. The larger time steps are advanced the earlier in order to generate necessary results at intermediate levels. The results at intermediate levels are obtained using linear interpolation. Kleb et al. report speedups between 4 and 10 for oscillating airfoils.

The present investigation extends the multi time stepping method of Maurits et al. to three dimensional aerodynamic simulations.

Apart from their efficiency, explicit multi time stepping algorithms are easy to implement in existing explicit steady state solvers. Moreover, they have an inherent parallelism and their simplicity leads one to expect excellent parallel efficiency.

The contents of the report is as follows. In Chapter 2 the multi time stepping algorithm used in this paper is described. Two algorithms are described, the basic algorithm and a simplified algorithm with less information exchange. In Chapter 3 the algorithms are validated. In the final chapter conclusions are drawn.

2 The multi time stepping algorithm

Basically, any algorithm in which different time steps are taken in different parts of the grid will be called a multi time stepping algorithm. Crucial are time levels at which (part of) the flow is synchronized to obtain time-accurate behaviour. Only at these synchronization levels information between the different parts is exchanged.

Two algorithms are tested in the present report. In the first algorithm, the basic algorithm, the information exchange is such that at all block boundaries the local stability conditions are satisfied. In the second algorithm, which is a simplification of the first, information exchange only takes place after the largest time step has been performed. The motivation to also consider the second algorithm is its simplicity: it requires less organization and is easier to implement.

The spatial discretization of the solver to which multi time stepping is added, is a cell-vertex Jameson scheme and block boundaries are part of both blocks they bound. Around each block two layers of dummy cells are added which contain the flow status in the bounding blocks (if any). Since the block boundaries belong to both blocks they bound, the block boundaries are multi-valued after an integration pass. After the integration pass the multi-valuedness is removed by averaging the different values. Moreover, the dummy cells at the internal block boundaries are refreshed.

The time integration scheme is the standard Runge-Kutta 4 algorithm. For linear problems this scheme is third order accurate over a fixed time interval. The accuracy of the multi-block implementation of this scheme has been measured for turbulent flows to be between second and third order.

2.1 The basic algorithm

In the description of the algorithms it is assumed that the computational domain is subdivided into blocks. The time step that is taken in one block will be called a block time step. The block time steps Δt_b are calculated in the following way. First the locally stable time steps are determined. Let Δt_c^b be the locally stable time step in cell c of block b. Let $\Delta t_{\min}^b$ be the minimum of Δt_c^b over the cells in block b. The synchronization time step Δt_{syn} is defined as the largest of these time steps. Finally, the time steps $\Delta t_{\min}^b$ for all blocks are decreased as to fit in the synchronization time step an integer number of times. The decreased time steps are the block time steps.

In each block the flow is advanced over as many block time steps that fit in the synchronization time step. At each synchronization level the block time steps are determined again.

In order to satisfy the local stability conditions at the block boundaries the block time steps are performed in a certain order. The dummy variables are refreshed after each block time step. The order of the time step computations is roughly determined by the elapsed physical time, which is measured by the so-called master clock. The master clock is advanced in time with steps equal to the smallest block time step $\Delta t_{\min}$, that is, $\Delta t_{\min} = \min_b \Delta t_b$. Below is a description of the part of the algorithm that advances the flow one synchronization time step.

repeat
  do for all blocks
    if time step should be set
      integrate flow in block one block time step
    endif
  enddo
  refresh dummies for all blocks
  apply boundary conditions
  advance time of master clock with $\Delta t_{\min}$
until one synchronization time step is elapsed at the master clock

The condition 'time step should be set' is true if the elapsed time T_b in block b lags too far behind with respect to the elapsed time on the master clock. Here we define T_b as the time on the master clock at which the latest time step in block b has been set. This implies that the next time step in this block should be set no later than $T_b + \Delta t_b$. Let T denote the elapsed physical time on the master clock, then we update block b at the time T for which

$$T < T_b + \Delta t_b \leq T + \Delta t_{\min},$$

that is, the block update is made at the latest possible moment. Using this strategy it follows that for any two blocks b and b'

$$|T_b - T_{b'}| \leq \max(\Delta t_b, \Delta t_{b'}).$$

This means that at each loop index the elapsed time difference between two blocks is less than the maximum of the block time steps in the two blocks. This implies that at the block boundaries the local stability conditions are always satisfied.

Notice that in between synchronization levels the flow in the different blocks is not synchronized as it is at the synchronization levels: the block time steps do not necessarily fit an integer time in each other. Hence, the exchanged info in between synchronization levels is not necessarily at

the same time level. There is, however, information exchange in between synchronization levels: the ‘refresh dummies’ command exchanges information between bounding blocks. An example of the order of the time steps is given in Figure 1.

