

PROCEEDINGS OF THE SIXTH SIAM CONFERENCE ON

PARALLEL

PROCESSING FOR

SCIENTIFIC

COMPUTING

Volume I

Edited by Richard F. Sincovec

Oak Ridge National Laboratory

David E. Keyes

Yale University

Michael R. Leuze

Oak Ridge National Laboratory

Linda R. Petzold

University of Minnesota

Daniel A. Reed

University of Illinois

siam.

Philadelphia

Society for Industrial and Applied Mathematics

Parallelization of a Godunov Type Method for the Solution of Systems of Conservation Laws*

Ken Bridges[†] Marcin Paprzycki[‡] Xuefeng Li[‡]

Abstract

High-order Godunov type methods are often used to solve systems of conservation laws. In their most common form these are explicit methods and so very well suited for parallelization. We discuss parallelization of a particular Godunov type method. Results of experiments performed on a 20 processor SEQUENT are reported.

1 Introduction

Consider the hyperbolic conservation law [1, 2]

$$(1) \quad \begin{aligned} \partial_t u + \partial_x f(u) &= 0, & (x, t) \in (-\infty, +\infty) \times (0, +\infty) \\ u(x, 0) &= u_0(x), & x \in (-\infty, +\infty) \end{aligned}$$

where $u = (u_1, u_2, \dots, u_s)^T$, $f(u) = (f_1(u), f_2(u), \dots, f_s(u))^T$. Assume that the x - t plane is discretized by introducing mesh points (x_j, t_n) . A class of numerical schemes of Godunov type to solve (1) can be defined as:

$$(2) \quad v_j^{n+1} = v_j^n - \lambda \left[f_{j+\frac{1}{2}}^n - f_{j-\frac{1}{2}}^n \right]$$

$$(3) \quad f_{j+\frac{1}{2}}^n = \sum_k \alpha_k f \left(v_{j+\frac{1}{2}}^{\gamma_k} \right)$$

where (3) is a numerical quadrature (with coefficients α_k and γ_k), the CFL restriction is $0 < \lambda < 0.5$, v_j^n is the approximate solution at time n , $v_{j+\frac{1}{2}}^{\gamma_k}$ is an approximation to the interface value between the grid cells, and $f_{j+\frac{1}{2}}^n$ represents the flux. Different numerical quadratures used in (3) lead to different methods.

In [2], one of the authors introduced a new method of Godunov type. His scheme is essentially non-oscillatory (ENO) and is based on replacing the evolution operator by an exact Riemann solver with Lagrangean interpolation. The new scheme uses adaptive moving stencil to obtain smooth information about the solution and a discontinuity sharpening technique to prevent the loss of information about shock and contact discontinuities (see [2], for more details). Analysis of the numerical properties of the new method are discussed in [2] and will be omitted here. The aim of this note is to present the preliminary results of an attempt to parallelize the new method.

* Computer time grant from the Department of Computer Science Engineering of the Southern Methodist University is kindly acknowledged.

[†] Department of Mathematics and Computer Science, University of Texas of the Permian Basin, Odessa, TX 79762

[‡] Department of Mathematics, Loyola University, New Orleans, LA 70118

2 Experimental results

The Godunov type methods are time explicit and as such should naturally yield to parallelization. For each subinterval, the solution for the next time step depends only on the solutions from the previous step and as such can be calculated independently on separate processors. We have attempted to parallelize the new scheme on a shared memory, 20-processor Sequent Symmetry computer. We have used the Sequent provided parallelization primitives (mainly *cfdoucrass*). Since one of the processors is running the operating system we have experimented with up to 19 processors. Each run was performed on an empty machine and the results presented here are averages of multiple runs.

The first problem solved was a scalar linear equation

$$\begin{aligned}\partial_t u + \partial_x u &= 0 \\ u(x, 0) &= u_0(x), \quad -1 \leq x \leq 1\end{aligned}$$

with periodic boundary conditions at $x = \pm 1$, and the initial function

$$u_0(x + \frac{1}{2}) = \begin{cases} x \sin\left(\frac{3}{2} \pi x^2\right), & -1 \leq x < -\frac{1}{3} \\ |\sin(2 \pi x)|, & -\frac{1}{3} \leq x < \frac{1}{3} \\ 2x - 1 - \frac{\sin(3 \pi x)}{6}, & \frac{1}{3} \leq x < 1 \end{cases}$$

Speedups obtained for varying number of meshpoints are presented in Figure 1.

