1995

WWW//PDEPACK A Web Based Problem Solving Environment for Partial Differential Equations

W. R. Dyksen

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
95-063

WWW//PDEPACK - A WEB BASED PROBLEM SOLVING ENVIRONMENT FOR PARTIAL DIFFERENTIAL EQUATIONS

Wayne R. Dyksen
Elias N. Houstis
John R. Rice

Department of Computer Science
Purdue University
West Lafayette, IN 47929

CSD TR-95-063
October 1995

WWW //PDEPACK

A Web Based Problem Solving Environment for Partial Differential Equations

W.R. Dyksen, E.N. Houstis, and J.R. Rice
Department of Computer Sciences
Purdue University

1 OVERVIEW

WWW //PDEPACK is a World Wide Web server that provides a sophisticated problem solving environment for partial differential equations (PDEs). It incorporates over 100 solvers of various types which cover all the common PDE applications in 2 and 3 dimensions. WWW //PDEPACK is a subset of *Parallel ELLPACK* in that not all the solvers are included and it is an extension of *Parallel ELLPACK* in that it allows for web based use. Parallel ELLPACK is an existing problem solving environment (see Section 3 for more details) that is now in alpha testing. A prototype of WWW //PDEPACK is now operational but its network performance is not yet satisfactory and its software generality is too limited (it now requires the user to have an X server).

The goal of this project is to implement the scenario illustrated in Figure 1. A user on the net accesses WWW //PDEPACK and the user interface is exported back. This interface is implemented in a net infrastructure code (e.g., Java from Sun) so it can run on any internet connected machine. This interface is large but still a tiny fraction of the entire WWW //PDEPACK system. The user formulates the PDE application using the interface and then it is solved at the server site. The server, in turn, accesses other servers (e.g., parallel machines) on the net during the solution process. The results are then returned to the user.

2 THE WWW //PDEPACK PROJECT

WWW //PDEPACK is a problem solving environment (PSE) for PDE based applications which is available for general use over the Net (World Wide Web). A user contacts http://pellpack.cs.purdue.edu for information, a demonstration, and to request an account. Once an account is established (this is part of the security system of WWW //PDEPACK), the user can then use WWW //PDEPACK.

The user interface of WWW //PDEPACK is exported to the user implemented using the Hot Java system. This approach provides two important features of WWW //PDEPACK. (1)

