

2

NASA Contractor Report 191552  
ICASE INTERIM REPORT NO. 25

AD-A274 649  
[Barcode]

# ICASE



## EXPERIENCE WITH PARAMETRIC BINARY DISSECTION

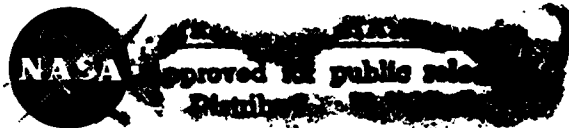
Shahid H. Bokhari

DTIC  
ELECTE  
JAN 11 1994  
S E D

NASA Contract No. NAS1-19480  
October 1993

Institute for Computer Applications in Science and Engineering  
NASA Langley Research Center  
Hampton, Virginia 23681-0001

Operated by the Universities Space Research Association



14px

94-01110



National Aeronautics and  
Space Administration  
Langley Research Center  
Hampton, Virginia 23681-0001

94 1 10 103

# Experience with Parametric Binary Dissection\*

Shahid H. Bokhari

*Department of Electrical Engineering  
University of Engineering & Technology  
Lahore, Pakistan*

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input checked="" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification _____	
By _____	
Distribution / _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

## Abstract

Parametric Binary Dissection (PBD) is a new algorithm that can be used for partitioning graphs embedded in 2- or 3-dimensional space. It partitions explicitly on the basis of  $nodes + \lambda \times (edges\ cut)$ , where  $\lambda$  is the ratio of time to communicate over an edge to the time to compute at a node. The new algorithm is faster than the original binary dissection algorithm and attempts to obtain better partitions than the older algorithm, which only takes nodes into account.

We compare the performance of parametric dissection with plain binary dissection on 3 large unstructured 3-d meshes obtained from computational fluid dynamics and on 2 random graphs. We show that the new algorithm can usually yield partitions that are substantially superior, but that its performance is heavily dependent on the input data.

\*Research Supported by NASA Contract NAS-1-19480, while the author was resident at the Institute for Computer Applications in Science & Engineering, NASA Langley Research Center, Hampton, Virginia.

# 1 Introduction

In order to fully utilize parallel computers, it is crucial to uniformly partition the domain over which computations are to be performed. This problem is known to be computationally intractable and a number of heuristics have been developed for its solution.

Binary dissection or orthogonal recursive partitioning, developed in 1985 by Berger & Bokhari [2, 3], is a partitioning technique that is in widespread use [1, 5, 6]. This is a fast and straightforward algorithm that carries out partitioning as a series of recursive bisections that minimize the load at each step. This algorithm does not take communication costs into account and can sometimes yield partitions that have poor communicate to compute ratio.

The solution of aerodynamic problems on unstructured meshes is an important area of research within the field of computational fluid dynamics. Unstructured meshes are graphs embedded in 2- or 3-dimensional space. The current requirement is to solve large ( $\approx 10^5$  node &  $10^6$  edge) problems on parallel computers such as the Intel iPSC-860 hypercube or PARAGON 2-d mesh. Efficient utilization of these parallel machines requires good partitioning of meshes over the processors of the system.

When binary dissection is used for partitioning, the nodes of the problem mesh are uniformly distributed over all processors but, of course, no attention is paid to the number of edges that are cut. Each edge that is cut by the partitioning results in an inter-processor communication requirement. We normalize the time required to compute at a node to 1, and denote by  $\lambda$  the time required to communicate over an edge. The normalized time required by a specific partitioning of a problem mesh is then equal to the maximum of  $nodes + \lambda \times (edges\ cut)$  over all subregions.

Parametric Binary Dissection (PBD) [4] is a new technique that attempts to take communication overhead into account by partitioning on the basis of load as well as communication cost. At each step of the dissection, an attempt is made to minimize the  $nodes + \lambda \times (edges\ cut)$  for the two subregions. A fast algorithm for PBD is given in [4]. Since PBD becomes ordinary binary dissection when  $\lambda = 0$ , this fast algorithm also serves to solve the original problem more rapidly.

In this paper we evaluate the performance of Parametric Binary Dissection on 3 unstructured meshes taken from aerodynamic applications. The meshes that we use for our evaluation are as follows.

