

Exploring Global Climate: A One-Dimensional Energy-Balance Model*

James E. Burt
Geography Department
University of Wisconsin-Madison

1 Introduction

This manual describes a computer model of the earth's climate appropriate for use in introductory climatology courses. Sections 1 through 5 provide essential information about the program, and are intended for all users. Details of the model are presented in Section 6. Though somewhat more mathematical than other parts, it will often provide the explanation for results which seem counter-intuitive. The appendices cover common user questions and hardware issues.

The model is typical of other zonal energy balance models, in which the atmosphere is represented as a slab extending from pole to pole. The fundamental assumptions are that the state of the atmosphere can be summarized entirely by surface temperature (T), and that all energy fluxes can be computed on the basis of surface temperature. The model is time-dependent—that is, the governing equation is one that gives the *rate of change* of temperature as a function of energy gains and losses. Latitude (ϕ) is the only spatial variable present in the model, thus heat is transferred poleward within the atmosphere, but not longitudinally nor vertically. The behavior of the model is governed by about 10 parameters (CO₂ concentration, orbital tilt, surface albedo, etc.), all of which default to “normal” values corresponding to current conditions. Parameters are easily manipulated by the user, either singly or in groups. The model can be run in a seasonal mode, in an annual average mode, or in a global average mode, with albedo either prescribed or computed internally. Tables and graphs can be produced on a variety of graphic devices and printers.

*Version 1.5. The development of this software was supported in part by a grant from the IBM Corporation. The assistance of IBM is gratefully acknowledged, as is the help of the Faculty Support Center. Manual and software copyright ©James E. Burt, 1990-1993.

2 Governing Equation

The model is based on the premise that the rate of change of temperature is proportional to the difference between energy gained and lost. The constant of proportionality defines a heat capacity (C) which controls how rapidly a place responds to changes in the energy balance. In the model this heat capacity is expressed in terms of the depth of ocean assumed to be well-mixed and freely exchanging energy with the atmosphere. There are three processes by which energy can be gained or lost: 1) absorption of solar radiation, 2) emission of terrestrial radiation, and 3) horizontal transport. More precisely,

$$\left(\begin{array}{c} \text{Rate of Change} \\ \text{of Temperature} \end{array} \right) = \left(\frac{1}{\text{Heat Capacity}} \right) \times \left(\begin{array}{ccc} \text{Absorbed} & \text{Emitted} & \text{Net} \\ \text{Solar} & - \text{Longwave} & - \text{Heat} \\ \text{Radiation} & \text{Radiation} & \text{Export} \end{array} \right)$$

To understand the above equation, it must be appreciated that some variables are functions of position ϕ , others vary with time t , and others are functions of both position and time. Consider, for example, heat capacity C . In the model, this is prescribed (set by the user) as a function of latitude only. There is no temporal variation in C , nor does it depend in any way upon temperature T . We will therefore write $C(\phi)$ to indicate dependence on latitude alone. On the other hand, incident solar radiation varies both seasonally and with latitude. We will write $Q(\phi, t)$ to denote spatial and temporal variation. With this notation, each of the terms in the basic equation can be described.

- Heat Capacity [$C(\phi)$]. As mentioned above, $C(\phi)$ is prescribed as a function of latitude. This term influences the amplitude of the seasonal cycle and the rate at which the model approaches equilibrium. More than anything else, it is a function of land/ocean distribution and depth of mixed layer. Normal values have been chosen in such a way that the seasonal cycle at each latitude is close to what is observed on earth. When this

parameter is changed, all latitudes are adjusted upward or downward by the same fraction.

- Absorbed Solar Radiation [$Q_{abs}(\phi, t)$ or $Q_{abs}(T, \phi, t)$]. Absorbed solar radiation is found as the product of radiation reaching the surface and surface absorptivity. Absorptivity is calculated in one of two ways, depending on whether surface albedo is specified *a priori* or calculated from T . In the first case, we are given albedo $\alpha(\phi, t)$, and absorptivity is simply $[1 - \alpha(\phi, t)]$. Notice that the notation here indicates albedo is a function of position and season. This corresponds to default conditions, in which albedo is prescribed and held fixed during a run. The default (normal) albedo values have been assigned from satellite measurements. By using observed albedos, we obtain energy fluxes and surface temperatures which everywhere are reasonably close to those actually observed. Notice that by default α does not vary with temperature, and thus there is no temperature-albedo feedback. In the real world, of course, one might expect high-latitude warming to produce lower surface albedo.

