(2)

CHAMMP Review

This document has been approved for public release and sale; its distribution is unlimited.

MITRE

CHAMMP Review

H. Abarbanel
P. Collela
A. Despain
S. Koonin
C. Leith
H. Levine
G. MacDonald
N. Metroplis
W. Nierenberg
G. North
O. Rothaus
A. Semtner
J. Vesecky

January 1992

JSR-90-306

JASON
The MITRE Corporation
7525 Colshire Drive
McLean, Virginia 22102-3481
(703) 883-6997

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 31, 1991

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
CHAMMP Review

5. FUNDING NUMBERS
PR - 8503Z

6. AUTHOR(S)
G. MacDonald, et al

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The MITRE Corporation
JASON Program Office A10
7525 Colshire Drive
McLean, VA 22102

8. PERFORMING ORGANIZATION REPORT NUMBER
JSR-90-306

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of Energy
Washington, DC 20585

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
JSR-90-306

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution unlimited; open for public release.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

CHAMMP (Computer Hardware, Advanced Mathematics and Model Physics) is a new DOE program designed to move climate models from the current generation of supercomputers to massively parallel computers of the future. The general computing goal of CHAMMP is to provide a ten thousandfold increase in computing speed. Within the current climate modeling community, the primary motivation for increased speed is the desire to achieve much higher geographical resolution in the models, which would allow the "regional" predictions desired by policy makers. As planning for CHAMMP has evolved, issues other than those of spatial resolution have received increased attention. These issues include predictability, improvement of model performance by use of modern software engineering, the relationship of CHAMMP to other proposed modeling efforts, etc. This report provides an overview of these issues.

14. SUBJECT TERMS
computer hardware, advanced mathematics, model physics, climate system modeling program (csmp),

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
SAR

NSN 7540-01-280-5500

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

Contents

1 INTRODUCTION 1

2 THE PREDICTABILITY ISSUE 5

3 MODELS 13

4 HARDWARE AND SOFTWARE 21

4.1 Introduction 21
4.2 Hardware 21
4.3 Graphics and Visualization Systems 22
4.4 Operating and Database Management Systems 22
4.5 Software 23
4.6 Tools of Software Engineering 23
4.7 Software Framework 24
  4.7.1 Approaches 24
  4.7.2 Modern Compilers 25
  4.7.3 The Programming Problem 26
  4.7.4 A Framework 27
  4.7.5 CFW Substructure 28
  4.7.6 CFW Development 31
4.8 Conclusions 32

5 A DEDICATED COMPUTER FOR CHAMMP? 33

6 RELATION OF CHAMMP TO THE CLIMATE SYSTEM MODELING PROGRAM (CSMP) 39

7 MODEL INTERCOMPARISON STUDIES AND CHAMMP 43

7.1 Climate Model Intercomparison 43
7.2 Atmospheric Model Intercomparison Initiative 43
7.3 Model Intercomparison in the Context of Parallel Computing . 44
7.4 Conclusions and Recommendations 45

8 LONG-TERM SCIENCE BASE FOR CHAMMP 47

9 SUMMARY COMMENTS ON CHAMMP 49

A NEW COMPUTER ARCHITECTURE FOR CHAMMP? 51

- A.1 Introduction 51
- A.2 Range of Algorithms 51
- A.3 Discussion of Algorithms 52
- A.4 Strawman Architecture 54
- A.5 Software 57
- A.6 Conclusions 59

1 INTRODUCTION

CHAMMP (Computer Hardware, Advanced Mathematics and Model Physics) is a new DOE program designed to move climate models from the current generation of supercomputers to massively parallel computers of the future. The general computing goal of CHAMMP is to provide a ten thousandfold increase in computing speed. Within the current climate modeling community, the primary motivation for increased speed is the desire to achieve much higher geographical resolution in the models, which would allow the "regional" predictions desired by policy makers. As planning for CHAMMP has evolved, issues other than those of spatial resolution have received increased attention. These issues include predictability, improvement of model performance by use of modern software engineering, the relationship of CHAMMP to other proposed modeling efforts, etc. This report provides an overview of these issues.

In Section 2, we present a discussion of what is meant by predictability in the context of the time and length scales over which atmospheric parameters can be predicted. The discussion is influenced by the last three decades of results in nonlinear dynamics. These results show that, in low-dimensional chaotic systems, predictability is limited to very short time scales. The question arises as to whether the coupled ocean-atmosphere system exhibits chaotic characteristics on the climate-change time scale. Because the investigation of such questions will require model runs of great duration, improvements in computing capacity are clearly needed.

Proceeding on the assumption that issues of predictability can be satisfactorily resolved, we discuss the desirable characteristics of future climate models in Section 3. Modeling the oceans at the length scales needed to resolve eddies will also require much-improved computing capacity. The future modeling of the atmosphere will involve more accurate treatments of the physical processes, which places a further high burden on computational capabilities.

In Section 4 we discuss in some detail the types of hardware and software required to implement CHAMMP. Because several styles of supercomputers of increasing power are expected to appear on the market in the next few years, it is probably unwise for CHAMMP to invest in the development of computer hardware. On the other hand, it is likely that the software needed for CHAMMP will be developed only with CHAMMP funding. We illustrate the kinds of considerations that should go into software development by describing a framework for CHAMMP software. The framework is designed to expose and standardize the internal system interfaces so that a large group of individuals can contribute to the development of a single software system for CHAMMP over a number of years.

An important issue is the advisability of including provisions for a dedicated computer in the planning for CHAMMP. This question is considered in general in Section 4, as noted above, and in detail in Section 5. Primarily because of the present rapid development of computer hardware, we believe it unwise to invest in a dedicated computer in the near future. In the longer term, such a decision will require re-examination in terms of both available and prospective hardware and of the progress that has been made in achieving CHAMMP's goals.

A group of atmospheric scientists has put forward a proposal for earth scientists, principally climate modelers, to design a program based chiefly on university climate modeling, with the goal of satisfying the aims of policy makers. This program, the Climate Systems Modeling Program (CSMP), in our view complements the aim of CHAMMP (see Section 6). We foresee CHAMMP developing the framework for models, including the software, and CSMP carrying out the calculations of future climate change for policy makers. However, such collaboration requires a commitment to dedicated cooperation, because the two efforts can be viewed as competing for restricted funding for climate modeling.

The testing of particular models depends in part on intercomparison with alternative models. Requirements for such intercomparison in the era of parallel computing are considered in Section 7. Finally, Section 8 underlines the requirement of building up the talent pool of scientists who work on climate models.

2 THE PREDICTABILITY ISSUE

One of the assumptions underlying work on climate modeling is that, with sufficient computer power, we can indeed predict future climate. Only if this belief is accepted can one sensibly use climate models to understand the effects of policy alternatives. In this section, we would like to define predictability more precisely and urge that much more attention be paid to testing of existing (and future) models, with the goals of quantifying their predictive capabilities.

A widespread consensus, following the work of Lorenz, holds that weather is chaotic, with a loss of coherence (for neighboring initial conditions) of one to two weeks. The question for climate modeling is then the following: Do length and time scales exist over which the distribution of averaged variables (temperature, pressure ...) is *not* extremely sensitive to initial conditions? That is, given initial conditions $T_0(x), p_0(x)$ with a small error, the system will be predictable if there exists τ, L such that

$$\overline{T}(x,t) = \frac{1}{L\tau} \int_{t-\frac{\tau}{2}}^{t+\frac{\tau}{2}} dt' \int_{x-\frac{L}{2}}^{x+\frac{L}{2}} dx'\, T(x',t') \qquad (2-1)$$

has a variance that is no bigger than a bounded, t-independent function of the variance of the original error. In particular, as t gets larger, this error should not grow.

Under "ergodic" assumptions, this criterion can be translated into a statement about the correlation function on some type of dynamical attractor for the system (i.e., some dynamical equilibrium state), specifically,

$$\text{Variance } \overline{T} = \frac{1}{\tau^2} \int dt'\, dt'' \left[< T(t')T(t'') > - < T(t') >< T(t'') > \right], \quad (2-2)$$

and if we replace the ensemble average by a time average,

$$\text{Variance } \overline{T} = \frac{1}{\tau^2} \int dt'\, dt'' \int e^{i\omega(t'-t'')}\, S(\omega) d\omega, \qquad (2-3)$$

where S(ω) is the spectral density of the fluctuating field. Evaluating the integrals gives

$$\int (\frac{\sin \omega\tau}{\omega\tau})^2 \, S(\omega) \, d\omega. \qquad (2-4)$$

The first factor acts as a low-pass filter, weighting all parts of the spectrum below τ^{-1}. Not surprisingly, this weighting shows that achieving predictability that is free from weather-induced chaos requires our averaging scale to be sufficiently long in time to leave the dynamical system free of significant spectral power after filtering.

The need for long time scales has several immediate consequences for the climate-modeling problem. First, it is absolutely essential that the spectrum grow at small ω (as ω goes to zero) no faster than $\frac{1}{\omega}^{1-\epsilon}$. This rate seems to be operating for global temperature averages (real data), where $S(\omega) \sim \omega^{-.7}$ (see Figure 1), and is also true of the drastically truncated models developed by Lorenz (Lorenz -27). Preliminary experiments with the atmospheric components of a typical GCM also produce spectra that are well-behaved in this regard. This behavior lends some credibility to predictability for time scales that are long compared to the natural scales of the system; for the uncoupled atmosphere this is probably on the order of a month.

Note that it is quite reasonable to expect a tradeoff between spatial resolution and predictability. As we ask questions about local quantities (finite-domain spatial averages), structures that move slowly across the system will contribute to local variability but not to global fluctuations. Even if one finds predictability of scale τ for the global average, there will probably be a length scale $L(\tau)$ below which predictability fails. One possibility that is hard to discount is that length scales exist for which there is no predictability, even for extremely large τ; consequently, there may well be a scale below which there is no climate. Such scales could be investigated within the context of the uncoupled atmosphere model mentioned earlier.

