

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

High-performance cluster computing, algorithms, implementations and
performance evaluation for computation-intensive applications
to promote complex scientific research on turbulent flows

by

Hao Wang

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Co-Majors: Computer Science; Water Resources

Major Professors: G.M. Prabhu and E.S. Takle

Iowa State University

Ames, Iowa

2001

Copyright © Hao Wang, 2001. All rights reserved.

UMI Number: 3016750

Copyright 2001 by
Wang, Hao

All rights reserved.

UMI[®]

UMI Microform 3016750

Copyright 2001 by Bell & Howell Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Graduate College

Iowa State University

This is to certify that the Doctoral dissertation of

Hao Wang

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

For the Co-major Program

Signature was redacted for privacy.

For the Co-major Program

Signature was redacted for privacy.

For the Graduate College

TABLE OF CONTENTS

LIST OF FIGURES	iv
LIST OF TABLES	vii
ACKNOWLEDGMENTS	ix
ABSTRACT	xi
CHAPTER 1. GENERAL INTRODUCTION	1
CHAPTER 2. SHELTERBELTS AND WINDBREAKS: MATHEMATICAL MODELING AND COMPUTER SIMULATIONS OF TURBULENT FLOWS	22
CHAPTER 3. IMPLEMENTATION AND PERFORMANCE EVALUATION OF PARALLEL COMPUTATIONS BY USING CLUSTER OF NETWORKED WORKSTATIONS	87
CHAPTER 4. DESIGN OF PARALLEL ALGORITHM AND PERFORMANCE MODELING & THEORETICAL ANALYSIS OF CLUSTER COMPUTING	104
CHAPTER 5. ON RELATIONSHIPS BETWEEN DRAG AND PRESSURE COEFFICIENTS FOR A POROUS SHELTERBELT	133
CHAPTER 6. MODELING THE EVAPOTRANSPIRATION AND ENERGY PARTITION OF INHOMOGENEOUS AGROECOSYSTEMS	170
CHAPTER 7. GENERAL CONCLUSIONS AND DISCUSSIONS	185

LIST OF FIGURES

CHAPTER 2

Figure 1a.	Flow pattern (unseparated flow) for medium-dense shelterbelt or windbreak	80
Figure 1b.	Flow pattern (separated flow) for dense shelterbelt or windbreak . .	80
Figure 2.	Wind speed within shelterbelts of different widths	81
Figure 3.	Horizontal profiles of wind speed around shelterbelts	82
Figure 4.	Wind direction rotation across shelterbelts	82
Figure 5.	Changes of shelter distance in oblique flow	83
Figure 6.	Perturbed pressure around shelterbelts of different shapes	83
Figure 7.	Wind reduction for shelterbelts of different densities	84
Figure 8.	Momentum budget for flow through shelterbelt	84
Figure 9.	Evapotranspiration as affected by shelterbelt	85
Figure 10.	Speed up ratio and its change with domain size and the number of processors in a cluster	86

CHAPTER 3.

Figure 1.	Speedup and its changes with domain size and number of processors	101
-----------	---	-----

CHAPTER 4.

Figure 1.	Theoretical and experimental parallel efficiencies and their changes with number of processors	125
-----------	--	-----

CHAPTER 5.

Figure 1.	Change of surface perturbed static pressure gradient, normalized by dynamic pressure (mean kinetic energy) of the undisturbed flow at the shelterbelt height H	158
Figure 2.	Schematic diagram of integration regions for momentum equation.	158
Figure 3.	Spatial distribution of the perturbed static pressure around a shelterbelt, normalized by dynamic pressure (mean kinetic energy) of the undisturbed flow at the shelterbelt height H	159
Figure 4.	Static pressure coefficients in the front of the shelter (C'_{pHF}) and behind the shelter (C'_{pHB}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).	159
Figure 5.	Comparison of drag coefficient (C_d) and static pressure-loss coefficient (C'_{pH}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).	161
Figure 6.	Comparison of drag coefficient (C_d) and total pressure-loss coefficient (C_{pH}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).	162
Figure 7.	Contributions of vertical advection, horizontal and vertical turbulent transports in partition of the drag for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c). Normalized by dynamic pressure (mean kinetic energy) of the undisturbed flow at the shelterbelt height H	164
Figure 8.	Perturbations of simulated turbulent stress for calculating drag by turbulent stress-loss method proposed by Seginer and Sagi (1971), normalized by turbulent stress of the upstream undisturbed flow.	165
Figure 9.	Simulated windspeed reduction profiles for calculating drag by turbulent stress-loss method proposed by Seginer and Sagi (1971), normalized by windspeed of the upstream undisturbed flow.	166

CHAPTER 6.

- Figure 1. Effects of medium-dense shelterbelts on latent and sensible heat fluxes (LE and H_s) for various soil moisture wetness(m). (a) $m=0.2$, (b)=0.3, (c) $m=0.5$, (d)=0.8. 175
- Figure 2. Influences of shelterbelt density on evapotranspiration (LE). (a) very dense shelterbelts and (b) very loose shelterbelts. 179

LIST OF TABLES

CHAPTER 1

Table 1. System disaster recovery efforts and their causes	6
--	---

CHAPTER 3.

Table 1. Computation profile for the shelterbelt with porosity of 50%.	90
Table 2a. Computation and communication profile on node 0.	95
Table 2b. Computation and communication profile on node 1.	95
Table 2c. Computation and communication profile on node 2.	96
Table 2d. Computation and communication profile on node 3.	96
Table 2e. Computation and communication profile on node 4.	97
Table 2f. Computation and communication profile on node 5.	97
Table 2g. Computation and communication profile on node 6.	98
Table 3. Comparisons of computed mean kinetic energy (MKE).	100
Table 4. Load balance ratio for domain size of $(256+2)*(128+2)$	102

CHAPTER 4.

Table 1. Computational load variability under different model parameter of porosity.	109
Table 2. Communication and idle time for functional decomposition.	114
Table 3. Idle time for functional decomposition.	122

CHAPTER 5.

Table 1. Mean departure of averaged static pressure-loss coefficient from drag coefficient.	167
Table 2. Mean departure of averaged total pressure-loss coefficient from drag coefficient.	167
Table 3. Drag coefficient determined by turbulent stress reduction method where turbulent stress calculated by the model directly from higher-order turbulence closure.	168
Table 4. Drag coefficient determined by turbulent stress reduction method where turbulent stress calculated indirectly from horizontal profiles of mean windspeed reduction.	169

ACKNOWLEDGMENTS

I would like to express my gratitude and appreciation to Professors G.M. Prabhu and E.S. Takle for their guidance, encouragement, and support during the course of this study, their time in revising the draft of my dissertation, as well as serving on my major professors. I also thank Professors S. Gadia, S. Tim, and W. Gutowski for their help and guidance as well as serving on my committee.

It is with great gratitude and appreciation that I was enlightened on computer science, database systems, scientific and industrial information systems, distributed and parallel systems and algorithms, distributed high-performance computing and programming, complex software system development, knowledge discovery and scientific discovery through high-performance computing by these professors. High-performance computing becomes a major approach, or sole approach in many cases, to understand the complex systems that classical theory and experiment could not solve; and such demands also promote the research on high-performance algorithms and computational methods as well as research on building complex and efficient software system to simulate the complicated natural and managed systems that we cannot touch before. I would like to thank Professor Takle for his guidance, helpful courses, and letting me work on the challenging new computer simulation model design and software system development and his financial support. I thank Professor Prabhu for his encouragement and his very helpful operating systems, computer architecture, and parallel programming courses that make my model software system become much more efficient and effective. I thank

Professor Gadia for his help and his advanced database system course that is also very helpful to me. I thank Professor Tim for his geographical information system (GIS) course and his help. I thank Professor Gutowski for his help and discussion. The success of my work is a result of the committee professors's kind help with their diversity expertise in these areas and their encouragement.

I have learned more than 140 credits in my graduate program at Iowa State University, I would like to extend my thanks to all of professors and graduate secretaries for their assistance.

ABSTRACT

Large-scale high-performance computing is a very rapidly growing field of research that plays a vital role in the advance of science, engineering, and modern industrial technology. Increasing sophistication in research has led to a need for bigger and faster computers or computer clusters, and high-performance computer systems are themselves stimulating the redevelopment of the methods of computation. Computing is fast becoming the most frequently used technique to explore new questions. We have developed high-performance computer simulation modeling software system on turbulent flows. Five papers are selected to present here from dozens of papers published in our efforts on complex software system development and knowledge discovery through computer simulations. The first paper describes the end-to-end computer simulation system development and simulation results that help understand the nature of complex shelterbelt turbulent flows. The second paper deals specifically with high-performance algorithm design and implementation in a cluster of computers. The third paper discusses the twelve design processes of parallel algorithms and software system as well as theoretical performance modeling and characterization of cluster computing. The fourth paper is about the computing framework of drag and pressure coefficients. The fifth paper is about simulated evapotranspiration and energy partition of inhomogeneous ecosystems. We discuss the end-to-end computer simulation system software development, distributed parallel computing performance modeling and system performance characterization. We design and compare several parallel implementations of our computer simulation system and show that the performance depends on algorithm design, communication channel pattern, and coding

strategies that significantly impact load balancing, speedup, and computing efficiency. For a given cluster communication characteristics and a given problem complexity, there is an optimal number of nodes. With this computer simulation system, we resolved many historically controversial issues and a lot of important problems.

CHAPTER 1. GENERAL INTRODUCTION

Introduction

Large-scale computing is a very rapidly growing field of research that plays a vital role in the advance of science, engineering, and modern industrial technology. Computing is fast becoming the most frequently used technique to explore new questions. Fast computers have stimulated the rapid growth of a new way of doing science. The two broad classical branches of *theoretical science* and *experimental science* have been joined by *computational science*. Computational scientists simulate on supercomputers phenomena too complex to be reliably predicted by theory and too dangerous or expensive to be reproduced in the laboratory (Anderson et al. 1998; Baker and Smith 1996; Barrios, et al. 1999; Bauer 1992; the Beowulf team 1996; Foster 1998, 2000; Lewis and El-Rewini 1992; Martin 1988).

Increasing sophistication in research has led to a need for bigger and faster computers. In this quest, high-performance computer systems are themselves stimulating the redevelopment of the methods of computation. Results in one area are quickly adapted for another. The effect is making *high-performance computing a multi-disciplinary adventure*. Research scientists in high-performance computing come from a variety of interests and background and have a large overlap between areas of computer science and mathematics and areas of engineering and physical sciences (Buyya 1999; Foster 1998, 2000).

Dissertation Organization

We choose 5 papers from a pool of dozens of papers for our efforts on building a high-performance computer simulation software system and the simulations of complex shelterbelt turbulent flows. For all the papers, I am the first author, responsible for all the processes from end-to-end of the research including software requirement analyses, model design, software specification, algorithm design, coding and implementation, functional and performance tests, integration and coupling, experiment design and execution, results analysis and writing.

The dissertation is organized in the threads of the complex model software system development and scientific discoveries through computer simulations. In Chapter 1, the general introduction is presented. Chapter 2 describes the end-to-end simulation system development and simulation results that help understand the nature of complex shelterbelt turbulent flows. Chapter 3 deals specifically with high-performance algorithm design and implementation in the cluster of computers. Chapter 4 discusses the design process of parallel algorithms and software system and the theoretical performance modeling of cluster computing. Chapter 5 is the framework of drag and pressure coefficients for a porous shelterbelt. Chapter 6 is about simulated evapotranspiration and energy partition of inhomogeneous agroecosystems. Finally, a general conclusion is given in Chapter 8. We not only present the detailed results and in-depth analysis, but also cover all steps and all aspects of scientific research through high-performance computing: from physical model to mathematical model, from mathematical model to computer model, computation algorithms, parallel programming, software implementation, performance evaluation and improvement, experiment design and execution, results analysis and interpretation.

Literature Review

The computing industry is one of the fastest growing industries and it is fueled by the rapid technological developments in the areas of computer hardware and software. The main reason for creating and using parallel computers is that parallelism is one of the best ways to overcome the speed bottleneck of a single processor. Higher-performance computing becomes an indispensable tool in the past few years for climate research and weather forecasting because the non-linear nature of atmosphere puts a lid on the theoretical research and also because the control experimental research is difficult to be done on the vast-volume and ever-changing atmosphere. In just the last few years, the inclusion of computer modeling has produced results that were inconceivable a few decades ago. Computer simulations are less expensive, can address a wider range of problems, and also provide an understanding of physical problems that cannot be obtained from experiments alone (Drake et al. 1993, 1995; Foster and Michalakes 1993; Foster and Toonen 1994; Foster and Worley 1993; Foster et al. 1992; Prusinkienicz 2000; Semtner 2000).

The development of faster computer is driven not only by simulations of complex systems such as ecosystems, weather, climate, turbulence, and mechanical processes, but also by the emerging commercial applications that require a computer to be able to process large amounts of data in sophisticated ways. High-performance computing is a common requirement for many applications.

There are three ways to do anything faster (Pfister 1998): (1) work harder – improve the speed of processor, (2) work smarter – improve the algorithms; (3) work together – do parallel processing. Parallel computing is an effective and efficient solution to ever-increasing demand

of computing power. Parallel processing, the method of having many small tasks solve one large problem, has emerged as a key technology in modern computing. The acceptance has been facilitated by two major developments: massively parallel processors (MPPs) and the widespread use of distributed cluster computing. These powerful parallel computing approaches have been developed and used to solve computational grand challenge problems such as climate modeling and drug design.

On one hand, the clock cycle times that determine the speed of processor are close to their physical limits due to the speed of light, and design of specialized processor is very expensive. On the other hand, high-performance microprocessor and high-speed interconnection network have been produced in large volume. In the past decade, the price/performance ratio of microcomputers has greatly improved. Therefore, distributed parallel computing or cluster computing is especially attractive (Boden et al. 1995; Farazdel et al. 1999; Mache 1999) .

Cluster is a collection of interconnected computers working together as a single system. It has a single system image spanning all its nodes. The initial idea leading to cluster computing was developed in the 1960s by IBM as a way of linking large mainframes (e.g. IBM HASP, JES, Parallel Sysplex). However, cluster computing did not gain momentum until the 1990s. Sometimes a number of conditions come to a confluence and allow the emergence of a new way of doing things: In 1994, NASA Earth and Space Science scientists first investigated the potentials of PC clusters for performing computational task (Becker et al 1995; Ridge et al. 1997; Sterling et al. 1995). Beowulf used a Pile of PCs (PoPC) to describe a loose ensemble or cluster of PCs and achieved the best overall cost/performance ratio for the cluster. This

successful initiative encouraged many others to establish the similar clusters of computers throughout the world. In the meantime, Berkeley group investigated the potentials of network of workstation (NOW) (Anderson et al. 1995). Although the history of cluster computing is short, its achievements are large and has significant impacts on the behavior and methods of computing. This represents the convergence of four trends: high-performance microprocessors, high-speed networks, ever-increasing needs of computing power for computational science and commercial applications, and standard tools for high-performance distributed computing.

In the past, scientific applications have been run on special purpose supercomputing hardware because they required not only fast processing, but also large memories and secondary storage capacities. But now the trend in parallel computing is to move away from specialized traditional supercomputers like Cray/SGI T3E to cheaper and general purpose systems consisting of loosely coupled components built up from commodity hardware like PCs and workstations. Cluster systems are the supercomputers of tomorrow. However, the determining performance bottle neck of clusters compared to MPP systems lies in its interconnection network and supporting software tools.

Beowulf clusters were motivated by the needs of the scientific community. The initial requirements of that community have been met, but some challenges still remain. First, clusters are not particularly easy to use, nor is it clear how to scale them to very large sizes. In addition, new classes of data-intensive applications outside of scientific computation are being explored as we start to take the next steps in Beowulf-class computing. Today's Beowulfs, while far more capable in every dimension, owe their heritage to this first generation of system

technologies and those computational scientists who first applied them to real-world problems. Load balancing is another way to ensure performance by distributing the work load or network traffic among the available nodes in the cluster. The distributed systems carry the load of high speed computations for scientific studies or industry workload on a corporation network or even today's typical desktop computing applications.

Table 1. System disaster recovery efforts and their causes

Causes	Services	Percentage
Hurricanes	96	50.8%
Hardware	50	20.6%
Power	23	10.6%
Earthquake	17	9.0%
Flood	7	3.2%
Software	5	2.6%
Fire	2	1.1%
Virus	1	0.5%
Storm	1	0.5%
Civil disorder	1	0.5%
Chemical spill	1	0.5%

While performance, scalability, and throughput continue to be major reasons for using clusters, recently clusters are also used to enhance system reliability and availability. Table 1 shows the disasters that IBM services involved to recover systems (IBM 2000). The importance of availability and reliability increases as Internet becomes a viable social force and major communication and transaction tool. Cluster computing provides a way to the automatic recovery of system that detects faults in hardware, network and software and handles these faults in a grace way (automatic failover) to minimize the disruption of services (Lee 1999; Libertone 2000; Marcus and Stern 2000; Pfister 1998). The increasing need for processing power for scientific computing combined with the large development of distributed information systems makes the computation distribution over workstation networks very attractive. Fault tolerance becomes an essential component. Many hours of computation can be lost not only if a hardware failure occurs, but also if one of the processors is rebooted, turned off or disconnected from the network. As a consequence, developing parallel programs also requires expertise in fault tolerance.

Cluster computing provides expandability. It provides an easy way to upgrade the cluster system by adding more nodes to increase computing power without incurring a lot of extra expense.

In the past decade, there has been dramatic shift from mainframe or host-centric computing to distributed or network-centric computing. Cluster computing is a rapidly maturing technology that seems certain to play an important part in the network-centric distributed computing future. Since NASA Earth and Space scientists first successfully explored the potential of cluster computing in scientific research, numerous institutions and

companies are investigating cluster computing and its various applications. Just as the most powerful supercomputer were developed for, first used and characterized in complex scientific research like climate simulation, we systematically explore the potentials of cluster computing by developing the end-to-end computer simulation system that addresses a complicated scientific research problem on turbulent flows.

Climate modeling is so computation-demanding that several parallel computing books written by computer scientists took climate simulation for illustrating concepts. For example, Foster (1995) demonstrated that refinements to climate model would increase the computational requirements by 10,000 times. On recent issues of the *Communications of the ACM*, Semtner (2000) discussed climate simulations and Prusinkiewicz (2000) discussed ecosystem modeling. Since ecosystems are related to climate, and ecosystems also exert impact on climate; moreover, human being's practices change both. If we combine both, we will demand not only more computer power, but also better computer algorithms and software infrastructure. Recently CCGRID development will provide promising future (Foster, 2000).

For drug discovery, many computer scientists and biological scientists have been doing very detailed research—down to gene-level; for the earth's health and sustainability, computer scientists will develop fastest computer systems and best simulation model systems to discover the man-environment relationships.

Department of Energy (DOE) Climate Prediction Program, NSF, and National Center for Atmospheric Research (NCAR) together with Los Alamos National Laboratories and Naval Postgraduate School used Parallel Ocean Model, Parallel Sea Ice Model, NCAR Parallel Community Climate Model version3 (PCCM3), and Flux Coupler to create Parallel Climate

Model (PCM) for MPP computers. PCM not only has very coarse resolution (>0.5 degree), but also lacks a lot of ecosystem and physics details (<http://www.cdg.ucar.edu>, 1997). Recently DOE laboratories (Argonne National Laboratory: Ian Foster and Jay Larson, Lawrence Berkeley National Laboratory: Chris Ding, Oak Ridge National Laboratories: John Drake, Los Alamos National Laboratories: Phil Jones, Lawrence Livermore National Laboratory: Doug Rotman) proposed to develop a modular, performance portable climate system model CSM (proposal to DOE Office of Science). Their two primary goals are (1) a performance-enhanced CSM-2, able to exploit microprocessor-based parallel computers; and (2) a design for current and future CSM versions that improves substantially over current practice. They pointed out that a substantial challenge in both areas will be to evolve software engineering practices.

Argonne National Laboratory (Ian Foster and John Michalakes, <http://www-climate.mcs.anl.gov/proj>) spent a lot of efforts to collaborate with NCAR to parallelize the mesoscale model version 4 (MM4) and later version 5 (MM5) on MPP computers. They decomposed MM4 in a single horizontal dimension and MM5 in both horizontal dimensions (Foster et al. 1992; Foster and Michalakes 1993). The parallelized MM4/5 can let investigators learn more details (resolution of up to 10-30km) of global climate change on the particular region than the PCM/CSM can provide (Argonne Regional Climate Center, <http://www-climate.mcs.anl.gov/proj/climate>).

The most useful and practical information comes from very-high resolution climate modeling and simulations that resolves all of interactions of climate, ecosystem, human being, and others. People not only can use this very high resolution model to understand the nature, but also can use this very high resolution model to provide the useful guide for practices to

improve climate and enhance productivity. Geographical Information System (or GIS) provides a powerful tool to get very high resolution data. Very high-resolution model is needed to use such data to investigate climate and environmental changes in details. However, the development of such very high resolution model demands very fast computing. Such model also should have a resolution high enough to be used in engineering.

Although Beowulf-class cluster have been used for scientific computing, scientific computing can generate enormous amounts of data as output, and also require similar amounts as input. It has been well-demonstrated that beowulf clusters are up to the task of running data-intensive programs. In light of these successes with scientific computation, many people have realized that they can apply these systems to other data processing. Beowulf clusters have been used as high-volume Web servers and high throughput data servers.

References

- Al-Tawil, K., 2001: Performance modeling and evaluation of MPI. *Journal of Parallel and Distributed Computing*, vol. 61, 202-223, 2001.
- Alverson, G.A., W.G. Griswold, C. Lin, D. Notkin, and L. Snyder, 1998: Abstractions for portable, scalable parallel programming. *IEEE Transactions on Parallel and Distributed Systems*, vol.9(1), 1998.
- Andersson, S. et al, 1998: *RS/6000 Scientific and Technical Computing: POWER3 Introduction and Tuning Guide*, IBM Redbook, SG24-5155-00, 1998, pp1-205.
- Anderson, T.E., D.E. Culler, D.A. Patterson, et al., 1995: A case for Now (Network of Workstations). *IEEE Micro*, 15:54-64, Feb. 1995.

Avresky, D.R., 1992: *Hardware and Software Fault Tolerance in Parallel Computing Systems*.

Ellis Horwood publisher, Chichester, West Sussex, England, 1992, pp334.

Baker, L. and B.J. Smith, 1996: *Parallel Programming*. New York, NY: McGraw-Hill.

Pp.381.

Barnard, S., R. Biswas, S. Saini, R. V. Wijngaart, M. Yarrow, L. Zechter, I. Foster, O. Larsson,

2000: *Large-scale distributed computational fluid dynamics on the information power grid using Globus*.

Barnard, S.T. and H.D. Simon, 1996: Fast multilevel implementation of recursive spectral

bisection for partitioning unstructured problems. *Concurrency: Practice and Experience*, vol.6(2), pp101-117.

Barrios, et al, 1999: *Scientific Applications in RS/6000 SP Environments*. IBM Redbook,

SG24-5611-00, 1999, pp.1-231.

Bauer, B.E., 1992: *Practical Parallel Programming*. New York: Academic Press. Pp.455.

Becker, D.J., T. Sterling, D. Savarese, E. Dorband, U.A. Ranawake, and C.V. Packe, 1995:

BEOWULD: A parallel workstation for scientific computation. *Proc. 1995 Int'l Conf. Parallel Processing*, pp. 11-14, 1995.

Becker, D.J., T. Sterling, D. Savarese, B. Fryxell, and K. Olson, 1995: Communication

overhead for space science applications on the Beowulf parallel workstation.

Proceedings, High Performance and Distributed Computing.

Beguelin A., E. Seligman, and P. Stephan, 1997: Application level fault tolerance in

heterogeneous networks of workstations. *J. Parallel and Distributed Computing*, vol.

43, no.2, pp.147-155, 1997.

- Beowulf team, 1996: *The Beowulf Project*, <http://www.beowulf.org>
- Berkeley Group, 1998: *The Case for Now*. <http://now.cs.berkeley.edu/Case/now.html>
- Berners-Lee, T. et al, 1994. The World-Wide Web. *Communications of the ACM*, vol.37(8), 76-82, 1994.
- Bhatt, S.N., G. Pucci, A. Ranade, and A.L. Rosenberg, 1993: Scattering and gathering messages in networks of processors. *IEEE Trans. On Computers*, C-42, 1993, pp.938-949.
- Boden, N.J., D. Cohn, R.E. Felderman, A.E. Kulawik, C.L. Seitz, J.N. Seizovic, and W.K. Su: Myrinet, 1995: A gigabit per second local area network. *IEEE Micro*, vol.15, no.1, pp.29-36, 1995.
- Bohossian, V., C. Fan, P.S. LeMahieu, M. D. Riedel, L. Xu, J. Bruck, 2001: Computing in the RAIN: A Reliable Array of Independent Nodes. *IEEE Transactions on Parallel and Distributed Systems*, vol.12, No.2, 2001.
- Bruck, J., R. Cypher, and C.T. Ho, 1993: Fault-tolerant meshes and hypercubes with minimal numbers of spares. *IEEE Trans. Computers*, vol.42, no.9, pp.1089-1104, Sep. 1993.
- Bubak, M., J. Dongarra, and J. Wasniewski (eds.), 1997: *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Lecture Notes in Computer Science, No.1332, 1-515.
- Buyya, R., 1999: *High-Performance Cluster Computing*, Prentice Hall, NJ: Upper Saddle River, 1999, pp840.
- Chandra, T.D. and S. Toueg, 1996: Unreliable failure detectors for reliable distributed systems. *J. ACM*, vol.43, no.2, pp.225-267, 1996.
- Chen, P.M., E.K. Lee, G.A. Gibson, R.H. Katz, and D.A. Patterson, 1994: Raid – high-

- performance, reliable secondary storage. *ACM Computing Surveys*, vol.26, no.2, pp.145-185, 1994.
- Colajanni, M., P.S. Yu, and D. M. Dias, 1989: Analysis of task assignment policies in scalable distributed web-server systems. *IEEE Transactions on Parallel and Distributed Systems*, vol.9(6), pp.585-600, 1989.
- Drake J., I. Foster, J. Michalakes, B. Toonen, and P. Worley, 1995: Design and performance of a scalable Parallel Community Climate Model. *Parallel Computing*, 21(10): 1571-1591, 1995.
- Drake, J.B., R.E. Flanery, I.T. Foster, J.J. Hack, J.G. Michalakes, R.L. Stevens, D.W. Walker, D. L. Williamson, and P. H. Worley, 1993: The message-passing version of the parallel community climate model. In *Parallel Supercomputing in Atmospheric Science*, eds. G-R. Hoffman and T. Kauranne, Singapore: World Scientific, 500-513.
- Drake, J.B., I.T. Foster, J.G. Michalakes, and P.H. Worley, 1995: Parallel algorithms for semi-Lagrangian transport in global atmospheric circulation models. In *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, eds. D.H. Bailey, P.E. Bjorstad, J.R. Gilbert, M.V. Mascagni, R.S. Schreiber, H.D. Simon, V.J. Torczon, and L.T. Watson, *SIAM*, 119-124.
- Dutt, S. and J.P. Hayes, 1992: Some practical issues in the design of fault-tolerant multiprocessors. *IEEE Trans. Computers*, vol.41, no.5, pp.588-598, 1992.
- Farazdel, et al, 1999: *Understanding and Using the SP Switch*. IBM Redbook SG24-5161-00, pp1-210, 1999.
- Archil, F., J.Ramanujam, and P. Sadayappan, 1990: Task allocation onto a hypercube by

- recursive mincut bipartitioning. *J. Parallel and distributed computing*, vol.10, pp.35-44, 1990.
- Foster, Ian, 1995: *Designing and Building Parallel Programs*. Addison-Wesley, 1995
- Foster, Ian, 1998: *The Grid: Blueprint for a new computing infrastructure*. Morgan-Kaufmann, 1998.
- Foster, Ian, 2000: Internet Computing and the Emerging Grid. *Nature*, Dec.7, 2000.
- Foster, I. And B. Toonen, 1994: Load balancing in climate model. *IEEE Proc. Scalable High-Performance Computing Conf.*, 1994.
- Foster, I. And J. Michalakes: Massively parallel implementation of the Penn State/NCAR Mesoscale Model. *Proc. 9th Intl Conf. On Interactive Information and Processing Systems for Meteorology*, SIAM, 1993.
- Foster, I. And J. Michalakes, 1993: MPMM: A massively parallel mesoscale model. *Parallel Supercomputing in the Atmospheric Sciences*, 354-363, World Scientific, 1993.
- Foster, I., W. Gropp, and R. Stevens, 1992: The scalability of numerical methods for climate modeling. *Proc. 5th SIAM Conf. On Parallel Processing*, 307-312, IEEE, 1992.
- Foster, I.T. and B. Toonen, 1994: Load balancing algorithms for climate models. In *Proceedings of the Scalable High Performance Computing Conference*, eds. J.J. Dongarra and D. W. Walker, IEEE Computer Society Press, 674-681.
- Foster, I.T. and P.H. Worley, 1993: Parallelizing the spectral transform method: a comparison of alternative parallel algorithms. In *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, eds. R.F. Sincovec, D.E. Keyes, M.R. Leuze, L.R. Petzold, and D.A. Reed, *SIAM*, 100-107.

- Garey, M.R. and D.S. Johnson, 1979: *Computers and Intractability, A Guide to Theory of NP-Completeness*. San Francisco: Freeman, 1979.
- Geist, A., A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, 1994: *PVM: Parallel Virtual Machine*. The MIT Press.
- Geist, A., J.A. Kohl, P.M. Papadopoulos, 1996: *PVM and MPI: a Comparison of Features*.
- Gilbert J.R. and E. Zmijewski, 1987: A parallel graph partitioning algorithm for a message-passing multiprocessor. *Int'l J. Parallel Programming*, vol.16(6), pp.427-449, 1987.
- Glenn R. Luecke, B. Raffin, and J. Coyle, 2000: Comparing the communication performance and scalability of a linux and a NT cluster of PCs, a Cray Origin 2000, an IBM SP, and a Cray T3E-600. *2000 IEEE Conference on Parallel Computing*, pp.26-35, 2000.
- Gropp, W., E. Lusk, N. Doss, and A. Skjellum, 1996: A high-performance , portable implementation of the MPI message passing interface standard. *Parallel Computing*, vol.22, No.6, pp.789-828, September 1996.
- Gropp, W., 1997: *Tutorial on MPI: The Message-Passing Interface*. Argonne National Laboratory.
- Gropp W. and E. Lusk, 1996: User's Guide for mpich, a Portable Implementation of MPI. Argonne National Laboratory, ANL/MCS-TM-ANL-96/6.
- Gropp, W., E. Lusk, and A. Skjellum, 1994: *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. Cambridge, MA: The MIT Press.
- Gustafson, J.L., 1988: Reevaluating Amdahl's law. *Communications of the ACM*, vol.31, no.5, pp.532-533, 1988.
- Hasegawa, T., H. Takagi, and Y. Takahashi, 1989: *Performance of distributed and parallel*

- systems*, New York: North-Holland, 1989.
- Hamidzadeh, B., L.Y. Kit, and D.J. Lilia, 2000: Dynamic task scheduling using online optimization. *IEEE Transactions on Parallel and Distributed Systems*, vol.11(11), 2000.
- Hammond, S.W., R.D. Loft, P.D. Tannenbaun, 2000: *Architecture and application: the performance of the NEC SX-4 on the NCAR Benchmark Suite*. NCAR Report, 2000.
- Hui, C.C. and S.T. Chanson, 1999: Hydrodynamic load balancing. *IEEE Transactions on Parallel and Distributed Systems*, vol.10(11), 1999.
- IBM 2000: SAP R/3 on DB2 for OS/390 Disaster recovery.
- IEEE Computer Society, 1996: *Proceedings of the Second MPI Developer's Conference*.
- Kafil, M. and I. Ahmad, 1998: Optimal task assignment in heterogeneous distributed computing systems. *IEEE Concurrency*, vol. 6, No.3, 1998, pp.42-51.
- Kothari, S., 1996: *Parallelization Agent for Legacy Codes*. Draft Technical Report, Iowa State University, Ames, Iowa.
- Krishna, C.M., 1996: *Performance Modeling for Computer Architects*. IEEE Computer Society Press, CA: Los Alamitos, 1996.
- Lee, Richard R, 1999.. *Windows NT Microsoft Cluster Server*. CA: Berkeley, Osborne/McGraw-Hill, 1999, pp.341.
- Lewis, T.G. and H. El-Rewini, 1992: *Introduction to Parallel Computing*. New York: Prentice Hall, 1992.
- Liao, C.J. and Y.C. Chung, 1999: Tree-based parallel load-balancing methods for solution-adaptive finite element graphs on distributed memory multicomputers. *IEEE*

- Transactions on Parallel and Distributed Systems*, vol.10 (4), 1999.
- Libertone D., 2000: *Windows 2000 Cluster Guidebook*. NJ: Upper Saddle River, Prentice-Hall, 2000, pp280.
- Mache.J, 1999.: An assessment of gigabit ethernet as cluster interconnect. *1999 IEEE Conference on Cluster Computing*, 1999,36-42.
- Marcus, E. and H. Stern, 2000: *Blueprints for High Availability*. Wiley Computer Publishing, 2000.
- Martin, J.L., 1988: *Performance Evaluation of Supercomputers*. Elsevier Science Publishers, Amsterdam, the Netherlands, 1988, pp419.
- Michalakes, J., 1991: *Analysis of Workload and Load Balancing Issues in the NCAR Community Climate Model*. ANL Technical Report ANL/MCS-TM-144.
- Minnich, R., M. Gokhale, A. Marks, J. Kaba, and J. Degood, 1998: Freeware for cluster computing", *USENIX 1998 Annual Technical Conference*.
- Mivakant, S., L. Fei, X. Lin, G. Xing, 2000: On group communication support in CORBA. *IEEE Transactions on Parallel and Distributed Systems*.
- Moon, S., B. So, and M.W. Hall, 2000: Evaluating automatic parallelization in SUIF. *IEEE Transactions on Parallel and Distributed Systems*, vol.11(1), 2000.
- Orduna, J.M., V. Arnau, J. Duato, 2000: Characterization of communications between processes in message-passing applications. *2000 Conference on Parallel Computing*, 91-98.
- Pacheco, P.S., 1997: *Parallel Programming with MPI*. San Francisco, CA: Morgan Kaufmann Publishers, Pp.418.

- Patel, N. and H. Sivaraman, 2000: A model of completion queue mechanisms using the virtual interface API. *2000 IEEE Conference on Parallel Computing*, pp.280-288.
- Pfister, G.F., 1998: *In Search of Clusters*. NJ: Upper Saddle River, Prentice-Hall, , 2nd edition, 1998, pp.516.
- Prieto, M., I.M. Liorente, and F. Tirado, 2000: Data locality exploitation in the decomposition of regular domain problems. *IEEE Transactions on Parallel and Distributed Systems*, vol.11(11), 1141-1150, 2000.
- Pritchard, D. and J. Reeve (Eds.), 1998: *Euro-Par'98 Parallel Processing*. Lecture Notes in Computer Science, No.1470, 1-1152.
- Prusinkienicz, P., 2000: Simulation modeling of plants and ecosystems. *Communications of the ACM*, pp.84-93, 43(7), July, 2000.
- Reschke, C., T. Sterling, D. Ridge, D. Savarese, D. Becker, and P. Merkey, 1996: A design study of alternative network topologies for the Beowulf parallel workstation. *Proceedings, High Performance and Distributed Computing*, 1996.
- Ridge, D., D. Becker, P. Merkey, T.S. Becker, and P. Merkey, 1997: Beowulf: harnessing the power of parallelism in a Pile-of-PCs. *Proceedings, IEEE Aerospace*.
- Semtner, A., 2000: Ocean and atmosphere modeling. *Communications of the ACM*, 43(4), pp.80-89, April, 2000.
- Snir, M., S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, 1996: MPI: *The Complete Reference*. Cambridge, MA.: The MIT Press. Pp.336.
- Sohn, J., T.G. Robertazzi, and S. Luryi, 1998: Optimizing computing costs using divisible load analysis. *IEEE Transactions on Parallel and Distributed Systems*, vol.8(3), pp.225-

234, 1998.

- Sterling, T., D.J. Becker, D. Savarese, M. R. Berry, and C. Res, 1996: Achieving a balanced low-cost architecture for mass storage management through multiple fast Ethernet channels on the Beowulf parallel workstation. *Proceedings, International Parallel Processing Symposium*.
- Sterling, T., D. Becker, D.F. Savarese, et al.: BEOWULF: A parallel workstation for scientific computation. *Proceedings of the International Conference on Parallel Processing*, 1995.
- Sunderam, V., 1996: PVM, A framework for parallel distributed computing. *Concurrency: Practice and Experience*, vol.2, no.4, pp.315-339
- Tam, A. and C.L. Wang, 1999: Realistic communication model for parallel computing on cluster. *1999 IEEE Conference on Cluster Computing*, 92-101, 1999.
- Wang, H. and J. Shen., 1997: *Parallelization of FFT with MPI and PVM*. Class project for CS625 instructed by Dr. M.P. Prabhu., Iowa State University, Ames, Iowa.
- Wang, H., G.M. Prabhu, and E.S. Takle, 1998: Parallelizing a very high resolution climate model using cluster of workstations with PVM and performance and load balance analyses. *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, CSREA Press, pp.1762-1765.
- Wang, H. G.M. Prabhu, E.S. Takle, R. Todi, 1999: Implementation and performance evaluation for a computation-intensive climate simulation application, *Proceedings of ParCo99 International Conference on Parallel Computing*, Netherlands, 1999.
- Wang, H., G.M. Prabhu, E.S. Takle, and J. Shen, 2000: Implementation and Performance

- Evaluation of Parallel Computations by Using Cluster of Networked Workstations. *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, CSREA Press, pp. 2007-2013.
- Wang, H. and E.S. Takle, 1995a: Boundary-layer flow and turbulence near porous obstacles: I. Derivation of a general equation set for a porous medium. *Boundary-Layer Meteorology*, **74**, 73-88.
- Wang, H. and E.S. Takle, 1995b: A numerical simulation of boundary-layer flows near shelterbelts. *Boundary-Layer Meteorology*, **75**, 141-173.
- Wang, H. and E.S. Takle, 1996a: On three-dimensionality of shelterbelt structure and its influences on shelter effects. *Boundary-Layer Meteorology*, **79**, 83-105.
- Wang, H. and E.S. Takle, 1996b: On shelter efficiency of shelterbelts in oblique wind. *Agric. Forest Meteorology*, **81**, 95-117.
- Wang, H. and E.S. Takle, 1996c: Numerical simulations of shelterbelt effects on wind direction. *Journal of Applied Meteorology*, **34**, 2206-2219.
- Wang, H. and E.S. Takle, 1997a: Momentum budget and shelter mechanism of boundary-layer flow near a shelterbelt. *Boundary-Layer Meteorology*, **82**, 417-435.
- Wang, H. and E.S. Takle, 1997b: Model-simulated influences of shelterbelt shape on wind-sheltering efficiency. *Journal of Applied Meteorology*, vol.36.
- Wang, H., E.S. Takle, and J. Shen, 2001: Shelterbelt and Windbreaks: Mathematical modeling and computer simulations of turbulent flows. *Annual Review of Fluid Mechanics*, **33**:549-586.
- Williamson, D.L., J.B. Drake, J.J. Hack, R. Jakob, and P.N. Swarztrauber, 1992: A standard test

- set for numerical approximations to the shallow water equations on the sphere. *J. Comp. Phys.*, **102**, 221-224.
- Watts J. and S. Taylor, 1998: A practical approach to dynamic load balancing. *IEEE Transactions on Parallel and Distributed Systems*, vol.9(3), 1998.
- Worley, P.H., I.T. Foster, and B. Toonen, 1995: Algorithm comparison and benchmarking using a parallel spectral transform shallow water model. In *Proceeding of the Sixth Workshop on Parallel Processing in Meteorology*, eds. G.-R. Hoffmann and N. Kreitz. Singapore: World Scientific, 277-289.
- Yap, T.K., O. Frieder, and R. L. Martino, 1998: Parallel computation in biological sequence analysis. *IEEE Transactions on Parallel and Distributed Systems*, vol.9(3), pp.283-294, 1998.
- Zhu, M., W. Cai, B.S. Lee, 1999: Key message algorithm: a communication optimization algorithm in cluster-based parallel computing. *IEEE Conference on Cluster Computing*, 121-128, 1999.

CHAPTER 2. SHELTERBELTS AND WINDBREAKS: MATHEMATICAL MODELING AND COMPUTER SIMULATIONS OF TURBULENT FLOWS

A paper published in the Annual Review of Fluid Dynamics, 2001, vol.33, 549-586

Hao Wang , Eugene S. Takle, and Jinmei Shen

Department of Computer Science and Department of Atmospheric Science

Iowa State University, Ames, Iowa 50011, USA

Abstract

Shelterbelts or windbreaks have been used for centuries to reduce wind speed, to control heat and moisture transfer and pollutant diffusion, to improve climate and environment, and to increase crop yields; but only within the last few decades have systematic studies considered the aerodynamics and shelter mechanisms of shelterbelts and windbreaks. This is a review of recent modeling and numerical simulation studies as well as the mechanisms that control flow and turbulence around shelterbelts and windbreaks. We compare numerical simulations with experimental data and explain the relationships between sheltering effects and the structure of shelterbelts and windbreaks. We discuss how and why the desired effects are achieved by using numerical analysis. This chapter begins with the derivation of a general equation set for porous shelterbelts and windbreaks; the numerical model and simulation procedure are developed; unseparated and separated flows are predicted and characterized; the momentum budget and

shelter mechanisms are analyzed; the effects of wind direction, density, width, and three dimensionality of shelterbelt structure on flow and turbulence are systematically described. Recent modeling and simulation of heat flux and evapotranspiration also are summarized. Finally, we discuss use of high-performance distributed and parallel computing as well as clusters of networked workstations to enhance performance of the model applied to simulations of shelterbelts and windbreaks.

