Final Report
ONR N00014-89-J-3202

Intelligence in Scientific Computing

The Artificial Intelligence Laboratory
Massachusetts Institute of Technology
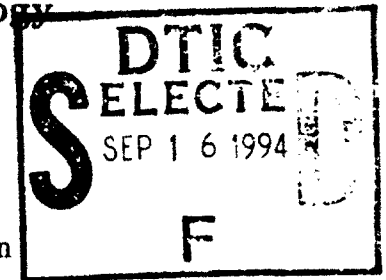
1 Project Objectives:

To develop new computer representations and reasoning mechanisms that enable intelligent systems to autonomously design, monitor, and understand complicated physical systems, through appropriate mixtures of numerical and symbolic computing.

As part of this research we demonstrated novel computational tools that autonomously monitor, understand, and control complex physical systems. These tools understand the models, monitor the execution of the simulations, and formulate qualitative explanations of the results. These tools interact with their human users in symbolic terms, allowing them to specify qualitative goals for measurement and intelligent control applications.

Work started under this effort is now being continued at MIT, Yale, University of Colorado at Boulder, Perdue, Xerox, and Hewlett Packard. Over the last three years five of our recent graduates have received National Young Investigator awards, largely based on their work on this project.

DTIC QUALITY

8

2 Summary Accomplishments

- For low-order dynamical systems, we have demonstrated that our programs can autonomously generate qualitative interpretations of dynamical behavior similar to those appearing in published papers in science and engineering.

  We have demonstrated that the automatic analysis of dynamical systems can be used to produce new results of current interest in theoretical hydrodynamics. In particular, we have isolated resonance overlap as an explanation of hitherto puzzling phenomena concerning surface waves in towing tanks.

- We have released to the DARPA community a compiler for Scheme, a kernel dialect of Lisp that is appropriate for direct realization in hardware and for parallel computing. Scheme has been used as an effective implementation vehicle for other computer languages, such as Common Lisp, and we have collaborated with another DARPA contractor who is using our implementation of Scheme in this way for parallel machines. The compiler achieves performance equivalent to the best-quality commercial Lisp compilers.

- In cooperation with Hewlett-Packard, we have completed the design and construction of a Supercomputer Toolkit memory-processor unit, host interface, and backplane. The Supercomputer Toolkit is a technology that permits automatic synthesis of a variety of special-purpose computers, each achieving supercomputer performance on the problem for which it is designed, yet at a fraction of the cost and complexity of a general-purpose supercomputer.

  Rajeev Surati extended our partial-evaluation compilation technique to automatically extract low-level parellelism from conventionally written programs. He demonstrated the effectiveness of this approach by generating automatically-parallelized code for the n-body problem. The code runs on the 8-processor Supercomputer Toolkit and achieves a factor of 6.2 speedup over an almost optimal uniprocessor computation, which compares with an average speedup factor of 4.0 for 8 processors obtained using manual restructuring of code in experiments performed at University of Illinois on a suite of benchmarks for the Cray YMP.

Gerald Jay Sussman and Jack Wisdom used the 8-processor Supercomputer Toolkit to perform a breakthrough computation in solar-system dynamics, integrating the complete Solar System, with General Relativity and Earth-Moon quadrupole correction, for 100 million years. The longest previous such integration, published in 1991, was for 3 million years.

- Feng Zhao has identified "flow pipes" as a technique for efficiently searching for and reasoning about phase-space structures. These group trajectories into a manageable number of equivalence classes that have similar dynamics. This helps plan gain-schedules for the design of controllers in nonlinear systems.

  Feng Zhao's MAPS (Modular Analyzer for Phase Spaces) program automatically synthesized a high-quality controller design for the German Transrapid magnetic levitation vehicle system. The new controller tolerates vehicle displacements that are twenty times larger than those allowed in a previous controller that was designed using classical linear-feedback technique.

- A series of experiments by Elmer Hung has demonstrated a technique for exploiting dynamics and simulation to achieve outstanding accuracy in measuring the parameters of physical systems. Beginning with imprecise initial estimates of device parameters in an electrical circuit, his method refines these to produce an improvement in relative error of five orders of magnitude.

  As part of the work on precision measurement, Thanos Siapas has discovered a new invariant that quantifies the dependence of global geometry of basins of attraction of dynamical systems on parameter variations. This is one of the first attempts to characterize fine changes in the geometry of phase space structures, as opposed to coarse changes in their topology.

- Elizabeth Bradley's program, Perfect Moment, automatically synthesizes control systems that achieve very fast target acquisition, taking advantage of the global phenomena and the inherent instabilities in nonlinear systems. In a sample demonstration, the controller synthesized by Perfect Moment acquires a target 300 times faster than the

DTIC QUALITY INSPECTED 3

controller based on local linear methods.

- Mike Eisenberg's Kineticist's Workbench program combines numerical simulation with a variety of symbolic methods in order to characterize reaction mechanisms in terms that are meaningful to the working chemist, suggesting ways to simplify complex mechanisms.

- Andy Berlin has demonstrated that active control may be used to increase the load-bearing strength of a structure by stabilizing the buckling modes of compressively-loaded structural members. Active control of buckling has the potential to produce structures that are both lighter and stronger than would otherwise be feasible. M.I.T. has applied for a patent on active techniques to control buckling.

- Phil Greenspun has developed a system, Site Controller, intended to assist civil engineers in the design, estimation, and construction of earthworks, including hazardous waste site remediation. The core of Site Controller is a site modelling system that represents existing and prospective terrain shapes, road and property boundaries, hydrology, important features such as trees, utility lines, and general user notations. Around this core are analysis, simulation, and vehicle control tools. The system works well enough that Caterpillar was able to use Site Controller to supervise operations of a 160-ton autonomous truck.

- Kleanthes Koniaris and Guillermo Juan Rozas have developed Vector Length Caching (VLC), a novel hardware technique, which provides high-speed bounds-checking for array accesses. M.I.T. is applying for a patent on this technique.

- Guillermo Juan Rozas introduced "translucent procedures" which are like ordinary procedures in all ways except that their opacity is violated in a controlled way, so as to allow a limited kind of structure matching against some representation of the procedure text and its environment. Dr. Rozas has implemented a system based on translucent procedures, and he has used this system to develop a novel equation-solving system, a compiler for threaded interpreters, and another visit to Roylance's method for the automatic construction of scientific subroutines to compute special functions.

4

3 Significant events

3.1 Increasing the strength of a column using active control

Andrew Berlin has used active control to increase the strength of a structure by preventing the catastrophic buckling of compressively loaded members. For many geometries, buckling is the factor that limits the maximum compressive force that may be applied to a member. The effectiveness of this technique has been demonstrated by actively controlling the first buckling mode of a small steel column using a tendon-based actuation strategy.

The anti-buckling technique may be employed to provide additional structural strength during events that cause unusually large compressive loads. For instance, when an aircraft makes an unusually hard landing, active control of buckling might be used to strengthen the struts in the landing gear. Another potential application is the reduction of metal fatigue in structures that undergo small-amplitude buckling due to external excitations, such as ships undergoing wave-induced whipping.

Berlin's experiments demonstrated the feasibility of using piezo-ceramic actuators to apply the actuation force. Berlin also studied the interactions among multiple active members in a railroad-style truss bridge.

Active control of buckling has the potential to produce structures that are both lighter and stronger than would otherwise be feasible. M.I.T. has applied for a patent on certain buckling control techniques.

3.2 New mathematics for parameter estimation

As part of our investigation of applications of intelligent simulation to precision measurement, Thanos Siapas has discovered a new invariant that quantifies the dependence of global geometry of dynamical systems on parameter variations. This is one of the first attempts to characterize fine changes in the geometry of phase-space structures, as opposed to coarse changes in their topology. Siapas interprets the evolution of basins of attraction as parameters change, in terms of convergence rates in an appropriate function space.