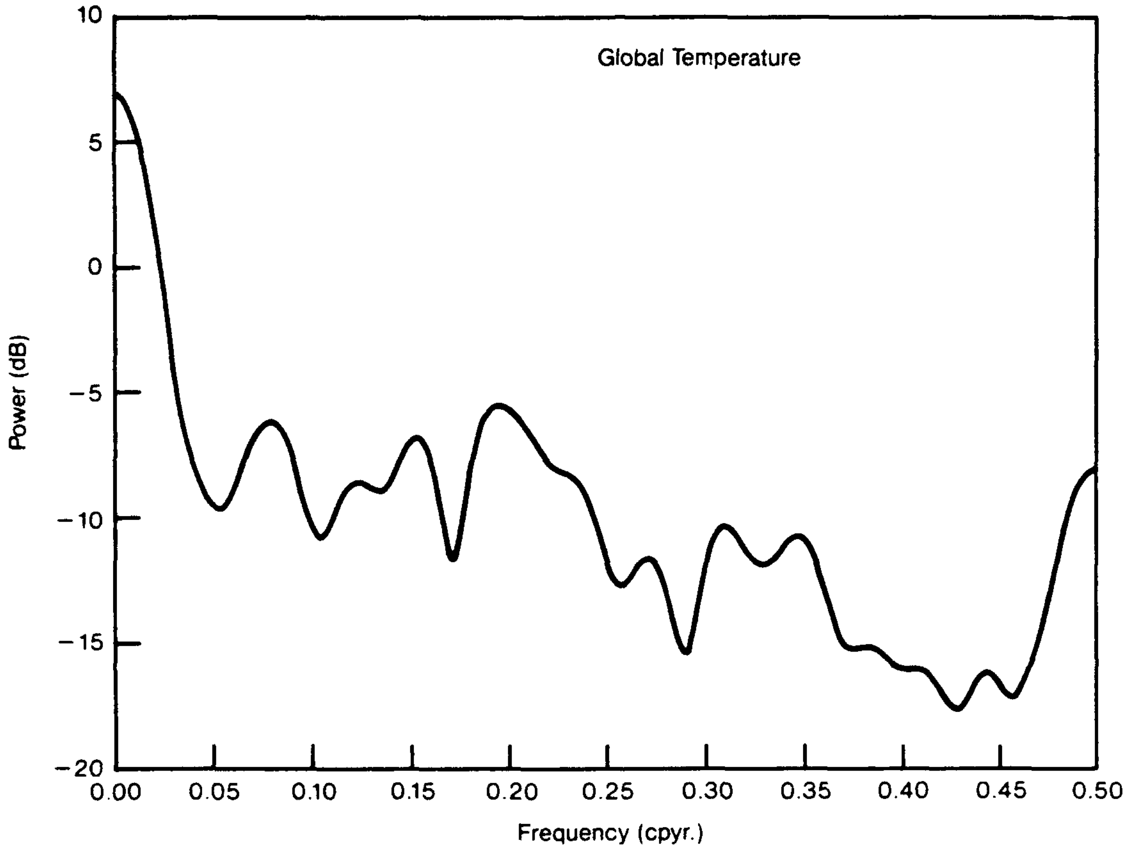

Figure 1. Power spectrum of global average temperature normalized to unit variance.

In order to test this type of predictability, one needs computer runs that are long compared to the time scales of typical physical processes. When one couples the ocean (and in particular the deep ocean) to the atmosphere, it becomes necessary to carry out runs of a minimum of several hundred and, more likely, several thousand years. Long runs of coupled ocean-atmosphere models should be an explicit part of the CHAMMP program; that is, one crucial use of the proposed 10^4 speedup in processing power should be devoted to the issue of natural variability on the ocean dynamic time scale. These long-term runs should be performed without adding a whole set of more complex physics "modules" to the model, which would then erode the capability of carrying out long runs.

In this sense, the problem of chaotic dynamics and the attendant loss of predictability is extremely serious when it comes to the oceans. If oceanic processes are truly chaotic on the scales of hundreds of years, our predictions of the climatic response to doubling CO_2 will be no more accurate than the total range of natural variability. Because this range appears from historical data to be quite large, natural variability may inflict a serious blow to the whole concept of prediction. The aforementioned global temperature data may imply that large variability does not occur, at least for global averages, over a 100-year period, but this must be tested in GCMs with deep oceans. In addition, we should stress once more that, even if the global average is salvaged from the record, local effects may be irretrievably noisy.

A more limited sense of predictability may slightly ameliorate this problem: If we cannot go to long enough time averages to eliminate chaotic dynamics, we might nevertheless be able to average over long enough times to reduce the chaos to a low-dimensional system. Techniques are emerging for short-range prediction based on the fact that low-dimensional systems can be reconstructed and integrated forward in time deterministically. Forward integration would allow prediction up to the time characterized by the inverse of the largest Lyapunov exponent. Thus, long-term capabilities would be lost, but short-term capabilities would be salvaged. Short-range prediction can be studied within GCMs and also can be modified by the spatial resolution of interest. Questions of exactly how best to carry out these studies are at the forefront of research in dynamical systems; hence, the climate modeling community must maintain contact with ongoing research in this area.

There is a general expectation in the climate modeling community that these "chaos" considerations will turn out to be annoying but manageable. Even if these considerations do prove manageable, one must quantify the relevant length and time scales; but let us for the moment assume that this has been done. There are still at least three other sources of error (leading to inaccurate predictions) that should be explicitly addressed within the program. The first source is related to the concept of sub-grid scale parameterization, which will inevitably remain part of any climate model. Even if one manages to increase the computing power and thereby decrease the grid size to encompass mesoscale physics, a whole set of phenomena remains that will never be resolved. The lack of resolution is not unique to climate models but is particularly critical in a problem for which the means of validating models is limited.

The sub-grid problem introduces a second predictability problem—that of the response of the deterministic dynamics to noise. This response is distinct from the ensemble average over different initial conditions, although a simple guess is that they may have similar consequences. The point is that one should not make simple guesses, but instead should perform numerical experiments to test how much "equation noise" matters. In simple dynamical systems, noise can cause transitions from one attractor to another, thereby producing a nonstationary probability distribution. Although such transitions are unlikely in a system with as many degrees of freedom as the climate, one should take the time for quantification and verification.

In Section 3 the requirement for reducing the grid size in order to increase regional predictability is discussed in detail. This issue of noise should be studied before one uses a large percentage of the increased computational power to decrease the grid size. If we discover that small scales are unpredictable with the relevant time scale averages (decades at a maximum), and if we discover a lack of sensitivity to sub-grid noise, then there is absolutely no reason to decrease the grid spacing. Having a small grid spacing can delude the casual observer into believing local-level predictions that are completely unreliable, and of course small grid spacing means more computing. In principle, one should use a grid spacing that is closer to the predictability limits, if such spacing is achievable via clever sub-grid parameterizations.

The last source of inaccurate predictions, incomplete physics, is the one which most practitioners feel is the most crucial. Whereas the sub-grid problem can in some sense be thought of as an AC driving of the climate model, systematic mistakes in the physics of various components in some sense constitute a DC problem, amplifying its effect over time.

For example, the preface to the ARM program document states that radiative forcing and feedbacks from clouds are not understood at the levels needed for reliable climate prediction. Although this statement is undoubtedly true, it is not quantitatively useful because it does not provide any idea of the magnitude of the effect. What one really needs to know is how a mistake in some parameterization alters the distribution of the averaged *predictable* results of the computation. Everyone seems to be operating under an assumption of "reasonable linear response," wherein a mistake by a factor of 2 leads to an error in the temperature shift (from some average value) by roughly the same factor.

For clouds, this hypothesis might be totally incorrect. Clouds affect heating by increasing albedo and by trapping infrared radiation. These two large effects almost cancel, as was recently discovered with some small residual studies using ERBE data. In any such situation, small mistakes can be disastrous. Unknown nonlinear feedbacks could also grossly alter the shape of the equilibrium distribution with small changes in the equations. There is even the outside possibility of going through a real bifurcation in the form of the attractor, though this is much more common in low-dimensional systems. Furthermore, although much effort will be focused on clouds, similar uncertainties are possible in parameterizations of sea ice and land vegetation, which also could drastically hinder predictability.

Again, no tools are currently available for studying the parameterization/prediction issue. In the absence of any theoretical framework, one will probably resort to numerical experiments to build up an intuition as to which parameterizations are critical and how accurately they should be known. The methodology is straightforward in principle, merely requiring changes in parameters and comparison of the distributions of predictable variables. The limitation is computational and cultural; computational in the sense that one needs the speedup of CHAMMP to perform these tests for any semi-realistic model (although studies of this sort in toy models should clearly be encouraged), and cultural in the sense that the atmospheric sciences community has shown no inclination to undertake these non-glamorous, systematic characterizations of existing models.

It is worthwhile noting that the issue of limitations in predictability due to incomplete physics might be most severe in regard to the understanding of extreme events. That is, even if we gain an understanding sufficient to find reliable distributions for predictable variables, the tails of these distributions will be much more uncertain. Clearly, we are very far away from being able to predict a significant change in, say, the number of severe hurricanes per year, and, as mentioned above, we should begin to study this issue within both toy models and existing GCMs.

In view of the gaps in our understanding of predictability, we recommend that numerical experiments of the sort outlined here be made a systematic part of the proposed program. Possible ideas about how to accomplish this include:

1. A joint workshop of CHAMMP atmospheric scientists and scientists in other disciplines (fluid mechanics, condensed-matter physics, mathematics of dynamical systems) on developing proper tools for analyzing predictability in spatially extended systems.
2. Access to the eventual model and its computational implementation by researchers who are specifically devoted to these issues and not to "production quality" output for policy discussions.
3. A stated goal of developing a consensus concerning the exact predictability limits of any "new generation" climate model and the exact response curves resulting from errors in the knowledge of different pieces of the climate system.

These issues are absolutely crucial if the CHAMMP program is to succeed in its stated goal of yielding real information quickly enough to have an impact on policy decisions. If the tests described are not carried out, it is likely that we will have a significantly more complex model running much faster on more expensive machines, but one which is no more effective at providing consensus answers than are the models of the current era.

