An Efficient Parallel Algorithm for Solving Unsteady Nonlinear Equations

Wilson Rivera
Department of Electrical and Computer Engineering
University of Puerto Rico
wrivera@ece.uprm.edu

Jianping Zhu, David H. Huddleston
Engineering Research Center for Computational Systems
Mississippi State University
{jzhu, hudd}@erc.msstate.edu

Abstract

This paper discusses the application of a new parallel non-overlapping domain decomposition algorithm, based on explicit predictors and implicit correctors, to the solution of nonlinear equations. The results demonstrate significant improvement in accuracy for calculating transient solutions using the new approach. In addition, the parallel algorithm scales well as the number of processors increases for large scale problems.

1. Introduction

When solving time dependent PDEs with non-overlapping subdomains, the domain decomposition method could either be used as a preconditioner for Krylov type algorithms [1], or as a means to decompose the original domain into subdomains and solve the PDEs defined in different subdomains concurrently [2]. When it is used as a preconditioner, the relevant PDE is discretized over the entire original domain to form a large system of algebraic equations, which is then solved by Krylov type iterative algorithms. The preconditioning step and the inner products involved in the solution process often incur a significant amount of communication overhead that could significantly affect the scalability of the solution algorithms.

On the other hand, if the original domain Ω is decomposed into a set of non-overlapping subdomains $\Omega_k, k = 1, \cdots, M$, it would be ideal that the PDEs defined in different subdomains could be solved concurrently. This often requires numerical boundary conditions at the boundaries between subdomains.

One way to generate those numerical boundary conditions is to use the solution values from the previous time step t_n to calculate the solutions at t_{n+1} [3]. This is often referred to as time lagging (TL). The other way to generate numerical boundary conditions is to use an explicit algorithm to calculate the solutions at the boundaries between subdomains, using the solutions from the previous time step, and then olve the PDEs defined on different subdomains concurrently using an implicit method [4]. This is referred to as the explicit predictor (EP) method.

In a previous paper [5], the authors showed that the stability and accuracy of the solution algorithm can be significantly affected by the TL and EP methods. The TL algorithm is stable, but in general reduces accuracy for calculating unsteady (transient) solutions. On the other hand, the EP method is accurate, but only conditionally stable. In the referred paper, a new method based on explicit predictor and implicit corrector (EPIC) for generating numerical boundary conditions was discussed. The EPIC method combines the advantages of both the TL (stability) and EP (accuracy) methods.

This paper discusses the application of the EPIC method to the solution of the system of Euler equations in flow simulation of an airfoil and a converging-diverging nozzle. The numerical solution of the Euler equations is discussed in section 2. The implementation details of the new algorithm applied to the solution of the Euler equations are presented in section 3. The numerical results obtained are given in section 4. Finally, the conclusions are listed in section 5.

2. Numerical Solution of the Euler Equations

The two-dimensional Euler equations expressed in Curvilinear coordinates are given by

$$\frac{\partial Q}{\partial \tau} + \frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} = 0, \tag{1}$$

where

$$\begin{aligned} Q &= J^{-1}\hat{Q}, \\ E &= J^{-1}(\xi_x \hat{E} + \xi_y \hat{F}), \\ F &= J^{-1}(\eta_x \hat{E} + \eta_y \hat{F}), \end{aligned} \tag{2}$$

with

$$\hat{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \quad \hat{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{bmatrix}, \quad \hat{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{bmatrix},$$

and equation of state given by

$$p = (\gamma - 1)\{e - \frac{\rho}{2}(u^2 + v^2)\}. \tag{3}$$

Here ρ is the mass density, u and v are the velocity components in the x and y directions, respectively, e is the total specific energy of the fluid, p is the pressure and γ is the ratio of specific heats. The jacobian J, which physically corresponds to the inverse of the cell volume, is given by

$$J = \xi_x \eta_y - \xi_y \eta_x, \tag{4}$$

and the metric terms are

$$\begin{aligned} \xi_x &= y_\eta J, \qquad \eta_x = -y_\xi J, \\ \xi_y &= -x_\eta J, \qquad \eta_y = x_\xi J. \end{aligned} \tag{5}$$

For flow fields which involve continuous variations of the flow field variables, central difference schemes work reasonably well. Explicit formulations of central differences are easy to implement. However, when discontinuities exist in the flow, central difference schemes usually fail because nonphysical oscillations around the discontinuities appear in the numerical solutions. In addition, the inherent conditional stability of these explicit schemes leads to unreasonably long computation times for many problems.