Introduction

Humans always have used trees and large vegetation structures for protection from sun, wind, sand, and snow. With the advent of the agricultural era, planting of trees and other vegetation offered a natural means of protecting not only humans but crops and animals as well. Increased sophistication in agriculture and landscape management along with more intense interest in the physical and microclimatic interactions in heterogeneous plant ecosystems calls for more advanced understanding of the mechanisms by which the flow of air is modified by vegetative structures. The general case of flow through a porous three-dimensional heterogeneous vegetative structure is very complex, but simplification to a two-dimensional living barrier, such as a shelterbelt or windbreak, offers opportunity both to advance the science of turbulent flow and to apply such advanced principles to practical applications. We review recent advances in the theory of turbulent flow through vegetation and its application to agricultural shelterbelts.

The primary effect of any shelterbelt or windbreak system is to reduce wind speed. Wind-speed reduction influences turbulent transport processes and modifies the microclimate in the sheltered zone. The amount of sheltering and the range of the sheltered zone depend on

the structure of the shelterbelt such as porosity, thickness, shape, and environmental conditions such as wind direction. Experimental studies of wind speed and turbulence have been done in the laboratory with wind-tunnel models and at full scale in the field. Van Eimern (1964), Rosenberg (1979), McNaughton (1988), and Heisler & Dewalle (1988) provide interpretations of experimental data that have accumulated over the last half century. Most shelterbelt research reported in the literature has been experimental, emphasizing wind-speed reduction produced by shelterbelts. A full understanding of shelterbelt aerodynamics has been slow to accumulate, even for the relatively simple artificial linear barrier of uniform porosity resting on a uniform surface of infinite extent.

Analytic solutions are somewhat easier to obtain for the far wake region ($x > 10 H$, where H is the height of the obstacle), well downstream of the obstacle itself; but for the region in which the protection is greatest ($x < 10 H$), the streamline shape becomes important, and quantitative results become rather complex and difficult to obtain analytically. Kaiser (1959) obtained an error-function formula by assuming that the momentum deficit in the sheltered region is replenished by diffusion of a passive scalar. This model does not have sufficient physics to describe complex turbulent flow interactions and cannot be expected to predict the location of maximum wind reduction. The complexity of the problem is discussed by Pate (1971) who identified more than 7 distinctive regimes of flow around shelterbelts or windbreaks.

Pioneering theoretical work on flow within vegetative canopies was reported by Wilson & Shaw (1977) who derived the one-dimensional equations for nonbuoyant flow with large horizontal-plane averaging. Raupach and Shaw (1982) extended this work for horizontally

homogeneous canopies, and Finnigan (1985) generalized the area average to a volume average within the canopy. Theoretical and numerical modeling research specifically focusing on shelterbelts has been reported by Kaiser (1959), Plate (1971), Counihan et al (1974), and Hagen et al (1981).

In the recent years, improvements in computers and computing techniques together with better numerical modeling and simulation techniques have allowed for deeper understanding of the aerodynamics and mechanisms of turbulent flow around shelterbelts and windbreaks. Development of a detailed basic understanding of the flow field is a complex analytical and numerical problem since a complete treatment requires a solution of the full turbulent Navier-Stokes equations.

Extensions of aerodynamic flow models to include non-neutral effects in the atmosphere requires that attention be given to effects of both heating and moisture. Evaporation from the soil surface and transpiration by plants depend on the availability and movement of moisture in the soil. And the partitioning of incoming solar radiation at the surface into reflected solar energy, sensible heating of the atmosphere, evaporation, and heat transfer to the soil require we also include multiple layers of soil to properly simulate surface processes. While these processes represent considerable complexity, they have been simulated with reasonable success for one-dimensional coupling of the atmosphere to the surface/vegetation for application to regional and global climate models (Bonan, 1996). These one-dimensional representations of flow near the boundary describe the aerodynamic properties of the surface/vegetation by resistance and drag coefficients and do not consider flow through the vegetation itself.

Simulating the aerodynamics of the heterogeneity of vegetation with the expectation one

vegetation component will modify the mean and turbulent flow field for vegetation in the downwind parts of the domain is an extension that had not been attempted until 1990's. Unlike the obstacle-free atmospheric boundary layer for which the Reynolds averaged Navier-Stokes equations are suitable, mean flow and turbulence in and around porous obstacles, such as forest or crops, are very dependent on physical properties of the vegetation. All physical variables that describe the flow, such as wind speed, pressure, temperature, and turbulent variables and the equations for mean and turbulent properties are defined in the interstitial air space within the porous obstacle but have no meaning in the space occupied by solid elements. These solid elements of the obstacle act as interior boundaries within the atmospheric boundary-layer flow and therefore present complex problem for applying boundary conditions. Only for two-dimensional artificial barriers with simple geometric shapes (such as a fence or plane having circular holes) is it possible to treat the details of the barrier boundaries to the flow. For natural tree shelterbelts and most artificial shelterbelts, the boundaries of the solid elements are too complex and irregular to be treated explicitly. Fortunately, it is unnecessary to treat all details of solid element boundaries, because the purpose of shelterbelt research is to describe the flow in the protected zone outside shelterbelt rather than in the shelterbelt itself. We can neglect the detailed structure of the complex solid boundaries within shelterbelt and consider them as an aggregate effect, described by a surface drag force on the airflow penetrating the shelter (Wang & Shen 1989, Wang 1991a,b, 1992, Wang & Takle 1994a,b, 1995 a,b,c,d,1996a,b,c,d,e, 1997a,b,c,d,e).

We reported derivations of a set of general equations of flow and turbulence for porous media (Wang & Takle 1995a) and developed a shelterbelt boundary-layer turbulent flow model

for studying the aerodynamics and mechanisms of shelterbelts and their dependence on shelterbelt structures (porosity, three dimensionality, shape) and environmental conditions such as wind direction as well as the momentum budget (Wang & Takle 1995b,c,d, 1996a,b,c,d,e, 1997a,b,c,d,e, 1998a, b, Wang et al 1998, 1999). We also extended this model to predict heat flux and evapotranspiration as affected by shelterbelts and windbreaks. The added model complexity requires optimization of model performance by using parallel and distributed computing techniques. In the following sections we give a summary of these results and suggest some future applications of numerical simulation of flow through heterogeneous vegetation.

Mathematical Modeling

Air-Phase High-Wavenumber Averaging

Air passing through a rigid but porous obstacle is a two-phase (air-solid) system. Performing the averaging process in this two-phase system is different from that in the pure fluid of single phase. The Navier-Stokes equations are valid only in the space occupied by air within the porous obstacle and are not valid in the space occupied by solid elements. Wilson & Shaw (1977), Raupach & Shaw (1982), Finnigan (1985), and Raupach et al (1986) performed spatially averaging by excluding solid elements and also recognized that the spatial averaging has physical significance beyond a purely statistical process. However, some differences between two-phase spatial averaging and the single-phase spatial averaging were not considered in these early studies. The volume-averaging process in a two-phase medium is fundamentally different from the conventional spatial averaging process commonly used for turbulent flows in fluid

dynamics, although both operate in some space volume. Conventional spatial averaging of turbulent flows is intended to produce steady-state turbulent statistics and mean variables. The introduction of air-phase averaging in air-solid two-phase medium avoids the inconvenience of addressing the existence of solid elements in porous-medium flows and produces a set of equations that include effects of solid elements (but not the elements themselves) and that are valid in the entire space. By this process we convert very complex solid-obstacle boundary effects into additional terms of the equations and make the physical variables of the equations continuous in the entire space. In order to distinguish multi-phase volume averaging from conventional single-phase spatial averaging and to avoid confusion with averaging over phase-angle of waves (e.g. Finnigan & Einaudi 1981), we follow the terminology from chemical engineering (Whitaker 1973) by referring to air-volume averaging in air-solid two-phase medium as *air-phase averaging*.

Clarification of the air-phase averaging process requires introduction of three length scales: d represents the characteristic scale of obstacle elements, L represents the characteristic scale of atmospheric boundary-layer turbulence, and D stands for the scale of the averaging volume. The value of D can be selected arbitrarily, and the averaging operator will smooth turbulence with wavenumber larger than the high-wavenumber $2\pi/D$. However, of most importance is wake turbulence generated by obstacle elements, its interactions with atmospheric shear turbulence produce distinctive characteristics of turbulence and result in changes of constants and parameters of turbulence closure schemes. Our averaging volume is small enough to eliminate only the effects of obstacle elements' structure with characteristic scale of d . For crops and forest, we have

$$L \gg d \quad (1)$$

and accordingly the eliminated minimum wavenumber of $2\pi/D$ is much higher than that of $2\pi/L$ in Reynolds averaging. Therefore, air-phase averaging not only allows the effects of obstacle elements to be described by equations that hold in the entire space but also acts as a high-wavenumber averaging. The high-frequency turbulence generated by vegetation has been revealed by a double peak in the turbulence spectrum observed in vegetation environments (e.g. Zhu et al 1992).

Generally speaking, when the space between obstacle elements is not large enough to produce steady turbulence statistics, we cannot use time averaging followed by space averaging to approximate the air-phase averaged equations. We also cannot use large-space averaging to get both time-steady mean values and turbulent statistics. Doing so misses the turbulence interactions and energy-cascade processes between multiple obstacle elements in the averaging space.

Following the above procedure we get the air-phase high-wavenumber-averaged continuity equation as

$$\frac{\partial \langle u_i \rangle}{\partial x_i} = 0 \quad (2)$$

and the air-phase-high-wavenumber-averaged equations of motion as

$$\begin{aligned}
\frac{\partial \langle u_i \rangle}{\partial t} = & -\langle u_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} - \frac{\partial \langle \tilde{u}_i \tilde{u}_j \rangle}{\partial x_j} - \varepsilon_{ikl} f_k \langle u_l \rangle - \beta g_i \langle \theta \rangle - \frac{\partial \langle p \rangle}{\partial x_i} \\
& + \nu \frac{\partial^2 \langle u_i \rangle}{\partial x_j^2} - \left(\frac{1}{V} \int_S p n_i dS' - \frac{\nu}{V} \int_S \frac{\partial u_i}{\partial n} dS' \right)
\end{aligned} \quad (3)$$

In Equation (3) the last two terms, which represent, respectively, the integration of the pressure and wind shear over the element surface within the averaging volume, are momentum sinks consisting of pressure drag and skin friction created by obstacle elements. The drag force may be expressed by the commonly used formula introduced by Thom (1975)

$$F_i = \rho_0 C_d A U u_i \quad (4)$$

where ρ_0 is the air density, C_d is a drag coefficient for unit plant area density, A is the plant area density, and U is the mean windspeed.

After air-phase-high-wavenumber-averaging, the equation of high-frequency turbulent kinetic energy, E , can be obtained as

$$\begin{aligned}
\frac{\partial E}{\partial t} = & -\langle u_k \rangle \frac{\partial E}{\partial x_k} - \langle \tilde{u}_i \tilde{u}_k \rangle \frac{\partial \langle u_i \rangle}{\partial x_k} - \beta g_i \langle \tilde{u}_i \tilde{\theta} \rangle - f_k \varepsilon_{ikl} \langle \tilde{u}_i \tilde{u}_l \rangle \\
& - \frac{\partial \langle \tilde{u}_k E \rangle}{\partial x_k} - \frac{\partial \langle \tilde{u}_i \tilde{p} \rangle}{\partial x_i} \\
& + \langle \tilde{p} \frac{\partial \tilde{u}_i}{\partial x_i} \rangle + \nu \frac{\partial^2 E}{\partial x_k^2} - \nu \langle \frac{\partial \tilde{u}_i}{\partial x_k} \frac{\partial \tilde{u}_i}{\partial x_k} \rangle + C_d A U \langle u_i \rangle^2
\end{aligned} \quad (5)$$

The last term in Equation (5) represents conversion of mean kinetic energy of air flow into high-frequency turbulent kinetic energy of the wake flow created by drag on the air flow past internal elements of the obstacle. These equations apply to the entire space, including the space

occupied by obstacle elements without making any assumptions, except for the drag force formula (which is widely accepted) and the drag coefficient (which has been measured to be constant by many field observations and laboratory tunnel studies for a wide variety of obstacles). This process changes a two-phase problem into a single-phase (pure air) problem, and produces equations of high-frequency wake turbulence generated by obstacle elements.

Reynolds Averaging

Field observations and laboratory measurements within or near porous obstacles have revealed a double peak in the turbulent spectrum, the high-frequency peak being generated by the obstacle-air interaction as described in the high-frequency stress equations given in the previous section. The high-frequency component dissipates more quickly and has less contribution to the total canopy turbulence at the actually observed turbulence level (Wilson & Shaw 1977, Raupach & Shaw 1982). However, it may interact with shear turbulence and change the turbulence structure and thereby cause changes in parameters and constants of turbulence closure schemes.

We have used air-phase high-wavenumber averaging to obtain a mathematically and physically consistent set of equations that hold in the entire space, including the space occupied by obstacle elements. We do not require the air-phase-high-wavenumber-averaging volume large enough to have a statistically steady mean flow and statistically steady turbulent quantities. As discussed in above section the averaging volume is the minimum with characteristic scale of D to eliminate only the effects of obstacle elements' structure on flow with minimum wavenumber of $2\pi/D$ which is much higher than that of shear turbulence. Therefore, we must

again average the air-phase-high-wavenumber-averaged equations over all-wavenumber by the commonly used space-averaging or time averaging methods to obtain equations for statistically steady mean variables and turbulence. Since we have transferred the Navier-Stokes equation set, which holds only in the air space of the porous-medium, to the porous-medium equation set which holds in the entire space just as in the pure atmosphere, the time-averaging or the space-averaging procedures are the same as are commonly used in the pure atmosphere. We here use Reynolds averaging as an example, and the equations of mean motion may be written as follows

$$\begin{aligned} \frac{\partial \langle u_i \rangle}{\partial t} = & -\langle u_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} - \frac{\partial (\langle \tilde{u}_i \tilde{u}_j \rangle + \langle u_i \rangle \langle u_j \rangle)}{\partial x_j} - \varepsilon_{ikl} f_k \langle u_l \rangle \\ & - \beta g_i \langle \theta \rangle - \frac{\partial \langle p \rangle}{\partial x_i} + \nu \frac{\partial^2 \langle u_i \rangle}{\partial x_j^2} - \mathbf{C_d A U} \langle u_i \rangle \end{aligned} \quad (6)$$

Compared to equations for the obstacle-free atmosphere, the continuity equation has the same form, but the equations of mean motion have two additional terms (shown in bold type). These two terms have clear physical significance: the last term is the well-known drag force exerted by obstacle elements on the mean air flow, and the other additional term associated with Reynolds stress is the averaged value of the high-frequency turbulent stress. This latter term describes the effect of averaged high-frequency turbulence generated by obstacle elements on the mean flow. Although our derivation produces additional terms that resemble previous results, the physical essence of these terms is different from dispersive terms of previous reports. The time and space scales of the high-frequency turbulence generated by obstacle elements are much smaller than those of the low-frequency turbulence generated by wind shear

and temperature stratification, so the spatial averaging of the time-averaged variables cannot represent the high-frequency turbulence component. Also the flow that interacts with the obstacle elements is not the steady time-averaged mean wind.

The low-frequency turbulent kinetic energy equation can be written as

$$\begin{aligned}
 \frac{\partial e}{\partial t} = & -\overline{\langle u_k \rangle} \frac{\partial e}{\partial x_k} - \overline{\langle u_i \rangle' \langle u_k \rangle'} \frac{\partial \overline{\langle u_i \rangle}}{\partial x_k} - \beta g_i \overline{\langle u_i \rangle' \langle \theta \rangle'} \\
 & - f_k \varepsilon_{ikl} \overline{\langle u_i \rangle' \langle u_l \rangle'} - \frac{\partial \overline{\langle u_k \rangle' e}}{\partial x_k} - \overline{\langle u_i \rangle'} \frac{\partial \overline{\langle \tilde{u}_k \tilde{u}_i \rangle'}}{\partial x_k} - \frac{\partial \overline{\langle u_i \rangle' \langle p \rangle'}}{\partial x_i} \\
 & + \overline{\langle p \rangle'} \frac{\partial \overline{\langle u_i \rangle'}}{\partial x_i} + \nu \frac{\partial^2 e}{\partial x_k^2} - \nu \frac{\partial \overline{\langle u_i \rangle'}}{\partial x_k} \frac{\partial \overline{\langle u_i \rangle'}}{\partial x_k} - 2 C_d A U e
 \end{aligned} \tag{7}$$

This equation includes two additional terms (shown in bold type) that were first reported by Wang & Takle (1995a). This equation represents changes in mean turbulence due to interaction with obstacle elements. The first additional term is the energy redistribution due to the interaction of the low-frequency velocity fluctuation from the space/time average with the fluctuation of the high-frequency velocity covariance from the space time average. The last additional term, the most important term in porous-obstacle flows, describes the turbulent-energy-cascade process of breaking down larger scales of motion into smaller scales of motion. Turbulent kinetic energy is lost by two processes, one being molecular dissipation and the other being the cascade process. There is no need to introduce a modified mixing length because the physical processes are captured by these additional terms of Equation (7). This procedure replaces the one-step averaging process of previous authors which led to a single turbulence kinetic energy equation that did not physically separate out obstacle-generated turbulence and

transfer of energy from low frequency to high frequency.

Wang & Takle (1995a) showed that the convention of using a single mixing length to represent both molecular dissipation and the turbulent energy cascade process is an unnecessary and unphysical simplification. Previous researchers had fit the observational data by modifying the mixing length of the molecular dissipation by trial and error methods. The procedure by which they fit the data may be demonstrated as follows. The turbulent energy dissipation and cascade terms can be rewritten in the following form:

$$2 C_d A U e + \epsilon_e = 2 C_d A U e + \frac{e^{3/2}}{\Lambda} = e^{3/2} (2 C_d A U e^{-1/2} + \frac{1}{\Lambda}) = \frac{e^{3/2}}{\Lambda'} \quad (8)$$

where Λ is the mixing length of the low-frequency turbulent molecular dissipation (including a constant), and Λ' is the "mixing length" which previous investigators adjusted to fit the observational data. Based on our results, Λ' has the following relationship with Λ and the drag produced by obstacle elements:

$$\frac{1}{\Lambda'} = \frac{1}{\Lambda} + 2 C_d A U e^{-1/2} = \frac{1}{\Lambda} + 2^{1/2} C_d \frac{A}{\sigma} \quad (9)$$

where σ is turbulence intensity. From formula (9) we can conclude that Λ' is not only related to the obstacle element ($C_d A$) and Λ , but it also is related to windspeed and turbulent kinetic energy. The last term shows that it is proportional to the inverse of the turbulence intensity. This suggests that the mixing length for porous-medium flows is not simply a combination of mixing lengths of the obstacle-free atmosphere and the size of the obstacle elements as commonly thought.

Numerical Simulation

Simplification of Governing Equations

The general equations for application to neutral flow derived in the preceding section were simplified in Wang & Shen 1989, Wang 1991a,b, 1992, Wang & Takle (1995b,c, 1996a,b,c, 1997a,b), and Wang et al (1998, 1999) are summarized here. A typical shelterbelt is about 10 m high, which is much less than the height of the atmospheric boundary layer, so the effect of Coriolis forces may be neglected. Shelterbelts generally are planted in rows perpendicular to the prevailing wind direction, and their length is at least one order larger than their height. We use a quasi-3-dimensional computational domain (vertical (z) and horizontal (x) perpendicular to the shelterbelt) in which the shelterbelt is infinitely long in the y-direction but allows fully 3-dimensional flow to be simulated. Flow oblique to the shelter can be simulated (not possible in 2-D simulations), but all calculated fields are uniform in the y direction (simpler and less general than fully 3-dimensional simulations). For these conditions, the basic equations for motion and continuity may be simplified as

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial x} - u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y} - w \frac{\partial u}{\partial z} - \frac{\overline{\partial u'^2}}{\partial x} - \frac{\overline{\partial u'v'}}{\partial y} - \frac{\overline{\partial u'w'}}{\partial z} - C_d A U \quad (10)$$

$$\frac{\partial v}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial y} - u \frac{\partial v}{\partial x} - v \frac{\partial v}{\partial y} - w \frac{\partial v}{\partial z} - \frac{\overline{\partial u'v'}}{\partial x} - \frac{\overline{\partial v'^2}}{\partial y} - \frac{\overline{\partial v'w'}}{\partial z} - C_d A U v \quad (11)$$

$$\frac{\partial w}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial z} - u \frac{\partial w}{\partial x} - v \frac{\partial w}{\partial y} - w \frac{\partial w}{\partial z} - \frac{\overline{\partial u'w'}}{\partial x} - \frac{\overline{\partial v'w'}}{\partial y} - \frac{\overline{\partial w'^2}}{\partial z} - C_d A U w \quad (12)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (13)$$

where u , v and w are mean windspeed components in x , y and z direction, respectively, and u' , v' and w' are their fluctuating values. For convenience we omit the overbar on mean values. p is the pressure perturbation, t is time, and ρ is air density. The last term in each of Eqs.(10), (11) and (12) is the parameterized drag force exerted by shelterbelts following Thom (1975), Wilson and Shaw (1977). Several investigators used this method to successfully simulate forest and crop flows (Wilson and Shaw 1977, Wilson 1985, Yamada 1982, Meyers and Paw U 1986, Naot and Mahrer 1991, Li et al 1989, Miller et al 1991). U is total mean windspeed, $A(x,y,z)$ is the leaf-area density (LAD), and C_d is the unit LAD form-drag coefficient.

Equations (10), (11), and (12) of mean motion include turbulent stress terms for which we must apply one of the boundary-layer turbulent-closure schemes. We select a K-E-l turbulence closure scheme which solves prognostic equations only for turbulence kinetic energy (TKE) and master length scale (Yamada 1982, Yamada & Mellor 1975).

Solution techniques

We solve a system of 8 equations including equations for horizontal motion, nonhydrostatic vertical motion, mass conservation, TKE, and mixing length. We use the finite-difference method to discretize these equations into a set of algebraic equations with tri-diagonal matrices, with forward differencing for the time terms, centered differencing for pressure terms, and upstream differencing for advection terms. The modified Crank-Nicholson scheme is used for

the turbulent flux terms (Paegle et al 1976). We use the Alternating Direction Implicit (ADI) method to solve these equations in both vertical and horizontal directions. Because they include dynamic pressure, the equations are of the mixed parabolic-elliptic type. The dynamic pressure perturbation gradients are dropped from the momentum equations, and a set of auxiliary velocity fields u^{aux} and w^{aux} are computed based on Chorin's (1968) scheme. We solve the dynamic pressure equation by the SOR method with the relaxation factor taken to be 1.75 and the successive convergence criterion set to $|\delta p_{max}| < 10^{-4} \text{ m}^2$.

Flow and Turbulence Around Shelter Belts and Windbreaks

Porosity and Resistance Coefficient

The resistance coefficients may be estimated as

$$k_r = \int_{-\infty}^{+\infty} C_d A dx \quad (14)$$

The relationship between porosity and resistance coefficient (k_r) has been revealed by many tunnel experiments. Based on Figure 1 of Heisler & Dewalle (1988) and Hoerner's (1965) formula, we estimate the porosities corresponding to form drag coefficients. We emphasize that porosity only represents the planar geometrical structure of shelterbelt, but that changes in flow are more correctly described by dynamic, not geometrical, parameters. The resistance coefficient is a dynamic parameter that depends not only on porosity but also the shape of the barrier elements. Barriers of equal porosity may have different k_r and different shelter effects. As reviewed by Heisler & Dewalle (1988), Baines & Peterson (1951) reported a range in k_r from 1.0 for a lattice of round material to 3.2 for a square bar lattice for a constant porosity of 0.5,

and Richards et al (1984) found a somewhat larger range in k_t from 1.5 for a screen of smooth round elements to 5.0 for flat, sharp-edged elements, also for a porosity of 0.5. However, porosity is easier to be understood and visualized. We will use porosity when possible in our description.

Flow Patterns and Streamlines for Shelterbelts with Different Porosity

Wang & Takle (1995b) examined the characteristics and patterns of flow and turbulence for the whole range of porous shelterbelts from nearly solid (porosity=0.0) to nearly open (porosity=1.0). The results divide into two regimes of flow patterns:

Unseparated Flow. Highly porous obstacles (Figure 1a) produce streamlines characteristic of typical unseparated flows. With decreasing porosity, streamline curvature increases: the upward streamlines become steeper and streamlines become compressed over the top of the shelterbelt. Accordingly, there are three zones: the windward wind-reduction zone, the leeward wind-reduction zone, and the over speed zone over the shelterbelt. When the porosity decreases to 0.3, recirculation is generated behind the shelterbelt.

Separated Flow. The characteristics of separated flow are important features of the physics and dynamics of shelterbelts. Differences of views about the relationship between shelter effects and shelterbelt density center on this regime. Figure 1b shows wind vector characteristics of typical separated flow. Recirculation at the separation point initially is very weak and occurs at a porosity of 0.30 (separating at $x=5 H$ and reattaching at $x=8 H$). The stagnation point is at $x=6.5 H$ and $z=0.15 H$. The center of the recirculation zone migrates gradually toward the shelter and up as the porosity decreases.

Available observations and smoke trace experiments have shown that dense shelterbelts with porosity less than 0.3 may produce a recirculation bubble in their lee (Heisler & Dewalle 1988, Perera 1981, Castro, 1971). The recirculation is observed to be rather weak, and the recirculation zone is quite small when it first appears at the critical porosity of 0.3. Even for a porosity of 0.27 (resistance coefficient of 4.0), the simulated maximum reverse wind speed is still less than 0.5 m/s. Quantitative measurements are very difficult in highly distorted flows for small reverse wind speed making it quite likely that small and weak recirculation far downstream escapes observation.

Perera (1981) made extensive wind-tunnel measurements of different model fences with porosities ranging from 0.0 to 0.5. He reported that the recirculating bubble detaches from the fence (at porosities less than 0.3) and moves downstream as the porosity of the fence increases. Perera's observations are consistent with earlier results of Castro (1971). Numerical simulations with the model previously described therefore agree with observed dependence on porosity of the onset of recirculation.

Wang & Takle (1995b) used a numerical model to construct a set of shelterbelt flow patterns and to describe their changes with porosity. The location of maximum wind-speed reduction, its changes with porosity, and the leeward wind-speed recovery rate are in general agreement with observations in the laboratory and in the field, as are the separated recirculation and its change in size and location with porosity. The model performed well for flows over and through shelters ranging from almost completely open, where the governing equations are parabolic, to almost solid shelterbelts, where the governing equations are elliptic and where the separated flow exists.

Flows and Pressure Interactions

A notable result of these simulations was the importance of correct simulation of the pressure field across (within) the shelter (Wang & Takle 1995b,d, 1996a,b,c,d, 1997a,b, Takle et al 1998, 1999). Although the issue has been debated, we assert that it is critically important that momentum be extracted locally and incrementally at numerous points within the shelter (rather than by bulk extraction) in order to obtain the appropriate convergence and divergence fields (and hence pressure) upwind and downwind of the barrier. The importance of the correct dynamic pressure is revealed by the wind-sheltering functions and characteristics. The interaction of penetrating flow with the perturbation pressure and flows over the shelter creates a point of maximum wind-speed reduction far behind the shelterbelt. Dynamic pressure resulting from convergence and divergence of the flow field alters the perturbation pressure field. The disturbed pressure controls not only the formation of the separated flow but also the location of maximum wind-speed reduction, streamline curvature, speed-up over the shelterbelt, and leeward wind-speed-recovery rate. The interaction of pressure and flows produces complex flow patterns, the characteristics of which are determined, to a great extent, by the ratio of the penetrating flow to undisturbed flow, or permeability. The permeability is controlled by shelterbelt structure, which traditionally is expressed in the terms of porosity or leaf-area index density.

Although the leeward wind-speed-reduction zone is most important for practical applications, as a whole, all parts of the flow pattern are interconnected. The overspeeding over the shelterbelt alters streamline curvature and hence affects the wind-speed-recovery rate. The

windward wind-speed reduction affects the overspeeding zone by changing the drag force and vertical transport of horizontal momentum.

The Effects of Three Dimensionality of Shelterbelt

Natural shelterbelts, unlike planar barriers, have a finite width, within which interactions among wind speed, drag force and pressure perturbations determine the net sheltering effect. Because experimental conditions are very difficult to control for natural tree shelterbelts in the open environment, numerous experiments have been conducted with artificial fences or screens either in laboratory wind tunnels (e.g., Raine & Stevenson 1977, Ogawa & Diosey 1980, Perera 1981) or in the atmospheric boundary layer (e.g., Bradley & Mulhearn 1983, Jacobs 1984). As a result, the shelter effects and aerodynamics of artificial fences and screens (nearly infinitely thin), including the role of shelterbelt density (porosity) in determining shelter effects, are largely understood. Natural shelterbelts, however, have width and three-dimensional spaces through which the wind flows across the width of the barrier, leading to distinctly different aerodynamic effects compared with artificial fences and screens with two dimensional gaps (Sturrock 1969, 1972, Heisler & DeWalle 1988). It is difficult if not impossible to separate the effect of width from the effect of overall density in both field observations and wind-tunnel measurements (van Eimern et al 1964, Heisler & DeWalle 1988). Wang & Takle (1996a) reported a set of numerical experiments using the model system as previously described to study the effects of three dimensionality of shelterbelt structure. These are summarized in this section.

Permeability and Its Dependencies on the Three Dimensionality of Shelterbelt Structure

The permeability (ϕ) is defined as a percentage of windspeed ($U_{x=0}$) at the back edge of the shelterbelt to the upstream undisturbed windspeed (U_0), and is sometimes used as a descriptor of shelterbelt density (van Eimern et al 1964). Modification of windspeed is a consequence of momentum loss due to drag of the shelterbelt, but the changed windspeed directly changes drag force as previously mentioned. Therefore, examination of the distribution of windspeed within shelterbelts and permeability contributes to a better understanding of dynamic processes and of the effects of the structure and width of shelterbelts. The permeability indicates the ability of shelterbelts to suppress flow penetration, so it is closely related to windspeed changes within shelterbelts. Figures 2 is relative windspeed across shelterbelts, which show four patterns of windspeed changes across shelterbelts: (1) monotonic increase (2) increase in the front but decrease in the back (3) monotonic decrease and (4) decrease in the front but increase in the back. The patterns of windspeed change depend on width of the shelterbelt and height of observation.

Perturbed Pressure and Its Dependencies on the Three Dimensionality of Shelterbelt Structure

The drag exerted on air by shelterbelts disturbs the pressure field; however, the disturbed pressure modifies the windspeed, and the modified windspeed again changes the drag. Moreover, the divergence of the wind field also causes changes in the pressure field, and this change in the pressure field occurs far beyond the shelterbelt unlike changes due to the drag that

occurs only within the shelterbelt. Therefore, the disturbed pressure plays an important role in the distribution of windspeed and shelter effects.

Pressure has a maximum immediately in front of the belt (Wang & Takle 1995b). In the lee, pressure gradually recovers to the undisturbed value. The recovery rate shows somewhat complex behavior with leeward distance, because divergence of windspeed also changes the perturbed pressure, which, in turn, causes windspeed reduction. For narrow shelterbelts, the perturbed pressure changes less between $x=1 H$ and $6 H$ and rapidly recovers beyond $x=7 H$. The location (X_{min}) of the minimum windspeed is far from the belt, so convergence of the wind field between $x=0$ and X_{min} produces an additional positive pressure perturbation, which tends to compensate the negative static pressure in the near lee. Beyond X_{min} , an additional negative pressure perturbation produced by the divergence of wind field adds to the pressure. Therefore, there is a plateau of pressure perturbation in the middle lee ($1-7 H$). We also find that the difference in the perturbed pressure between shelterbelts narrower than $3 H$ is small. However, the perturbed pressure changes with width are larger for wide shelterbelts, and the leeward plateau of perturbed pressure is no longer obvious for shelterbelts wider than $5 H$.

Shelter Effects and Its Dependencies on the Three Dimensionality of Shelterbelt Structure

Shelter effects may be expressed by many different terms, but the most commonly used one is the shelter distance (d_{20}) over which windspeed in the lee is reduced by 20%. The minimum windspeed (U_{min}) (i.e., the maximum windspeed reduction) and its location (X_{min}) also are used to characterize the shelter effects (Heisler & DeWalle 1988). These indices describe only

specific characteristic points, even though these points are closely related to the whole horizontal profile of windspeed.

Change in width of shelterbelts causes modification of horizontal profiles of windspeed shown in Figure 3. The location of the minimum windspeed (X_{\min}) moves toward the belt with increasing height: for $W_{\text{SB}}=0.1 H$, $X_{\min}=6 H$ at $z=0.1 H$. With increasing width, the location of the minimum windspeed (X_{\min}) also moves toward the belt and may move into the wider shelterbelt (negative X_{\min}) at higher levels. The location of minimum windspeed moves from $6 H$ for $W_{\text{SB}}=0.1 H$ to $1.5 H$ for $W_{\text{SB}}=10 H$. The shelter distance (d_{20}) also decreases with increasing width, but the difference is only $2.5 H$ between $W_{\text{SB}}=0.1$ and $10 H$. With increasing width, the location of minimum windspeed (X_{\min}) moves upstream, and the minimum windspeed may occur inside the wide shelterbelt (negative X_{\min}). The maximum difference in the minimum windspeed is 6%, the shelterbelt with width $5 H$ having the lowest value.

As cited in van Eimern et al.'s (1964) Figure 12, Naegeli (1954) reported the observed differences in horizontal profiles of windspeed between a narrow medium-dense belt and a coniferous forest block with a width of $21.4 H$. The location of X_{\min} was $4 H$ for the narrow belt and $-2 H$ (inside the shelter) for the wide forest block. These are in good agreement with our simulated results. However, shelter distances (d_{20}) of $16 H$ for the narrow belt and $10 H$ for the wide forest block is larger than our simulated results, because the effects of the density have not been taken into account in the observed data. A forest block $21.4 H$ wide is likely very dense and may result in rapid recovery of windspeed in the lee. Takahashi (1978) measured windspeed near 68% porous vinyl nets of widths $0.5 H$, $2.5 H$, and $5 H$ in a wind-tunnel and observed the location of minimum windspeed for $z=0.25 H$ to be at $5 H$, $4H$, and $1 H$,

respectively, and the difference in shelter distance (d_{20}) to be small (within 2 H). These are in good agreement with our simulated results. Our simulated coefficient is also consistent with Taylor's (1988).

Variations of Wind Direction Around Shelterbelts

Shelterbelts influence the flow field by reducing windspeed and changing wind direction. The former effect has been intensively investigated, but the latter has been studied very little. Quantitative observation of wind direction is more difficult than the observation of windspeed, especially in wind-tunnel experiments, on which most existing knowledge of shelterbelt effects is based. Only few investigators reported qualitatively and quantitatively the observational difference in wind direction between sheltered and unsheltered sites (Bringmann & Kaiser 1955, cited by van Eimern et al 1964, Nord 1991). Shelterbelt modification of microclimate depends on wind direction (van Eimern et al 1964), because the drag force exerted by shelterbelts is strongly dependent on the wind trajectory through the barrier. However, almost all previous theoretical and numerical modeling work considered only the relatively simple example of a uniformly porous, artificial barrier resting on a uniform surface with wind blowing at right angle (Kaiser 1959, Plate 1971, Taylor 1988, Counihan et al 1974, Hagen et al 1981, Wilson 1985, Wang & Shen 1989, Wang 1991a,b, 1992, Wang & Takle 1995b,c,d, 1996a). We define the incidence angle (α) as an angle between undisturbed wind direction and the normal line (x axis) of the shelterbelt, and local wind angle (γ) as an angle between local wind direction and the normal line of the shelterbelt. We calculated flow fields for 19 incidence angles ranging from 0° to 90° in 5° intervals for a shelterbelt with porosity of 50%.

The distribution of local wind angle (γ) along the normal line (x axis) of the shelterbelt, reveal three wind-direction shift zones that have distinctive characteristics (Figure 4). The approaching air rotates toward the direction parallel to the shelterbelt, with a maximum departure just in front of the barrier; this is followed by an abrupt rotation in the opposite direction passing through the incidence angle to a maximum departure in the opposite direction just behind the shelterbelt. Beyond this point, the wind rotates back toward the incidence angle at a few H downstream. Rotation continues past the incidence angle, the wind becoming more parallel to the shelterbelt, and produces a maximum shift at a distance of 2-12 H. Further leeward, the wind slowly rotates back to the undisturbed angle.

The local wind direction twice passes through the undisturbed approach flow direction, the two zero-departure points being in the front of the shelterbelt and in the near lee. The changes of wind direction may be divided into three zones separated by these two points. We define the upwind zone where the attack angle is greater than the undisturbed incidence angle as the F (front) zone. The zone within and just behind the shelterbelt where the wind direction becomes less oblique (less than the upwind undisturbed incidence angle) we define as B (behind) zone. The region beyond the B zone where the wind direction becomes more parallel to the shelterbelt (greater than upwind undisturbed incidence angle) we define as W (wake) zone.

Nord (1991) reported quantitative observations of the effect of shelterbelts on wind-direction. Three-component anemometers equipped with light propellers of the Gill type were used to measure wind velocity under neutral stability at 2 m at several sites along a line perpendicular to a multiple-row shelterbelt with WNW-ESE orientation situated in the south

of Sweden. This belt was composed of birches and spruce with averaged height of 12 m, length of 100 m, and width of 20 m. The porosity of this shelterbelt varied with height, and the porosity, determined by matching the horizontal wind profile between the wind-tunnel model and the full-scale field shelterbelt, was estimated to be 0.55 in the upper part and 0.23 in the lower part.

The direction of the drag force for oblique flow always is opposite to the local flow; by contrast, the resulting gradient of the perturbation pressure (hence the pressure force) always is perpendicular to the shelter. This pressure force deflects oblique flow upwind of the shelter to a more shelter-parallel direction. As an air parcel enters the shelter, the drag force reduces its speed but does not alter its direction. However, the large pressure gradient within the shelter (Wang & Takle 1995b), being perpendicular to the barrier and therefore oblique to the flow, rotates the flow more normal to the barrier within the shelter. After exiting the shelter, the air parcel enters the leeward region of pressure plateau where its direction rotates back toward the incident angle due to vertical (downward) advection; but then it encounters a region where a weak pressure force of opposite sign deflects it slightly parallel to the barrier. Beyond this point the upwind direction gradually is recovered.

The Shelter Efficiency of Shelterbelts in Oblique Wind

A few papers have reported the observed changes of shelter effects in oblique wind. Lawrence (1955) observed that the shelter effect, defined as the distance over which wind speed was reduced by 20%, decreased 5% , 5-15%, 40-50%, and 60-70% at incidence angles (IA) of the approach wind (angle from normal) of 15° , 30° , 45° , and 60° , respectively. Seginer (1975) also

reported a decrease in the shelter distance for oblique flow. However, Karschon (1956), Karschon and Heth (1958), and Franken and Kaps (1957) reported that the shelter distance is not influenced as long as the wind does not vary more than 25° to 50° from the normal. On the other hand, Gorsenin (1946) reported the decrease of the sheltered distance in oblique flows to be less than $\cos(IA)$, but Seginer (1975) observed the decrease rate to be larger than $\cos(IA)$.

Wind Speed Reduction in Oblique Flow

Three factors may contribute to the different effects of oblique flows (compared to normal-incidence flows) on wind-speed reduction in the lee: (1) an increase of total drag due to the longer path through the shelter for oblique flow, (2) a less effective reduction of the component of wind speed parallel to the belt compared to the component perpendicular to the belt (pressure perturbation of the shelter affects only the normal component), (3) rotation of the wind vector as the flow recovers to the undisturbed flow as discussed in above section. The roles of the first two factors in wind-speed reduction are opposite: higher IA causes larger drag which produces larger wind reduction, whereas higher IA creates a larger component of wind parallel to the belt which is less efficiently diminished by the shelter, thereby suppressing the wind-speed reduction. For lower density shelterbelts, the former is more important, resulting in an increase of the maximum wind-speed reduction. However, the latter is more important for higher density shelterbelts, resulting in a decrease of the maximum wind-speed reduction.

Shelter Distance in Oblique Flow

We define the shelter distance (d) as the distance, in units of shelterbelt height, from the belt to the downwind point at which wind speed recovers to 80% of the undisturbed airflow speed as the shelter distance. We further define the shelter distance for approach flow perpendicular to the shelterbelt as the normal shelter distance (d_0), and the shelter distance for oblique flows as an effective oblique shelter distance (d_{IA}). Figure 5 shows the changes of the shelter distance for a medium-dense shelterbelt. The simulated normal shelter distance (d_0) is between 15.5 and 19.5 H at levels below 0.5 H for a medium-dense shelterbelt, in good agreement with all available observations (van Eimern et al 1964, McNaughton 1988, Heisler & DeWalle 1988). The shelter distance decreases with increasing height and IA . The rate of decrease of the shelter distance with IA also diminishes with height. The range of IA s within which the shelter distance is hardly changed increases from 10° at $z=0.1 H$ to 60° at $z=1.0 H$. Therefore, observations made at different levels may lead to different conclusions about the effects of oblique flow on the shelter distance. This may explain the apparently conflicting results of observational studies.