As an alternative to prescribed albedo, the program will calculate albedo from the local temperature. In this case absorptivity becomes $[1 - \alpha(T)]$. The program uses a simple equation in which albedo decreases sharply as temperature increases through a transition zone near 0°C . At temperatures far from 0°C , the albedo-temperature curve is nearly flat (see Figure 1). It is important to understand that when albedo feedback is turned on, no information other than T is used in computing albedo. In particular, no knowledge of the local land/ocean ratio or elevation is exploited. In addition, the program assumes that albedo responds instantaneously to T . Because of these and other limitations, unrealistic temperatures will occasionally appear, especially during high latitude summers.

The program allows the user to vary the slope of the albedo-temperature curve, and thereby change the strength of albedo feedback. One will typically use negative slopes, which creates a positive feedback loop:

$$\text{higher } T \rightarrow \text{lower } \alpha \rightarrow \text{more solar absorbed} \rightarrow \text{higher } T$$

When the feedback is strengthened by making the slope more negative, the transition zone becomes narrower, and the model climate becomes less stable. Depending on the values of other parameters, one can quite easily create a world whose climate slips off to completely ice-covered or ice-free. Here (as elsewhere) extreme caution must be used when interpreting model output.

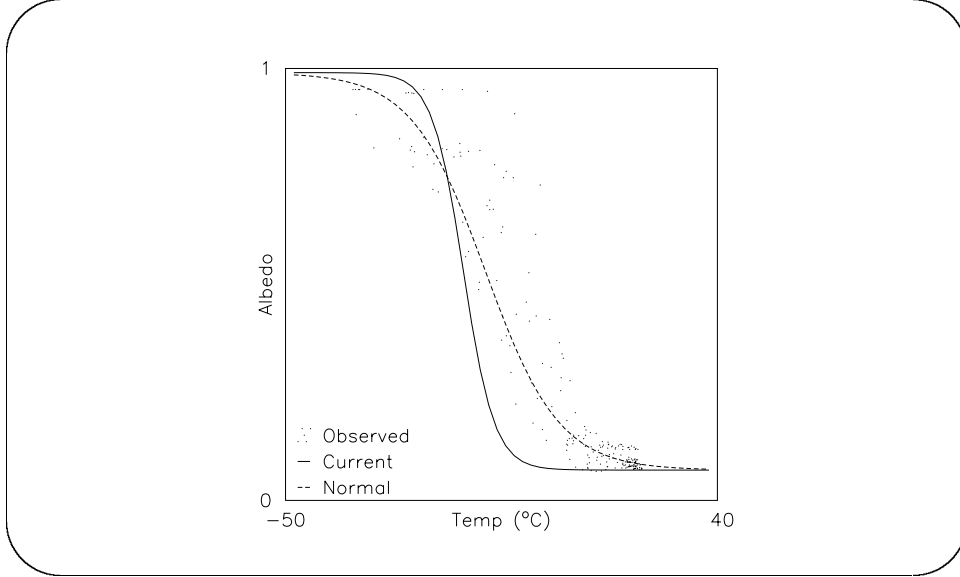


Figure 1: A panel taken from the albedo feedback screen. The user is requesting lower global albedo together with strong feedback—note the narrow zone of transition temperatures.

Solar radiation at the surface $Q(\phi, t)$ is the product of radiation reaching the top of the atmosphere $Q_o(\phi, t)$ and atmospheric transmission Ψ . The transmission coefficient varies with atmospheric transmissivity τ and optical air mass m according to

$$\Psi = \tau^{m(\phi, t)}$$

Because air mass varies with solar position, it is a function of both latitude and time. Atmospheric transmissivity can be adjusted upward or downward, corresponding to changing particulate loads and other factors influencing atmospheric reflection of solar radiation. Radiation on top of the atmosphere varies with the solar constant S , solar zenith angle $Z(\phi, t)$, and earth-sun distance $d(t)$. The user specifies S in the usual way as the flux density (in W/m^2) at the annual average earth-sun distance (\bar{d}). Letting $V(t)$ be the normalized earth-sun distance, $d(t)/\bar{d}$, we get

$$Q(\phi, t) = \Psi Q_o(\phi, t) = \Psi S V^{-2}(t) \cos Z(\phi, t).$$

The earth's orbital elements (eccentricity, obliquity, date of perihelion) modulate zenith angle, and have been set such that normal conditions correspond to present-day values. The orbital configuration can be changed, either by direct manipulation of the three parameters, or by changing the year of simulation. When a new year is requested, the program determines new values for the three orbital parameters using an approximate solution to the laws of planetary motion. Collecting the terms influencing Q_{abs} , we use one of the following formulas:

$$\begin{aligned}
 Q_{abs}(\phi, t) &= [1 - \alpha(\phi, t)] Q(\phi, t) && \text{(no albedo feedback)} \\
 &\text{or} \\
 Q_{abs}(T, \phi, t) &= [1 - \alpha(T)] Q(\phi, t) && \text{(with albedo feedback)}
 \end{aligned}$$