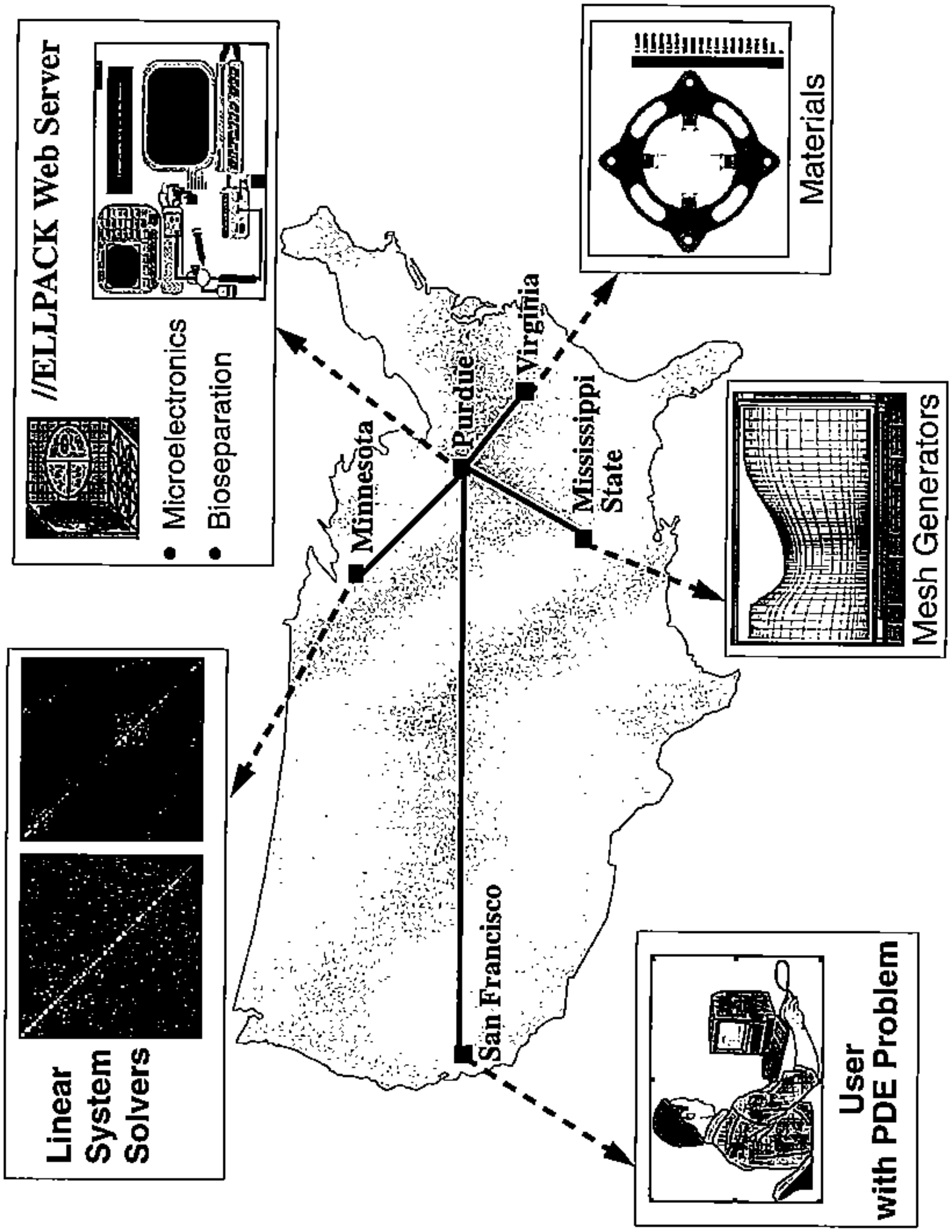

Figure 1: Schematic view of the WWW //PDEPACK system operating over the Net.

Generality – any machine connected to the internet can use the interface without concerns about language or machine compatibility. (2) *Interaction* – the user can specify the PDE with normal interaction speeds (as this is done locally) for the local machine. The amount of code exported to support the user interface is substantial (several megabytes) but it is only a tiny fraction of the WWW //PDEPACK system. If the user has no graphics capability then the text based interface tools must be used; these are less convenient but still practical to use.

As the PDE problem is being specified information is sent to the server. The server might request additional information but once the problem is completely specified it is solved on the server. The server is a common workstation (currently a Sun Sparc 5 with 64 Mbytes of memory) and it might use other machines on the internet to compute the solution. Once the PDE is solved the user can either view output generated by the server or request that the solution (normally a large data set) be sent for local use. This approach provides two more important benefits of WWW //PDEPACK. (3) *Access to High Performance Computers* – any user can access machines with sufficient power to solve the PDE problem. Even if the solution is too large to be sent to the user (or if there are no local visualization tools), the solution can be explored over the net. (4) *No Code Portability Problems* – the user does not need to have the code in the local language, the software infrastructure need operate only on the server. Even when other machines are used by the server, these are few in number and only specialized parts of WWW ///PDEPACK run on the other machines. The code that runs on an auxiliary server itself need only be for that machine.

There are several concerns and technical issues in creating WWW //PDEPACK which we discuss briefly:

- Performance of the user interface: There is a clear trade off in user interface performance between exporting code to the user's machine and executing code on the server. Our existing prototype shows that communicating each mouse click back to the server for processing provides unsatisfactory interactive performance due to network delays. Our analysis indicates that almost all of the interaction can be run locally by exporting a moderate amount of code. The user interface does use tools that are both time consuming to execute and which are too large to export. Examples are Maxima (used to transform mathematical equations) and domain processors (used to create meshes or grids in geometric domains). These tools usually require pauses in response even without a network and the added delay due to networks is unlikely to be significant.