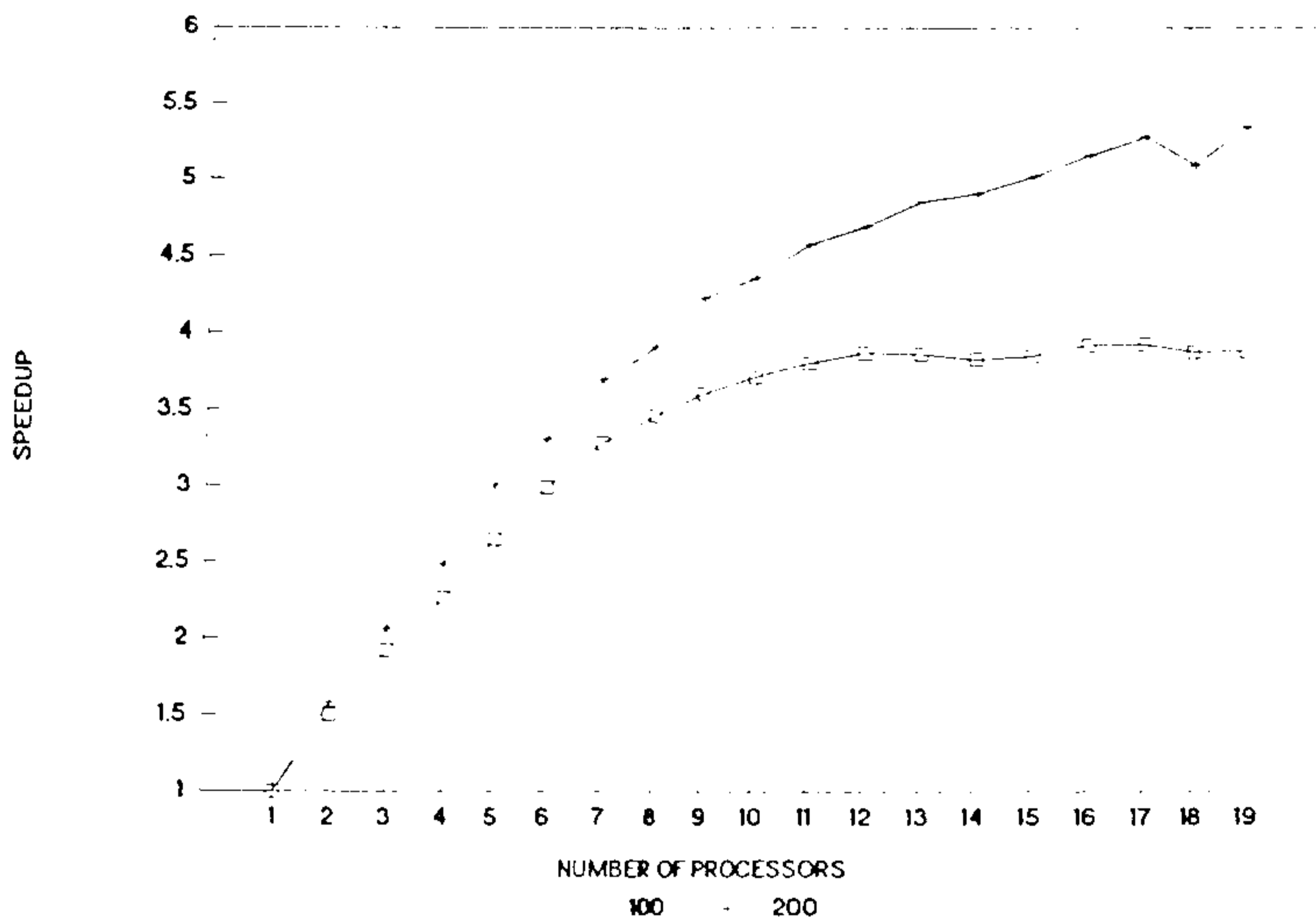


FIG. 1. Speedup for varying meshsizes (scalar linear equation).

It is worth noting that 100 meshpoints are all that is necessary to obtain an accurate solution of this problem. We have run this problem for up to 1200 meshpoints and we have found that for more than 400 meshpoints the maximum speedup becomes fixed between 6 and 7. This is most likely an indication that the communication network (system bus) was saturated.

We have also solved a Riemann problem

$$\begin{aligned} \partial_t u + \partial_x f(u) &= 0, & u(x, 0) &= u_l, & 0 \leq x < 0.1 \\ & & u(x, 0) &= u_m, & 0.1 \leq x < 0.9 \\ & & u(x, 0) &= u_r, & 0.9 \leq x \leq 1.0 \end{aligned}$$

where

$$\begin{aligned} u &= (\rho, \rho q, e)^T, & f(u) &= (\rho q, \rho q^2 + p, q(e + p))^T, \\ e &= \rho \epsilon + \frac{1}{2} \rho q^2 & \text{and} & \quad \epsilon = \frac{p}{(\delta - 1)\rho} \end{aligned}$$

where ρ is density, p — pressure, q — velocity, e — total energy per unit volume, ϵ — internal energy per unit mass, $\delta (> 1)$ — heat ratio; and the initial values

$$(\rho_l, q_l, p_l) = (1.0, 0.0, 1000.0)$$

$$(\rho_m, q_m, p_m) = (1.0, 0.0, 0.01)$$

$$(\rho_r, q_r, p_r) = (1.0, 0.0, 100.0)$$

Speedups obtained for varying numbers of meshpoints are summarized in Figure 2.