For large scale applications an implicit finite volume formulation is a more convenient choise. In the finite volume formulation, the computational domain is divided into a large number of control volumes so that the partial differential equation representing a conservation law is then integrated over these volumes. The direct discretization of the integral form of the conservation laws ensures conservation of mass, momentum and energy in the discrete level.

The implicit discretized integral form of the two-dimensional Euler equations is given by

$$\frac{Q_{i,j}^{n+1} - Q_{i,j}^{n}}{\Delta \tau} + \frac{E_{i+\frac{1}{2},j}^{n+1} - E_{i-\frac{1}{2},j}^{n+1}}{\Delta \xi} + \frac{F_{i,j+\frac{1}{2}}^{n+1} - F_{i,j-\frac{1}{2}}^{n+1}}{\Delta \eta} = 0 \quad ,$$

or

$$\Delta Q^n + \Delta \tau(\delta_\xi E^{n+1} + \delta_\eta F^{n+1}) = 0, \tag{6}$$

where $\Delta Q^n = Q^{n+1} - Q^n$, and the central difference operator notation indicates that the flux vectors are evaluated at cell faces. The problem of computing the cell-face fluxes for a control volume can be treated as a series of local Riemann problems.

Roe [6] proposed a method of solving a linear problem approximating the original nonlinear Riemann problem. The Roe approximation is expressed as

$$\frac{\partial Q}{\partial t} + \bar{A}(Q_L, Q_R)\frac{\partial Q}{\partial x} = 0, \tag{7}$$

where $\bar{A}(Q_L, Q_R)$ is a constant matrix whose components are evaluated using averaged values of Q at the interface given by

$$\begin{aligned} \bar{\rho} &= \sqrt{\rho_L \rho_R}, \\ \bar{u} &= \frac{\sqrt{\rho_L}u_L + \sqrt{\rho_R}u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \\ \bar{v} &= \frac{\sqrt{\rho_L}v_L + \sqrt{\rho_R}v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \\ \bar{H} &= \frac{\sqrt{\rho_L}H_L + \sqrt{\rho_R}H_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \end{aligned} \tag{8}$$

where $H = \frac{e+p}{\rho}$ is the total enthalpy per unit volume.

The calculation of the flux at cell interfaces can be split into contributions across negative and positive wave speeds. Thus, the required flux can be computed from either of the following expressions

$$\begin{aligned} \bar{E}_{i+\frac{1}{2}} &= E_i + \sum_{j=1}^{m} \alpha_j \lambda_j^{(-)} r_j, \\ \bar{E}_{i+\frac{1}{2}} &= E_{i+1} - \sum_{j=1}^{m} \alpha_j \lambda_j^{(+)} r_j, \end{aligned} \tag{9}$$

where the $(-)$ and $(+)$ superscripts indicate that the calculations involve only negative or positive eigenvalues, respectively. The Roe's numerical flux is then either of the expressions in (9) evaluated using the averages in (8).

The approach discussed previously is only first order accurate due to the assumption of a constant variation of flow properties across a grid cell. Second order (Third order) of accuracy can be obtained by assuming a linear (quadratic) variation of flow properties across a given cell. However,

numerical results obtained using these extensions of the numerical flux to high order accuracy exhibit the oscillatory behavior in the vicinity of discontinuities similar to that encountered with central differences.

In order to avoid oscillations it is necessary to introduce nonlinear components, referred to as flux limiters, to restrict the amplitude of the gradients appearing in the original high order schemes. Osher and Chakravarthy [7] introduced a family of high order accurate schemes using Roe averaging.

3. Domain Decomposition

The algorithm applied to the entire domain without domain decomposition is an implicit finite volume formulation, first order accurate in time with an approximate Riemann solver based on Roe flux approximation to achieve up to third order spatial accuracy as reported by Whitfield et al. [8], that is, the implicit cell-centered finite volume discretization of the Euler equations can be written as

$$\begin{aligned}\mathcal{F}(Q^{n+1}) &= \frac{Q^{n+1}_{i,j} - Q^{n}_{i,j}}{\Delta\tau} + \frac{E^{n+1}_{i+\frac{1}{2},j} - E^{n+1}_{i-\frac{1}{2},j}}{\Delta\xi} \\ &+ \frac{F^{n+1}_{i,j+\frac{1}{2}} - F^{n+1}_{i,j-\frac{1}{2}}}{\Delta\eta} = 0, \end{aligned} \quad (10)$$

where flux vectors at cell faces are calculated using the Roe's approximation defined in (8) and (9) with third-order spatial accuracy as defined in [7].