It would be convenient to assume that knowledge of normal shelter distance could be extended to oblique flows. However, we can derive simple relationships between d_0 and d_{IA} only by assuming that (1) the shelterbelt does not affect the horizontal wind direction and that (2) effective density of the shelterbelt does not change with IA . With the further assumption that (3a) the shelterbelt reduces wind speed both parallel to and perpendicular to the belt with the same efficiency, i.e., $\delta U/U_0 = \delta u_1/\delta u_{10} = \delta u_2/u_{20}$, where δ denotes the reduction of wind and its components and subscript 0 denotes the upstream undisturbed wind speed, we can project the

sheltered distance for oblique flow with an incidence angle of IA on the normal line of the shelter and obtain the relationship

$$d_{IA} = d_0 \cos(IA) \quad (15)$$

Alternatively, if we assume that (3b) the shelterbelt reduces only the component of wind perpendicular to the belt and does not affect the component parallel to the belt and the total wind reduction is equal to the reduction of the normal component, the relative reduction of total wind speed is $\delta U/U_0 = \delta u_1/U_0$. Under assumption 3b, local wind direction is changed; but if we still assume the shelterbelt does not affect the horizontal wind direction, we have $\delta U/U_0 = \cos(IA) \delta u_1/u_{10}$ and project the sheltered distance for oblique flow with an incidence angle of IA on the normal line of the shelter and obtain

$$d_{IA} = d_0 \cos^2(IA) \quad (16)$$

Obviously, all above assumptions are unrealistic, but these functional relationships help us to analyze the results.

Influence of Shelterbelt Share

Shelterbelt structure can be characterized by a combination of the internal structure (i.e., porosity) and the external structure (i.e., shape). Intensive studies of shelterbelt porosity have concluded that medium-dense shelterbelts have maximum average wind-speed reduction over the total distance influenced by the shelterbelt. Aeronautical studies of the effects on flow due to shape for solid obstacles show that smooth shapes or streamlined obstacles have significantly

smaller influence on perturbation pressure and flow field (Hoerner 1965). Most investigators suggested that smooth-shaped or streamlined shelterbelts produce smaller wind-reduction than vertical-sided shelterbelts because the resistance for smooth-shaped or streamlined shelterbelts is smaller (e.g., Caborn 1957, 1965, Jensen 1974). However, Woodruff & Zingg (1953) reported that their wind-tunnel results gave only small differences of wind reduction for shelterbelts with very different shapes. They studied many streamlined shapes except, unfortunately, the rectangle shape, which now commonly is suggested for maximum wind reduction.

Wang & Takle (1997a) conducted numerical simulations of shelterbelt shape by using seven common shapes divided into three profile categories: rectangular, triangular, and streamlined. We denote them as H, A, and S, respectively. The triangular and streamlined shapes can be further subdivided. The letter J is used to symbolize a vertical side on the leeward edge of the shelter, and its mirror image, "L", denotes vertical sides on the windward side. "H" represents vertical sides at both edges, and "A" denotes a shelter with symmetrically sloping sides. By considering the general shape profile (first symbol) and symmetry of windward and leeward shape (second symbol) separately, we create seven shapes labeled as HH, AA, AL, AJ, SA, SL, and SJ.

We were surprised to find that although the shapes of shelterbelts were very different and drag forces for different shapes also were significantly different (Wang & Takle 1997a), there were only small differences in wind-speed reduction by shelterbelts of different shapes, even in the region of maximum wind reduction. The effect of shelterbelt shape on shelter distance, for the configurations examined, is negligible. The shelterbelts with windward vertical

sides cause the maximum wind reduction at lower levels. However, at higher levels, shape HH produces the maximum wind reduction, and above the canopy level AL produces the maximum, and AJ produces the minimum increase in wind. Shelterbelt shape significantly affects the perturbed pressure around the shelter(Figure 6). Pressure increases as air approaches the belt, with the AL shape causing the most rapid increase in the pressure and smooth-shaped shelterbelts causing smaller perturbations.

The pressure-loss coefficient or resistance coefficient (C_p), which is defined as mean pressure drop across an obstacle normalized by dynamic pressure or MKE of the undisturbed flow at the height of the top of the obstacle, is often used to characterize the aerodynamics of obstacles (Hoerner 1965). The pressure-loss coefficient is strongly dependent on shelterbelt shape. The triangle-shaped shelterbelt with a windward vertical side (AL) has a 33% larger C_p than the smooth-shaped shelterbelts (Wang & Takle 1997a). Our calculated results are consistent with aerodynamic measurements for solid obstacles, which were summarized by Hoerner (1965) as follows: $C_d=1.00, 1.03, 1.28, 1.2$, and 0.8 for solid shapes AA, AJ, AL, HH, and SA, respectively. It would seem reasonable to infer from pressure-loss coefficients, as many previous investigators have, that shelterbelt shape significantly affects shelter efficiency and that smooth-shaped or streamlined shelterbelts significantly reduce the shelter efficiency. However, as demonstrated in our analysis of numerical simulations and Woodruff and Zingg's (1953) measurements (summarized by Heisler & DeWalle 1988), shelter efficiency is affected very little by shelterbelt shape. An analysis of the momentum budget for different shelterbelt shapes will clarify this apparent discrepancy.

Momentum Budget and Shelter Mechanism

Plate (1971) reviewed several qualitative aspects of shelterbelt aerodynamics. The drag exerted by a shelterbelt on airflow extracts momentum from the air. The effectiveness of a shelter is determined not only by its total drag but also by the distribution of the drag-generated momentum deficit in the sheltered area. The distribution of momentum deficit in the near- and mid-wake directly determines the effective shelter characteristics. Kaiser (1959) obtained an error-function formula for wind speed by assuming that the momentum deficit in the sheltered region is replenished by diffusion of a passive scalar, which follows from the governing equations that the shear stress gradient shapes the mean velocity profile as in homogeneous surface boundary layer. From this theory, the momentum deficit is eliminated and the undisturbed flow is reestablished only by turbulent processes: more turbulence leads to faster recovery to the undisturbed background flow. This theory is somewhat oversimplified, but it is frequently used to understand and interpret experimental data. The wind-reduction profiles illustrated by data from Nageli's (1946) field experiments showed that a medium-dense shelterbelt has the largest shelter effects (van Eimern, et al 1964, Plate 1971, Heisler & DeWalle 1988). This was explained as follows: although a dense shelterbelt by its larger drag produces larger momentum deficit immediately behind the shelter, it also produces more turbulence which may increase downward diffusion of momentum and cause faster recovery of the undisturbed wind speed. However, more recent measurements with artificial barriers in the atmosphere (Hagen & Skidmore 1971) and in wind tunnels (Jensen 1974, Raine & Stevenson 1977) demonstrated that the shelter distance for dense and solid fences is only slight shorter

than that for medium-dense fences, and that the difference is much smaller than that suggested by Nageli's data.

Wilson's (1985) numerical modeling results suggested that the shelter distance increases with increasing density. Evaluation of the root causes of the sheltering effect requires evaluation of the relative contributions of turbulence, pressure gradient, and advection to the wind-speed recovery process in different regions of lee. Failure of early modeling studies to accurately characterize the incremental momentum extraction in the shelter (and the resulting pressure field) precluded in-depth analysis of the momentum budget. Wang & Takle (1997b) reported analyses the momentum budgets for shelterbelt flow and documented the contributions of each term to the recovery process. These results are of more than theoretical interest since they provide information that may be used in the design of new shelterbelt and the interpretation of experimental data.

The shelterbelt extracts momentum from the wind field by means of the drag force, so larger drag would seem to suggest greater wind reduction in the incompressible boundary-layer atmosphere. However, maximum effectiveness of a shelter does not necessarily mean reducing the wind to zero. Rather, the goal is an optimum wind-reduction (i.e., that wind is reduced below the dangerous level) over a maximum distance in a thin air layer near the ground where the crops need protection. Optimizing wind reduction requires understanding of how the momentum deficit created by shelterbelt drag is replenished in the lee and how the perturbed wind recovers to the undisturbed status. It has been suggested that shelterbelt density is the most important factor to determine shelter distance (van Eimern et al 1964, Heisler & DeWalle 1988). We first analyze the change of drag with shelterbelt density and the resulting wind-

reduction and then analyze the processes responsible for wind speed recovery in the lee and evaluate their relative magnitudes at various locations.

Drag Force and Wind-Reduction

Conventional wisdom suggests that, although a dense shelterbelt causes greater wind reduction than its more open counterpart, it allows to a more rapid rate of recovery towards the undisturbed flow. This leads to an optimum shelterbelt density of about 40-50% (van Eimern et al 1964, Heisler & DeWalle 1988) as having the longest sheltered distance. Our simulated results of wind reduction normalized by the undisturbed wind speed (Wang & Takle 1997b) are shown in Figure 7. Several points are noteworthy:

- (1) maximum wind-reduction at $z=0.5 H$ occurs 1-5 H leeward, not immediately behind the belt. The location of the maximum wind-reduction moves toward the belt when the shelterbelt density increases. But this feature is not obvious for loose shelterbelts because the curves for loose shelterbelts have a wide "valley" whereas the curves for dense shelterbelts have a deep "valley".
- (2) maximum wind-reduction increases with increasing shelterbelt density (k_r), however, the downstream extent of the wind-reduction curve does not always increase with increasing shelterbelt density.
- (3) wind-reduction curves for dense and loose shelterbelts cross except for very open shelterbelts which have very small wind-reduction ($k_r=0.1$, corresponding to porosity of 92% according to Hoerner's (1965) formula).

For dense shelterbelts, the windspeed recovers faster in the near lee (0-10 H), and more slowly in the mid lee (10-30 H) than for loose shelterbelts. Therefore, the shelterbelt with $k_r=2.0$ (corresponding to porosity of 50%) extends farthest within the range of 50-80% of u/u_0 . The shelter distance is generally defined as a downstream distance where wind-reduction is at least 20% (van Eimern et al 1964, Heisler & DeWalle 1988). The maximum shelter distance is 16.5 H for shelterbelt with porosity of 50% ($k_r=2.0$), and the shelter distance slightly decreases to 14.5 H when shelterbelt porosity decreases to 10% ($k_r=100$).

There has been speculation in previous reports (van Eimern et al 1964, Heisler & DeWalle 1988), i.e., the effect of advection-turbulent diffusion processes of momentum deficit. However, simple advection-turbulent diffusion processes can not account for the location of maximum wind-reduction. Moreover, Wilson (1985) argued that the turbulence kinetic energy generated near the barriers is at small scales, which contribute little to transport of momentum and dissipate rapidly. Analysis of momentum budget (Wang & Takle 1997b) helps to understand aerodynamic processes and shelter mechanism.

Momentum Budget

For steady state flow outside the shelter, Equation (10) becomes

$$\underbrace{-u \frac{\partial u}{\partial x}}_I - \underbrace{w \frac{\partial u}{\partial z}}_II - \underbrace{\frac{\partial \overline{u'^2}}{\partial x}}_III - \underbrace{\frac{\partial \overline{u'w'}}{\partial z}}_IV - \underbrace{\frac{1}{\rho_0} \frac{\partial p}{\partial x}}_V = 0 \quad (17)$$

where first two terms (I and II) are horizontal and vertical advection of mean horizontal momentum, respectively; the middle two terms (III and IV) are horizontal and vertical turbulent

transport of turbulent horizontal momentum, respectively; and the last term on the left is the pressure gradient. The effective sheltered region (e.g., the region where wind-reduction is at least 20%) is located in the near and middle lees within 30 H of the shelter. Horizontal dependence of terms of the momentum equation at different levels within 30 H leeward are shown in Figure 8 for a medium-dense shelterbelt ($k_t=2.0$, corresponding to porosity of 50%). Values shown in Figure 8 have been normalized by shelterbelt height (H) and mean kinetic energy (MKE_H) of the undisturbed flow at the height of shelterbelt top (MKE_H/H), and hence give the dimensionless acceleration. If a term is positive, it contributes to recovery of the wind to the undisturbed background equilibrium, and negative terms reduce the mean wind.

Analysis of components of the momentum budget at the level corresponding to the middle of the shelter (Figure 8) reveals the following (Wang & Takle 1997):

- (1) Horizontal mean transport (horizontal advection) is positive from 0-5 H leeward, is maximum immediately behind the shelter, goes to zero at the position of the maximum wind-reduction ($x=5 H$), and further decreases to its minimum at $x=10 H$.
- (2) Vertical mean transport (vertical advection) is negative from 0-7 H leeward and positive beyond 7 H with maximum at $x=11 H$, thereby nearly compensating for horizontal mean transport over the whole lee.
- (3) Vertical turbulent transport was thought to be a dominant factor in the recovery of the wind field (van Eimern et al 1964, Heisler & DeWalle 1988); however, Figure 8 suggests that other processes may play roles as large as vertical turbulent transport.
- (4) Horizontal turbulent transport is very small and is negligible in the momentum equation except at higher levels where other terms also are small (Figure 8).

- (5) The pressure gradient has two minima in the lee, one being immediately behind the shelterbelt and the other at $10 H$ leeward. Beyond $17.5 H$ the pressure gradient has a small positive value. This reveals the significant role of the pressure gradient in the middle lee, which is in contrast to prior common understanding that the pressure gradient plays larger role in the near lee (Wilson 1985). However, as summarized in Plate's (1971) Figure 9, measurements showed that static pressure downstream of the shelter is approximately constant over a distance of about $5 H$, and rises very rapidly from 5 - $15 H$, and then slightly overshoots the free-flow pressure after $15 H$. These measurements are in agreement with the simulated results reported in Wang & Takle (1997b).

Momentum transport is very large near the shelter (0 - $2 H$ leeward) and in the middle lee (7 - $20 H$ leeward). Horizontal advection and large values of negative pressure gradient in the middle lee combine to extend the wind-reduction zone much farther on the leeward side than on the windward side of the shelterbelt. Vertical advection is a larger factor than turbulent transport in helping wind-speed recovery in the middle lee (9 - $23 H$).

These simulations clarify the roles of various processes contributing to the wind-speed recovery at different locations. It is clear that perturbed pressure, turbulence, and vertical velocity all play important roles in the recovery of wind speed.

Wang and Takle (1997b) also analyzed the vertical momentum budget and pointed out that the difference between ground and free stream pressure gives rise to a vertical force on the flow that tends to compress streamlines near the ground (the Coanda effect). The Coanda effect is rather weak and is confined only within the near lee (1 - $10 H$) for medium-dense shelterbelts,

but the counter vertical pressure gradient makes vertical velocity reduced to zero in the middle lee (10-30 H). For a dense shelterbelt, horizontal and vertical advection and vertical turbulent transport all play significant roles. Even horizontal turbulent transport, which is always negligible for loose and medium-dense shelterbelts, has a large effect on the momentum budget. Recirculation and the related strong turbulence produce multiple peaks of the vertical momentum budget.

Influence on Evapotranspiration and Surface Energy Partition

Water consumption in irrigated agriculture commonly accounts for 80-85% of all agricultural use, with on-farm efficiencies estimated to be in the range of 10-40%. Because water is becoming the most critical of all resources for food production, and crop production is limited more often by water than anything else, and it is necessary to increase crop water-use efficiency to increase crop yields (Sturrock 1988). Agroforestry has been seen as a possible means of preserving forest resources increasing agricultural production and to reduce deforestation (Von Maydell 1987). Additional advantages of trees in an agricultural landscape, such as for sequestering carbon and offering a potential cash crop for biomass energy production suggest agroforestry practices likely will increase.

Reduced evaporation of soil moisture and reduced transpiration of plants are benefits of shelterbelts in both warm dry and cool wet periods (Gagarin 1949). Preservation of soil moisture may be the main reason for enhanced plant growth and crop yield in wind-sheltered areas. Reductions of evaporation in the lee of windbreaks of between 10 and 40% have been measured by pan and Piche evaporimeters (Bates 1911, Long & Persaud 1988). Miller et al

(1973) reported that during six days of measurements, shelter caused a mean 20% decrease in evapotranspiration. Observations in large-scale shelterbelt networks in China showed regional evaporation was reduced by 14%. Windbreaks and shelterbelts have been suggested as practical means to increase water-use efficiency of sheltered crops (Rosenberg 1967). However, Marshall (1967) noted several exceptions and George (1971) presented data indicating increases in seasonal evapotranspiration by shelter. Dixon & Grace (1984) demonstrated transpiration rates increased with decreasing wind speed. Recent work by Brenner et al (1995) also does not support the hypothesis that water is conserved behind a windbreak. Cleugh (1998) reviews the observed effects of shelters on microclimate and points out the value of modeling studies and the need for extension of a model such as described herein to include effects of temperature, moisture, and CO₂.

Although this problem has been treated by numerous authors owing to its great importance, most measurements of evaporation are limited to the assessment of the capacity for evaporation, and have been carried out with atmometers or evaporimeters. However, plants can close their stomata to reduce transpiration, so we must distinguish between the influence of belts on potential evaporation and evapotranspiration.

The effect of windbreaks on evaporation is complicated by turbulence induced by the barrier, barrier porosity, and availability of water to evaporation sites. Possible consequences of shelter for water use are several and difficult to predict, and may not always be beneficial to crop growth in water-limited environments. Therefore, it is essential to gain an understanding of how evapotranspiration responds to shelter.

Wang et al (1997c,d, 1998a) extended the previously described shelterbelt turbulent flow model to drive *Soil-vegetation-atmosphere Processes*:

$$Flux = \frac{Potential\ difference}{Resistance} \quad (18)$$

Various resistances are defined and determined following Sellers et al (1986). For sensible heat flux (H_s), the potential difference is the temperature difference between the leaf surface or ground and air; for latent heat flux (LE), the potential difference is the vapor pressure difference between the leaf surface or ground and air. These values are connected to soil moisture availability, which is predicted by soil moisture budget equations, in terms of ratio of actual soil water to saturated soil water (m)(Sellers et al 1986, Shen et al 1997). The shelter is simulated as a barrier to the flow and does not participate in the balances of moisture and energy. Processes simulated included the effects of shelterbelts on evapotranspiration and heat flux and their dependence on soil moisture availability and shelterbelt structure as well as atmospheric, soil, and vegetation conditions.

Temporal and spatial variability of evapotranspiration and heat flux as a function of soil moisture availability

For a very dry soil ($m=0.2$, m is moisture availability in soil), shelterbelts cause a decrease in sensible heat flux. This effect extends to 15 H leeward from the belt, and the maximum decrease of H occurs at $x=6$ H. The shelter effect on H_s also change significantly with time, and the maximum effect occurs at $t=1300$ LST. Accordingly, latent heat flux increases in the sheltered zone, and the maximum increase is at $x=6$ H and $t=1100$ LST. Although the

evapotranspiration is very small for dry surface, LE in the sheltered zone is as large as 122% of that in the unsheltered zone.

With increasing soil wetness, the situations begin to change. For $m=0.3$, the increase in LE is still as large as 20%, but the maximum increase of LE occurs at $t=1300$ LST. At $t=0900$ LST, a significant decrease in LE occurs in 0-17 H lee with the maximum decrease at $x=6$ H, and at $t=1700$ LST, a slight smaller LE occurs in the sheltered zone. H_s decreases at noon but increases in early morning and late afternoon in sheltered zone for $m=0.3$. For $m=0.4$ (figures omitted), the situation is similar to that of $m=0.3$, but the noontime increase in LE is very small, and the decreases in LE become significant both in the morning and afternoon. Maximum decreases in LE are as large as 18% and 25% at $t=1000$ LST and $t=1530$ LST.

For $m=0.5$ (Figure 9), the noontime increase in LE disappears, and evapotranspiration is reduced all during the daytime in the near lee. However, in the middle lee (15-23 H), LE increases in the afternoon because of wake turbulence. Accordingly, H_s increases in the sheltered zone and it is more obvious for $m=0.6$ case, where a center of large H forms at $x=6$ H around noon. Evapotranspiration during 0900-1300 LST is further reduced with the maximum reduction as large as 40% (figures omitted). For $m=0.8$, LE is significantly reduced during the daytime in the sheltered zone extending to 25 H leeward side, and the maximum reduction of 50% occurs in the early afternoon at 4-7 H leeward, where reductions of wind and turbulence also reach their maxima (Wang & Takle 1995b). By contrast, H_s increases in the sheltered zone and forms a center. The situation is nearly identical for further increasing soil wetness (figures omitted). The maximum evapotranspiration occurs at 0930, 1030, 1230 for $m=0.3$, 0.5, and 0.8, respectively. Crop stomata are closed when water is limited. For dry soil,

the stomata are closed earlier than for moderately wet soil. For wet soil, the stomata are not closed and therefore the evapotranspiration reaches its maximum during the maximum radiation and temperature.

Influence of shelterbelt structure on evapotranspiration and heat flux

Turbulence and flow structures near shelterbelts are very dependent on shelterbelt density, and wind-shelter effect reaches its maximum for medium-dense shelters (Wang & Shen 1989, Wang 1991a,b, 1992, Wang & Takle, 1994a,b, 1995a,b,c,d, 1996a,b,c, 1997a,b). Wang et al (1997d, 1998a) demonstrated the dependence of the shelter evapotranspiration effect on shelterbelt density. Very dense shelterbelts cause complicated changes in both LE and H_s in the near lee, which obviously relate to recirculations and separation of streamlines (Wang & Takle 1995b). In the recirculation zone, LE is significantly larger and a center is formed at $x=3 H$ around noon, which diminishes the reduction of evapotranspiration by the shelter. H_s is enhanced with two centers located on $x=1.5$ and $5.5 H$. For very loose shelterbelts, LE is still reduced, but the effects of shelterbelts on both H_s and LE is quite small.

Agricultural practices such as shelterbelts networks and strip-cropping can significantly affect evapotranspiration. However, the effects are not straightforward, and our simulations show the complicated temporal and spatial variability of both latent and sensible heat fluxes. Aerodynamic shelter effects and their interactions with energy, water and mass transfer in soil-vegetation-atmosphere system cause significant variability and are controlled, to great extent, by soil moisture availability. These simulations demonstrate that soil moisture controls not only the magnitude of evapotranspiration but also the direction of evapotranspiration-shelter effect:

shelterbelts decrease evapotranspiration for wet soil, increase it for dry soil, and increase it around noon but decrease it in the morning and afternoon for moderately wet soil. The model also illustrated that recirculation significantly diminished the reduction of evapotranspiration for very dense shelterbelts and that very loose shelterbelts exerted small effects on the partition of energy. Therefore, medium-dense shelterbelts also have maximum evapotranspiration-shelter efficiency. Regional scale agricultural and forestry practices may significantly affect water and energy cycles and may be used to advantage to suppress negative impacts of climate change.

High-Performance Computing - Parallelization and Cluster Computing

Fast computers have stimulated the rapid growth of a new way of doing science. The two broad classical branches of theoretical science and experimental science have been joined by computational science. Computational scientists simulate on supercomputers phenomena too complex to be reliably predicted by theory and too dangerous or expensive to be reproduced in the laboratory. With the rapid development of microcomputers and fast networking with high-speed switches, parallel processing on distributed networks of workstations has emerged as a cost-effective method of high-performance computing – cluster computing. The shelterbelt turbulent flow model systems has been parallelized and run with very good load balance on a cluster of 128 processors network of workstations by using both MPI (Message-Passing Interface) and PVM (Parallel Virtual Machine) techniques (Wang et al 1998, 1999).

Analysis of the computational demands of the code revealed that most of computing time was consumed, not in the tri-diagonal equation solver, but in solving non-linear terms and

computing dynamic pressure perturbation. For a porous shelterbelt, the computation time of dynamic pressure perturbation accounts for 66% total computation time; however, for a dense shelterbelt, the computation time of dynamic pressure perturbation accounts for as high as 91.4% total computation time (Tables omitted).

Functional Decomposition Parallel Programming (FDPP)

Functional parallelism computations allocates separate tasks or functions to be executed simultaneously on different processors to achieve higher speed. Functions outside the time-step iteration loop are not worth parallelizing because they execute only once for the whole run but must process large volumes of data. The root processor was allocated all these functions and transit between time steps, while other six functions were put on 6 different machines. These functions take inputs from the previous time-step values, and run independently within the same time step. They are synchronized at the end of completion of each time step so that for every time step, different functions use the same previous time-step values as inputs. At the completion of the function for each time step, the processor will broadcast the updated values to all other processors so that other processors can use the newest values computed by other processors. For 6 of 7 nodes, the communication overhead took more than 97% of CPU time and the speedup was less than 1 for the parallel code. The major portion of the computation time is consumed by calculating dynamic pressure perturbation. Computing any of the model variables requires information on other variables. Therefore, there are $O(n^2)$ message passing. Moreover, the program should be synchronized at the completion of each function to the same starting point so that all functions can get the same time-step values as new initial values. The

computation of pressure is time-consuming and leaves other processors idle, waiting for the completion of the pressure computation. The average load balance ratio is only 51%. The parallel time is determined by the maximum time of the most time-consuming function computing plus communication overhead. FDPP evidently is not a good strategy for this kind on program.

Domain Decomposition Parallel Programming (DDPP)

Domain decomposition refers to spatially partitioning the computational domain. The domain size of $n \times n$ is sliced it into m small regions, each region having size $n/m \times n$. Each region is assigned to a processor, thereby reducing computation load and enhancing performance; however, this region needs to exchange information with its neighbors, and communication load increases, with additional message complexity of $O(2n)$. Therefore, there are tradeoffs also for domain decomposition, and the final performance depends on the relative effects of both factors. Figure 10 shows the changes of speedup ratio with number of processors and domain size for the shelterbelt model. The curve with squares is for the domain size of $(128+2) \times (64+2)$, and the curve with solid circles is for the domain size of $(256+2) \times (128+2)$. As shown from the figure, speedup increases with increasing number of processors. For small number of processors, the curves are close to the ideal line; but, with the increasing number of processors, the curves depart from the ideal line, and the performance gains slow down, especially for small domain size. This is the result of tradeoffs between reduced computation load and increased communication load after the domain is decomposed. When the number of processors is small or the domain is large, the reduced computation dominates. When the number of processors is

large, each processor processes only a few grids, the performance gains are limited, but the increased communications dominate. We achieved very good load balancing (>0.90 , table omitted) with domain decomposition parallelism. The load balance ratio improves slightly with large number of processors.

Summary and Future Applications

Takle and Kao (1998) have outlined a conceptual model for linking physical processes over many spatial scales in simulating plants, soil, and climate in fully interacting natural systems. The capability of simulating local and incremental momentum extraction within a vegetative canopy in a non-hydrostatic model of the non-neutral atmospheric boundary layer provides an essential link for simulating physical interactions in heterogeneous plant ecosystems. Correct representation of drag produces the proper pressure fields in and around the vegetative barrier, which then enables mean and turbulent flow fields to be properly represented. An understanding of the turbulent motions produced by a single component of a plant community permits quantification of advective effects that are critical to such interactions. Extensions to non-neutral flow bring complications relating to incremental light interception by the barrier, which are yet to be resolved in a boundary layer model. However, this is of secondary importance to the advances that now are possible in simulating energy budgets, including soil moisture, over surfaces represented by heterogeneous plant communities.

Applications now possible using the techniques described in this paper include transport of particles and passive scalars through heterogeneous vegetation. Impact of vegetation on particulate transport, movement of pesticides, dispersion of agricultural or industrial odors, drift

patterns of aerial sprays, and movement of spores or pollen all can be simulated with additional precision. Simulations of pollination processes for crop breeding involving conventional or genetically altered materials can be done in advance of, or perhaps even in place of, more costly, time consuming, and controversial field experiments. Agricultural techniques for conserving soil moisture now can be assessed numerically given the ability to more accurately simulate heterogeneous evaporation and crop transpiration due to a knowledge of lateral as well as vertical mean and turbulent transport processes in such environments. Evaluating alternative management strategies may be done for manipulating microclimates in agricultural, horticultural, or forest environments to create more favorable environments for plant growth and yield. The ability to represent large pressure gradients in two-phase systems renders such a model applicable to simulation of regions of strong convection as in forest fires (Clark 1996).

The impact of global climate change on plants, animals, and humans is experienced at the local, rather than regional or global scale. Dynamical downscaling of global climate change to regional scales has been achieved by use of regional climate models (Wang et al 1997e). A microscale climate model with ability to simulate non-neutral effects completes the downscaling hierarchy and allows impacts of global changes to be interpreted in local agricultural, forest, and natural ecosystems.

Finally, models, such as is described in this paper are readily adaptable for execution on the latest generation of parallel computers, thereby promoting rapid advances in understanding managed and natural heterogeneous plant systems.

Acknowledgments

This research was supported by the US Department of Agriculture Grant 96351083892, the US Department of Energy NIGEC Grant DE-FC 0390ER61010, and the NRI Competitive Grant 93-37101-8954. The final work was done on the ALICE network of workstations in the Scalable Computing Laboratory. The ALICE computer facility was maintained by the USDOE Ames Laboratory.

Literature Cited

- Baines WD, Peterson EG. 1951. An investigation of flow through screens. *Trans. Am. Soc. Mech. Eng.* 73:467-80
- Bates CG. 1911. Windbreak - their influence and value. *Bull. Forest Service*, US Dept. Agric., 1-22
- Bonan GB. 1996. *A land surface model (LSM version 1.0) for ecological, hydrological, and atmospheric studies: technical description and user's guide*. NCAR Technical Note NCAR/TN-417+STR. National Center for Atmospheric Research, Boulder, Colorado. 150 pp.
- Bradley EF, Mulhearn PJ. 1983. Development of velocity and shear stress distributions in the wake of a porous shelter fence. *J. Wind Eng. Ind. Aerodyn.* 15:145-56
- Brenner AJ, Jarvis PG, van der Beldt RJ. 1995. Windbreak-crop interactions in the Sahel. 2. Growth response of millet in shelter. *Agric. For. Meteorol.*, 75: 235-62
- Brinmann K, Kaiser H. 1955. Maize strips as windbreaks, meteorological investigations and yield. *Zeitschr. Acker-u. Pflanzenbau*, 99:321-34

- Caborn JM. 1957. Shelterbelts and microclimate. *For. Comm. Bull. (Edinburgh)*, no.29, 129pp.
- Caborn JM. 1965. *Shelterbelts and Windbreaks*. Faber and Faber, London. 288 pp.
- Castro IP. 1971. Wake characteristics of two-dimensional perforated plates normal to an airstream. *J. Fluid Mech.* 46:599-609
- Chorin AJ. 1968. Numerical solution of the Navier-Stokes equations. *Math. Comp.* 23:341-54
- Clark T L, Jenkins MA, Coen J, Packham D. 1996. A coupled atmosphere-fire model: Convective feedback on fire-line dynamics. *J. Appl. Meteorol.* 35:875-901
- Cleugh HA. 1998. Effects of windbreaks on air flow, microclimate and crop yields. *Agroforestry Systems* 41:55-84
- Counihan J, Hunt JCR, Jackson PS. 1974. Wakes behind two-dimensional surface obstacles in turbulent boundary layers. *J. Fluid Mech.* 64: 529-63
- Dixon M, Grace J. 1984. Effect of wind on the transpiration of young trees. *Ann. Bot.*, 53: 811-19
- Gagarin E. 1949. Planting trees for protecting fields in Russia. *Forstwiss Centralblatt.* 68:571-602
- Finnigan JJ. 1985. Turbulent transport in flexible plant canopies. In *The Forest-Atmosphere Interaction*, eds. BA Hutchison, BB Hicks. Dordrecht / Boston / Lancaster: D. Reidel. pp.443-80
- Finnigan JJ, Einaudi F. 1981. The interaction between an internal gravity wave and the planetary boundary layer. Part II. Effect of the wave on the turbulence structure. *Quart. J. R. Meteorol. Soc.* 107:807-32

- Franken E, Kaps E. 1957. Investigations on shelterbelts in the Emsland. *Berichte Deutscher Wetterd.* 5: 33-37
- Gorsenin NM. 1946. Principles of planting forest shelterbelts on arable slopes. *Nauchnyi otcet vniaini, Sel'hozgiz.*
- Hagen LJ, Skidmore EL. 1971. Turbulent velocity fluctuations and vertical flow as affected by windbreak porosity. *Trans. ASAE.* 1971: 634-37
- Hagen LJ, Skidmore EL, Miller PL, Kipp JE. 1981. Simulation of effect of wind barriers on airflow. *Trans. ASAE.* 24:1002-8
- Heisler GM, Dewalle DR. 1988. Effects of windbreak structure on wind flow. *Agriculture, Ecosystems and Environment.* 22/23:41-69
- Hoerner SF. 1965. *Fluid Dynamic Drag.* Library of Congress Catalog Card Number 64-19666.
- Jacobs AFG. 1984. The flow around a thin closed fence. *Boundary-Layer Meteorol.* 28:317-28
- Jensen M. 1974. The Aerodynamics of Shelter. In *FAO Report on the FAO/DANIDA Inter-regional Training Center on Heathland and Sand Dune Afforestation*, FAO/DEN/TF. 123pp.
- Kaiser H. 1959. Die stromung an windschutzstreifen. *Ber. Deut. Wetterdienstes.* 7:1-36
- Karschon R. 1956. Les eucalyptus et la protection des cultures agricoles. Fin. Rep. World Eucalypt. Conf., Rome.
- Karschon R, Heth D. 1958. Wind speed, wind-borne salt and agricultural crops as affected by windbreaks. *Lay-Yaaran*, 8: 8-13 & 38-42
- Lawrence EN. 1955. Effects of a windbreak on the speed and direction of wind. *Meteorol. Mag.* 84: 244-51

- Li Z, Lin JD, Miller DR. 1989. Air flow over and through a forest edge: a steady-state numerical simulation. *Boundary-Layer Meteorol.* 46:333-54
- Long SP, Persaud N. 1988. Influence of neem (*Azadirachta indica*) windbreaks on millet yield, microclimate, and water use in Niger, West Africa. *Challenges in dryland agriculture -- a global perspective*, eds. PW Unger, TV Sneed, JR Jordan, R Jensen. Amarillo, TX: Texas Agricultural Experimental Station. Pp313-14.
- Marshall JK. 1967. The influence of shelter on the productivity of grasslands and field crops. *Field Crop Abstr.* 20:1-14
- McNaughton KG. 1988. Effects of windbreaks on turbulent transport and microclimate. *Agriculture, Ecosystems and Environment.* 22/23:17-39
- Mellor GL, Yamada T. 1974. A hierarchy of turbulence closure models for planetary boundary layers. *J. Atmos. Sci.* 31:1791-1806
- Mellor GL, Yamada T. 1982. Development of a turbulent closure model for geophysical fluid problems. *Rev. Geophys. Space Sci.* 20:851-75
- Meyers T, Paw U KT. 1986. Testing of a higher-order closure model for modeling airflow within and above plant canopies. *Boundary-Layer Meteorol.* 37:297-311
- Miller DR, Lin JD, Lu ZN. 1991. Air flow across an alpine forest clearing: a model and field measurements. *Agricultural and Forest Meteorol.* 56:209-25
- Miller DR, Rosenberg NJ, Bagley WT. 1973. Soybean water use in the shelter of a slat-fence windbreak. *Agric. Meteorol.* 11:405-18
- Nageli W. 1946. Further investigation of the wind conditions in the range of shelterbelts. *Mitt. Schweiz. Anst. Forstl. Versuchswesen.* 24:660-737

- Naegeli W. 1954: The braking effect of the wind by a large forest. A contribution to the problem of the width of shelterbelts. *Berichte 11*, Kongress Intern. Union Forestl. Forschungsanstalten, Rome, Firenze.
- Naot O, Mahrer Y. 1991. Two-dimensional microclimate distribution within and above a crop canopy in an arid environment: modeling and observational studies. *Boundary-Layer Meteorol.* 56:223-44
- Nord M. 1991. Shelter effects of vegetation belts - results of field measurements. *Boundary-Layer Meteorol.* 54:363-85
- Ogawa Y, Diosey PG. 1980. Surface roughness and thermal stratification effects on the flow behind a two-dimensional fence: II. a wind tunnel study and similarity considerations. *Atmos. Environ.* 14:1309-20
- Paegle J, Zdunkowski WG, Welch RM. 1976. Implicit differencing of predictive equations of the boundary layer. *Mon. Wea. Rev.* 104:1321-24
- Perera MDAS. 1981. Shelter behind two-dimensional solid and porous fences. *J. Wind Eng. Ind. Aerodyn.* 8:93-104
- Plate EJ. 1971. The aerodynamics of shelterbelts. *Agricultural Meteorol.* 8:203-22
- Raine JK, Stevenson DC. 1977. Wind protection by model fences in a simulated atmospheric boundary layer. *J. Indust. Aerodyn.* 2:159-80
- Raupach MR, Coppin PA, Legg BJ. 1986. Experiments on scalar dispersion within a model plant canopy. Part I: the turbulence structure. *Boundary-Layer Meteorol.* 35:21-52
- Raupach MR, Shaw RH. 1982. Averaging procedures for flow within vegetation canopies. *Boundary-Layer Meteorol.* 22:79-90

- Richards PJ, Kay EF, Russell D, Wilson GRC. 1984. Porous artificial windbreaks in oblique winds. *Paper 67/84 for IPENZ Conf.*, Hastings, New Zealand, 10pp.
- Rosberg NJ. 1967. The influence and implications of windbreaks on agriculture in dry regions. *Ground level climatology*, RH Shaw, ed. Am. Assoc. Advan. Sci., Symposium, 327-49.
- Rosenberg NJ. 1979. Windbreaks for reducing moisture stress. In *Modification of Aerial Environment of Plants*, eds. BJ Barfield, JF Gerber. ASAE, St Joseph, pp.538
- Seginer I. 1975. Flow around a windbreak in oblique wind. *Boundary-Layer Meteorol.* 9:133-41
- Sellers PJ, Mintz Y, Sud YC, Dalcher A. 1986. A simple biosphere model (SiB) for use within general circulation models. *J. Atmos. Sci.* 43:505-31
- Shen J, Takle ES, Arritt RW, Wang H. 1997. A simulation study of the relationship among surface heterogeneity, heat fluxes, and boundary-layer turbulence structure. 12th Symposium on Boundary Layers and Turbulence, Amer. Meteor. Soc. Vancouver. 569-70.
- Sturrock JW. 1969. Aerodynamics studies of shelterbelts in Nealand-1: low to medium height shelterbelts in Mid-Canterbury. *N.Z.J.Sci.* 12:754-76
- Sturrock JW. 1972. Aerodynamics studies of shelterbelts in Nealand-2: medium height to tall shelterbelts in Mid-Canterbury. *N. Z. J. Sci.* 15:113-40
- Sturrock JW. 1988. Shelter and its management and promotion. *Agric. Ecosys. Environ.*, 22/23: 17-39
- Takahashi H. 1978. Wind tunnel test on the effect of width of windbreaks on the wind speed distribution in leeward. *J. Agr. Meteorol. (Japan)* 33:183-87

- Takle ES, Kao DT. 1998. A synthesis of models for describing multi-scale interactions within natural systems. In *1998 Conference on Mission Earth: Modeling and Simulation of the Earth System*, eds. A Sydow, JY Yu. Society for Computer Simulation International, San Diego, CA. 29-34.
- Takle ES, Wang H, Schmidt RA, Brandle JR, Litvina IV, Jairell RL. 1997. Pressure perturbations around shelterbelts: Measurements and model results. 12th Symposium on Boundary Layers and Turbulence, Amer. Meteor. Soc. Vancouver. 563-64.
- Takle ES, Wang H, Brandle JR, Zhou X, Litvina IV. 1999. Numerical simulations and measurements of effects of wind direction and shelter structure on shelterbelt efficiency. Sixth Conference on Agroforestry in North America. Hot Springs AK
- Taylor PA. 1988. Turbulent wakes in the atmospheric boundary layer. In *Flow and Transport in the Natural Environment: Advances and Applications*, eds. WL Steffen, OT Denmead. Berlin: Springer-Verlag. 270-92.
- Thom AS. 1975. Momentum, mass and heat exchange of plant communities. *Vegetation and the Atmosphere*. Academic. 1:1-278
- Van Eimern J, Karschon R, Razumova LA, Robertson GW. 1964. *Windbreaks and Shelterbelts*. World Meteorological Organization Technical Note No.59, pp.188
- Von Maydell HJ. 1987. Agroforestry in the dry zones of Africa: past, present and future. *Agroforestry, a decade of development*, eds. HA Steppeler, PKR Nair. ICRAF, Nairobi, 89-116.
- Wang H, Shen J. 1989. A two-dimensional numerical study of the wind sheltering effects of shelterbelts. *Acta Meteorologica Sinica* 3:498-505

Wang H. 1991a. The effects of shelterbelts on the atmospheric turbulent exchange coefficient.

Acta Geographica Sinica 46:107-14

Wang H. 1991b. A numerical simulation of the wind sheltering effects of multiple parallel

shelterbelts. *Journal of Nanjing University* 481-88

Wang H. 1992. On methods for the study about the protection effects of shelterbelts.

Promoting Agriculture through the Progress of Science and Technology, Eastern China Agricultural Association, Ed., Jiangsu Sci.&Tech. Press, 369-72.

Wang H, Takle ES. 1994a. Interaction of synoptic flow and mesoscale circulation as a function

of synoptic windspeed.. Sixth Conference on Mesoscale Processes, Portland, Amer. Meteor. Soc., J36-J39.

Wang H, Takle ES. 1994b. Mesoscale and boundary-layer flows over inhomogeneous surfaces

consisting of porous obstacles. Sixth Conference on Mesoscale Processes, Portland, Amer. Meteor. Soc., 262-65.

Wang H, Takle ES. 1995a. Boundary-layer flow and turbulence near porous obstacles. I.

Derivation of a general equation set for a porous medium. *Boundary-Layer Meteorol.* 74:73-88

Wang H, Takle ES. 1995b, A numerical simulation of boundary-layer flows near shelterbelts.

Boundary-Layer Meteorol. 75:141-73

Wang H, Takle ES. 1995c. Equations for mean and turbulent flow through and over porous

obstacles. Eleventh Symposium on Boundary Layers and Turbulence, Charlotte. Amer. Meteor. Soc.

