A Locally Implicit Method for Fluid Flow Problems

K.C.Reddy
Reddy and Associates
Rt.2, Box 253
Tullahoma, TN 37388

Final Report of Contract NO. NAS8-36648
for Period: July 1985-July 1986

Submitted to
George C Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

By
Reddy and Associates
Rt.2, Box 253
Tullahoma, TN 37388

November 1986

(NASA-CR-178989) A LOCALLY IMPLICIT METHOD FOR FLUID FLOW PROBLEMS Final Report, Jul. 1985 - Jul. 1986 (Reddy and Associates) 16 p CSCL 20D

N87-15443

Unclas
G3/34 40295

A Locally Implicit Method For Fluid Flow Problems

K. C. Reddy

November 1986

Table of Contents

A Locally Implicit Method For Fluid Flow Problems

1. Introduction

The fluid flow inside the space shuttle main engine (SSME) traverses through a complex geometrical configuration. The flow is compressible, viscous and turbulent with pockets of separated regions. Computation of such a flow field requires the application of the state-of-the-art CFD technology. Several computer codes$^{(1-4)}$ are being developed to solve three dimensional Navier–Stokes equations with different turbulence models for analyzing the SSME internal flow. Most of these computer codes require extensive computational effort and are rather slow to converge if accurate solutions are to be obtained in three dimensional domains. This report outlines an algorithm which has the potential of solving fluid flow problems with rapid convergence to steady-state solutions.

The computational methods used in the Navier–Stokes codes fall into two major categories: finite difference and finite element methods. Each have their own strengths and weaknesses. Some of the algorithms are designed to solve the unsteady Compressible Navier–Stokes equations, either by explicit or by implicit factorization methods, for several hundred or thousands of time steps to reach a steady-state solution asymptotically. Other algorithms attempt to reach the steady-state solution by relaxation methods. All of them require body-fitting curvilinear grids with sufficient resolution. Grid requirements, however, differ greatly with the algorithm used. Implicit factorization methods, typically use numerical transformations whereby the transformed grid in the computational space is uniform and rectilinear. This requires the grid to have indices which are separable in three directions for three dimensional problems, and also be reasonably smooth. However, such requirements may introduce grid singularities when complicated domains are discretized. Flow solver algorithm will have to deal with such grid singularities. Explicit schemes and finite element algorithms have less stringent requirements on the grid structure. However, their efficiency and accuracy also depend heavily on the grid structure.

Explicit algorithms for solving time-dependent Navier–Stokes equations are usually bound by stability restrictions and take thousands of time steps to converge to a steady-state for three-dimensional flow problems. Most of the current implicit algorithms use approximate factorization techniques and require index separable grid structure. The approximate factorization techniques are reasonably efficient for two-dimensional problems, but can be quite slow in convergence for three dimensional problems. We propose a locally implicit algorithm, which is based on a relaxation procedure, for solving time dependent partial differential equations to obtain steady-state solutions. The basic algorithm which is locally implicit but globally explicit is computationally efficient and is outlined in the next section. It is suitable for the application of convergence acceleration procedures such as the multi-grid method which is also outlined.

Test calculations with the current method for solving an elliptic partial differential equation, the diffusion equation, which simulates many features of the subsonic flow problems are presented in the third section. Both single grid and multi-grid results are presented. Solutions for inviscid and viscous flow problems from subsonic to supersonic flow problems are being computed successfully by this method and those results will be published in due course.

2. Locally Implicit Scheme

A relaxation scheme known as the locally implicit scheme is described below. The compressible Navier–Stokes equations can be written in the conservation form.

$$\frac{\partial G}{\partial t} + \nabla \cdot \vec{f} = 0 \tag{2.1}$$

where G is the conservation variable vector and $\vec{F} = (F_x, F_y, F_z)$ represents the flux vectors, including inviscid and viscous terms. Addition of the equation of state makes this system complete. Integrating over space and using the Gauss Theorem,

$$\frac{\partial}{\partial t}\int_v G\ dv + \int_s \vec{F} \cdot d\vec{s} = 0 \tag{2.2}$$

where v is any closed volume element and s is its surface. In the finite volume discretization, the computational domain of the problem is covered by a body conforming grid, typically of quadrilateral bricks. For the current scheme the grid need not be index separable. We only need a grid which provides reasonable resolution of the flow field. Finite element type of grids are acceptable as are also the well structured curvilinear grids used by many of the finite difference codes. Euler implicit discretization of the equation (2.2) is written as

$$v_{j,k,\ell}\left(\frac{G^{n+1} - G^n}{\Delta t}\right)_{j,k,\ell} + \mathcal{F}_{j,k,\ell}\left(G^{n+1}\right) - \mathcal{D}_{j,k,\ell}\left(G^{n+1}\right) = 0 \tag{2.3}$$