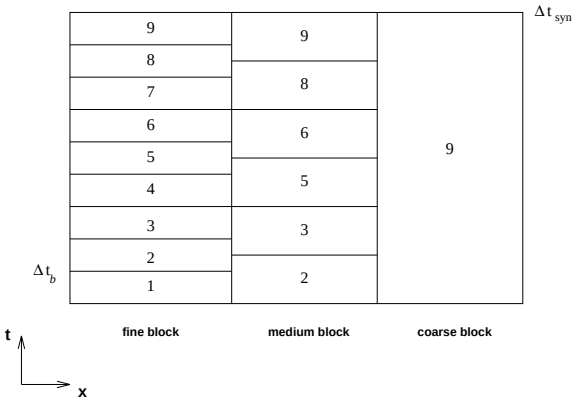

Fig. 1 Order of the block time step integration. The numbers refer to the order of the block time integration.

Finally note that in the above algorithm no averaging takes place at the multi-valued block boundaries. Since the boundaries should be accurately integrated in time, averaging is felt to be unnecessary. Moreover, averaging is an accelaration technique for steady flow computations, and as such obsolete for unsteady computations.

2.2 The simplified algorithm

In the simplified algorithm the information exchange in the loop over the block time steps is moved outside the repeat-until loop. Hence, in between synchronzation time steps the information at the block boundaries is frozen.

The above algorithm applied to arbitrary block decompositions may cause violation of local stability conditions at block boundaries bounding blocks with block time steps strictly smaller than the synchronization time step.

3 Experiments and validation

The algorithm described in Chapter 2 is implemented in the structured, multi-block solver EDDS, developed at NLR Ref. 5.

3.1 Stability

3.1.1 The basic algorithm

Turbulent flow is simulated around the RAE2822 airfoil under the flow conditions of Case 10 of Ref. 2. The nonuniform grid that is used is divided in 18 blocks, in three layers normal to the solid, each layer consisting of six blocks. The flow conditions are Re $= 6.2 \cdot 10^6$, $\alpha = 2.8°$, and the Mach number M $= 0.75$. The transition points are located at 3% chord. An initial (non converged) solution was made with local time stepping. Then ten multi-time steps using the simplified algorithm were taken.

The ratio r between the synchronization time step and the smallest block time step is 915. The flow remains stable for ten multi time steps (elapsed non-dimensional time: 0.11); flow results are displayed in Figure 2.

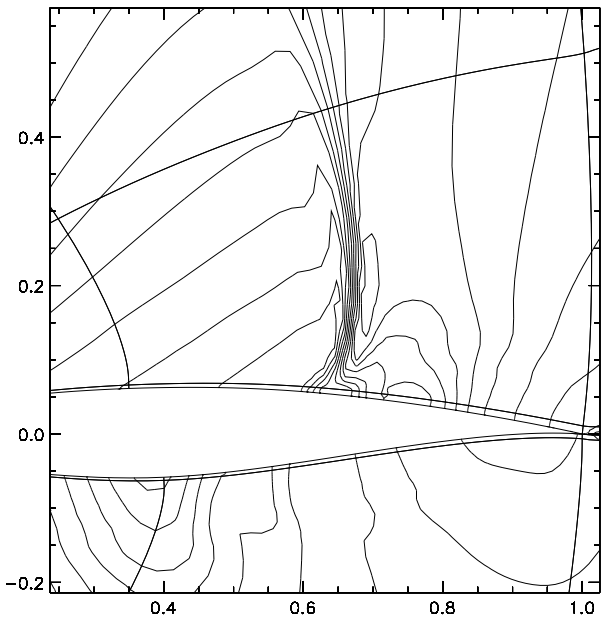

Fig. 2 Pressure distribution after 10 multi time steps with elapsed physical time 0.11.

3.1.2 The simplified algorithm

The same case as in the previous section is simulated using the simplified algorithm. Purpose of this simulation was to examine to what extent information exchange can be postponed. Unfortunately, judging from the wiggles in the shock region, the flow becomes instable.

To investigate the instability the synchronization level is decreased without affecting the smallest block time steps. Consequently, the ratio r between the synchronization time step and the smallest block time step decreases. Tested cases are shown in Table 1. All computations with smaller synchronization time steps are stable.