The Siapas invariant is equal to the dimension of the imbedding space minus the Hausdorff dimension of the boundary of the basin. Therefore the spatial geometric complexity of the basin boundary directly influences its

local parameter dependence. The invariant determines a power law governing the change in basin geometry as parameters very. Siapas has shown how such a power law arises mathematically.

This phenomenon occurs in a wide class of systems. Siapas has investigated the driven damped pendulum, which models Josephson junctions, charge-density-wave transport, and the phase-locked loop configuration of a voltage controlled oscillator (VCO); and a forced two-well potential system, which describes the motion of a buckled elastic beam or an electron in a plasma.

3.3 Automatic design of high-performance controllers

Feng Zhao has constructed and demonstrated a suite of programs that automatically analyzes and designs controllers for high-performance, global control of nonlinear systems. These programs combine powerful techniques from numerical and symbolic computation with novel representation and reasoning mechanisms from artificial intelligence. The two major programs in the suite of tools – Phase Space Navigator and MAPS (Modular Analyzer for Phase Spaces) – work together to visualize and model the phase-space structure of a given system. They reason about and manipulate the phase-space geometry and topology and identify optimal control paths that connect different system states.

The key novel idea in these programs is to decompose phase space into structures called flow pipes – groups of trajectories that have the same qualitative effect. This transforms the infinite number of distinct behaviors into a manageable finite set that serves as the basis for establishing reference trajectories and for navigating the system along the planned trajectories.

To demonstrate the power of this approach, Zhao used these programs to automatically synthesize (in simulation) a high-performance controller for a magnetic levitation system – the German Transrapid system. The new control system can stabilize maglev vehicles with displacements of up to 20 times those allowed in a previous controller design that was developed for the same system using linear-feedback techniques. A paper by Feng Zhao and Richard Thornton about the maglev controller designed by his program was presented at the 31st IEEE conference on Decision and Control.

4 Reports published under this effort

H. Abelson, A. Berlin, J. Katzenelson, W. McAllister, G. Rozas, G.J. Sussman, and J. Wisdom "The Supercomputer Toolkit: A general framework for special-purpose computing", *International Journal of High-Speed Electronics*, vol. 3, no. 3, 1992, pp. 337–361. Also available as MIT Artificial Intelligence Laboratory Memo number 1329, November 1991, and Hewlett-Packard Laboratories Technical Report Number HPL-94-30.

The Toolkit is a family of hardware modules (processors, memory, interconnect, and input-output devices) and a collection of software modules (compilers, simulators, scientific libraries, and high-level front ends) from which high-performance special-purpose computers can be easily configured and programmed. The hardware modules are intended to be standard, reusable parts. These are combined by means of a user-reconfigurable, static interconnect technology. The Toolkit's software support, based on novel compilation techniques, produces extremely high-performance numerical code from high-level language input, and will eventually automatically configure hardware modules for particular applications.

We have completed fabrication of the Toolkit processor module, and an eight-processor configuration is running at MIT. As a demonstration of the power of the Toolkit approach, we have used the prototype Toolkit to perform an integration of the motion of the Solar System in a computation that extends previous results by nearly two orders of magnitude.

H. Abelson, A. Berlin, J. Katzenelson, W. McAllister, G. Rozas, and G.J. Sussman, "The Supercomputer Toolkit and its Applications", *Proc. of the Fifth Jerusalem Conference on Information Technology*, Oct. 1990. Also available as AI Memo 1249.

The Supercomputer Toolkit is a proposed family of standard hardware and software components from which special-purpose machines can be easily configured. Using the Toolkit, a scientist or an engineer, starting with a suitable computational problem, will be able to readily configure a special purpose multiprocessor that attains supercomputer-class

performance on that problem, at a fraction of the cost of a general purpose supercomputer.

While the Toolkit project is not complete, we believe our results show evidence that generating special-purpose computers from standard modules can be an important method of performing intensive scientific computing. This paper briefly describes the Toolkit's hardware and software modules, the Solar System simulation, and conclusions and future plans.

H. Abelson "The Bifurcation Interpreter: A step towards the automatic analysis of dynamical systems", *Int. J. of Computers and Mathematics with Applications*, vol. 20, no. 8, June 1990, pp. 13–35. Also available as MIT Artificial Intelligence Laboratory Memo no. 1174, September 1989.

The Bifurcation Interpreter is a computer program that autonomously explores the steady-state orbits of one-parameter families of periodically-driven oscillators. To report its findings, the Interpreter generates schematic diagrams and English text descriptions similar to those appearing in the science and engineering research literature. Given a system of equations as input, the Interpreter uses symbolic algebra to automatically generate numerical procedures that simulate the system. The Interpreter incorporates knowledge about dynamical systems theory, which it uses to guide the simulations, to interpret the results, and to minimize the effects of numerical error.

H. Abelson, T. Cline, W. Harris "Symbolic computing in engineering design," *AI EDAM*, vol. 3, no 3, 1989, pp. 195–206.

Computer programs that combine traditional numeric methods with symbolic algebra and with specific knowledge of application-based techniques can provide new levels of computational support for engineering design. We illustrate this with a computer-based "control-engineer's assistant." Although this program is focussed on control-system design, it demonstrates techniques that should be widely applicable across many engineering disciplines. In particular, we show how, with symbolic computing, a computer-aided design system can usefully simulate engineering models early in the design process, before all (or any) system parameters have been specified numerically. Our system employs a

flexible, extensible, object-oriented representation for control systems, which admits multiple mathematical models of designs and provides a framework for integrating tools that operate on diverse representations.

H. Abelson, M. Eisenberg, M. Halfant, J. Katzenelson, E. Sacks, G.J. Sussman, J. Wisdom, K. Yip, "Intelligence in Scientific Computing", *Communications of the ACM*, **32**, May 1989. Also available as AI Memo 1094.

Combining numerical techniques with ideas from symbolic computation and methods incorporating knowledge of science and mathematics leads to a new category of intelligent computational tools for scientists and engineers. These tools autonomously prepare simulation experiments from high-level specifications of physical models. For computationally intensive experiments, they automatically design special-purpose numerical engines optimized to perform the necessary computations. They actively monitor numerical and physical experiments. They interpret experimental data and formulate numerical results in qualitative terms. They enable their human users to control computational experiments in terms of high-level behavioral descriptions.

A. Berlin, "Towards Intelligent Structures: Active Control of Buckling", Ph.D. Thesis, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, August 1994.

A. Berlin and R. Surati, "Partial Evaluation for Scientific Computing: The Supercomputer Toolkit Experience," *Proc of ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation 1994*, Orlando, FL 1994. Also MIT Artificial Intelligence Laboratory Memo 1487, May 1994.

We describe the key role played by partial evaluation in the Supercomputer Toolkit, a parallel computing system for scientific applications that effectively exploits the vast amount of parallelism exposed by partial evaluation. The Supercomputer Toolkit parallel processor and its associated partial evaluation-based compiler have been used extensively by scientists at M.I.T., and have made possible recent results in astrophysics showing that the motion of the planets in our solar system is chaotically unstable.

A. Berlin and G.J. Sussman, "Increasing the Compressive Strength of a Column via Active Control", *Proceedings of the Third International Conference on Adaptive Structures*, Technomic Publishing Company, San Diego, CA, Oct 1992.