- Wang H, Takle ES, 1995d: Simulations of mean and turbulent properties of oblique flows near agricultural shelterbelts. Eleventh Symposium on Boundary Layers and Turbulence, Charlotte. Amer. Meteor. Soc.
- Wang H, Takle ES. 1996a. On three-dimensionality of shelterbelt structure and its influences on shelter effects. *Boundary-Layer Meteorol.* 79:83-105
- Wang H, Takle ES. 1996b. On shelter efficiency of shelterbelts in oblique wind. *Agric. Forest Meteorol.* 81:95-117
- Wang H, Takle ES. 1996c. Numerical simulations of shelterbelt effects on wind direction. *Journal of Applied Meteorol.* 34:2206-19
- Wang H, Takle ES. 1996d. Influences of shelterbelts on turbulent flow and shelter mechanism. *22nd Conference on Agricultural and Forest Meteorology with Symposium on Fire and Forest Meteorology* (8.12), Atlanta, Georgia, 28 January - 2 February 1996.
- Wang H, Takle ES. 1996e. Modeling the evapotranspiration and energy partition of inhomogeneous agroecosystems. *22nd Conference on Agricultural and Forest Meteorology*, Atlanta, GA, Amer. Meteor. Soc
- Wang H, Takle ES. 1997a. Model-simulated influences of shelterbelt shape on wind-sheltering efficiency. *Journal of Applied Meteorol.* 36:695-704
- Wang H, Takle ES. 1997b. Momentum budget and shelter mechanism of boundary-layer flow near a shelterbelt. *Boundary-Layer Meteorol.* 82:417-35
- Wang H, Takle ES, Shen J. 1997c. Radiative characteristics of an agroforestry ecosystem. *Ninth Conference on Atmospheric Radiation*. Amer. Meteorol. Soc., 2-7 February, Long Beach, CA. 308-309

- Wang H, Shen J, Takle ES. 1997d. Influences of agroforestry ecosystem on evapotranspiration and soil moisture. 13th Conference on Hydrology. Amer. Meteorol. Soc., 2-7 February, Long Beach, CA. 360-61
- Wang H, Shen J, Takle ES. 1997e. High-resolution regional climate simulations using RegCM2 with different scale couplings of soil, vegetation, and atmospheric boundary-layer processes. Preprints, 12th Symposium on Boundary Layers and Turbulence, Amer. Meteor. Soc. Vancouver. 573-74
- Wang H, Prabhu GM, Takle ES. 1998. Parallelization a very-high-resolution climate model using clusters of workstations with PVM and performance and load balance analyses. Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA'98), CSREA Press. 1:1762-65
- Wang H, Takle ES. 1998a. Agroforestry shelterbelts and its influence on evapotranspiration. ASAE Annual International Meeting, Orlando. 24 pp.
- Wang H, Takle ES. 1998b. Agricultural shelterbelt's protection functions. ASAE Annual International Meeting, Orlando.
- Wang H, Prabhu GM, Takle ES, Todi R. 1999. Implementation and performance evaluation for a computation-intensive climate simulation application. Proceedings of the international conference on parallel computing (parco99). Pp1-8.
- Whitaker S. 1973. The transport equations for multi-phase system. *Chem. Eng. Sci.* 28:139-147
- Wilson NR, Shaw RH. 1977. A higher order closure model for canopy flow. *J. Applied Meteorol.* 16:1197-205

- Wilson JD. 1985. Numerical studies of flow through a windbreak. *J. Wind Eng. Ind. Aerodyn.* 21:119-54
- Woodruff NP, Zingg AW. 1953. Wind tunnel studies of shelterbelt models. *J. For.* 53:173-78
- Woodruff NP, Read RA, Chepil WS. 1959. Influence of a field windbreak on Summer wind movement and air temperature. *Kansas Agric. Exp. Stn. Tech. Bull.* No. 100, 24pp.
- Yamada T. 1982. A numerical model study of turbulent airflow in and above a forest canopy. *J. Meteor. Soc. Japan* 60:438-54
- Yamada T, Mellor GL. 1975. A simulation of the wangara atmospheric boundary layer data. *J. Atmos. Sci.* 32:2309-29
- Zhu T (ed). 1992. *The Study of Ecological Effects and Physical Characteristics in the Boundary Layer of Protective Forest System.* Chinese Meteorological Press, pp.1-266

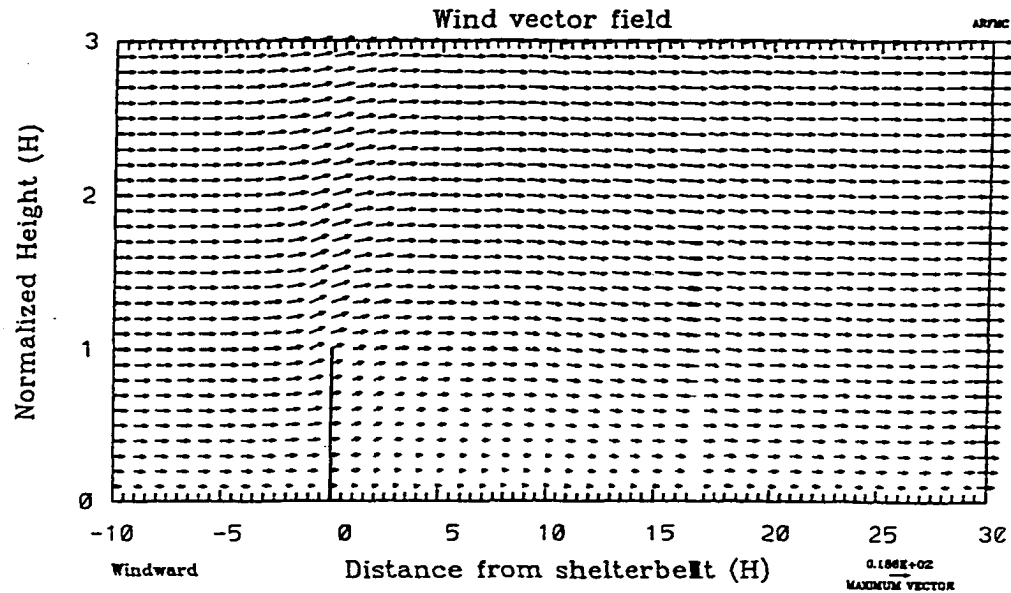


Figure 1a. Flow pattern (unseparated flow) for medium-dense shelterbelt or windbreak

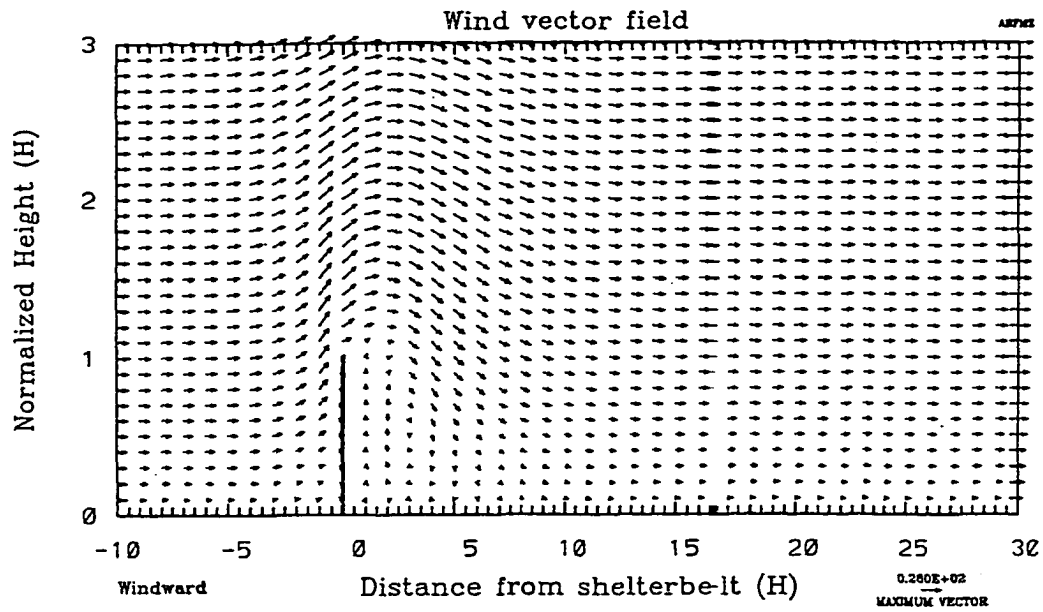


Figure 1b. Flow pattern (separated flow) for dense shelterbelt or windbreak

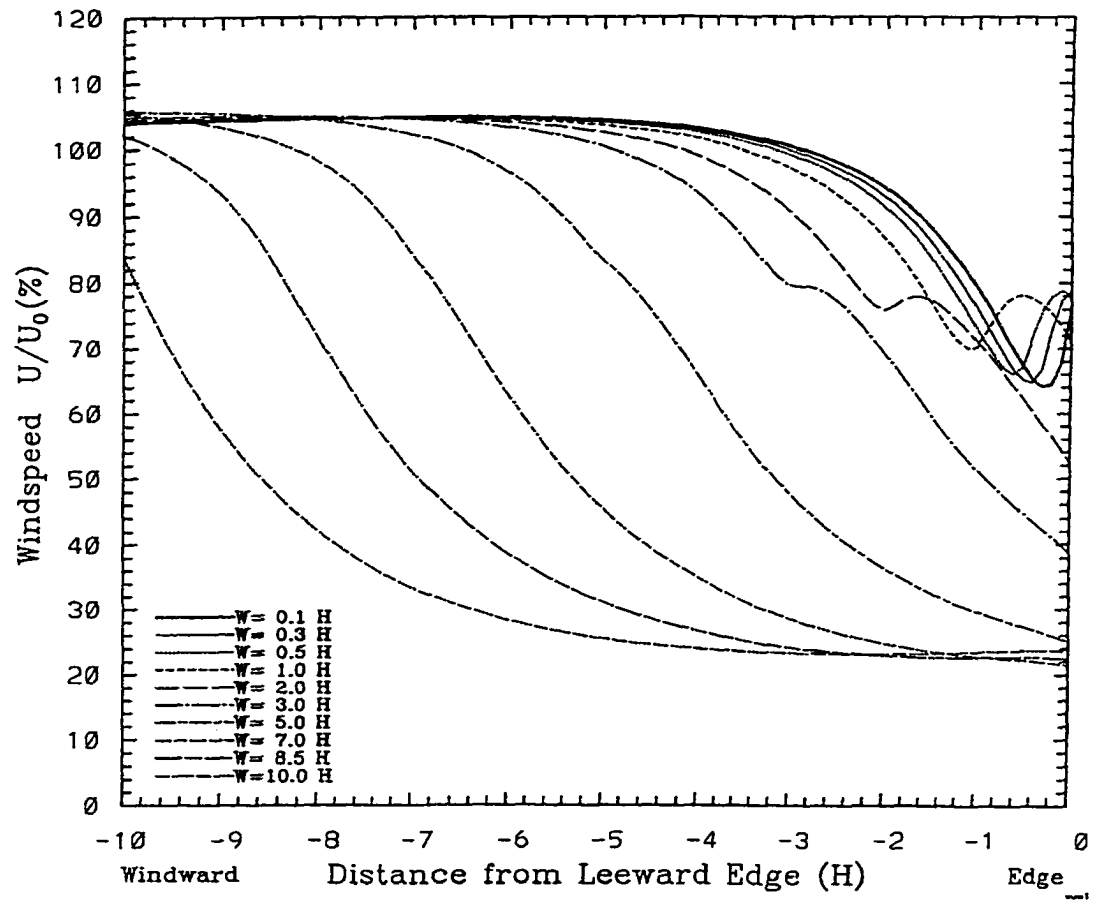


Figure 2. Wind speed within shelterbelts of different widths

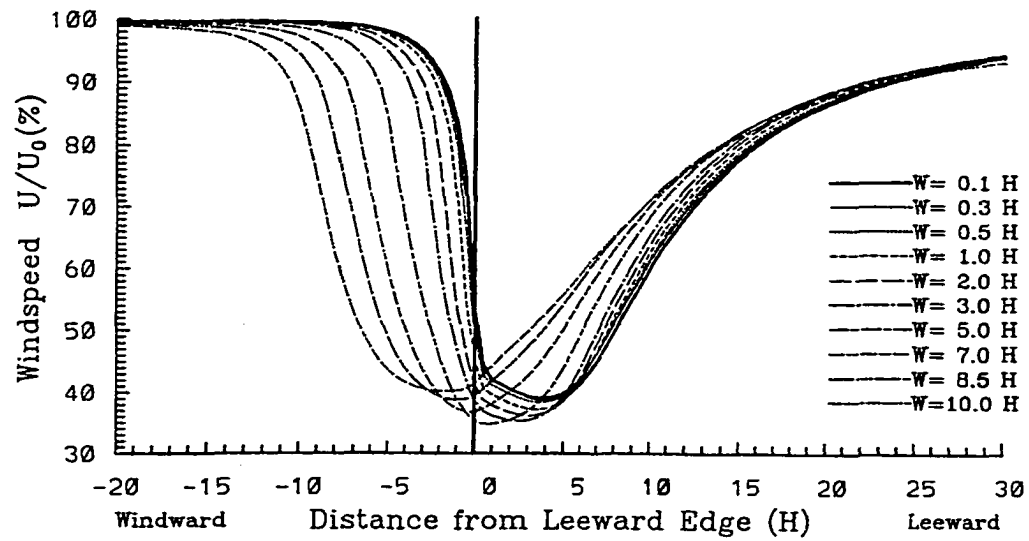


Figure 3. Horizontal profiles of wind speed around shelterbelts

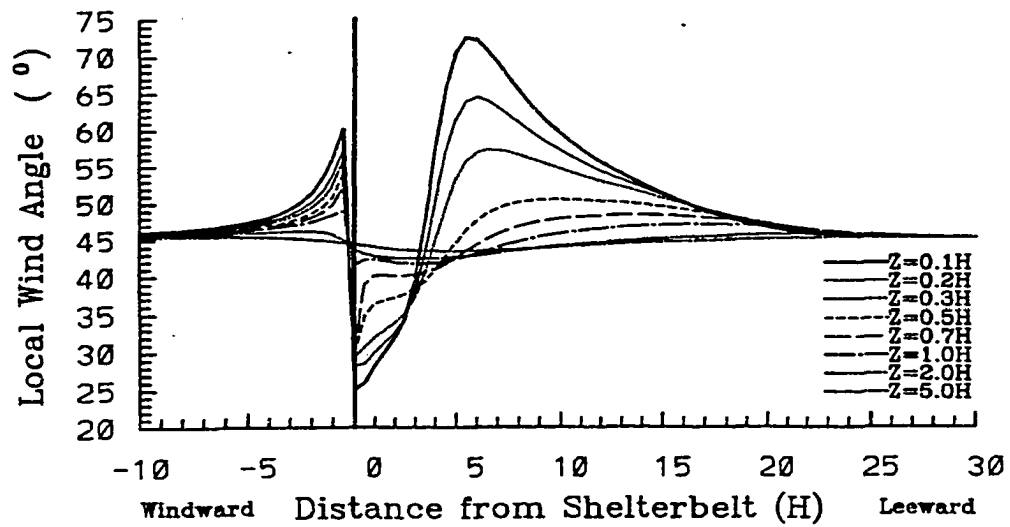


Figure 4. Wind direction rotation across shelterbelts

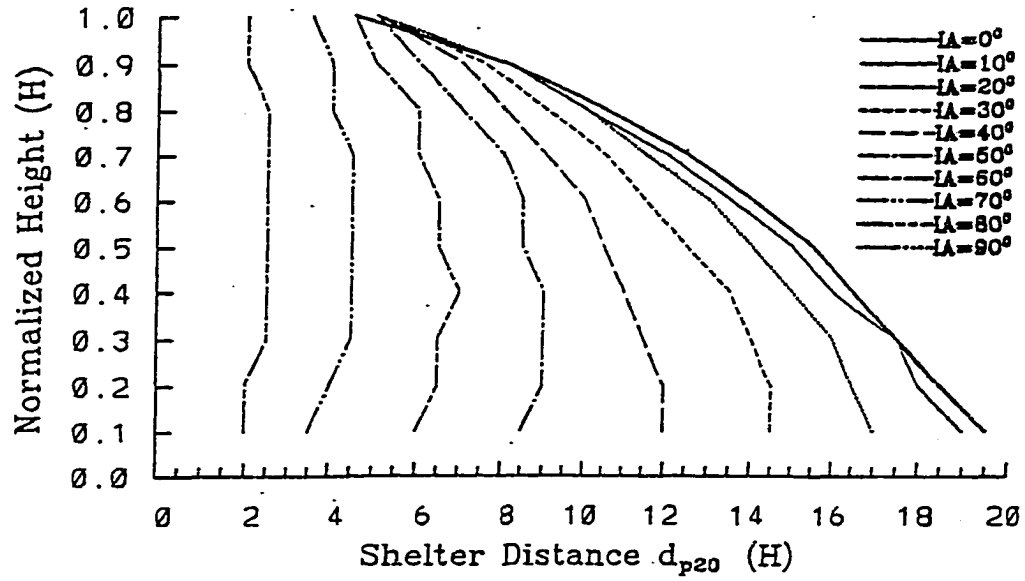


Figure 5. Changes of shelter distance in oblique flow

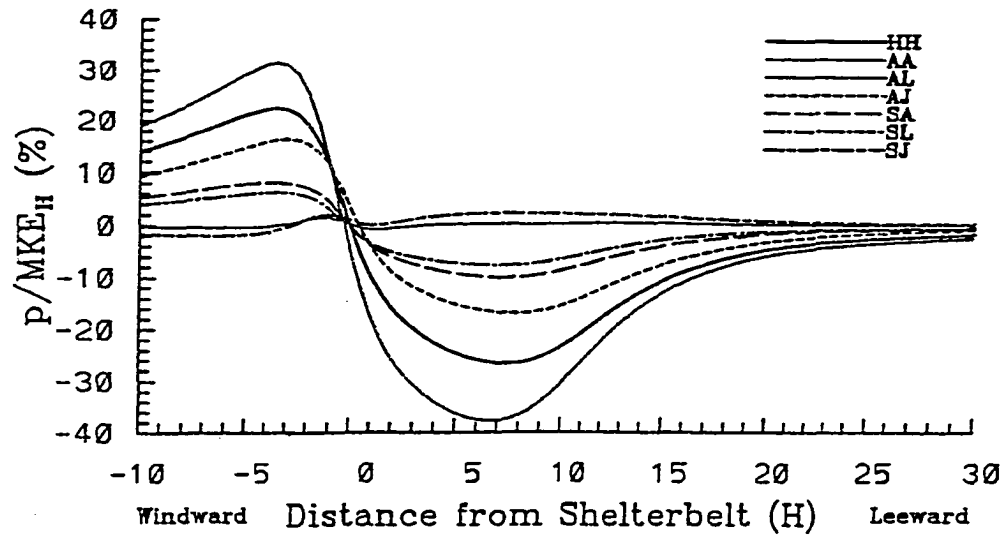


Figure 6. Perturbed pressure around shelterbelts of different shapes

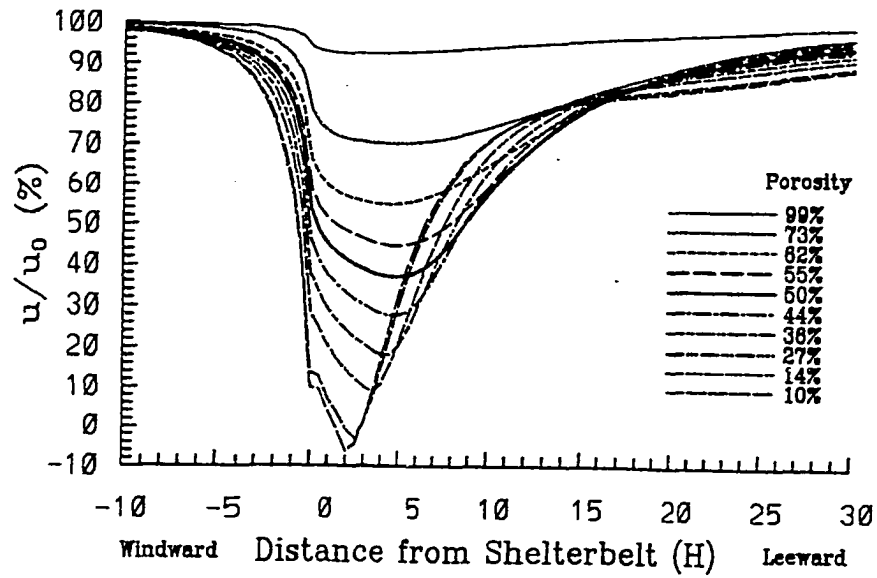


Figure 7. Wind reduction for shelterbelts of different densities

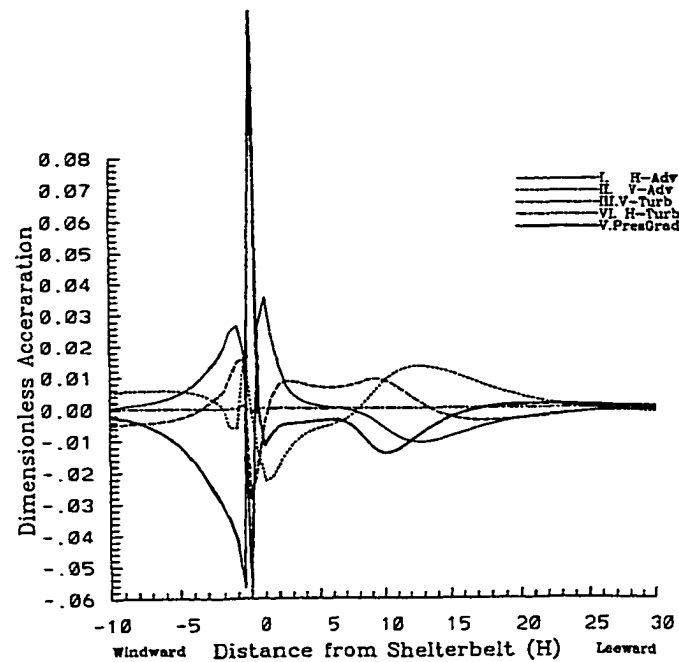


Figure 8. Momentum budget for flow through shelterbelt

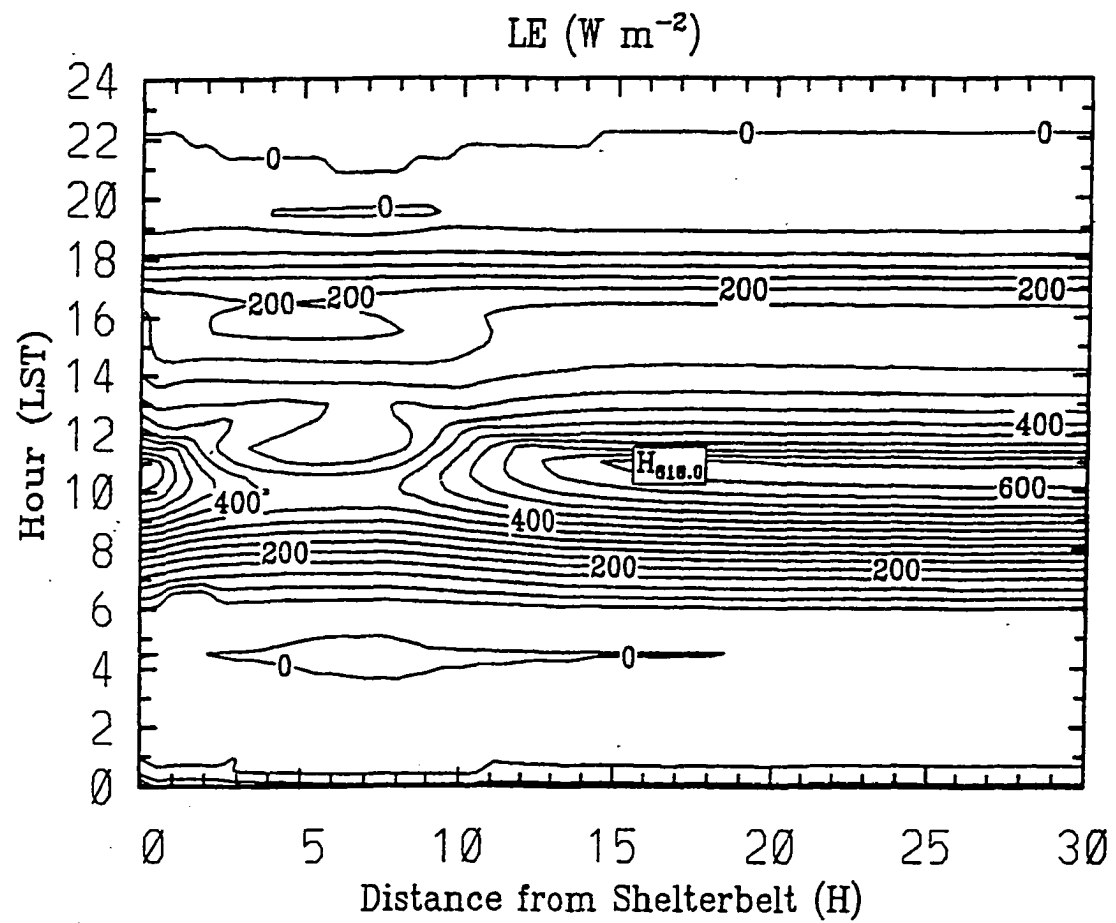


Figure 9. Evapotranspiration as affected by shelterbelt

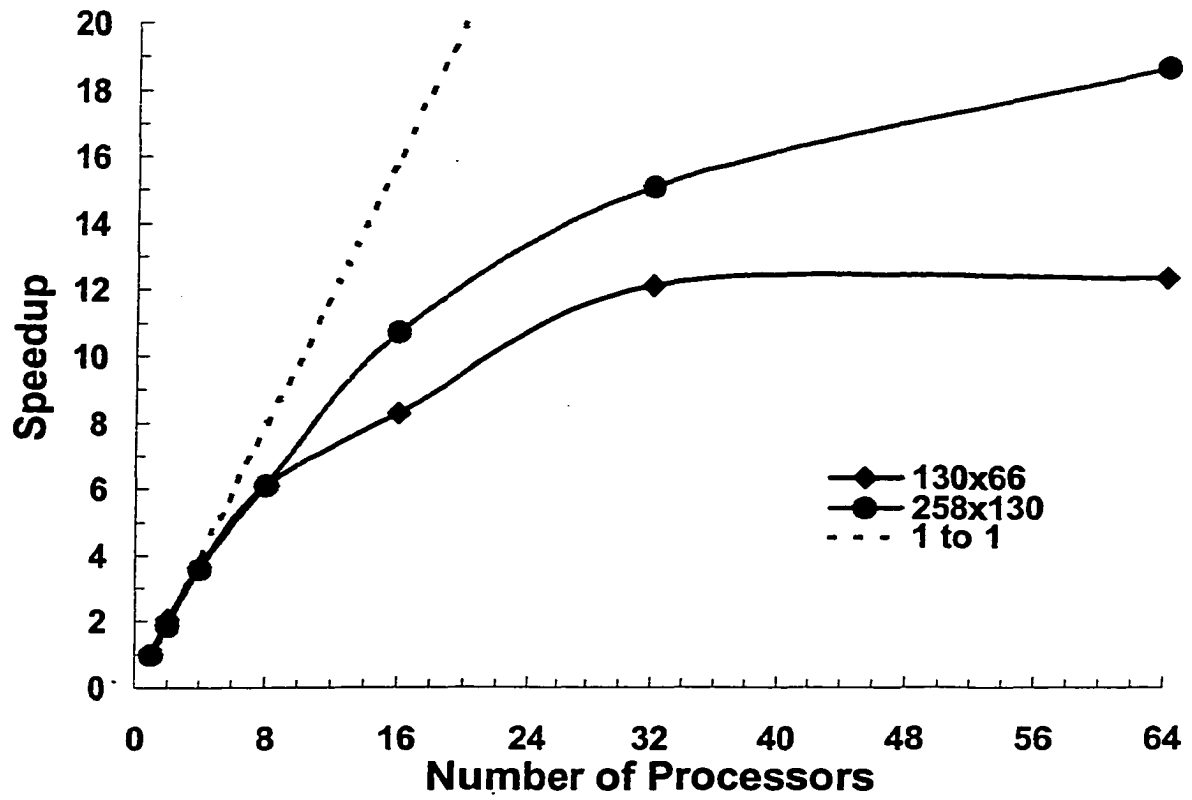


Figure 10. Speed up ratio and its change with domain size and the number of processors in a cluster

CHAPTER 3. IMPLEMENTATION AND PERFORMANCE EVALUATION OF PARALLEL COMPUTATIONS BY USING CLUSTER OF NETWORKED WORKSTATIONS

A paper published in the Proceedings of International Conference on
Parallel and Distributed Processing Techniques and Applications

Hao Wang ^{1,2,3}, G.M. Prabhu ², E.S. Takle ³, and J. Shen ¹

¹ IBM, Rochester, MN 55901, USA; ² Department of Computer Science,

³ Department of Geo.& Atmospheric Sciences, Iowa State University,
Ames, IA50011, USA

Abstract

We design, implement, and compare several parallel version codes of our shelterbelt turbulent flow model system which involves extremely strong non-linear processes, solid-air interactions, flow-pressure linkages, and turbulent feedback mechanisms. We evaluate the performance of each implementation and show that the parallel code's performance depends on its design and coding strategies. The performance gains of the domain decomposition parallel version of our model system are quite satisfactory. For functional decomposition, we even obtained parallel performance loss. We analyze the speedup and its change with domain size and number of processors. There is tradeoff between reduced computation load and increased communication load, and the relative weight, which is related to domain size, number of processors, and problem specific features, determines the speed ratio and load balance ratio.

1. Introduction

Fast computers have stimulated the rapid growth of a new way of doing science. The two broad classical branches of theoretical science and experimental science have been joined by computational science. Computational scientists simulate on supercomputers phenomena too complex to be reliably predicted by theory and too dangerous or expensive to be reproduced in the laboratory. With the rapid development of microcomputers and fast networking with high-speed switches, parallel processing on distributed networks of workstations has emerged as a cost-effective method of high-performance computing – cluster computing.

Previous work on parallelizing old climate models under USDOE “grand challenges” project was focused mainly on relatively simple linear shallow water model with very low spatial resolution. In the past several years, we were developing non-linear shelterbelt turbulent flow model with strong flow-pressure and solid-air interactions. This new model is very computation-intensive. In this paper, we describe our work to seek the better parallelization of this non-linear, strong flow-pressure and very-high-resolution model system. This problem can be widely used in real applications.

2. The Shelterbelt Turbulent Flow Model

This model system was developed to solve the real-world complex turbulence and flows around vegetation and climate problems. The real earth’s surface is not an ideal homogeneous flat surface as required by classical atmospheric boundary-layer theory. Ideal flows and turbulence are themselves complex, but this complexity is significantly increased for boundary-layer flows and turbulence around vegetation. We derived a general set of equations for a

porous medium and developed a model system. Limited by the length of this paper, we could not describe it in details. Please refer our previous journal papers (Wang and Takle 1995a, 1995b, 1996a, 1996b, 1996c, 1997a, 19997b) for the model's physics and mathematics.

3. Problem Analyses and Sequential Computation Timing

In order to understand the roles of various processes in this application and the corresponding computational requirements, we analyze the sequential code before we parallelize it.

3.1 The solver and Computation Time

Most people who work on parallel computing focus on the solver. Recently, there are a few publications that describe parallel algorithms of linear algebra (tridiagonal system). We also began with parallelizing this solver in our model system in the last year. However, no significant speedup was achieved by parallelizing this solver, no matter what kind of parallel algorithms were used. Therefore, it is necessary for us to find out why.

We also know that the solver function is called most frequently. That is true as shown in Table 1: the solver is called 636 times more than other functions! Therefore, we also thought that most frequently called function also would be most important. But, that is not true, especially for a complicated nonlinear system with feedback and interactions. Usually the solver only accounts for 1-5% length of the system code, and the linear equation solver is very fast.

Table 1. Computation profile for the shelterbelt with porosity of 50%

Time in seconds	+children	calls	name
20468.309 (88.5%)	20468.309	5000	Calc-P
488.933 (2.1%)	550.361	5000	Calc-W
488.689 (2.1%)	551.616	5000	Calc-U
424.070 (1.8%)	471.217	5000	Calc-V
407.818 (1.8%)	466.372	5000	Calc TKE
322.261 (1.4%)	322.261	5000	CalcContinuous
230.056 (1.0%)	230.056	3180000	run-Solver
208.895 (0.9%)	208.895	5000	CalcP-UVW
84.380(0.4%)	84.380	5000	boundCondition
8.203(0.0%)	8.203	1	CalcStorage
5.581 (0.0%)	23137.430	1	Total
0.234 (0.0%)	0.234	1	initial

3.2 Non-linear Processes and Theoretical Analyses

The interactions and feedback between different variables complicate the computation and theoretical analyses. For large spatial scales, the simple and linearized shallow-water equation is used to analyze the theoretical performance of parallel computing. That is feasible because there exist few and weak feedback and interactions for large spatial scale. As dx

(spatial resolution) is large, both $u \, du/dx$ (nonlinear advection) and dp/dx (pressure driving force) are small. When dx is small, the nonlinear term and driving force term become extremely large. Moreover, what we want to study and reproduce (simulate) are interaction mechanisms and processes. Therefore, if we simplify the equations by linearizing them, we cannot get the mechanisms and nonlinear processes from our simulations.

We try to analyze our nonlinear feedback model system. We list the main functional parts: (1) parameters inputs, (2) grid generator, (3) initialization, (4) compute TKE, (5) compute dynamic pressure perturbation, (6) compute U, (7) compute V, (8) compute W, and (9) the solver. Please keep in mind that we cannot isolate each part and ignore feedback and interactions between different parts. Assume the domain size is $n \times n$, we estimate computation of each part as:

(1) parameters input

$O(1)$ – constant time

(2) grid generator

$O(n)$

(3) initialization

$O(n^2)$

(4) compute TKE

$O(n^2)$

(5) compute dynamic pressure perturbation

$O(n^2)*M$ -- consider feedback and interactions between wind and pressure, this term may cause $O(n^3)$, where M depends on domain size and magnitude of physical forcing.

(6) compute U

$$O(n^2)$$

(7) compute V

$$O(n^2)$$

(8) compute W

$$O(n^2)$$

(9) the solver

$$O(n)$$

Therefore, the total sequential computation time is $O(Mn^2) + O(n^2) + O(n) = O(Mn^2) + O(n^2)$, where M is not a small number, it depends on physical conditions.

3.3 Computation of Dynamic Pressure Perturbation

From above estimates, we see that the dynamic pressure perturbation module is very time-consuming and it depends on the magnitude of physical forcing. We calculated the various magnitudes of physical forcing (shelterbelts). For a porous shelterbelt, the computation time of dynamic pressure perturbation accounts for 66% total computation time; however, for a dense shelterbelt, the computation time of dynamic pressure perturbation accounts for as high as 91.4% total computation time (Tables omitted). We listed the computation time for a medium-dense shelterbelt in Table 1, where the computation time of dynamic pressure

perturbation accounts for 88.5%. The denser the shelterbelt, the stronger interactions between flow and pressure. Pressure gradient force drives the flow, and the change in flow also modifies the pressure. So, when wind (U , V , or/and W) changes, we need compute the divergence or convergence; the divergence or convergence causes the changes of pressure; after computing new pressure perturbation, we also need adjust wind according to current pressure field.

4. Functional Decomposition Parallel Programming (FDPD)

4.1 Functional Decomposition

Functional parallelism computations are based on different operations and functions. Separate tasks or functions must be executed at the same time on different processors for achieving higher speed. Before decomposing the functional modules, we must understand the relationships and data dependency between different functions.

As we discussed in the above section, we mainly have 9 different functional modules. The functions outside the time-step iteration loop are not worth parallelizing because they execute only once for the whole run. Also these functions must process the large amount of data. If we parallelize them, they will have heavy communication overhead because all data needed are sent to all processors in the group. We let the root processor handle all these functions and transit between time steps, while other six functions take 6 different machines. These functions take inputs from the previous time-step values, and run independently within the same time step. They are synchronized at the end of completion of each time step so that for every time step, different functions use the same previous time-step values as inputs. At the end of completion of the function for each time step, the processor will broadcast the updated

values to all other processors so that other processors can use the newest values computed by other processors.

In our original sequential code, we used just-computed values for the same time step but the computation was done before another variable's prediction began. By using newest available values of variable, we can improve the performance. When we rewrote the sequential code into functional parallel code, we did not use currently available new values of the same time step for minimizing the computation overhead.

4.2 Speedup, Communication Overhead, and Analyses

Let us compute the speedup ratio for our 7-processor as listed in Table 2 vs a single processor as listed in Table 1.

Serial execution time = 23137.430

Parallel execution time = 29696.437

Therefore, $\text{speedup} = 23137.430 / 29696.437 = 0.779$, which is less than 1. That means that the parallel code took more time than the serial one. What cause such inefficiency? Let us make more detailed analyses of data in Table 2.

The major portion of the computation time is consumed by calculating dynamic pressure perturbation as shown in Table 2f. Computing dynamic pressure perturbation consumed 17424.77 seconds for the functional decomposition parallel code and 20468.309 seconds for the serial code, thus gaining 14.9% speedup in terms of computation of dynamic pressure perturbation. However, node 5 which computed dynamic pressure perturbation needs extra 11922.345 seconds to broadcast and receive the whole updated new time-step values

Table 2a. Computation and communication profile on node 0

time in seconds	+children	calls	name
29346.581 (98.8%)	29346.581	30000	communication-ove
266.310 (0.9%)	266.310	5000	CalcContinuous
70.558 (0.2%)	70.558	5000	boundCondition
10.651 (0.0%)	10.651	1	CalcStorage
2.060 (0.0%)	29696.437	1	Total

Table 2b. Computation and communication profile on node 1

time in seconds	+children	calls	name
28965.449 (97.5%)	28965.449	30000	communication-ove
339.322 (1.1%)	379.669	5000	Calc tke
264.619 (0.9%)	264.619	5000	CalcContinuous
73.206 (0.2%)	73.206	5000	boundCondition
40.347 (0.1%)	40.347	800000	run-Solver
10.601 (0.0%)	10.601	1	CalcStorage
2.392 (0.0%)	29696.176	1	Total

Table 2c. Computation and communication profile on node 2

time in seconds	+children	calls	name
28892.673 (97.3%)	28892.673	30000	communication-ove
412.246 (1.4%)	453.147	5000	Calc-U
263.458 (0.9%)	263.458	5000	CalcContinuous
73.945 (0.2%)	73.945	5000	boundCondition
40.901 (0.1%)	40.901	800000	run-Solver
10.651 (0.0%)	10.651	1	CalcStorage
2.358 (0.0%)	29696.474	1	Total

Table 2d. Computation and communication profile on node 3

time in seconds	+children	calls	name
28957.078 (97.5%)	28957.078	30000	communication-ove
353.075 (1.2%)	384.800	5000	Calc-V
264.503 (0.9%)	264.503	5000	CalcContinuous
76.625 (0.3%)	76.625	5000	boundCondition
31.724 (0.1%)	31.724	800000	run-Solver
10.600 (0.0%)	10.600	1	CalcStorage
2.651 (0.0%)	29696.497	1	Total

Table 2e. Computation and communication profile on node 4

time in seconds	+children	calls	name
28888.417 (97.3%)	28888.417	30000	communication-ove
411.778 (1.4%)	452.978	5000	Calc-W
263.083 (0.9%)	263.083	5000	CalcContinuous
78.639 (0.3%)	78.639	5000	boundCondition
41.200 (0.1%)	41.200	80000	run-Solver
10.616 (0.0%)	10.616	1	CalcStorage
2.668 (0.0%)	29696.642	1	Total

Table 2f. Computation and communication profile on node 5

time in seconds	+children	calls	name
17424.722 (58.7%)	17424.722	5000	Calc-P
11922.345 (40.1%)	11922.345	30000	communication-ove
262.622 (0.9%)	262.622	5000	CalcContinuous
73.155 (0.2%)	73.155	5000	boundCondition
10.599 (0.0%)	10.599	1	CalcStorage
2.860 (0.0%)	29696.542	1	Total

Table 2g. Computation and communication profile on node 6

time in seconds	+children	calls	name
29156.474 (98.2%)	29156.474	30000	communication-ove
267.219 (0.9%)	267.219	5000	CalcContinuous
181.901 (0.6%)	181.901	5000	CalcP-UVW
76.128 (0.3%)	76.128	5000	boundCondition
10.814 (0.0%)	10.814	1	CalcStorage
2.581 (0.0%)	29695.358	1	Total

to/from all other processors which also compute other functions independently. Because u , v , and w affect TKE and p , p drives u , v , and w , and TKE also affects u , v , and w ; there exist very complicated interactions and feedback. Computing any of these variables needs the information of other variables. Therefore, there are $O(n^2)$ message passing. Moreover, the program should be synchronized at the completion of each function to the same starting point so that all functions can get the same time-step values as new initial values. The computation of pressure is time-consuming, so other processors just sit there to wait for the completion of the pressure computation. The average load balance ratio is only 51%. The parallel time is determined by the maximum time of the most time-consuming function computing plus communication overhead.

5. Domain Decomposition Parallel Programming (DDPP)

One of the most critical design decisions in any parallel program is how to partition your problem. A poor or rigid partitioning choice might result in disastrous parallel performance. As we saw from the last section, functional decomposition not only is not flexible (dependent on how many main function components; we had 7 main functions for shelterbelt turbulent flow problem), but also causes very poor performance (parallel processing with 7 processors needs longer time than serial processing with only one processor). A flexible partitioning choice, designed to optimize real world problems will result in much better parallel performance.

