Several aspects of adaptive methods for partial differential equations implemented on vector and parallel computers were investigated on this effort. A new techniques for mapping mesh points to processors in a static way has been developed, this takes advantage of the structure of the family of solutions without singling out any one solution. Three publications and two technical reports resulted from this effort, as well as two conference proceedings papers and four presentations. Papers included such titles as "A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation", "Local uniform mesh refinement on loosely-coupled parallel processors", and "Dynamic grid manipulation for partial differential equations on hypercube parallel processors".
Abstract

Over the period of this contract, William Gropp has investigated several aspects of adaptive methods for partial differential equations on vector and parallel computers. The work has included the porting of several adaptive codes to run in parallel on several parallel machines and the comparison of various domain decomposition algorithms for elliptic partial differential equations. A new technique for mapping mesh points to processors in a static way has been developed; this takes advantage of the structure of the family of solutions without singling out any one solution. Experience with the parallel codes has shown that good efficiencies can be reached without a disproportionate amount of work; however, data communication times are important and can reduce the benefit of using parallel computers. Early results with domain decomposition methods indicate that they do work well on parallel computers, but they don't have an overwhelming advantage over more conventional approaches.

1. Research Objectives

The purpose of this work is to study the application of a particular adaptive method, Local Uniform Mesh Refinement (LUMR) to vector and parallel computers. The specific areas to be studied were

- Solution of elliptic equations.
- Where and how to refine, preserving the effectiveness of vector and parallel computers.
- Programming and data structures.

Some of the work is also applicable to other adaptive methods; it is hoped that these results may point out how other adaptive techniques may be used on vector and parallel computers.

2. Progress

2.1. LUMR on parallel processors

A version of the LUMR code which handles the data structure manipulations was implemented and tested on the Intel iPSC-860 hypercube which was acquired by the Computer Science Department at Yale. These experiments showed that, just as in vector computing, there is a minimum size or granularity below which various overheads become dominant. The results were presented in a technical report, to appear in a SIAM proceedings.

An analysis of the time complexity of running a 2-d adaptive mesh refinement code on a loosely-coupled parallel computer was developed. The model of a parallel computer used included data communication delays; these terms proved crucial in predicting the observed performance. This analysis allowed suggested specific implementations of certain operations in the code. To test these results, a 2-d version of a Local Uniform Mesh Refinement code was developed and ported to a ring of Apollo Workstations. The results of these tests validated the theory, and suggest that loosely-coupled parallel processors will work even for adaptive techniques if they are designed with enough communications bandwidth. Further, it was relatively easy to move a properly designed serial code to a parallel processor. This suggests that, once the newness of parallel computing fades, programming a parallel computer may not be much harder than a single computer as long
as some care is taken in the initial design. William Gropp’s results give a way to determine what
is enough for these applications. These results were presented in a technical report, and submitted
for publication to Computers and Mathematics.

Another result of these studies was an appreciation that a memory access model serves as a
good way to view both message passing and shared memory parallel computers. This is similar to
the way in which algorithms for vector computers have been discovered by optimizing the memory
access patterns in existing techniques. This was discussed by William Gropp at a workshop, and
will appear in the proceedings of that meeting.

2.2. Domain Decomposition

One difficult part of solving partial differential equations in parallel is solving the linear systems
that arise from implicit discretizations. With Dr. David Keyes, William Gropp has been evaluating
most of the published methods of Domain Decomposition, examining their relative performance
and their suitability for real (rather than theoretical) parallel computers. The early results of this
work show that these techniques are indeed applicable to the problem of solving linear systems on
parallel computers, but do not offer a major advance, particularly on general problems.

2.3. Mesh distribution

With Dr. Ilse Ipsen, William Gropp has been investigating a new method for distributing
work among parallel processors. This work is based on viewing an adaptive method for partial
differential equations as a graph problem, where the graphs are not arbitrary, but rather have
a certain locality based on the continuity of the solution. The method is a new form of Gray-
code, called an interleaved Gray-code, which allows easy labeling of graph nodes even when the
maximal level of refinement is unknown, allows easy determination of nearby nodes in the graph, is
completely deterministic, and often (in a well-defined sense) distributes the graph efficiently across
a hypercube. This work has been presented at an international meeting on parallel computing, and
will appear as a technical report.
3. Publications and reports

Publications


Proceedings


Technical reports


Presentations


END
12-87
DTC