> We have used active control to increase the strength of a structure by preventing the catastrophic buckling of compressively loaded members. For many geometries, buckling is the factor that limits the maximum compressive force that may be applied to a member; for long slender members, the strength limitation imposed by buckling is several orders of magnitude more important than other load-limiting factors, such as plastic deformation. Experimental results obtained using a prototype actively-controlled column indicate that this approach has the potential to produce structures that are both lighter and stronger than would otherwise be feasible.

A. Berlin, "A compilation strategy for numerical programs based on partial evaluation", MIT Artificial Intelligence Laboratory Technical Report no 1144, July 1989.

> This work demonstrates how partial evaluation can be put to practical use in the domain of high-performance numerical computation. I have developed a technique for performing partial evaluation by using placeholders to propagate intermediate results, and have implemented a prototype compiler based on this technique. For an important class of numerical programs, this compiler improves performance by an order of magnitude over conventional compilation techniques. I also show that by eliminating inherently sequential data-structure references, partial evaluation exposes the low-level parallelism inherent in a computation. I have implemented a parallel program generator, as well as several analysis programs that study the tradeoffs involved in the design of an architecture that can effectively utilize this parallelism. I present these results using the 9-body gravitational attraction problem as an example.

A. Berlin and D. Weise, "Compiling Scientific Code using Partial Evaluation", *IEEE Computer* December 1990. Also available as MIT Artificial Intelligence Laboratory Memo no., 1145, July 1989.

We ha.e developed a compiler that uses partial evaluation and parallel sch.. luling techniques. Where conventional compilers compile a pro-gram without any knowledge of the data the program will be run on. our system uses information about the data when transforming the pro-gram. This technique, by eliminating nearly all the user's control and data abstractions, produces high-performance code. For an important class of numerical programs, partial evaluation dramatically improves performance: we have achieved speedups over conventionally compiled code that range from seven times faster to ninety one times faster. We also show how partial evaluation can be applied to the program-ming of parallel computers. By eliminating inherently sequential data structure references and their associated conditional branches, partial evaluation exposes the low-level parallelism inherent in a computation. We present the results of applying a parallel scheduler to a partially evaluated program that simulates the motions of nine bodies under mutual gravitational attraction.

A. Berlin, "Partial Evaluation Applied to Numerical Computation", *Proc. 1990 Lisp and Functional Programming Conference*, Nice France, June 1990.

There have been many demonstrations that the expressive power of Lisp can greatly simplify the process of writing numerical programs, but at the cost of reduced performance. I show that by coupling Lisp's abstract, expressive style of programming with a compiler that uses par-tial evaluation, data abstractions can be eliminated at compile time. producing extremely high-performance code. For an important class of numerical programs, partial evaluation achieves order-of-magnitude speed-ups over conventional Lisp compilation technology. This ap-proach has proven to be especially effective when used in conjunction with schedulers for VLIW and highly pipelined architectures, because the elimination of data structures and procedural abstractions exposes the low-level parallelism inherent in a computation.

M. Birkholz "Emacs Lisp in Edwin Scheme", MIT Artificial Intelligence Lab Report TR-1451, September 1993.

The MIT-Scheme program development environment includes a general-purpose text editor, Edwin, that has an extension language, Edwin

Scheme. Edwin is very similar to another general-purpose text editor, GNU Emacs, which also has an extension language, Emacs Lisp. The popularity of GNU Emacs has lead to a large library of tools written in Emacs Lisp. The goal of this thesis is to implement a useful subset of Emacs Lisp in Edwin Scheme. This subset was chosen to be sufficient for simple operation of the GNUS news reading program.

E. Bradley, "Using Chaos to Broaden the Capture Range of a Phase-Locked Loop", *IEEE Trans. Circuits and Systems*, November 1993. Also UColorado CU-CS-669-93 Memo, September 1993.

Chaos is common in physical systems, but control engineers have, until very recently, deemed it undesirable and gone to great lengths to avoid it. Such tactics can represent a needless sacrifice in performance — chaos has a variety of useful properties that can significantly enhance engineering designs. In particular, phase-space trajectories on a chaotic attractor densely cover a set of non-zero measure, making all points in that set *reachable* from any initial condition in its basin of attraction. Moreover, the size, shape, and position of the attractor are affected by changes in system parameters, following certain highly characteristic patterns. These properties have been used, in simulations, to broaden the capture range of the common phase-locked loop circuit. An external modulating input is used to throw the unlocked loop into a chaotic regime that overlaps the original capture range. The chaos-inducing modulation is then turned off, allowing the loop's original dynamics to capture the signal. This technique is not limited to this system or even to this branch of engineering; it applies, modulo a few constraints and limitations, to any system that exhibits chaotic behavior and that is subject to design requirements.

E. Bradley, "Taming Chaotic Circuits", Ph.D. Thesis, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, 1992. Also available as MIT AITR 1388, September 1992.

Control algorithms that exploit chaotic behavior can vastly improve the performance of many practical and useful systems. The program *Perfect Moment* is built around a collection of such techniques. It autonomously explores a dynamical system's behavior, using rules embodying theorems and definitions from nonlinear dynamics to zero in

on interesting and useful parameter ranges and state-space regions. It then constructs a reference trajectory based on that information and causes the system to follow it. This program and its results are illustrated with several examples, among them the phase-locked loop, where sections of chaotic attractors are used to increase the capture range of the circuit.

E. Bradley and F. Zhao, "Phase-Space Control System Design" *Proceedings of the 1992 IEEE Symposium on Computer-Aided Control System Design*, Napa, California, March 1992. Also in *IEEE Control Systems Magazine*, April 1993.

This paper describes a computational environment that has been developed to aid control system design for a particular class of nonlinear applications. The analysis and design tools that comprise this environment are based upon knowledge about phase-space dynamics of nonlinear and chaotic systems. We describe two implemented, complementary programs that exploit the special properties of such systems to automatically synthesize powerful control systems. *Phase Space Navigator* visualizes phase-space dynamics through flow pipes and navigates systems along automatically synthesized reference trajectories. *Perfect Moment* identifies and uses chaotic phase-space features like strange attractors in its segmented control trajectories, gaining otherwise-unobtainable performance. Though the phase-space paradigm is very powerful, its global and computationally-intensive nature makes some of the techniques that exploit it difficult to implement. Fast computers and powerful computational techniques that combine symbolic/numeric and algebraic/geometric computing with new reasoning mechanisms from artificial intelligence make this paradigm feasible in spite of its inherent demands.

E. Bradley, "A Control Algorithm for Chaotic Physical Systems" *Proceedings of the First Experimental Chaos Conference*, Washington DC, October 1991, World Scientific. Also available as MIT Artificial Intelligence Lab Memo 1323.

Control algorithms that exploit chaos's unique properties can vastly improve the performance of many practical and useful systems. The program *Perfect Moment* is built around such an algorithm. Given a

13

differential equation and two points in the system's state space, it automatically maps the space, chooses a set of trajectory segments from the maps, uses them to construct a composite path between the points, and causes the system to follow that path by monitoring the state and switching parameter values at the segment junctions. The creation of and search through the maps are computationally intensive processes. However, the sensitivity of a chaotic system's state-space topology to the parameters of its equations and the sensitivity of the paths of its trajectories to the initial conditions make this approach rewarding in spite of its computational demands. This program and its results are illustrated with several examples, among them the driven single pendulum and its electronic analog, the phase-locked loop. In this particular case, strange attractor bridges, which traverse boundaries of basins of attraction and thus alter the reachability of different state space points, can be used to broaden the capture range of the circuit.

E. Bradley, "Control Algorithms for Chaotic Systems", *Lecture Notes in Control and Information Sciences*, Vol. 165, G. Jacob and F. Lamnabhi-Lagarrigue, Eds., Springer-Verlag, December 1991. Also available as MIT Artificial Intelligence Lab Memo 1278.

