H. G. Kolsky

Some Computer Aspects of Meteorology

Abstract: A long history of large-scale scientific computing is associated with numerical weather prediction. Recently, interest in this field has been renewed as a result of international studies concerning the feasibility of a global observation and analysis experiment preliminary to the World Weather Watch. This paper describes the physical phenomena occurring in the atmosphere and the problems of modeling them for computer analysis. The numerical methods commonly used in general circulation models are described briefly and the relative advantages discussed. Finally, an analysis of the computer requirements for global weather calculations is developed and the need pointed out for very fast computers capable of executing the equivalent of hundreds of millions of instructions per second.

Introduction

Even in this age of scientific superlatives it is hard to find a field more far-reaching, with more interesting problems and more difficulties, than that of numerical weather prediction. The associated atmosphere physics behind it is literally world-wide. A large number of physical disciplines interact with each other in a most complex way. Fluid dynamics, which describes the major motions of the atmosphere and oceans, is considered classical physics. However, the energy sources and frictional forces which must be included introduce quantum mechanics and diffusion theory as well as numerical analysis.

Although one cannot underestimate the importance of the physical theory and numerical methods, the real history of numerical weather prediction has been essentially tied to that of the speed and capacity of the computers available. Richardson's attempt[R1] in the 1920's to perform numerical weather calculations starting from the primitive equations of hydrodynamics is well known. He proposed tying the calculations into an operational network. The magnitude of his effort and the difficulties which he encountered were enough to discourage serious work in this area for over twenty years. However, reading his account now is an interesting experience because it sounds surprisingly modern. One would have to make relatively few serious changes in Richardson's book to bring it out as a modern treatise in the field. There is, however, an enormous difference between the theory and the accomplishments of these early reports and those of today—the growth in speed and capacity of the available computers has been the key ingredient.

The purpose of this paper is to survey the basic physical phenomena concerning the atmosphere and some of the numerical and computer design problems that arise in attempting to model it. The emphasis is on general circulation research problems rather than on existing operational methods. Our feeling is that these research problems are the ones which stretch the state of the art, place the heaviest demands on computers, and point the way to the future.

Physical phenomena

In discussing the requirements for improved weather forecasting, Phillips[P1] offers the following comment: "Unfortunately, faster computing machines are not the only requirements for improved weather predictions. The basic physical equations, which are nonlinear, presuppose an extremely detailed knowledge of the state of the atmosphere at the beginning of the forecast. For example, the viscosity term in the Navier-Stokes hydrodynamic equation is of fundamental importance because it is ultimately responsible for the frictional dissipation of kinetic energy in the atmosphere. However, it can perform this vital function in a numerical calculation only if the latter includes motion on scales as small as a millimeter. Analogous difficulties appear in other equations, especially those describing condensation of water vapor and precipitation (where the fundamental physical laws apply to individual raindrops) and radiation effects (where the molecular spectra are extremely complicated). The most important weather phenomena, on the other hand, have horizontal scales of 10^5

to 10^7 meters, and experience has shown that it is necessary to consider conditions over almost an entire hemisphere to predict the weather several days in advance. It is obviously impractical to allow for this scale ratio of 10^{10} in any conceivable computation scheme."

- *Scaling approximations*

To deal with these difficulties of scale many kinds of approximations in the field of dynamic meteorology have been developed. They are all concerned with including or omitting certain physical quantities from the model. As a consequence, the solutions of such approximate, discrete systems do not describe all physical phenomena associated with the complete system of differential equations. Such approximations are said to "filter out" entire ranges of phenomena, e.g. sound waves, which should not have an effect on the answers of interest, e.g. large-scale cyclonic motion.

In the scale-analysis approach, it is assumed that all dependent variables—such as the velocity—are characterized by a "well-defined rate of variation," i.e., a *scale*, in space and time. In particular, it is assumed that a partial derivative of a quantity will have an order of magnitude at most equal to the magnitude of the quantity divided by the appropriate scale length. This scale-analysis method has the advantage of maintaining a relatively clear and unambiguous relationship between the "true" variables in the atmosphere and the variables in the simplified equation. It still requires some physical intuition, however, since definite statements about the order of magnitude of various quantities are necessary.

Figure 1, from a graph by Arakawa,[A1] shows in a clear way how many of the regions of approximation are related. In particular, it shows where ordinary general circulation calculations fit into the range of physical phenomena. (Their characteristic time scale is from 10^3 to 10^5 seconds and their characteristic length scales from 10^5 to 10^7 meters). These are motions in which the Coriolis force from the earth's rotation is important.

Fortunately, motions on this large scale are not only responsible for most day-to-day weather changes (and are therefore worth forecasting), but it seems that their behavior can be predicted satisfactorily over periods of several days without too much detailed consideration of the unknown smaller-scale phenomena. "Small scale" phenomena in this case include individual thunderstorms, tornados, and even hurricanes. Except for special warning networks, they fall outside the regular observations and realistic computing models.

- *Fundamental conservation equations*

The physical and mathematical basis of all methods of dynamical weather prediction lies in the principles of conservation of momentum, mass, and energy. Applied to the

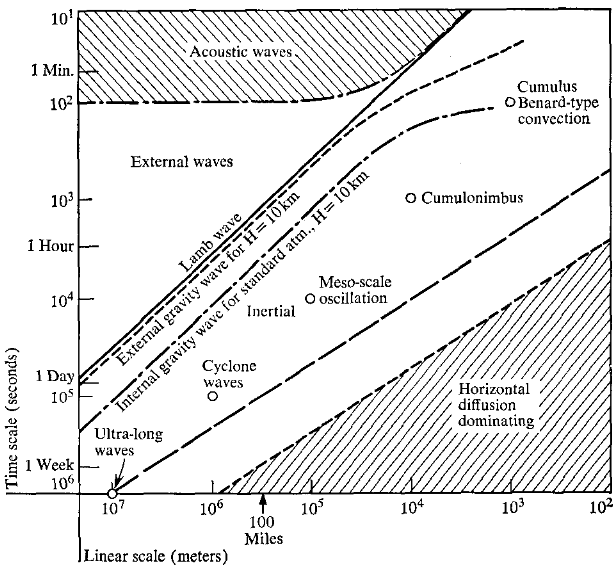


Figure 1 A graph showing the scales of various atmospheric motions in wavelength and time (from Ref. A1).

quasicontinuous statistical motion of an assemblage of liquid or gas molecules (through the methods of kinetic theory and statistical mechanics), these fundamental principles are expressed mathematically in Newton's equations of motion for a continuous medium, the equation of continuity (for mass conservation), and the thermodynamic energy equation. So far as is known, these equations are universal, in that they evidently apply to all fluids in normal ranges of pressure, temperature, and velocity, without regard to composition, container, or state of motion.

The basic equations contain terms which describe such physical entities as heat influx, water vapor influx and frictional forces, which after discretization and scaling assumptions are no longer implicitly described by the system. Such phenomena are said to be "parameterized" in the model. For example, the fact that the transport of water vapor into the atmosphere from water surfaces is accomplished by small eddies means that even the release of latent heat by precipitation cannot be successfully treated without including the effects of turbulence.

A detailed discussion of the conservation equations of fluid dynamics and their numerical difference formulations is not considered necessary in this paper since they have been discussed at length by Quarles and Spielberg,[Q1] Thompson,[T1] and Smagorinsky.[S1]

The importance of the nonadiabatic effects, friction and precipitation is discussed by Phillips.[P1]

- *Atmospheric heating and cooling by radiation*

The effects of radiative cooling in the atmosphere are less well formulated in existing models, but progress is being made.[S2] Although the hydrodynamic energy of the atmosphere is dominant and its motion can be calculated adiabatically to a fairly reasonable approximation, one must keep in mind that the earth receives virtually all of its energy from space in the form of electromagnetic radiation from the sun.

The transformation of the incident solar radiation into scattered and thermal radiation, and the consequent thermodynamic effects on the earth's gaseous envelope, are very complicated phenomena, requiring the most advanced methods of molecular physics and quantum mechanical calculations. Absorption along a real atmospheric path, where pressure, temperature and composition all vary, presents problems only a few of which have been solved. (See Goody[G1]).

• *Calculation of cloud effects*

If a cloud layer is present, the drops of water of which a cloud consists are comparable in size to the wavelengths of thermal radiation, and their number per unit volume is quite high. In this case consideration of the scattering is, therefore, of great importance. The accurate solution of the problem of radiative heat transfer in clouds can be obtained only by using a detailed equation for radiative energy transfer. Calculations indicate that a cloud is "active" with respect to thermal radiation only around its edges. The flux of thermal radiation entering the cloud is completely absorbed in a distance of a few tens of meters.

Probably the most formidable computational problems meterologists are likely to face will arise from the calculation of cumulus convection. In such problems no simple hydrostatic approximation for vertical motion can be assumed. The phase changes of water in air will have to be carried in four forms—vapor, ice, water and droplets. Even electrostatic forces between droplets may have to be calculated, at least in some average sense. These microscale calculations are as demanding of computer time as the general circulation problems.

• *Air-sea interface*

Although the most important driving force for the atmosphere is ultimately the radiation from the sun, the second most important is no doubt the atmosphere's interactions with the ocean. The tremendously large heat capacity represented by the oceans of the earth provide its stability and relative uniformity of temperature. The oceans and the air can be considered a two-fluid system coupled relatively loosely but coupled in a very important way.