5.1 Domain Decomposition and Efficiency

Domain decomposition refers to spatially partitioning the computational domain. The effectiveness of spatial decomposition methods can be explained in terms of volume and surface area. When we consider partitioning a numerical grid problem spatially, the volume enclosed by a particular partition will correspond roughly to the number of cells. And the surface area of the section will correspond roughly to the communications needed for that partition to communicate with its neighbors. The domain is vertically sliced in our application (Figure omitted). Let us assume the domain size is $n \times n$. We sliced it into m small regions, each region has size $n/m \times n$ (we can divide it into different size regions; but for convenience of analysis, we assume the regions have the same size). Then we distribute each region to a processor, so the computation load is reduced and performance is enhanced; however, this region needs to exchange information with its neighbors, and communication load increases, with additional message complexity of $O(2n)$ (please note that the functional decomposition brings additional

messages of $O(n^2)$). Therefore, there are tradeoffs also for domain decomposition, and the final performance depends on the relative effects of both factors.

We ran our domain decomposition MPI parallel version shelterbelt turbulent flow model with different processors and different sizes. The parallel performance gains are significant for all runs. Because the timing data are huge, we summarize the key performance here. We also evaluate the correctness and errors of multiple processing. As shown in Table 3, the differences of computed results by using different number of processors are very small.

Table 3. Comparisons of Computed Mean Kinetic Energy (MKE)

#Proc.	1	2	4	8	16	32	64
MKE	16.3762	16.3739	16.3735	16.3731	16.2728	16.3729	16.3726

5.2 Speedup Ratio and Its Changes with Domain Size and Number of Processors

Figure 1 shows the changes of speedup ratio with number of processors and domain size. The curve with squares is for the domain size of $(128+2) \times (64+2)$, and the curve with solid circles is for the domain size of $(256+2) \times (128+2)$. We also plot the ideal 1 to 1 line. As shown from the Figure, speedup increases with increasing number of processors. For small number of processors, the curves are close to the ideal line; but, with the increasing number of processors, the curves depart from the ideal line, and the performance gains slow down, especially for small domain size. This is the result of tradeoffs between reduced computation load and increased communication load after the domain is decomposed. When the number of

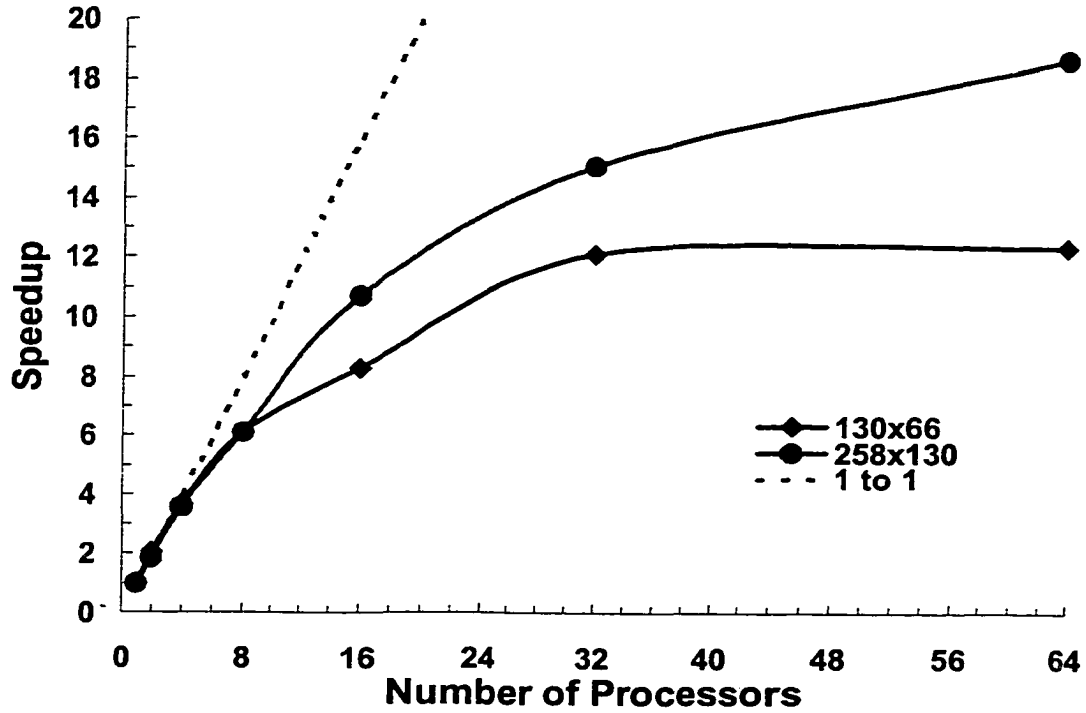


Figure 1. Speedup and its changes with domain size and number of processors

processors is small or the domain is large, the reduced computation dominates. When the number of processors is large, each processor processes only a few grids, the performance gains are limited, but the increased communications dominate.

5.3 Load Balance and Analyses

Table 4 lists the changes of load balance with the number of processors. We can see that the load balance ratio improves a little bit with large number of processors.

Table 4. Load balance ratio for domain size of $(256+2)*(128+2)$

#Processors	2	4	8	16	32	64
Load Balance	0.89	0.90	0.89	0.91	0.93	0.93

6. Summary and Conclusions

We designed, implemented, and compared several parallel version codes of our shelterbelt turbulent flow model system which involves extremely strong non-linear processes, solid-air interactions, flow-pressure linkages, and turbulent feedback mechanisms. There is a need for running this code in parallel because this code is very time-consuming and because we can use it to research many cutting-edge scientific problems. This paper described the implementation and performance evaluation on a cluster of workstations. The code used MPI for message passing and three design approaches were implemented: most-frequently-used module, functional decomposition, and spatial decomposition. The spatial decomposition was found to give the best performance for our application.

Acknowledgments

This research was supported by the US Department of Agriculture Grant 96351083892, the US Department of Energy NIGEC Grant DE-FC 0390ER61010, and the NRI Competitive Grant 93-37101-8954. The final work was done on the ALICE network of workstations in the Scalable Computing Laboratory. The ALICE computer facility was maintained by the USDOE Ames Laboratory.

References

- Baker, L. and B.J. Smith, 1996: *Parallel Programming*. New York, NY: McGraw-Hill. Pp.381.
- Gropp, W., E. Lusk, and A. Skjellum, 1994: *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. Cambridge, MA: The MIT Press.
- Pritchard, D. and J. Reeve (Eds.), 1998: *Euro-Par'98 Parallel Processing*. Lecture Notes in Computer Science, No.1470, 1-1152.
- Wang, H. and E.S. Takle, 1995a: Boundary-layer flow and turbulence near porous obstacles: I. Derivation of a general equation set for a porous medium. *Boundary-Layer Meteorology*, **74**, 73-88.
- Wang, H. and E.S. Takle, 1995b: A numerical simulation of boundary-layer flows near shelterbelts. *Boundary-Layer Meteorology*, **75**, 141-173.
- Wang, H. and E.S. Takle, 1996a: On three-dimensionarity of shelterbelt structure and its influences on shelter effects. *Boundary-Layer Meteorology*, **79**, 83-105.
- Wang, H. and E.S. Takle, 1996b: On shelter efficiency of shelterbelts in oblique wind. *Agric. Forest Meteorology*, **81**, 95-117.
- Wang, H. and E.S. Takle, 1996c: Numerical simulations of shelterbelt effects on wind direction. *Journal of Applied Meteorology*, **34**, 2206-2219.
- Wang, H. and E.S. Takle, 1997a: Momentum budget and shelter mechanism of boundary-layer flow near a shelterbelt. *Boundary-Layer Meteorology*, **82**, 417-435.
- Wang, H. and E.S. Takle, 1997b: Model-simulated influences of shelterbelt shape on wind-sheltering efficiency. *Journal of Applied Meteorology*, **36**, 695-704.

CHAPTER 4. DESIGN OF PARALLEL ALGORITHM AND PERFORMANCE MODELING & THEORETICAL ANALYSIS OF CLUSTER COMPUTING

A paper to be submitted to SIAM Journal on Scientific Computing

Hao Wang, G.M. Prabhu, and E.S. Takle

Department of Computer Science and Department of Geological and Atmospheric Sciences

Iowa State University, Ames, Iowa 50011

Abstract

In this paper, we discuss processes and best practices for using cluster computing to solve a computation-intensive scientific problem. We assess the impact of cluster computing on speedup, and load balance in every step of parallelization, including the formulation of a problem, mathematical modeling, computer modeling, algorithm development and implementation, task partitioning, and mapping. We also develop a theoretical performance model for this application. Our results show that for MPP-based parallel computing, the parallel computation time decreases with increasing number of processors; for cluster-based parallel computing, the parallel computation time first decreases with increasing number of processors; however, with further increase in the number of processors, the parallel computation time increases because of the increasing communication overhead. For MPP-based parallel computing, the speedup increases with increasing number of processors as predicted by John Gustafson who revised Amdahl's law. There exists a maximum speedup for a given problem

with a given computation algorithm that occurs at the turning point number of processors in a cluster parallel computing environment. The maximum speedup is less than half of the turning point number of processors, which is related to communication latencies, bandwidths, size of problem, and computation load. The more computation intensive the problem, the higher the turning point number of processors, the higher the speedup. We also examine our theory with experiments, and the results show that for large domains, the maximum speedup is 19.6 and the turning point number of processors is 105; for small domains, the maximum speedup is 12.4 and the turning point number of processors is 56.

1. Introduction

Computing is fast becoming the most frequently used technique to explore new questions. Fast computers have stimulated the rapid growth of a new way of doing science. The practical issues involved in parallel computing are much the same as those of a group of people attempting to work together on a single project. How should the work be divided up and in what order? How will people communicate and coordinate work? How do we accommodate workers completing tasks at different speeds? How can output from different tasks be merged to form the final product? Who will oversee the overall project? There are many fundamental problems regarding parallelism and computing. In this paper, we discuss the best practices and processes for using cluster computing to solve a computation-intensive problem and assess its impact on performance and load balance; we also develop a parallel performance model and do theoretical analysis.

2. Design Process of Parallel Computing

2.1 Mathematical Modeling of Physical Problem

This step is very important because if the mathematical model does not capture the essentials of physical system, we cannot get the best results even if we have the best computation with the best computer algorithm. This step doesn't have direct impact on the load balance of computation. However, the complexity of the mathematical model dictates the demand of computation resources. A complex mathematical model is more difficult to compute (complicated computer algorithms) than a simple mathematical model. Therefore, a complex mathematical model leads to a harder issue of load balance. For example, the detailed shelterbelt turbulent flow problem has complicated solid-air interactions that induce complicated flow-pressure interactions; hence, the computation of dynamic pressure perturbation takes a lot of computer time; other functional computation waits idly for the pressure values, causing load imbalance in the functional decomposition parallelism [Wang et al. 2000]. In addition, since the complex mathematical model has the complicated dependencies of variables, we need to spend more time on message-passing design in cluster computing. In summary, while we need to verify the laws of physics in the derived mathematical model, this step also provides the requirement analysis of software development. We derived a set of governing equations for porous turbulent flows [Wang and Takle 1995a].

2.2 Computer Modeling of Mathematical Problem

From step 1, we obtain the continuous mathematical equations that govern the physical system. Until we reach the day that computers can think effectively (really a thinking-machine; but even at that time, how does the thinking-machine resolve to non-linear problems that people still don't know how to obtain the analytical solutions to?), we need to discretize these equations into a form that computers can store and recall. Usually we set up the computational domain and divide it into many grids or elements, and let grids/elements hold values. There are a lot of algorithms on the computational methods of differential equations. The major issue is the numerical stability of computation for non-linear equations. We developed the computer model of shelterbelt turbulent flow model [Wang and Takle 1995b, 1996a,b,c, 1997a,b; Wang et al. 1998a, b, 2000, 2001].

This step has significant impact on data exchanges for different tasks and load balance for parallel computing. It is this step that determines the message channel structure and tasks for parallel computing. Usually we get the five-point stencils for the staggered grid system, and nine-point stencils for the un-staggered grid system. The former has better efficiency and better performance than the latter for parallel computing either on cluster of computers or MPPs, because the former needs to exchange data only with its 4-neighbors, while the latter needs to exchange data with 8 points around it. In other words, the amount of data for message passing in the latter case is twice as much that in the former case. Unfortunately, many computer scientists as mentioned before who parallelized CCM (PCM) and MM4/5 (MPMM) from the serial codes hardly changed such 9-point horizontal communication pattern [Drake et al. 1993, 1995; Foster 1995; Foster and Gropp 1992; Foster and Michalakes 1993; Foster and

Toonen 1994; Foster and Worley 1993]. Therefore, even the smartest load balancing algorithms cannot make up such big communication costs. Another issue is related to the moving average for smoothing some perturbations to enhance the stability of computational model. We should minimize the use of the moving average since it represents communication overheads for message passing in parallel computing, especially in distributed parallel computing such as cluster computing where bandwidth is limited. This step is similar to software design. In our model, we used the five-point stencils [Wang and Takle 1995b, 1996a,b, 1997a; Wang et al. 1998, 1999, 2000, 2001].

2.3 Analysis of Sequential Algorithms and Methodological Design of Parallel Algorithms

Not each line of code in the program weighs the same for parallel computing. There is a 20/80 rule that says 20% code takes 80% computation time (or sometimes, 10/90 rule). In order to maximize the parallel processing performance, the parallelization should be focused on the parts that take the most time in the corresponding serial code. The profiling of serial code also helps us to find the communication-intensive but computation-light parts so that we can consider duplicating the computations to remove the heavy communication costs in distributed parallel computing or cluster computing. The rule is that if the duplicate computation takes less time than the possible communication costs when we don't do the duplicate computation, then we duplicate the computation to remove the communication costs in cluster computing. This step impacts load balance and parallel performance directly. We also need to evaluate the variability of computation load under different model parameters so that we can optimize them in cluster computing.

Table 1. Computation load variability under different model parameter of porosity

Component	Porosity of 90%		Porosity of 50%		Porosity of 10%	
	Time (s, %)	Calls	Time (s, %)	Calls	Time (s, %)	Calls
Calc-P	5093.5(66%)	5000	20468.3(88%)	5000	22767.7(91%)	5000
Calc-W	491.3(6.4%)	5000	488.9(2.1%)	5000	402.2(1.6%)	5000
Calc-U	490.6(6.4%)	5000	488.7(2.1%)	5000	403.5(1.6%)	5000
Calc-V	420.6(5.5%)	5000	424.1(1.8%)	5000	344.2(1.4%)	5000
Clac-TKE	410.7(5.3%)	5000	407.8(1.8%)	5000	333.9(1.3%)	5000
Cac-Cont	316.2(4.1%)	5000	322.3(1.4%)	5000	252.1(1.0%)	5000
Calc-UVW	206.2(2.7%)	5000	208.9(0.9%)	5000	171.7(0.7%)	5000
Run-Solver	181.8(2.4%)	3180000	230.1(1.0%)	3180000	152.1(0.6%)	3180000
B-Conditions	88.5(1.1%)	5000	84.4(0.4%)	5000	68.2(0.2%)	5000
Calc-Storage	11.1(0.1%)	1	8.2(0.0%)	1	8.2(0.0%)	1
Initial	0.2(0.0%)	1	0.2(0.0%)	1	0.2(0.0%)	1
Total	7717.8		23137.4		24905.4	

As listed in Table 1, if we run the same code 5000 times under different model parameter of porosity, the computational time is different: 7,717.8 seconds for porosity of 90%, 23,137.4 seconds for porosity of 50%, and 24,905.4 seconds for porosity of 15%. From this same table, we also can find that the computational load distribution among model functional components are different for different model parameter of porosity: 66% computational time

on the computation of pressure for porosity of 90%, and 91.4% computational time on the computation of pressure for porosity of 10%. From Table 1, we also can find that even though the linear solver is called 318,000 times, it takes only 0.6-2.4% of computational time. It needs to be pointed out that the pressure computation is controlled dynamically by the preset precision it reaches. For such run-time dynamically controlled termination condition of computation, it is difficult to estimate the computation load, that is why computation time varies largely with model parameter of porosity. Physical reason is that the dense shelterbelt causes strong turbulent flow pressure that needs more iterations and long time to get the balance.

We need a new concept of performance, i.e., we need parallel algorithms that perform well for all possible range of model input parameters that may impact parallel performance. As we discussed above, there is a significant change in computational time for different model parameter. Foster and Toonen [1994] and Foster [1995] discussed that clouds and radiation process caused the load imbalance in parallel computing.

We also do theoretical order analysis of computation load for each individual components as in the paper [Wang et al. 2000].

2.4 Analysis of Functional and Data Dependencies

The understanding of functional and data dependencies is very important to the design of parallel algorithms. Parallel processing needs to divide a big task into many small tasks so that multiple processors can process these small tasks concurrently. The functional and data dependencies can be analyzed from mathematical equations and the methods of computation applied. For example, we analyzed U , V , W , P , TKE , and their relationships, as well as the

order of computation time. Forward differencing for the time terms, centered differencing for the pressure term, upstream differencing for the advection terms, and modified Crank-Nicholson scheme for the turbulent flux terms, all of these determine the message-passing structure for parallel processing [Wang et al. 2000]. This step impacts load balance and parallel performance since good analysis leads to good partitions that minimize communication costs and divide the big task into small tasks of equal computation load that are easily load-balanced.

2.5 Partitioning of Computation to Basic Elements

The partitioning of a design is to divide the problem into small pieces to expose opportunities for parallel processing so that multiprocessors can compute the problem together to gain efficiency and performance. The decomposition of a problem is to divide the computation associated with a problem and the data on which this computation operates. There are two partitioning techniques: (1) functional decomposition and (2) domain decomposition.

This step has the most important impact on load balance and parallel performance: If the divided pieces have very different computation load for each individual piece of a problem, the load balancing for processors in the cluster or MPP is a major issue. If the number of divided pieces is much larger (at least one order larger) than the number of processors, you will have flexibility to use commercial load balancing software or simple cyclic mapping (like what ANL computer scientists did for PCM and MMPM [Foster and Toonen 1994]) to distribute several small pieces of uneven computational load to a processor. However, we will discuss later that the large number of the partitioned small pieces has significantly large communication overhead that impacts the speedup performance of parallel computing. Although a lot of computer

scientists studied the load balance problem, a proved NP-Complete problem that cannot have a tractable solution [Colajanni et al. 1989; Archil et al. 1990; Foster 1995; Garey and Johnson 1979; Hamidzadeh et al. 2000; Hui and Chanson 1999; Kafil and Ahmad 1998; Liao and Chung 1999; Michalakes 1991; Sohn 1998; Watts and Taylor 1998], they either assumed the special computational and data structure or assumed a large number of independent small piece tasks available. Most of them studied the load balancing problem separately from parallel speedup performance. We must keep in mind the performance goal of parallel computing. If the communication overhead is substantially large, even if the load balancing is 100%, the overall speedup performance of parallel processing is still limited. We need to seek the problem-specific partitioning and load balancing solution to increase load balance and to minimize the communication costs at the same time so that we can obtain high overall performance of parallel processing. This is especially true for cluster computing since the interconnection of clusters (usually Ethernet) has larger latency and lower bandwidth compared to the switch bars of MPP. We will discuss this later.

2.5.1 Functional Decomposition

Functional parallelism is a mode of parallelism which involves partition by task, or module in the system, i.e. first decomposing the computation to be performed and then dealing with the data. The computations are based on different operations and functions. Separate tasks or functions must be executed at the same time on different processors for achieving higher speed. This is a natural choice for the design of a complex program, utilizing the expertise of experts of individual components in a complex problem. If we are successful in dividing the

computation into disjoint tasks, then we examine the data requirements of these tasks. If these data requirements are also disjoint, partition is complete. If the data requirements have overlap, we need to consider the communications to avoid replication of data. A functional decomposition partitions not only the computation to be performed but also the code that performs that computation. It affects program structure and reduces the complexity of the overall design. Each individual can write individual piece of code independently and then we collect all pieces of codes for different functions written by different component experts and connect them via interfaces. For example, a simulation of climate may comprise components of atmosphere, land ecosystem, ocean, hydrology, ice, and so on. Each component code is developed by different experts in the corresponding area, especially for the situation where the component code already exists in this area, then functional decomposition is a natural choice.

However, functional decomposition has several problems: (1) the computational load of each function is different, and the load balance is an issue; (2) it doesn't yield a large number of tasks and there is not a flexibility in functional decomposition since the functions in a problem are fixed, so traditional load balancing technique is hardly applied to them without decomposing them into more pieces of small tasks using other techniques such as domain decomposition; (3) communication costs may be huge, the complexity of communication possibly is the same order as the computation data. We discussed these in detail in our paper [Wang et al. 2000].

We divided our model into 9 disjoint components or functions. We also used an important technique in parallel computing: duplicate computation to minimize the communication costs [Wang et al. 2000]. The functions outside the time-step iteration loop are

not worth parallelizing because they execute only once for the whole run. These functions also process a large amount of data. If we parallelize them, there will be heavy communication overhead because all data needed are sent to all processors in the group. We duplicate them and let the root processor handle the transit between time-steps. Table 2 lists the summary of the results regarding communication costs and idle time during the time a processor waits for data from other processors to synchronize on each individual time step.

From Table 2, we can see that communication costs take about 40% of the machine time. This is because the all the computed data needed to be exchanged among processors that handle different functions. While node 5 is busy in computing the pressure, all other nodes are idle waiting for the pressure data from node 5, and the average load balance ratio is only 51%, and the speed up is 0.779, slower than the serial code execution. This is due to both the complexity of the problem and the deficiencies of functional decomposition as discussed above.

Table 2. Communication and Idle time for functional decomposition

	Node0	Node1	Node2	Node3	Node4	Node5	Node6
Communication + Idle Time	98.8%	97.5%	97.3%	97.5%	97.3%	40.1%	98.2%

2.5.2 Domain Decomposition

Domain decomposition refers to spatially partitioning the computational domain, i.e., first decomposing the data associated with a problem, then working out how to associate computation with data. This partitioning yields a number of tasks, each comprising some data and a set of operations on that data. An operation may require data from other tasks, and

communications are needed to move the data. Usually this partitioning method is flexible and scalable. It can create a large set of small tasks that can be used easily for load balancing as PCM/MMPM did [Foster and Toonen 1994; Michalakes 1991].

However, a large amount of small tasks solves the load balance problem easily but degrades the parallel performance due to excessive communication costs, and makes the solution not scalable especially in cluster computing that has large communication latency and small bandwidth. And we point out that speedup is limited due to communication overhead in the cluster computing environment, not like in the MPP environment where speedup increases with the number of processors.

Certain very fundamental parallel principles are at work when we decompose a problem spatially. The effectiveness of spatial decomposition methods can be explained in terms of volume and surface area. When we consider partitioning a numerical grid problem spatially, the volume enclosed by a particular partition will correspond to the number of cells, and the surface area of the section will correspond to the communications needed for that partition to communicate with its neighbors. For example, if we divide 100×100 into 10s of 10×10 , for each piece of 10×10 , we need to change 2×100 data with neighbors in order to compute 10×10 , the ratio of computation data points over communication data points is $10 \times 100 / (2 \times 100) = 5$. If we divide 100×100 into 20s of 5×100 , then the ratio becomes $5 \times 100 / (2 \times 100) = 2.5$. The smaller the granularity, the bigger the communication overhead. As seen from our paper [Wang et al. 2000], the parallel performance of 258×130 grid is significantly better than that of 130×66 grid.

There is a method that combines functional decomposition and domain decomposition.

Especially when you develop a system consisting of several sub-systems for which codes have already been developed. The design and integration of whole complex system are difficult and need knowledge beyond the ability of any individual. The combined functional and domain decomposition may reduce the complexity of algorithm design. However, load balance and speedup performance may not be optimal.

2.6 Assessment of Communication Needs

The tasks generated by a partition are intended to execute concurrently but cannot, in general, execute independently. The computation to be performed in one task will typically require data associated with another task. There is a need for communication in order for the computation to proceed. There are two kinds of communications: (1) *local communication* that each task communicates with a small set of other tasks (e.g., its neighbors); (2) *global communication* that requires each task to communicate with many tasks or even all the other tasks. Communication is an overhead that reduces the parallel performance, especially in cluster computing environment. Communication patterns are very important to mapping and load balancing algorithm design. The communications can be classified as structured communication that has regular structure or unstructured communication that may have arbitrary graphs of networks. It can also be classified as static communication structure that doesn't change in runtime or dynamic communication structure that changes in runtime. According to the coordination fashion, the communications can be synchronous (producers and consumers cooperate in data transfer operations) or asynchronous (without the cooperation of the producer).

We used the 5-point stencils in the finite difference computation [Wang and Takle 1995b, 1996a,b,c, 1997a, b; Wang et al. 1998, 2000, 2001], so we need to exchange data with two neighbors. This is a local communication. We also need to compute global mean kinetic energy and global turbulent kinetic energy for model monitoring as well as global pressure perturbation to determine when the preset precision reaches. These are global communications.

2.7 Agglomeration to Reduce Communication Costs

Although a large amount of small tasks yields good load balance, it also creates large communication overhead that we need to minimize. We need to look into tasks and communication patterns to see if we can combine, or agglomerate, tasks identified by the partitioning analysis. This step is very important to load balance and parallel performance. Despite great advances in multicomputer architecture design, interprocessor communication remains a notoriously unavoidable overhead in the execution of parallel programs. This overhead is incurred when tasks of the parallel program assigned to different processors exchange data. Since the communication cost between tasks assigned to the same processor is considered to be negligible, task duplication is one way of reducing the interprocessor communication overhead. This can potentially reduce the start times of waiting tasks and eventually improve the overall completion time of the entire program. Duplication-based scheduling can be useful for systems, such as networks of workstations, that have high communication latencies and low band widths. With task duplication, considerable improvements in speedups have been reported [Ahmad and Kwok, 1998; Lewis and El-Rewini 1992].

On the other hand, we need to keep enough tasks to preserve load balance, flexibility, and scalability of parallel processing. Best of all, we should keep the size of agglomerated tasks dynamically so that we can adapt it to different machines, different load distributing techniques, overlapping communication and computation. From software engineering point of view, if we parallelize the existing codes, we need to minimize the changes to reduce software engineering costs.

As discussed in our paper [Wang et al. 2000], we developed codes from scratch, so we have more freedom to choose optimal parallelization approach. We keep the size of tasks dynamically to adapt to the size of processors, and maintain the partitioning boundaries dynamically to adapt to load balance. We obtained 90%+ load balance rate.

2.8 Mapping Tasks to Processors

In the final stage of parallel algorithm design, we need to specify where each task is to execute. This mapping problem does not arise on uniprocessor or on shared-memory computers that provide automatic task scheduling. Since the mapping problem is known to be NP-complete [Garey and Johnson 1979], many heuristic methods were proposed to find satisfactory suboptimal solutions [Barnard and Simon 1996; Archil et al. 1990; Gilbert and Zmijewski 1987; Liao and Chung 1999]. However, most of them assumed a specific communication structure. Therefore, we need to seek a problem specific solution. Mapping is closely related to load balance. If we map the computation load exactly at the speed of each of processor, and such load doesn't change with time, we don't need load balancing.

Load balancing techniques can be classified as either static or dynamic. Static load

balancing requires complete information on the computing system and workload characteristics to compute optimal schedules. Dynamic load balancing makes little assumption on the system and workload, and scheduling decisions are made in real time based on measured loading information. The nearest-neighbor approach is a dynamic load balancing technique that allows the processors to communicate and migrate tasks with their neighbors only. Each processor balances the workload with its neighbors so that the whole system will be balanced after a number of iterations. Since this technique does not require a global coordinator, it is inherently local, fault tolerant and scalable. Therefore, this approach is a natural choice for load balancing in a highly dynamic environment.

Foster and Toonen [1994], Foster [1995], and Michalakes [1991] used static cyclic mapping to solve load imbalance due to the day/night cycle (radiation calculations are performed only in sunlight) and local physical changes for NCAR CCM2. We used dynamic nearest-neighbor approach to balance our load due to the strong solid-air interactions that demand more computation [Wang et al. 2000].

2.9 Implementation

First we need to choose an implementation environment and to find a way to implement data communications in the parallel environment. PVM, or Parallel Virtual Machine, and MPI, or Message Passing Interface, are two integrated sets of software tools and libraries [Al-Tawil 2001; Bubak 1997; Geist et al. 1994, 1996; Gropp 1997; Gropp and Lusk 1996; Gropp et al. 1994, 1996; Minnich et al 1998; Pacheco 1997; Sunderam 1996]. We have reviewed them and implemented our model both on PVM [Wang et al. 1998] and on MPI [Wang et al. 2000]. The

implementation may impact performance and load balance due to the differences of communication speed and topologies, but we did not find the impact to be big. The functionality is more important for choosing implementation environment, and we finally chose MPI.

Recently some people tried other implementation, but the results are not encouraging because of large overhead of communication channel implementation. For example, Mivakant et al. [2001] evaluated the effect of CORBA on the performance of a group communication service, and concluded that there is a substantially high performance cost, but this performance cost can be reduced to a certain extent by carefully choosing a design and tuning various protocol parameters such as buffer size and timer values.

2.10 Evaluation and Redesign

In adopting good software development practice, we tested the results and went through steps 1-9 to improve or fine-tune the design and performance of our parallel program. During this cycled-process, we always compared our results with the results of the serial code and with previously implemented parallel codes.

3. Parallel Performance Modeling and Theoretical Analysis

Amdahl's law stated that for a fixed problem size, the maximum speedup is limited to $1/s$ when the number of processors is infinite, where s is the fraction of serial work that cannot be parallelized in the given problem. According to their experience on MPP (massive parallel processors) with scientific and engineering computing, Gustafson [1988] correctly argued that

the problem size should not be fixed, parallel machine should be used to solve big problem; therefore, he obtained the speedup= $s+pg$, where g is the fraction of parallel task, and p is the number of processors. In the above models, they did not consider communication costs. Let us consider the communications costs and idle time due to the imbalance of computation load.

We can divide the parallel time T_p into 3 parts: computation time (T_{comp}), communication time (T_{comm}) and idle time (T_{idle}), and we assume that we have p processors, i.e.

$$T_p = \frac{T_{comp} + T_{comm} + T_{idle}}{p} = \frac{\sum_{i=0}^{p-1} T_{comp}^i + \sum_{i=0}^{p-1} T_{comm}^i + \sum_{i=0}^{p-1} T_{idle}^i}{p} \quad (1)$$

The computation time of an algorithm (T_{comp}) is the time spent on performing computation. If we have a sequential program that performs the same computation as the parallel algorithm, we can determine T_{comp} by T_s (the execution time of the sequential code). It depends on the size of a problem. For our case, we have $M \times N$ grid, therefore

$$T_{comp} = t_c MN \quad (2)$$

where t_c is a coefficient.

The communication time of an algorithm (T_{comm}) is the time that its task spends on sending and receiving messages. There are two distinct kinds of communications: interprocessor communication and intraprocessor communication. We map one task per processor, so we consider interprocessor communication. In order to send/receive a message

of length (L), we need to initiate the communication channel link (costing t_s , i.e. latency) and send the message at t_w per word (bandwidth), therefore, the time required T_{msg} is

$$T_{msg} = t_s + t_w SL \quad (3)$$

where S is the factor of competition for bandwidth, representing the number of processors needing to send concurrently over the same wire.

For the cluster of computers that use Ethernet, $S=p$, since every time only one processor is permitted to transfer data; for crossbar-based MMPs, every processor has its channel, $S=1$.

We need to exchange $2N$ data with its two neighbors, therefore:

$$T_{comm}^{cluster} = pt_s + 2t_w p^2 N \quad (4)$$

$$T_{comm}^{MPP} = pt_s + 2t_w p N \quad (5)$$

The idle time is the difference between the computation and communication times of the heaviest load processor multiplied by the number of processors and the total actual computation and communication times of all processors. For the functional decomposition, the idle time is as high as 102,769 seconds, or 49.4% of total machine runtime as listed in Table 3.

Table 3. Idle time for functional decomposition

	Node0	Node1	Node2	Node3	Node4	Node5	Node6
Idle Time (s)	17424	17063	16970	17043	17035	0	17234

For the domain decomposition, we got very good load balance. Since it is difficult to analyze idle time analytically, we omit it in the following analysis.

$$T_p^{cluster} = \frac{t_c MN}{p} + t_s + 2t_w pN \quad (6)$$

$$T_p^{MPP} = \frac{t_c MN}{p} + t_s + 2t_w N \quad (7)$$

For the MPP-based computing, computation time decreases linearly with increasing number of processors. However, for the cluster computing, the communication cost is substantially high and increases with the number of processors. Therefore, at some point, with the increasing number of processors, the increased communication cost will outperform the decreased computation load, and the cluster parallel time will increase. At this turning point, we obtain the maximum speedup:

$$p_{tp}^{cluster} = \sqrt{\frac{t_c M}{2t_w}} \quad (8)$$

From equation (8), we can see that the lower the network bandwidth, the lower the turning point. The larger the domain, the higher the turning point. The more intensive computation the higher the turning point. Therefore, the speedup, S , for cluster computing is:

$$S = \frac{p}{1 + \frac{t_s p}{t_c MN} + \frac{2t_w p^2}{t_c M}} \quad (9)$$

and the maximum speedup at the turning point is:

$$S_{\max}^{cluster} = \frac{P_{tp}^{cluster}}{2 + \sqrt{\frac{t_s^2}{2t_c t_w MN^2}}} \quad (10)$$

Therefore, the maximum speedup is less than half of the turning point number of processors. The bigger domain, the higher the maximum speedup. However, the large network latency will reduce the speedup. If the interconnect is 0 latency, then the maximum speedup is exactly half of the number of processors at the turning point.

From equation (9), we obtain:

$$p/S = \frac{1}{E} = 1 + \frac{t_s p}{t_c MN} + \frac{2t_w p^2}{t_c M} \quad (11)$$

where E is the efficiency that is defined as the fraction of time that processors spend doing useful work.

It characterizes the effectiveness with which an algorithm uses the computational resources of a cluster of computers. From Equation (11), we can see that the efficiency decreases with increasing number of processors. We ran our model on a cluster of 64 processors for two sizes of domains (130x66 and 258x130), and we used the data to fit the Equation (11) and obtained the fitting curves (see Figure 1):

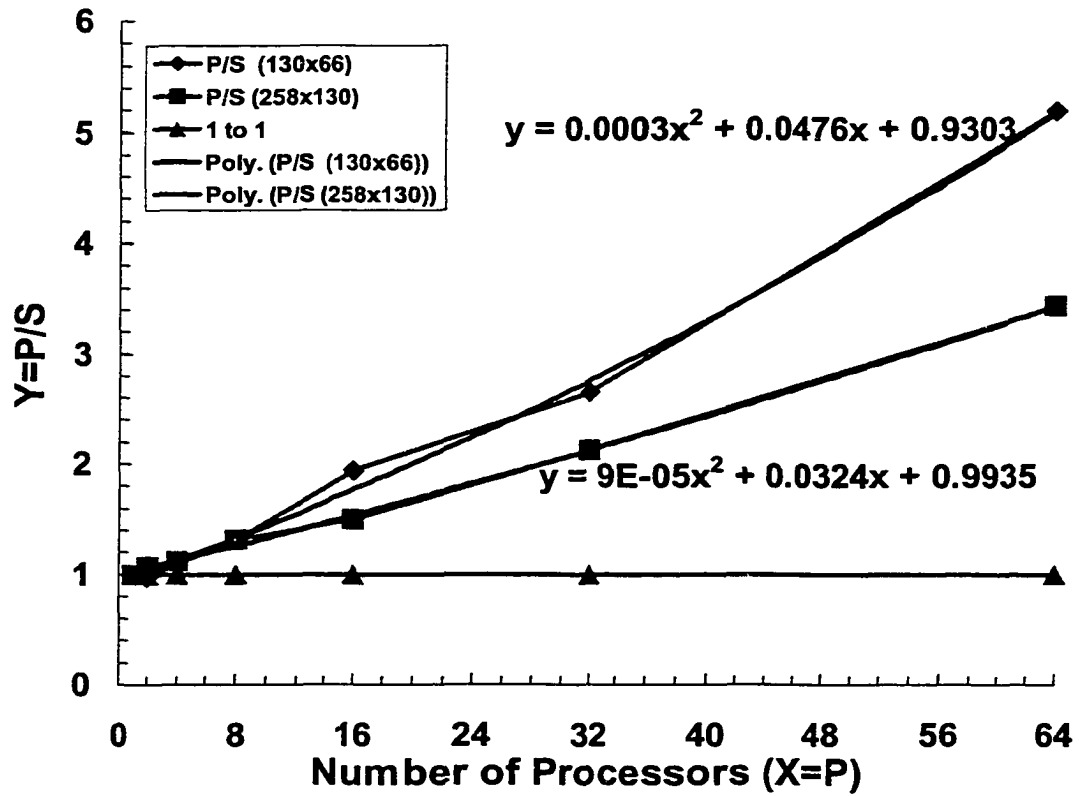


Figure 1. Theoretical and experimental parallel efficiencies and their changes with number of processors

$$p/S = 0.9935 + 0.0324p + 0.00009p^2 \quad (12)$$

for the larger domain, and

$$p/S = 0.9303 + 0.0476p + 0.0003p^2 \quad (13)$$

for the smaller domain.

We solve Equations (12) and (13) to get the maximum speedup by:

$$\frac{\partial S}{\partial p} = 0 \quad (14)$$

and we obtain the maximum speedup for the large domain is 19.6 and the corresponding p is 105; and for the small domain the maximum speedup is 12.4 and the corresponding p is 56.

4. Conclusions

We have discussed the processes for using cluster computing to solve a computation-intensive problem and their impacts on parallel performance and load balance. We developed theoretical performance model for cluster computing and the results show that there exists a maximum speedup for cluster computing that equals less than half of the number of processors. Because of limited communications channel, high latencies, and low bandwidths, it is crucial to design effective parallel algorithms for cluster of computers that minimize communications costs and maximize load distributions. With the development of next generation fast routers and optical communication, latencies will be substantially reduced, and bandwidth will be substantially improved. Therefore, the turning point will be larger, speedup will be higher, and parallel efficiency will be further improved. Cluster computing will be even more attractive due to its further reduced communication overhead, high accessibility and low costs.

5. References

- Al-Tawil, K., 2001: Performance modeling and evaluation of MPI. *Journal of Parallel and Distributed Computing*, vol. 61, 202-223, 2001.
- Barnard, S.T. and H.D. Simon, 1996: Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and Experience*, vol.6(2), pp101-117.

- Bubak, M., J. Dongarra, and J. Wasniewski (eds.), 1997: *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Lecture Notes in Computer Science, No.1332, 1-515.
- Colajanni, M., P.S. Yu, and D. M. Dias, 1989: Analysis of task assignment policies in scalable distributed web-server systems. *IEEE Transactions on Parallel and Distributed Systems*, vol.9(6), pp.585-600, 1989.
- Drake J., I. Foster, J. Michalakes, B. Toonen, and P. Worley, 1995: Design and performance of a scalable Parallel Community Climate Model. *Parallel Computing*, 21(10): 1571-1591, 1995.
- Drake, J.B., R.E. Flanery, I.T. Foster, J.J. Hack, J.G. Michalakes, R.L. Stevens, D.W. Walker, D. L. Williamson, and P. H. Worley, 1993: The message-passing version of the parallel community climate model. In *Parallel Supercomputing in Atmospheric Science*, eds. G-R. Hoffman and T. Kauranne, Singapore: World Scientific, 500-513.
- Drake, J.B., I.T. Foster, J.G. Michalakes, and P.H. Worley, 1995: Parallel algorithms for semi-Lagrangian transport in global atmospheric circulation models. In *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, eds. D.H. Bailey, P.E. Bjorstad, J.R. Gilbert, M.V. Mascagni, R.S. Schreiber, H.D. Simon, V.J. Torczon, and L.T. Watson, *SIAM*, 119-124.
- Archil, F., J.Ramanujam, and P. Sadayappan, 1990: Task allocation onto a hypercube by recursive mincut bipartitioning. *J. Parallel and distributed computing*, vol.10, pp.35-44, 1990.
- Foster, Ian, 1995: *Designing and Building Parallel Programs*. Addison-Wesley, 1995

- Foster, Ian, 1998: *The Grid: Blueprint for a new computing infrastructure*. Morgan-Kaufmann, 1998.
- Foster, Ian, 2000: Internet Computing and the Emerging Grid. *Nature*, Dec.7, 2000.
- Foster, I., and B. Toonen, 1994: Load balancing in climate model. *IEEE Proc. Scalable High-Performance Computing Conf.*, 1994.
- Foster, I. And J. Michalakes: Massively parallel implementation of the Penn State/NCAR Mesoscale Model. Proc. 9th Intl Conf. On Interactive Information and Processing Systems for Meteorology, SIAM, 1993.
- Foster, I. And J. Michalakes, 1993: MPM: A massively parallel mesoscale model. *Parallel Supercomputing in the Atmospheric Sciences*, 354-363, World Scientific, 1993.
- Foster, I., W. Gropp, and R. Stenvens, 1992: The scalability of numerical methods for climate modeling. *Proc. 5th SIAM Conf. On Parallel Processing*, 307-312, IEEE, 1992.
- Foster, I.T. and B. Toonen, 1994: Load balancing algorithms for climate models. In *Proceedings of the Scalable High Performance Computing Conference*, eds. J.J. Dongarra and D. W. Walker, IEEE Computer Society Press, 674-681.
- Foster, I.T. and P.H. Worley, 1993: Parallelizing the spectral transform method: a comparison of alternative parallel algorithms. In *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, eds. R.F. Sincovec, D.E. Keyes, M.R. Leuze, L.R. Petzold, and D.A. Reed, SIAM, 100-107.
- Garey, M.R. and D.S. Johnson, 1979: *Computers and Intractability, A Guide to Theory of NP-Completeness*. San Francisco: Freeman, 1979.