3 MODELS

The principal goal of CHAMMP is to move climate models onto massively parallel computers, which are up to 10,000 times faster than present supercomputers, in order to enable prediction of climate change with regional accuracy. Two main components of a climate model, in terms of computational load, are the physical submodels for the atmosphere and ocean, which together constitute the physical foundation of the coupled climate system. (Augmentation of these components with atmospheric chemistry, ocean geochemistry, and global biology can easily double a model's computational needs.) An atmospheric model is computationally demanding because of the requirement for accuracy on length scales of tens of kilometers, and because assessing long-term climatic means and aspects of intrinsic variability requires these models to be integrated for hundreds of years. An ocean model is computationally demanding because of the need to resolve unstable energetic disturbances, which have length scales of order 10 km, and because of the requirement for some 1,000-year time integrations to approach equilibrium of the deep ocean. (Ocean models are essential in pursuing the CHAMMP objective, because the overall magnitude, spatial distribution, and time evolution of climate change depend critically on ocean dynamics.)

An approximate requirement for predictive atmospheric models within CHAMMP is that they have an effective grid spacing of 50 km or less. However, as discussed in Section 2, limitations on parameterization, noise, and chaotic behavior may make the 50-km goal unreachable. The comparable grid-spacing requirement for ocean models is one-half or even one-quarter that of the atmospheric models. Considering these spacing requirements, one can expect the ocean part of the coupled climate system to use at least half of all the available computer resources. One can also expect a coupled integration of 1,000 years to take many months of computation at a sustained execution rate exceeding that of existing (1990) computers by a factor of 1,000. Many integrations on various time scales can be envisioned in evaluating the ability of the models to meet CHAMMP objectives. Reproducing present and past climatic states in order to predict future states and their dependence on important physical parameters will indeed require an overall computing speedup of a factor of 10,000.

Improvements in climate modeling depend, in part, on much-enhanced capabilities for real (computational) time visualization. Such visualization is a first step in ensuring that the "human brain is in the loop." The concept of real-time visualization requires fast communication as well as advances in software development (see Section 4). This type of visualization can serve as a geodesic in establishing the basis for understanding the very complicated phenomena involved in such problems as ocean-atmosphere interaction and the formation and dissipation of clouds.

Atmosphere and ocean climate models appear to map rather naturally onto massively parallel computers because they can be based on horizontally local space-time physics. Horizontally based models allow complete avoidance of global sequential calculation, and only neighborhood information is needed to advance the model in time. The global domain can be decomposed into regional components distributed over computing nodes. Only border information requires communication by message-passing across the interfaces between regions. The ratio between communication and arithmetic, which is critical for parallel computing, scales favorably here as a surface-to-volume ratio for large models with many grid points per computing node. Message-passing serves also as an implicit synchronization mechanism and is not much more difficult to program than the conventional input-output data flow needed for models that are too large to fit in memory when run on single-processor computers.

A number of issues are relevant to the adaptation of atmospheric and oceanic models for massively parallel machines. These include:

1. Whether or not a common framework exists for the hydrodynamical, physical, and numerical treatments of the fluid in question.
2. How many tasks will be available for simultaneous execution on all processors almost all of the time.
3. What degree of imbalance exists among the various tasks of a problem at any given time.
4. What computational strategy will maximize overall scientific productivity.
5. How long will a particular formulation be effective in pursuit of CHAMMP goals.

It is useful to discuss these issues in turn, as they apply to the oceanic and atmospheric components of a coupled climate model.

Ocean modeling has always been severely hampered by computational constraints. A number of approaches, such as the use of limited domains and limited physics, have been used in the past to make problems tractable. Within the past few years, runs of ocean models with relatively complete physics and global-scale coverage have been completed, but they strained the capacity of the best available supercomputers. Most of the models in use have a common formulation (Navier-Stokes equations with active thermohaline processes are solved by finite-difference techniques) and derive mainly from an original ocean model developed at Princeton's Geophysical Fluid Dynamics Laboratory. The original model has been refined and improved over the years by numerous investigators throughout the world. Thus, there is a large measure of commonality among existing models, and many problems that were previously intractable, in terms of grid resolution and time integration, await solution within the same modeling framework.

The existence of a common framework for attacking many oceanographic problems should not preclude the development of innovative numerical techniques, such as finite-element and adaptive-grid methods. New techniques might provide computational alternatives in ocean modeling, but the need for enhancement of computer power will persist, regardless of methodology.

In assessing how many tasks are available in an ocean-modeling context, we keep in mind that massively parallel machines may be expected to have from 50,000 to 1,000,000 processors. Efficient utilization of all processors requires either (a) that the tasks be well-balanced and occur in integral multiples of the number of processors, or (b) that the number of tasks greatly outnumbers the processors, without a stringent need for balanced tasks. Global ocean models with horizontal grid spacing of one-quarter degree have a million grid points at each level. Higher-resolution models routinely would have several million tasks in the horizontal. It thus appears that, if organized around simultaneous processing in the horizontal, ocean models can attain condition (b). The degree of task imbalance in an ocean model should be relatively low, because many sources of potential imbalance, such as surface fluxes, near-surface mixing, and surface height, are minimized by a horizontal treatment. Remaining aspects of imbalance, due to irregular lateral boundaries and possible convective overturning, can be handled with masking, at a cost of some unnecessary arithmetic.

The foregoing describes how in ocean modeling, which has a fairly common framework for attacking climate-related problems, a large number of relatively well-balanced tasks can be treated through a horizontal orientation. Because much investigation is needed in high-resolution simulation of long-term ocean behavior, an ocean model adapted to massively parallel architectures should have considerable longevity. Refinements in physical processes, which typically involve the treatment of vertical mixing processes, can be amalgamated into an existing model framework without enormous effort and without adversely affecting the parallel aspects of the computation.

The overall situation for the parallelization of atmospheric models is somewhat more complicated than for ocean models. Many formulations of atmospheric models exist for climate studies. Numerical treatments of hydrodynamics divide into the two options of finite differencing versus spectral (spherical harmonic) decomposition. Spectral models are currently popular because of their efficiency at moderate horizontal resolution, but this advantage would disappear at a very high resolution of about one-quarter degree grid spacing. Because spectral tranforms may not run in parallel as effectively as finite-difference methods on some computer architectures, the two methods may be computationally equivalent at grid resolutions that are important to climate. The end result is that a common framework is less obvious for atmospheric modeling than for ocean modeling. Serious consideration must be given to developing models based on both finite-difference and spectral methodologies.

The physical processes in atmospheric models dealing with radiation, clouds, convection, and the planetary boundary layer have been treated in many ways by different modeling groups. Such processes are almost always carried out in physical space coordinates; however, adequate program modularity is necessary to allow for evolution in the diversified treatment of many physical processes.

The number of tasks in an atmospheric model that has an effective grid spacing of one-half degree (about 50 km) in the horizontal is about a quarter million per level. Hydrodynamic algorithms, such as advection by the large-scale flow, are independent at each level. Millions of tasks are available for these algorithms from simultaneous processing of all levels, and situation (b) exists as described above (many more tasks than processors). A level-by-level treatment of advection would also be possible following situation (a), with the tasks being an integral multiple of the number of processors, given that advection is a well-balanced operation. However, physical algorithms with vertical interdependency, such as radiation and convection, are often poorly balanced in atmospheric models. These types of processes will require considerable effort in model development if they are to be highly parallel as well as relatively modular. One possible approach is to compute certain hydrodynamical and physical effects simultaneously. Such an approach would require relatively sophisticated, multiple-instruction, multiple-data machine architectures. Alternatively, machines with fewer but faster processors would have a natural advantage over those with more but slower processors. In general, a non-trivial amount of masking and unnecessary arithmetic may be required to make architectures adequately parallel.

Regarding the issue of longevity of an atmospheric model for climate studies, finite-difference may emerge as the clear choice for the treatment of hydrodynamics over several years, allowing concentration on only one formulation. Continuing experimentation with physical parameterizations will undoubtedly be required to attain the goal of accurate regional modeling. Thus, a continuing need exists for modularization of model physics within a core hydrodynamics model.

In summary, the CHAMMP goal of modeling the coupled ocean-atmosphere system on massively parallel computers is reasonable and feasible. A major portion of the computations will deal with the oceanic part of the system. Exploiting computer upgrades will enhance resolution and extend integrations in a common framework that is quite amenable to massively parallel architectures. For the atmospheric portion of the computations, considerable attention must be given to making parallel certain processes with inherent task imbalance and to preserving modularity for increasingly accurate treatments of physical processes. When coupled together, such models may substantially improve our ability to simulate and predict global climate on a regional basis, provided the predictability issue (raised in Section 2) can be resolved.

4 HARDWARE AND SOFTWARE

4.1 Introduction

The goal of CHAMMP is to create a new generation of climate models that can utilize new developments in computer technology such as algorithms, software environments, operating, database and visualization systems, as well as supercomputer hardware. It is very clear, as discussed in Sections 2 and 3, that CHAMMP must concentrate on the question of the feasibility of climate model calculations and algorithm development. Assuming that the feasibility of such calculations is proven, large improvements in both computer hardware and software technology will be needed to achieve CHAMMP's goals.

4.2 Hardware

It is conceivable that CHAMMP could sponsor the development of a computer especially designed to calculate climate models efficiently and quickly. Such a computer might be justified if a single algorithm (or narrow class of algorithms) for calculating climates were agreed upon and if execution of the algorithm would be continuously needed for several years. However, these conditions do not currently apply to CHAMMP. Nevertheless, we present a new machine architecture (see Appendix A) that could be considered for development by CHAMMP, if such an approach were later deemed desirable.

Another hardware issue is whether the CHAMMP program should buy a single supercomputer for the CHAMMP community or several machines for individual CHAMMP research groups to share. Due to the current rapid evolution of high-performance computer systems, it is unlikely that one such machine would prove optimal for CHAMMP. It seems better to design the CHAMMP models, algorithms, and software to be portable, so that advantage can be taken of several different types of supercomputers, both current and future models. Because several styles of supercomputers of continuously increasing power are expected to appear on the market, there may be no advantage in having a single, dedicated CHAMMP machine. This issue is more fully discussed in Section 5.

4.3 Graphics and Visualization Systems

Hardware and software systems that display high-resolution, three-dimensional, color images are rapidly being commercially developed. Such systems can be purchased off the shelf, as needed, by the CHAMMP program. Interfacing these systems with the algorithms employed in climate models will require some special software development.

4.4 Operating and Database Management Systems

The only reasonable choice for an operating system is one of the UNIX variants. For database management, a mature, well-supported commercial system (such as ORACLE) that is compatible with UNIX and its file system should be chosen after a careful evaluation of compatibility across the spectrum of machines that will be used for CHAMMP.