Mesh	Nodes	Edges	Provided by
Wing & Pod	106064	697992	Dimitri Mavriplis
F-18	316399	2106889	Clyde Gumbert
Wing & Store	121200	818066	Neil Frink

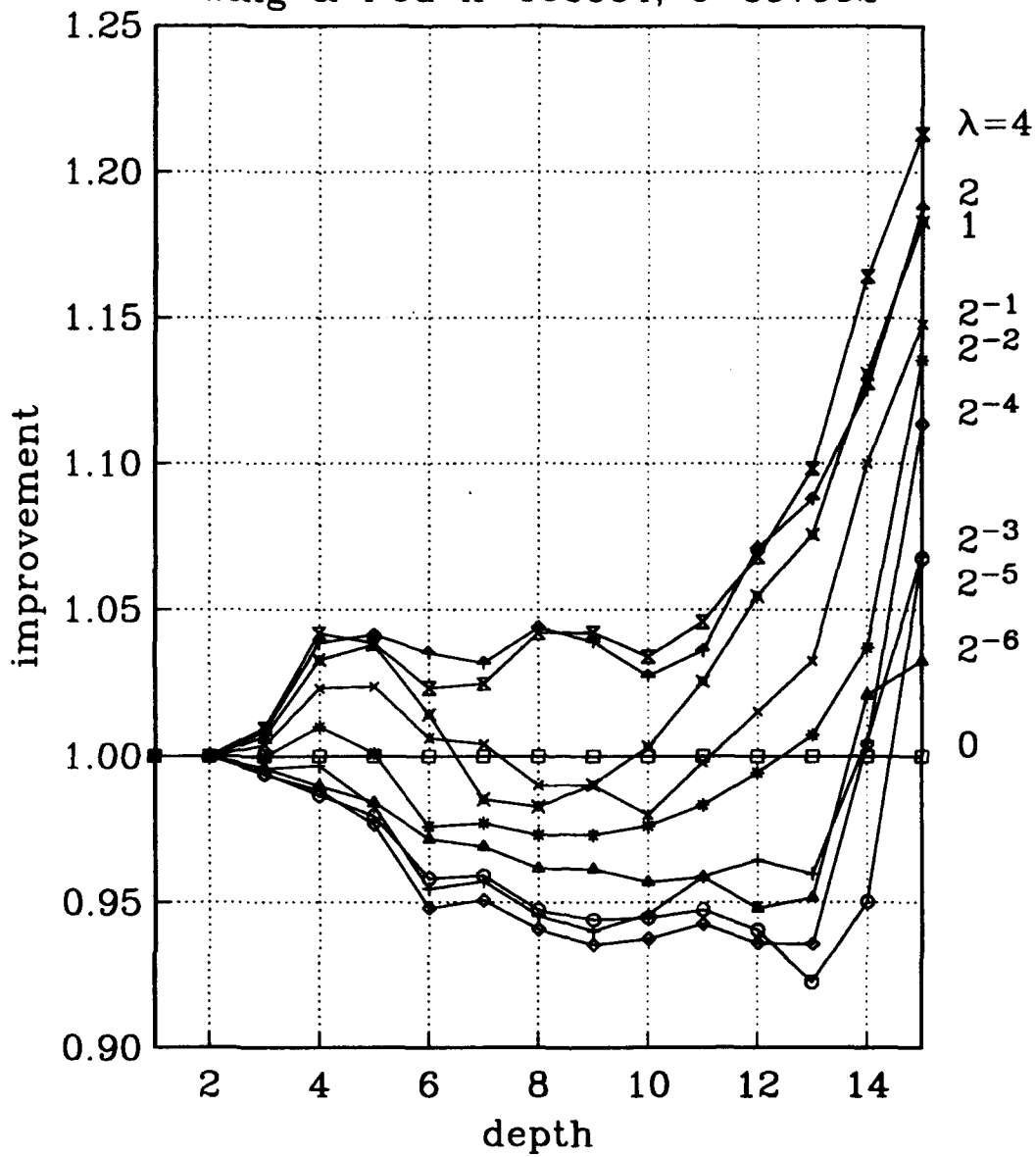
We also provide results for the dissection of two large random 3-d graphs.