where

$$\mathcal{F}_{j,k,\ell}(G) = (\vec{F}.\vec{s}\,)_{j+1/2,k,\ell} - (\vec{F}.\vec{s}\,)_{j-1/2,k,\ell} + (\vec{F}.\vec{s}\,)_{j,k+1/2,\ell}$$
$$-(\vec{F}.\vec{s}\,)_{j,k-1/2,\ell} + (\vec{F}.\vec{s}\,)_{j,k,\ell+1/2} - (\vec{F}.\vec{s}\,)_{j,k,\ell-1/2}\ ,$$
$$\vec{s}_{j+1/2,k,\ell} = (\vec{n}ds)_{j+1/2,k,\ell}\ ,\ \text{etc.}$$

and $\vec{n}$ is the unit normal vector pointing in the increasing j direction. In the above, $\mathcal{D}_{j,k,\ell}(G)$ are the artificial dissipation terms similar to those used by Jameson, et al[5]. They are smaller than the trucation error terms of the scheme and are designed to suppress the nonlinear instabilities which arise due to the central difference approximation of the convective flux terms, without affecting the accuracy of the solution. The implicit difference equations (2.3) for all the cells of the domain represent a set of coupled non-linear algebraic equations. The coupling of the equations for each cell is primarily with the equations of the neighboring nodes. Thus one can devise Gauss–Seidel type of iterations where equations at a group of neighboring nodes can be solved implicitly with the latest available iterates from the neighboring zones. The simplest scheme would be to solve the linearized form of the equations (2.3) one at a time with an inner iteration loop.

Locally implicit approximation: The equation (2.3) is linearized and written in the form

$$L(\Delta G)_{j,k,\ell} = Res^{n}_{j,k,\ell} \tag{2.4}$$

where

$$\Delta G = G^{n+1} - G^{n} \tag{2.5}$$

$$\begin{aligned} L(\Delta G)_{j,k,\ell} = {} & CC \cdot \Delta G_{j,k,\ell} \\ & + CJM \cdot \Delta G_{j-1,k,\ell} + CJP \cdot \Delta G_{j+1,k,\ell} \\ & + DJM \cdot \Delta G_{j-2,k,\ell} + DJP \cdot \Delta G_{j+2,k,\ell} \\ & + CKM \cdot \Delta G_{j,k-1,\ell} + CKP \cdot \Delta G_{j,k+1,\ell} \\ & + DKM \cdot \Delta G_{j,k-2,\ell} + DKP \cdot \Delta G_{j,k+2,\ell} \\ & + CLM \cdot \Delta G_{j,k,\ell-1} + CLP \cdot \Delta G_{j,k,\ell+1} \\ & + DLM \cdot \Delta G_{j,k,\ell-2} + DLP \cdot \Delta G_{j,k,\ell+2} \end{aligned} \tag{2.6}$$

$$Res^{n}_{j,k,\ell} = -\mathcal{F}_{j,k,\ell}(G^{n}) + \mathcal{D}_{j,k,\ell}(G^{n}) \tag{2.7}$$

The coefficients CJM, CJP, etc. are matrices which include the Jacobian matrices on the cell faces of the (j, k, ℓ) cell and the dissipation coefficients. DJM, DJP etc., are only diagonal matrices with artificial dissipation coefficients at the cell faces of (j, k, ℓ), and CC is a matrix which can be approximated by a diagonal matrix which includes the time step term and the dissipation coefficients, both real and artificial. The linearized equations (2.4) are solved by an inner iteration for each time step by a modified symmetric Gauss–Seidel iteration.

$$C \cdot dG^{(m)}_{j,k,\ell} = Res^{n}_{j,k,\ell} - L(\Delta G^{(\ )})_{j,k,\ell} \tag{2.8}$$

$$\Delta G^{(m)} = \Delta G^{(m-1)} + dG^{(m)}, m = 1, \ldots, 8 \tag{2.9}$$

The superscript ( ) for ΔG inside the terms of the L operator of (2.8) is either (m) or $(m-1)$ whichever is the latest available value depending on the direction of the iteration sweep. The coefficient C is a modified form of the diagonal matrix CC at j, k, ℓ designed to enhance the stability and convergence characteristics of the scheme for large Courant numbers. Local time stepping enhances the convergence of the scheme for obtaining the steady state solutions. The eight inner iterations start from each of the eight corners of the computational domain to make the scheme symmetric. The initial $\Delta G^{(0)}$ can either be zero or can be the ΔG computed in the previous time step. The symmetric inner iteration is stable and removes any possible instabilities which may arise due to sweep direction being opposite to the flow direction.