This paper presents techniques that actively exploit chaotic behavior to accomplish otherwise-impossible control tasks. The state space is mapped by numerical integration at different system parameter values and trajectory segments from several of these maps are automatically combined into a path between the desired system states. A fine-grained search and high computational accuracy are required to locate appropriate trajectory segments, piece them together and cause the system to follow this composite path. The sensitivity of a chaotic system's state-space topology to the parameters of its equations and of its trajectories to the initial conditions make this approach rewarding in spite of its computational demands. Boundaries of basins of attraction can be breached, vastly altering both global and local convergence properties. Strange attractor bridges can be found that connect previously unreachable points. Examples of both are shown.

E. Bradley, "Causes and Effects of Chaos", MIT Artificial Intelligence Lab Memo 1216, September, 1990.

Most of the recent literature on chaos and nonlinear dynamics is written either for popular science magazine readers or for advanced mathematicians. This paper gives a broad introduction to this interesting and rapidly growing field at a level that is between the two. The graphical and analytical tools used in the literature are explained and demonstrated, the rudiments of the current theory are outlined and that theory is discussed in the context of several examples: an electronic circuit, a chemical reaction and a system of satellites in the solar system.

D. Chasman, R.J. Silbey and M. Eisenberg, "Massively Parallel Time Dependent Wave-Packet Calculations in a Classically Chaotic Potential", *Chem. Phys. Lett.*, No 175, December, 1990.

The time dependent solution to Schrödinger's equation is obtained using a parallel algorithm on a massively parallel computer. The algorithms used are reviewed and their running speed on serial and parallel architectures is contrasted. The system studied has a classical analogue which undergoes a gradual transition from regular to chaotic motion as a function of a parameter of the Hamiltonian. The decay of the autocorrelation function is studied. In addition, the time dependence of the information measure and spatial correlation is studied.

M. Eisenberg, "The Kineticist's Workbench: Combining Symbolic and Numerical Methods in the Simulation of Chemical Reaction Mechanisms", Ph.D. Thesis, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, 1991. Also availible as MIT AITR 1306.

The Kineticist's Workbench is a program that expands the expressiveness of computer simulation: it combines symbolic and numerical techniques in simulating a particular class of complex systems—chemical reaction mechanisms. The Workbench assists chemists by predicting, generating, and interpreting numerical data. Prior to simulation, it analyzes a given mechanism to predict that mechanism's behavior; it then simulates the mechanism numerically; and afterward, it interprets and summarizes the data that it has generated. In performing these tasks, the Workbench brings to bear a wide variety of techniques: graph-theoretic algorithms (for the analysis of mechanisms), traditional

numerical simulation methods, and algorithms that examine the simulation results and reinterpret them in qualitative terms. Moreover, the Workbench can use symbolic procedures to help guide or simplify the task of numerical simulation; and it can sometimes use its summary of numerical results to suggest additional numerical analysis. Thus, it serves as a prototype for a new class of scientific computational tools—tools that provide symbiotic collaborations between qualitative and quantitative methods.

M. Eisenberg, "Combining Qualitative and Quantitative Techniques in the Simulation of Chemical Reaction Mechanisms", *Proc. Eastern Multi-Conference (AI and Simulation)*, Nashville, TN, April 1990.

In simulating chemical reaction mechanisms, a combination of both qualitative analysis and traditional numerical techniques can produce insights not obtained with either approach alone. For instance, if a program can deduce on qualitative grounds that a system must reach equilibrium, it can use this information to steer numerical simulations, to choose appropriate numerical methods, and to produce qualitative interpretations of numerical results.

This paper describes a program that implements mixed qualitative-quantitative analysis of chemical mechanisms, whose simulation usually entails numerically integrating (possibly large) systems of coupled nonlinear differential equations. The program guides the integration by means of qualitative information that it deduces from the structure of the mechanisms prior to numerical simulation.

M. Eisenberg, "Descriptive Simulation: Combining Symbolic and Numerical Methods in the Analysis of Chemical Reaction Mechanisms", *Artificial Intelligence in Engineering*, July 1990. Also available as AI Memo no. 1171.

The Kineticist's Workbench is a computer program currently under development whose purpose is to help chemists understand, analyze, and simplify complex chemical reaction mechanisms. This paper discusses one module of the program that numerically simulates mechanisms and constructs qualitative descriptions of the simulation results. These descriptions are given in terms that are meaningful to the working chemist

16

(e.g. steady states, stable oscillations, etc.); and the descriptions (as well as the data structures used to construct them) are accessible as input to other programs.

P. Greenspun, "Site Controller: A system for computer-aided civil engineering and construction", MIT Artificial Intelligence Laboratory Technical Report 1408, May 1993.

A revolution in earthmoving, a $100 billion industry, can be achieved with three components: the GPS location system, sensors and computers in earthmoving vehicles, and Site Controller, a central computer system that maintains design data and directs operations. The first two components are widely available; I built Site Controller to complete the triangle and describe it here.

Civil engineering challenges computer scientists in the following areas: computational geometry, large spatial databases, floating-point artihmetic, software reliability, management of complexity, and real-time control. Site Controller demonstrates that most of these challenges may be surmounted by the use of state-of-the-art algorithms, object databases, software development tools, and code-generation techniques. The system works well enough that Caterpillar was able to use Site Controller to supervise operations of a 160-ton autonomous truck.

Site Controller assists civil engineers in the design, estimation, and construction of earthworks, including hazardous waste site remediation. The core of Site Controller is a site modelling system that represents existing and prospective terrain shapes, road and property boundaries, hydrology, important features such as trees, utility lines, and general user notations. Around this core are analysis, simulation, and vehicle control tools. Integrating these modules into one program enables civil engineers and contractors to use a single interface and database throughout the life of a project.

This area is exciting because so much of the infrastructure is in place. A small effort by computer scientists could cut the cost of earthmoving in half, enabling poor countries to build roads and rich countries to clean up hazardous waste.

C. Hanson, "MIT Scheme Reference Manual", MIT Artificial Intelligence

17

Laboratory Technical Report 1281, January 1991.

MIT Scheme is an implementation of the Scheme programming language that runs on many popular workstations. The MIT Scheme Reference Manual describes the special forms, procedures, and datatypes provided by the implementation for use by application programmers.

C. Hanson, "Efficient Stack Allocation for Tail-Recursive Languages" , *Proc. ACM Conference on Lisp and Functional Programming*, pp. 106-118, Nice France, June 1990.

The Scheme dialect of Lisp is properly tail-recursive – it relies entirely on procedure calls to express iteration. As elegant as tail-recursion may be from the perspective of the programmer or the theoretician, it poses challenges for the compiler designer. This paper describes stack-allocation techniques for compiling tail-recursive languages. We have implemented these techniques in the MIT Scheme compiler. We show that the performance is comparable to implementations of non-tail-recursive languages. In particular, the code sequences generated for tail-recursive procedure calls are as efficient as those that implement the special-purpose iteration constructs on non-tail-recursive languages.

J. Katzenelson, "Computational Structure of the N-body Problem", *SIAM J. Sci. Stat. Computing*, vol. 10, no. 4, 1989, pp. 787–815. Also available as AI memo number 1039.

This work combines tree algorithms for the N-body problem where the number of particles is on the order of a million. The main concern of this work is the organization and performance of these computations on parallel computers. The work introduces the formulation of the N-body problem as a set of recursive equations based on a few elementary functions. It is shown that both the algorithm of Barnes-Hut and that of Greengard-Rokhlin satisfy these equations using different elementary functions. The recursive formulation leads directly to a computational structure in the form of a pyramid-like graph, where each vertex is a process, and each arc a communication link.