We evaluate PBD by applying it to each mesh for  $\lambda = 0, 2^{-6}, \dots, 1, \dots, 2^2$ , for depths of partitioning varying from 1 to about 18. Since we are dealing with binary dissection, each level of partitioning doubles the number of regions. Thus a depth 4 partitioning results in  $2^4$  regions and would be targeted to a 16 processor system. For each partition we obtain the maximum number of edges cut and the maximum number of nodes over all regions. The normalized run time of a partition is then  $\max \text{ nodes} + \lambda \times (\max \text{ edges cut})$ . For  $\lambda = 0$  PBD degenerates into plain binary dissection. We can thus compare the performance of plain and parametric dissection by dividing the normalized run time at every depth for  $\lambda = 0$  with the run time at various non-zero values of  $\lambda$ . These ratios give us the improvement of PBD over plain dissection and are plotted in the following Sections.

## 2 Wing and Pod

This mesh, provided by Dimitri Mavriplis, is based on half a fuselage, wing and an engine. It has 106064 nodes and 697992 edges. When  $\lambda = 0$ , Parametric dissection is the same as ordinary dissection and there is no performance advantage. For depths 7-11 there is degradation in performance, except for large values of  $\lambda$ . For depths 3-6 and 11-15 there is performance improvement and this increases with  $\lambda$ . The time taken for PBD to depth 16 on this mesh is 230 seconds on a Sparcstation-10. This time includes 67 seconds to input the mesh.

Wing & Pod n=106064, e=697992



### 3 F-18

The F-18 mesh was provided by Clyde Gumbert and has 316399 nodes and 2106889 edges. The Parametric Dissection algorithm could not provide *any* performance improvement for this mesh. Careful analysis of the mesh revealed that the wings are exactly at the midpoint (in terms of nodes) of the domain. Thus the first two cuts tend to pass *through* the wings which, of course, contain no mesh elements. The following figure is a simplified representation of a slice through the mesh (grey area) with the first cut passing through the wing. The nose of the plane points towards the observer—only half the plane is shown.