The original code, provided by the Computational Simulation and Design Center at Mississippi State University, has been modified to deal with domain decomposition. It is important to point out that we are not attempting to improve the original solver. Our focus is on providing fast turn around time for CFD (Computational Fluid Dynamics) simulations without degrading accuracy and convergence.

The scheme used to predict the boundary values between subdomains is the two-step MacCormack scheme written as

$$\begin{aligned}\hat{Q}^{n+1}_{i,j} &= Q^{n}_{i,j} - \Delta\tau\delta_i E(Q^n) - \Delta\tau\delta_j F(Q^n), \\ Q^{n+1}_{i,j} &= \frac{1}{2}\Big[\hat{Q}^{n+1}_{i,j} + Q^{n}_{i,j} - \Delta\tau\delta_{i-1}E(\hat{Q}^{n+1}) \\ &- \Delta\tau\delta_{j-1}F(\hat{Q}^{n+1})\Big]. \end{aligned} \quad (11)$$

The subdomain solutions are obtained using Newton's iterations [8] for the equation $\mathcal{F}(Q^{n+1}) = 0$, that is, for $m = 1, 2, \ldots$

$$\mathcal{F}'(Q^{n+1,m})(Q^{n+1,m+1} - Q^{n+1,m}) = -\mathcal{F}(Q^{n+1,m}), \quad (12)$$

where $\mathcal{F}'(Q^{n+1,m})$ is the Jacobian matrix of the vector $\mathcal{F}(Q^{n+1,m})$. As a consequence, the resulting formulation for the equation in (10) is

$$\begin{aligned}&-F'_{j-\frac{1}{2},j-1}\Delta Q^{n+1,m}_{i,j-1} - E'_{i-\frac{1}{2},i-1}\Delta Q^{n+1,m}_{i-1,j} \\ &\left\{\frac{I}{\Delta\tau} + E'_{i+\frac{1}{2},i} + F'_{j+\frac{1}{2},j} - E'_{i-\frac{1}{2},i} - F'_{j-\frac{1}{2},j}\right\}\Delta Q^{n+1,m}_{i,j} \\ &+E'_{i+\frac{1}{2},i+1}\Delta Q^{n+1,m}_{i+1,j} + F'_{j+\frac{1}{2},j+1}\Delta Q^{n+1,m}_{i,j+1} \\ &= -\left\{\frac{Q^{n+1}_{i,j} - Q^{n}_{i,j}}{\Delta\tau} + \delta_\xi E(Q^{n+1}) + \delta_\eta F(Q^{n+1})\right\}, \end{aligned} \quad (13)$$

where

$$\begin{aligned}\Delta Q^{n+1,m} &= Q^{n+1,m+1} - Q^{n+1,m}, \\ E'_{i+\frac{1}{2},i} &= \frac{\partial E_{i+\frac{1}{2},j}}{\partial Q_{i,j}}, \\ F'_{j+\frac{1}{2},j} &= \frac{\partial F_{i,j+\frac{1}{2}}}{\partial Q_{i,j}}. \end{aligned} \quad (14)$$

The Jacobian E' and F' are calculated numerically by using difference quotients. The first subscript of E' and F' refers to the location of the cell face and the second subscript refers to the location of the dependent variable vector that the numerical flux vector is differentiated with respect to.

Finally, the corrector step based on the Roe's approximate Riemann solver is defined as

$$Q^{n+1}_{i,j} = Q^{n}_{i,j} - \Delta\tau\delta_\xi E(Q^{n+1}) - \Delta\tau\delta_\eta F(Q^{n+1}). \quad (15)$$

The communication between processors is carried out using the MPI standard library [9] to ensure maximum portability. Communications are mostly implemented using the *MPI_Sendrecv* routine. Since this is a locally blocking routine, high synchronization is achieved between processors.

4. Experimental Results

In this section, we presents numerical results obtained for flow simulation of an airfoil and a converging-diverging nozzle followed by a discussion regarding performance.

4.1. NACA 0012 Airfoil

In order to demonstrate performance, applicability and accuracy of the EPIC approach for solving Euler equations, a series of computations for transonic flow about a NACA0012 airfoil pitching about the quarter chourd point are carried out.