- Security for the server: While we control somewhat the material received from a user, the server is clearly subject to attack. We place the server on a separate subnet and access licensed software through a gateway. Since we know exactly what is to be sent to whom via an RPC, it is possible to protect this licensed software. Even if a user succeeds in becoming "root", access to other machines is not possible. Of course, network file systems and similar tools are not used. Our process of "registering" users when we give them accounts provides us with a chance to screen users before providing them access to WWW //ELLPACK.

- Security for the user: This requires each user to be completely isolated from all others. Each user on the server runs in a virtual file system as "nobody" (a Unix term). Thus each user appears to have the entire machine (all the common software, including the operating systems, is replicated in the different virtual file systems) and thus the protection mechanisms between machines actually protects users from one another. This approach provides security at the cost of using much more memory than normally necessary.

- Software ownership and fair use: We prevent the copying of software by placing, if necessary, source code on another machine or another network and using secure RPC. The question of "fair use" is still murky as we may, in theory, allow thousands of people to use our single machine copy of a privately owned or commercial code. Most license agreements do not address this issue and we do not intend to do anything special.

- Payment for computing services: The WWW //PDEPACK server is provided free to users as well as time on associated servers used for security purposes. We do not foresee a need to charge users for time on these machines. If large numbers of users contend for service then they will be queued and the cost of the servers is clearly limited. However, there is a real problem when we access parallel machines which act as compute servers. Initially, WWW //PDEPACK uses local machines (a 64 processor Ncube2, a 140 processor Paragon, a 20 processor SP-2) and a user can easily pose a problem that uses 10s of hours on one of these machines. We intend to access off site machines in the future (e.g., Purdue is a member of Concurrent Super Computing Consortium and our group has access to its 540 processor Paragon). When the amount of use of these compute servers becomes a problem, we will require users to obtain accounts on them. This is a nuisance now but we believe the Net infrastructure will evolve soon to simplify such administrative problems.

There are three technical issues in implementing WWW //PDEPACK: First, the user interface must be clearly separated from the rest of the system. Our system is very modular in nature and we have already essentially completed this task. Second, we must create an efficient, exportable user interface. We have already made a prototype exportable user interface which is neither efficient nor general. It assumes the user has an X-windows server and it requires excessive network communication. We have studied the Java system recently released by Sun and believe we can use it to obtain both efficiency and generality on the network (e.g., Netscape has announced it will provide Java connections with its network browser). Even though Java is clearly a "first generation" system, it appears to be quite usable.

Third is the problem of dealing with the visualization of very large data sets over the network. With WWW //PDEPACK a person with a simple PC can generate a PDE solution consisting of millions of data points in 3-D. In our own group we have 155 Mbit/sec ATM networks and expensive graphics workstations visualize such solutions. We see two ways to provide visualization service to the user neither of which is always satisfactory. (1) We have visualization tools to slice, rotate, color, etc., data for viewing. We can send these images back over the Net. But the user might have a slow network connection or a black and white display; the viewing process would be painfully slow. (2) We can send the data set to the user. A two million point solution is not rare and its data set would be at least 25–50 Mbytes. The transmission time could be prohibitive if the user has slow network connections. The user might not have space to store the solution. The user might not have any visualization tools that can handle the data. We believe that visualization over the Net will be a severe problem for some users and it is one we cannot solve. We also believe that this is a common problem and that the Net infrastructure will provide solutions in a few years.

In summary, we have an operational prototype of WWW //PDEPACK and a plan providing a very useful and innovative network service using it. The implementation of the plan does not require new science or technology and it can be accomplished with reasonable cost and time.

3 PARALLEL PDEPACK SUMMARY

A. History.

This software system has its origins in the ELLPACK system [6, 7, 9] for solving second order, linear, elliptic PDEs. This system has been licensed to about 200 sites in over 20 countries.