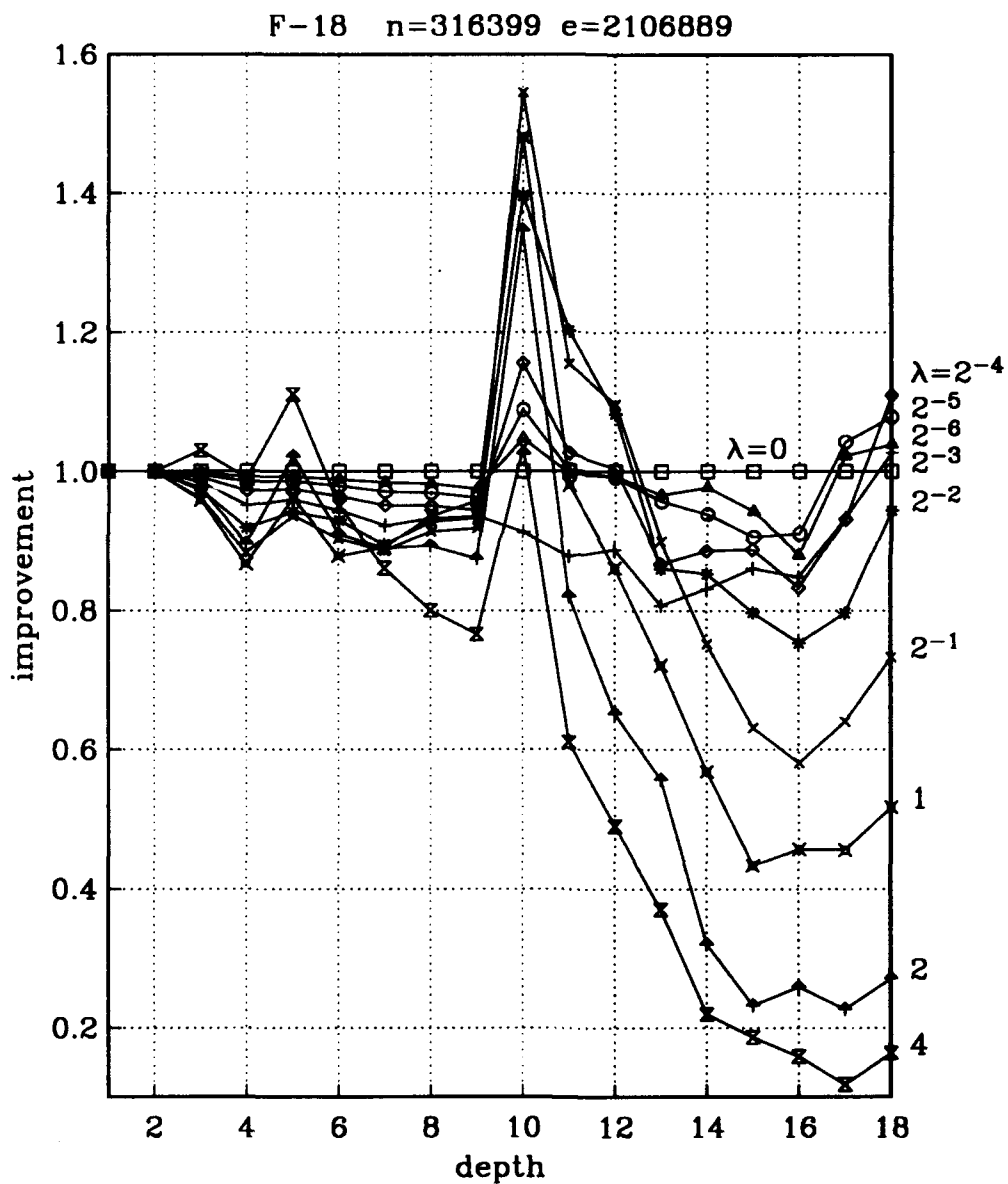


To verify this conclusion, we redid the experiment after applying a rotation to the node coordinates. The rotation that we used was an arbitrarily chosen  $79^\circ$  each about the  $x$ ,  $y$  and  $z$  axes, in that order <sup>1</sup>. The performance on the rotated mesh is better, yielding very high performance improvements

---

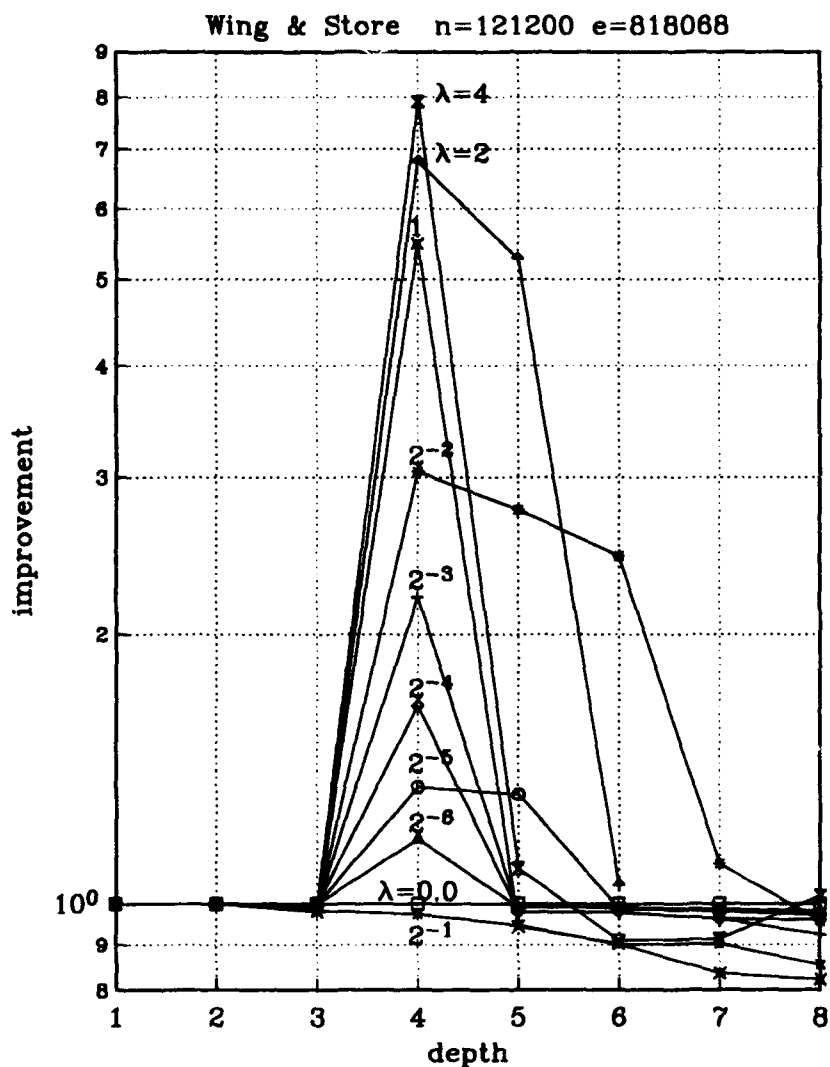
<sup>1</sup>Any set of rotations large enough to move the first 2 or 3 cuts out of the body of the plane will suffice.

for certain depths, but is not as dramatic as for the previous case. The time required for a depth 18 dissection of this mesh is 505 seconds on a 50-MHz MIPS R4000 processor; this includes 82 seconds for input.