A Government report edited by Benton[B1] emphasized the fact that the atmosphere and the oceans together form a single mechanical and thermodynamic system, and that an understanding of the way in which energy, gases, particles, and electric charges move across the interface between the two is essential to the development of geophysics. Energy transfer, in the form of radiation or latent heat, affects the circulation of both the atmosphere and the oceans. The physical phenomena are complex and have far-reaching consequences. For example, water vapor, oxygen, carbon dioxide, and other gases move across the interface and influence both the composition of the atmosphere and the life cycle of marine organisms. Salt particles from the oceans provide condensation nuclei for precipitation. Charge separation in ocean spray may be significant in the development of differences in electrostatic potential between the atmosphere and the earth's surface.

• *Weather and climate modification*

Certainly no discussion of the future of meteorological computations would be complete without mentioning the enormous implications of weather and climate modification. Several studies have been made in recent years concerning the general problems of weather modification. The most recent study, headed by G. J. F. MacDonald,[N1] has resulted in an excellent report which is recommended to anyone interested in the subject.

The major portion of atmospheric energy exchange is due to the release of instabilities inherent in the preferred states of the atmosphere. These dynamically unstable situations are looked upon as "levers" or "soft spots" in the system where man's efforts might be able to trigger a chain of nature reactions.

Man has exploited these instabilities on a limited scale in the belief that the effects would be as short-lived as the phenomena themselves and that the energies released would not escalate to the level which would change the weather permanently.

It is obvious from geological evidence that the earth-atmosphere system can support radically different climatic regimes, some of which could be disastrous to civilization. We do not yet know what can cause a shift from one climatic regime to another, whether change can occur in an "instant" of geologic time or only as a secular cyclic process; our few theories still hang on the most tenuous evidence.

Numerically integrated mathematical models of the atmosphere have come to be regarded as necessary tools for research in modification of the atmosphere. This is particularly true in areas where actual experimentation would be too costly, take too long, or possibly be irreversible.

Numerical methods

Because the partial differential equations (or in the case of radiation, integral equations) for a fluid are nonlinear and possess difficult initial and boundary conditions, they must be solved by numerical methods for practical cases. This involves converting the equations describing the particular physical model under consideration into a form which can be solved by numerical algorithms on a computer.

Although it is a simple matter to convert a given system of differential equations to some finite difference form, it is a far more difficult task to obtain physically meaningful results even if the integration method proves to be stable. The hydrodynamical equations of motion contain among their solutions the high-speed sound and gravity waves which, meteorologically speaking, are spurious information. Unless an unrealistically small time step (several seconds) is used, these solutions have a tendency to amplify in time and overshadow the physically meaningful results. It is necessary to find a set of difference equations which will be stable in the sense that the calculation can go on for an indefinite time without nonsensical results developing. This is a particularly sensitive matter in an atmospheric model since there are conditions under which the atmosphere itself can be temporarily unstable so that small disturbances really do grow. This must be permitted in the numerical model also, but in a reasonable way.

By far the most serious obstacle to solving the hydrodynamical equations arises from the properties of the atmosphere itself. The large-scale horizontal accelerations of air are about an order of magnitude less than either of the forces per unit mass taken individually, i.e., the Coriolis force due to the rotating earth and horizontal pressure-gradient force. The atmosphere maintains itself near some state of balance and much more nearly so than is revealed by direct measurement.[C5]

For similar reasons, the horizontal divergence of the wind velocity cannot be computed accurately from direct measurements of the wind. It can be inferred indirectly that the sum $(\partial u/\partial x + \partial v/\partial y)$ is generally an order of magnitude less than either $\partial u/\partial x$ or $\partial v/\partial y$ taken individually, i.e., the latter tend to compensate each other almost completely. Thus, in order to compute $\nabla \cdot V = \partial u/\partial x + \partial v/\partial y$ to within 10% accuracy, the wind components must be measured to within 1% accuracy. Winds, however, are not measured and reported to within better than 10% accuracy. As a result, the vertical air speed cannot be computed accurately unless spuriously large fluctuations of divergence can somehow be suppressed.

Present global general-circulation models use equations in which the *hydrostatic assumption* is made. (This approximate set of equations is somewhat misleadingly referred to as the "primitive equations" in meteorological literature.) This is an accurate approximation for motions with horizontal scales of 25 miles or longer. Vertically propagating sound waves are excluded by this technique but gravity waves are retained. The size of the time step permitted is proportional to the horizontal space increment; for a latter value of 125 miles, the time step should be less than 10 minutes.

• *Finite difference solutions and nonlinear instability*

At the present time no single finite-difference analog of the primitive equations has emerged which meets *all* the objectives of the meteorological community. Such a set of difference equations must be, in addition to being "physically" acceptable, mathematically *accurate* and *stable*. By accuracy is meant consistency, and by stability, convergence.

In 1956, Phillips[P2] in his early attempt at long-term integration of the meteorological equations encountered an unexpected difficulty. After about 20 simulated days the solution began to show a structure termed "noodling," in which the motion degenerates into eddies of elongated, filamented shapes. Once formed, the eddies intensify without limit, causing a nonlinear computational instability and explosive growth of the total kinetic energy. Phillips showed that the instability is caused by "aliasing" or misrepresentation of the shorter waves because a finite grid cannot properly resolve them. Phillips showed further that the instability could not be reduced by shortening the time interval.

For a discussion of nonlinear stability and how it can be overcome using the "leapfrog" scheme, see Leith's fractional time step method,[L1], Arakawa's method[A2] and the discussion by Kolsky.[K3]

• *Brief descriptions of existing general circulation research models*

At the present time there are four research groups in the U. S. actively working on general circulation models. Abroad there is one large group working in the USSR at Novosibirsk. Many other groups here and abroad are working on the theoretical aspects of general circulation or are using numerical flow calculations for operational forecasts. Research in general circulation calculations obviously is heavily dependent upon the availability of large computers and upon the existence of a wealthy sponsor (usually some government agency).

The present researchers in the field all owe a great debt to the work of Charney and Phillips[C2] at the Institute for Advanced Study in the early 1950's. They also owe much to the early operational models of Shuman[S5] and Cressman and Bedient,[C4] who verified the importance of numerical prediction. The following descriptions (listed in alphabetical order) give some of the main features of recent programs.

The Kasahara-Washington model—This model is being developed by the newest group in the field, that of Drs. A. Kasahara and W. Washington[H2] at NCAR. The model has been in an evolutionary state in which more complicated physical approximations are being included one by one.

The hydrodynamic and thermodynamic evolution of a dry atmosphere is computed over the whole globe, including solar heating, surface boundary-layer effects, and a simple prescription for the latent heat release due to precipitation. The prognostic variables, that is, the ones which are used to advance the model in time, are pressure, temperature, and horizontal wind velocity on a five-degree spherical mesh similar to that used by Leith, with two levels in altitude.

Kasahara and Washington use a three-level "leapfrog" scheme for 50 time steps—which is inherently unstable—followed by one cycle of the Lax-Wendroff procedure, whose damping properties prove to be sufficiently strong as to render the whole procedure stable. This procedure requires that the data be stored at two time levels.

The Leith model—An intermediate-term general circulation model computer program has been developed over the past five years by Dr. Cecil Leith[L1] of the Lawrence Radiation Laboratory at Livermore.

The hydrodynamic and thermodynamic evolution of a moist atmosphere is computed over the whole globe, taking into account such effects as solar heating, latent heat of evaporation and precipitation, surface friction of the earth, and eddy viscosity. The prognostic dependent variables are temperature, water vapor content, isobaric wind velocity, and surface pressure, and the diagnostic dependent variables are isobaric wind divergence, vertical velocity, and geopotential. Independent variables are latitude and longitude at five-degree intervals, pressure at six levels from 1.0 to 0.1 bars, and time at ten-minute intervals.

The time-development of the prognostic dependent variables is obtained by numerical integration of the primitive equations by a semi-implicit method of fractional time steps which is second-order in time but requires data stored at only a single time level. The thermodynamic heat source contains approximations to the absorption of solar energy by water vapor—a geometric computation without clouds—and a prescribed radiative cooling rate which is a function of pressure alone. Rainfall is assumed to develop at every point of local supersaturation and the so-obtained latent heat is found to be an important thermodynamic source. No mountains are considered in the present calculation.

The mesh is scanned vertically for each horizontal mesh point and all longitudes for each latitude, starting at the equator and going to the North Pole, then back to the equator and proceeding to the South Pole to complete one time iteration. Three complete latitudes of data at one time step are in core at one time.

The Mintz-Arakawa model—This is a two-level global model developed by Professors Y. Mintz[M1] and A. Arakawa of UCLA. The upper boundary condition is best summarized by picturing the stratosphere as a layer of weightless cork floating on the troposphere. Thus there is no infusion of mass, momentum, or moisture into the troposphere from above, and the upper boundary is an isobaric surface (0.2 bar).

The lower boundary condition is quite complicated, for the model accounts for orography (i.e., land elevation) and for air-sea-ice difference. The continental land masses are assumed to have zero thermal conductivity, and hence zero heat capacity. Consequently, the surface temperature of the land is a computed quantity, viz., that temperature which gives zero net heat flux through the air-ground interface. Conversely, it is assumed that the oceans have infinite heat capacity and zero advection, and hence that the ocean surface temperature is a prescribed function of season. Where the surface is ice, the ice is also assumed to have zero thermal conductivity. However, an upper limit of 0° C is taken for the ice.