The NACA0012 airfoil is prescribed to be pitching at $M_\infty = 0.755$, $k = 0.1628$, $\alpha_m = 0.016$, and $\alpha_0 = 2.51$. The numerical results are compared with the experimental data by Landon [10].

The unsteady calculations are started from a converged steady state solution and afterward a CFL of 1000 was kept constant during the simulation. Figures 1 and 2 show the

pressure distribution for different angles of attack as one pitching cycle advances. Eight subdomains are used in the domain decomposition for this test case.

The pressure distribution obtained with the EPIC approach matches that obtained with the Roe's approximate solver without domain decomposition for each of the angles of attack considered. The TL approach, on the other hand, produces a considerable error in shock locations.

In order to evaluate the significance of the CFL condition in the behavior of the different methods discussed here, unsteady calculations also are carried out maintaining a CFL of 1 constant during the simulation. Figures 3 and 4 show the pressure distribution for the different angles of attack. The results indicate that the TL method is more sensitive to variations of CFL conditions compared to the EPIC method. The EPIC method demonstrates the same high quality results for the two CFL numbers, while the TL method demonstrates a reduction in quality as the CFL number is increased.

4.2. Converging-Diverging Nozzle

As an additional example of temporal accuracy, we consider the movement of unsteady normal shocks around the boundaries between subdomains.

A converging-diverging nozzle is used for the experiment. The dimensions of the nozzle are 80 units of length, 1 unit in the throat section, and 3 units in the exit section. Flow conditions are determined by the ratio of the exit pressure to the inlet stagnation pressure. If the inlet pressure is fixed, the exit pressure can be adjusted to produce various possible flow regimes. The first critical pressure ratio corresponds to the mode of operation where the Mach number increases in the converging part of the nozzle from nearly zero far upstream to 1.0 at the throat, and then decreases again in the diverging portion of the nozzle. For any pressure ratio above the first critical, the nozzle has subsonic flow throughout. As the pressure ratio is lowered bellow the first critical, the flow becomes supersonic just downstream of the throat in the diverging part of the nozzle. A normal shock wave forms somewhere downstream of the throat, and the flow jumps from supersonic to subsonic across this normal shock. As the pressure ratio is lowered further, the shock continues to move toward the exit. Thus, the location of a normal shock in the diverging part of the nozzle can be changed by regulating the exit pressure of the nozzle. This fact is used to move the normal shock through a boundary between subdomains in order to evaluate the behavior of the methods before unsteady shocks.

The computational domain is divided into 4 subdomains. The steady state solution is obtained after 1000 local time steps at a $CFL = 10$. The pressure ratio is equal to 0.90043, and the normal shock occurs at the nozzle axial location where the area ratio is equal to 2.4 just before the boundary between the third and fourth subdomains. At this point the time is set up to the start time t_o, and unsteady calculations are initiated by regulating the pressure ratio. Figure 5 shows the steady state solution at t_0. Figures 6 and 7 show the distribution of the pressure ratio for different time steps. The pressure ratio distribution obtained with the EPIC approach matches that obtained with the Roe's approximate solver without domain decomposition. On the other hand, the TL method exhibits an error in the pressure ratio around the normal shock.

4.3. Scalability

The following couple of figures compare the performance of the TL algorithm vs. the EPIC algorithm on an SGI Power Challenge XL parallel computer with 16 processors.

Figure 8 shows the speedup obtained using a coarse 129×31 C-grid for the computation of the steady state solution at Mach number $M_\infty = 0.85$ and angle of attack $\alpha = 1.0$.

Speedup is defined as the ratio between the sequential execution time and the parallel execution time. For fixed size problems, speedup is limited because of the overhead which grows with increasing number of processors or because the number of processors exceeds the degree of concurrence of the algorithm, that is, the maximum number of tasks which can be executed simultaneously. As a consequence the problem size should be increased in order to achieve an improvement in performance. Efficiency, which is the ratio between the sequential execution time and the cost of the parallel system, may be maintained constant by increasing the problem size as the number of processors increases.

It is clear from the figures that the EPIC algorithm scales well as the number of processors increases for large scale problems. Even a slight superlinear speedup can be observed from the Figure 9. This is mainly due to the nonuniform access latency for different levels of cache and memory on SGI Power Challenge. When more processors are used, the array sizes of the code on each processor become smaller, which preserves data locality better and results in more efficient utilization of local cache and memory.