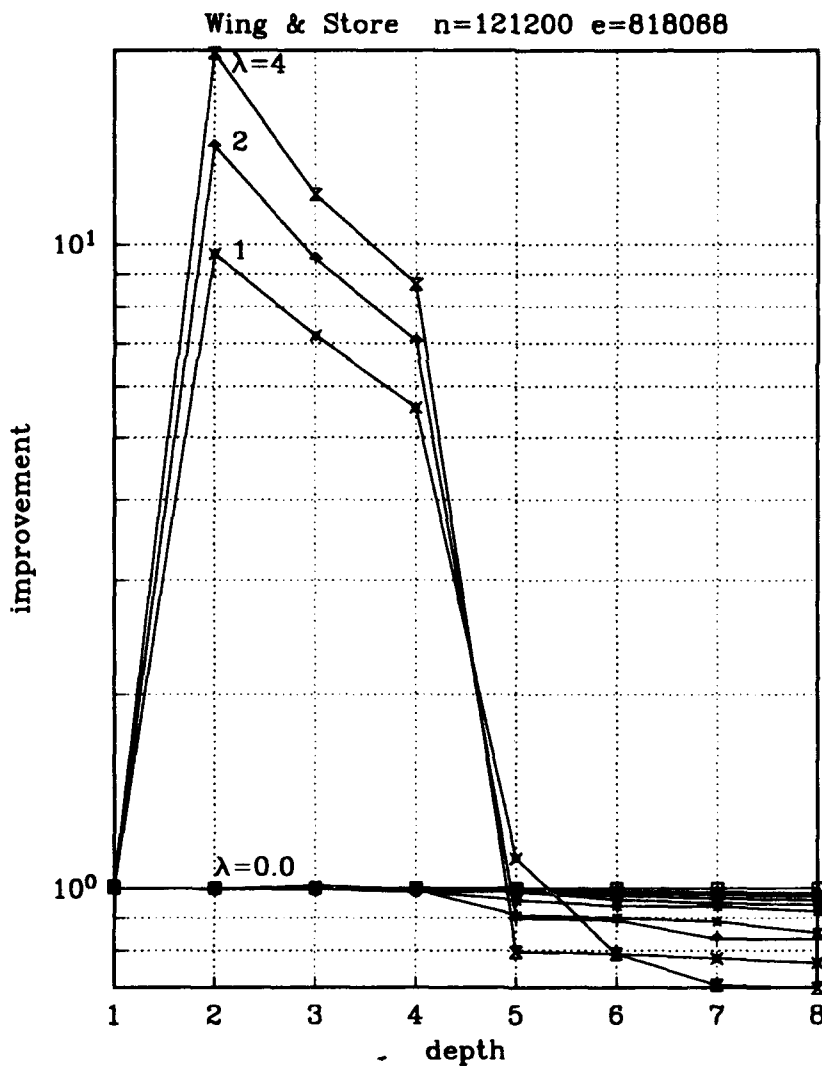
## 4 Wing and Store

This mesh was provided by Neil Frink and has 121200 nodes and 818068 edges. The algorithm was able to show good performance improvements for several values of depth for this mesh. The time required is 301 seconds for depth 15 (88 seconds input time) on a Sparcstation-10.