- Longwave Radiation [$I(T)$]. Infrared emission from the planet I is a simple grey-body formulation $I(T) = \epsilon(\text{CO}_2)\sigma T^4$. Because I depends strongly on T , this term represents a powerful negative feedback. Planetary emissivity $\epsilon(\text{CO}_2)$ is adjusted on the basis of atmospheric CO_2 concentration using the results of a detailed radiative-convective model.
- Heat Export [$F(T, \phi)$]. When more energy leaves than arrives by horizontal transport, there is net heat export due to horizontal motion. In the model, northward heat flux F is assumed to be proportional to the local temperature gradient:

$$F(T, \phi) = -\kappa(\phi) \frac{\partial T}{\partial \phi}$$

In this equation, $\kappa(\phi)$ is an eddy diffusion coefficient, representing the vigor of poleward motions. Larger values imply more mixing by the combined effects of atmospheric and oceanic processes. Net export is found from the gradient of F . Default values of $\kappa(\phi)$ have been assigned by requiring that the model reproduce observed annual average meridional transport. Note that $\kappa(\phi)$ does not vary seasonally, so that seasonal variations in heat transport arise entirely from seasonal variations in the local temperature gradient.

3 Solution Method

The governing equation is a nonlinear partial differential equation in latitude and time. The computer program does not attempt an analytical, or closed form

solution. In other words, it does not employ the sort of symbolic manipulations that might be learned in a calculus course. Instead the program creates an algebraic problem which approximates the true problem, and proceeds to solve the approximate problem. The program will solve this approximate problem to practically any required level of accuracy (up to about 14 significant digits). However, given that one is operating on an approximation, there is no point in demanding highly accurate solutions.

The approximation process is as follows. The planet is divided into 10° latitude bands, and a separate energy balance equation is written for each. So long as κ is not zero, these equations are coupled to one another, and form a system of ordinary differential equations in time. The system is solved numerically, with the solution marching forward in time from some set of initial values. The system is always initialized using current observed temperatures. The time step used by the program is adjusted internally, so that numerical errors are held below a small value set by the user. (Smaller error tolerances make for longer running times.)

Because of the strong negative feedback caused by temperature-infrared coupling, it is almost always the case that the solution moves quickly from an initial state to some equilibrium climate. When a seasonal cycle is present in Q and α , there will be a seasonal cycle in T even at equilibrium. Nevertheless, corresponding days in successive years will be virtually identical (within the error tolerance). Exceptions to this trend toward equilibrium will occur only if the user has set a parameter to some pathological value (e.g., negative C).

4 Running the Model

The model is usually started by typing “ebmodel”. (Depending on the local computing environment, some instructors might install a slightly different startup mechanism.) Regardless of how the program is invoked, users are presented initially with a menu showing the current model setup (see Figure 2). By moving the cursor around the menu, the user can change parameter values or branch to other parts of the program. Branches are shown in the bottom row of the menu. They are selected by placing the cursor on a choice and striking the “Enter” key. Alternatively, one can branch immediately to another module using shortcut keys

shown in capital letters (e. g., “R” begins a run, “X” exits the program—see Figure 3).

The main menu (Figure 2) indicates that albedo feedback is assigned a default value of zero. In this case albedo is fixed throughout the run. By entering a nonzero value, the user indicates that albedo should be a function of T . As mentioned earlier, the program uses a simple function in which albedo changes rapidly across a narrow zone of transition or “melting” temperatures. The strength of feedback is specified by giving the slope of this curve at the midpoint of the melting zone. (Because albedo is dimensionless, the slope has units of $1/^\circ\text{C}$.) The slope may be set to either a positive or negative value. A value of about $-0.13/^\circ\text{C}$ provides a reasonable fit to observed temperatures and albedo. Increasing the slope away from zero in either direction will have the effect of narrowing the melting zone, whereas decreasing the slope will widen the zone. If the slope is decreased to zero, one again has no feedback. When albedo feedback is changed, the user is presented with a graph showing observed (satellite) albedos, a normal $\alpha(T)$ curve (slope $-0.13/^\circ\text{C}$), and the $\alpha(T)$ curve one associated with the new albedo feedback value (Figure 1). The user may adjust the curve by means of the arrow keys, changing global albedo and/or feedback. The “X” key returns the user to the main menu.

Note that the top row of the main menu establishes any spatial or temporal averaging to be used during the simulation. As mentioned in the introduction, the model can be run in a seasonal mode, with no averaging at all. It can also be run in an annual mode, where incident solar radiation and albedo at each latitude are averaged over the year (no seasonal cycle). Finally, the model can be run in a global mode, with variables averaged spatially as well (no latitudinal or temporal variation). Annual and global runs are much faster than seasonal runs, because large time steps are possible as the solution approaches equilibrium.