Koniaris, K. and Rozas, G.J., "Vector Length Caching: A Means for Fast and Safe Array Access", to be AI Memo, Massachusetts Institute of

Technology Artificial Intelligence Laboratory. Cambridge, MA, April 1993.

Vector Length Caching (VLC) is a technique which provides high-speed hardware bounds checking for array accesses. The entries of a vector V, of length N, starting at address $\mathcal{V}$, are usually accessed (read/written) by means of LOAD/STORE instructions. VLC introduces two additional instructions, VLOAD/VSTORE, and a simple software convention, that N is stored at address $\mathcal{V} - 1$. These new instructions look and work like LOAD/STORE, except that they also provide automatic bounds checking. The first time that VLOAD/VSTORE accesses an entry of V, there will be an automatic one-time hardware-imposed penalty of a read from address $\mathcal{V} - 1$, where N is placed into the VLC; afterwards, these new instructions run at exactly the same speed as their conventional counterparts, for *any* element of V. VLC enables many programs to become either more robust or faster, particularly if they are written in *Lisp* or *ML*.

LaMacchia, B. and Nieh, J., "The Standard Map Machine", AI Memo 1165, Massachusetts Institute of Technology Artificial Intelligence Laboratory. Cambridge, MA, September 1989.

We have designed the Standard Map Machine (SMM) as an answer to the intensive computational requirements involved in the study of chaotic behavior in nonlinear systems. The high-speed and high-precision performance of this computer is due to its simple architecture specialized to the numerical computations required of nonlinear systems. In this report, we discuss the design and implementation of this special-purpose machine.

M. J. Coster, B. A. LaMacchia, A. M. Odlyzko and C. P. Schnorr, "Improved Low-density Subset-Sum Algorithms", *Advances in Cryptology, Proceedings of Eurocrypt '91*, 1991. Also in Computational Complexity **2** (1992) no. 2, 111-128, with A. Joux and J. Stern.

The general subset sum problem is NP-complete. However, there are two algorithms, one due to Brickell and the other to Lagarias and Odlyzko, which in polynomial time solve almost all subset sum problems of sufficiently low density. Both methods rely on basis reduction algorithms to find short non-zero vectors in special lattices. The

Lagarias-Odlyzko algorithm would solve almost all subset sum problems of density $< 0.6463\ldots$ in polynomial time if it could invoke a polynomial-time algorithm for finding the shortest non-zero vector in a lattice. This paper presents two modifications of that algorithm, either one of which would solve almost all problems of density $< 0.9408\ldots$ if it could find shortest non-zero vectors in lattices. These modifications also yield dramatic improvements in practice when they are combined with known lattice basis reduction algorithms.

B. A. LaMacchia and A. M. Odlyzko, "Computation of Discrete Logarithms in Prime Fields", *Designs, Codes and Cryptography*, No. 1, 1991.

The presumed difficulty of computing discrete logarithms in finite fields is the basis of several popular public key cryptosystems. The secure identification option of the Sun Network File System, for example, uses discrete logarithms in a field $GF(p)$ with p a prime of 192 bits. This paper describes an implementation of a discrete logarithm algorithm which shows that primes of under 200 bits, such as that in the Sun system, are very insecure. Some enhancements to this system are suggested.

B. A. LaMacchia and A. M. Odlyzko, "Solving large sparse linear systems over finite fields", *Advances in Cryptology: Proceedings of Crypto '90*, S. Vanstone, ed., to be published.

Many of the fast methods for factoring integers and computing discrete logarithms require the solution of large sparse linear systems of equations over finite fields. This paper presents the results of implementations of several linear algebra algorithms. It shows that very large sparse systems can be solved efficiently by using combinations of structured Gaussian elimination and the conjugate gradient, Lanczos, and Wiedemann methods.

M. Lee "Summarizing Qualitative Behavior from Measurements of Nonlinear Circuits", AI TR 1125, MIT Artificial Intelligence Laboratory, May 1989

The process of exploring the behavior of nonlinear, dynamical systems can be a time-consuming and tedious process. In this thesis, I have

written a program which automates much of the work of an experimental dynamicist. In particular, the program automatically characterizes the behavior of any driven, nonlinear, electrical circuit exhibiting interesting behavior below the 10 Mhz range. In order to accomplish this task, the program can autonomously select interesting input parameters, drive the circuit, measure its response, perform a set of numeric computations on the measured data, interpret the results and decompose the circuit's parameter space into regions of qualitatively distinct behavior. The output is a two-dimensional portrait summarizing the high-level, qualitative behavior of the nonlinear circuit for every point in the graph as well as an accompanying textual explanation describing any interesting patterns observed in the diagram. In addition to the graph and the text, the program generates a symbolic description of the circuit's behavior. This intermediate data structure can then be passed onto other programs for further analysis.

J. Miller, C. Niedner, and J. London, "The organization engine: Virtual data integration", Submitted to the *Symposium on Computer Applications in Medical Care*, November 1992.

J. Miller and G.J. Rozas, "Free Variables and First-Class Environments", *Journal of Lisp and Symbolic Computation*, No. 4, 1991.

A simple set of extensions to the SCHEME language removes the need for a distinguished top level interaction environment by providing first-class environments. These extensions also provide a powerful mechanism for code packaging and may be used to implement simple object-oriented systems. In addition, a mechanism is presented that implements compiled references to free variables as efficiently as in languages like C, provided the code does not directly manipulate first-class environments. The mechanism requires a simple static analysis performed by the compiler and meshes with a slower mechanism used by both interpreted code and compiled code that manipulates first-class environments.

J. Nieh, "Using Special-purpose Computing to Examine Chaotic Behavior in Nonlinear Mappings", AI Technical Report 1139, Massachusetts Institute of Technology Artificial Intelligence Laboratory. Cambridge, MA, September 1989.

Most physical phenomena are nonlinear in nature and exhibit the complicated and seemingly random behavior known as chaos. Studying chaotic behavior in nonlinear systems requires numerous computations in order to simulate the behavior of such systems. The Standard Map Machine (SMM) was designed and implemented as a special computer for performing these intensive computations with high-speed and high-precision. SMM's impressive performance is due to its simple architecture specialized to the numerical computations required of nonlinear systems. This report discusses the design and implementation of the Standard Map Machine and its use in the study of nonlinear mappings, in particular, the study of the standard map.

J. Rees, "Implementing Lexically Scoped Macros", *Lisp Pointers* VI(1), ACM SIGPLAN, 1993.

J. Rees, "The June 1992 Meeting [of Scheme Report authors]", *Lisp Pointers* V(4), ACM SIGPLAN, 1992.

J. Rees, "User-Defined Data Types", *Lisp Pointers* V(3): 39–46, ACM SIGPLAN, 1992.

J. Rees and B. Donald, "Program mobile robots in Scheme", *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2681–2688, 1992.

J. Rees and W. Clinger, "Macros That Work", *Conference Record of the Eighteenth Annual ACM Symposium on Principles of Programming Languages*, 1991.

This paper describes a modified form of Kohlbecker's algorithm for reliably hygienic (capture-free) macro expansion in block-structured languages, where macros are source-to-source transformations specified using a high-level pattern language. Unlike previous algorithms, the modified algorithm runs in linear instead of quadratic time, copies few constants, does not assume that syntactic keywords are reserved words, and allows local (scoped) macros to refer to lexical variables in a referentially transparent manner.

Syntactic closures have been advanced as an alternative to hygienic macro expansion. The problem with syntactic closures is that they are

inherently low-level and therefore difficult to use correctly, especially when syntactic keywords are not reserved. It is impossible to construct a pattern-based, automatically hygienic macro system on top of syntactic closures because the pattern interpreter must be able to determine the syntactic role of an identifier (in order to close it in the correct syntactic environment) before macro expansion has made that role apparent.