The Arakawa differencing scheme[A2] is applied to these equations in their source-free form, thus assuming that the averaged variations in the conserved quantities do indeed result from the physical sources and sinks rather than from truncation error. The Matsuno approximated backward-differencing scheme is used; this gives almost no spurious damping of the meteorologically importance motions.

The Smagorinsky models—The group headed by Dr. Joseph Smagorinsky at the Geophysical Fluid Dynamics Laboratory of ESSA represents the oldest and most experienced research group currently working in the field. They have experimented with many models over the years. The most recent model, published by Smagorinsky, Manabe and Holloway,[S2] uses nine vertical levels distributed so as to resolve surface boundary layer fluxes as well as radiative transfer by ozone, carbon dioxide, and water vapor. The lower boundary is a kinematically uniform land surface without any heat capacity. The stabilizing effect of moist convection is implicitly incorporated into the model by requiring an adjustment of the lapse rate (the vertical gradient of temperature) whenever it exceeds the moist adiabatic value. The numerical integrations are performed for the mean annual conditions over a hemisphere starting with an isothermal atmosphere at rest. The grid points of the calculation are located on a stereographic projection plane centered on the pole. The spatial distribution of gaseous absorbers is assumed to have the annual mean value of the actual atmosphere and to be constant with time.

Operational models—Operational numerical weather prediction first came into its own with the formation of the Joint Numerical Weather Prediction unit (JNWP) in 1954. The U. S. Weather Bureau, the Air Force and the Navy jointly established the JNWP to capitalize on the research[C5] which had been done at the Institute for Advanced Study, the Air Force's Cambridge Research Center, and under the late C. G. Rossby at Stockholm. The three main operational groups in existence today in the United States are direct descendants of the JNWP. They are: the ESSA Weather Bureau's National Meteorological Center at Suitland, the 3rd Weather Wing at Offutt AFB, and the Fleet Numerical Weather Facility at Monterey.

Of course, there is a tremendous difference between doing research on general circulation and issuing operational forecasts every twelve hours. The pressure of the latter has forced the use of simpler models which have been tried and found reliable under operational conditions. The operational models are steadily expanding as techniques are improved by research and as better data become available. For more details, see for example papers by Cressman,[C5] Wolff,[W1] O'Neil,[O1] Stauffer[S6] and their collaborators.

The main difference between research and operations can be related to the "real data" problem. The operational groups are geared to the world-wide data acquisition and communication networks which furnish the initial conditions for their prediction models. A very important phase of data collection is the verification and smoothing of the raw measurements for the entry into the numerical model. For example, the National Meteorological Center uses a combination of three IBM computers to perform this task. The distribution of the completed products of the centers to their customers is another large topic which we can only mention here.

- *Nonfinite difference methods in numerical weather forecasting*

In investigating the future computer requirements of numerical weather calculations, one always has the uneasy feeling that some radical departure from the present finite-difference methods will make the computing estimates completely invalid. The hope is that some entirely different numerical method might reduce the computations required, by many orders of magnitude.

One suggestion which arises repeatedly is that Fourier transform methods might be employed for the calculations. This was considered as early as 1950 by Charney, Fjørtoft and von Neumann[C1] as a means for solving the Poisson equation arising in their geostrophic model.

The necessity of solving the Poisson equation on each time step which arises in the geostrophic models, is being abandoned by the primitive equation approaches currently in vogue.

Fourier methods, however, have continued to be used to analyze and treat stability problems in numerical procedures. These yield a good description of the nonlinear instability arising from a cascading of energy from low- to high-frequency waves due to "aliasing" mentioned earlier. This phenomenon results from the fact that products of functions yield Fourier components outside the spatial frequency range corresponding to the grid size. These components then contribute erroneously to the frequencies, modulo the number of grid points, lying in the spectrum corresponding to the grid. This suggests filtering out these high frequencies by the introduction of diffusion terms in the equations or by periodically sweeping over the grid with some averaging process. But it also suggests a less artificial means, namely the carrying out of the calculation in the spatial frequency domain with the simple expedient of dropping Fourier components outside the frequency range being considered.

Calculations of Fourier transforms of actual weather maps do not indicate any particular wavelength range that can always be discarded.

More recently, spectral methods have become interesting because new methods for computing Fourier transforms have been devised by Cooley and Tukey[C3] which reduce the number of operations from N^2, where N is the number of data points, to $N \log_2 N$. It has been found that Fourier transform methods, applied to the solution of the Poisson equation by R. W. Hockney[H1] have resulted in computation speeds ten times as great as the best iterative methods.

The most serious objection to the use of spectral methods is that they are generally not extendable to even slight complications in the equations, such as the introduction of nonconstant coefficients or extra terms. This property, more than any other, makes it unlikely that they can be applied to the primitive equations in a realistic way. Nevertheless, the possibility that someone may have a brilliant idea keeps interest alive.

High-speed computer characteristics for meteorological calculations

It is futile, of course, to try to describe a computer in terms of a single speed number or to quote storage in terms of a single value.[K1] However, if such numbers are not taken too seriously, they can be helpful in giving an idea of the capabilities of these devices. The secret is to emphasize the important effects; that is, the ones whose inclusion results in a factor of two or more, while trying not to get lost in the myriad 10% effects.

Figures 2 and 3 show the progress of large machines versus time, both in the raw speed (measured in terms of millions of instructions per second executed) and in the internal storage (rated in terms of characters of storage regardless of size of the words). Word size itself can affect useful storage by as much as a factor of two effect but usually it does not have this effect for scientific applications. The graphs show a steady increase in speed and storage and, if one overlooks occasional plateaus and fluctuations, there is almost a constant rate of increase of performance and storage. Experts in the computer field have been predicting a coming saturation in computer speed and size ever since the beginning. It has not yet happened, however, and there are no indications that it will necessarily happen in the next generation of computers. The velocity of light, which is often quoted as the main barrier to computer speed, is still there, but it has not proved to be the barrier we expected. There are ways of obtaining an effectively high instruction operation rate other than simply making the individual components go faster and faster.

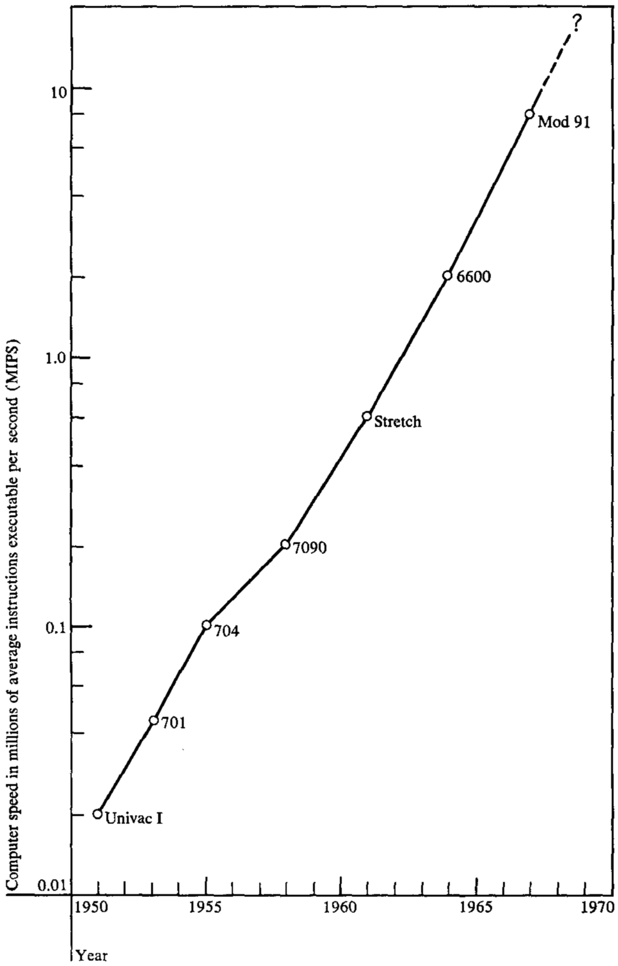


Figure 2 Growth of computer speed vs time.

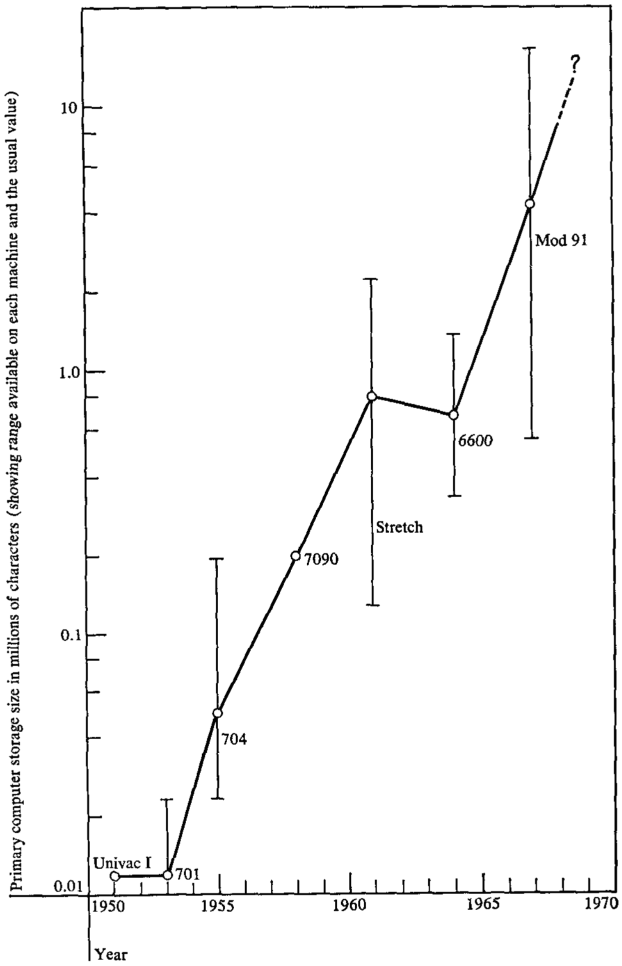


Figure 3 Growth of computer storage vs time.