5. Conclusions

A series of computations for unsteady flow were carried out to demonstrate performance, applicability and accuracy of the EPIC approach for solving systems of nonlinear equations. The numerical results showed that for transient problems the boundary treatment developed here yields significant improvement in accuracy compared to the traditional

time-lagging method. The EPIC method showed better accuracy than the TL method for detecting unsteady shocks. Also, the EPIC method demonstrated high quality solutions at high CFL conditions, while the TL method demonstrated a reduction in quality as the CFL number was increased. In addition, the results showed that the new parallel algorithm is scalable as the number of processors increases.

References

[1] T. F. Chan and T. P. Mathew, "Domain decomposition algorithms," *Acta Numerica*, vol. 3, pp. 1–143, 1994.

[2] D. Drikakis and E. Schreck, "Development of parallel implicit Navier-Stokes solvers on MIMD multiprocessor systems," *AIAA*, 93-0062.

[3] R. Pankajakshan and W. R. Briley, "Parallel solution of viscous incompressible flow on multi-block structured grids using MPI," in *Parallel Computational Fluid Dynamics: Implementation and Results Using Parallel Computers*, pp. 601–608, Elseiver Science, 1996.

[4] C. N. Dawson, Q. Du, and T. F. Dupont, "A finite difference domain decomposition algorithm for numerical solution of the heat equation," *Mathematics of Computation*, vol. 57, pp. 63–71, 1995.

[5] W. Rivera, J. Zhu, and D. Huddleston, "An Efficient Parallel Algorithm with Application to Computational Fluid Dynamics," To appear in *Computers and Mathematics with Applications*.

[6] P. L. Roe, "Approximate Riemann solvers, parameter vector, and difference schemes," *Journal of Computational Physics*, vol. 43, pp. 357–372, 1981.

[7] S. Osher and S. Chakravarthy, "Very high order accurate TVD schemes," Report 84-44, ICASE, 1984.

[8] D. L. Whitfield, J. M. Janus, and L. B. Simpson, "Implicit finite volume high resolution wave split scheme for solving the unsteady three-dimensional Euler and Navier-Stokes equations on stationary or dynamic grids," Engineering and Industrial Research Report MSSU-EIRS-ASE-88-2, Mississippi State University, 1988.

[9] W. Gropp, M. Snir, B. Nitzberg, and E. Lusk, *MPI: The Complete Reference*. MIT Press, 1998.

[10] R. H. Landon, "NACA0012 oscillation and transient pitching," in Compendium of Unsteady Aerodynamic Measurements, Advisory Report 702, AGARD, 1982.

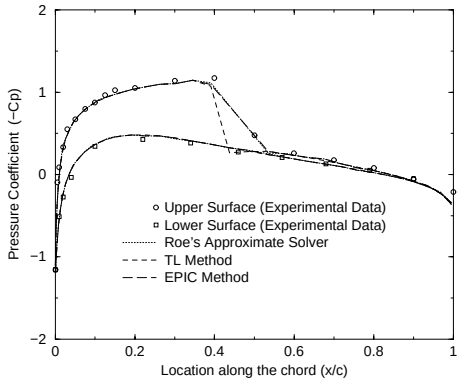

Figure 1. NACA 0012 unsteady pressure distribution: $\alpha = 2.34$**.**

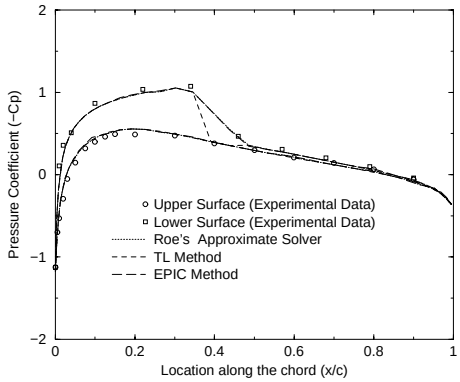

Figure 2. NACA 0012 unsteady pressure distribution: $\alpha = -2.41$**.**

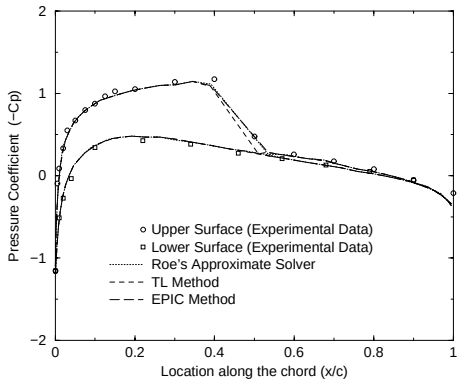

Figure 3. NACA 0012 unsteady pressure distribution: $\alpha = 2.34$**,** $CFL = 1$**.**