The equation (2.8) is now a set of scalar equations and we have explicit expressions for computing the corrections to the solution at each point. Thus the scheme is globally explicit while the inner iteration gives it a locally implicit character. Stability analysis of the model equations indicates that the scheme is unconditionally stable. In practice inner and outer relaxation parameters are introduced for equations (2.5) and (2.9) as follows:

$$G^{n+1} = G^n + \omega_{out} \cdot \Delta G \tag{2.10}$$

$$\Delta G^{(m)} = \Delta G^{(m-1)} + \omega_{in} \cdot dG^{(m)} \tag{2.11}$$

Under relaxation for ω_{out} in the range of 0.67 to 0.95 and over relaxation for ω_{in} from 1 to 1.2 are indicated for good convergence of the scheme at Courant numbers of order 10 and local time stepping.

3. Multi-Grid Technique

One of the most effective methods to speed up the convergence of iterative procedures for the solution of partial differential equations is the multi-grid technique. From a chosen fine grid, a sequence of coarser grids are formed by combining groups of adjacent cells in a systematic manner. For example, by omitting every other mesh line in an even cell numbered grid one can produce a coarse grid and repeating the process the next coarser level grid can be obtained. A set of iteration equations are solved on each of the grids where information from the fine grid is passed to the coarser grids and vice versa in each cycle. The highest frequency errors corresponding to each grid are dissipated rapidly in the iteration process at that grid level and thus errors of all frequencies are damped much more rapidly in the multi-grid mode than in a single grid iteration. For fluid flow problems, Ni[6] and Jameson[7] have applied the multi-grid technique for inviscid transonic flow problems. A summary of the multi-grid technique is outlined below.

At a particular grid level denoted by a mesh spacing parameter h, one or more time step calculations are carried out and the solution is updated. The solution G_h is injected to a coarser grid level indicated by the subscript $2h$ by the volume weighted average.

$$G_{2h} = \left(\sum v_h \, G_h\right) / v_{2h} \tag{3.1}$$

and the residual is injected by the summation over the fine grid cells,

$$RI_{2h} = \sum R_h \tag{3.2}$$

A forcing function FF is defined on the coarse grid as the difference between the injected residual and the Res function (equation (2.7)) calculated on that grid using the injected solution.

$$FF_{2h} = RI_{2h} - Res_{2h} \tag{3.3}$$

The residual for advancing the solution by one or more time steps on the coarse grid is defined by

$$R_{2h} = Res_{2h} + FF_{2h} \tag{3.4}$$

The linearized equation corresponding to (2.4) on the coarse grid is

$$L(\Delta G)_{2h} = R_{2h} \tag{3.5}$$

which is solved for one or more time steps by the locally implicit method outlined in the previous section. This process is repeated till the coarsest grid solution is completed. The coarse grid information is passed back to the finer grids as follows. The difference in the injected solution and the computed solution on the coarse grid is interpolated at the grid points of the next finer mesh and is added to the solution which was previously computed at that level. The solution is recomputed at that level for one or more time steps by the equations (3.4) and (3.5), using the forcing function FF which already exists at that level. This process is continued till the finest grid level is reached where the R and Res functions are the same. This completes one multi-grid cycle. In practice there are many variations to this method. One of the effective methods for fluid flows involves the solution advancement at each grid level for only one time step. Multi-grid scheme propagates the information from one boundary to another in each cycle and allows it to adjust to the boundary conditions quickly to reach a steady-state solution.

4. Three-Dimensional Diffusion Equation

As a test case for the performance of the algorithm, a time dependent diffusion equation is solved with Dirichlet boundary conditions.

$$\frac{\partial u}{\partial t} = \nu \nabla^2 u, \ in D : 0 \leq x, y, z \leq 1 \tag{4.1}$$

$$u = 0 \text{ on } \partial D$$

At $t = 0$, u = Random numbers in (-1, 1) at all grid points

The solution $u \to 0$ as $t \to \infty$. This problem has components at all frequencies supported by the grid and provides the most severe test for the process of convergence of the solution to steady state. The discretization chosen is the Euler implicit scheme, given by

$$\frac{1}{\Delta t}\, \Delta u^n_{i,j,k} = \frac{\nu}{h^2}\, \left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) u^{n+1}_{i,j,k} \tag{4.2}$$

where Δ is forward difference and δ^2 is central difference operator. The scheme has no limitation on grid sizes, which can be non-uniform in all directions. Constant grid size h is indicated here for the presentation of results for the test cases. The scheme (4.2) can be rewritten in the Δ-form as follows:

$$\left\{I - R\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right)\right\} \Delta u^n_{i,j,k} = Res^n_{i,j,k} \tag{4.3}$$

$$Res^n_{i,j,k} = R\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) u^n_{i,j,k}$$

$$R = \frac{\nu \Delta t}{h^2}$$

<u>Locally implicit approximation</u>: The implicit equations (4.3) are solved for Δu's by one or more Gauss-Seidel iterations. This test problem being relatively simple, one Gauss-Seidel iteration per time step without any further modification is adequate. It is stable for any R and is insensitive to the iteration sweep direction, though 8 symmetric sweeps (starting from each of the eight corners of the computational domain) per time step are more desirable for difficult problems, particularly with convection. The iteration for (4.3) can be written as