Since the beginning of the development of modern computers there has always been at least one machine at any given time which could be called the "supercomputer" of its period. These machines are always "stretching the state of the art" of the technology at their time. The STRETCH computer (IBM 7030) obtained its name from this phenomenon. Every supercomputer built to date, and probably most new ones to be discussed in the future, has one thing in common, i.e., their protagonists use the numerical weather problem as one of the reasons for developing their machines. It has the following advantages for this purpose: It can saturate the biggest computer. It is very important for the nation to solve it. It resembles many other problems based on partial differential equations and is completely unclassified from the national security point of view.

A cynic might say that numerical weather prediction seems to be a "feedback loop" in which the careful analysis of physical theories and numerical methods is aimed mainly at justifying the calculations which the available machine can do.

From the computer architecture viewpoint, almost all supercomputers made to date have been serial instruction-stream computers. The System/360 Model 91 is the largest single-instruction-stream computer presently announced. It is however, highly parallel in its internal structure. There are large numbers of buffers and arithmetic units, all capable of operating concurrently on the instruction and data streams. This internal overlapping is roughly equivalent to a parallelism of 8- to 10-fold over a simple nonoverlapped design. The important point is that it is still logically equivalent to a serial instruction-stream. The programmer or programming system need not consider the parallel nature of the machine directly.

Two other machine design philosophies have been seriously proposed which overtly act upon many sets of numbers at once. One design is that characterized by a single-instruction-issuing serial computer which causes the execution of a given instruction by many processing units, each of which operates on its own data. This design is frequently called a "network processor." The SOLOMON is the best known example.[S4] Computers of this design may perform with dramatic effect on calculations of relatively unbranching structure which can match the structure of the machine.

The second design has been called a "vector processor" or sometimes a "pipeline" machine. One such has been described by Senzig and Smith.[S1] As its name implies, it operates on vectors of numbers. The design has more flexibility than the network processor. Both of these designs are capable of very high effective MIPS rates (millions of instructions per second), because many of the instructions are occurring simultaneously. Both a network and a vector machine may form the sum $X_i = A_i + B_i$ for $1 \leq i \leq N$, where N is the number of arithmetic units. The network processor is usually considered with $N = 128$ to 1024, while in a vector machine the individual arithmetic units are not related to the number of components.

The speed of a serial machine can often be evaluated by consideration of representative kernels from typical programs. For a parallel machine, it may be necessary to look at the program as a whole. The network machine, in particular, is much more dependent on the nature of the problem and the skill of the programmer than is a conventional machine.

There is another more general class of parallel computers in which the separate arithmetic units have separate instruction-streams. These are usually referred to as "multiprocessor systems." However, Schwartz[S3] has recently introduced the term "Athene-type" for this class of machine (as contrasted to "Solomon-type"). The arithmetic registers in such a machine would appear to the user as multiple logically independent computers which could, however, all be put to work on parts of the same problem. It would include operations of the network or pipeline types as special cases. The difficulty usually comes in trying to prove that the combination will really produce more useful work done than N completely independent machines, or than one large machine processing the same amount of hardware as the N processors.

- *Relationship between storage and speed*

Perhaps the main property which differentiates a computer from any other piece of electronic equipment is the large number of internal storage states which it possesses. The growth of computer capability has been directly tied to the development of storage technology. This includes not just the high-speed core storage but also the peripheral bulk storage devices such as disks, drums and tapes.

One of the imponderables in analyzing computer performance has always been how to relate memory size to the speed of the arithmetic and logic of the system. Most knowledgeable people admit that storage is very important, but usually fall back to purely arithmetic comparisons when asked to compare two machines. Speed is not a substitute for large storage in general, but a large storage is frequently a substitute for speed. Knight[K1] has made an attempt to quantify this.

The basic effect can perhaps be summarized by this simple statement: "*Any* computer program can be made simpler by the addition of a larger core storage." The level of complexity of any given calculation is reduced by not having to worry about the blocking and ordering of data, or using overlaid programs, etc. Even very small programs which are not themselves changed will profit indirectly by the existence of more efficient compilers and monitors. This could be stated as an equally general statement: "An increase in arithmetic speed or computer parallelism without an increase in storage and other data flow capacity will result in an *increase* in program complexity."

- *Complexity threshold*

One should couple these statements with another basic rule, which can be stated as follows: Problem originators each have a "complexity threshold" beyond which a problem will not be attempted. This is so not because the problem cannot be solved nor that it cannot be properly modeled nor blocked into arrays, but simply because it has reached a certain point of complexity at which the problem originator decides not to attempt it and will do a simpler job or a different problem instead. This threshold, of course, is quite different for different people and at different times for the same person. However, it is certainly a very important phenomenon, because no matter how many programmers or assistants are put upon a large calculation, it is always a relatively small group, often a single individual—a senior scientist—who really originates the calculation and lays out its over-all structure. His threshold of complexity will govern the calculations which are attempted by his group. If we lower the level of complexity of every single problem, by providing a larger random access store, we may extend the frontier of complexity into a whole new range. There is a point where size is no longer simply a question of scaling problems, but where it really opens up a whole new class of calculations.

- *Flow of data between primary and secondary storage*

The total data storage required for large weather problems is so far beyond the internal storage capabilities presently available or foreseen in any supermachine that we will probably always have the requirement for keeping most of the problem data in secondary bulk storage devices. This places a serious requirement upon the data transmission between such secondary devices and the internal high-speed storage. This, in turn, is very heavily dependent upon the nature of the calculation and the organization of the problem to make effective use of external storage. If a problem is poorly organized, the computer speed may be limited solely by the access time of the first word of each storage block. An ideal problem organization would be one in

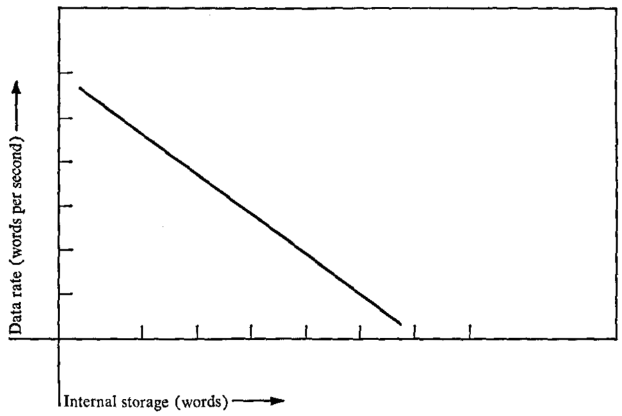
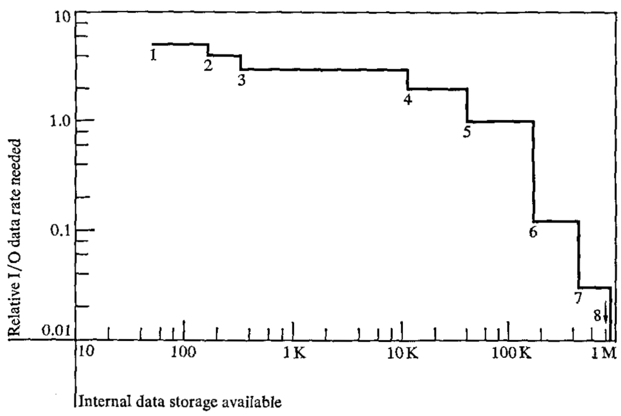


Figure 4 Relation between data rate and internal storage, plotted on log-log coordinates.

Figure 5 Data rates needed between primary and secondary storage as a function of size of internal data storage and problem formulation.

Example formulations of a 2° mesh problem:

1. 1 Point
2. 1 Vertical column
3. 5 Vertical columns
4. 1/4 Long.-3 lat.
5. 3 Latitude lines
6. 15 Latitude lines
7. Hemisphere
8. Entire sphere



which the utilization of a given number is directly proportional to the time which it spends in the internal storage. The data rate would be determined by the computer speed and the average amount of computation per data word. This can be represented in Fig. 4, which is a log-log plot. In practice, however, a problem has a number of discontinuous breaks related to natural block boundaries in the calculations. An example of such an analysis is given[K3] in Fig. 5.

• *The need for supercomputers*

It has been pointed out in a recent National Academy of Sciences report[N1] that there are a number of problems within the context of geophysical hydrodynamics which are now being worked on which may provide us with the basis for an estimate of the computation demands that can be anticipated within the next decade. The characteristic time and space scales range from those concerning turbulent exchange processes to those responsible for maintaining the large-scale ocean circulations.