- Geist, A., A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, 1994: *PVM: Parallel Virtual Machine*. The MIT Press.
- Geist, A., J.A. Kohl, P.M. Papadopoulos, 1996: *PVM and MPI: a Comparison of Features*.
- Gilbert J.R. and E. Zmijewski, 1987: A parallel graph partitioning algorithm for a message-passing multiprocessor. *Int'l J. Parallel Programming*, vol.16(6), pp.427-449, 1987.
- Gropp, W., E. Lusk, N. Doss, and A. Skjellum, 1996: A high-performance , portable implementation of the MPI message passing interface standard. *Parallel Computing*, vol.22, No.6, pp.789-828, September 1996.
- Gropp, W., 1997: *Tutorial on MPI: The Message-Passing Interface*. Argonne National Laboratory.
- Gropp W. and E. Lusk, 1996: User's Guide for mpich, a Portable Implementation of MPI. Argonne National Laboratory, ANL/MCS-TM-ANL-96/6.
- Gropp, W., E. Lusk, and A. Skjellum, 1994: *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. Cambridge, MA: The MIT Press.
- Gustafson, J.L., 1988: Reevaluating Amdahl's law. *Communications of the ACM*, vol.31, no.5, pp.532-533, 1988.
- Hamidzadeh, B., L.Y. Kit, and D.J. Lilia, 2000: Dynamic task scheduling using online optimization. *IEEE Transactions on Parallel and Distributed Systems*, vol.11(11), 2000.
- Hui, C.C. and S.T. Chanson, 1999: Hydrodynamic load balancing. *IEEE Transactions on Parallel and Distributed Systems*, vol.10(11), 1999.
- IEEE Computer Society, 1996: *Proceedings of the Second MPI Developer's Conference*.

- Kafil, M. and I. Ahmad, 1998: Optimal task assignment in heterogeneous distributed computing systems. *IEEE Concurrency*, vol. 6, No.3, 1998, pp.42-51.
- Lewis, T.G. and H. El-Rewini, 1992: *Introduction to Parallel Computing*. New York: Prentice Hall, 1992.
- Liao, C.J. and Y.C. Chung, 1999: Tree-based parallel load-balancing methods for solution-adaptive finite element graphs on distributed memory multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, vol.10 (4), 1999.
- Michalakes, J., 1991: *Analysis of Workload and Load Balancing Issues in the NCAR Community Climate Model*. ANL Technical Report ANL/MCS-TM-144.
- Minnich, R., M. Gokhale, A. Marks, J. Kaba, and J. Degood, 1998: Freeware for cluster computing", *USENIX 1998 Annual Technical Conference*.
- Mivakant, S., L. Fei, X. Lin, G. Xing, 2000: On group communication support in CORBA. *IEEE Transactions on Parallel and Distributed Systems*.
- Pacheco, P.S., 1997: *Parallel Programming with MPI*. San Francisco, CA: Morgan Kaufmann Publishers, Pp.418.
- Sohn, J., T.G. Robertazzi, and S. Luryi, 1998: Optimizing computing costs using divisible load analysis. *IEEE Transactions on Parallel and Distributed Systems*, vol.8(3), pp.225-234, 1998.
- Sunderam, V., 1996: PVM, A framework for parallel distributed computing. *Concurrency: Practice and Experience*, vol.2, no.4, pp.315-339
- Wang, H. and J. Shen., 1997: *Parallelization of FFT with MPI and PVM*. Class project for CS625 instructed by Dr. G.M. Prabhu., Iowa State University, Ames, Iowa.

- Wang, H., G.M. Prabhu, and E.S. Takle, 1998: Parallelizing a very high resolution climate model using cluster of workstations with PVM and performance and load balance analyses. *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, CSREA Press, pp.1762-1765.
- Wang, H. G.M. Prabhu, E.S. Takle, R. Todi, 1999: Implementation and performance evaluation for a computation-intensive climate simulation application, *Proceedings of ParCo99 International Conference on Parallel Computing*, Netherlands, 1999.
- Wang, H., G.M. Prabhu, E.S. Takle, and J. Shen, 2000: Implementation and Performance Evaluation of Parallel Computations by Using Cluster of Networked Workstations. *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, CSREA Press, pp. 2007-2013.
- Wang, H. and E.S. Takle, 1995a: Boundary-layer flow and turbulence near porous obstacles: I. Derivation of a general equation set for a porous medium. *Boundary-Layer Meteorology*, **74**, 73-88.
- Wang, H. and E.S. Takle, 1995b: A numerical simulation of boundary-layer flows near shelterbelts. *Boundary-Layer Meteorology*, **75**, 141-173.
- Wang, H. and E.S. Takle, 1996a: On three-dimensionality of shelterbelt structure and its influences on shelter effects. *Boundary-Layer Meteorology*, **79**, 83-105.
- Wang, H. and E.S. Takle, 1996b: On shelter efficiency of shelterbelts in oblique wind. *Agric. Forest Meteorology*, **81**, 95-117.
- Wang, H. and E.S. Takle, 1996c: Numerical simulations of shelterbelt effects on wind direction. *Journal of Applied Meteorology*, **34**, 2206-2219.

- Wang, H. and E.S. Takle, 1997a: Momentum budget and shelter mechanism of boundary-layer flow near a shelterbelt. *Boundary-Layer Meteorology*, **82**, 417-435.
- Wang, H. and E.S. Takle, 1997b: Model-simulated influences of shelterbelt shape on wind-sheltering efficiency. *Journal of Applied Meteorology*, vol.36.
- Wang, H., E.S. Takle, and J. Shen, 2001: Shelterbelt and Windbreaks: Mathematical modeling and computer simulations of turbulent flows. *Annual Review of Fluid Mechanics*, 33:549-586.
- Watts J. and S. Taylor, 1998: A practical approach to dynamic load balancing. *IEEE Transactions on Parallel and Distributed Systems*, vol.9(3), 1998.

CHAPTER 5. ON RELATIONSHIP BETWEEN DRAG AND PRESSURE COEFFICIENTS FOR A POROUS SHELTERBELT

A paper to be submitted to Boundary-Layer Meteorology

Hao Wang, E.S. Takle, and G.M. Prabhu

Department of Computer Science and

Department of Geological & Atmospheric Sciences

Iowa State University, Ames, Iowa 50011

Abstract

Drag produced by flow through porous obstacles is the root cause of spatial inhomogeneities in atmospheric flow in the natural landscape. We analyzed the physics of drag near porous obstacles where the perturbation of the flow field is very strong and nonlinear. We analyzed forces across a shelterbelt by numerical simulation and found that the drag of the belt is balanced mainly by static pressure difference and horizontal advection across the belt. With increasing shelterbelt density, horizontal advection decreases and static pressure difference balances the drag. Total (dynamic plus static) pressure difference across the belt is close to the drag, and the vertically averaged difference between drag and pressure coefficient is within 8%. Horizontal turbulent transport and vertical advection generally play a lesser role. However, large difference exists near the top and bottom of the belt, especially near the top, where vertical turbulent transport plays an important role. Numerical simulations give high-resolution, physically consistent fields, which allow us to examine previous drag calculation

methods. For the far wake, we also use a surface stress-loss method proposed to estimate drag. The results show that the estimated values from indirectly derived stress or stress of higher-order momentum closure or drag plate-measured stress are much smaller than the drag.

1. Introduction

The need for increased accuracy of surface fluxes in mesoscale and global models has brought recent attention to the aggregation of fluxes in an inhomogeneous surface layer. A shelterbelt represents a single inhomogeneity that can be used to study the drag properties of a heterogeneous surface. Many publications have reported modifications of flows around shelterbelts and their linkages with shelterbelt structure. Several review papers are available (e.g., van Eimern et al., 1964; Rosenberg, 1983; McNaughton, 1988; Heisler and DeWalle, 1988, Wang et al 2001). Shelterbelt drag is the root cause of all changes to the flow as a result of a shelterbelt. How to quantitatively estimate drag is crucially important to shelterbelt design and understanding of shelterbelt aerodynamics. Fundamental drag characteristics of obstacles in a uniform free flow have been intensively investigated for over half a century to meet the need of rapid development of aeronautics. However, additional complications for determination of drag of shelterbelts and windbreaks emerge because these obstacles are embedded within a pre-existing atmospheric turbulent boundary layer which itself is complicated. Much of the study on drag due to obstacles in turbulent boundary layers is relatively recent and is limited to relatively simple far wake; however, drag within the shelter, which causes the flow perturbation, needs further study (Taylor, 1988).

Woodruff et al. (1963) determined drag force in the field by neglecting pressure and friction drag and then applying Betz's method originally developed for profile drag determination in wind tunnels:

$$F_D = \rho \int_{z_0}^{z_t} (u_1^2 - u_2^2) dz \quad (1)$$

where F_D is drag force exerted on airflow by the shelterbelt, ρ is air density, u_1 is the velocity at an upwind location x_1 , u_2 is the velocity at a downwind point x_2 , and z_t is a height above the shelterbelt where the flow is reasonably undisturbed, and z_0 is the height at which the wind speed is zero. They obtained good agreement between their calculation and direct drag measurement. Hagen and Skidmore (1971) used a similar method to calculate drag. However, Seginer and Sagi (1971) questioned some of the assumptions made in adapting Betz's method to obstacles embedded in atmospheric turbulent flow. Shear flow and pre-existing turbulence in which shelterbelts are embedded add complications to the momentum method. Seginer (1972) evaluated this method by using Naegeli's data and concluded that calculated drag is very sensitive to distance from the shelterbelt and to the height above the shelterbelt. The calculated drag decreases to zero as x_2 becomes arbitrarily large. Seginer contended that the agreements of observations and calculations by Woodruff et al. (1963) and Hagen and Skidmore (1971) must have been fortuitous because the integration apparently was not carried throughout the length of the wake.

Seginer and Sagi (1971) reformulated the momentum equation and proposed a method for calculating drag. They assumed that the pressures at x_1 and x_2 , given by p_1 and p_2 , are

constant with height and that the horizontal pressure gradient is constant. The drag can be expressed as

$$F_D = (p_1 - p_2)z_t + \int_{x_1}^{x_2} (\tau_t - \tau_0) dx \quad (2)$$

where τ_t and τ_0 are the turbulent stresses at the undisturbed top boundary and at the ground, respectively. If the difference of p_1 and p_2 is negligible, Equation (2) can be simplified to

$$F_D = \int_{x_1}^{x_2} (\tau_t - \tau_0) dx \quad (3)$$

For convenience, they assumed that the logarithmic profile of wind describes the wake region:

$$u = \frac{u_{*0}}{k} \ln\left(\frac{z}{z_0}\right) \quad (4)$$

where u_{*0} is the friction velocity of the undisturbed flow, z_0 is the roughness length of the surface, and k is the von Karman constant. Equation (3) can be approximated as

$$F_D = \rho u_{*0}^2 \int_{x_1}^{x_2} \left(1 - \left(\frac{u(x,z)}{u_0(z)}\right)^2\right) dx \quad (5)$$

and the mean drag coefficient, C_d , can be calculated from

$$C_{dH} = \frac{F_D}{0.5\rho u_H^2 H} \quad (6)$$

where u_H is windspeed of the undisturbed flow at the shelterbelt height, and H is the height of the shelterbelt.

Seginer (1972) used Naegeli's field data to evaluate methods of calculating drag and concluded that neither the method of Woodruff et al. (1963) nor that of Seginer and Sagi (1971) are suitable for determining drag of shelterbelts.

Taylor (1983) used a method similar to Equation (2) to numerically study wakes well downstream of obstacles (far lee) in the neutrally stratified larger scale planetary boundary layer. However, he did not consider pressure, porous obstacles, or their interactions.

It is very difficult to measure drag directly for natural shelterbelts. The only direct measurements of drag force were for simple, artificial model fence (Seginer, 1975; Jacobs, 1985). Many attempts have been made to indirectly infer the drag of shelterbelts and windbreaks from field measurements of wind speed and the formula previously discussed (e.g., Tani, 1952; Woodruff et al., 1963; Hagen and Skidmore, 1971; de Bray, 1971). Boundary-layer flows, even under a constant wind direction, are highly distorted by shelterbelts and have large spatial changes of physical variables. Typical fluctuations in wind direction further complicate field measurements of drag due to changes in path length through the shelter and

characteristics of the wake region. Measurements at isolated points may miss critically important features of the physical fields and lead to large errors in calculations of drag.

In previous reports, we derived governing equations of air motion in porous obstacles (Wang and Takle, 1995a), developed a nonhydrostatic shelterbelt turbulent model, and studied patterns of airflow through a shelterbelt (Wang and Takle, 1995b). In this paper, we evaluate previous derivations of drag and attempts to reformulate the momentum equation on which the methods for calculating drag are based. We illustrate the importance of non-continuity of variables across the shelterbelt for determination of drag. We then focus on the process of drag generation and its effect on the surrounding atmosphere through pressure, flow, and turbulence. We use our model to evaluate the relationship between drag and pressure coefficients as well as contributions of turbulence, advection, and convection to drag. Numerical simulations provide a physically consistent means of examining previous drag calculations. Finally, we used simulated surface turbulent flux reduction and wind reduction as well as drag plate-measured data to evaluate Equations (3) and (5).

2. Theoretical Framework: Drag and Its Relation to Local Perturbations

Previous derivations have used the assumption that all variables and their gradients in the momentum equation are continuous, even in the volume containing the shelter. However, the gradients of variables are not continuous across porous shelterbelts because of the action of drag. We take the pressure gradient as an example. As demonstrated in Figure 1 which shows the simulated distribution of pressure gradient along the x direction (perpendicular to the belt), the pressure gradient reaches its maximum and minimum at the same place - front

edge of the shelterbelt - because pressure increases as airflow approaches the shelterbelt and pressure decreases abruptly when airflow passes through the shelterbelt. At the rear edge, the pressure gradient also is discontinuous. Therefore, the integration of horizontal gradient of pressure in the momentum equation by

$$\int_{z_0}^{z_t} \int_{x_1}^{x_2} \frac{\partial p}{\partial x} dx dz = \int_{z_0}^{z_t} (p_2 - p_1) dz \quad (7)$$

is inappropriate.

We consider a shelterbelt of width of W_B and height H with its front edge at $x=x_f$ and the rear edge at $x=x_b$, so that $W_B=x_b-x_f$ as shown in Figure 2. We divide the domain into 4 zones: (1) front zone, $x=x_1 \rightarrow x_f^-$, and $z=z_0 \rightarrow z_t$; (2) lee zone, $x=x_b^+ \rightarrow x_2$, and $z=z_0 \rightarrow z_t$; (3) upper zone, $x=x_f^- \rightarrow x_b^+$, and $z=H \rightarrow z_t$; and (4) shelterbelt zone, $x=x_f^+ \rightarrow x_b^-$, and $z=z_0 \rightarrow H$. Subscripts f^- and b^+ stand for the location immediately before the front edge at $x=x_f$ and immediately after the rear edge at $x=x_b$ of shelterbelt, respectively, and subscript t , 0 , and H^+ stand for top boundary at $z=z_t$, ground at $z=z_0$, and immediately above the top of the shelterbelt, respectively. All terms of the momentum equation are continuous within each of the four zones.

For simplicity let us consider the equations of motion for stationary conditions in neutral stratification. A typical shelterbelt height of about 10 m is much less than the height of the atmospheric boundary layer, so the effect of Coriolis forces may be neglected. For zones I, II, and III, the stationary, nonhydrostatic, turbulent atmospheric equation of horizontal motion may be written as

$$-\frac{1}{\rho_0} \frac{\partial p}{\partial x} - \frac{\partial uu}{\partial x} - \frac{\partial uw}{\partial z} - \frac{\partial \overline{u'^2}}{\partial x} - \frac{\partial \overline{u'w'}}{\partial z} = 0 \quad (8)$$

where prime indicates fluctuation from the time average, the overline represents the time average (we have omitted the overlines of mean variables). Integration of Equation (8) in zones I, II, and III, respectively, gives

$$\begin{aligned} & \int_{z_0}^{z_i} \langle \rho(u_i^2 - u_{f-}^2) + \rho(\overline{u_i'^2} - \overline{u_{f-}'^2}) + (p_i - p_{f-}) \rangle dz - \\ & \int_{x_1}^{x_f^-} \rho \langle u_i w_i - (\overline{u'w'}_i - \overline{u'w'}_0) \rangle dx = 0 \end{aligned} \quad (9)$$

$$\begin{aligned} & \int_{z_0}^{z_i} \langle \rho(u_{b+}^2 - u_2^2) + \rho(\overline{u_{b+}'^2} - \overline{u_2'^2}) + (p_{b+} - p_2) \rangle dz - \\ & \int_{x_b^+}^{x_2} \rho \langle u_i w_i - (\overline{u'w'}_i - \overline{u'w'}_0) \rangle dx = 0 \end{aligned} \quad (10)$$

and

$$\begin{aligned} & \int_H^{z_i} \langle \rho(u_{f-}^2 - u_{b+}^2) + \rho(\overline{u_{f-}'^2} - \overline{u_{b+}'^2}) + (p_{f-} - p_{b+}) \rangle dz - \\ & \int_{x_b^+}^{x_f^-} \rho \langle (u_i w_i - u_{H+} w_{H+}) - (\overline{u'w'}_i - \overline{u'w'}_{H+}) \rangle dx = 0 \end{aligned} \quad (11)$$

The sum of Equations (9), (10), and (11) gives

$$\begin{aligned}
 & \int_{z_0}^{z_t} \langle \rho(u_1^2 - u_2^2) + \rho(\overline{u'^2_1} - \overline{u'^2_2}) + (p_1 - p_2) \rangle dz - \\
 & \int_{x_1}^{x_2} \langle \rho(u_t w_t - \overline{u'w'_t} - \overline{u'w'_0}) \rangle dx + \\
 & \int_0^{H^*} \langle \rho(u_{b+}^2 - u_{f-}^2) + \rho(\overline{u'^2_{b+}} - \overline{u'^2_{f-}}) + (p_{b+} - p_{f-}) \rangle dz - \\
 & \int_{x_f^*}^{x_b^-} \langle \rho(u_{H^*} w_{H^*} - \overline{u'w'_{H^*}} - \overline{u'w'_{f_0}}) \rangle dx = 0
 \end{aligned} \tag{12}$$

We assume that lateral boundaries at $x=x_1$ and $x=x_2$ and the top boundary at $z=z_t$ are beyond the disturbed region of the shelterbelt and that the top boundary is within the constant flux layer of atmospheric boundary layer. These assumptions are questionable as many investigators have discussed (Seginer and Sagi, 1971; Seginer, 1972; Miller, et al., 1975; Taylor, 1983), but under these assumptions, Equation (12) has a very simple form as

$$\begin{aligned}
 & \int_{x_1}^{x_2} \langle \overline{u'w'_t} - \overline{u'w'_0} \rangle dx + \\
 & \int_0^{H^*} \langle \rho(u_{b+}^2 - u_{f-}^2) + \rho(\overline{u'^2_{b+}} - \overline{u'^2_{f-}}) + (p_{b+} - p_{f-}) \rangle dz - \\
 & \int_{x_f^*}^{x_b^-} \langle \rho(u_{H^*} w_{H^*} - \overline{u'w'_{H^*}} - \overline{u'w'_{f_0}}) \rangle dx = 0
 \end{aligned} \tag{13}$$

From Equation (13), we can see that total surface turbulent flux deficit (first integral) of the sheltered region is balanced by the differences of pressure, flow, and turbulence across

the shelterbelt (second and third integrals). Drag of the shelterbelt causes these differences of pressure, flow, and turbulence across the shelterbelt.

Within porous shelterbelts, the governing equation of airflow may be expressed as follows (Wang and Takle, 1995a, 1995b)

$$0 = -\frac{1}{\rho} \frac{\partial p}{\partial x} - \frac{\partial uu}{\partial x} - \frac{\partial uw}{\partial z} - \frac{\partial \overline{u'^2}}{\partial x} - \frac{\partial \overline{u'w'}}{\partial z} - F_1 \quad (14)$$

Compared with Equation (8), which does not consider the presence of porous obstacles, Equation (14) has an additional term F_1 arising from the form drag causes by the porous obstacles. This force can be expressed (Wang and Takle 1995a) as

$$F_i = \frac{1}{V} \int \int_S p n_i dS' - \frac{v}{V} \int \int_S \frac{\partial u_i}{\partial n} dS' \quad (15)$$

This is the sum of the integration of pressure and wind shear over the obstacle elements' surface area (S) within averaging volume (V), where \mathbf{n} is a unit normal vector outward from S and n_i is its component in the i direction, u_i represents the three components of wind, i.e., u, v, and w. F_1 in Equation (14) is the horizontal component of F_i . This force represents momentum sinks of form drag and viscous skin-friction drag on the mean flow by the plant canopy elements. Following Thom (1975), the drag force per unit surface area may be expressed by the commonly used formula

$$F_1 = \rho C U u \quad (16)$$

where C is a drag coefficient for an obstacle element, and U is the mean windspeed defined as

$$U = \sqrt{u^2 + v^2 + w^2} \quad (17)$$

For applications to vegetation, (16) can be rewritten as

$$F_1 = \rho C_d A U u \quad (18)$$

where C_d is a drag coefficient for unit leaf-area density, and A is the leaf-area index density, which is the leaf-area index divided by the height of the shelter.

Integrating Equation (14) over the whole space occupied by the shelterbelt gives

$$\begin{aligned} & \int_0^{H_-} \langle \rho(u_{f+}^2 - u_{b-}^2) + \rho(\overline{u_{f+}^2} - \overline{u_{b-}^2}) + (p_{f+} - p_{b-}) \rangle dz - \\ & \int_{x_f^-}^{x_b^-} \rho \langle (u_0 w_0 - u_{H-} w_{H-}) - (\overline{u' w'_0} - \overline{u' w'_{H-}}) \rangle dx - F_D = 0 \end{aligned} \quad (19)$$

where F_D is drag force exerted on airflow by the whole shelterbelt. Equation (19) clearly shows the relationship between shelterbelt drag and the changes of pressure, wind, and turbulence across the shelterbelt.

Although the gradients of pressure and other variables are discontinuous on interfaces of the shelterbelt, pressure, windspeed, and turbulent flux themselves are continuous, i.e.,

$$\begin{aligned} p_{f-} &= p_{f+} ; & p_{b+} &= p_{b-} ; \\ u_{f-} &= u_{f+} ; & u_{b+} &= u_{b-} ; \\ u_{H+} &= u_{H-} ; & w_{H+} &= w_{H-} ; \\ \overline{u' w'_{H+}} &= \overline{u' w'_{H-}} ; \\ \overline{u_{f-}^2} &= \overline{u_{f+}^2} ; \\ \overline{u_{b+}^2} &= \overline{u_{b-}^2} ; \end{aligned} \quad (20)$$

Therefore, the sum of Equations (13) and (19) yields

$$F_D = - \int_{x_1}^{x_2} \rho (\overline{u'w'_t} - \overline{u'w'_0}) dx = \int_{x_1}^{x_2} (\tau_t - \tau_0) dx \quad (21)$$

i.e., shelterbelt drag causes surface turbulent flux reduction in the sheltered region, and increase in form drag due to the shelterbelt is eventually compensated by loss of surface friction. However, Equation (19) gives more physical insight than does equation (21). Mechanical drag itself exists only within the shelterbelt; however, this force influences flow outside the shelter through perturbations of the pressure, flow, and turbulence fields.

Equations (19) and (21) give a theoretical basis for determination of shelterbelt drag by indirect measurements. Equation (21) requires surface turbulent stress data with sufficient accuracy and resolution over the whole sheltered region with very large horizontal range as discussed by Seginer and Sagi (1971), Seginer (1972), and Taylor (1983). This requirement generally is difficult to meet. Direct measurement of surface turbulent stress is rather difficult. Previous investigators (Seginer and Sagi, 1971; Seginer, 1972) estimated the surface turbulent stress by measuring the mean windspeed in the undisturbed atmospheric boundary-layer and using Equation (2) to approximate the disturbed wake of the shelterbelt. This can lead to large errors because the vertical change of turbulent stress is extremely large in the wake boundary layer while turbulent stress is constant in undisturbed atmospheric boundary layer. We will discuss these in next section. Furthermore, measurement of stress to points up and down wind of the shelter where effect of the shelter is negligible demands measurements over a very large volume.

Equations (12) and (19), on the other hand, require measurements only over region IV in Figure 2, but these measurements must include profiles of pressure, windspeed, and turbulence. The assumptions that p_1 and p_2 are constant with height and that the pressure gradient in the horizontal direction is constant as used in Seginer and Sagi (1971) and Seginer (1972) can lead to large errors, and violate conservation of mass if w_i is set to zero. The simulated distribution of perturbed pressure as shown in Figure 3 shows that isobars are not perfectly vertical near the shelter, and therefore the pressure gradient is not constant with height. However, we can calculate the drag from Equation (19), if we have pressure, turbulence, and windspeed data measured at both lateral surfaces of the shelterbelt from surface to height H . In the following section, we use a nonhydrostatic shelterbelt turbulent flow model to study the interaction among drag, pressure, flow, and turbulence. We will focus on interfaces of the shelterbelt to study the relative importance of each physical process. We will illustrate that only the differences of pressure and horizontal mean wind across the shelterbelt in Equation (19) are needed to determine the drag.

3. Numerical Simulation: Drag and Pressure

Previous theoretical and numerical work has been restricted to consideration of the far wake where nonlinear interactions and perturbed pressure are weak enough to allow use of small perturbations and linearized methods (e.g., Taylor, 1985). Around the shelterbelt, however, strong perturbed pressure and very nonlinear interactions dominate the changes in momentum. We use a nonhydrostatic time-dependent shelterbelt turbulent flow model as

described in Wang and Takle (1995a, 1995b) to study the interactions among drag, pressure, flow, and turbulence in the entire domain.

As shown in Figure 2, the computational domain is from upstream 60 H to downstream 100 H, with a top at 12 H.

We follow Hoerner (1965) and define static pressure coefficients C'_{pHF} and C'_{pHB} as the differences of pressure between front (upwind) edge of the shelter (p_F) and back (down wind) edge of the shelter (p_B) from ambient pressure (p_{amb}), respectively, normalized by mean kinetic energy (MKE) of the undisturbed flow at height of the shelterbelt top.

$$\begin{aligned} C'_{pHF} &= \frac{P_F - P_{amb}}{0.5\rho u_H^2} \\ C'_{pHB} &= \frac{P_B - P_{amb}}{0.5\rho u_H^2} \end{aligned} \quad (22)$$

Figure 4 illustrates the changes of static pressure coefficient with height and shelterbelt porosity. As shown in Figure 4a for a loose shelterbelt with porosity of 91.6%, the absolute values of both C'_{pHF} and C'_{pHB} increase with height from the ground to $z=0.7$ H and then decrease with further increase of height. The maximum pressure drop across the shelter occurs at $z=0.6-0.8$ H (see Figure 3). However, the two curves shown in Figure 4a may be approximated by vertical lines with maximum departures of 0.01 (15%) and 0.012 (30%) for a vertically uniform shelterbelt.

The location of maximum C'_{pHF} moves down slightly with increasing shelterbelt density to $z=0.6$ H for a medium-dense shelterbelt having porosity of 52.3% (Figure 4b) and to $z=0.5$ for a dense shelterbelt with porosity of 0.5 H (Figure 4c). The maximum vertical changes are

0.08 (20%) and 0.25 (55%), respectively. C'_{pHF} decreases with height more rapidly near the shelterbelt top for dense shelters.

The location of the maximum absolute value of C'_{pHB} moves up to $z = 1.0 H$ for a shelterbelt with porosity of 61.8% (Figure omitted) and is at the shelterbelt top for medium-dense and dense shelterbelts (Figures 4b and 4c). The maximum vertical changes of C'_{pHB} are 0.11 (35%) and 0.39 (30%) for medium-dense and dense shelterbelts, respectively. The most obvious vertical change also is near the shelterbelt top where the absolute value of C'_{pHF} increases with increasing shelterbelt density. Absolute values of perturbed pressure both in front of and behind the shelter always increase with increasing shelterbelt density. For loose to medium-dense shelterbelts, C'_{pHF} increases slightly faster than the absolute value of C'_{pHB} ; however, the absolute value of C'_{pHB} increases much faster than C'_{pHF} for medium-dense to dense shelterbelts.

Static pressure-loss coefficient (C'_{pH}) across the belt is defined as

$$C'_{pH} = \frac{P_F - P_B}{0.5 \rho u_H^2} = C'_{pHF} - C'_{pHB} \quad (23)$$

For loose and medium-dense shelterbelts, C'_{pH} first increases with height, reaches its maximum at $z = 0.7 - 0.8 H$, then decreases with further increasing height (Figure 5a). However, with increasing shelterbelt density, the relative vertical change of C'_{pH} becomes smaller (Figures 5b and 5c), and becomes nearly a vertical line for porosity of 27% (figure omitted). With further increase of shelterbelt density, C'_{pH} near the shelterbelt top rapidly increases

because the absolute value of the perturbed pressure near the shelterbelt top rapidly increases for very dense shelterbelts (Figure 5c).

Equation (14) may be rewritten by use of Equations (23) and (6) to give

$$C_{dH} = C'_{pH} - \left(u \frac{\partial u}{\partial x} - w \frac{\partial u}{\partial z} - \frac{\overline{\partial u^2}}{\partial x} - \frac{\overline{\partial u'w'}}{\partial z} \right) / (0.5u_H^2) \quad (24)$$

which demonstrates the relationship between drag and static pressure coefficients. The second, third, fourth, and fifth terms on the right hand side of Equation (24) are associated with horizontal advection, vertical advection, horizontal and vertical turbulent transport, respectively. For laminar flow and solid obstacles, these four terms vanish giving the drag coefficient equal to the static pressure-loss coefficient. However, for a porous shelterbelt embedded in atmospheric turbulent boundary layer, all four terms are non-zero.

The simulated differences between drag and static pressure-loss coefficients are shown in Fig.5. For a loose shelterbelt with porosity of 91.6% (Figure 5a), $C_{dH} < C'_{pH}$ below $z=0.25$ H. Above $z=0.25$ H, $C_{dH} > C'_{pH}$, and the departure increases with increasing height. For a medium-dense shelterbelt with porosity of 52.3% (Figure 5b), the height at which $C_{dH} = C'_{pH}$ has decreased to $z=0.18$ because large increase of drag with increasing shelterbelt density makes turbulence dominant. These characteristics is more clearly seen for a more dense shelterbelt with porosity of 9.9% shown in Figure 5c. From Figures 5a-c, we also can see that with increasing shelterbelt density, the relative difference between C_{dH} and C'_{pH} decreases below the shelterbelt top, but increases at the shelterbelt top. We averaged C_{dH} and C'_{pH} over the shelterbelt height (H) and calculated mean departures of C_{dH} from C'_{pH} for different

densities as listed in Table 1. With decreasing shelterbelt porosity from 99% to 10%, the mean departure decreases from 29% to 8%.

The departures are in part due to changes of dynamic pressure across porous shelterbelts. Total pressure equals to sum of static pressure and dynamic pressure (mean kinetic energy), and we define the total pressure-loss coefficient (C_{pH}) as

$$C_{pH} = \frac{(p_F + 0.5\rho u_F^2) - (p_B + 0.5\rho u_B^2)}{0.5\rho u_H^2} \quad (25)$$

where u_F and u_B are windspeed immediately in front of the shelter and behind the shelter, respectively. Therefore, Equation (24) may be rewritten as

$$C_{dH} = C_{pH} - \left(-w \frac{\partial u}{\partial z} - \frac{\partial \overline{u'^2}}{\partial x} - \frac{\partial \overline{u'w'}}{\partial z} \right) / (0.5u_H^2) \quad (26)$$

The difference between drag and total pressure-loss coefficients is shown in Figure 6. Comparison of Figure 6a and Figure 5a shows that total pressure-loss coefficient decreases below $z=0.2 H$, because windspeed increases rather than decreases due to strong pressure difference across the belt. Above that level, C_{pH} is always larger than C'_{pH} . Obvious differences between C_{dH} and C_{pH} exist only near the shelterbelt top and, somewhat less obviously, near the shelterbelt bottom.

Figure 7 shows the vertical profiles of contributions to drag from advection and horizontal and vertical turbulent transports normalized by mean kinetic energy of the undisturbed flow at height of the shelterbelt top. The maximum changes occur near the shelterbelt top. Vertical turbulent transport has the largest contribution among the three terms

and increases rapidly with increasing shelterbelt density (Figures 7b and 7c). The depth downward into the shelterbelt to which this strong canopy turbulent transport extends decreases with increasing shelterbelt density. Vertical advection has its largest relative contribution for medium-dense shelterbelts. For a loose shelterbelt, vertical velocity is very small due to small drag, and therefore, vertical advection transport is small. On the other hand, a very dense shelterbelt, although its drag is very large, does not permit a large enough vertical velocity to support vertical transport. Horizontal turbulent transport is always smallest.

We averaged the drag coefficient C_{dH} and total pressure-loss coefficient C_{pH} over the whole shelterbelt height and obtained mean drag and pressure-loss coefficients as listed in Table 2. From Table 2, we can see that the departure of total pressure-loss coefficient from drag coefficient is quite small, the maximum departure being about 8% for very dense shelterbelts. Comparison to Table 1 shows that dynamic pressure has larger contribution for loose and medium-dense shelterbelts, but less contribution for dense shelterbelts. For example, both Tables 1 and 2 list the departures to be 8% for porosity of 10%. In summary, Equation (26) may be approximated as

$$C_{dH}=C_{pH}=C'_{pH}+\frac{u_F^2-u_B^2}{u_H^2} \quad (27)$$

We use this analysis of behavior and partition of the drag in the near wake to examine the stress-loss method proposed by Seginer and Sagi (1971) and Bradley and Mulhearn's (1983) drag plate data. Figure 8 gives the simulated horizontal profiles of turbulent stress at $z=0.1, 0.2, 0.3, 0.4, 0.5 H$ levels for a medium-dense shelterbelt with porosity of 50%. The

shelterbelt reduces the turbulent stress, and the horizontal range of turbulent stress reduction decreases with increasing height. At $z=0.4 H$, turbulent stress is reduced out to $8 H$ in the lee beyond which ($8-16 H$) the turbulent stress increases. This is in agreement with previous experimental results as summarized in McNaughton (1988). Figure 8 clearly shows that a constant flux relationship no longer exist around shelterbelts.

We also calculated the drag coefficients at different heights according to Equation (3). Results are listed in Table 3, where C_{dH0} , C_{dH1} , C_{dH2} , C_{dH3} , and C_{dH4} are the calculated results at $z=0.0 H$, $z=0.1 H$, $z=0.2 H$, $z=0.3 H$, and $z=0.4 H$, respectively. C_{dH0} increases with decrease of porosity from 99% to 36% but hardly changes beyond 36%. The maximum C_{dH0} is 0.1020 for a dense shelterbelt with porosity of 10%. C_{dH1} , C_{dH2} , C_{dH3} , and C_{dH4} increase with decreasing porosity to maxima for medium-dense shelterbelts, then decrease with further decreasing porosity. The calculated drag coefficients rapidly decrease with increasing height. Negative values occur at higher levels for either very dense or loose shelterbelts where total turbulent stress increases due to shelterbelt wake. The drag coefficients calculated by the stress-loss method are considerably smaller than these calculated from the pressure loss (Table 2). We also use Bradley and Mulhearn's (1983) field drag-plate-measured turbulent stress data from the region $0-50 H$ in the lee of a 50% porosity fence as shown in their Figure 4. Based on these data, we calculate the corresponding drag coefficient to be 0.061 according to Equation (3). This value is reasonably comparable to 0.0991 calculated directly from our simulated turbulent stress as listed in Table 3, if we consider that the measured turbulent stress data are only for leeward $0-50 H$ and do not include windward and far-lee turbulent stress data.

Therefore, we think that our simulation of surface turbulent stress is reasonable and that the stress-loss method is questionable.

All previous investigators who calculated drag coefficient by stress-loss method used the procedure summarized in Equations (4) and (5) where the assumption of a logarithmic windspeed in the wake was made. To examine this assumption, we calculate the drag coefficients (Table 4) based on our simulated horizontal profiles of mean windspeed reduction as shown in Figure 9. We plot data only from $x=-10$ to $30 H$ to clearly show the wind reduction, but the horizontal range for calculating drag coefficient is much larger. We have compared our simulation of horizontal wind profile with the observations and found good agreement (Wang and Takle, 1995b). Table 4 gives C_{dH1} , C_{dH2} , C_{dH3} , C_{dH4} , and C_{dH5} calculated according to Equation (5) with the simulated wind reduction profiles at $z=0.1, 0.2, 0.3, 0.4$, and $0.5 H$, respectively. All the drag coefficients calculated by this procedure increase with decreasing porosity, with a maximum value of 0.1755 for a dense shelterbelt with porosity of 10%.

The calculated results compared well with those of previous investigators (Seginer and Sagi, 1971; Seginer, 1975; Miller, et al., 1975) based on measured wind reduction profile. However, comparison with Table 3 shows significant differences that arise from the assumption of a logarithmic wind profile in the wake. The drag coefficients calculated from our simulated wind-reduction data (Table 3) and those calculated from measured wind-reduction profiles of previous investigators are larger than those calculated directly from our simulated stress-loss data (Table 3) and the drag coefficient calculated from drag-plate-measured stress-loss data as mentioned above. However, those drag coefficients listed in Table

4 are still much smaller than those listed in Table 2 which are thought to be closer to actual values. Therefore, the drag coefficients determined by the actual stress-loss method of Equation (3) are even smaller than those previous investigators estimated by using Equation (5). On the other hand, the height of anemometers is different for different field measurements, and Seginer and Sagi (1971) assumed that this would have no significant effect on the calculated drag coefficient as long as the anemometers were positioned below $0.5 H$. As shown in Table 4, however, calculated values of C_{dH} decrease significantly with increasing height. The decrease in the calculated drag coefficient is faster for loose shelterbelts than dense shelterbelts (the relative differences between C_{dH1} and C_{dH5} are 40.5% and 22.3% for porosities of 99% and 10%, respectively). In Figure 4 of Seginer and Sagi (1971), a dense shelterbelt with porosity of 0.15-0.20 was used to illustrate the variation of wind-reduction profiles with height, so the height dependence should be minimized. On the other hand, their isolated measurements of windspeed did not reveal recirculation, which would have been likely for this porosity (Wang and Takle, 1995b). Therefore, comparison between drag coefficients calculated from wind-reduction profiles at different heights should be used with extreme caution.

4. Summary

We have analyzed the balance of forces perturbed by the drag near the shelterbelt and evaluated them quantitatively by numerical simulation. The mechanical drag exerted by shelterbelts on airflow only exists within shelterbelts. This drag causes complicated interactions among pressure, flow, and turbulence within the shelter but these processes

transmit the effects of drag beyond the boundaries of obstacle. We have partitioned the drag into perturbed pressure, horizontal and vertical mean transports, and horizontal and vertical turbulent transports. The drag exerted on airflow by the belt is balanced mainly by static pressure difference and horizontal advection across the belt. With increasing shelterbelt density, horizontal advection decreases, leaving static pressure difference to balance the drag. Total (dynamic plus static) pressure difference across the belt is close to the drag, and the vertically-averaged difference between drag and pressure coefficient is within 8%. Generally, horizontal turbulent transport and vertical convection are smaller. However, great difference exists near the top and bottom of the belt, especially near the top, where vertical turbulent transport plays an important role. Therefore, we can estimate drag of a shelterbelt with high accuracy by measuring only the changes of static pressure and windspeed across the shelterbelt.

High-resolution and physically consistent data from numerical simulations allow us to examine previous drag-calculation methods. We also have systematically evaluated the surface stress-loss method proposed by Seginer and Sagi (1971), and found that the results from this method is unreasonably small. Moreover, we find that assuming logarithmic profile of windspeed in the wake to derive the turbulent stress can lead to large errors. The coefficient calculated by this method heavily depends on the measurement height. The drag coefficient calculated from turbulent stress derived from mean wind-reduction profile is larger than calculated directly from the simulated turbulent stress of higher-order turbulent closure. However, it is still unreasonably small compared to the actual value. The coefficient calculated by this method and both procedures heavily depends on the measurement height. Our

calculation using drag-plate-measured turbulent stress data gave the same results. Therefore, the stress-loss method to determine drag coefficient is questionable.

Acknowledgment This research was supported by USDA/CSRS NRI Competitive Grant # 93371018954 and USDOE/NIGEC grant No. DE-FC0390ER61010.

References

- Bradley, E.F. and Mulhearn, P.J.: 1985, 'Development of Velocity and Shear Stress Distributions in the Wake of a Porous Fence', *J. Wind Eng. Ind. Aerodyn.*, **15**, 145-156.
- De Bray, R.G.: 1971, 'Protection by Fences', *Wind Effects on Buildings and Structures Seminar*, 25-27 May 1971, Univ. of Auckland, New Zealand.
- Hagen, L.J. and Skidmore, E.L.: 1971, 'Turbulent Velocity Fluctuations and Vertical Flow as Affected by Windbreak Porosity', *Trans. ASAE*, **14**, 634-637.
- Heisler, G.M. and Dewalle, D.R.: 1988, 'Effects of Windbreak Structure on Wind Flow', *Agriculture, Ecosystems and Environment*, **22/23**, 41-69.
- Hoerner, S.F.: 1965, *Fluid Dynamic Drag*, Library of Congress Catalog Card Number 64-19666.
- Jacobs, A.G.: 1985, 'The Normal-Force Coefficient of a Thin Closed Fence', *Boundary-Layer Meteorol.*, **32**, 329-335.
- McNaughton, K.G.: 1988, 'Effects of Windbreaks on Turbulent Transport and Microclimate', *Agriculture, Ecosystems and Environment*, **22/23**, 17-39.