$$(1+6R)\Delta u^{(m)}_{i,j,k} = Res^n_{i,j,k} + R\Big(\Delta u^{(\,)}_{i-1,j,k} + \Delta u^{(\,)}_{i,j-1,k} + \Delta u^{(\,)}_{i,j,k-1} + \Delta u^{(\,)}_{i+1,j,k} + \Delta u^{(\,)}_{i,j+1,k} + \Delta u^{(\,)}_{i,j,k+1}\Big) \tag{4.4}$$

where the superscript ( ) for Δu on the right side is either (m) or $(m-1)$ depending on the iteration sweep direction. A simple iteration would be with $m = 1$, $\Delta u^{(0)} = 0$ and the iteration sweep carried out in the directions of increasing indices. The multi-grid technique outlined in the last equation is applicable for this problem and is simpler since the equation (4.1) is scalar and linear. In each multi-grid cycle, the number of iterations carried out while going from fine to coarse grids are $1, 2, 2^2$, etc., and the same number in the reverse process. Figures 1–3 show the convergence of the solution u from random numbers to its asymptotic state $u = 0$. $DELU = Max\,|u - u_{asy}| = Max\,|u|$. Work unit is the work required to compute the solution on the finest grid for one time step. For a 20^3 mesh, convergence to more than 4 orders of magnitude is achieved in 60 work units for a single grid method where as it takes only 18 work units in a multi-grid mode with 3 grids. With 40^3 mesh it takes 500 iterations for the same level of convergence with a single grid while it takes only 21 work units in the multi-grid mode with 4 grids. Even with $40 \times 40 \times 80$ mesh it takes only 21 work units with a 4 level multi-grid scheme while it takes more than 600 work units by a single grid scheme. Fig. 4 shows the convergence of $Max\ |Residue|$ against work units for a 40^3 mesh, which again shows a convergence of 4 orders of magnitude in 21 work units.

The locally implicit scheme can also be applied for a group of nodes at a time instead of one node at a time(5). For example, the equations (4.3) at a group of eight adjacent nodes can be solved with a locally implicit approximation. Fig. 5 shows the convergence of such a zonal scheme in both single grid and multi-grid modes. While the convergence is slightly better in terms of number of work units than the single point scheme, actual computational work is many times larger, since the locally implicit equations for each zone have to be solved simultaneously which is time consuming. Thus the single point scheme seems to be the computationally efficient scheme.

5. Conclusions

The locally implicit scheme is a computationally efficient scheme which converges rapidly in multi-grid modes for elliptic problems. It has the promise of providing a rapidly converging algorithm for steady-state viscous flow problems.