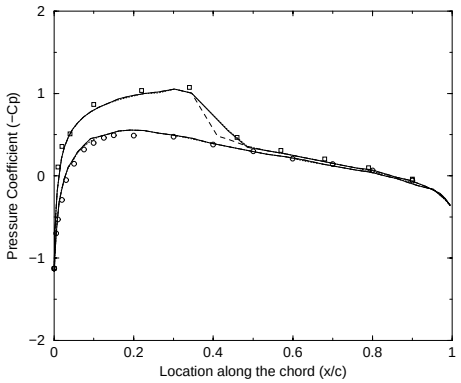

Figure 4. NACA 0012 unsteady pressure distribution: $\alpha = -2.41$, $CFL = 1$.

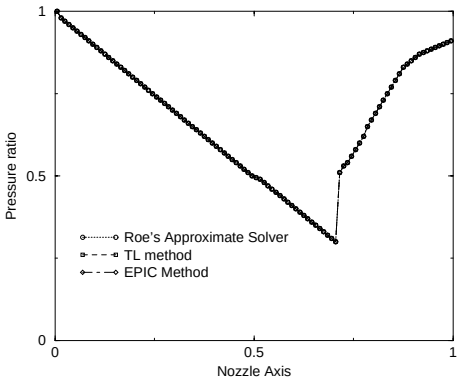

Figure 5. Pressure ratio at $t = t_0$.

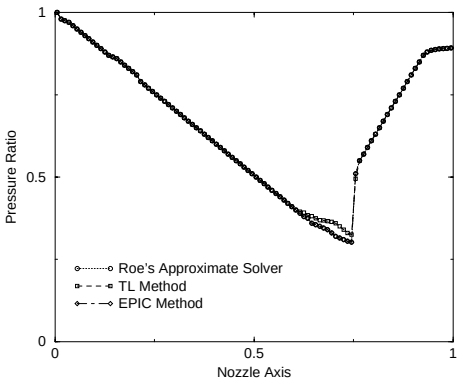

Figure 6. Pressure ratio at $t = t_0 + 0.0001$.

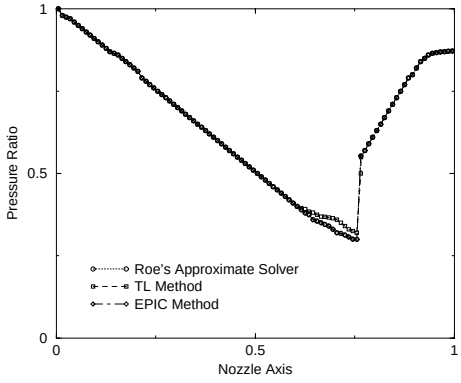

Figure 7. Pressure ratio at $t = t_0 + 0.0002$.

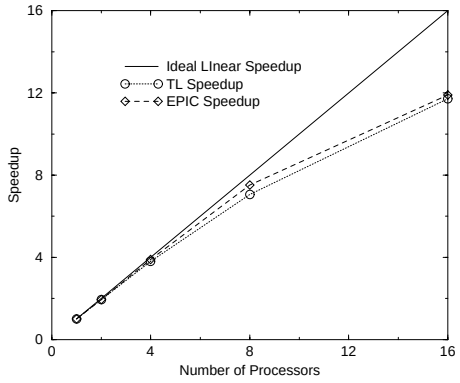

Figure 8. Speedup: 129×31 grid for the NACA0012 airfoil

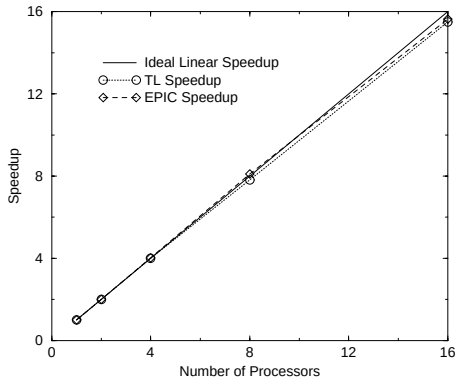

Figure 9. Speedup: 290×81 grid for the NACA0012 airfoil