It is of interest that, despite a span of simulated-experiment time ranging from 10 sec to 30 years (a ratio of 10^{-8}), the number of dependent variables generated during the experiment is roughly invariant, i.e., 10^{10}. Thus, the problems are of similar computational magnitude. It is also typical of and common to such hydrodynamical calculations that approximately 200 computer operations are required to generate one dependent variable at each new time step, so that somewhat in excess of 10^{12} computer operations are necessary to complete a single experiment.

We must define a "reasonable time" to spend in doing a numerical simulation experiment. The purpose of doing experiments is to study the nonlinear response of the numerical model to changes in parameters. By this process one develops the sought-after insight. Since a series of such experiments is necessary in order to span a physically realizable range of each of the parameters, we can arrive subjectively at a threshold of tolerance. Obviously 1,000 hours of machine time (one-half year of first-shift time) for a single experiment is intolerable. The threshold is more likely to be 100 hours (2.5 weeks of first-shift time), but a convenient time is closer to 10 hours. Operational considerations also impose similar limits. Hence, we need to be able to perform 10^{12} computer operations in 10 hours, or approximately one operation every 30 nsec, or 33 MIPS.

This figure exceeds by a factor of 50 the power of the fastest of the present computing systems.

Estimates of computer requirements in the future

• *Main properties contributing to speed and storage requirements*

A really accurate estimate of the computer speeds needed for future global weather calculations is probably not possible no matter how much detail one puts into the estimates. Even though the estimated average values may be fairly accurate, the over-all speed of a system often depends upon *combinations* of interactions. In other words, system performance depends on the distribution of the variables and their correlations as well as their mean values. An unfortunate coincidence of slow events can spoil the running time of the whole calculation, thus making accurate estimates very difficult. Even so, one can usually arrive at numbers bracketing the computer speeds and requirements that are reasonably safe as typical values.

For the global weather calculations which we are considering, the following appear to be the main properties which contribute to the computer speed and storage requirements. Some trade-off between them is possible:

(a) The complexity of the numerical model being used, including the number of physical effects being approximated and the difficulty of each algorithm used to compute them.
(b) The number of computer operations needed per program and their types, compared to those available on the computer under consideration.
(c) The total number of space points and time steps needed in a calculation.
(d) The desired ratio of speed of the computed solution of the atmospheric motion compared to real time.

There are other important factors which one could list, such as ease of programming, reliability, convenience of the display of results, etc. These are very important in building a satisfactory system but have only a secondary effect on the estimates of raw computer speed and storage capacity.

• *Complexity of the model*

The most important property that gives the character of the calculation is, of course, the numerical model of the weather problem being solved. It is made up of mathematical approximations to the physical quantities, the particular numerical algorithms being used to solve the mathematical equations, and the interactions allowed between the equations being solved.

It is customary to consider the dependent variables in a problem as either prognostic or diagnostic variables. There are five fundamental prognostic variables which must be evaluated and stored each time for each mesh point in the three-dimensional grid. They are: The three components of wind velocity, the temperature and the water vapor content. In addition there are two or more quantities which are two-dimensional in nature. These are kept in storage corresponding to the horizontal grid. An example would be the surface pressure.

In addition to the variables there are large numbers of constants. Usually no three-dimensional constants are carried but there can be a number of two-dimensional arrays of constants. Surface properties such as the differences between land and sea, the number of daylight hours, radiation properties, etc., can be kept as constants.

The diagnostic variables can vary in number considerably from one problem to another depending on what is being studied. Diagnostic variables are defined as quantities which can be determined completely from the prognostic variables, or as intermediate results from the computation of the prognostic variables. Most of them are not stored for the full three-dimensional or two-dimensional meshes, and many are not saved except as contributions to averages or limits.

Because of the number of special two-dimensional cases concerned with the surface conditions it is easier to describe the number of quantities for a vertical column of points rather than for each point in the three-dimensional mesh. On this basis, Leith's program[L1] stores 31 variables per horizontal point containing 6 vertical levels, and Smagorinsky[S2] uses 37 variables for 9 levels plus other diagnostic variables which are not computed every cycle.

Additional physical effects usually enter into the equations as diffusion terms in the energy equations. Since the vertical cases are handled separately, there will usually be two additional horizontal diffusion terms per point which must be stored. The inclusion of additional quantities and physical effects may or may not have an appreciable effect on the computing time. The storage may also be organized to avoid reading in slowly varying quantities more often than they are needed.

For five-day forecasts the effects of radiation are usually considered not very important. Radiation effects, however, are uniform, consistent and slowly varying, and so are cumulative. For forecasts longer than five days the radiation drives the atmosphere and must be included in the calculation. There are also cases of shorter duration where radiative effects can be dominant if other forces are weak.

Adequate treatment of radiation requires a suitable statistical treatment of local cloud cover. A reasonable guess by Kasahara is that this would add four dependent prognostic variables to the problem—three numbers representing the fractional grid coverage by each of three cloud types and a radiative heat source term—at every mesh point. The slow variation in time of these variables allows a time step of a few hours. Although the cloud and radiation coding will be lengthy and difficult, he estimates that the infrequency of calculation would result in only a few per cent increase in the over-all computation time. The doubling of the number of dependent variables, however, is certainly a significant consideration in terms of storage requirements and data transmission rates within the computer.

A two-fold increase in the linear resolution can cause an increase of up to a factor of 10 in total computation. However, by Smagorinsky's estimate an increase in the number of dependent variables by one (e.g., adding ozone in a general circulation model) will increase the number of calculations by only 20 per cent. Likewise, an increase of vertical resolution in the quasihorizontal models has a relatively small impact on computing time. Increases in physical complexity and fidelity in formulating the slow-acting processes (e. g., radiation, ocean coupling on the atmospheric general circulation) are also trivial in the total number of computations because they are done rather infrequently.

• *Number of computer operations executed*

For rough estimates the best parameter seems to be the number of instructions executed per mesh point per cycle. This is a complicated average to compute from a program, but it is easy to measure and very easy to use in making estimates. After examining several programs and making allowances for cases of Input/Output limitations, the following seem to be typical.

For a straightforward hydrodynamics program with no complications, approximately 300 instructions are executed per mesh point in a three-dimensional grid per time step. The most complicated programs with many complex physical effects included, run less than 3000 instructions executed per mesh point per time step. The programs themselves could be different in total size by more like a factor of 100. The important point is that the running time of a calculation in terms of computer operations per point per cycle does not seem to range over more than a factor of ten in practice.

The particular mixture of computer operations that are used to advance a typical mesh point are made up of the necessary floating-point arithmetic calculations plus the logical instructions to handle the branching and indexing. Even with the same numerical algorithm there are a number of ways in which individual equations may be programmed in machine language. A carefully written program will ordinarily use combinations of instructions which run the fastest. Thus there can be fairly strong feedback between the particular computer operation code characteristics and the precise way in which the numerical algorithm for the problem is set up. As a result, the product of the typical number of instructions executed per mesh point times the execution time of a typical instruction may give a total time which can differ considerably from the actual time needed to compute the mesh point. This is true because the two averages are not independent quantities. A well-coded program will tend to use more of the highest speed instructions. More details are given in Ref. K3.

• *Mapping problem*

One problem which is somewhat unique to the general circulation model is that of choosing a coordinate system to represent the surface of the earth in a convenient fashion in the computer. At first glance it seems that a trivial change of coordinates should solve it. However, like many such "trivial" matters it has caused a great deal of serious thought and planning, since the whole structure of the problem storage and the details of the difference equations depend on the form of mapping used.

Three types of mappings are used. The easiest to understand is the latitude-longitude grid such as that used by Leith and Mintz. It has the problem that the longitude lines get closer together as one goes toward the poles. Some means of cutting down on the number of computed zones as one approaches the poles is necessary. This creates artificial discontinuities in the mesh which can (and frequently does) cause problems.

The second method, which has been used by the Weather Bureau, the Air Force and the Fleet Numerical Weather Facility, is that of projecting the northern part of the globe onto a plane centered at the North Pole, then subdividing the plane in some regular way. The Southern Hemisphere, of course, can be done the same way. This method gives the simplest differencing scheme and gives best resolution in the upper and middle latitudes. It has difficulties in patching the two hemispheres together. For a detailed discussion of mapping, see Quarles and Spielberg[Q1].

The third method is that of using some algorithm for spacing points equally (almost) over the whole globe. Kurihara[K2] has described such a system which preserves latitude lines while spacing points almost equally. Smagorinsky's newer programs will use this approach since there is an appreciable storage saving promised.

In the estimates given later we compute the number of mesh points, simply assuming that the earth is spread out in a 360° by 180° rectangular plane. For example, the rectangular grid just mentioned gives 2592 horizontal mesh points for 5° mesh. The actual 40×40 mesh used by Smagorinsky results in 2514. Leith's zoning scheme results in 2036, because of the reduction in number of mesh points toward the poles. Kurihara's proposed equal spacing scheme would be able to get by with 1682 total points for the same approximate resolution. This is almost a factor of 1/3 reduction, although it is bought at the expense of longer interpolation calculations in the mapping equations. This is a good example of the trade-off between complexity and storage requirements.

• *Time and space resolution problem*

In the finite difference methods of solution which we have been considering, one of the main characteristics determining the running time of the problem and the accuracy of its results is the number of mesh points used in the calculation. Any of the models are capable of arbitrarily fine refinement in both space and time. Within the limitations of round-off error, the answers will improve, the finer the resolution. This creates a certain insatiability in the calculational requirements for such problems. In the Table 1 we have taken a number of typical mesh sizes which have either been done or discussed as future plans and have computed the total number of spatial mesh points.