6. References

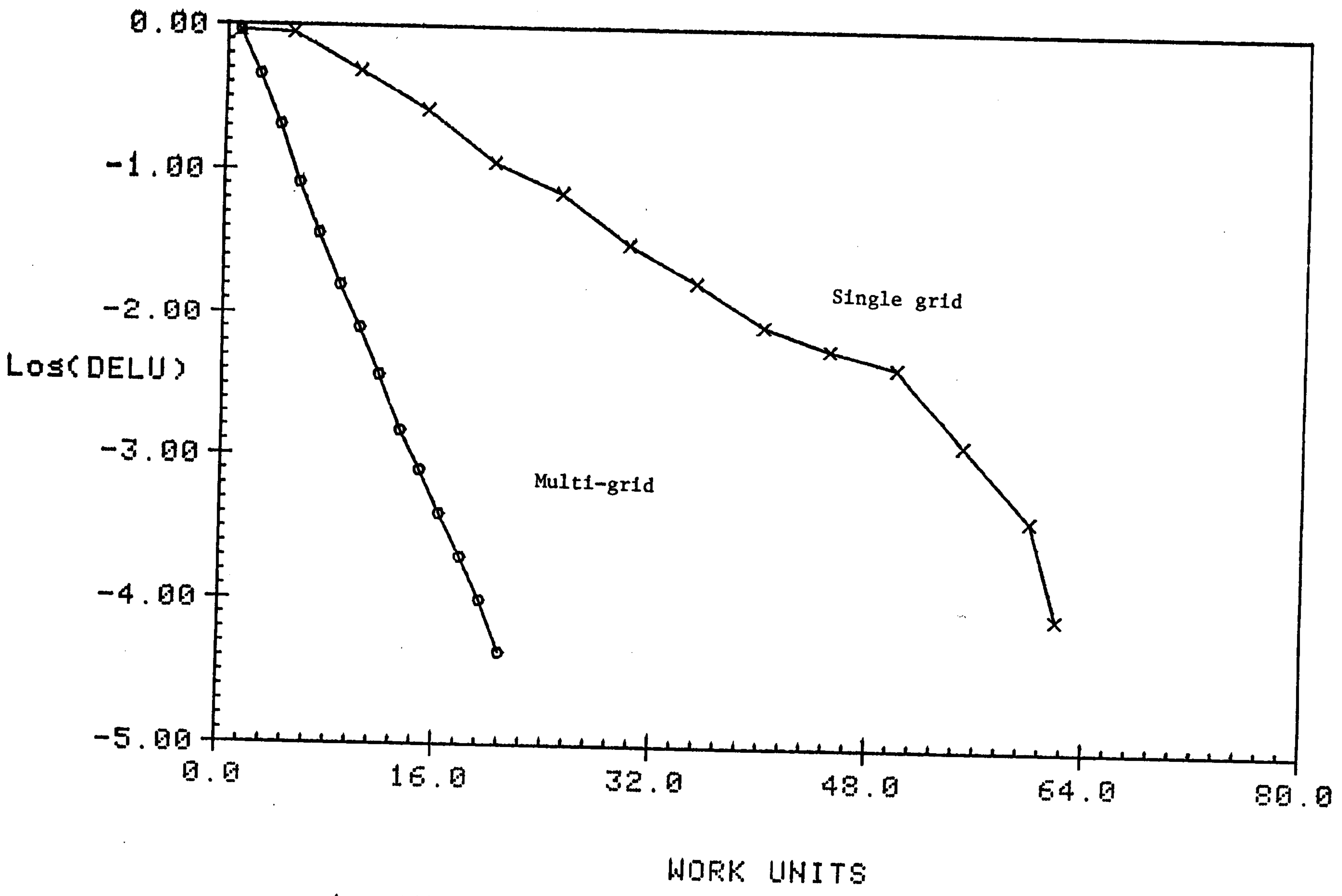
1. Chang, L. C., Kwak, D., Dao, S. C., and Rosen, R., "A Three-Dimensional Incompressible Flow Simulation Method and Its Application to the Space Shuttle Main Engine, Part 1 - Laminar Flow", AIAA-85-0175, AIAA 23rd Aerospace Sciences Meeting, Reno, January 1985.

2. Spradley, L. W., "Applicatin of Computational Methods to Internal and External Fluid Flows", Proceedings of a Workshop on Computational Fluid Dynamics in Aerospace Design, UTSI, Tullahoma, TN, June 1985.

3. Mukhenjee, T., Przekwas, A. J., Tam, L. T., Holland, R. L. and Costes, N. C., "A Multidomain Global-Model Analysis of SSME, Workshop on Computational Fluid Dynamics, NASA Marshall Space Flight Center, Huntsville, AL, April 1986.

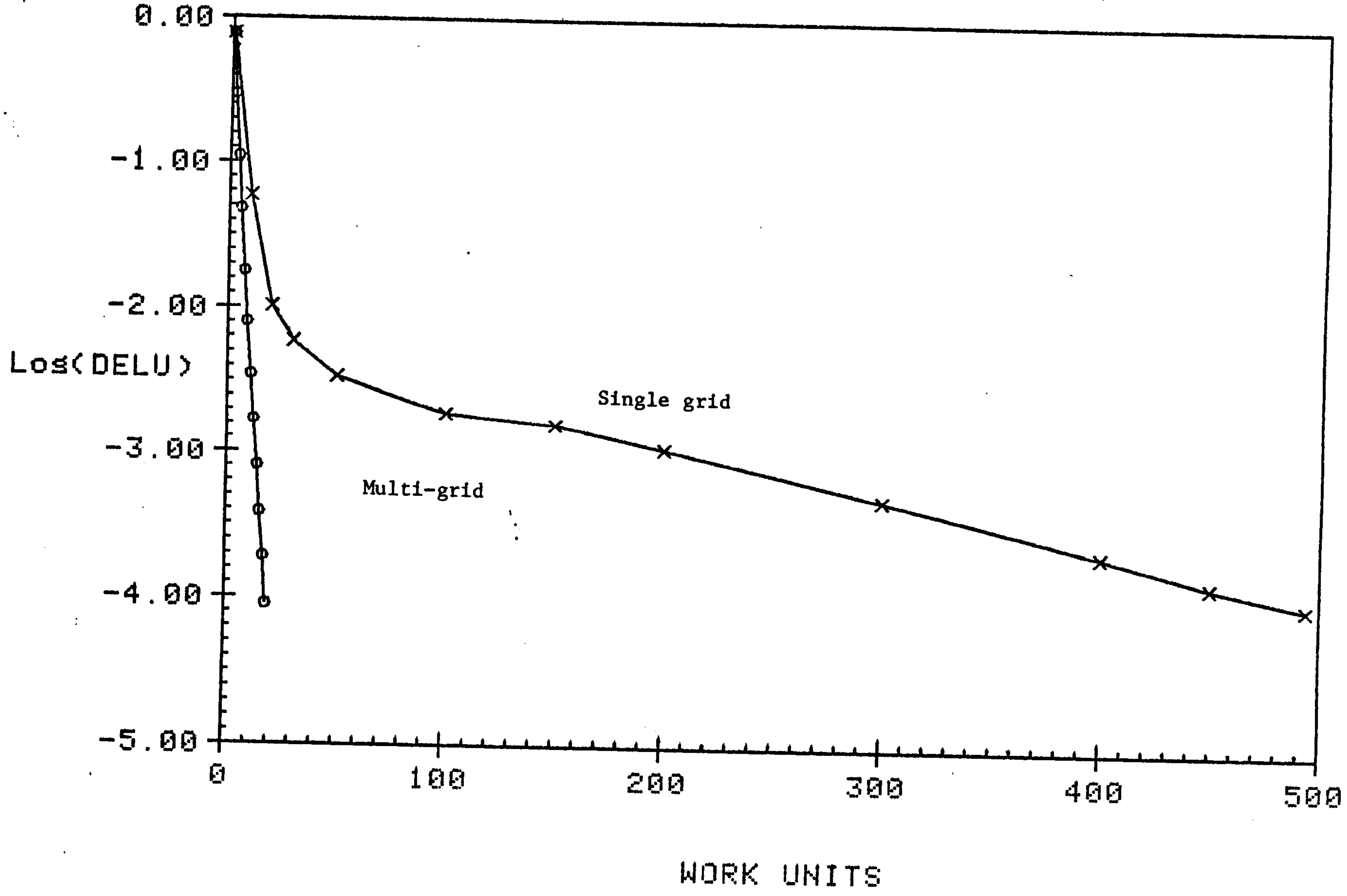
4. Rhie, J., "A General Purpose Navier-Stokes Solver for Internal Propulsion System Flows", AIAA 86-0207, Aerospace Sciences Meeting, January 1986.