Kohlbecker's algorithm may be viewed as a book-keeping technique for deferring such decisions until macro expansion is locally complete. Building on that insight, this paper unifies and extends the competing paradigms of hygienic macro expansion and syntactic closures to obtain an algorithm that combines the benefits of both.

Several prototypes of a complete macro system for Scheme have been based on the algorithm presented here.

J. Rees and W. Clinger, eds., The revised[4] report on the algorithmic language Scheme", *Lisp Pointers* 4(3), ACM, 1991.

The report gives a defining description of the programming language Scheme. Scheme is a statically scoped and properly tail-recursive dialect of the Lisp programming language invented by Guy Steele Jr. and Gerald Jay Sussman. It was designed to have an exceptionally clear and simple semantics and few different ways to form expressions. A wide variety of programming paradigms, including imperative, functional and message passing styles, find convenient expression in Scheme.

J. Rees. "Modular Macros", S.M. thesis, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, May 1989.

Macros provide a general way to extend the syntax of a language. A macro defines a new language construct via a syntactic transformation function that rewrites an instance of the new construct as an expression using other language constructs. However, because they are textual rewrite rules, macros suffer from the drawbacks of call-by-text: lexical scoping rules are not obeyed, so free variables in both the uses of the macro and in its definition may be inadvertently captured. Thus the details of the implementation of the macro may be exposed to the

23

macro's clients. A solution to this difficulty is given by a device called *syntactic closures*. Like an Algol 60 thunk, a syntactic closure packages an expression with its proper lexical context before the expression is substituted into a different context. I have implemented an extended Scheme dialect that supports the practical use of syntactic closures for writing macros in modularly constructed programs.

M. Resnick, F. Martin, and B. Silverman, "Braitenberg Bricks: A LEGO-based Creature-Construction Kit", *Proc. Artificial Life Conference II*, Santa Fe NM, February 1990.

We developed LEGO/Logo as a type of Artificial Life Construction Kit for children. Using LEGO/Logo, children build artificial creatures using LEGO bricks, motors, gears, and sensors, and they write Logo programs to control the creatures' behaviors. The creatures are wired to a personal computer through a custom interface box.

Braitenberg Bricks is a alternate construction kit that puts computational power inside the LEGO bricks themselves – enabling children to create autonomous, free-ranging creatures. With Braitenberg Bricks, children can construct creatures resembling those described in Valentino Braitenberg's Vehicles. Children "program" the creatures by selecting various Braitenberg Bricks, plugging them onto the creature, and wiring them together. The B-Brick set includes effectors (motors, lights, beepers), sensors (sound, light, touch), and logic elements (and's, or's, flip-flop's).

M. Resnick, "Logo: A Children's Environment for Exploring Self-Organizing Behavior", *Proc. Artificial Life Conference II*, Santa Fe NM, February 1990.

In a growing number of research fields (including Artificial Life), researchers are adopting self-organization as a paradigm for understanding the workings of the world. But there is a problem: most people have difficulty thinking about and understanding self-organizing phenomena. One solution is to provide people with computational environments in which they can simulate and "play with" self-organizing systems. This paper describes such an environment, designed particularly for pre-college students. The environment, called *Logo, is based

on two primary types of objects: turtles (creatures in the world) and patches (pieces of the world). Users can simulate self-organizing behavior by programming local interactions among the turtles and patches. The paper presents three sample *Logo simulations of self-organizing systems. The goal of these simulations (and *Logo in general) is not to perfectly simulate life-like behavior, but rather to help people explore core concepts of self-organization.

M. Resnick, "MultiLogo: A Study of Children and Concurrent Programming", *Interactive Learning Environments*, Vol. 1, No. 3, 1991.

In a growing number of computer applications (such as animation, robotics, and desktop video), people need to control several objects at the same time. But concurrent actions are difficult to program with traditional sequential programming languages. This paper describes an extension of Logo, called MultiLogo, that provides new constructs and metaphors for controlling concurrent actions. MultiLogo is designed with non-experts in mind; it places special emphasis on language learnability. To explore how children appropriate new ideas about concurrency, I conducted an experimental study with a group of elementary-school students. The students used MultiLogo to control simple robotic devices built out of LEGO bricks. In analyzing the children's work, I develop three primary categories of MultiLogo programming bugs: problem-decomposition bugs, synchronization bugs, and object-oriented bugs. Based on the results, I recommend ways to improve the design and teaching of concurrent programming languages for non-experts.

M. Resnick, "Animal Simulations with *Logo: Massive Parallelism for the Masses", *From Animals to Animats*, J.A. Meyer and S. Wilson editors, MIT Press, 1991.

L. Rodriguez and G. Kiczales, "Efficient Method Dispatch in PCL", *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, pages 99-105, June 1990. Also to appear in *Object Oriented Programming*, Andreas Paepcke, editor. MIT Press, forthcoming.

This paper describes a mechanism for implementing CLOS method dispatch efficiently on stock hardware, in the current generation of Com-

mon Lisp implementations. This mechanism is implemented in the newest version of PCL, a portable implementation of CLOS, and runs in more than ten Common Lisps. This work is based on a careful analysis of the behavior of existing CLOS programs. The method dispatch mechanism differs from previously published work in three important ways. First, the use of a new hashing algorithm improves memoization table density and distribution. Second, the selection of memoization table format based on the dynamic history of each generic function makes it possible to store information in the memoization tables more efficiently and do the runtime method dispatch more quickly. Third, lazy updating techniques are used to speed interactive programming environment response without undue degradation of program execution.

Luis H. Rodriguez Jr. "A Study on the Viability of a Production-Quality Metaobject Protocol-based Statically Parallelizing Compiler" *Proceedings of the IMSA '92 Workshop on Reflection and Meta-level Architectures*, November, 1992.

A compiler that automatically chooses a program's parallelization is often unable to choose either the best one or the particular one that a programmer has in mind. This has led to systems that provide the programmer with explicit control over a program's parallelization, for example, via compiler pragmas. The pragma approach is like the metaobject protocol (MOP) approach in that pragmas provide control over what would otherwise be hidden aspects of an implementation. However, it differs because the set of pragmas is fixed, thereby limiting the amount of control provided. We investigated whether it was possible to increase the amount of control using the full MOP approach. We were in fact successful, but the resulting MOP differs from previous ones in that it is present at compile-time rather than at run-time. In this paper, we compare the MOP approach with other approaches, and discuss what is needed in order to produce a production-quality MOP-based statically parallelizing compiler.

Luis H. Rodriguez Jr. "Coarse-Grained Parallelization Using Metaobject Protocols" *MIT Master's Thesis*, September, 1991. Also appears as Xerox PARC System Sciences Laboratory Technical Report SSL-91-

06.

Currently, there exist different approaches to parallelizing a computation at a coarse-grain. One approach is to parallelize a computation by placing compiler declarations in the source code. This approach makes it possible to conceptually separate the code describing a computation from the code (i.e. declarations) describing its parallelization. This approach also makes it possible to explicitly control a computation's parallelization, thereby providing an opportunity to increase the computation's performance. However, there are cases when it is not reasonable to expect a desired mechanism for concurrency to be supported by the available, and fixed set of declarations, such as when the desired mechanism is highly specialized to a given computation and target architecture. In these cases, the declaration-based approach fails. In this thesis, we demonstrate that *metaobject protocols* (MOPs) can solve this problem. Under the MOP approach, a computation is parallelized by *marking* source code expressions with marks supported by the compiler. Marks, like declarations, are used to separate a computation from its parallelization. When the supported marks can not be used to express a desired mechanism for concurrency, the MOP is used to incrementally augment the compiler's parallelization strategy to support the desired mechanism. The MOP is a model of the compiler's parallelization strategy that provides the knowledge necessary to augment the strategy incrementally, without exposing arbitrary or irrelevant implementation details. In order to demonstrate the effectiveness of the MOP-based approach, we present Anibus, a MOP-based compiler. We give several examples of using marks, and of incrementally augmenting Anibus's parallelization strategy. The examples include two implementations of the n-body gravity problem.