In general, if a linear dimension is reduced, the time step must be reduced in direct proportion. This is strictly true for explicit hyperbolic-type equations. Normally, the weather circulation models are run at a finer time step than that which would be called for simply by the stability condition, mainly because of convenience. Also, the stability condition is usually finer in some parts of the mesh than others. Normally the entire mesh is carried at a single time step for logical convenience.

Table 1 The number of mesh points required by numerical weather problems as a function of resolution

Scale		Number of Points						
Spacing, degrees latitude	N[1]	Spacing, kilometers (at equator)	No. of horizontal mesh points[2]	No. of vertical levels	Total no. of mesh points	Time step,[3] min.	No. of points advanced to compute 1 hr. of real time	Typical programs
10°	18	1111.	648.	3	1.9×10^3	20	5.8×10^3	Programs of early 1950's. Present stream function codes.
8°	22	889.	1,000.	2	2.0×10^3	12	1.0×10^4	Mintz's 2-level code.
5°	36	556.	2,592.	6	1.5×10^4	10	9.3×10^4	Leith, Smagorinsky's $N = 20$ code
3°	60	333.	7,200.	9	6.5×10^4	6	6.5×10^5	Experimental calculations 1965-69
2°	90	222.	16,200.	12	1.9×10^5	4	2.9×10^6	Future operational region 1970's
1°	180	111.	64,800.	18	1.2×10^6	2	3.5×10^7	Future experimental region 1970's

[1] N equals twice the number of mesh spaces between the pole and equator.
[2] The number of horizontal points is computed for a 360° × 180° square grid. An actual problem can have 10 to 30% fewer horizontal points, depending on the mapping method used.
[3] The assumption is that each point is advanced every time step. In larger problems, the time step would probably be variable.

Since these problems can saturate any machine, the most serious question then becomes, What resolution is *really* needed? The detailed five-day forecast studies of Smagorinsky et al. were performed with a 40 × 40 horizontal mesh per hemisphere per altitude (approximately 5° mesh). It was found that the sub-mesh energy transport by eddy diffusion accounted for approximately 5 per cent of the total energy transport. Since the eddy diffusion in the atmosphere is not really known, the practical philosophy adopted is that the grid scale should be chosen small enough that a change by a factor of two or more in the eddy viscosity results in no significant change in the meteorologically significant large-scale effects. Ideally one would continue to halve the mesh interval until computed results become insensitive (although the physical model might become inappropriate in the process). Manabe feels an 80 × 80 horizontal mesh per hemisphere per altitude level is probably sufficient. This corresponds approximately to a two-degree mesh interval at the equator.

In order to represent the important temperature gradients in the ocean at least two vertical levels into the ocean are necessary. A total of 18 to 20 levels for both the ocean and the atmosphere is probably adequate. A time step of about 3 or 4 minutes would go with this resolution.

These estimates lead one to a spatial mesh of $80 \times 80 \times 20 = 1.3 \times 10^5$ points. Four dependent variables (eight with radiation and clouds) would be stored and at least five variables computed per mesh point each time cycle. A five-day forecast would require 2400 time steps, yielding a total of 3×10^8 mesh-point computations for a complete problem. If one wishes to perform this calculation at *100:1* times real time (1 hour 12 minutes for a 5-day forecast) then each mesh point must be advanced in an average of 14.4 μsec! If there are 3000 instructions to be executed to advance one mesh point, then the computer must execute instructions at the rate of 210 million instructions per second!

Of course, one can reduce this incredible figure by backing off on the above criteria. In particular the easiest to relax is the 100:1 speed criterion. A rate of *10:1* should *still* be quite reasonable for an operational system.

- *Ratio of the speed of calculation to real time*

The ratio of calculation speed to real time is one of the simplest numbers to state, yet next to the spatial resolution it is the most important in setting the computer requirements. Present models range from 1-to-1, that is, the calculation proceeding at the same rate as the actual weather, to perhaps 10-to-1. Older, greatly simplified models can be integrated on present-day computers at a much higher ratio—100-to-1 or more.

Tables 2 and 3 show examples of computer speed in terms of millions of instructions per second for different complexity models and different spatial resolutions. Some of these results are shown in the graphs where they become straight lines on log-log paper. The graphs are given for particular combinations of vertical resolution and time step corresponding to diagonal values in the table. (See Fig. 6).

These curves show perhaps better than any other means the true open-endedness of the computer requirements for general circulation models. They also show, however, that if one wishes to operate at speeds between *10:1* and *100:1*, with a 2° mesh resolution or better with a fairly complicated model, one must certainly get into the range of 20–200 million instructions per second executed.

Table 2 Approximate computer speeds required for a fast model computing at 10:1 times real time

Horizontal resolution. Spacing in degrees latitude	MIPS for various vertical levels: 3 Levels	6 Levels	9 Levels	12 Levels	18 Levels
10°	0.016	0.032	0.049	0.065	0.097
5°	.13	.26	.39	.52	.78
3°	.60	1.2	1.8	2.4	3.6
2°	2.0	4.0	6.1	8.1	12.
1°	16.2	32.	49.	65.	97.

Table 3 Approximate computer speeds required for a fast model computing at 100:1 times real time

Horizontal resolution. Spacing in degrees latitude	MIPS for various vertical levels: 3 Levels	6 Levels	9 Levels	12 Levels	18 Levels
10°	0.16	0.32	0.49	0.65	0.97
5°	1.3	2.6	3.9	5.2	7.8
3°	6.0	12.	18.	24.	36.
2°	20.	40.	61.	81.	120.
1°	162.	320.	490.	650.	970.

A "fast model" is defined as one requiring 1000 computer operations to advance one mesh point one time step on the average.

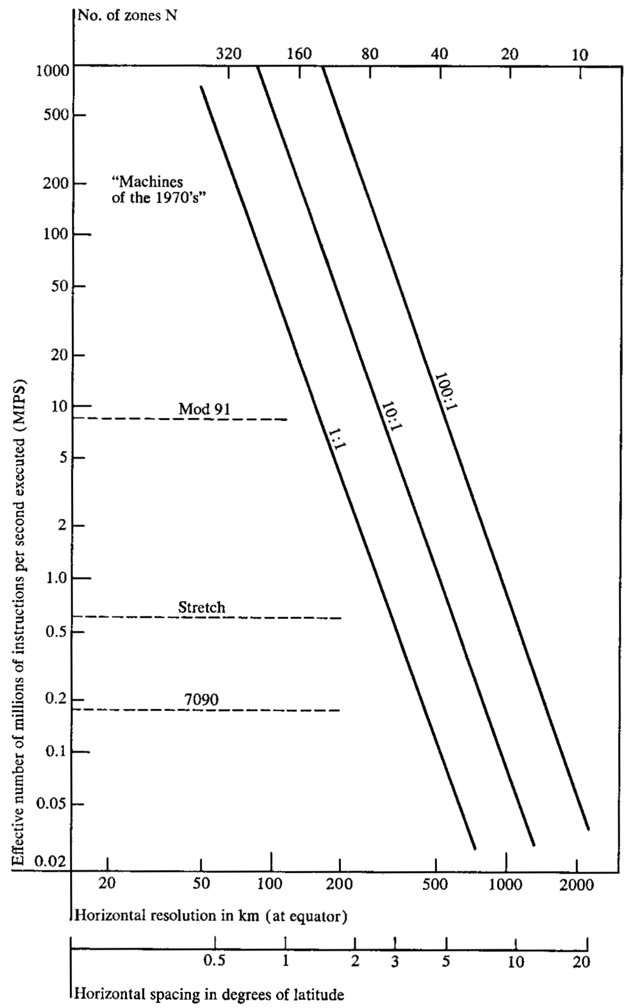


Figure 6 Computer speed vs. horizontal resolution on the globe for a complex model for various values of $t_{\text{real}}/t_{\text{calculated}}$.

Programming considerations

• *Higher-level languages and efficiency*

The question of programming-system efficiency is very serious for a large problem. The millions of instructions per second used up by an inefficient programming system are just as real as those used on a complicated physical calculation. In general, one is willing to spend more time in the programming of a large operational program that will be run many, many times than one is for a small program or a "one shot" experiment.

It is generally agreed today that it is unreasonable to expect people to do large-scale scientific programming in machine language. It is also agreed that they should not be penalized too heavily for using FORTRAN or PL/1. In practice, a combination of FORTRAN supplemented by key data handling subroutines written in machine language achieves most of the efficiency while retaining the ability to modify and improve the large production problem.

There should also perhaps be a set of special instructions which the average user could ignore but which could be used to streamline the inner loops of big programs. Examples of such instructions might be: multiplying by an integral power of a number; calculating A^x where the ranges of A and x are limited, say, between 0.1 and 1.0. Careful attention should be paid to the little "bottleneck" situations in weather calculations. Better diagnostic programs for locating bottlenecks automatically, and perhaps even eliminating them automatically in big calculations, would seem to be a very fruitful area for investigation.

The real need in a programming system is better coupling between the problem formulators and the computing equipment. The real goal should be to reduce the time from problem formulation to useful answers—not to increase "speed" or "turn-around time" or any of the other usual measures of computing, although they would also improve. Paper improvement in computer efficiency forced by rigid schedules and control program constraints is often exactly the wrong way to measure real progress on a scientific calculation.