5. Reddy, K. C., "A Locally Implicit Scheme for Elliptic Problems", A Workshop on Computational Fluid Dynamics, NASA Marshall Space Flight Center, Huntsville, AL, April 1986.



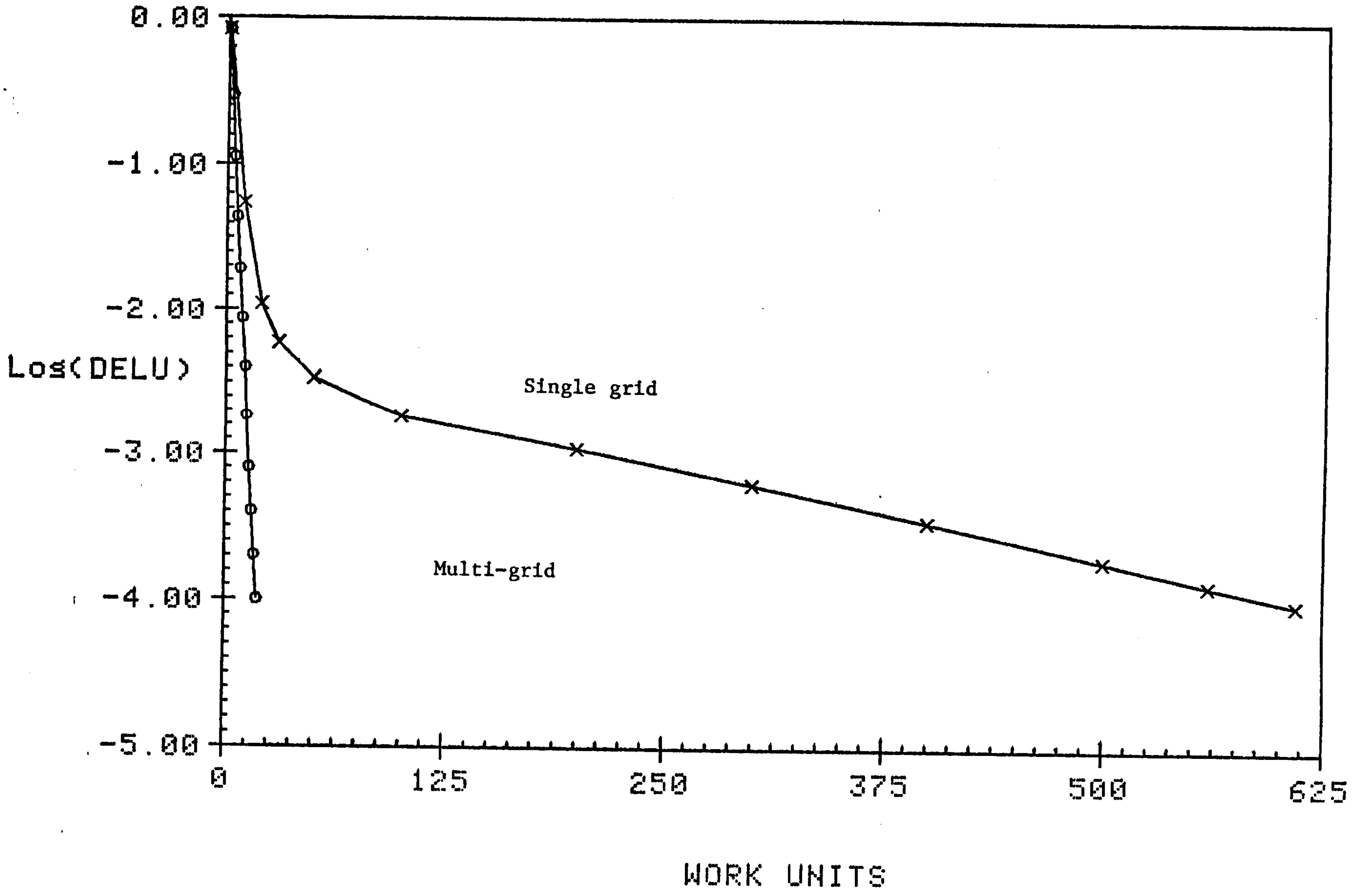
Log(MAX. DELU) VS #WORK UNITS (20x20x20)