G.J. Rozas, "Translucent Procedures, Abstraction without Opacity", Ph.D. Thesis, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, May 1993.

This report introduces *translucent procedures* as a new mechanism for implementing behavioral abstractions. Like an ordinary procedure, a translucent procedure can be invoked, and thus provides an obvious way to capture a *behavior*. Translucent procedures, like ordinary pro-

cedures, can be manipulated as first-class objects and combined using functional composition. But unlike ordinary procedures, translucent procedures have structure that can be examined in well-specified non-destructive ways, without invoking the procedure.

I have developed an experimental implementation of a normal-order lambda-calculus evaluator augmented with novel reflection mechanisms for controlled violation of the opacity of procedures. I demonstrate the utility of translucent procedures by using this evaluator to develop large application examples from the domains of graphics, computer algebra, compiler design, and numerical analysis.

A. Siapas, "A Global Approach to Parameter Estimation of Chaotic Dynamical Systems", AI Memo 1402, December 1992.

We present a novel approach to parameter estimation of systems with complicated dynamics, as well as eveidence for the existence of a universal power law that enables us to quantify the dependence of global geometry on small changes in the parameters of the system. This power law gives rise to what seems to be a new dynamical system invariant.

P.A. Skordos, "Initial and boundary conditions for the lattice Boltzmann method." *Physical Review E*, Vol. 48, No. 6, pp. 4823-4842. December 1993.

An alternative approach of implementing initial and boundary conditions for the lattice Boltzmann method is presented. The basic idea is to calculate the lattice Boltzmann populations at a boundary node from the fluid variables that are specified at this node using the gradients of the fluid velocity. The numerical performance of the lattice Boltzmann method is tested on several problems with exact solutions and is also compared to an explicit finite difference projection method. The discretization error of the lattice Boltzmann method decreases quadratically with finer resolution both in space and in time. The roundoff error of the lattice Boltzmann method creates problems unless double precision arithmetic is used.

P. A. Skordos, "The Cost of Compressible Dynamics, Time-reversibility, Maxwell's demon, and the second law", Physical Review E, August 1993.

A tantalizing version of Maxwell's demon is presented which appears to operate reversibly. A container of hard core disks is separated into two chambers of equal volume by a membrane that selects which disk can penetrate depending on the disk's angle of incidence. It is shown that the second law of thermodynamics requires the incompressibility of microscopic dynamics or an appropriate energy cost for compressible microscopic dynamics.

P. A. Skordos and W. H. Zurek, "Maxwell's Demon, Rectifiers, and the Second Law: Computer Simulation of Smoluchowski's Trapdoor" American Journal of Physics, **60** (10) 876—882, October, 1992. Also as MIT AI Memo 1319, September 1991, and as Los Alamos Report LA-UR-91-2553.

An automated Maxwell's demon inspired by Smoluchowski's ideas of 1912 is simulated numerically. Two gas chambers of equal area are connected via an opening that is covered by a trapdoor. The trapdoor can open to the left but not to the right, and is intended to rectify naturally occurring variations in density between the two chambers. The simulation results confirm that though the trapdoor behaves as a rectifier when large density differences are imposed by external means, it can not extract useful work from the thermal motion of the molecules when left on its own.

R. Surati, "A Parallelizing Compiler Based on Partial Evaluation", S.B. Thesis, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, May 1992. Also available as TR-1377, MIT Artificial Intelligence Laboratory, July, 1993.

This thesis demonstrates a compiler that uses partial evaluation to achieve outstandingly efficient parallel object code from very high-level source programs. The source programs are ordinary Scheme numerical programs, written abstractly, with no attempt to structure them for parallel execution. The compiler identifies and extracts parallelism completely automatically; nevertheless, it achieves speedups equivalent to or better than the best observed results achieved by previous supercomputer compilers that require manual restructuring of code.

This thesis represents one of the first attempts to capitalize on partial evaluation's ability to expose low-level parallelism. To demonstrate

the effectiveness of this approach, we targeted the compiler for the Supercomputer Toolkit, a parallel machine with eight VLIW processors. Experimental results on integration of the gravitational n-body problem show that the compiler, generating code for 8 processors, achieves a factor of 6.2 speedup over an almost optimal uniprocessor computation, despite the Toolkit's relatively slow interprocessor communication speed. This compares with an average speedup factor of 4.0 on 8 processors obtained at University of Illinois using manual code restructuring of a suite of benchmarks for the Cray YMP.

G.J. Sussman and J. Wisdom, "Chaotic Evolution of the Solar System", *Science*, Volume 257, July 1992. Also AIMemo 1359, March 1992.

The evolution of the entire planetary system has been numerically integrated for a time span of nearly 100 million years. This calculation confirms that the evolution of the solar system as a whole is chaotic, with a time scale of exponential divergence of about 4 million years. Additional numerical experiments indicate that the Jovian planet subsystem is chaotic, although some small variations in the model can yield quasiperiodic motion. The motion of Pluto is independently and robustly chaotic.

G.J. Sussman, "The last three minutes: Issues in Gravitational Wave Measurements of coalescing compact binaries," C. Cutler, T.A. Apostolatos, L. Bildsten, L.S. Finn, E.E. Flanagan, D. Kennefick, D.M. Markovic, A. Ori, E. Poisson, G.J. Sussman, and K.S. Thorne, Physical Review Letters, Vol 70, No. 20, p 2984-2987, May 1993.

W-T. Tsai, D. Yue, K. Yip, "Resonantly excited regular and chaotic motions in a rectangular wave tank", *J. Fluid Mech.*, Vol. 216, pp 343-380, (1990).

We consider the resonant excitiation of surface waves inside a rectangular wave tank of arbitrary water depth with a flap-type wavemaker on one side. Depending on the length and width of the tank rlative to the sinusoidal forcing frequency of the wave paddle, three classes of resonant mechanisms can be identified. The first two are the well-known synchronous, resonantly forced longitudinal standing waves, and subhormonic, parametrically excited transverse (cross) waves. These have

been studied by a number of investigators, notably in deep water. We rederive the governing equations and show good comparisons with the experimental data of Lin and Howard (1960). The third class is new and involves the simultaneous resonance of the synchronous longitudinal and subharmonic cross-waves and their internal interactions. In this case, temporal chaotic motions are found for a broad range of parameter values and initial conditions. These are studied by local bifurcation and stability analysis, direct numerical simulations, estimations of the Lyapunov exponents and power spectra, and examination of Poincare sections. To obtain a global criterion for widespread chaos, the method of resonance overlap (Chirikov, 1979) is adopted and found to be remarkably effective.

Henry M. Wu, "A Method for Eliminating Skew Introduced by Non-uniform Buffer Delay and Wire Lengths in Clock Distribution Trees", MIT Artificial Intelligence Laboratory Memo no. 1422, April, 1993.

Non-uniformities in buffer delays and wire lengths introduce skew in clock distribution trees. Previous techniques exist for eliminating skew introduced by each of these causes, not both. This method uses a pair of matched variable delay lines to eliminate skew caused both by differing buffer delays and wire lengths.