• *Life and death of a large production program*
The numerical weather prediction programs which are presently being run are all the result of a long series of technical iterations involving the physical problem being solved, the equations arising from theoretical considerations, and experimental observations that are made of various physical phenomena. During the design iterations, new numerical differencing and approximation schemes are derived to solve the equations, experimental results are used to calibrate certain constants, then the calculations are refined, the equations improved, and so on. Some of the larger programs, such as Smagorinsky's which have been in existence for a number of years, have actually gone through several cycles of this physical-mathematical-experimental-programming iteration. One result of this history is that when new computing equipment becomes available, it is not as easy to take advantage of all of its power as might be expected, simply because one cannot break with the past completely in these practical design problems.

The life cycle of a large calculation usually starts from an experimental phase in which variations and improvements on past calculations are consolidated in a new code. It will become proved out throughout a number of exploratory calculations, comparisons against experimental results, previous calculations, etc., until it becomes accepted for its own right by the scientists who are making the practical decisions as a good tool for analysis. Often there may be two or three active versions of the same code in operation at one time, depending on the particular set of approximations or perhaps the machine upon which it is being run. It is thus hard to identify exactly what is being referred to in any given program.

When a production program is being replaced it does not really die suddenly. Usually what will happen is that a better approximation or a better program will gradually become used more and more and the older program will be relegated largely to the role of reruns of past examples. In cases where comparisons against earlier problems or designs are desired, the designers may wish to run them on the same old code as the originals were run. Also, calibration problems for the new code will be another type of use.

In a sense, any program which has at one time been used for heavy production becomes really completely obsolete only when people essentially have *forgotten* how to run it. This can happen because the machine on which it was written is no longer available, or the problem originators or the problem users have really forgotten how to run problems of interest on it. At this point the program is truly dead.

Example of the use of a numerical weather model
The use of electronic computers as tools to solve the data processing and physical analysis problems associated with daily operational forecasts is certainly understood and con-

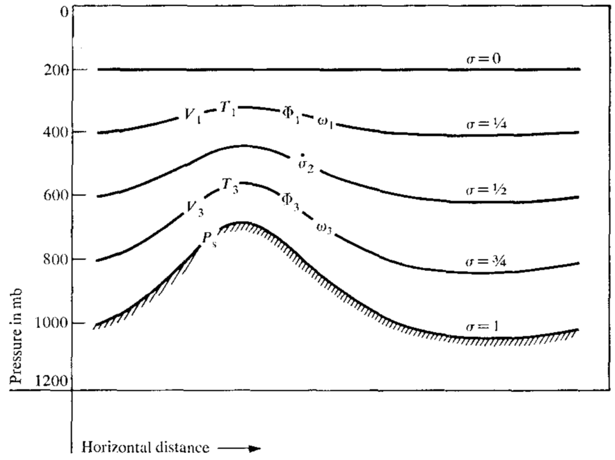


Figure 7 Configuration of vertical surfaces in the Mintz-Arakawa model showing how mountain ranges are described. Definition: $\sigma = (p - 200)/(p_s - 200)$. From Mintz.[M1]

sidered worthwhile by the public today. The role of general circulation calculations which use unreal initial conditions on an "artificial" world is probably less well understood.

Such calculations should be considered in the same spirit as laboratory experiments in which a model of a complex non-linear system is studied by carefully varying individual parameters to determine the basic behavior of the system. The goal is to isolate the key physical phenomena from among all the confusing details. As with any active field, there are many discussions and disagreements among numerical meteorologists concerning the relevancy of their various models and their results.

An excellent example of the use of a numerical model to study the cause of certain large global patterns is given by Mintz[M1] in his calculations demonstrating the importance of the Himalayas for the winter surface pressure pattern over Siberia (known as the "Siberian High" by meteorologists).

Prof. Mintz used a modified version of a coordinate system originally proposed by Phillips in 1957. The vertical dimension in the model is expressed in "sigma coordinates" as shown in Fig. 7. The ground is represented by $\sigma = 1$. It includes a smoothed representation of mountain ranges as shown. $\sigma = 0$ represents the 0.2 bar pressure surface and is the upper limit of the calculation. The horizontal grid is relatively coarse, being 9° of longitude at the equator and 7° of latitude, giving a total of 1000 horizontal mesh points. A time step of 12 minutes was used.

The calculation was started so that in the beginning the air was everywhere at rest, everywhere the same temperature (250°K), and everywhere having the same surface pressure of one atmosphere (1.013 bar). The sun's declination was set for the northern hemisphere winter. Initially the mountains were left out and the globe was a smooth sphere, although land, sea, and ice were differentiated.

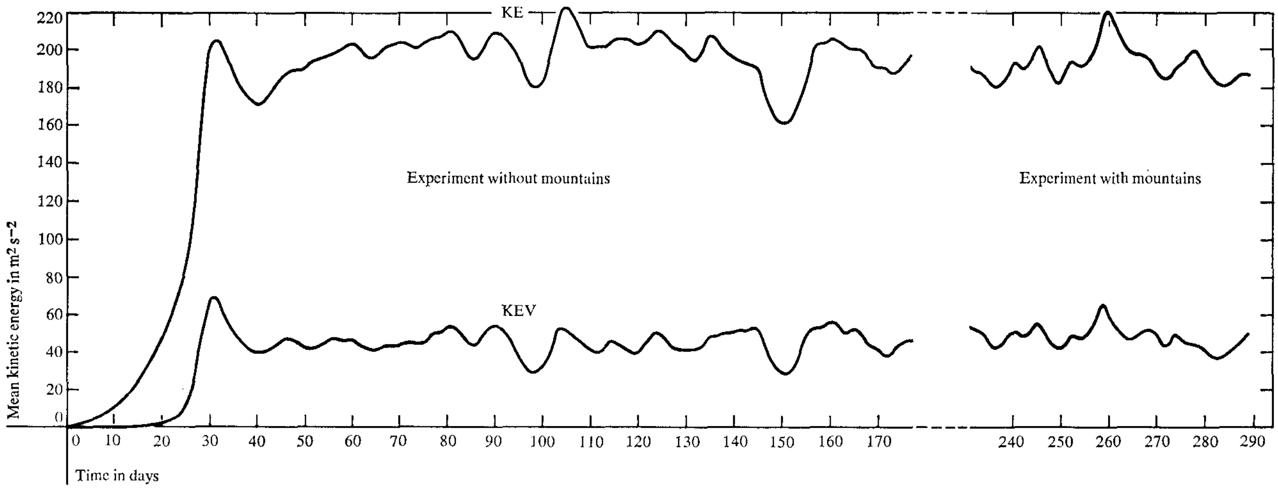


Figure 8 Mean kinetic energy of the entire atmosphere, as a function of time. KE is the global mean of the kinetic energy, per unit mass, of the total (vector) wind. KEV is the global mean of the kinetic energy of only the meridional (south to north) component of the wind. The left part of the figure is for the experiment without mountains. The right part is for a period many days after the large scale mountain systems had been added. From Mintz[M1].

From this initial state of rest the kinetic energies rose to their long-term steady state values in about 40 days. (See Fig. 8.) When this run reached a steady state, the mountain experiment was performed. The results showed that the Siberian High does not appear on the mean sea level pressure map when there is no Himalayan mountain chain. In Prof. Mintz's words: "After the experiment with the smooth, mountainless globe, I changed the surface data cards, at a given instant in the middle of a run, and in this way suddenly raised into place the large scale mountain systems of the Earth. As soon as this was done the air began to pour down the mountain sides (as would water down the sides of an island emerging from the sea), producing large gravity waves. After some days these large external gravity waves died out and only the familiar meteorological motions remained, as the synoptic charts will show. In terms of the global mean kinetic energies, the presence or absence of mountains seems to make little difference, as may be seen by comparing the right and left hand sides of Fig. 8."

The time-averaged results of the mountain experiment are shown in Fig. 9. The surface pressure reduced to sea-level shows a number of features which have been observed experimentally. These include the Siberian High; the Alaska-Yukon High; the Icelandic Low; the North Pacific Low; the North Australian Low, and others. Other features are not in agreement with experiment; particularly noticeable are the High over Greenland and the Low over the Great Lakes, the absence of observed low centers over South America and South Africa, etc.

For a numerical experiment, the results which disagree can be as significant as those which agree. The details for the above example are discussed in the original report.[M1] As is often the case in physical experiments, each numerical experiment often raises more questions than it answers. This phenomenon helps maintain the insatiability for computing time which characterizes meteorology.

Future outlook

Numerical weather prediction has progressed in the past 15 years from theoretical speculations to fully operational networks. The plans for the next ten years, which include the World Weather Watch, automated data collection and communications, promise to outshine the accomplishments of the past.

Concerning the numerical models to be used, the trend toward the finite difference solution of primitive equations will probably continue. Other new methods, such as those based on Fourier transforms, seem to offer little hope. One can expect that more and more detailed physics will be included. Coupled air-sea calculations will be commonly used. Clouds and moisture will be handled much more realistically. The emphasis will probably be on the incorporation of more satellite data, and other exotic measurements, directly into the models.