- Miller, D.R., Rosenberg, N.J., and Bagley, W.T.: 1975, 'Wind Reduction by a Highly Permeable Tree Shelterbelt', *Agric. Meteorol.*, **14**, 321-333.
- Rosenberg, N.J., Blad, B.L., and Verma, S.B.: 1983, *Microclimate: The Biological Environment*, John Wiley, New York, 495pp.
- Seginer, I.: 1975, 'Atmospheric-Stability Effect on Windbreak Shelter and Drag', *Boundary-Layer Meteorol.*, **8**, 383-400.
- Seginer, I. and Sagi, R.: 1971, 'Drag on a Windbreak in Two-Dimensional Flow', *Agric. Meteorol.*, **9**, 323-333.
- Seginer, I.: 1972, 'Windbreak Drag Calculated From the Horizontal Velocity Field', *Boundary-Layer Meteorol.*, **3**, 87-97.
- Tani, N: 1952, 'On the Wind Tunnel Test of the Model Shelter-Hedge', *Bull. Nat. Inst. Agri. Sci. Japan*, Ser.A, **6**, 75-80.
- Taylor, P.A.: 1983, 'On Wakes and the Net Forces Produced by Surface-Mounted Obstacles in Neutrally Stratified Atmospheric Boundary Layers', *Boundary-Layer Meteorol.*, **27**, 393-412.
- Taylor, P.A.: 1988, 'Turbulent Wakes in the Atmospheric Boundary Layer', *Flow and Transport in the Natural Environment: Advances and Applications*, Steffen, W.L. and O.T. Denmead, Eds., Springer-Verlag, Berlin, 270-292.
- Thom, A.S.: 1975, *Vegetation and the Atmosphere*, Vol.1. Academic Press, pp.1-278.
- van Eimern, J., Karschon, R., Razumova, L.A., and Robertson, G.W.: 1964, *Windbreaks and Shelterbelts*, World Meteorological Organization Technical Note No.59, pp.188.

- Wang, H. and Takle, E.S.: 1995a, 'Boundary-Layer Flow and Turbulence Near Porous Obstacles. I. Derivation of a General Equation Set For a Porous Medium', *Boundary-Layer Meteorol.*, **74**, 73-88.
- Wang, H. and Takle, E.S.: 1995b, 'A Numerical Simulation of Boundary-Layer Flows Near Shelterbelts', *Boundary-Layer Meteorol.*, **75**, 141-173.
- Woodruff, N.P., Fryrear, D.W., and Lyles, L.: 1963, 'Engineering Similitude and Momentum Transfer Principles Applied to Shelterbelt Studies', *Trans. Am. Soc. Agric. Eng.*, **6**, 41-47.

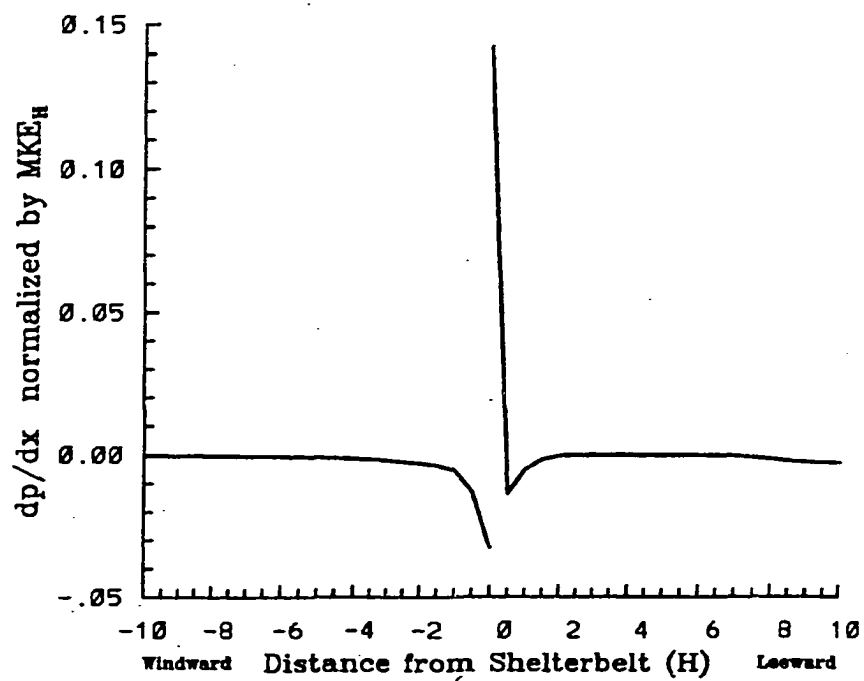


Figure 1. Change of surface perturbed static pressure gradient, normalized by dynamic pressure (mean kinetic energy) of the undisturbed flow at the shelterbelt height H .

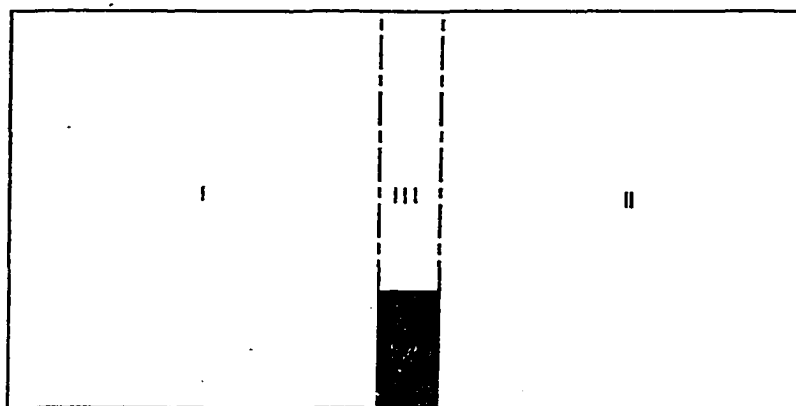


Figure 2. Schematic diagram of integration regions for momentum equation.

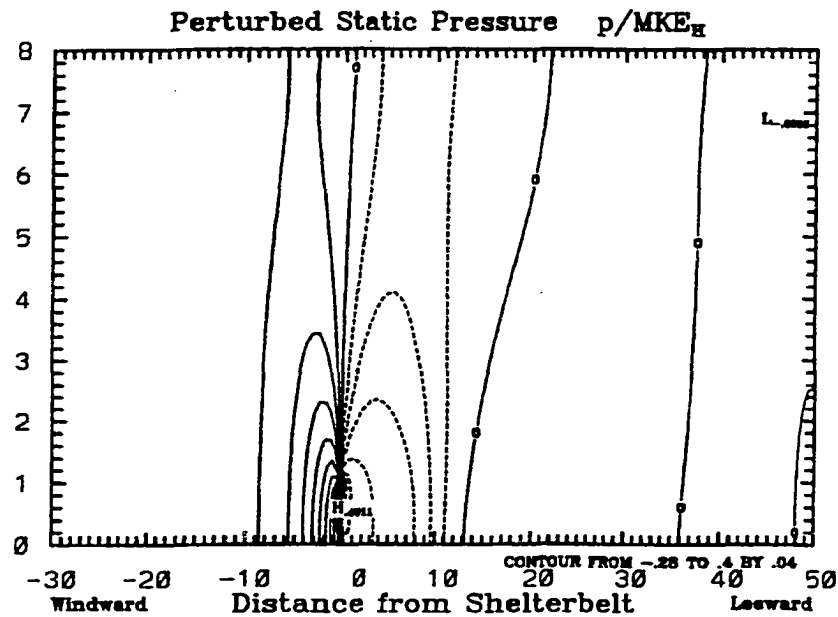


Figure 3. Spatial distribution of the perturbed static pressure around a shelterbelt, normalized by dynamic pressure (mean kinetic energy) of the undisturbed flow at the shelterbelt height H .

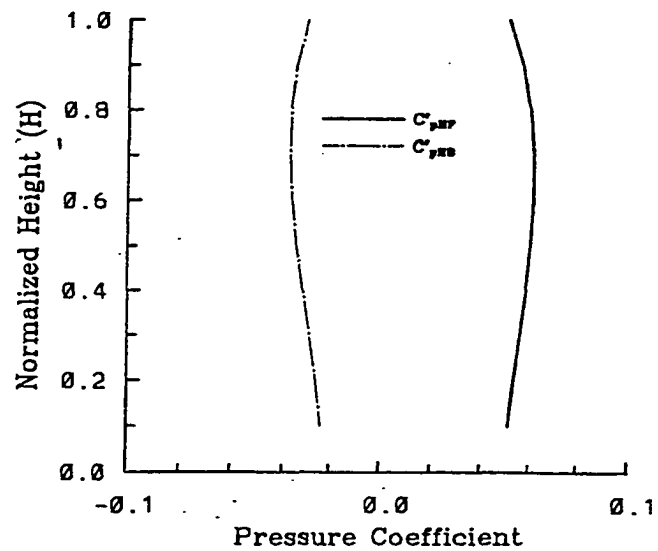


Figure 4a. Static pressure coefficients in the front of the shelter (C'_{pHF}) and behind the shelter (C'_{pHB}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

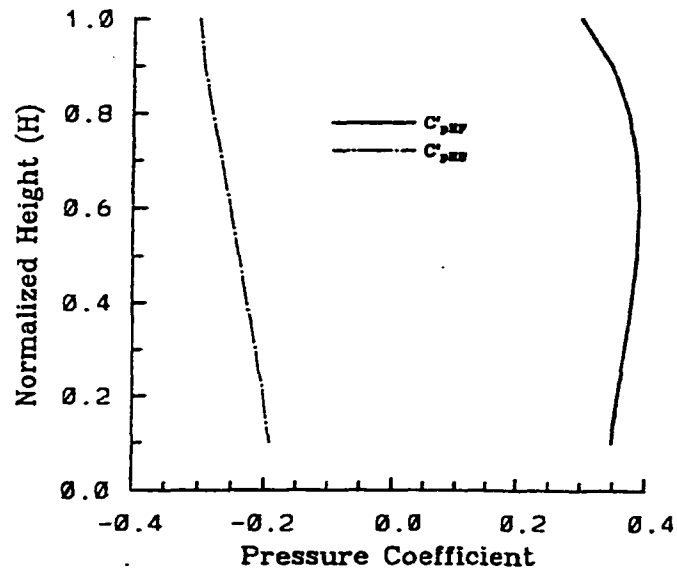


Figure 4b. Static pressure coefficients in the front of the shelter (C'_{pHF}) and behind the shelter (C'_{pHB}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

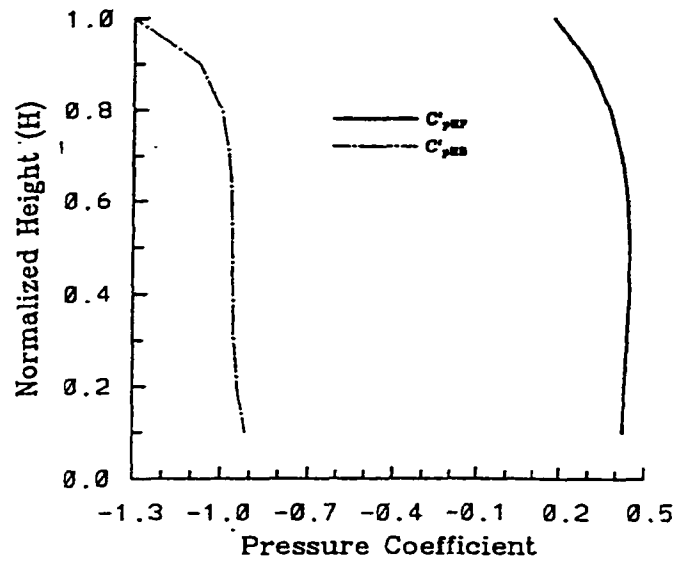


Figure 4c. Static pressure coefficients in the front of the shelter (C'_{pHF}) and behind the shelter (C'_{pHB}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

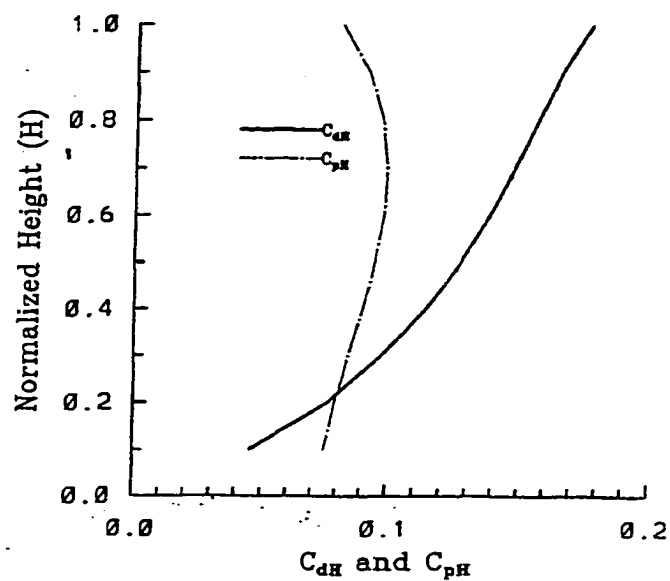


Figure 5a. Comparison of drag coefficient (C_d) and static pressure-loss coefficient (C'_{pH}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

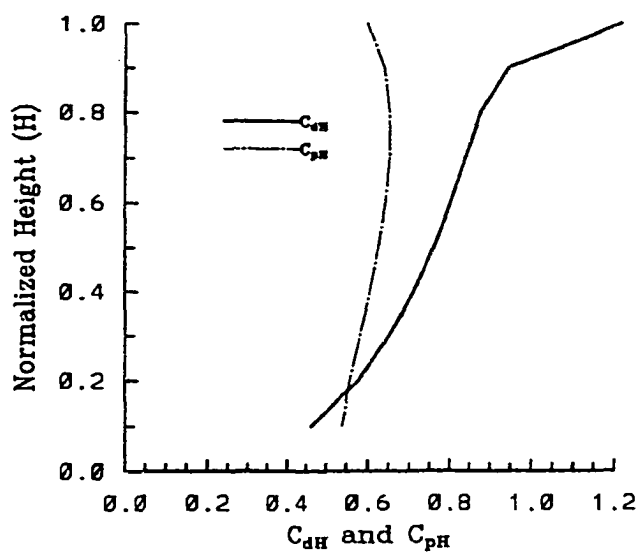


Figure 5b. Comparison of drag coefficient (C_d) and static pressure-loss coefficient (C'_{pH}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

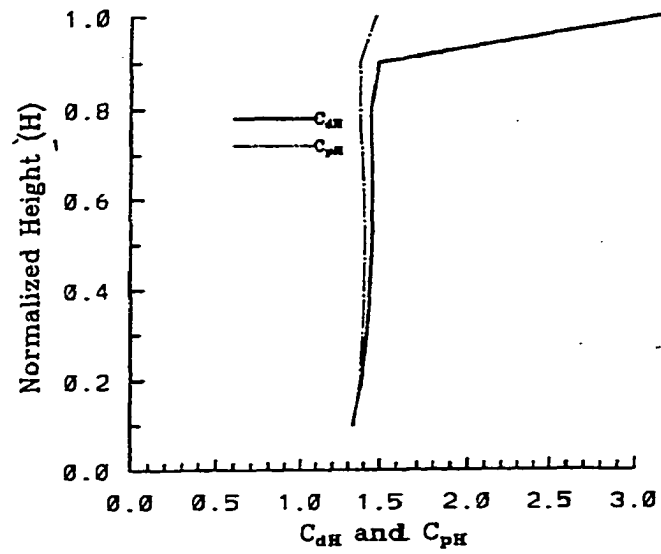


Figure 5c. Comparison of drag coefficient (C_d) and static pressure-loss coefficient (C'_{pH}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

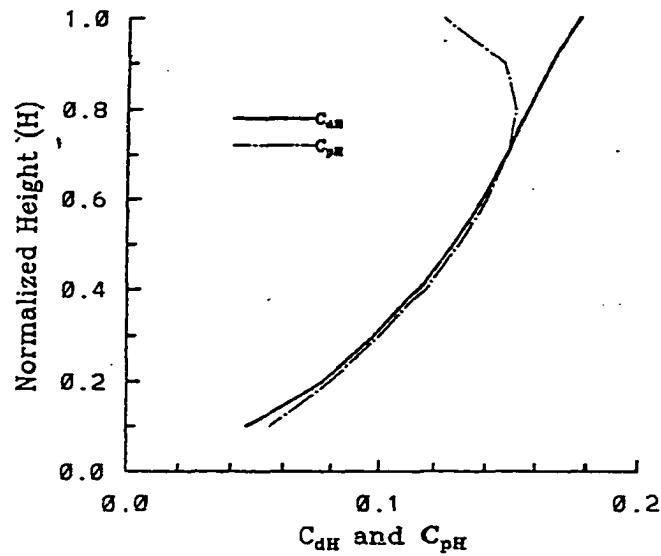


Figure 6a. Comparison of drag coefficient (C_d) and total pressure-loss coefficient (C_{pH}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

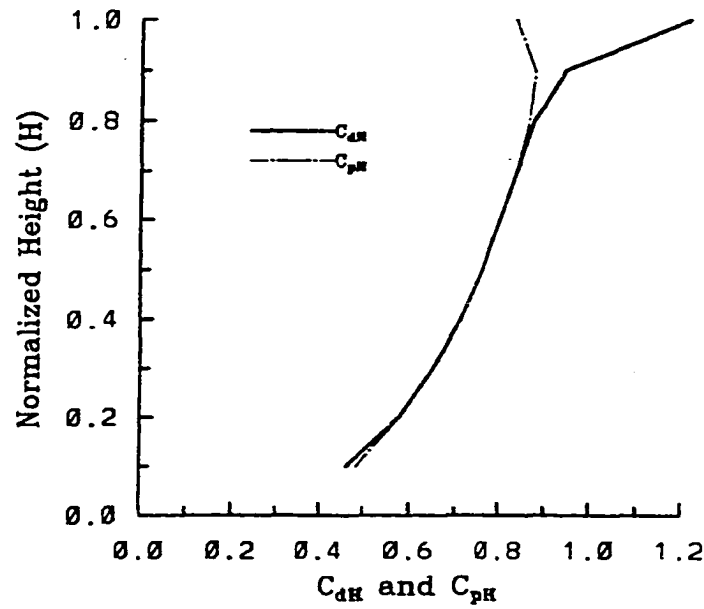


Figure 6b. Comparison of drag coefficient (C_d) and total pressure-loss coefficient (C_{pH}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

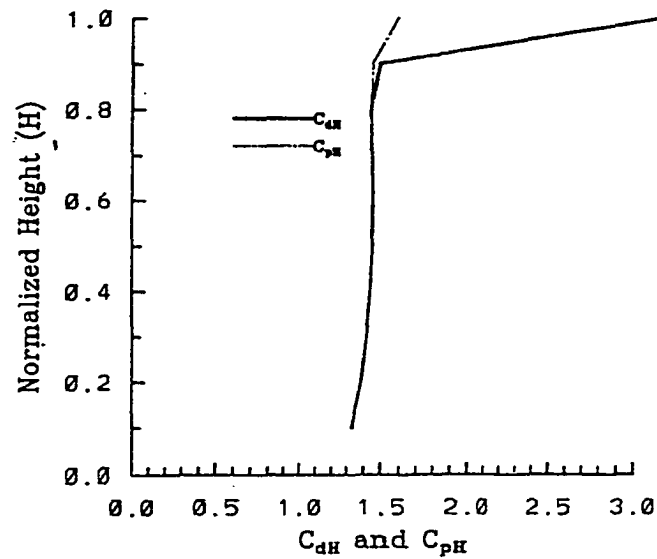


Figure 6c. Comparison of drag coefficient (C_d) and total pressure-loss coefficient (C_{pH}) for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c).

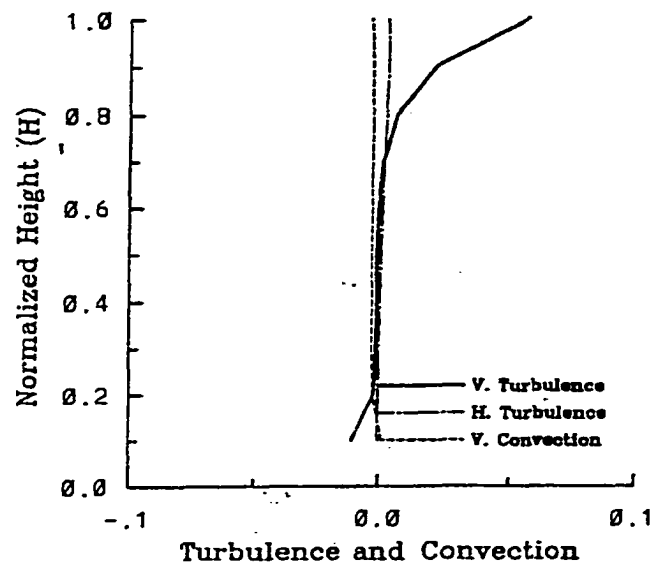


Figure 7a. Contributions of vertical advection, horizontal and vertical turbulent transports in partition of the drag for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c). Normalized by dynamic pressure (mean kinetic energy) of the undisturbed flow at the shelterbelt height H .

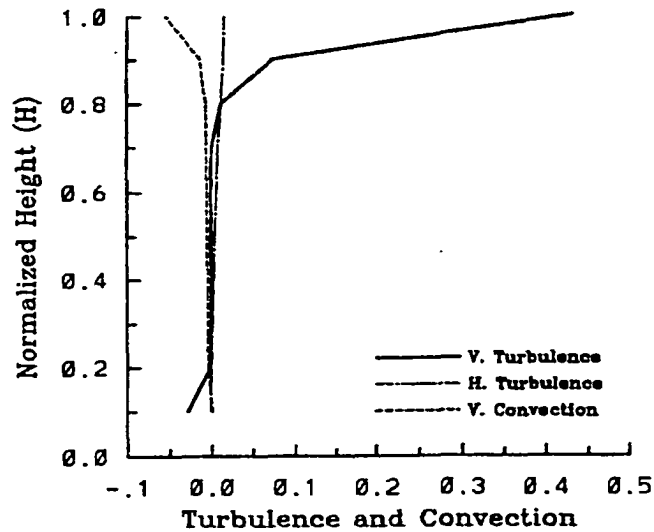


Figure 7b. Contributions of vertical advection, horizontal and vertical turbulent transports in partition of the drag for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c). Normalized by dynamic pressure (mean kinetic energy) of the undisturbed flow at the shelterbelt height H .

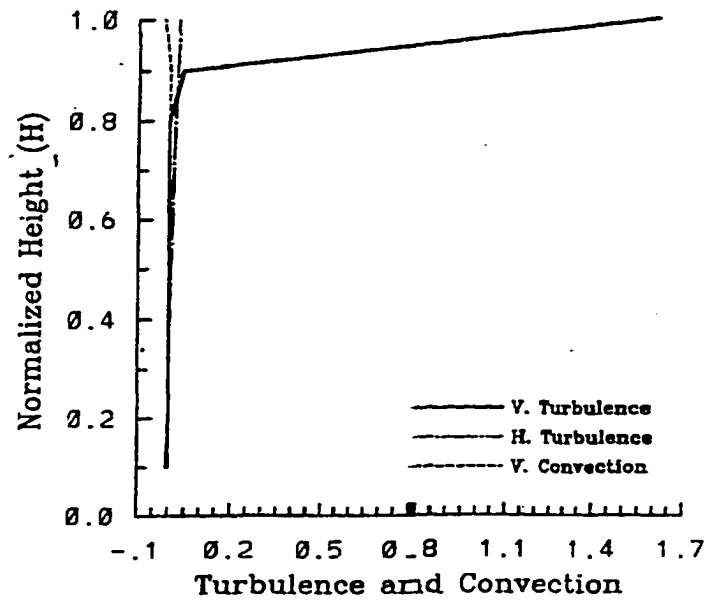


Figure 7c. Contributions of vertical advection, horizontal and vertical turbulent transports in partition of the drag for shelterbelts with porosities of 91.6% (a), 52.3% (b), and 9.9% (c). Normalized by dynamic pressure (mean kinetic energy) of the undisturbed flow at the shelterbelt height H .

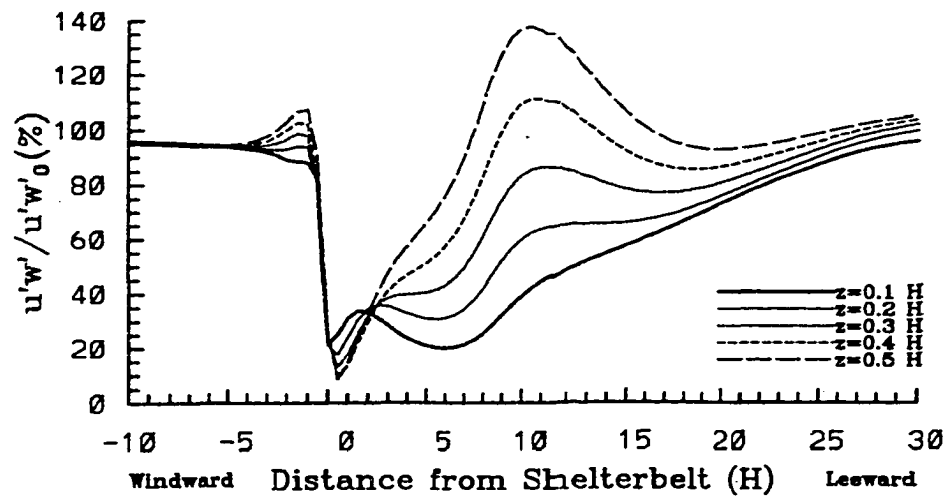


Figure 8. Perturbations of simulated turbulent stress for calculating drag by turbulent stress-loss method proposed by Seginer and Sagi (1971), normalized by turbulent stress of the upstream undisturbed flow.

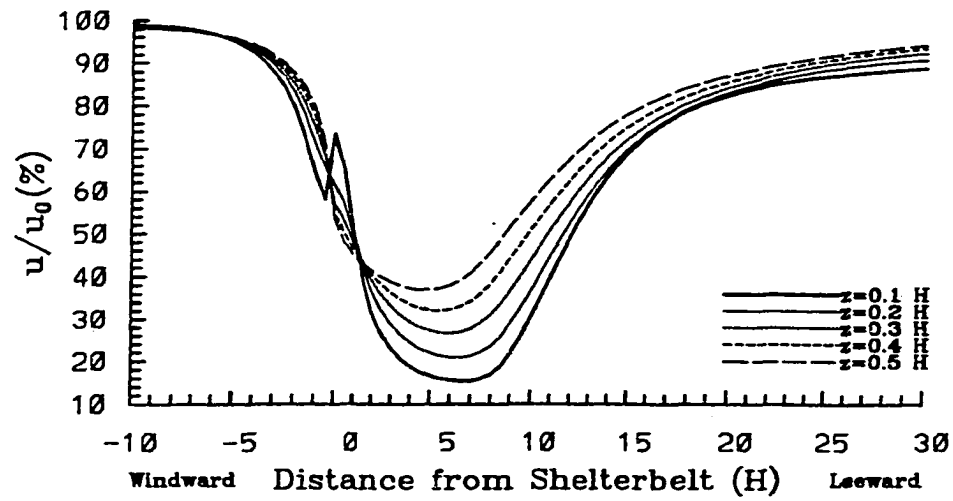


Figure 9. Simulated windspeed reduction profiles for calculating drag by turbulent stress-loss method proposed by Seginer and Sagi (1971), normalized by windspeed of the upstream undisturbed flow.

Table 1. Mean departure of averaged static pressure-loss coefficient from drag coefficient

Porosity (%)	99	92	81	73	62	50	44	36	27	21	14	10
Departure (%)	29	28	26	23	20	17	15	13	12	10	8	8

Table 2. Mean departure of averaged total pressure-loss coefficient from drag coefficient

Porosity (%)	99	92	81	73	62	50
C_{dH}	.0128	.1156	.2926	.4108	.6057	.7989
C_{pH}	.0122	.1126	.2820	.3995	.5901	.7731
Departure (%)	5	3	4	3	3	3
Porosity (%)	44	36	27	21	14	10
C_{dH}	.9070	1.0242	1.1576	1.2662	1.3798	1.4451
C_{pH}	.8784	.9882	1.1108	1.2081	1.2871	1.3283
Departure (%)	3	4	4	5	7	8

Table 3. Drag coefficient determined by turbulent stress reduction method where turbulent stress calculated by the model directly from higher-order turbulence closure

Porosity (%)	99	92	81	73	62	50
C_{dH0}	.0016	.0174	.0455	.0649	.0865	.0991
C_{dH1}	.0011	.0130	.0350	.0497	.0668	.0733
C_{dH2}	.0006	.0091	.0254	.0364	.0493	.0507
C_{dH3}	.0002	.0050	.0155	.0231	.0320	.0293
C_{dH4}	-.0003	.0004	.0047	.0088	.0137	.0075
Porosity (%)	44	36	27	21	14	10
C_{dH0}	.1011	.1015	.1008	.1008	.1015	.1020
C_{dH1}	.0724	.0689	.0641	.0602	.0565	.0549
C_{dH2}	.0476	.0414	.0325	.0252	.0179	.0145
C_{dH3}	.0244	.0156	.0333	-.0071	-.0175	-.0226
C_{dH4}	.0108	-.0099	-.0246	-.0375	-.0509	-.0577

Table 4. Drag coefficient determined by turbulent stress reduction method where turbulent stress calculated indirectly from horizontal profiles of mean windspeed reduction

Porosity (%)	99	92	81	73	62	50
C_{dH1}	.0042	.0318	.0752	.1010	.1272	.1508
C_{dH2}	.0036	.0284	.0695	.0940	.1192	.1425
C_{dH3}	.0032	.0257	.0642	.0871	.1111	.1322
C_{dH4}	.0028	.0233	.0592	.0806	.1033	.1234
C_{dH5}	.0025	.0211	.0544	.0743	.0957	.1150
Porosity (%)	44	36	27	21	14	10
C_{dH1}	.1576	.1628	.1674	.1707	.1738	.1755
C_{dH2}	.1478	.1526	.1571	.1605	.1639	.1659
C_{dH3}	.1383	.1428	.1470	.1503	.1538	.1558
C_{dH4}	.1293	.1335	.1375	.1406	.1439	.1459
C_{dH5}	.1207	.1248	.1284	.1212	.1344	.1363

CHAPTER 6. MODELING THE EVAPOTRANSPIRATION AND ENERGY PARTITION OF INHOMOGENEOUS AGROECOSYSTEMS

A paper published in the 22nd Conference on Agricultural & Forest Meteorology

and the 12th Conference on Biometeorology and Aerobiology

A paper to be submitted to Journal of Applied Meteorology

Hao Wang, E.S. Takle, and G.M. Prabhu

Department of Computer Science and Department of Geological & Atmospheric Sciences

Iowa State University, Ames, Iowa 50011

Abstract

Water is the most important resource for food production; water consumption in irrigated agriculture commonly accounts for 80-85% of all agricultural use, with on-farm efficiencies estimated to be in the range of 10-40%. Numerous researchers focused on measurements of water usage under the sheltered environment, but with some inconsistent results: some reported that shelters preserved water while others reported an increase of measured evapotranspiration under shelter. We developed a high-resolution computer simulation system to investigate for the first time by use of basic conservation laws, the physical processes governing evapotranspiration under shelter and have been able to resolve some historical inconsistencies. Agricultural practices such as shelterbelts can significantly

affect evapotranspiration. However, the effects are not straightforward, and our simulations show the complicated temporal and spatial variability of both latent and sensible heat fluxes. We have demonstrated that soil moisture controls not only the magnitude of evapotranspiration but also the direction of evapotranspiration-shelter effect: shelterbelts lead to evapotranspiration decrease for wet soil, increase for dry soil, and increase around noon but decrease in the morning and afternoon for moderately wet soil.

1. Introduction

Water consumption in irrigated agriculture commonly accounts for 80-85% of all agricultural use, with on-farm efficiencies estimated to be in the range of 10-40%. Because crop production is limited more often by water than anything else, it is necessary to increase crop water-use efficiency to increase crop yields (Sturrock, 1988). Agroforestry has been seen as a possible means of preserving forest resources, increasing agricultural production, and reducing deforestation (Von Maydell, 1987).

Shelterbelts and windbreaks have been widely used to improve agricultural sustainability. Reduction of evaporation of soil moisture and transpiration of plants is one of the benefits of belts in both warm dry and cool wet periods (Gagarin, 1949). Of all microclimatic influences of shelterbelts, enhanced soil moisture may be the main reason for the observed increased yields in the wind sheltered areas. Effect of belts on moisture balance may well be more important than the aerodynamic effect of wind protection about which more is known. Reductions of evaporation in the lee of windbreaks of between 10 and 40% have been measured by pan and Piche evaporimeters (Bates, 1911; Long and Persaud, 1988).

Miller et al. (1973) reported that during six days of measurements, shelter caused a mean 20% decrease in evapotranspiration. Observations in large-scale shelterbelt networks in China showed regional evaporation was reduced by 14%. Windbreaks and shelterbelts have been suggested as practical means to increase water-use efficiency of sheltered crops (Rosenberg, 1967). However, Marshall (1967) noted several exceptions and George (1971) presented data indicating increases in seasonal evapotranspiration by shelter. Dixon and Grace (1984) demonstrated transpiration rates increased with decreasing wind speed. Recent work by Brenner et al. (1995) also does not support the hypothesis that water is conserved behind a windbreak.

Although this problem has been addressed by numerous researchers, most measurements of evaporation are limited to the assessment of the capacity for evaporation and have been carried out with atmometers or evaporimeters. However, plants can close their stomata to reduce transpiration, so we must distinguish between the influence of belts on potential evaporation and evapotranspiration.

The effect of windbreaks on evaporation is complicated by turbulence induced by the barrier, barrier porosity, and availability of water to evaporation surfaces. Possible consequences of shelter for water use are several and difficult to predict, and may not always be beneficial to crop growth in water-limited environments. Therefore, it is essential to understand how evapotranspiration responds to shelter.

This study evaluates the temporal and spatial variability of shelter effects on evapotranspiration. Our goal is to assess the impact of agricultural and forestry practices on large-scale, mesoscale, and local climate changes and interactions.

2. The Model

2.1 Shelterbelt turbulent flow model

Our shelterbelt aerodynamic turbulence model and simulation domain are described in previous papers (Wang and Takle, 1995a, 1995b, 1995c, 1995d). Shelterbelts severely distort surface flow and generate a complex turbulence structure characterized by a triangle-shaped "quiet" zone of turbulence extending from the top of the shelter to the surface on the leeward side at 8 H (H is shelterbelt height) from the belt and a wake zone of significantly increased turbulence above the quiet zone (Wang and Takle, 1995d). Denser shelterbelts also cause separation and reattachment of streamlines and generate recirculations (Wang and Takle, 1995b). All these affect the transfer of heat and mass and change the partition of surface energy, including evapotranspiration.

2.2 Soil-vegetation-atmosphere processes

We calculate sensible and latent heat fluxes by using the electrical resistance analogue:

$$Flux = \frac{Potential\ difference}{Resistance} \quad (1)$$

We calculated both C3 (soy bean) and C4 (corn) vegetation situations and reported here the results of soy bean. For sensible heat flux (H_s), the potential difference is the temperature difference between the leaf surface or ground and air; for latent heat flux (LE), the potential difference is the vapor pressure difference between the leaf surface or ground and air. These values are connected to soil moisture availability, which is predicted by soil moisture budget equations, in terms of ratio of actual soil water to saturated soil water (m), following Sellers

et al. (1986). The shelter is simulated as a barrier to the flow and does not participate in the balances of moisture and energy. The vegetated surface leeward and windward of the shelter has no mechanical influence on the flow field except through its value of surface roughness.

3. Results and Discussion

We simulated the effects of shelterbelts on evapotranspiration and heat flux and their dependence on soil moisture availability and shelterbelt structure as well as atmospheric, soil, and vegetation conditions. We present only the simulated distributions of evapotranspiration and heat flux and the effects of soil moisture availability and shelterbelt density. The computational domain including both windward and leeward sides is much larger than that shown in Figs. 1 and 2. Because the effects of shelterbelts occur within $20 H$ leeward from the shelter, we plot only that part of the domain.

3.1 Temporal and spatial variability of evapotranspiration and heat flux as a function of soil moisture availability

For a very dry soil ($m=0.2$), as shown in Fig. 1a, shelterbelts cause a decrease in sensible heat flux (H_s). This effect extends to $15 H$ leeward from the belt, and the maximum decrease of H_s occurs at $x=6 H$. The shelter effect on H_s also changes significantly with time, and the maximum effect occurs at $t=1300$ LST. Concurrently, latent heat flux (LE) increases in the sheltered zone, with maximum increase at $x=6 H$ and $t=1100$ LST. Although the evapotranspiration is very small for dry surface, LE in the sheltered zone is as large as 122% of that in the unsheltered zone.