4.5 Software

Although hardware systems that are faster and more parallel will undoubtedly be developed independent of CHAMMP, it is likely that the software needed for CHAMMP will be developed only with CHAMMP funding. It also seems to be the consensus of the CHAMMP investigators that most of the needed performance improvements will come not from hardware developments, but from a combination of algorithm and software developments. In fact, it appears that most CHAMMP funding will go towards the construction of a model composed of 100,000 to 500,000 lines of code (if written in FORTRAN). Because different groups will try various approaches, and because algorithms will be tried and discarded, it is likely that CHAMMP will sponsor the writing of more than 1,000,000 lines of code.

4.6 Tools of Software Engineering

The task of writing the equivalent of 1,000,000 lines of FORTRAN code can be approached in two ways:

1. If a single model code is needed as soon as possible and cost is not a consideration, then the best general-purpose tools of software engineering that are available should be adopted, and the work should be systematically organized.
2. If the code is to be developed by a large, diverse group of scientists, the life span of the software is to be long, the users will be numerous, and code development will require much experimentation, then the general-purpose tools of software engineering will be inadequate. A better approach is to develop a special framework and set of tools designed

just for the climate modeling task. This approach is obviously best for CHAMMP and is discussed next.

4.7 Software Framework

The primary motivation for constructing a special software development framework is efficiency in developing a computer program to model climate. Such a framework can expose and standardize internal system interfaces so that, over a number of years, a large group of diverse individuals can cooperatively develop a single software system that can be used by all the CHAMMP community.

4.7.1 Approaches

A CHAMMP software framework could be approached in several ways:

1. It could be built around a general-purpose, relational database management system (DBMS), such as ORACLE.
2. An object-oriented language such as C++ could be adopted, and a framework could be built up out of objects.
3. A manual framework could be built up around existing software engineering tools, such as "make" and "SCCS," which are available on UNIX systems.
4. A program-generator system could be created to compile from high-level specifications and adapted to climate model programs expressed

in FORTRAN (or C). This option, which we recommend, is discussed further below. Such a software system could be called:

(a) The CHAMMP Program Generator System,
(b) The CHAMMP Climate Model Compiler, or
(c) The CHAMMP Software Development Framework.

This program generator could also be thought of as a specialized language compiler specifically designed to optimally express climate models. More importantly, a framework would allow modular development and provide specialized computer programs on demand. We will adopt for this software the name "CHAMMP Framework" (CFW).

4.7.2 Modern Compilers

Today, complex compilers are structured as a series of analysis and transformation modules, with several internal interfaces comprising intermediate formal languages. Sometimes such intermediate languages (ordinarily one at most) are disclosed to the world outside of the compiler development group. One of the best-known examples of this is the "P-Code" language used in almost all Pascal compilers. The great advantage of using a common language in compilers is that they are relatively easy to re-target to various host machines by re-writing only one or a few of the compiler modules.

A typical modern compiler for a high-level language, as viewed by its developers (but generally hidden from the users), is a series of stages, in each of which fairly simple transformations and annotations of the input specifications (high-level program) occur. Such a description is critical for the developers to control the complex task of generating the compiler and to allow assignment of subtasks to team members.

4.7.3 The Programming Problem

The CHAMMP research program has a much more difficult overall programming task than simply building a high-level language compiler. The research program leadership should thus insist that the CHAMMP community (all research groups funded by the CHAMMP research program) first agree upon a common framework for software development, with "publicly" identified internal modules and interfaces. Each member of the CHAMMP program can then benefit from modules produced by others. When superior modules are produced, all others can adopt them immediately.

If such a high level of interaction is to be achieved, an extraordinary effort will be needed to define and develop the framework. To this end, we recommend that a special framework group be established. This work would be best carried out by an independent contractor with no particular climate model ax to grind, but with a highly competent staff of computer scientists skilled in modern computer language and compiler development. The framework group should be advised by a policy committee with representatives from the CHAMMP community.

We also strongly recommend that a formal language grammar be completely specified in both syntax and semantics at each module interface. Constructing a new language grammar may seem an unnecessary encumbrance to many programmers, but will prove to be critical for coordinating the large, diverse group development effort of CHAMMP.

Either attributed grammars (e.g., definite clause grammar) or denotational semantic specification will suffice if done correctly. Any one of several languages can be used to write the translation modules: PROLOG offers a simple method for semantic specification through its support for definite clause grammars; LISP is popular with some compiler writers; ADA would be robust and brings along a good software engineering environment; and the "C" language is universally known and runs on many machines.

4.7.4 A Framework

To illustrate the type of framework that we are recommending, we present a "strawman" framework; i.e., an example to provoke discussion and later be replaced by a more carefully planned and detailed framework. This is the CFW.

Viewed globally, CFW is a series of language translation (compiler) blocks with a high-level language program, benchmark data, and target execution environment visible at each level. The top-level language is specialized for describing various high-level climate models. Its terms, relationships, operations, and state transitions fall in the realm of the climate/atmosphere/ocean physical scientists and not those of programmers, numerical analysts, or computer scientists. At the lowest level, the output of the framework is a program generated in FORTRAN (or C) language, with operating system annotations and/or operating system command scripts to schedule parallel activity. Development of the CFW could encompass the FORTRAN-to-machine-code levels, but this is not recommended because limited CHAMMP program money can be better spent on the specialized higher levels that no one outside CHAMMP will produce.

The CFW will span several times as many levels of abstraction as a modern compiler. A modern compiler typically spans 6 to 10 levels—CFW will need 20 to 40 levels. Beginning at the FORTRAN level and progressing upward a few levels, there would be modules to schedule parallel instructions for MIMD and SIMD architectures as advocated by R. Stevens at ANL and others. At higher levels one would expect to see a language level similar in abstraction to FIDIL. Proceeding upward a further dozen or so levels, one would expect to see a translation module (and corresponding languages), such as advocated by Balahan et al., for deriving discrete approximation schemes. Higher levels of CFW might include some of the features developed by K. Wilson in his Gibbs Project as they apply to climate modeling.

Each translation block in CFW should be transparent, modular, and publically accessible. It should be easy for any member of the CHAMMP community to rewrite any module (in any computer language of his choosing) to improve its performance or enhance its functionality. Both the input and output of each module should be a UNIX file (or possibly a DBMS relation), as should each program module. All modules should be centrally archived and accessible to all.

A central authority should determine a directory structure, naming conventions, a version control scheme, and an archive method for the CFW. Each CHAMMP site could have its own working version of CFW, and send new modules to the CFW authority for validation, distribution, and archiving.

4.7.5 CFW Substructure

The substructure of each level of CFW should allow the possible inclusion of various analysis modules, simulation drivers, visualization displays, and tools to aid debugging and human analysis. Each level will need an interface to the file (and/or DBMS) system of CHAMMP. *Each* level of CFW should include a program translation module, a simulation module, a visualization module, a data (problem) translation module, a performance analysis module, a knowledge base of heuristics, and environment translation module. These modules are described below.

4.7.5.1 Program Translator

It is the function of the program translator to check and accept the specification for a program and translate this into a more detailed specification. This specification is itself a high-level language program which is translated to a lower-level form. Generally, a large number of semantically correct, lower-level translations are possible. Each will be correct in terms of producing a program that, when executed, gives a valid result. However, some resulting programs will be more efficient than others (i.e., will run faster). Thus, the program translator must consult heuristic rules, procedures, and prior-level analysis of benchmark results to select efficient implementations from its available choices. These rules, etc., are packaged in the heuristic module to be discussed below.

4.7.5.2 Simulator

The function of the simulator (actually a specialized program interpreter) is to simulate the execution of the program and its benchmark data in order to measure the program's efficiency. This information is fed back to the same-level analysis module, so that estimates, etc., can be updated, and also to the next-lower-level analysis module, to aid in the next cycle of design decisions. The simulator also feeds raw results to the visualization module, where they are displayed for viewing.

4.7.5.3 Visualization

The visualization module processes the raw output of the simulator and generates graphics and other presentations of the effectiveness and accuracy of generated (compiled) programs relative to the benchmark problem data.

4.7.5.4 Data Translation

The data translator module accepts both benchmark data and a compiled program and, while consulting its database, it expands the problem data into a form suitable for the next-level program to execute. For example, a climate model at a high level might begin with a formula that expresses temperature as a function of altitude. A lower-level program may need a discrete (tabular) representation of the same temperature profile.

4.7.5.5 Performance Analysis

The function of the performance analysis module is to accept the raw results of program simulation and input data, and to reduce this information to appropriate performance estimates, so that the heuristics module can provide good design choices to the program translator module.

4.7.5.6 Heuristics

The heuristic module has access to an extensive database that includes rules supplied by experts, results of past simulations, and performance estimation functions. It receives an analysis of benchmark executions from both above and below its level. Its function is to decide between sets of design choices provided by the program translator module, based upon the heuristic rules, its historical record, and current analysis data.

4.7.5.7 Execution Environment Translation

The specification of the execution environment, like that of the program translator, must also be translated from very high-level abstract terms into low-level operating system scripts and annotations for insertion into the FORTRAN code. For example, at the top level, the specification might only identify a network of connection machines and specify a priority for execution. The execution environment translator module (and its database) communicates with both the program translator and analysis modules to produce a more detailed execution environment specification.

4.7.6 CFW Development

Developing all the levels of the CFW system is a very large undertaking, especially at the upper levels, where many unknown problems are involved. The greatest payoff will come from careful development of the lowest levels, where manual programming is massive, tedious, and error prone. As the lower levels become mature, and experience is gained, it will be easier to add the more difficult upper levels—provided the discipline of the formal framework is maintained.

The first tasks in setting up the framework are selecting a site and choosing an initial computer system, programming language, formal language specification methodology, and database management system (if used).

The CHAMMP community should also quickly develop a consensus on the requirements for the first high-level language, which will be translated into annotated FORTRAN (or C). This language should be on a level only slightly higher than FORTRAN, and well below a language like FIDIL, but should provide some relief from the details of programming required by FORTRAN. It is important that this be only a simple extension. For example, features of such a language might be built-in vector data types and operations. Benchmark data for programs in this language will be needed, as will a scheme for specifying the execution environment. Valuable experience can be gained by quickly developing all the modules for a prototype of the lowest-level translation block.