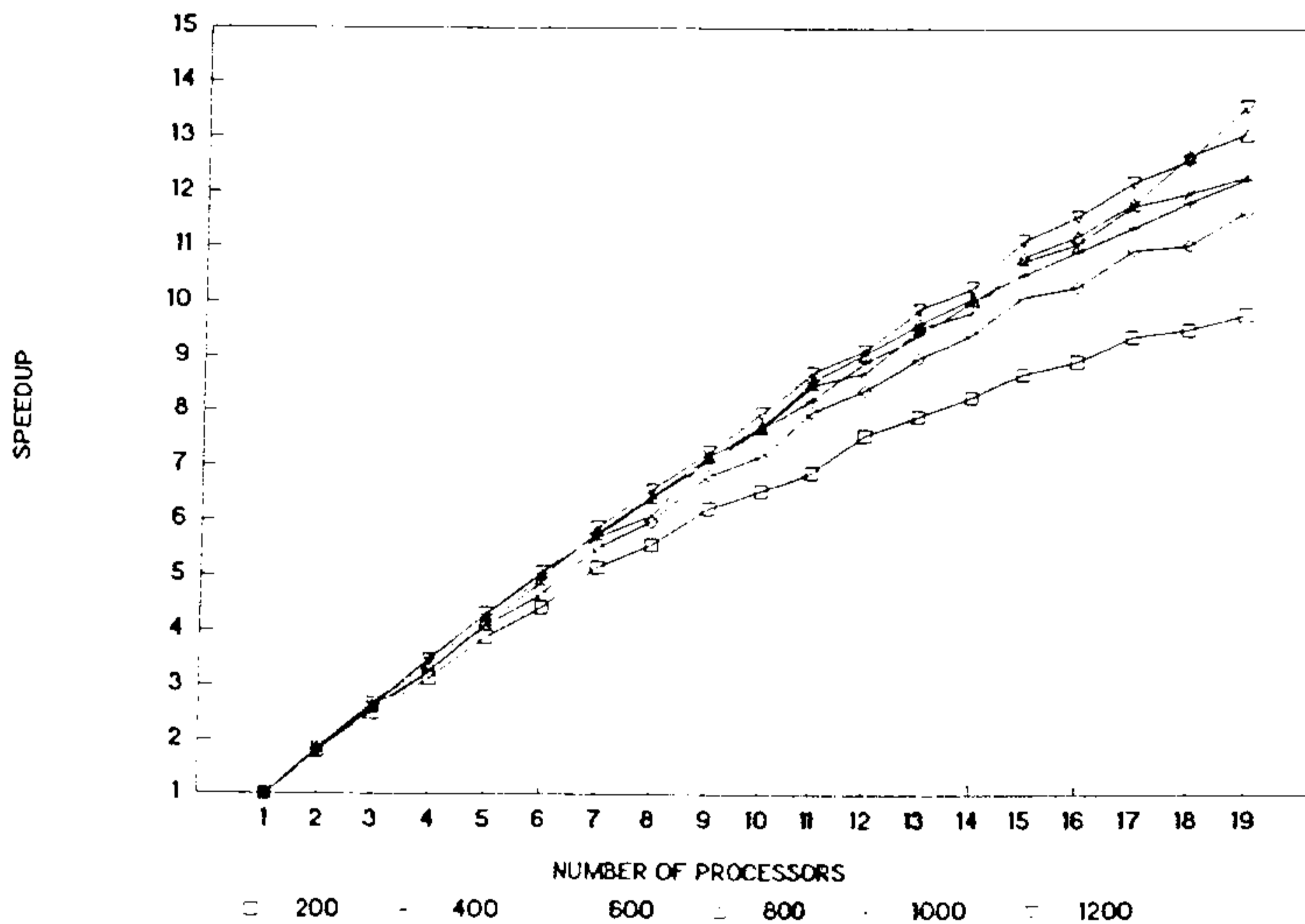


FIG. 2. Speedup for varying meshsizes (Riemann problem).

It should be noted that the number of meshpoints necessary to solve this problem accurately is 1200. It can be observed that for more than 800 meshpoints the maximum speedup (on 19 processors) becomes fixed between 12 and 14. The fact that the Riemann problem leads to higher speedups can be explained by the fact that, as a system of equations, it requires greater amount of work for each grid point thus reducing the ratio of communication to the work performed by each processor.

3 Conclusions

The presented results indicate that Godunov type methods for the solution of systems of conservation laws should parallelize quite nicely on shared memory MIMD computers. The primary problem one is confronted with is the bus saturation when only a minimal amount of work is necessary for each grid cell.

In the follow-up, we plan to run more experiments on the Sequent as well as attempt to parallelize Li's method on a Cray. We will also attempt to develop a mathematical model of the behavior of our code for a variety of problems on shared memory parallel computers.

References

- [1] LeVeque, R.J., *Numerical Methods for Conservation Laws*, Birkhauser, Basel, 1992.
- [2] Li, X., *Efficient ENO Schemes for Hyperbolic Conservation Laws*, submitted for publication.