Fig. 1



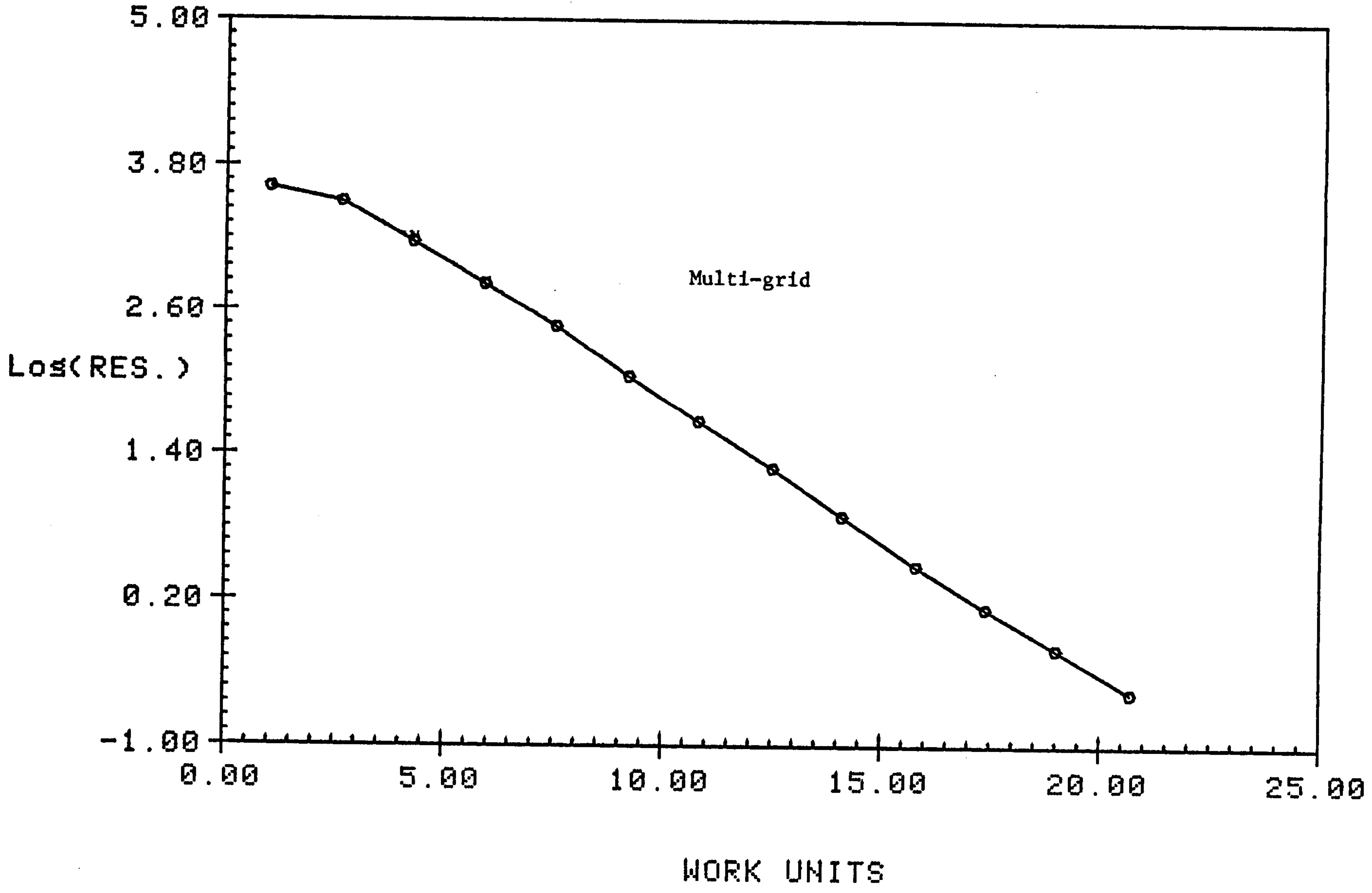
Log(MAX. DELU) VS #WORK UNITS (40x40x40)