4.8 Conclusions

It should be possible for CHAMMP to meet its goal of developing a computer model that can calculate likely patterns of future climate change. To accomplish this, CHAMMP should establish a software development framework for use by its research community. Such a framework has the potential of greatly speeding up the software development process and enhancing the value of the software that will be developed under CHAMMP.

5 A DEDICATED COMPUTER FOR CHAMMP?

The main objectives of the CHAMMP program include transporting existing climate models to massively parallel computers, building new climate models with algorithms tuned to this new generation of machines, and developing new climate models that incorporate additional physics and higher spatial resolution. The latter objective will become feasible when machines have megafloppage of order 10,000 or more times that of current Cray YMP class computers.

In the first two to four years of CHAMMP, the first two objectives will be carried out on serial vector machines with multiple processors such as the eight-processor Cray-2 at LLNL, which is being purchased partially with CHAMMP funds. Other machines include the existing Intel Touchstone Gamma machine, now at ORNL, and the TC-2000 from BBN, now installed at LLNL. In this immediate time frame we cannot see a requirement for a machine dedicated to CHAMMP. However, any dedicated machine would be of the massively parallel type for the following reasons.

1. Full-time code development or numerous production runs would presumably require a dedicated, improved-model machine. The only machines available for these activities as of 1990, and for the next few years, are serial, slightly parallel devices such as the multiprocessor Crays. Such machines are available through NERSC and other locations for the development needs of the program. The work of Chervin and Semptner on parallel ocean modeling demonstrates this clearly.
2. An immediate investment in a large, dedicated, massively parallel machine would be wasted in terms of the long-range goal of the project. Any machines we purchase now will have inadequate computing power

to provide factors of 10^4 or more in floating-point throughput for CHAMMP and would thus require replacement as machines in the desired class become available.

3. The financial requirements of the infrastructure (support staff, buildings, etc.) required for a dedicated facility would divert resources from the scientists working on demonstrating the possibility of using such a machine for the goals of CHAMMP. In other words, the cost of such a dedicated facility cannot be justified at the outset of CHAMMP if funding it destroys the possibility of proving it is required.

The most cost-effective approach is using the available computational horsepower—both serial and parallel—until the next generation of machines is available. Purchasing portions of NERSC machines to assure CHAMMP time on the suite of computers available to ER generally and CHAMMP in particular seems sound for the near future. This policy should be reexamined each year to ensure that the investment does not divert resources from pursuit of the central scientific issues of CHAMMP. Also, as we shall argue in a moment, the time will come sooner rather than later when a dedicated CHAMMP machine does make sense—too much investment now in serial machines would be unwise.

In addition to supporting high-performance serial computers, such as the Crays at NERSC, some funds should be put into massively parallel machines as well. DOE could arrange for software development on these machines (general operating software and specific scientific libraries and algorithms) that directly supports climate model development and exploration. The existing NSF Supercomputer Centers serve as generic examples of this mode of investment. One can envision the partial purchase of a massively parallel machine at one of these Centers and the subsequent support of appropriate scientific and system software on this machine, as discussed in Section 4. The advantage of this kind of investment would be that the required model development would occur and, because these Centers deliver cycles to users who are widely distributed geographically, one would gain real-world experience in making these massively parallel computing capabilities available to users.

After a period of four or five years in this mode, a quite clear picture should emerge of what hardware will be available for CHAMMP goals and of what software will have been developed and will become available for CHAMMP modeling purposes. At this point, a dedicated CHAMMP computer may become quite attractive.

If massively parallel machines show real promise of providing the 10^4 times or more floating-point throughput desired by CHAMMP, then an upgrade path to the hardware should be clear. A "lend-lease" plan with the manufacturer will be required so that a scalable upgrade path is achieved with existing resources. If an eventual purchase is deemed inappropriate, then the lease arrangement can terminate naturally. One should not be tied to the machine of 1995 when the machine of 1998 may be the one CHAMMP wants.

In the same four-to-five year time period, it should become clear from the experimentation with machines at DOE labs, and perhaps at NSF-like Centers, whether or not the software CHAMMP requires will also be available. Both system software to allow ease of code development and scientific software to allow efficient computation are essential.

For a target date on which to decide the attractiveness of purchasing a dedicated CHAMMP machine, let us choose four years from 1 October 1990 — adequate time for analyzing the availability of these hardware and software items. Also high on the list of requirements is a clear indication that the community of climate modelers and climate model analysts will have become large enough to use a machine dedicated to CHAMMP productively. In other words, the machines must be available, the software must be available, and the community must be large enough and ready to work on massively parallel machines for scientific production (see Section 8).

If these requirements are met, two additional arguments can be made for a dedicated CHAMMP machine:

1. PRODUCTIVITY: With a sound base of software, hardware, and personnel, the work of modeling in CHAMMP can proceed rapidly by using machines solely dedicated to long and/or high-resolution model runs on massively parallel machines. The piecemeal, experimental state will have passed and production can get underway.
2. APPEARANCE: If the scientific base is sound, obtaining funding for climate prediction studies will be assisted by the presence of a dedicated machine and the supporting scientific and systems staff that it will require. The clearly identifiable effort of CHAMMP will give the progam political substance, which it may well require in competing for the large but finite resources devoted to climate change studies. Eventually, a distributed computational effort will be vulnerable to attrition by bits and pieces in the budget process; use of a demonstrated facility as a central feature of the effort will diminish such vulnerability.

If the requirements indicated are not met in the fall of 1994, as determined by the DOE program manager's review process, then we propose that the question be revisited on an annual basis, with any decision made in FY(N) to be realistically funded in FY(N+2). The dollars required depend on several variables not knowable in 1990: cost of the machine from the commercial manufacturer, including software and support; desired staff and facility size; and networking requirements and availability of high-bandwidth national networks. A dedicated machine could be placed at a DOE laboratory or NSF or other existing center to take advantage of present infrastructures, or, for political or other purposes, the CHAMMP Center could be made a "standalone."

6 RELATION OF CHAMMP TO THE CLIMATE SYSTEM MODELING PROGRAM (CSMP)

The Universal Corporation for Atmospheric Research (UCAR) has proposed a Climate System Modeling Program (CSMP) to accelerate progress toward reliable predictions of global and regional climate changes. If formed, CSMP will consist of a team of scientists who will conduct research, data analysis, and coordination activities in collaboration with existing research organizations. The goal of CSMP is the provision of an integrating mechanism to ensure that the pieces of the earth system climate model come together to meet the anticipated needs of society, particularly policy makers. The first-year planning budget is proposed at $724,000, with later years requiring support at the annual level of $24 million to $95 million. Initially, the support is expected to come from the federal agencies that make up the Committee on Earth and Environmental Sciences (CEES). In later years, an attempt will be made to secure funding (as much as one-third) from non-governmental sources. The proposal grew out of two workshops attended by scientists active in earth science modeling. Francis Bretherton, of the University of Wisconsin, has been designated as the Scientific Director of CSMP.

CSMP and CHAMMP can be viewed as both complementary and competitive efforts. The competitive element arises from budgetary considerations. Only a limited amount of federal funds will be available to support climate change modeling, though the precise amount is not currently known. Both CHAMMP and CSMP will have to compete with several other modeling efforts, including NOAA's Geophysical Fluid Dynamics Laboratory, NASA's work at both the Goddard Institute for Space Studies and the Goddard Space

Flight Center, NCAR's activities, and efforts at a number of universities, including UCLA, Florida State, and the Universities of Illinois and Maryland. In addition, a number of programs are under development, including NOAA's and NSF's Dynamic Extended Range Forecasting Program, NSF's Integrated Climate Modeling Analysis and Prediction Program, and NOAA's Climate Modeling and Analytical Centers. Such competition is unavoidable and can be beneficial, provided a serious effort is maintained to encourage collaboration among participants and to avoid unnecessary duplication.

Although the planning for CSMP is not as far along as that for CHAMMP, the programs could clearly complement each other. The goals for CHAMMP include the investigation of the predictability of climate, the study of massively parallel machines and their applicability to climate models, the analysis of new software developments to support climate models, and the construction of a new-generation climate model. The main effort in CSMP would be to run models so as to assist the policy maker, while drawing on available data in the various applicable earth sciences. CSMP is aimed particularly at university scientists, while CHAMMP is more broadly aimed at universities, national laboratories, and industry. CSMP will seek broad-based funding support, while CHAMMP has a single sponsor, DOE. CSMP will be a centered activity with network links to university participants, while CHAMMP, at least initially, will support research at a number of institutions.

The large number of modeling activities noted above makes it imperative that at least one of the programs focus significant effort on fundamental modeling issues: What is meant by prediction in highly nonlinear systems? What parameters can be predicted and over what time and length scales? How can models be designed to test the limits, if any, to predictability? CHAMMP can provide a prime focus for work on these questions. Another fundamental issue is how the best hardware can be applied to modeling efforts. Future developments in computers will be toward massively parallel machines. Again, the results from CHAMMP can provide the guidance to CSMP and other modeling efforts as to desirable avenues in hardware and software selection. Software engineering is progressing slowly in the areas of validation and automated programming. Application of these developments, together with the use of accepted good practices in software engineering, can aid the overall modeling effort. CHAMMP can provide leadership in these developing areas.

7 MODEL INTERCOMPARISON STUDIES AND CHAMMP

7.1 Climate Model Intercomparison

The need for climate model intercomparison is clear. Model intercomparisons are widely quoted and have revealed areas of significant disagreement among model results. This disagreement has focused attention on specific physics that needs improvement. For example, recent work by Cess et al. revealed differences among prominent climate models in cloud response to changes in boundary conditions. The Program for Climate Model Diagnosis and Intercomparison (PCMDI) at LLNL has also produced useful intercomparison results. For example, Grotch (1989) compared the results of five different GCMs in terms of changes in surface temperature and precipitation resulting from a doubling of CO_2. He noted that although the five GCMs all produced similar results when averaged over large scales, they were significantly different on the scale of subcontinental regions. Work, such as the intercomparisons under PCMDI at LLNL, provides DOE a useful entree into the climate modeling community in addition to its financial sponsorship of modeling efforts.