Fortunately the projected computer speeds, storage capacities and data rates for the 1970's seem to match the projected needs in terms of resolution and speed for the global weather problems of the same period. Perhaps this

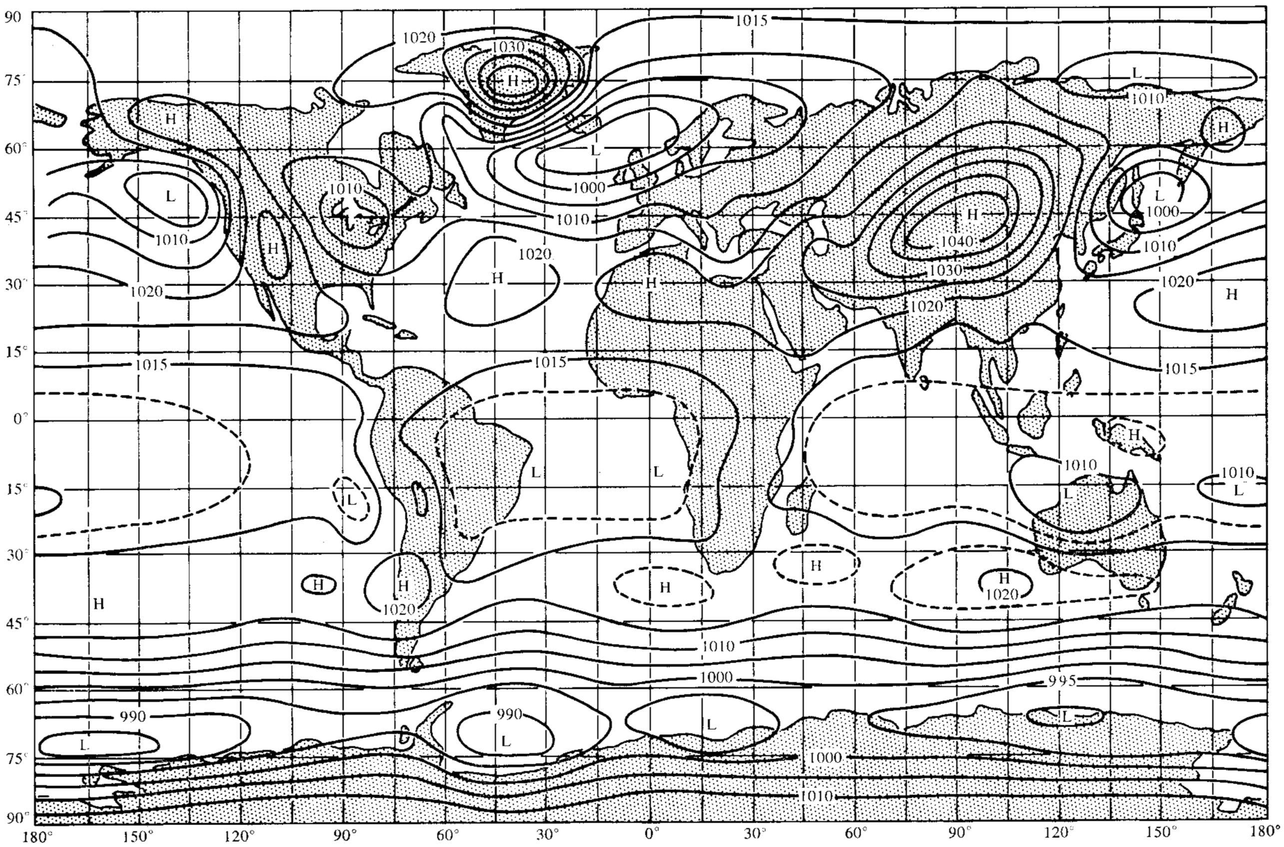


Figure 9 Some results of Mintz' numerical general calculation experiment with mountains. The curves are surface pressure isobars reduced to sea level, for Northern Hemisphere winter and Southern Hemisphere summer, in millibars. (The broken lines are intermediate 2½ mb isobars.) The curves represent the 30 day mean (from day 256 to 285) computed in the numerical experiment. From Mintz.[M1]

is another case of "feedback" as mentioned earlier, although it will take a tremendous effort from many groups and individuals to bring it to pass.

Acknowledgments

The author wishes to express his thanks to C. C. Leith of LRL, Livermore, Y. Mintz of UCLA, J. M. Gary of NCAR, and J. Smagorinsky of ESSA for the data, discussions, and encouragement they gave to this project.

Bibliography

A1. Arakawa, A., "Advanced Topics in Numerical Weather Prediction," Lecture Course Notes (as prepared by W. E. Langlois, UCLA, 1965.)

A2. Arakawa, A., "Computational Design for Long-Term Numerical Integration of the Equations of Fluid Motion: Two Dimensional Incompressible Flow," *Journal of Computational Physics* **1,** 119–143 (1966).

B1. Benton, George S., Chairman, "Interaction Between the Atmosphere and the Oceans," Publication 983, National Academy of Sciences, National Research Council, Washington, D. C., 1962.

C1. Charney, J. G., R. Fjørtoft, and J. von Neumann, "Numerical Integration of the Barotropic Vorticity Equations," Tellus **2,** No. 4, 237–254 (1950).

C2. Charney, J. G. and N. A. Phillips, "Numerical Integration of the Quasi-Geostrophic Equations for Barotropic and Simple Baroclinic Flows," *J. of Meteo.* **10,** No. 2, 71–99 (April, 1953).

C3. Cooley, J. W. and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. of Comp.* **19,** No. 90, 297–301 (April, 1965).

C4. Cressman, G. P. and H. A. Bedient, "An Experiment in Automatic Data Processing," *Monthly Weather Review* **85,** 333–340 (1957).

C5. Cressman, G. P., "Numerical Weather Prediction in Daily Use," *Science* **148,** 319–327 (1965).

G1. Goody, R. M., *Atmospheric Radiation*, Oxford Clarendon Press, 1964.

H1. Hockney, R. W., "A Fast Direct Solution of Poisson's Equation Using Fourier Analysis," Tech. Report CS6, Computer Science Division, Stanford University, April 14, 1964.

H2. Houghton, D., A. Kasahara and W. Washington, "Long-Term Integration of the Barotropic Equations in Eulerian Form," *Monthly Weather Review* **94,** No. 3 (March, 1966).

K1. Knight, Kenneth E., "Changes in Computer Performance," *Datamation* **12,** No. 9, 40 (1966).

K2. Kurihara, Y., "Numerical Integration of the Primitive Equations on a Spherical Grid," *Monthly Weather Review* **93,** No. 7, 399 (July, 1965).

K3. Kolsky, H. G., "Computer Requirements in Meteorology," IBM Palo Alto Scientific Center Report No. 38.002 (July, 1966).

L1. Leith, C., "Numerical Simulation of the Earth's Atmosphere," Lawrence Radiation Laboratory, UCRL 7986-T, 1964.
M1. Mintz, Y., "Very Long-Term Global Integration of the Primitive Equations of Atmospheric Motion," World Meteorological Organization Technical Note No. 66, pp. 141–167 (1964); to be republished as Amer. Meteor. Soc. Meteorological Monograph, vol. 8, No. 30 (1968).
N1. National Academy of Sciences, "Weather and Climate Modification—Problems and Prospects," report by Panel on Weather and Climate Modification, G. J. F. MacDonald, Chairman, NAS-NRC, Publ. 1350, Washington, D. C., 1966.
O1. O'Neil, H. M., "The Air Weather Service Six Level Model," Air Weather Service Tech. Report No. 188, p. 37 (November, 1966).
P1. Phillips, Norman A., "Numerical Weather Prediction," article in book: Alt, F. (editor), *Advances in Computers*, Vol. 1, Academic Press, 1960, p. 43ff.
P2. Phillips, Norman, "An Example of Nonlinear Computational Instability," *The Atmosphere and the Sea in Motion*, Oxford Press, 1959, p. 501–504.
Q1. Quarles, D. A. and K. Spielberg, "A Computer Model for Global Study of the General Circulation of the Atmosphere," *IBM Journal* **11,** No. 3, 312–336 (1967).
R1. Richardson, L. F., *Weather Prediction by Numerical Process*, Cambridge Univ. Press, London, 1922, 236 pp., reprinted by Dover.
S1. Senzig, D. N., and R. V. Smith, "Computer Organization for Array Processing," *Proc. Fall Joint Computer Conference*, Spartan Books, 1965.
S2. Smagorinsky, J., S. Manabe and J. L. Holloway, "Numerical Results from a Nine-Level General Circulation Model of the Atmosphere," *Monthly Weather Review* **93,** No. 12, 727–768 (December, 1965).
S3. Schwartz, J., "Large Parallel Computers," *J. of the ACM* **13,** No. 1, 25–32 (January, 1966).
S4. Slotnick, D., W. C. Borck and R. C. McReynolds, "The SOLOMON Computer," *Proc. Eastern Joint Computer Conference*, Spartan Books, 1962.
S5. Shuman, F. G., "Numerical methods in weather prediction," *Monthly Weather Review* **85,** 357–361, also 229–234 (1957).
S6. Stauffer, R. B., and T. H. Lewis, "MET-WATCH: a technique for processing and scanning meteorological data with a digital computer," *Proc. IFIP Congress* 62, Munich, 1962, p. 242–246.
T1. Thompson, Philip D., *Numerical Weather Analysis and Prediction*, The MacMillan Company, 1961.
W1. Wolff, P. M., L. P. Carstensen and T. Laevastu, "Analyses and forecasting of sea surface temperature," FNWF Tech. Note No. 8, 1965, 30 pp.

Received March 30, 1967.