The idea behind averaging is that one will make a seasonal run only when it is important to see details of the seasonal cycle. Therefore, for a fixed set of parameters, we want the three types of run to correspond as closely as possible to one another. This requires that average albedo be constructed by weighting monthly albedo according to incident solar radiation. This will insure that total absorbed solar radiation is unchanged by the averaging process, and thus planetary heating will be the same for all three types of run. Of course, because of nonlinearities in the model, correspondence in model forcing doesn’t ensure correspondence in model response, and small differences in global equilibrium

Type of Run: Annual Run		
Parameter Name	Current Setting	Change from Normal
Solar Constant	1367.0 W/m ²	0.0%
Atmospheric Transmissivity	0.89	0.0%
Carbon Dioxide Concentration	350.0 ppm	0.0%
Global Albedo	0.121	0.0%
Albedo Feedback	0.0 /°C	0.0/°C
Eddy Diffusion Factor	1.0	0.0%
Ocean Mixed Layer Depth	75.0 meters	0.0%
Year	1990.0 AD	0.0 Years
Eccentricity	0.017	0.0%
Day of Perihelion	2.8	0.0 Days
Obliquity	23.45°	0.00°

Run	eXit	Graphs	Tables	reSet	Info
-----	------	--------	--------	-------	------

Figure 2: A panel taken from the main menu. All parameters are shown at standard (normal) values, corresponding to present conditions.

temperature will emerge.

When albedo feedback is present, there is no averaging of albedo; it is simply calculated from the temperature at each point in space and time. Because there is no way to force the three types of run to have identical absorbed radiation, albedo feedback will lead to larger differences between the three types of run. In addition, as the perceptive reader will have realized, with albedo feedback turned on there is no guarantee that the planetary albedo requested on the main menu is in fact achieved. Though one could iterate on the nominal albedo, to do so would require multiple runs. Instead the program uses an $\alpha(T)$ function which produces the requested global radiation-weighted albedo under normal conditions. Whenever computed temperatures depart from normal, the actual global albedo will depart from what is requested.

When satisfied with the setup, the user begins a simulation by hitting “R” or by placing the cursor on RUN and striking the “Enter” key. The program first calculates incident solar radiation on the basis of the current orbital configuration and solar parameter. The user is then asked to specify the total length of run and how frequently a solution should be printed on the screen. Lastly, the user supplies a value for the internal error tolerance. (Default values are suggested for all of these).