7.2 Atmospheric Model Intercomparison Initiative

An initiative proposed by W. Lawrence Gates of LLNL concerns the intercomparison of atmospheric models using a standard set of boundary conditions. The conditions are those recommended by WGNE/WCRP, namely the observed monthly sea-surface temperature and sea-ice distributions over the years 1982 to 1989. Climate modeling groups participating in the comparison would produce a standard time series of output variables and compute a standard set of statistics. Examples of output would be the distributions of the monthly means of surface temperature and wind. The intercomparisons would be designed to isolate model features and parameterizations of particular interest. For example, the atmospheric model intercomparison isolates the atmosphere by using measured ocean boundary conditions. Each group would run its own model on LLNL computers under standard conditions to produce a required set of outputs. This initiative appears to be a useful one by a group with a good track record; however, it does not address issues related to parallel computing.

7.3 Model Intercomparison in the Context of Parallel Computing

For many purposes, intercomparisons need not be run on parallel machines. However, as modelers use parallel machines to develop more complex models and to attack problems on longer time scales, it will become impractical to run model intercomparisons on anything but parallel machines. Parallel computing introduces another set of issues into the intercomparison process, significantly increasing the effort required to make intercomparisons. These new issues are in addition to the existing issues involving geophysics and numerical algorithms. The new issues relevant to parallel computing involve questions such as the following: Do parallel and sequential versions of the same algorithm yield the same answers? Which scheme of adapting a particular model subsystem or algorithm to a parallel machine is the most accurate and efficient? How robust and portable are model codes in terms of running on different parallel machines?

Intercomparing climate models using parallel computing will require the use of somewhat standardized hardware. Cray XMP class machines could be used, but it would be very difficult for modelers using machines with different parallel architectures to intercompare codes. For example, codes running on Alliant machines could not be run on Crays without great effort. Given that different modeling groups use the same or compatible parallel machines, we think they would be willing to work on machines at a comparison site, such as LLNL, using wideband data links to compatible machines. LLNL has experience with such data links to outside users.

As parallel computing becomes standard with small numbers of parallel processors (Cray XMP class), one needs to look forward to the issue of massively parallel machines, such as the large-scale parallel machine proposed by Intel. As this stage is approached, one may benefit from NASA's experience with the massively parallel processor (MPP) at Goddard Space Flight Center. In terms of intercomparisons, a massively parallel machine devoted primarily to intercomparisons must be considered. The PCMDI group at LLNL would be a good candidate for such a dedicated machine.

7.4 Conclusions and Recommendations

We summarize our conclusions and recommendations as follows:

1. Model intercomparisons are difficult, especially when parallel computing is involved. However, intercomparisons are very useful both in identifying model errors (and hence features that need improvement) and in understanding how much credibility GCM results deserve.
2. Intercomparisons can address a number of issues regarding the use of parallel computing in climate modeling, to wit:

(a) Are certain parallel implementations, architectures, or machines particularly well suited to climate models?

(b) Does a given model or subsystem run more effectively with alternative shemes for implementing parallel operations?

(c) Can robust and transportable codes for GCMs be achieved on parallel machines?

(d) Do parallel versions of a given model yield equivalent results for both sequential and parallel versions?

8 LONG-TERM SCIENCE BASE FOR CHAMMP

CHAMMP is an ambitious program that seeks to apply developing capabilities in computing and numerical analysis to the problem of climate modeling. It will be of no use to compute inadequate models more rapidly, nor to devise models that are beyond foreseeable computational capacity. To succeed, the effort must be balanced.

As CHAMMP begins, balance is being achieved by recruiting established experts in the relevant fields of computer science, numerical analysis, and climate modeling. As is natural and was to be expected, these scientists are being drawn largely from the National Laboratories and NCAR, the groups with whom DOE has always had close contacts. The current specialists appear to be of good or high caliber within their fields, and the level of cooperation looks reasonable. It is, however, essential to expand the program and its scientific base rapidly by bringing in other scientists, by ensuring that new and young scientists (graduate students and post-doctorates) can get involved in this developing interdisciplinary area, and even by recruiting and supporting foreign laboratories and investigators. If such expansion does not occur, the program will resemble new model development by closed commercial corporations, with very few surprises anticipated or in store.

Indeed, the history of modeling climate over the past ten years is badly marred by a lack of systematic evaluation and comparison of competing models, as though these were competing corporations. Although one may hope for a change of attitude now, there is every reason to ensure an attitude of common goals and sharing, and of intellectual ferment and discovery, by introducing new, well-chosen people and capabilities into the field.

9 SUMMARY COMMENTS ON CHAMMP

One may hope for a number of outcomes from the CHAMMP program. At the head of the list is, of course, its nominal goal—to determine decisively the effect on earth's climate of the injection into the atmosphere of CO_2 and other gases of anthropogenic origin.

This is by no means a modest goal. Its successful completion will require substantial increases in computing power and parallel processing skills, as well as much more detailed information and modeling of the dynamic variables responsible for weather and climate. The most important new modeling considerations may be the climatic roles of clouds and the ocean, but biospheric influences on climate also appear to be significant. Furthermore, it is not at all clear at what scales (grid size) cloud physics must be modeled to keep simulation errors under control, and if the available increase in computational power will be adequate to accommodate whatever resolution is required. As is noted in Section 2, it is not even certain that climate is a "predictable" process.

Although uncertainties are attendant upon any research program, with CHAMMP they give rise to the distinct possibility that the program's main goal cannot be realized, or perhaps will not be realized in time to serve any purpose. However, if the program is appropriately framed, there could be very valuable scientific, technologic, and economic spin-offs (better weather forecasting, better seasonal predictions) during CHAMMP's lifetime.

Desirable spin-offs are likely to be realized if we stress the pieces of the model as they are developed, and carry out as much sensitivity analysis along the way as is possible. Through such testing and analysis, we will not only also learn important things about the model, but if the modeling is correct, we may learn important things about weather and climatology, without having to wait until the model is triumphantly completed.

In order to have useful consequences during its lifetime, the CHAMMP program must avoid having only one narrow goal, and it must avoid, to the greatest extent possible, getting locked into large, immutable, poorly understood computer codes. Instead, the use of a carefully devised, modularized code, such as the one alluded to in Section 4.7 above, might allow the ready acceptance and incorporation of any computational improvements discovered independently along the way. A poorly designed code would either reject such improvements, or would require painful and substantial rewriting.

A NEW COMPUTER ARCHITECTURE FOR CHAMMP?

A.1 Introduction

As discussed in other sections of this report, calculating global warming through numerical simulation is a very difficult but critical task. Because many millions of dollars of computer time (on conventional supercomputers) clearly will be needed for this task, a question arises: Could a special-purpose system architecture be developed that would be much more cost effective for this specialized calculation problem?

The answer to this question is surely yes, if a single algorithm is settled upon, and if the architecture is specifically designed to support this algorithm. On the other hand, if nothing is known about which algorithm, or even which class of algorithms, is to be used (i.e. if the architecture can only be a massive, general-purpose architecture), then the development of yet another large-scale, general-purpose architecture would be unwise.

Below we assume a new architecture is to be developed and examine what would be possible in the near term.

A.2 Range of Algorithms

The range of algorithm types that should be considered for models of global warming is indeed wide:

1. A "Lattice-Gas" model implemented on either a virtual or actual Cellular Automata machine, using only a "bit" representation.
2. A "finite-element or finite difference" representation of Newton's Laws (Navier Stokes equations, etc.) in the form of floating-point parameters on a (possibly adaptively sized) grid.
3. A "low-level object" or property representation (vorticity, etc.) on a grid with floating-point parameters.
4. A "model or spectral" representation on a sphere (that represents the earth) of "Newton's Laws" or low-level objects.
5. A "high-level object" representation of the large-scale dynamical elements of climate such as:
    - polar ice caps
    - hemispheric jet stream
    - large ocean circulation vortices
    - low- and high-pressure centers with their associated wind fields
    - continental masses
    - earth's core (heat surface)
    - solar flux.

Both floating-point numbers and symbolic (typed) identifiers would be employed.

A.3 Discussion of Algorithms

The "Lattice-Gas" algorithm would require a very special-purpose architecture to be effective (see JASON report on Lattice-Gas by Max and

Despain, 1988). Because no effective "Lattice-Gas" algorithm has yet been developed, this approach will not be considered further. (However, it might be argued that such algorithms should be investigated.)

All remaining algorithms can be encompassed in a fairly general set for which a moderately special architecture is appropriate. To support the remaining algorithms in a computer, the following architecture features are needed:

1. A powerful, single-chip processor with a single-chip, vector, floating-point co-processor attached. This pair of chips, plus a large fast-cache memory system, would provide about 50 MIP and 100 MFLOP of peak computer power, respectively—assuming a 50 MHZ clock and CMOS chips.
2. About 10,000 such processors organized to run in parallel, providing a TERAFLOP.
3. A hierarchical, heterogeneous memory system with the following parameters:
    - CPU pipeline registers, 2 nanoseconds
    - CPU register files, 5 nanoseconds
    - CPU cache, 10 nanoseconds
    - co-processor system cache, 20 nanoseconds
    - main memory, 100 nanoseconds
    - disc buffer memory, 500 nanoseconds
    - disc (bus-limited) memory, 30 msec/block.
4. A scaleable architecture, in the sense that a very wide range of computing power (at proportional cost) can be chosen, and the architecture

can be demonstrated for a modest cost before large expenditures are needed for a full-size machine.

5. Very good support of linear algebra (vector and matrix operations), especially in large-scale configurations. This implies careful attention to the network topology that connects the processing nodes.
6. Multiple instruction, multiple data (MIMD), shared-address, multiprocessor architecture (as opposed to a "message passing" multi-computer), which requires a global scheme for maintaining cache coherence.
7. Computation nodes composed of the following custom-designed VLSI CMOS chips:
   - main CPU chip
   - floating-point vector co-processor
   - cache controller
   - bus controller.
8. Commodity memory chips for each computation node:
   - static RAM chips (10 nanoseconds) for the node cache
   - dynamic RAM chips for the main memory and disc buffers.
9. A special interconnect bus system.

A.4 Strawman Architecture

There are, of course, many possible architectures that could meet the above requirements. However, for purposes of discussion (only) we propose the following "strawman"[1] architecture.