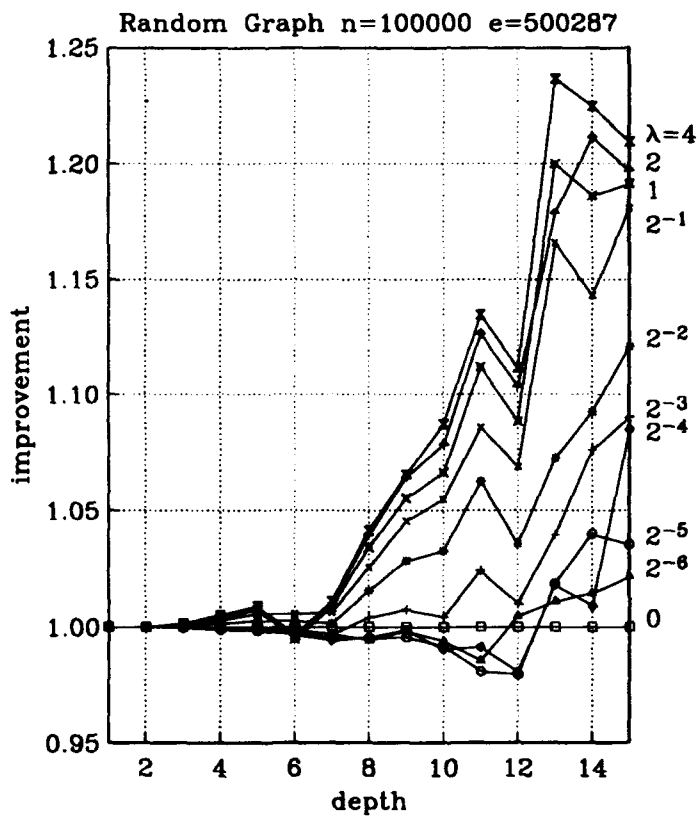
## 4.1 Wing and Store: modified algorithm

Careful examination of the plots presented so far will show that there is no performance improvement for depth 1. This is because the PBD implementation we have been using ignores edges for the first cut. This is important because, as explained in [4], taking edges into account for the first cut, usually yields very poor partitions. However, in some cases this is not true. For the Wing and Store mesh, taking edges into account during the first cut can yield very large performance improvements for  $\lambda \geq 1$ .

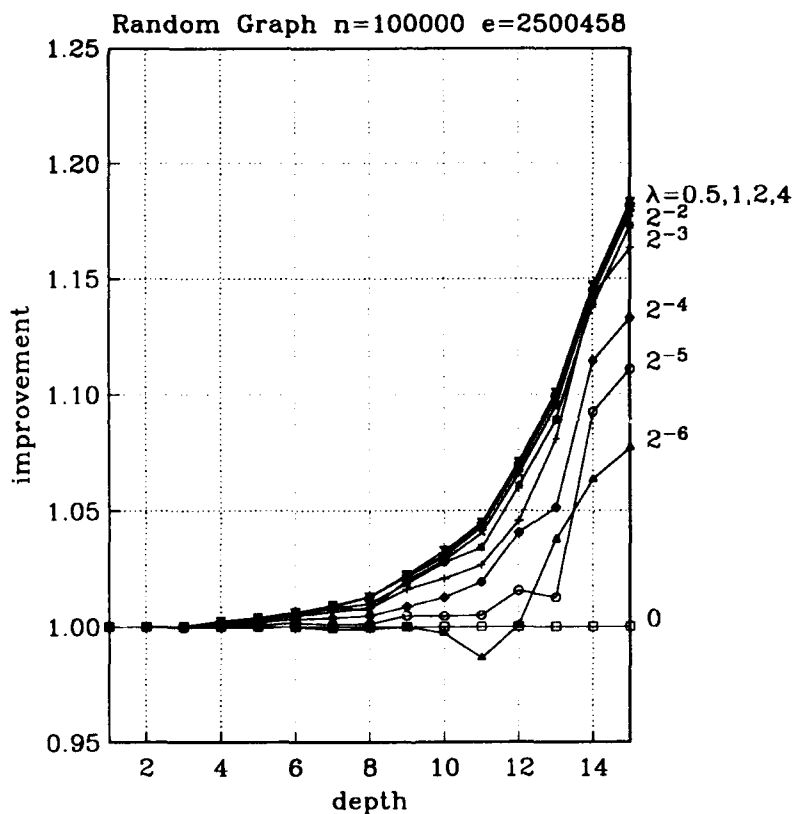


## 5 Random Graphs

Parametric Binary Dissection was also run on two randomly generated graphs. The first graph has 100000 nodes and 500287 edges. We obtain performance improvements at depths  $> 6$  for all but the lowest values of  $\lambda$ .



The second random graph has 100000 nodes and about 2.5 million edges. Except for  $\lambda = 2^{-6}$ , there is performance improvement for all values of lambda, though not as great as in the smaller random graph. The improvement saturates towards a smooth curve above  $\lambda = 0.5$ .