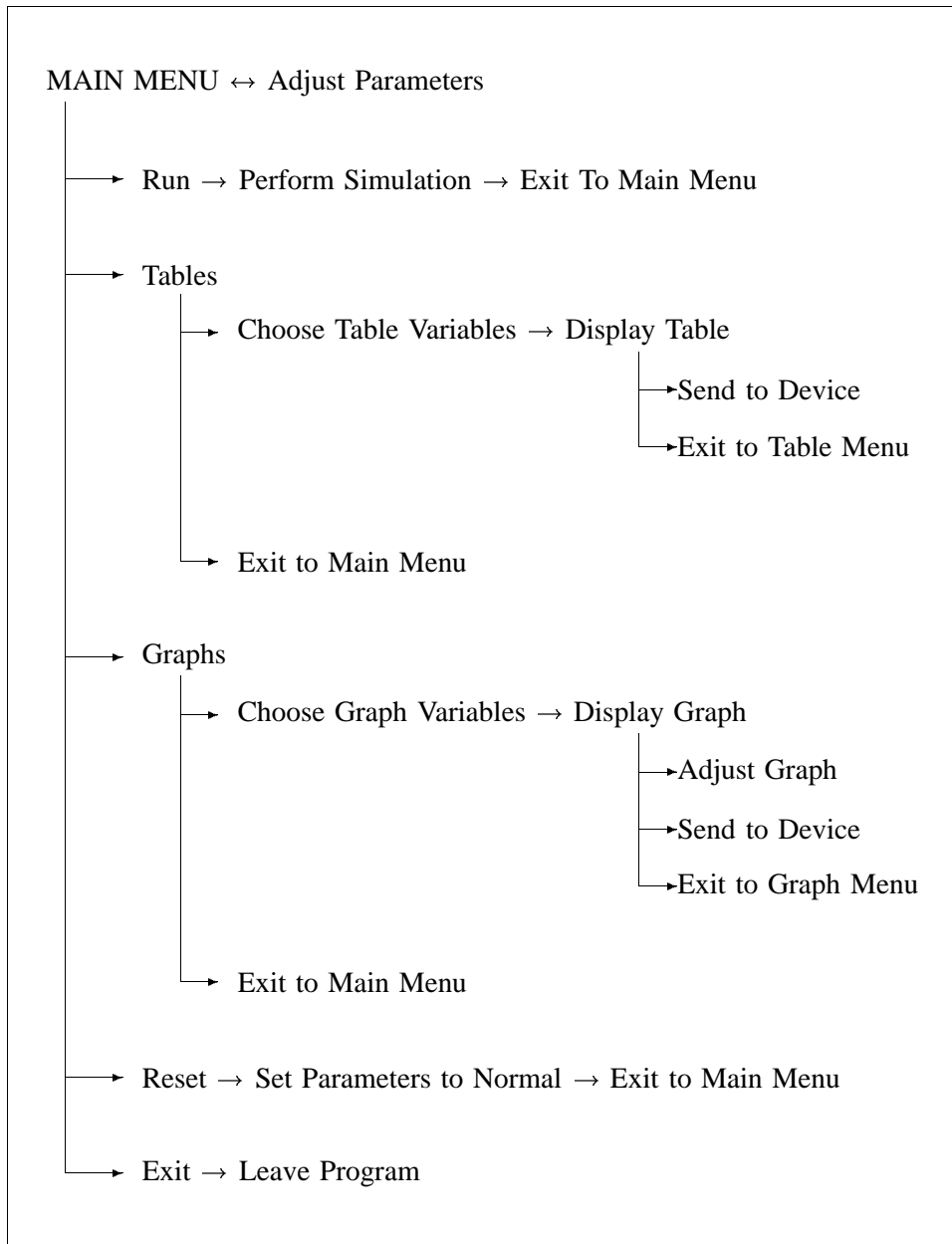


Figure 3: Flow diagram of program. Vertical lines indicate choices, horizontal lines indicate sequential operations. Parameters are adjusted in the main menu. From the main menu, one either performs a simulation run, produces tables or graphs, resets parameters to normal values, or leaves the program.

All simulations begin with temperatures assigned to their normal values. As the run proceeds, a table is displayed on the screen showing the evolution of planetary temperature and several other quantities useful for determining when equilibrium has been reached. A run can be interrupted by hitting the key labeled “Esc”. This causes the simulation to terminate at the next convenient point in time.

5 Model Output

The main menu provides options for graphical and tabular output of model variables (bottom row, Figure 2). These options lead to two very similar submenus, one for making tables, one for making graphs (see Figure 3). In both cases normal values are held in memory along with results from the most recent global, annual and seasonal runs. For comparative purposes, normal values are always displayed together with simulation results. In the case of an annual run, results are available by latitude zone. Seasonal results are available as monthly values by latitude, annual averages by latitude, and as zonal averages by month. Obviously, a global run can provide only global average values. Please note that until a simulation is performed, no simulation results are available. Tables will display “N.A.” to indicate unavailable entries; graphs will show only normal conditions when simulation data are not available.

5.1 Graphic Output

All plots are x-y graphs with either latitude or month on the horizontal axis and some model variable like T on the vertical axis. The graph menu is used to select the type of graph, after which the program enters its graphics mode. A graph is immediately drawn on the screen with a solid line showing simulation results (if present) and a dotted line showing normal values. Many commands are available at this point for changing colors, size, font, and so forth. (A list of all available commands can be obtained by issuing the *help* command.) All graphics instructions are entered by typing the command name followed by the “Enter” key (↵). For example, to change the plot title, one might type “title 10000 Years BP ↵”.

Most graphics commands can be safely ignored, but two deserve special mention: *x* (or *exit*), and *plot*. The first of these simply returns the user to the plot menu, where a new variable or type of graph might be selected. The *plot* command is used to produce hardcopy output. This command creates instructions which will drive one of two popular graphics devices. Each time the *plot* instruction is issued, the current plot is sent to an external device or file. If data are sent to a device, the printer/plotter must be online and ready to receive data. If data are sent to a file, it becomes the responsibility of the user to route all plot files to the appropriate printer/plotter after leaving the climate model program. How this is accomplished differs from site to site, and is not discussed here.

5.2 Tabular Output

As mentioned above, the table menu controls the contents of printed output just as the plot menu does for graphic output. Tabular output is always displayed first on the screen. After each table is displayed, the user may elect to print the table (type “P” to print). Normally, tables are sent to a default printer port established when the program starts. This will usually be a printer connected to your PC. If desired, tables can be sent to an alternative port or file. To change the port, select “New Port” from the table menu. You will be prompted for a file or device name. For example, to save tables on a diskette, you might enter “a:solar.txt” for a file name. As new data is sent to a file, it is simply appended to whatever has been sent previously.

Most of the tables display annual or global averages along with monthly or zonal values. Please note that global averages are always constructed by weighting the zonal values by surface area, and are not simple arithmetic averages. As mentioned above, albedo is also weighted by incident solar radiation, so its annual average is not the arithmetic average of its monthly values.