1. The full machine would have ≈ 10,000 computation nodes arranged in either three dimensions of 24 × 24 × 24 or in four dimensions of 10 × 10 × 10 × 10. (See Figure A-1 for a 10 × 10 version.)
2. Each dimension connects into computation nodes by a synchronization bus that maintains cache consistency throughout that dimension via a "snoopy bus" scheme. This arrangement requires that each bus support 10 to 24 processing nodes.
3. Computation nodes also function as agents to maintain cache consistency between buses as needed, employing a directory scheme.
4. This bus organization subsumes an orthogonal-bus topology, thus supporting large-scale, vector-matrix operations with minimal memory access interference.
5. Each computation node (≈ 10,000 altogether) would be composed of about 100 to 200 chips on a medium-sized, printed circuit board, with five connections per board (four for the four bus connections and one for a test I/O port).
6. Each computation node would have:
    - a custom VLSI CMOS main processor with a performance of about 50 MIPS
    - a custom VLSI CMOS vector floating-point co-processor with a performance of about 100 MFLOP
    - about 60 static RAM chips for the cache memory system (≈ 10 nsec access time)

---

[1] Note: The term "strawman" is meant to imply that this architectural proposal is intended to provoke critical discussion that will expose its flaws and lead to the development of a viable architecture in the future.

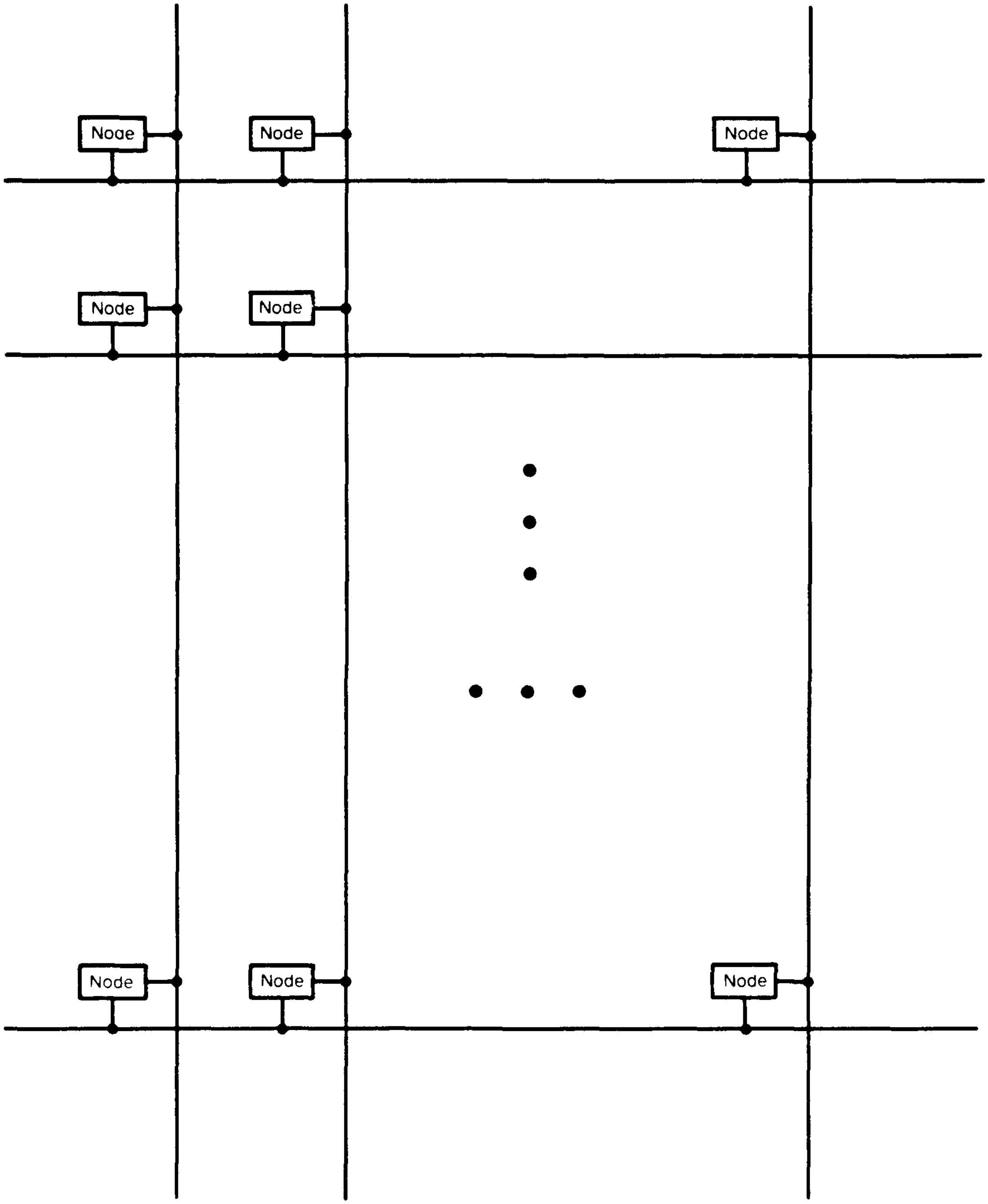

Figure A-1. Two-dimensional version of the strawman bus architecture.

- a custom VLSI CMOS cache-controller chip
- about 64 4M-bit dynamic RAM chips to provide for a 32M-byte main memory
- a standard dynamic RAM controller chip
- four custom VLSI-CMOS bus-controller/interface chips
- a number of standard bus-driver and receiver chips
- miscellaneous glue-logic chips.

A-2 illustrates the computational node architecture. Each computational node is a stand-alone computer system, yet it is designed to cooperate and share its memory address space with up to about 10,000 other such nodes. A prototype of this architectural concept could be cheaply verified experimentally before a massive system is constructed.

A.5 Software

Although a global shared-address memory scheme will greatly simplify the software problem, the cost of the strawman system software is likely to far exceed the hardware cost. The following approach to software development is suggested:

1. Adopt MACH as the operating system. This UNIX-style operating system was developed by DARPA at CMU to replace UNIX on future architectures and currently has a large support community. The strawman will be much easier to adapt to MACH than to one of the UNIX systems.
2. Develop a FORTRAN compiler, then a C compiler, and finally a compiler for a new language, described below.

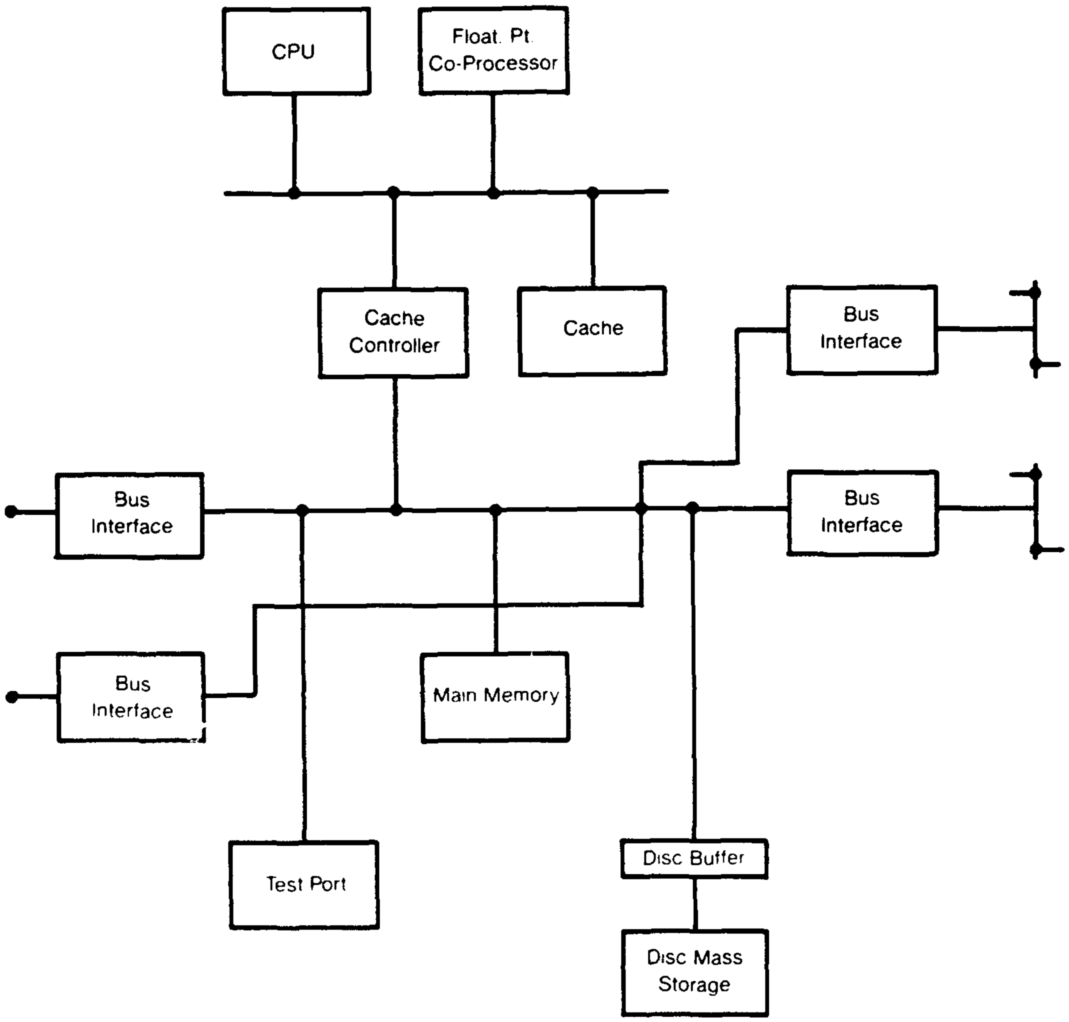

Figure A-2. Computational node

3. Develop a formal, specialized computer language to directly express global warming calculations at a very high level. This language should be a narrow, specialized language that formalizes the informal language employed by experts in global warming.
4. Pay particular attention to graphical display and interaction software (user interface), which is especially critical.

A.6 Conclusions

It appears that a special architecture to support the calculation of global warming could be quite effective for large-scale calculations. Such a system could be developed in a period of three to five years and is potentially ten times more cost-effective than commercial general-purpose machines, although *further study with simulations* is needed to determine the expected cost-effectiveness. In addition, other architecture variations should be proposed and evaluated.