Table 1 Synchronization time steps and ratio's for RAE2822

Δt_{syn}	ratio r	stable
$112.6 \cdot 10^{-4}$	915	no
$40.8 \cdot 10^{-4}$	332	maybe
$20.4 \cdot 10^{-4}$	166	yes
$2.1 \cdot 10^{-4}$	18	yes

In the case of $\Delta t_{\text{syn}} = 2.1 \cdot 10^{-4}$ the block time steps in the blocks in the two outer layers are all equal to the synchronization time step. Hence in this case, the local stability conditions at the block boundaries in normal direction are satisfied. In all other cases the synchronization level is larger than the locally stable time steps at one or more block boundaries. Apparently, the flow in the block can damp instabilities at block boundaries to a certain degree for a certain time. If the information exchange is postponed too long, the computation becomes instable.

3.2 Accuracy

Because of the results of the stability tests, the accuracy is tested using the basic algorithm.

The flow obtained in Section 3.1.1 with multi time stepping is used for restarts using multi time stepping with different time steps. Two experiments are performed.

In the first experiment the block time steps in the outer blocks are decreased in such a way that the synchronization time step is halved, and successively quartered. The block time steps in the inner blocks near the solid are unaltered. Hence this experiment tests the dependence of the accuracy on the synchronization level.

In the second experiment the block time steps in *all* blocks are halved and successively quartered. Hence this experiment tests the dependence of the accuracy on the block time steps.

In all experiments the flow is integrated from the non dimensional time 0.113 to 0.117, which corresponds with one multi time step with unaltered block time step, or with 1000 single time steps. The flow results of the experiments are compared to the flow results using single time stepping.

In Figures 3 and 4 the time evolution of the aerodynamic coefficients for the different experiments is monitored. The agreement for both coefficients is excellent.

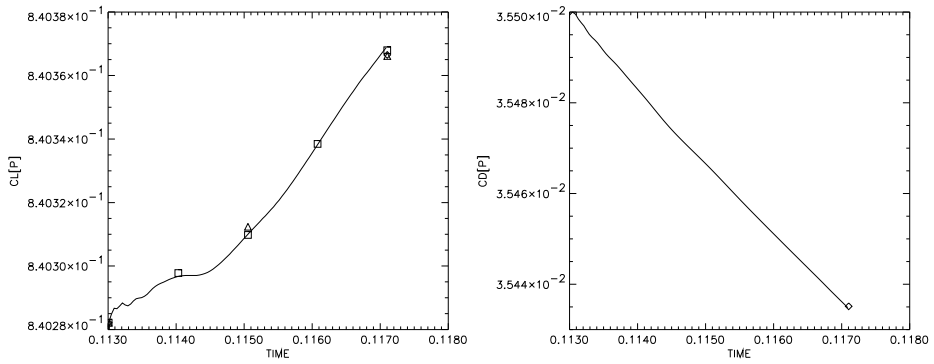

Fig. 3 Time evolution of the aerodynamic coefficients for the experiment in which the synchronization time step has been halved successively. Solid line: single time stepping; ◇: Δt_{syn}; △: $\frac{1}{2}\Delta t_{\mathrm{syn}}$; □: $\frac{1}{4}\Delta t_{\mathrm{syn}}$.

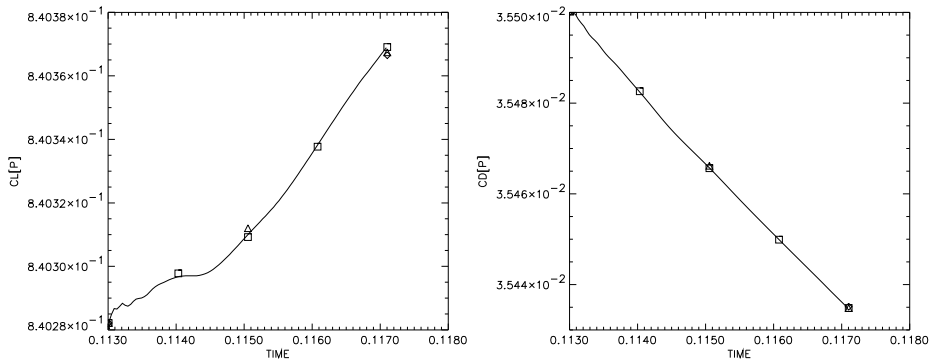

Fig. 4 Time evolution of the aerodynamic coefficients for the experiment in which all block time steps have been halved successively. Solid line: single time stepping; ◇: Δt_{syn}; △: halved block time steps; □: quartered block time steps.

In order to determine the formal accuracy of the multi time stepping method the following quotient was computed: $(u_1 - u_{\frac{1}{2}})/(u_{\frac{1}{2}} - u_{\frac{1}{4}})$. Here, u_1 refers to the solution with the default synchronization time step, $u_{\frac{1}{2}}$ with half the default synchronization time step, resp. half the block time step, and $u_{\frac{1}{4}}$ with quart default synchronization time step, resp. quart block time step. The ^{2} log of this quotient is the formal order of the accuracy of the multi time stepping method.