6 Model Details

The model is similar to the 1-dimensional zonally-averaged energy balance models of Schneider and Gal-Chen (1976), Schneider and Thompson (1979), and

Thompson and Schneider (1979). Letting a be the radius of the earth and \mathbf{div} be the spherical divergence operator, the basic equations are:

$$\begin{aligned}
C(\phi) \frac{\partial T}{\partial t} &= Q_{abs}(T, \phi, t) - I(T) - \mathbf{div} F(T, \phi) & (1) \\
Q_{abs}(T, \phi, t) &= (1 - \alpha) \tau^{m(\phi, t)} S V^{-2}(t) \cos Z(\phi, t) \\
\alpha &= \begin{cases} \alpha(\phi, t) & \text{no feedback} \\ \alpha(T) & \text{with feedback} \end{cases} \\
I(T) &= \epsilon(\text{CO}_2) \sigma T^4 \\
\mathbf{div} F(T, \phi) &= \frac{1}{\cos \phi} \frac{\partial}{\partial \phi} \left[\frac{\cos \phi}{a} \left(-\kappa(\phi) \frac{\partial T}{\partial \phi} \right) \right]
\end{aligned}$$

Atmospheric transmissivity τ includes the effects of all processes that deplete the solar beam. It is defined here as the proportion of extraterrestrial solar radiation that reaches the ground under a vertical sun. The model assumes uniform transmissivity (τ is a constant), but adjusts for variations in path length (yielding nonuniform transmission) using Atwater and Ball's (1978) formula for optical air mass:

$$m(Z) = \frac{35}{\sqrt{1 + 1224 \cos^2 Z(\phi, t)}}$$

In determining solar radiation, orbital parameters are either specified directly by the user or calculated from the year of simulation. In the latter case, the astronomical formulas of Berger (1977, 1978) are used to calculate orbital eccentricity e , obliquity ω , and day of perihelion. To determine the sun's position, we follow Hopkins (1985) in first expressing a day of the year as the mean anomaly M , which is simply its angular departure from the day of perihelion. We then compute the radius vector V and solar declination δ from the true anomaly ν

$$\begin{aligned}
\nu &\approx M + \left(2e - \frac{e^3}{4}\right) \sin M + \frac{5}{4}e^2 \sin 2M + \frac{13}{12}e^3 \sin 3M \\
V &= \frac{1 - e^2}{1 + e \cos \nu} \\
\delta &= \arcsin [\sin \omega \sin(\nu - \nu_{ve})]
\end{aligned}$$

where ν_{ve} is the true anomaly on the vernal equinox. The only time-varying processes in the model are seasonal, thus it suffices to compute solar flux just often

enough to resolve the seasonal cycle. Accordingly, the program computes solar flux once per month at each latitude. When needed, solar flux on intermediate days is interpolated using cubic splines, one for each latitude. Because there is no diurnal cycle, for each month we have to calculate only mean flux density \bar{Q} over the day. If H is the hour angle of sunset, we seek

$$\bar{Q} = 2 S V^{-2} \int_0^H \tau^{m(Z)} \cos Z dh$$

where

$$H = \arccos \left[\frac{-\sin \delta \sin \phi}{\cos \delta \cos \phi} \right]$$

$$\cos Z = \sin \phi \sin \delta + \cos \phi \cos \delta \cos h$$

(See for example, Paltridge and Platt, 1976; or Sellers, 1965). For each latitude-month pair, the program integrates over the half-day using Simpson's rule. Annual average albedo for a zone, α_j , is obtained from

$$\alpha_j = \frac{\sum_{i=1}^{12} \alpha_{ij} \bar{Q}_{ij}}{\sum_{i=1}^{12} \bar{Q}_{ij}}$$

where α_{ij} and \bar{Q}_{ij} are albedo and incident solar radiation for month i , zone j . Likewise, given the area of a zone, A_j , global albedo is found from

$$\alpha_G = \frac{\sum_{i=1}^{12} \sum_{j=1}^{18} \alpha_{ij} \bar{Q}_{ij} A_j}{\sum_{i=1}^{12} \sum_{j=1}^{18} \bar{Q}_{ij} A_j}$$

When fixed albedo is used, normal values α_{ij} are assigned from Robock's (1980) monthly climatology. When albedo feedback is nonzero, albedo is calculated from a logistic equation in T :

$$\alpha(T) = \alpha_{min} + \frac{\alpha_{max} - \alpha_{min}}{1 + e^{\gamma(T_0 - T)}}$$

where γ is the albedo feedback value specified by the user and α_{min} and α_{max} are fixed. The inflection point T_0 defines the melting zone "midpoint", where

the slope of the curve is γ . Thus one may interpret γ as the slope through the melting zone. (It is easy to show that $|\gamma|$ is also the steepest slope.) Notice that if γ alone changes, global albedo changes. In order to permit γ and α_G to be changed more or less independently of one another, the program finds a new T_0 whenever γ is changed. More specifically, given α_G and γ , the program searches for a T_0 which will allow the following equation to be satisfied:

$$\alpha_G = \frac{\sum_{i=1}^{12} \sum_{j=1}^{18} \alpha(T_{ij}) \bar{Q}_{ij} A_j}{\sum_{i=1}^{12} \sum_{j=1}^{18} \bar{Q}_{ij} A_j}$$

where T_{ij} is the normal temperature for month i , zone j . This lets us obtain an $\alpha(T)$ function which produces the requested α_G and γ under normal temperatures. Of course, because the final computed temperatures will differ from normal, the final α_G will differ from what was requested. Equally obviously, there are (α_G, γ) combinations for which no T_0 will satisfy the above equation. In those cases the user's α_G will be overwritten to a new value which is as close as possible to what was requested. This will happen only for unusual (α_G, γ) combinations, and is not believed to be unduly constraining.

Planetary emissivity is calculated from CO₂ concentration relative to its standard value (350 ppm), as described in Burt and Vecelli (1979). In that paper, Ramanathan's (1976) radiative-convective model was run under increasing CO₂ values at fixed temperature, giving decreasing amounts of longwave flux to space. From the flux values one can arrive at effective emissivity as a function of CO₂. Letting X be $\ln(\text{CO}_2/350)$, one obtains

$$\epsilon(\text{CO}_2) = 1 + 2(-3.825X + 0.43878X^2)/120$$

Standard diffusion coefficients were determined from $\kappa = -F(\phi)/(\partial T/\partial \phi)$, using values of T and F presented in Thompson and Schneider (1979). Note that unlike those authors, we do not assume κ varies with the local temperature gradient. The remaining parameters, τ and $C(\phi)$ were assigned by tuning the model to observed temperatures. The former was chosen by making a global run and requiring that computed global annual temperature agree as closely as possible with the current observed value. $C(\phi)$ was assigned by matching the amplitude of the annual temperature cycle at each latitude. Integration of Equation (1) is accomplished by the method of lines. That is, $\partial T/\partial \phi$ is approximated by finite differencing, resulting in a system of ordinary differential equations in time. These are solved using a predictor-corrector method whose time step is

dynamically adjusted so that local integration error remains within the specified tolerance. For annual and global runs, the allowable time step is constrained mainly by how far the solution happens to be from equilibrium, with the result that internal time steps grow progressively longer as equilibrium is approached. In most cases time steps several decades in length become possible near the end of a run. On the other hand, because activity of the solution is governed mainly by temporal variations in solar radiation, seasonal runs demand smaller and nearly constant time steps, typically on the order of 2-3 weeks. If albedo feedback is present, still smaller time steps are needed, often as small as 2-3 days.

7 References

Atwater, M. A. and J. T. Ball, 1978: A Numerical Solar Radiation Model Based on Standard Meteorological Observations. *Solar Energy*, 21, 163-70.

Berger, A. L., 1977: Long-term Variation of the Earth's Orbital Elements. *Celestial Mechanics*, 15, 53-74. Berger, A.L., 1978: Long-term Variations of Caloric Insolation Resulting from Earth's Orbital Elements. *Quaternary Research*, 9, 139-167.

Burt, J.E., and J. Viecelli, 1979: Magnitude of Ocean-Temperature Feedback Effects in a Coupled Carbon Budget-Energy Balance Climate Model for the Period 1800–2100. *Climatic Change*, 2, 267-274.

Gal-Chen, T., and S. H. Schneider, 1976: Energy Balance Climate Modeling: Comparison of Radiative and Dynamic Feedback Mechanisms. *Tellus*, 28, 108-121.

Hopkins, E., 1985: Analysis of Astronomically-Induced Monthly and Zonally Averaged Extra-Atmospheric Irradiance Variations for the Earth Over the Last 500,000 Years. Unpublished Ph.D. dissertation, University of Wisconsin-Madison, 919 pp.

Paltridge, and Platt, 1976. *Radiative Processes in Climatology and Meteorology*. Amsterdam: Elsevier, 318 pp.

Ramanathan, V., 1976: Radiation Transfer within the Earth's Troposphere and Stratosphere: A Simplified Radiative-Convective Model. *Journal Atmospheric Sciences*, 33, 1330-1346.

Robock, A., 1980: The Seasonal Cycle of Snow Cover, Sea Ice, and Surface Albedo. *Monthly Weather Review*, 108, 267-285.

Sellers, W. D., 1965: *Physical Climatology*. Chicago: University of Chicago Press, 272 pp.

Schneider, S. H., and S. L. Thompson, 1979: Ice Ages and Orbital Variation: Some Simple Theory and Modeling. *Quaternary Research*, 12, 188-203.

Thompson, S. L., and S. H. Schneider, 1979: A Seasonal Zonal Energy Balance Climate Model With an Interactive Lower Layer. *Journal Geophysical Research*, 84, 2401-2414.