If it is deemed that a new architecture is desirable, we recommend that a workshop be organized to explore all the ideas in the computer architecture community that might be applied to this problem. The strawman architecture proposed herein could serve as a starting point for discussion at such a workshop.

DISTRIBUTION LIST

Dr Henry D I Abarbanel
Institute for Nonlinear Science
Mail Code R002/Building CMRR/Room 115
University of California/San Diego
La Jolla, CA 92093-0402

Director Ames Laboratory [2]
Iowa State University
Ames, IA 50011

Mr John M Bachkosky
Deputy DDR&E
The Pentagon
Room 3E114
Washington, DC 20301

Dr Joseph Ball
Central Intelligence Agency
Washington, DC 20505

Dr Arthur E Bisson
DASWD (OASN/RD&A)
The Pentagon
Room 5C675
Washington, DC 20350-1000

Dr Albert Brandenstein
Chief Scientist
Office of Natl Drug Control Policy
Executive Office of the President
Washington, DC 20500

Mr Edward Brown
Assistant Director
Nuclear Monitoring Research Office
DARPA
3701 North Fairfax Drive
Arlington, VA 22203

Dr Herbert L Buchanan III
Director
DARPA/DSO
3701 North Fairfax Drive
Arlington, VA 22203

Dr Curtis G Callan Jr
Physics Department
PO Box 708
Princeton University
Princeton, NJ 08544

Dr Ferdinand N Cirillo Jr
Central Intelligence Agency
Washington, DC 20505

Mr Phillip Colella
Dept of Mechanical Engineering
University of California/Berkeley
Berkeley, CA 94720

Brig Gen Stephen P Condon
Deputy Assistant Secretary
Management Policy &
Program Integration
The Pentagon Room 4E969
Washington, DC 20330-1000

DISTRIBUTION LIST

Ambassador Henry F Cooper
Director/SDIO-D
Room 1E1081
The Pentagon
Washington, DC 20301-7100

DARPA
RMO/Library
3701 North Fairfax Drive
Arlington, VA 22209-2308

Mr John Darrah
Senior Scientist and Technical Advisor
HQAF SPACOM/CN
Peterson AFB, CO 80914-5001

Dr Alvin M Despain
Electrical Engineering Systems
SAL-318
University of Southern California
Los Angeles, CA 90089-0781

Col Doc Dougherty
DARPA/DIRO
3701 North Fairfax Drive
Arlington, VA 22203

DTIC [2]
Defense Technical Information Center
Cameron Station
Alexandria, VA 22314

Mr John N Entzminger
Director
DARPA/TTO
3701 North Fairfax Drive
Arlington, VA 22203

CAPT Kirk Evans
Director Undersea Warfare
Space & Naval Warfare Sys Cmd
Code PD-80
Department of the Navy
Washington, DC 20363-5100

Mr F Don Freeburn
US Department of Energy
Code ER-33
Mail Stop G-236
Washington, DC 20585

Dr Dave Galas
Associate Director for
Health & Environmental Research
ER-70/GTN
US Department of Energy
Washington, DC 20585

Dr S William Gouse
Sr Vice President and General Manager
The MITRE Corporation
Mail Stop Z605
7525 Colshire Drive
McLean, VA 22102

LTGEN Robert D Hammond
CMDR & Program Executive Officer
US Army/CSSD-ZA
Strategic Defense Command
PO Box 15280
Arlington, VA 22215-0150

DISTRIBUTION LIST

Mr Thomas H Handel
Office of Naval Intelligence
The Pentagon
Room 5D660
Washington, DC 20350-2000

Maj Gen Donald G Hard
Director of Space and SDI Programs
Code SAF/AQS
The Pentagon
Washington, DC 20330-1000

Dr Robert G Henderson
Director
JASON Program Office
The MITRE Corporation
7525 Colshire Drive Z561
McLean, VA 22102

Dr Barry Horowitz
President and Chief Executive Officer
The MITRE Corporation
Burlington Road
Bedford, MA 01730

Dr William E Howard III [2]
Director For Space
and Strategic Technology
Office/Assistant Secretary of the Army
The Pentagon Room 3E474
Washington, DC 20310-0103

Dr Gerald J Iafrate
US Army Research Office
PO Box 12211
4300 South Miami Boulevard
Research Triangle Park, NC 27709-2211

Technical Information Center [2]
US Department of Energy
PO Box 62
Oak Ridge, TN 37830

JASON Library [5]
The MITRE Corporation
Mail Stop W002
7525 Colshire Drive
McLean, VA 22102

Dr George Jordy [25]
Director for Program Analysis
US Department of Energy
MS ER30 Germantown
OER
Washington, DC 20585

Dr O'Dean P Judd
Los Alamos National Lab
Mail Stop A-110
Los Alamos, NM 87545

Dr Steven E Koonin
Kellogg Radiation Laboratory
106-38
California Institute of Technology
Pasadena, CA 91125

Dr Chuck Leith
LLNL
L-16
PO Box 808
Livermore, CA 94550

DISTRIBUTION LIST

Dr Herbert Levine
Department of Physics
Mayer Hall/B019
University of California/San Diego
La Jolla, CA 92093

Technical Librarian [2]
Argonne National Laboratory
9700 South Cass Avenue
Chicago, IL 60439

Research Librarian [2]
Brookhaven National Laboratory
Upton, NY 11973

Technical Librarian [2]
Los Alamos National Laboratory
PO Box 1663
Los Alamos, NM 87545

Technical Librarian [2]
Pacific Northwest Laboratory
PO Box 999
Battelle Boulevard
Richland, WA 99352

Technical Librarian [2]
Sandia National Laboratories
PO Box 5800
Albuquerque, NM 87185

Technical Librarian [2]
Sandia National Laboratories
PO Box 969
Livermore, CA 94550

Technical Librarian [2]
Lawrence Berkeley Laboratory
One Cyclotron Road
Berkeley, CA 94720

Technical Librarian [2]
Lawrence Livermore Nat'l Lab
PO Box 808
Livermore, CA 94550

Technical Librarian [2]
Oak Ridge National Laboratory
Box X
Oak Ridge, TN 37831

Chief Library Branch [2]
AD-234.2 FORS
US Department of Energy
Washington, DC 20585

Dr Gordon J MacDonald
Institute on Global Conflict
& Cooperation
UCSD/0518
9500 Gilman Drive
La Jolla, CA 92093-0518

DISTRIBUTION LIST

Mr Robert Madden [2]
Department of Defense
National Security Agency
Attn R-9 (Mr Madden)
Ft George G Meade, MD 20755-6000

Dr Arthur F Manfredi Jr [10]
OSWR
Central Intelligence Agency
Washington, DC 20505

Mr Joe Martin
Director
OUSD(A)/TWP/NW&M
Room 3D1048
The Pentagon
Washington, DC 20301

Dr Nicholes Metropolis
Los Alamos National Laboratory
M/S B210
Los Alamos, NM 87545

Dr Julian C Nall
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311

Dr William A Nierenberg
Director Emeritus
Scripps Institution of Oceanography
0221
University of California/San Diego
La Jolla, CA 92093

Dr Jerry North
College of Geosciences
Dept of Meteorology
Texas A&M University
College Station, TX 77843-3146

Dr Gordon C Oehler
Central Intelligence Agency
Washington, DC 20505

Oak Ridge Operations Office
Procurement and Contracts Division
US Department of Energy
(DOE IA No DE-AI05-90ER30174)
PO Box 2001
Oak Ridge, TN 37831-8757

Dr Peter G Pappas
Chief Scientist
US Army Strategic Defense Command
PO Box 15280
Arlington, VA 22215-0280

Dr Aristedes Patrinos [20]
Director of Atmospheric
& Climate Research
ER-74/GTN
US Department of Energy
Washington, DC 20585

Dr Bruce Pierce
USD(A)/D S
Room 3D136
The Pentagon
Washington, DC 20301-3090

DISTRIBUTION LIST

Mr John Rausch [2]
Division Head 06 Department
NAVOPINTCEN
4301 Suitland Road
Washington, DC 20390

Records Resources
The MITRE Corporation
Mailstop W115
7525 Colshire Drive
McLean, VA 22102

Dr Oscar S Rothaus
Math Department
Cornell University
Ithaca, NY 14853

Dr Fred E Saalfeld
Director
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Dr John Schuster
Technical Director of Submarine
and SSBN Security Program
Department of the Navy OP-02T
The Pentagon Room 4D534
Washington, DC 20350-2000

Dr Barbara Seiders
Chief of Research
Office of Chief Science Advisor
Arms Control & Disarmament Agency
320 21st Street NW
Washington, DC 20451

Dr Philip A Selwyn [2]
Director
Office of Naval Technology
Room 907
800 North Quincy Street
Arlington, VA 22217-5000

Dr Albert Semtner
Department of Oceanography
Naval Post Grad School
Code 68
Monterey, CA 93949

Superintendent
CODE 1424
Attn Documents Librarian
Naval Postgraduate School
Monterey, CA 93943

Dr George W Ullrich [3]
Deputy Director
Defense Nuclear Agency
6801 Telegraph Road
Alexandria, VA 22310

Ms Michelle Van Cleave
Asst Dir/National Security Affairs
Office/Science and Technology Policy
New Executive Office Building
17th and Pennsylvania Avenue
Washington, DC 20506

Dr John Fenwick Vesecky
Dir Space Physics Res Lab
University of Michigan
1424A Space Research Bldg
Ann Arbor, MI 48109-2143

DISTRIBUTION LIST

Mr Richard Vitali
Director of Corporate Laboratory
US Army Laboratory Command
2800 Powder Mill Road
Adelphi, MD 20783-1145

Dr Edward C Whitman
Dep Assistant Secretary of the Navy
C3I Electronic Warfare & Space
Department of the Navy
The Pentagon 4D745
Washington, DC 20350-5000

Mr Donald J. Yockey
U/Secretary of Defense
For Acquisition
The Pentagon Room 3E933
Washington, DC 20301-3000

Dr Linda Zall
Central Intelligence Agency
Washington, DC 20505

Mr Charles A Zraket
Trustee
The MITRE Corporation
Mail Stop A130
Burlington Road
Bedford, MA 01730