Fig. 2



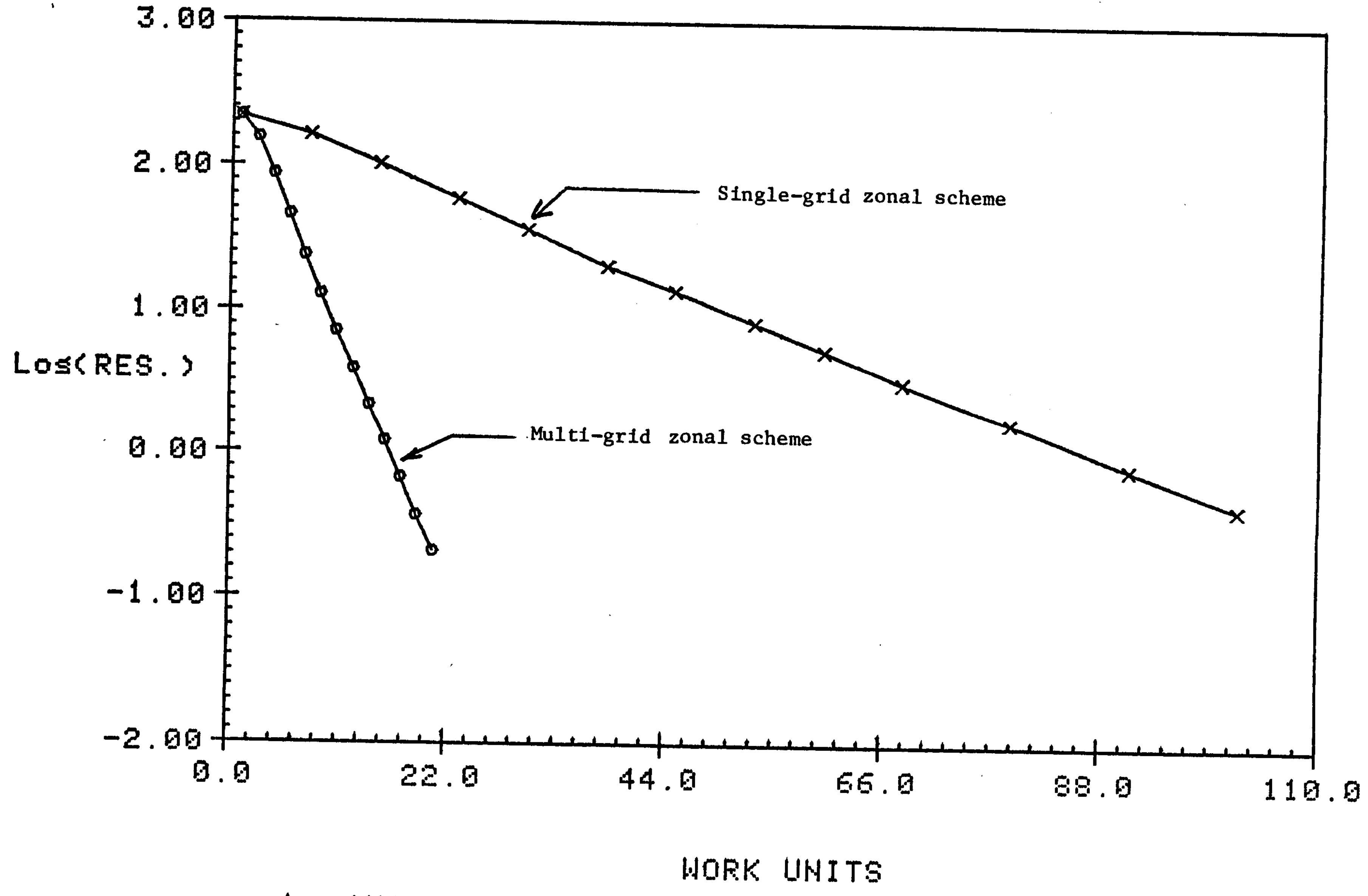
Log(MAX. DELU) VS #WORK UNITS (80x40x40)

Fig. 3



Log(MAX. RESIDUE) VS #WORK UNITS (80x40x40)

Fig. 4



40 x 40 x 40

Fig. 5