A number of follow-up systems have been developed [8]: *Interactive ELLPACK* [2], *Parallel ELLPACK* [4], [5], and *PDELab* [1]. These have all been prototypes to develop the methodology of PSEs for PDE based applications and of handling parallelism in solving PDEs. Thus they have not been distributed for use outside Purdue University. The Parallel ELLPACK (//ELLPACK) system is being prepared [3] for wider distribution and is now in alpha testing at two sites outside Purdue.

B. Size and Description.

The //ELLPACK system consists of almost one million lines of code and thus it is impractical to describe it in detail. One may obtain a summary description on the Web with the URLs http://pellpack.cs.purdue.edu and http://www.cs.purdue.edu/research/cse/pellpack/pellpack.html. The first URL also indicates how WWW //PDEPACK is used over the network.

The names and sizes of the problem solvers in //ELLPACK is given in Table 1 along with a summary characteristics. The table is organized into three parts: (1) public domain software, (2) propriety software that will be included in WWW //PDEPACK for a trial period, (3) specialized or propriety software that will not be part of WWW //PDEPACK. This table does not show the 250,000 lines of C, Lisp, and Fortran code in the problem solving environment that are not part of a specific solver. The total lines of source code is about 900,000.

The solvers included in WWW //PDEPACK give broad coverage for partial differential equation problems, PDEs covered include

- Second order, linear elliptic problems on general domains in 2 or 3 dimensions,
- Second order evolutionary equations of general type with 2 or 3 space dimensions,
- Structural mechanics equations on general domains in 2 or 3 dimensions,
- Navier Stokes equations on general domains in 2 or 3 dimensions.

References

[1] Catlin, A., C. Chui, C. Crabill, E.N. Houstis, S. Markus, J.R. Rice, and S. Weerawarana, PDELab: An object-oriented framework for building problem solving environments for PDE

PDE PROBLEM SOLVERS

	Discretizer											Solver				
	Elliptic	Parabolic	Nonlinear	Systems of Equations	2-D Rectangular	2-D General	3-D Rectangular	3-D General	FEM	FD	Parallel	Direct	Iterative	Parallel Iterative	Parallel Direct	Size of Code Thousands of Lines
Public Domain																
NSPCG														X		34
SPARKIT														X		12
LINPACK													X			23
PDECOL	X	X	X	X							X					2
PDEONE	X	X	X	X							X					3
PDETWO	X	X	X	X							X					2
Private with Permission to Use in WWW //PDEPACK Server																
ELLPACK	X	X	X		X	X	X			X		X	X			100
//ELLPACK	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	43
//ITPACK															X	6
VECFEM	X	X	X	X	X	X	X	X	X				X			61
FIDISOL	X	X	X	X	X		X			X			X			30
CADSOL	X	X	X	X	X	X				X			X			142
MGGHAT	X				X	X			X				X			10
NSPCG													X			34
NPARC	X	X	X	X	X	X	X	X	X				X			38
CFD-TURBULENT	X	X	X	X	X	X			X				X			1
Commercial Products Licensed for Server																
VERSE		X	X	X	X				X				X			11
PATRAN	X	X	X	X	X	X	X	X	X				X			Unknown
MPPCGPACK														X		Unknown

Table 1: PDE problem solvers in Parallel ELLPACK showing their principal areas of application. The solvers are grouped according to availability for WWW //PDEPACK. The approximate number of lines of code is also given.

based applications, in *2nd Objet-Oriented Numerics Conf.*, (Vermeulen, ed.), Rogue Wave, Corvallis, OR (1994), 79–92.

[2] Dyksen, W.R. and C.J. Ribbens, Interactive ELLPACK: An interactive problem-solving environment for elliptic partial differential equations, *ACM Transactions on Mathematical Software*, **313**, (1987), 113–132.

[3] //**ELLPACK PSE**, Tech. Rpt., Department of Computer Sciences, Purdue University, (1996), 500 pages, in preparation.

[4] Houstis, E.N., J.R. Rice, and T.S. Papatheodorou, Parallel ELLPACK: An expert system for parallel processing of partial differential equations, *Math. Comp. Simulation*, **31** (1989), 497–508. Reprinted in *Intelligent Mathematical Software Systems*, (Houstis, Rice and Vichnevetsky, eds.), North Holland, Amsterdam (1990), 63–73.