## 6 Conclusions

In this paper we have presented a brief and by no means exhaustive evaluation of Parametric Binary Dissection (PBD) on unstructured meshes. Our experimental results indicate that the performance of PBD is highly problem dependent but that it can often provide very good improvements over plain binary dissection. For meshes based on aircraft, the position of the wings can be a troublesome problem but can be overcome to some extent by randomly rotating the mesh points. Very good performance can sometimes be obtained by using a slightly modified variant of the algorithm, as explained in Section 4.1.

The PBD algorithm is very fast and it is feasible for the practitioner to try out several partitions to choose the best one for his or her application. It should also be recognized that it may be better to run the problem on a smaller number of processors, if a very good partition has been obtained for a depth lower than the maximum depth possible. For example, in Section 4.1, given a 32 processor system, and  $\lambda = 2$ , it would be preferable to use only 16 of these processors. This is because the depth 4 partition is 7 times faster than the depth 5 partition, while doubling the number of processors can at most halve the time.

### Acknowledgements

I am grateful to Dimitri Mavriplis, Clyde Gumbert and Neil Frink for providing me with large 3-d unstructured meshes for experimentation; to Tom Roberts, Mark Sanetrik and Tom Crockett for their assistance; and to Tom Zang and Yousuff Hussaini for their encouragement.

### References

- [1] Krishan Belkhale and Prithviraj Banerjee. Recursive partitions on multiprocessors. In *The Fifth Distributed Memory Computing Conference*, pages 930–938, April 1990.
- [2] Marsha J. Berger and Shahid H. Bokhari. A partitioning strategy for pdes across multiprocessors. *Proceedings of the 1985 International Conference on Parallel Processing*, pages 166–170, August 1985.

- [3] Marsha J. Berger and Shahid H. Bokhari. A partitioning strategy for non-uniform problems across multiprocessors. *IEEE Transactions on Computers*, C-36:570-580, May 1987.
- [4] Shahid H. Bokhari, Thomas W. Crockett, and David M. Nicol. Parametric binary dissection. Technical Report 93-39, ICASE, July 1993. NASA Contractor Report 191496.
- [5] Karen M. Dragon and John L. Gustafson. A low-cost hypercube load-balance algorithm. In *The Fifth Conference on Hypercubes, concurrent computers and applications*, pages 583-589, March 1989.
- [6] J. Saltz, S. Petiton H. Berryman, and A. Rifkin. Performance effects of irregular communications on massively parallel processors. *JPDC*, 13:202-212, 1991.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

<b>1. AGENCY USE ONLY (Leave blank)</b>	<b>2. REPORT DATE</b> October 1993	<b>3. REPORT TYPE AND DATES COVERED</b> Contractor Report	
<b>4. TITLE AND SUBTITLE</b> EXPERIENCE WITH PARAMETRIC BINARY DISSECTION		<b>5. FUNDING NUMBERS</b> C NAS1-19480 WU 505-90-52-01	
<b>6. AUTHOR(S)</b> Shahid H. Bokhari		<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b> ICASE Interim Report No. 25	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23681-0001		<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b> NASA CR-191552 ICASE Interim Report No. 25	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> National Aeronautics and Space Administration Langley Research Center Hampton, VA 23681-0001		<b>11. SUPPLEMENTARY NOTES</b> Langley Technical Monitor: Michael F. Card Final Report	
<b>12a. DISTRIBUTION/AVAILABILITY STATEMENT</b> Unclassified-Unlimited  Subject Category 63,66		<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (Maximum 200 words)</b> Parametric Binary Dissection (PBD) is a new algorithm that can be used for partitioning graphs embedded in 2- or 3-dimensional space. It partitions explicitly on the basis of $nodes + \lambda \times (edges\ cut)$ , where $\lambda$ is the ratio of time to communicate over an edge to the time to compute at a node. The new algorithm is faster than the original binary dissection algorithm and attempts to obtain better partitions than the older algorithm, which only takes nodes into account.  We compare the performance of parametric dissection with plain binary dissection on 3 large unstructured 3-d meshes obtained from computational fluid dynamics and on 2 random graphs. We show that the new algorithm can usually yield partitions that are substantially superior, but that its performance is heavily dependent on the input data.			
<b>14. SUBJECT TERMS</b> binary dissection; color image quantization; graph partitioning; load balancing; orthogonal recursive bisection; parametric binary dissection; partitioning; parallel processing		<b>15. NUMBER OF PAGES</b> 13	
		<b>16. PRICE CODE</b> A03	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b>	<b>20. LIMITATION OF ABSTRACT</b>

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18  
298-102