For the successive halvenings of the synchronization time step the quotient varied between 6.4 for the density and 15 for the energy. For the successive halvenings of the block time steps the quotient varied between 1.7 for the energy and 2.1 for the x-velocity.

Hence, the present multi time stepping method is first order accurate in the block time steps, and effect of the synchronization time step is negligible. In comparison with the single time stepping integration scheme the multi time stepping method loses one order of accuracy.

In principle, the multi time stepping method allows for accurate simulation of physical quantities within each block with an accuracy of the block time step. In order to assess this accuracy the time evolution during a synchronization time step of the normal velocity was monitored at a point below the airfoil and near the trailing edge. In Figure 5 the time evolution of this quantity is displayed together with the time evolution as given by single time stepping. For multi time stepping the velocity is printed every block time step, which is roughly a thousandth of the synchronization time step. The agreement between single and multi time stepping is excellent.

3.3 Efficiency

The single time stepping run of the previous section takes 3276 seconds, while the multi time stepping run with the default synchronization time step takes 731.2 seconds. Hence, a speedup of 4.5 is obtained.

When the block layer around the airfoil is halved in the normal direction to create a 24 blocks topology, the speedup with respect to single time stepping is increased to 5.5.

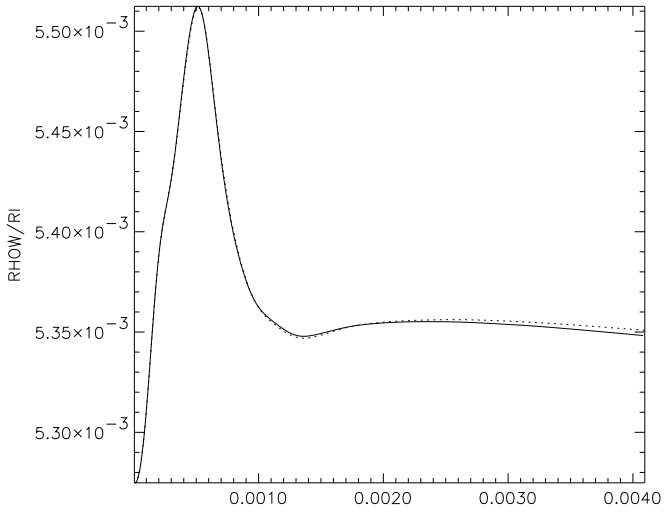

Fig. 5 Time evolution of the normal velocity at a point below the airfoil near the trailing edge. — single time stepping, - - - multi time stepping. The size of the block time step of the block in which the point lies is roughly equal to the single time step. The extent of the x-axis is equal to one synchronization time step.

4 Conclusions

The multi time stepping algorithm presented in this paper has been proven to be an accurate, efficient and easy-to-implement algorithm.

The algorithm is stable whenever the block time steps are determined by the local stability conditions of the time integration scheme in the block. A simplification of the algorithm where the information exchange is postponed to the synchronization levels proved to be unstable.

The accuracy is determined by the block time steps and not by the synchronization time step. This allows for the accurate simulation of phenomena with a smaller time scale than the synchronziation time step. The accuracy of the multi time stepping method over a fixed time interval is first order in the block time steps.

The efficiency is expressed in a speedup of five with respect to single time stepping.

The algorithm can be simply implemented in any block-structured flow solver.

References

1. T. Belytschko and Y.Y. Lu. Explicit multi-time step integration for first and second order finite element semidiscretizations. *Comp. Methods Appl. Mech. Eng.*, 108:353–383, 1993.
2. P.H. Cook, M.A. McDonald, and M.C.P. Firmin. Aerofoil RAE2822. Pressure distributions, and boundary layer and wake measurements. Technical Report AR-138, AGARD, 1979.
3. W. L. Kleb, J.T. Batina, and M.H.Williams. Temporal adaptive Euler/Navier-Stokes algorithm involving unstructured dynamic meshes. *AIAA Journal*, 30(8):1980–5, 1992.
4. N.M. Maurits, H. van der Ven, and A.E.P. Veldman. Explicit multi time stepping methods for convection-dominated flow problems. *submitted to Comp. Methods in Appl. Mech. and Eng.*, 1995.
5. M.E.S. Vogels. Technical design of edds, a system for the solution of the compressible flow equations on a three-dimensional block-structured grid. Technical Report TR 93235 L, Netherlands National Aerospace Laboratory NLR, 1993.