[5] Houstis, E.N. and J.R. Rice, Parallel ELLPACK: A development and problem solving environment for high performance computing machines, in *Programming Environments for High-Level Scientific Problem Solving*, (Gaffney and Houstis, eds.), North-Holland, Amsterdam (1992), 229–241.

[6] Rice, J.R., ELLPACK: A cooperative effort for the study of numerical methods for elliptic partial differential equations, in *Proc. Army Numerical Analysis and Computers Conf.*, ARO Rpt. 77-3 (1977), 165–169.

[7] Rice, J.R., ELLPACK: Progress and plans, in *Elliptic Problem Solvers*, (M. Schultz, ed.) Academic Press (1981), 135–162.

[8] Rice, J.R., ELLPACK: An evolving problem solving environment, in *Problem Solving Environments for Scientific Computing* (Ford and Chatelin, eds), North-Holland, Amsterdam (1987), 233–245.

[9] Rice, J.R. and R.F. Boisvert, *Solving Elliptic Problems Using ELLPACK*, Springer-Verlag, New York (1985), 497 pages.

4 RELATED LITERATURE

This section lists publications related to the research and systems of //ELLPACK, PDELab and ELLPACK since 1990. This research and systems have been supported by the National Science Foundation in some way since the early 1980s. The most recent (and current) NSF support is by grants CCR 91-23502, *SoftLab – A Laboratory for Computational Science*, and CCR 92-02536, *Problem Solving Environments and Methods for the Development of PDE Based Applications on Parallel Machines.*

1. Byun, H. S.K. Kortesis, and E.N. Houstis, A workload partitioning strategy for PDE computations by a generalized neural network, *Journal on Neural, Parallel, and Scientific Computations*, 1, No. 2 (1993), pp. 209–226.

2. Byun, H. S.K. Kortesis, E.A. Vavalis, and E.N. Houstis, A virtual parallel environment for implementing neural network computations on parallel machines, *Journal on Neural, Parallel, and Scientific Computations*, 1, No. 3, (1993), pp. 301–323.

3. Catlin, A., C. Chui, C. Crabill, E.N. Houstis, S. Markus, J.R. Rice, and S. Weerawarana, PDELab: An object-oriented framework for building problem solving environments for PDE based applications, *2nd Object-Oriented Numerics Conf.*, (A. Vermeulen, ed.), RogueWare Software, Corvallis, OR (1994), 79–92.

4. Chen, A. and J.R. Rice, On grid refinement at point singularities for h-p methods. *Intl. J. Num. Meth. Engr.*, **33**, (1992), 39–57.

5. Chrisochoides, N., C.E. Houstis, E.N. Houstis, S.K. Kortesis, P. Papachiou, and J.R. Rice, Domain decomposer: A software tool for mapping PDE computations to parallel architectures, in *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations* (R. Glowinski, Y. Kuznetsov, G. Meurant, J. Periaux, and O. Widlund, eds.), SIAM Publications, (1991), 341–357.

6. Chrisochoides, N., E.N. Houstis, S.B. Kim, M.K. Samartzis, and J.R. Rice, Parallel iterative methods, in *Computer Methods for Partial Differential Equations VII* (R. Vichnevetsky, ed.), IMACS, New Brunswick, NJ (1992), 134–141.

7. Chrisochoides, N. and J.R. Rice, Partitioning heuristics for PDE computations based on parallel hardware and geometry characteristics, in *Computer Methods for Partial Differential Equations VII* (R. Vichnevetsky, ed.), IMACS, New Brunswick, NJ (1992), 127–133.

8. Chrisochoides, N., E.N. Houstis, and J.R. Rice, Mapping algorithms and software environments for data parallel PDE iterative solvers. *J. Par. Dist. Comp.*, **21**, (1994), 75–95.