Henry M. Wu, "A Multiprocessor Architecture Using Modular Arithmetic for Very High Precision Computation", *1989 IEEE Symposium on Computer Architecture and Digital Signal Processing.* Also available as MIT Artificial Intelligence Laboratory Memo no. 1119, April, 1989.

We outline a multiprocessor architecture that uses modular arithmetic to implement numerical computation with 900 bits of intermediate precision. A proposed prototype, to be implemented with off-the-shelf parts, will perform high-precision arithmetic as fast as some workstations and mini-computers can perform IEEE double-precision arithmetic. We discuss how the structure of modular arithmetic conveniently maps into a simple, pipelined multiprocessor architecture. We present techniques we developed to overcome a few classical drawbacks of modular arithmetic. Our architecture is suitable to and essential for the study of chaotic dynamical systems.

K. Yip, "KAM: Automatic Planning and Interpretation of Numerical Experiments using Geometrical Methods", Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, August 1989.

KAM is a computer program that can automatically plan, monitor, and interpret numerical experiments with two degrees of freedom. The program has recently helped solve an open problem in hydrodynamics – the prediction of onset of chaos in a resonantly excited rectangular wave tank of finite depth. Unlike other approaches to qualitative reasoning about physical system dynamics, KAM embodies a significant amount of knowledge about nonlinear dynamics. KAM's ability to control numerical experiments arises from the fact that it not only produces pictures for us to see, but it also looks at (in its "mind's eye") the pictures that it draws to guide its own actions. By combining techniques with sophisticated dynamical invariants, KAM is able to exploit mathematical knowledge, represented in terms of a "grammar" that dictates consistency constraints on the structure of the phase space and parameter space. KAM is organized in three semantic levels: orbit recognition, phase-space searching, and parameter-space searching. Within each level spatial properties and relationships that are not explicitly represented in the initial representation are extracted by applying three operations – (1) aggregation, (2) partition, and (3) classification – iteratively.

F. Zhao, "Automatic Analysis and Synthesis of Controllers for Dynamical Systems Based on Phase-Space Knowledge", Ph.D. Thesis, Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, 1991. Also availible as MIT AITR 1385.

Zhao, F. and Thornton, R. "Automatic Design of a Maglev Controller in State Space." In *Proc. 31st IEEE Conf. on Decision and Control*, pp. 2562–1567, Tucson, Arizona, Dec. 1992.

We describe the automatic synthesis of a global nonlinear controller for stabilizing a magnetic levitation system. The synthesized control system can stabilize the maglev vehicle with large initial displacements from an equilibrium, and possesses a much larger operating region than the classical linear feedback design for the same system.

The controller is automatically synthesized by a suite of computational tools. This work demonstrates that the difficult control synthesis task can be automated, using programs that actively exploit knowledge of nonlinear dynamics and state space and combine powerful numerical and symbolic computations with spatial-reasoning techniques.

F. Zhao, "Automatic Design of a Maglev Controller in State Space." *AIM-1303*, MIT Artificial Intelligence Laboratory, December 1991.

We describe the automatic synthesis of a global nonlinear controller for stabilizing a magnetic levitation system. The synthesized control system can stabilize the maglev vehicle with large initial displacements from an equilibrium. and possesses a much larger operating region than the classical linear feedback design for the same system.

The controller is automatically synthesized by a suite of computational tools. This work demonstrates that the difficult control synthesis task can be automated, using programs that actively exploit knowledge of nonlinear dynamics and state space and combine powerful numerical and symbolic computations with spatial-reasoning techniques.

F. Zhao, "Machine Recognition as Representation and Search – A Survey", *International Journal of Pattern Recognition and Artificial Intelligence*, 5(5):715-747, December 1991.

Generality, representation, and control have been the central issues in machine recognition. Model-based recognition is the search for consistent matches of the model and image features. We present a comparative framework for the evaluation of different approaches, particularly those of Acronym, Raf, and Ikeuchi et al. The strengths and weaknesses of these approaches are discussed and compared and the remedies are suggested. Various tradeoffs made in the implementations are analyzed with respect to the systems' intended task-domains. The requirements for a versatile recognition system are motivated. Several directions for future research are pointed out.

F. Zhao, "Extracting and Representing Qualitative Behaviors of Complex Systems in Phase Spaces", *Proc. of the 12th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 1991. Also available as *MIT-AIM-1274*, MIT Artificial Intelligence Laboratory, March 1991.

33

We develop a *qualitative* method for understanding and representing phase space structures of complex systems. To demonstrate this method, a program called MAPS has been constructed that understands qualitatively different regions of a phase space and represents and extracts geometric shape information about these regions, using deep domain knowledge of dynamical system theory. Given a dynamical system specified as a system of governing equations, MAPS applies a successive sequence of operations to incrementally extract the qualitative information and generates a *complete, high level symbolic description* of the phase space structure, through a combination of numerical, combinatorial, and geometric computations and spatial reasoning techniques. The high level description is *sensible* to human beings and *manipulable* by other programs. We are currently applying the method to a difficult engineering design domain in which controllers for complex systems are to be automatically synthesized to achieve desired properties, based on the knowledge of the phase space "shapes" of the systems.

F. Zhao, "Phase Space Navigator: Towards Automating Control Synthesis in Phase Spaces for Nonlinear Control Systems", *Proc. of the 3rd IFAC International Workshop on Artificial Intelligence in Real Time Control*, Pergamon Press, 1991. Also available as *MIT-AIM-1286*, MIT Artificial Intelligence Laboratory, April 1991.

We develop a novel autonomous control synthesis strategy called Phase Space Navigator for the automatic synthesis of nonlinear control systems. The Phase Space Navigator generates global control laws by synthesizing flow shapes of dynamical systems and planning and navigating system trajectories in the phase spaces. Parsing phase spaces into trajectory flow pipes provide a way to efficiently reason about the phase space structures and search for global control paths. The strategy is particularly suitable for synthesizing high-performance control systems that do not lend themselves to traditional design and analysis techniques.

F. Zhao and L. Johnsson, "The Parallel Multipole Method On the Connection Machine", *SIAM Journal on Scientific and Statistical Computing*, 12(6):1420-1437, November 1991. Technical Report CS89-6, Thinking Machines Corporation, October, 1989. Also Technical Report

YALEU/DCS/TR-749, Dept. of Computer Science, Yale University.

This paper reports on a fast implementation of the three-dimensional non-adaptive Parallel Multipole Method (PMM) on the Connection Machine system model CM-2. The data interactions within the decomposition tree are modeled by a hierarchy of three dimensional grids forming a pyramid in which parent nodes have degree eight. The base of the pyramid is embedded in the Connection Machine as a three dimensional grid. The standard grid embedding feature is used. For 10 or more particles per processor the communication time is insignificant. The evaluation of the potential field for a system with 128k particles takes 5 seconds, and a million particle system about 3 minutes. The maximum number of particles that can be represented in 2G bytes of primary storage is $\sim$ 50 million. The execution rate of this implementation of the PMM is at about 1.7 Gflops/sec for a particle-processor-ratio of 10 or greater. A further speed improvement is possible by an improved use of the memory hierarchy associated with each floating-point unit in the system.

Feng Zhao, "Machine Recognition as Representation and Search", *MIT-AI-Memo-1189*, Artificial Intelligence Lab, MIT, December, 1989.

Generality, representation, and control have been the central issues in machine recognition. Model-based recognition is the search for consistent matches of the model and image features. We present a comparative framework for the evaluation of different approaches, particularly those of ACRONYM, RAF, and Ikeuchi et al. The strengths and weaknesses of these approaches are discussed and compared and the remedies are suggested. Various tradeoffs made in the implementations are analyzed with respect to the systems' intended task-domains. The requirements for a versatile recognition system are motivated. Several directions for future research are pointed out.