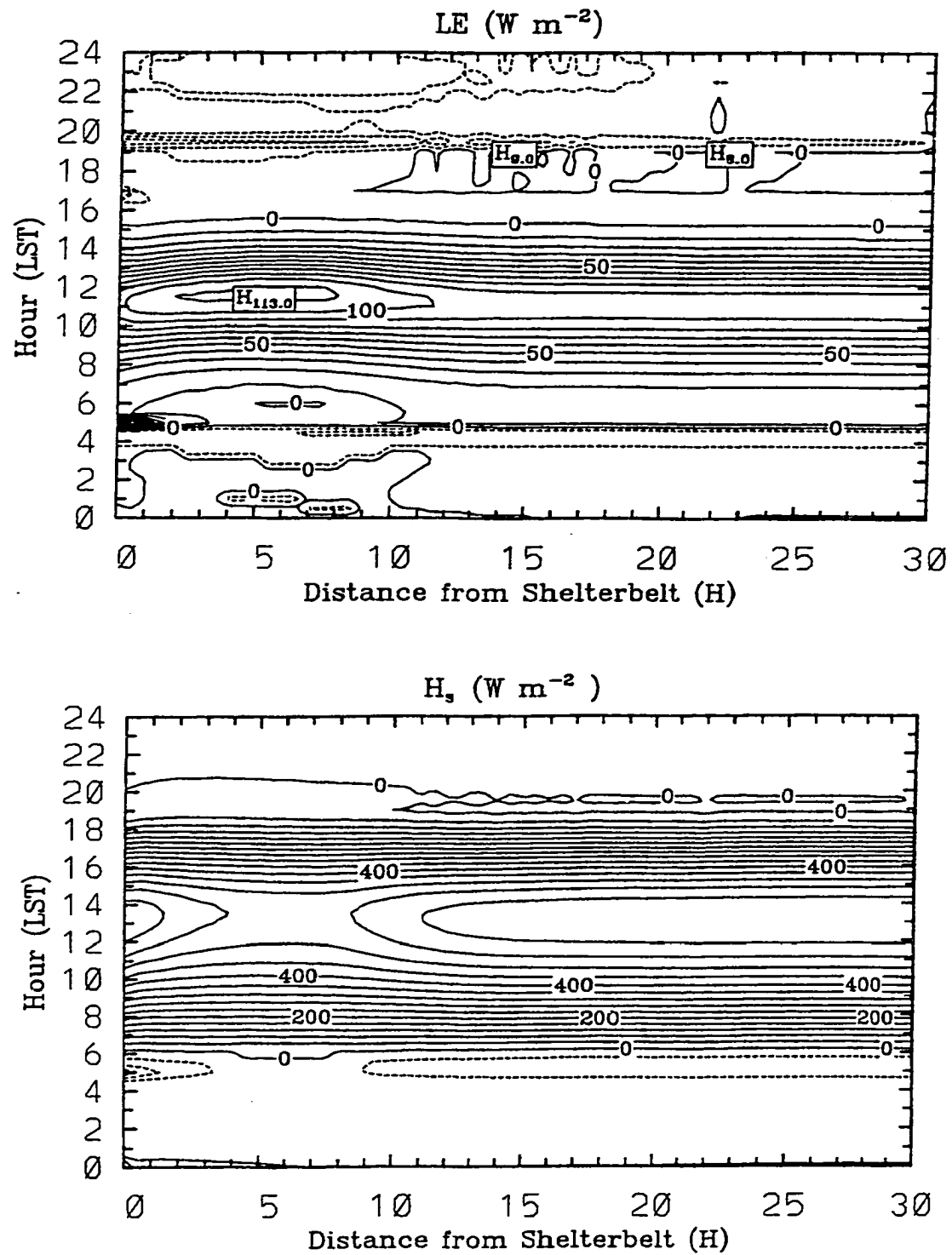


Figure 1a. Effects of medium-dense shelterbelts on latent and sensible heat fluxes (LE and H_s) for various soil moisture wetness (m).
 (a) $m=0.2$, (b) $m=0.3$, (c) $m=0.5$, (d) $m=0.8$

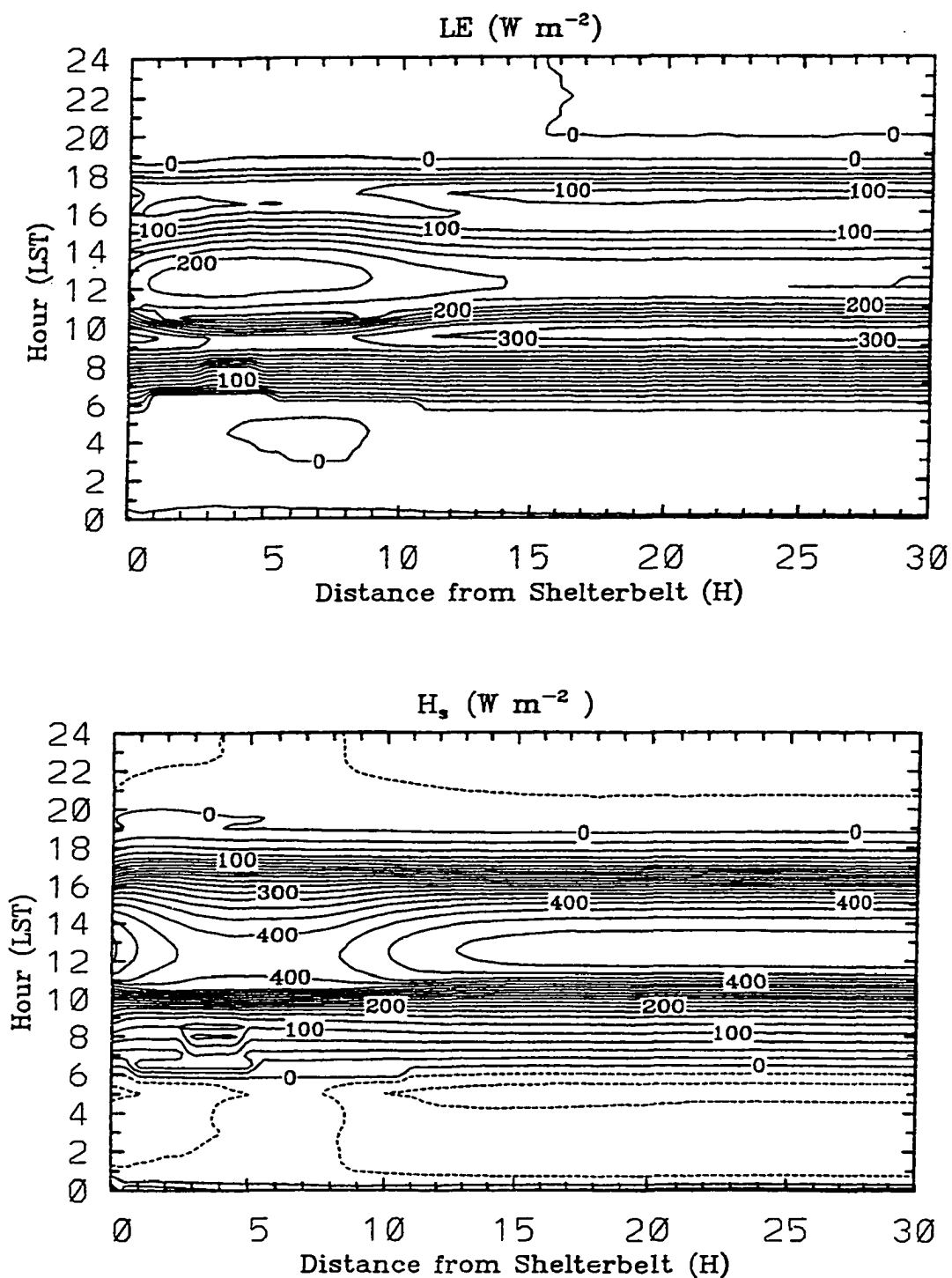


Figure 1b. Effects of medium-dense shelterbelts on latent and sensible heat fluxes (LE and H_s) for various soil moisture wetness (m).
 (a) $m=0.2$, (b) $m=0.3$, (c) $m=0.5$, (d) $m=0.8$

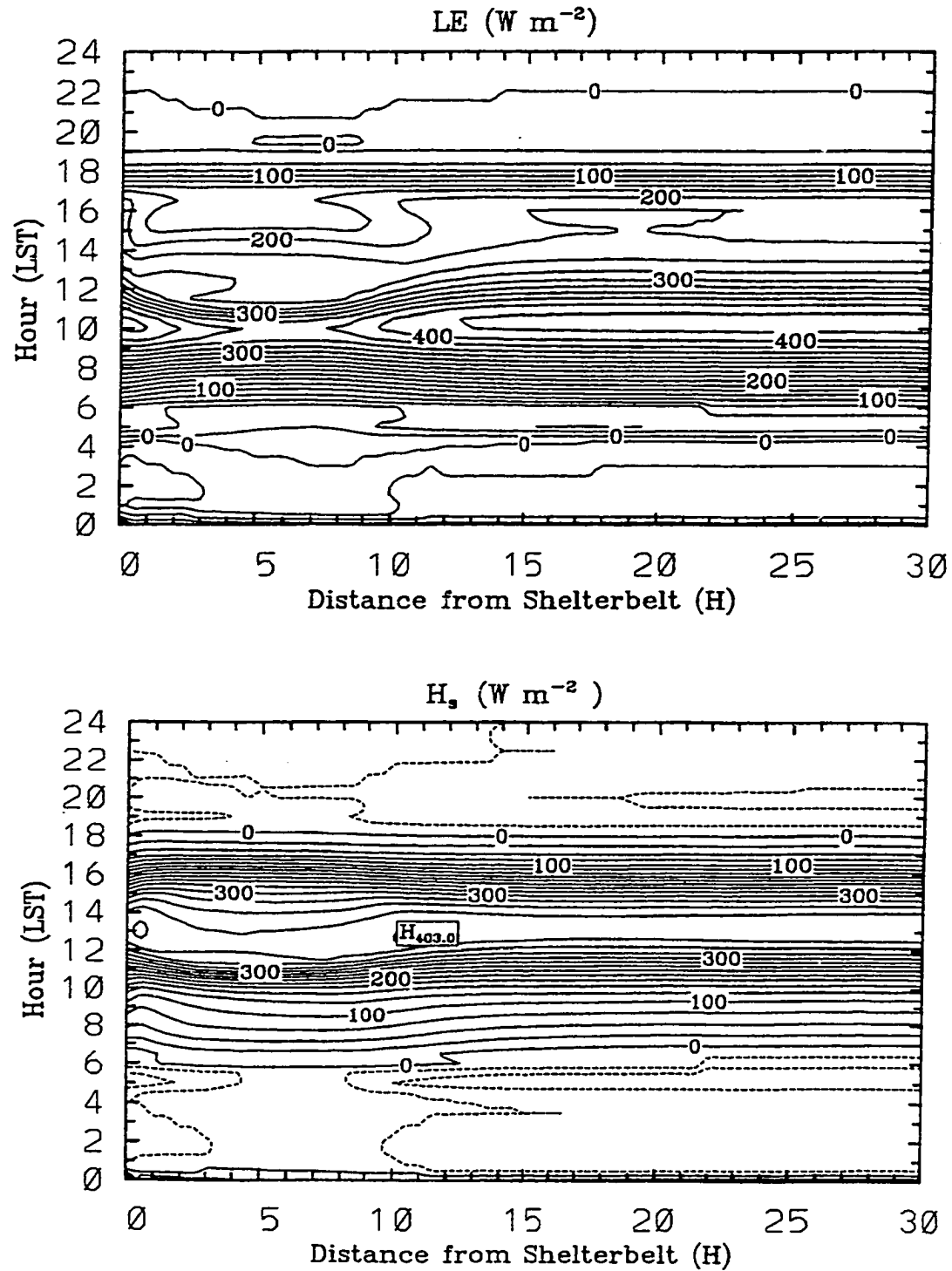


Figure 1c. Effects of medium-dense shelterbelts on latent and sensible heat fluxes (LE and H_s) for various soil moisture wetness (m).
(a) m=0.2, (b) m=0.3, (c) m=0.5, (d) m=0.8

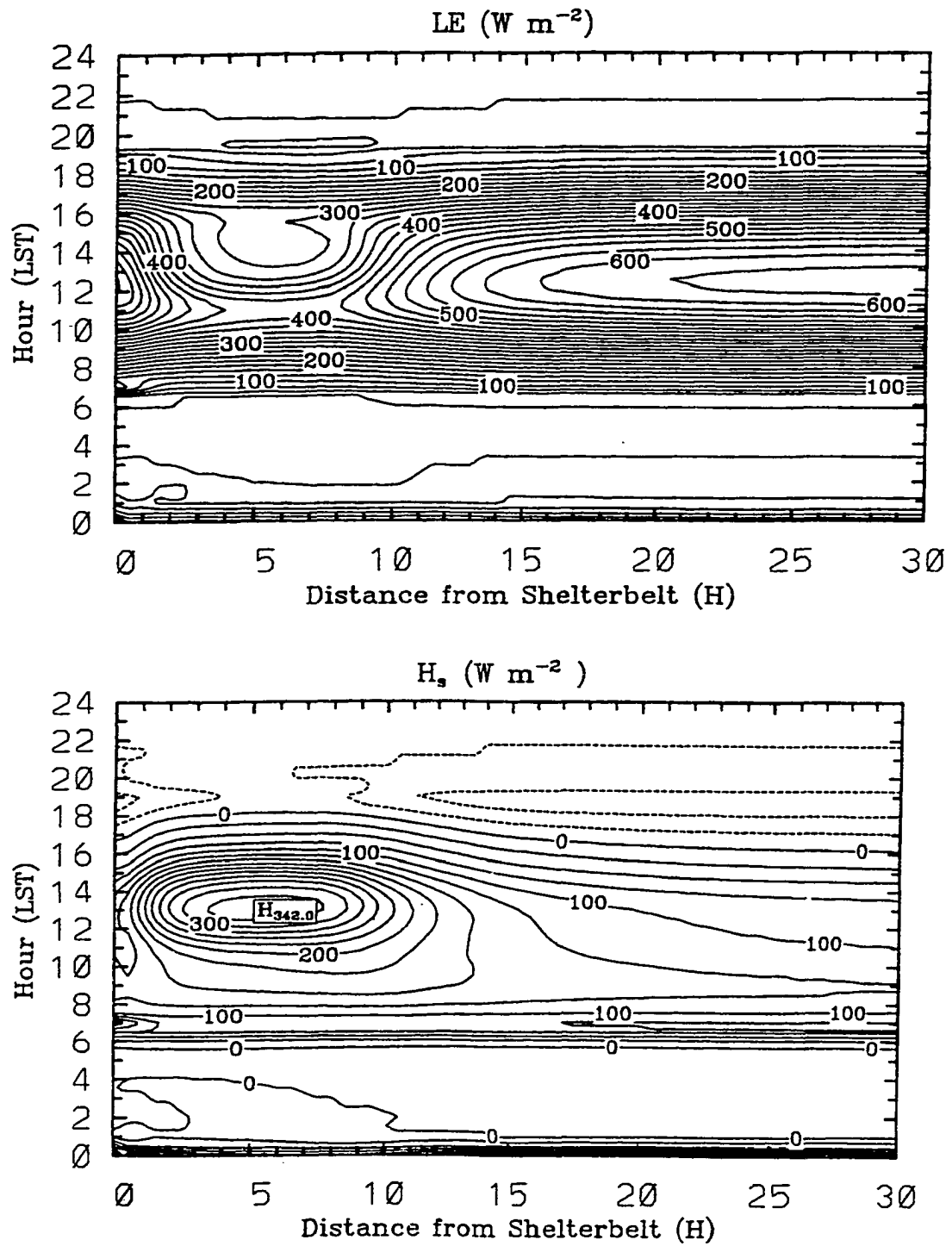


Figure 1d. Effects of medium-dense shelterbelts on latent and sensible heat fluxes (LE and H_s) for various soil moisture wetness (m).
 (a) $m=0.2$, (b) $m=0.3$, (c) $m=0.5$, (d) $m=0.8$

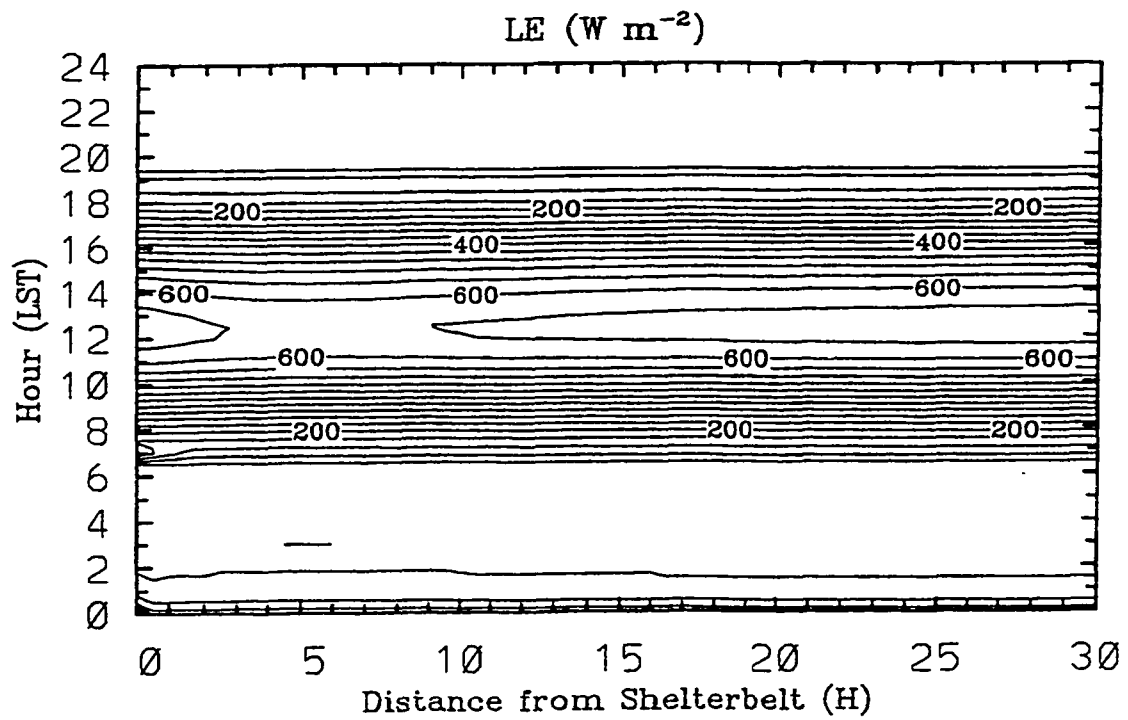
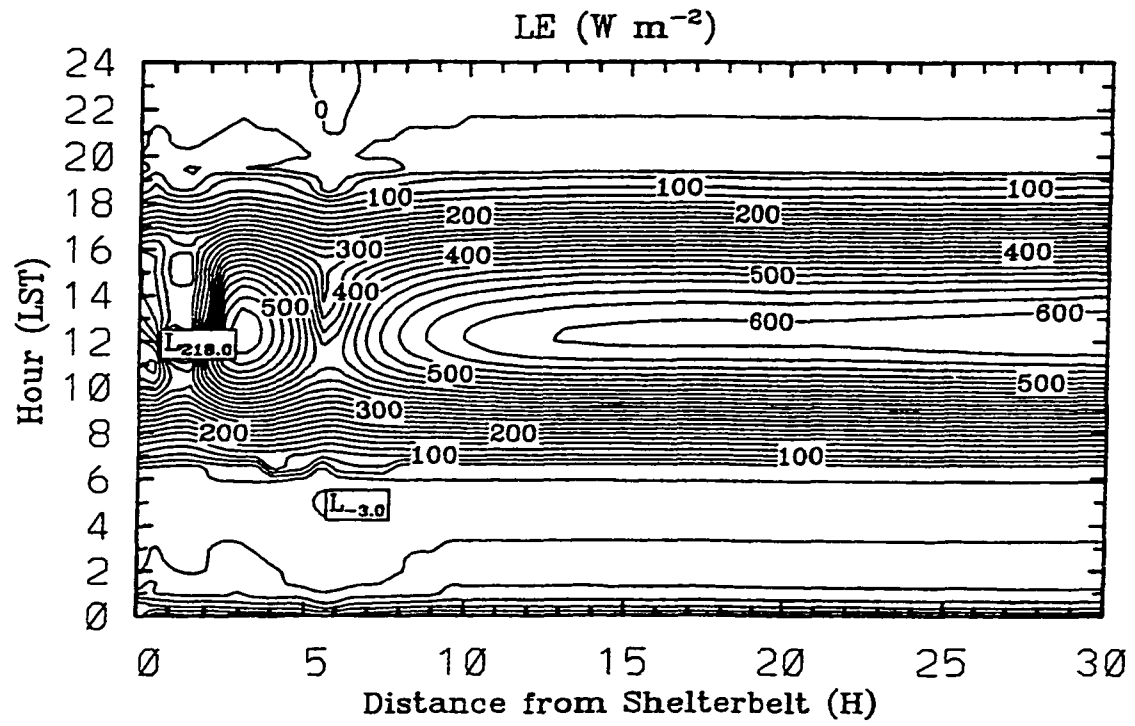


Figure 2. Influences of shelterbelt density on evapotranspiration (LE). (a) very dense shelterbelts and (b) very loose shelterbelts.

With increasing soil wetness, the situations begin to change. For $m=0.3$, the increase in LE is still as large as 20%, but the maximum increase of LE occurs at $t=1300$ LST. At $t=0900$ LST, a significant decrease in LE occurs between 0-17 H in the lee with the maximum decrease at $x=6$ H, and at $t=1700$ LST a slightly smaller LE occurs in the sheltered zone (Fig.1b). H_s decreases at noon but increases in early morning and late afternoon in the sheltered zone for $m=0.3$. For $m=0.4$ (figures omitted), the situation is similar to that of $m=0.3$, but the noontime increase in LE is very small, and the decreases in LE become significant both in the morning and afternoon. Maximum decreases in LE are as large as 18% and 25% at $t=1000$ LST and $t=1530$ LST.

For $m=0.5$ (Fig.1c), the noontime increase in LE disappears, and evapotranspiration is reduced all during the daytime in the near lee. However, in the middle lee (15-23 H), LE increases in the afternoon because of wake turbulence. Accordingly, H_s increases in the sheltered zone (Fig.1c), which is more obvious for $m=0.6$ case (figure not shown), where a center of large H_s forms at $x=6$ H around noon. Evapotranspiration during 0900-1300 LST is further reduced with the maximum reduction as large as 40% for $m=0.6$. For $m=0.8$ (Fig.1d), LE is significantly reduced during the daytime in the sheltered zone extending to 25 H leeward side, with maximum reduction of 50% occurring in the early afternoon at 4-7 H leeward, where reductions of wind and turbulence also reach their maxima (Wang and Takle, 1995b). By contrast, H_s increases in the sheltered zone and forms a center shown in Fig.1d. The situation is nearly identical for further increasing soil wetness (figures omitted). From Fig.1a-d, we also can see that the maximum evapotranspiration occurs at 0930, 1030, 1230 for $m=0.3$, 0.5, and 0.8, respectively. Crop stomata are closed when water is limited. For dry soil,

the stomata are closed earlier than for moderately wet soil. For wet soil, the stomata are not closed and therefore the evapotranspiration reaches its maximum during the maximum radiation and temperature.

3.2 Influence of shelterbelt structure on evapotranspiration and heat flux

Turbulence and flow structures near shelterbelts are very dependent on shelterbelt density, and wind-shelter effect reaches its maximum for medium-dense shelters (Wang and Takle, 1995a,b,c,d). Fig.2 shows the changes of LE and H_s at $m=0.8$ for shelterbelts with two extreme densities: very dense and very loose. Combined Fig.1d which shows the case of a medium-dense shelterbelt, we demonstrate the dependence of the shelter evapotranspiration effect on shelterbelt density. As shown in Fig.2a, very dense shelterbelts cause complicated changes in both LE and H_s in the near lee, which obviously relate to recirculations and separation of streamlines (Wang and Takle, 1995b). The recirculation zone severely suppresses the sheltering gain of a medium-dense shelter within 5 H of the shelter. A region centered at 3H has nearly the same evapotranspiration as an unsheltered area, although the region from 5-10 H has evapotranspiration reduced by 10-30%. H_s is enhanced with two centers located on $x=1.5$ and 5.5 H. For very loose shelterbelts, as shown in Fig.2b, LE is still reduced, but the effects of shelterbelts on both H_s and LE is quite small.

4. Summary and Conclusion

Agricultural practices such as shelterbelts networks and strip-cropping can significantly affect evapotranspiration. However, the effects are not straightforward, and our simulations show the complicated temporal and spatial variability of both latent and sensible heat fluxes. Aerodynamic shelter effects and their interactions with energy, water and mass transfer in a soil-vegetation-atmosphere system causes significant variability and are controlled, to great extent, by soil moisture availability. We have demonstrated that soil moisture controls not only the magnitude of evapotranspiration but also the direction of evapotranspiration-shelter effect: shelterbelts cause evapotranspiration to decrease for wet soil, increase for dry soil, and increase around noon but decrease in the morning and afternoon for moderately wet soil. We also illustrated that recirculation significantly diminished the reduction of evapotranspiration for very dense shelterbelts and that very loose shelterbelts exerted small effects on partition of energy. Therefore, combined with our previous results showing optimum wind reduction for medium-dense shelters, we conclude that medium-dense shelterbelts have maximum combined evapotranspiration-shelter efficiency. We also concluded that regional scale agricultural and forestry practices may significantly affect water and energy cycles.

Acknowledgment This research was supported by USDA/CSRS NRI Competitive Grant # 93371018954 and USDOE/NIGEC grant No. DE-FC0390ER61010.

References

- Bates, C.G., 1911: Windbreak -- their influence and value. *Bull. Forest Service*, US Dept.Agric., 1-22.
- Brenner, A.J., P.G. Jarvis, R.J. van deb Beldt, 1995: Windbreak-crop interactions in the Sahel. 2. Growth response of millet in shelter. *Agric. For. Meteorol.*, **75**, 235-262.
- Dixon, M., and J. Grace, 1984: Effect of wind on the transpiration of young trees. *Ann. Bot.*, **53**, 811-819.
- Gagarin, E., 1949: Planting trees for protecting fields in Russia. *Forstwiss Centralblatt*, **68**, 571-602.
- Hagen, L.J. and E.L. Skidmore, 1974: Reducing turbulent transfer to increase water-use efficiency. *Agric. Meteorol.*, **14**, 153-168.
- Judd, M.J., P.T. Prendergast, and K.J. McAneney, 1993: Carbon dioxide and latent heat flux measurements in a windbreak-sheltered orchard. *Agric. For. Meteorol.*, **66**, 193-210.
- Long, S.P., and N. Persaud, 1988: Influence of neem (*Azadirachta indica*) windbreaks on millet yield, microclimate, and water use in Niger, West Africa. *Challenges in dryland agriculture -- a global perspective*, Unger, P.W., T.V. Sneed, J.R. Jordan, and R. Jensen, eds. Texas Agricultural Experimental Station, Amarillo, TX, 313-314.
- Marshall, J.K., 1967: The influence of shelter on the productivity of grasslands and field crops. *Field Crop Abstr.*, **20**, 1-14.
- Miller, D.R., N.J. Rosenberg, and W.T. Bagley, 1973: Soybean water use in the shelter of a slat-fence windbreak. *Agric. Meteorol.*, **11**, 405-418.

- Rosberg, N.J., 1967: The influence and implications of windbreaks on agriculture in dry regions. *Ground level climatology*, R.H. Shaw, ed. Am. Assoc. Advan. Sci., Symposium, 327-349.
- Sellers, P.J., Y.Mintz, Y.C. Sud, and A. Dalcher, 1986: A simple biosphere model (SiB) for use within general circulation models. *J. Atmos. Sci.*, **43**, 505-531.
- Sturrock, J.W., 1988: Shelter and its management and promotion. *Agric. Ecosys. Environ.*, **22/23**, 17-39.
- Von Maydell, H.-J., 1987: Agroforestry in the dry zones of Africa: past, present and future. *Agroforestry, a decade of development*. Steppeler, H.A., and P.K.R. Nair, eds. ICRAF, Nairobi, 89-116.
- Wang, H. and E.S. Takle, 1995a: Boundary-layer flow and turbulence near porous obstacles. I. Derivation of a general equation set for a porous medium. *Boundary-Layer Meteorol.*, **74**, 73-88.
- Wang, H. and E.S. Takle, 1995b: A Numerical Simulation of boundary-layer flows near shelterbelts. *Boundary-Layer Meteorol.*, **75**, July 1995.
- Wang, H. and E.S. Takle, 1995c: Numerical simulation of shelterbelt effects on wind direction. *J. Appl. Meteorol.*, Oct. 1995
- Wang, H. and E.S. Takle, 1995d: Influences of shelterbelts on turbulent flow and shelter mechanism. *22nd Conference on Agricultural and Forest Meteorology with Symposium on Fire and Forest Meteorology* (8.12), Atlanta, Georgia, 28 January - 2 February 1996.

CHAPTER 7. GENERAL CONCLUSIONS AND DISCUSSIONS

We have systematically presented the development of high-performance parallel software system that simulates complex shelterbelt turbulent flow and we used this computer simulation system to reproduce almost all observed phenomena, to find the optimal design, to understand the nature and mechanism of shelterbelt turbulent flows, and to resolve historically controversial issues. In Chapter 1, we gave the general introduction and literature review, and pointed out that many computer scientists from national laboratories have spent a lot of efforts on parallelizing PCM and MPMM for MPPs, and now they have proposal for parallelizing these models for clusters. In Chapter 2, we presented our computer simulation model system, and demonstrated the capacities of our computer simulation model system. In Chapter 3, we presented the details of parallel algorithm design and performance evaluation for several different algorithms including sequential coding analyses, functional decomposition parallel programming, and domain decomposition parallel programming. Through profiling, we found that the computation of dynamic pressure takes most of the computer time and the solver of linear algebra takes very little time. In Chapter 4, we summarized the development processes of parallel software and developed parallel performance modeling. Combined theory and experiments of parallel performance, we obtained interesting results for cluster computing performance. Then, in Chapter 5, we used this computer simulation model to reevaluate previous researchers' work on drag --- the root cause for dynamic systems, and provided a

framework for drag calculation using high-resolution simulation data. In Chapter 6, we simulated evapotranspiration and heat fluxes, the most important elements to food production and ecosystems, and resolved historically disputed issues.

Recently we developed the theoretical analysis and modeling of MPP-based and Cluster-based computing performance. Our results showed that for MPP-based computing, the speedup increases with increasing number of processors; for cluster-based computing, the speedup first increases with increasing number of processors, however, with further increases of the number of processors, the speedup will decrease. There exists a turning point where the speedup reaches the maximum, and we also pointed out that the maximum speedup is less than half of the number of processors. Because of limited communications channel, high latencies, and low bandwidths, it is crucial to design effective parallel algorithms for cluster of computers that minimize communication costs and maximize load distributions.

The performance of parallel programs is typically studied in one of two ways. Asymptotic analysis considers the behavior of an algorithm on very large numbers of processors; empirical studies seek to extrapolate from results obtained on a small number of processors to predict large-scale parallel performance. Unfortunately, it is rare that either approach provides accurate prediction of program performance. As simple asymptotic analysis drops lower-order cost terms, it is often inaccurate on realistic numbers of processors. Extrapolation from empirical results without guidance from analytic studies is always dangerous. We combined both approaches to avoid the deficiencies of experimental analysis and theoretical analysis. We used our experimental data to fit our theoretical analysis and we obtained the maximum speedup for the large domain is 19.6 and the corresponding number of

processors (p) is 105; and for the small domain the maximum speedup is 12.4 and the corresponding p is 56. We concluded that cluster computing is good for big computation-intensive tasks, since the ratio of computation load over communication overhead is higher at this situation. We need to choose the optimal number of processors according to the computation load in the cluster computing environment. On the other hand, when we develop a simulation model system, we need to determine the optimal complexity of computer simulation model system according to available computing resources.

The costs of coordinating computation and sharing data depend critically on the way in which the algorithm is decomposed, that is, how its data and computation are distributed among processors of a parallel machine or network of workstations. It is the rate at which these costs determine an algorithm's scalability. We discussed the design of algorithms in details and compared the performance of sequential, functional decomposition, and domain decomposition codes. The parallel code's performance depends on the design and coding strategies very much. Different designs and implementations have different parallel performance gains. For functional decomposition, we even got the parallel performance loss. The advantage obtained by concurrent computation was overwhelmed by the large communication overhead between nodes because functional decomposition needs additional $O(n^2)$ message exchanges. Functional decomposition parallel code is not flexible and scalable. We also implemented domain decomposition parallel code of our model system. The performance gains are quite satisfactory. We also highlighted that MPI communication functions make code parallelizing easier and more effective. We analyzed the speedup and its change with domain size and number of processors. Speedup ratio increases with increasing

number of processors; the increase is faster for small number of processors; with further increase of number of processors, the performance gains slow down, especially for small domain. There is a tradeoff between reduced computation load and increased communication load, and the relative weight, which is related to domain size, number of processors, and problem-specific features, determines speedup ratio and load balance ratio.

The implementation and coding of parallel algorithms can be accomplished by using data transfer mechanisms either implicitly or explicitly with software and tools. In principle, any communication channel and mechanism can be used for data transfer needed for partitioned tasks. We implemented our model system with PVM, or Parallel Virtual Machine (Wang et al. 1988), and MPI, or Message Passing Interface (Wang et al. 1999, 2000). PVM and MPI have their own individual merits.

The overall objective of the PVM system is to enable such a collection of computers to be used cooperatively for concurrent or parallel computation. Briefly, the principles upon which PVM is based include the following: (1) User-configured host pool -- the application's computational tasks execute on a set of machines that are selected by the user for a given run of the PVM program. (2) Both single-CPU machines and hardware multiprocessors (including shared-memory and distributed-memory computers) may be part of the host pool. The host pool may be altered by adding and deleting machines during operation (an important feature for fault tolerance). (3) Translucent access to hardware: Application programs either may view the hardware environment as an attribute-less collection of virtual processing elements or may choose to exploit the capabilities of specific machines in the host pool by positioning certain computational tasks on the most appropriate computers. (4) Process-based computation: The

unit of parallelism in PVM is a task, an independent sequential thread of control that alternates between communication and computation. No process-to-processor mapping is implied or enforced by PVM; in particular, multiple tasks may execute on a single processor. (5) Explicit message-passing model: Collections of computational tasks, each performing a part of an application's workload using data-, functional-, or hybrid decomposition, cooperate by explicitly sending and receiving messages to one another. Message size is limited only by the amount of available memory. (6) Heterogeneity support: The PVM system supports heterogeneity in terms of machines, networks, and applications. With regard to message passing, PVM permits messages containing more than one data type to be exchanged between machines having different data representations. (7) Multiprocessor support: PVM uses the native message-passing facilities on multiprocessors to take advantage of the underlying hardware. Vendors often supply their own optimized PVM for their systems, which can still communicate with the public PVM version.

There are several advantages by using MPI: (1) portability – MPI offers a degree of portability across different machines. This means that the same message-passing source code can be executed on a variety of machines as long as the MPI library is available. Though message passing is often thought of in the context of distributed-memory parallel computers, the same code can run well on a shared-memory parallel computer. It can run on a network of workstations, or as a set of processes running on a single workstation. All these give a high degree of flexibility in code development, debugging, and in choosing a platform for program runs. (2) compatibility – it is the ability to run transparently on heterogeneous systems, that is, collections of processors with distinct architectures. It is possible for an MPI

implementation to span such a heterogeneous collection, yet provide a virtual computing model that hides many architectural differences. The user need not worry whether the code is sending messages between processors of like or unlike architecture. (3) efficiency – MPI was carefully designed so as to avoid a requirement for large amounts of extra information with each message, or the need for complex encoding or decoding of message headers. MPI also avoids extra computation or tests in critical routines. (4) performance – MPI was designed to encourage overlap of communication and computation so as to take advantage to intelligent communication agents, and to hide communication latencies. This is achieved by the use of non-blocking communication calls, which separate the initiation of a communication from its completion. (5) scalability – it is an important goal of parallel processing. Message passing is a programming paradigm used widely on parallel computers, especially scalable parallel computers with distributed memory, and networks of workstations (NOWs).

We have compared our simulation results with a lot of available observations, our computer simulation model system simulated all the observed phenomena, reproduced, interpreted, and resolved many seemingly controversial observations, gave the whole picture of complex shelterbelt turbulent flows, and predicted many characteristics that were later found to be true in the field (Wang and Takle 1995a,b; Wang and Takle 1996a, b,c; Wang and Takle 1997a,b; Wang et al. 1998, 1999, 2000, 2001). We successfully simulated recirculations, flow attachment and separation, the relationships between porosity and wind reductions, optimal porosity, wind direction rotation, pressure field, and turbulent structure as well as the effects of three dimensionality, shape, width, and oblique flows on shelter efficiencies and we found the shelter mechanisms through momentum analysis of the simulated

results. All of these have been reported in the published papers. Here we summarize only the drag force, a root cause for flow interactions that is very important to regional and global climate modeling, and evapotranspiration and soil moisture, the most important climate elements to crop production and ecosystems. Both are very difficult to be quantified and observed, the previous research led to the controversial theories because they are related to too many factors and they have very complicated spatial and temporal changes.

High-resolution and physically consistent data from numerical simulations allow us to examine previous drag-calculation methods. We systematically evaluated the surface stress-loss method proposed by Seginer and Sagi, and found that the results from this method is unreasonably small. Moreover, we found that assuming logarithmic profile of windspeed in the wake to derive the turbulent stress can lead to large errors. The coefficient calculated by this method heavily depends on the measurement height. The drag coefficient calculated from turbulent stress derived from mean wind-reduction profile is larger than that calculated directly from the simulated turbulent stress of higher-order turbulent closure. However, it is still unreasonably small compared to the actual value. The coefficient calculated by both procedures heavily depends on the measurement height. Our calculation using drag-plate-measured turbulent stress data gave the same results. Therefore, the stress-loss method to determine drag coefficient is questionable.

High-resolution simulations also provided us the opportunity to analyze the balance of forces perturbed by the drag near the shelterbelt and let us evaluate them quantitatively. The mechanical drag exerted by shelterbelts on airflow only exists within shelterbelts. This drag causes complicated interactions among pressure, flow, and turbulence. These processes

transmit the effects of drag beyond the boundaries of obstacle. We have partitioned the drag into perturbed pressure, horizontal and vertical mean transports, and horizontal and vertical turbulent transports. The drag exerted on airflow by the belt is balanced mainly by static pressure difference and horizontal advection across the belt. With increasing shelterbelt density, horizontal advection decreases, leaving static pressure difference to balance the drag. Total (dynamic plus static) pressure difference across the belt is close to the drag, and the vertically-averaged difference between drag and pressure coefficient is within 8%. Generally, horizontal turbulent transport and vertical convection are smaller. However, great difference exists near the top and bottom of the belt, especially near the top, where vertical turbulent transport plays an important role. Therefore, we can estimate drag of a shelterbelt with high accuracy by measuring only the changes of static pressure and windspeed across the shelterbelt.

Agricultural practices such as shelterbelt networks and strip-cropping can significantly affect evapotranspiration. However, the effects are not straightforward, and our simulations show the complicated temporal and spatial variability of both latent and sensible heat fluxes. Aerodynamic shelter effects and their interactions with energy, water and mass transfer in soil-vegetation-atmosphere system cause significant variability and are controlled, to great extent, by soil moisture availability. We have demonstrated that soil moisture controlled not only the magnitude of evapotranspiration but also the direction of evapotranspiration-shelter effect: shelterbelts make evapotranspiration decrease for wet soil, increase for dry soil, and increase around noon but decrease in the morning and afternoon for moderately wet soil. We also illustrated that recirculation significantly diminished the reduction of evapotranspiration for very dense shelterbelts and that very loose shelterbelts exerted small effects on partition of

energy. Therefore, medium-dense shelterbelts also have maximum evapotranspiration-shelter efficiency. We also concluded that regional scale agricultural and forestry practices may significantly affect water and energy cycles.

We have demonstrated that our computer simulation model system captured very detailed nature of shelterbelt turbulent flows, drag, pressure, evapotranspiration, heat fluxes, and resolved several historically controversial issues. We have several directions for the further research, all of them are demanding ever greater computing resources and better computing efficiencies: (1) applying this computer simulation model to different situations for guiding to practices such as how people can improve climate and soil conditions; (2) using this computer simulation model into some other fields such as pollution control, flux aggregation, snow drifting control, and some engineering problems; (3) further expanding this computer simulation model to include more physical variables; (4) linking our computer simulation model to regional and global models so that we can really assess the impacts of global changes on the plant community that only local climate matters; (5) exploring climate-ecosystem interactions in great details as outlined in the Communications of the ACM; (6) exploring new computer systems and new computer algorithms that can be used more efficiently to solve scientific and engineering problems. All of these will promote the development of cluster computing as in Beogulf cluster project that was first used by NASA Earth and Space Scientists several years ago. Our theoretical and experimental analyses demonstrated that communication latencies and bandwidth still are the factors that limit the performance of cluster computing. There is a lot of space to improve these on current techniques. If we can develop new techniques and build multiple communication channels on one network interface

card (NIC) and one physical line that connects to computer, and this will solves the competing for communication channel issue in cluster of computers, then cluster computing will be the only computing environment in the world and MPPs will disappear.

References

- Foster, Ian, 1998: *The Grid: Blueprint for a new computing infrastructure*. Morgan-Kaufmann, 1998.
- Foster, Ian, 2000: Internet Computing and the Emerging Grid. *Nature*, Dec.7, 2000.
- Prusinkienicz, P., 2000: Simulation modeling of plants and ecosystems. *Communications of the ACM*, pp.84-93, 43(7), July, 2000.
- Semtner, A., 2000: Ocean and atmosphere modeling. *Communications of the ACM*, 43(4), pp.80-89, April, 2000.
- Wang, H. and J. Shen, 1989. A two-dimensional numerical study of the wind sheltering effects of shelterbelts. *Acta Meteorologica Sinica* 3:498-505
- Wang, H., 1991a. The effects of shelterbelts on the atmospheric turbulent exchange coefficient. *Acta Geographica Sinica* 46:107-14
- Wang, H., 1991b. A numerical simulation of the wind sheltering effects of multiple parallel shelterbelts. *Journal of Nanjing University* 481-88
- Wang, H., 1992. On methods for the study about the protection effects of shelterbelts. *Promoting Agriculture through the Progress of Science and Technology*, Eastern China Agricultural Association, Ed., Jiangsu Sci.&Tech. Press, 369-72.

- Wang, H. and E.S. Takle,. 1994a. Interaction of synoptic flow and mesoscale circulation as a function of synoptic windspeed. Sixth Conference on Mesoscale Processes, Portland, Amer. Meteor. Soc., J36-J39.
- Wang, H. and E.S. Takle, 1994b. Mesoscale and boundary-layer flows over inhomogeneous surfaces consisting of porous obstacles. Sixth Conference on Mesoscale Processes, Portland, Amer. Meteor. Soc., 262-65.
- Wang, H. and E.S. Takle, 1995a. Boundary-layer flow and turbulence near porous obstacles. I. Derivation of a general equation set for a porous medium. *Boundary-Layer Meteorol.* 74:73-88
- Wang, H. and E.S. Takle, 1995b, A numerical simulation of boundary-layer flows near shelterbelts. *Boundary-Layer Meteorol.* 75:141-73
- Wang, H. and E.S. Takle, 1995c. Equations for mean and turbulent flow through and over porous obstacles. Eleventh Symposium on Boundary Layers and Turbulence, Charlotte. Amer. Meteor. Soc.
- Wang, H. and E.S. Takle, 1995d: Simulations of mean and turbulent properties of oblique flows near agricultural shelterbelts. Eleventh Symposium on Boundary Layers and Turbulence, Charlotte. Amer. Meteor. Soc.
- Wang, H. and E.S. Takle, 1996a. On three-dimensionality of shelterbelt structure and its influences on shelter effects. *Boundary-Layer Meteorol.* 79:83-105
- Wang, H. and E.S. Takle, 1996b. On shelter efficiency of shelterbelts in oblique wind. *Agric. Forest Meteorol.* 81:95-117

- Wang, H. and E.S. Takle, 1996c. Numerical simulations of shelterbelt effects on wind direction. *Journal of Applied Meteorol.* 34:2206-19
- Wang, H. and E.S. Takle, 1996d. Influences of shelterbelts on turbulent flow and shelter mechanism. *22nd Conference on Agricultural and Forest Meteorology with Symposium on Fire and Forest Meteorology* (8.12), Atlanta, Georgia, 28 January - 2 February 1996.
- Wang, H. and E.S. Takle, 1996e. Modeling the evapotranspiration and energy partition of inhomogeneous agroecosystems. *22nd Conference on Agricultural and Forest Meteorology*, Atlanta, GA, Amer. Meteor. Soc
- Wang, H. and E.S. Takle, 1997a. Model-simulated influences of shelterbelt shape on wind-sheltering efficiency. *Journal of Applied Meteorol.* 36:695-704
- Wang, H. and E.S. Takle, 1997b. Momentum budget and shelter mechanism of boundary-layer flow near a shelterbelt. *Boundary-Layer Meteorol.* 82:417-35
- Wang, H., E.S. Takle, and J. Shen, 1997c. Radiative characteristics of an agroforestry ecosystem. *Ninth Conference on Atmospheric Radiation*. Amer. Meteorol. Soc., 2-7 February, Long Beach, CA. 308-309
- Wang, H., J. Shen and E.S. Takle, 1997d. Influences of agroforestry ecosystem on evapotranspiration and soil moisture. *13th Conference on Hydrology*. Amer. Meteorol. Soc., 2-7 February, Long Beach, CA. 360-61
- Wang, H., J. Shen, and E.S. Takle, 1997e. High-resolution regional climate simulations using RegCM2 with different scale couplings of soil, vegetation, and atmospheric boundary-

- layer processes. Preprints, 12th Symposium on Boundary Layers and Turbulence, Amer. Meteor. Soc. Vancouver. 573-74
- Wang, H., G.M. Prabhu, and E.S. Takle , 1998. Parallelization a very-high-resolution climate model using clusters of workstations with PVM and performance and load balance analyses. Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA'98), CSREA Press. 1:1762-65
- Wang, H. and E.S. Takle., 1998a. Agroforestry shelterbelts and its influence on evapotranspiration. ASAE Annual International Meeting, Orlando. 24 pp.
- Wang, H. and E.S. Takle , 1998b. Agricultural shelterbelt's protection functions. ASAE Annual International Meeting, Orlando.
- Wang, H., G.M. Prabhu, E.S. Takle, and R. Todi, 1999. Implementation and performance evaluation for a computation-intensive climate simulation application. Proceedings of the international conference on parallel computing (parco99). Pp1-8.
- Wang, H., E.S. Takle , and J. Shen, 2001: Shelterbelt and Windbreaks: Mathematical modeling and computer simulations of turbulent flows. *Annual Review of Fluid Mechanics*, 33:549-586.
- Wang, H., G.M. Prabhu, E.S. Takle, and J. Shen, 2000: Implementation and Performance Evaluation of Parallel Computations by Using Cluster of Networked Workstations. *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, CSREA Press, pp. 2007-2013.