9. Drashansky, T., S. Weerawarana, A. Joshi, R. Weerasinghe, E.N. Houstis, Software architecture of ubiquitous scientific computing environments for mobile platforms, CSD-TR-95-032, Department of Computer Sciences, Purdue University, 1995.

10. Gallopoulos, E., E.N. Houstis, and J.R. Rice, Computer as thinker/doer: Problem solving environments for computational science, *IEEE Comp. Sci. Engr.*, 1, (1994), 11–23.

11. Hadjidimos, A., E.N. Houstis, E.A. Vavalis, and J.R. Rice, Iterative line cubic spline collocation methods for elliptic partial differential equations in several dimensions. *SIAM J. Sci. Stat. Comp.* **14** (1993), 715–734.

12. Hadjidimos, A., E.N. Houstis, Y-L. Lai, and J.R. Rice, General interior hermite collocation methods for second order elliptic partial differential equations, *Applied Numerical Methods*, **16** (1994), 183–200.

13. Hadjidimos, A., E.N. Houstis, S.B. Kim, and J.R. Rice, The performance of parallel stationary iterative methods for distributed memory machines, *Proc. Intel Supercomputer User's Group* (D. Marinescu and R. Frost, eds) (1994), 169–173.

14. Hoffmann, C.M., E.N. Houstis, J.R. Rice, A.C. Catlin, M. Gaitatzes, S. Weerawarana, N-H.L. Wang, C.G. Takoudis, and D.G. Taylor, SoftLab – A virtual laboratory for computational science, *Math Comp. Simulation* **36** (1994), 479–491.

15. Houstis, C.E., E.N. Houstis, J.R. Rice, P. Varodoglou, and T.S. Papatheodorou, Athena: A knowledge base system for //ELLPACK, in *Symbolic-Numeric Data Analysis and Learning* (E. Diday and Y. Lechevallier, eds.), Nova Science, New York (1991), 459–467.

16. Houstis, C.E., E.N. Houstis, J.R. Rice, M.K. Samartzis, and D. Alexandrakis, The algorithm mapper: A system for modeling and evaluation parallel application/architecture pairs, in *Intelligent Mathematical software Systems* (Houstis, Rice and Vichnevetsky, eds.), North Holland, (1990), 87–101.

17. Houstis, E.N., J.R. Rice, N. Chrisochoides, H. Karathanasis, P. Papachiou, M.K. Samartzis, E.A. Vavalis, K. Wang, and W. Weerawarana, //ELLPACK: A numerical simulation programming environment for parallel MIMD machines, in *Supercomputing '90* (J. Sopka, ed.), ACM Press (1990), 96–107.

18. Houstis, E.N. and J.R. Rice, The engineering of modern interfaces for PDE solvers, in *Artificial Intelligence, Expert Systems and Symbolic Computing*, (E. Houstis and J. Rice, eds.), North-Holland, Amsterdam (1992), 89–94. Extended abstract in *Proc. IMACS World Congress*, IMACS, Rutgers University, New Brunswick, NJ, Vol. 3, (1991), 1037–1038.

19. Houstis, E.N. and J.R. Rice, Parallel ELLPACK: A development and problem solving environment for high performance computing machines, in *Programming Environments for High-Level Scientific Problem Solving* (P. Gaffney and E. Houstis, eds.), North-Holland, Amsterdam (1992), 229–241.

20. Houstis, E.N. and J.R. Rice, The architecture of PDE solving systems, in *Computer Methods for Partial Differential Equations VII* (R. Vichnevetsky, ed.), IMACS, New Brunswick, NJ (1992), 363–370.

21. Houstis, E.N., J.R. Rice, and S. Weerawarana, A software platform for integrating symbolic computation with a PDE solving environment, *Proc. 14th IMACS World Congress*, IMACS **1**, (1994), 482–485.

22. Houstis, E.N., J.R. Rice, and S. Weerawarana, An open structure for PDE solving systems, *Proc. 14th IMACS World Congress*, **3** (1994), 1296–1299.

23. Houstis, E.N., Parallel methodologies for "legacy" PDE software, *Proceedings of the Intel Supercomputer Users Group*, 1995, Albuquerque, to appear.

24. Kim, S., A. Hadjidimos, and E.N. Houstis, Domain decomposition method for scalar waves, *Domain Decomposition Conference*, October 1993.

25. Lai, Y.-L., A. Hadjidimos, E.N. Houstis, and J.R. Rice, On the iterative solution of Hermite collocation equations, *SIAM J. Matrix Anal. Appl.*, **16**, (1995), 254–277.

26. Markus, S., A.C. Catlin, S. Weerawarana, and E.N. Houstis, On the software engineering of multiplatform parallel/distributed software, *Proc. First Intl. Conf. on Neural, Parallel & Scientific Computation*, 1995, to appear.

27. McFaddin, A. and J.R. Rice, RELAX: A platform for software relaxation, in *Expert Systems for Scientific Computing*, (Houstis, Rice, and Vichnevetsky, eds.), North-Holland, Amsterdam (1992), 175–194.

28. McFaddin, H.S. and J.R. Rice, Collaborating PDE solvers. *Applied Numerical Mathematics*, **10**, (1992), 279–295.

29. Mu, M. and J.R. Rice, Performance of PDE sparse solvers on hypercubes, in *Unstructured Scientific Computations on Scalable Multiprocessors* (J. Saltz, ed.), MIT Press (1992), 345–370.

30. Mu, M. and J.R. Rice, Row oriented Gauss elimination on distributed memory multiprocessors. *Intl. J. High Speed Comp.* **4**, (1992), 143–68.

31. Mu, M. and J.R. Rice, A PDE sparse solver benchmark for massively parallel distributed memory multiprocessors, in *Computer Methods for Partial Differential Equations VII* (R. Vichnevetsky, ed.), IMACS, New Brunswick, NJ (1992), 546–552.

32. Mu, M. and J.R. Rice, An organization of sparse Gauss elimination for solving partial differential equations on distributed memory machines, *Numer. Meth. Part. Diff. Eqns.* **9** (1993), 175–189.

33. Mu, M. and J.R. Rice, Preconditioner construction with rational approximation, in *Parallel Processing for Scientific Computing, II* (Sinovec, et. al., eds.), SIAM Pub., Philadelphia (1993), 678–682.

34. Mu, M. and J.R. Rice, Preconditioning for domain decomposition through functional approximation, *SIAM J. Sci. Comp.* **15** (1994), 1452–1466.

35. Mu, M. and J.R. Rice, Modeling with collaborating PDE solvers – Theory and practice. *Contemporary Mathematics*, **180**, (1994), 427–438.

36. Sharma, R. and J.R. Rice, Numerical technique to solve nonlinear elliptic PDE's arising from semiconductor device modeling, in *Proc. Intl. Workshop on Computational Electronics*, Beckman Inst., Univ. Illinois (1992), 123–126.

37. Weerawarana, S. E.N. Houstis, and J.R. Rice, An interactive symbolic-numeric interface to parallel ELLPACK for building general PDE solvers, n *Symbolic and Numerical Computation for Artificial Intelligence*, (Donald, Kapur, and Mundy, eds.), Academic Press, (1992), 303–321.

38. Weerawarana, S., *Problem Solving Environments for Partial Differential Equation Based Systems*, Ph.D. Thesis, Department of Computer Sciences, Purdue University, 1994.

39. Wu, P., E.N. Houstis, and J.R. Rice, EPPOD: A parallel problem solving environment for the electronic prototyping of physical objects design, *Proc. DAGS '94 Symposium*, (F. Makedon, ed.), Dartmouth Inst. Adv. Grad. Studies, Dartmouth, NH (1994), 135–151.

40. Wu, P. and E.N. Houstis, A parallel mesh generation and decomposition methodology, *Proceedings of Mesh Generation Conference*, Albuquerque, Oct 1994.

41. Wu, P., Parallel shape optimization, *Proceedings of International Conference on Parallel Algorithms (ICPA'95)*, Wuhan-China, 1995, to appear.