A Frequently Asked Questions

1. *When I graph results, why do I see only normal values on the plot (a dashed red line)?* This happens if there are no simulation results available. Perhaps you made an annual run, but are trying to plot results from a seasonal run.

2. *When I make a graph, why do I see only the new results (solid green line)? Where are the normal values?* The program first draws normal values, then new values. Your simulation results are so close to normal, they appear on top of the normal curve. Make a larger change in the parameters, and the curves will diverge.

3. *Is there a way to print tables to a file, rather than have them go directly to the printer?* Yes. From the table menu, redirect output to a file using the "New Port" command.

4. *Why don't saved tables appear on my diskette? I'm sure I redirected output from the screen to a file.* You probably aren't giving the disk drive name in the file specification. If your diskette is in drive A, try "A:filename". If you just

enter “filename”, the file is saved on the current device, which might be some other drive.

5. *I can print tables with no problem, but when I look at them later, I can't remember what parameters were used in the run. Is there some way to put parameter values on the table?* No, but you can print the parameter table itself. Maybe you should do that before you print any results from a run.

6. *Is there a way to plot graphs on paper?* Graphs are saved from the plot screen itself, not from the plot menu. Issue the *plot* command at the plot prompt. Depending on your lab setup, the graph will be sent to a plotter or saved in a file for plotting later.

7. *Why don't I see a strong precessional effect in solar radiation? For example, when I set the year to 10000 AD, there's not much change in June incoming solar. I thought the seasons are reversed every 10000 years or so.* Yes, but because we name our months relative to the equinoxes and solstices, June is always Northern Hemisphere summer. June radiation will be different in 10000 AD only because the June earth-sun distance and planetary tilt will have changed.

8. *Why doesn't the program always calculate solar radiation before making a run?* New radiation values are computed only when you change a parameter that affects solar radiation.

9. *What aspects of the albedo function are under my control?* You can change the slope of the curve and you can shift it left and right, but the maximum and minimum values are fixed. When you change slope, you change the magnitude of the feedback. Left/right shifts cause global mean albedo to change.

10. *Why is it that whenever I change albedo feedback, the planet becomes very cold, e.g., -179°C ?* Your value for albedo feedback is probably a large negative number. This makes for an unstable system. Once the planet starts to cool, the cooling reinforces itself until all latitudes are “ice-covered”.

11. *Why does global average F change so much, even when earth climate has hardly changed at all?* For the planet as a whole, northward transport roughly balances southward transport. As a result, global average F tends to be small, and can show a large percentage change in the presence of a small climate change.

B Technical Issues

B.1 Hardware Requirements

The computer program runs on IBM-PC's, AT's, PS/2's and compatibles using DOS 2.0 or higher. About 385 KB of available memory is required, some of which is allocated after the program is loaded. A math co-processor (Intel 80x87) is strongly recommended, though not essential. (In the absence of a math chip, floating point instructions are emulated using much slower integer operations). The program is overlaid, and reads code repeatedly from the current disk directory during execution. Although it's possible to run from a floppy disk, a fixed disk (or ram disk) is preferable.

For graphical output, one of the following IBM-type adapters is required: Enhanced Graphics Adapter, Professional Graphics Adapter, Video Graphics Array, or 8514/A (with memory expansion option). Except for the PGA, all are assumed to be single-monitor systems. The PGA is treated as a dual adapter system, with text written to a monochrome adapter. Before the program can use an 8514/A, the 8514/A Adapter Interface Code must be installed. This software is shipped by IBM with the adapter, and is often installed at boot time. None of the other adapters require special device drivers—the program writes directly to the hardware or uses built-in adapter software (rom bios). Unless instructed to the contrary, the program assumes that a VGA is present. Please note that a graphics adapter is not needed to run the program, nor is one needed for tabular output.

Hardcopy text and graphics data are sent to external “files” using DOS I/O routines. By default, tabular data is sent to lpt1:, and graphics data is sent to lpt2:. These can be changed by a command-line option (see below), so that data are routed to alternative ports or files for printing later. If data are store in a file, instructors must provide students with some mechanism for transmitting files to the appropriate devices. (In many cases, the DOS “print” or “copy” command will suffice.) Please note that DOS drivers do not necessarily handshake with external devices, or make any attempt to determine if a device is in fact ready to receive data. As a result, data overruns may be experienced when plotting complicated graphs on devices with small I/O buffers. The file specification can include a drive and pathname, but must not reference an existing file.

Tables are printed after being displayed on the screen. They will print on any 80-column printer capable of reproducing the IBM character set. Except for linefeeds, formfeeds and carriage returns (decimal 10, 12 and 13), no special formatting characters are used. However, because hardcopy tables are intended to be screen images, it is assumed that the full IBM character set can be printed (including decimal 128 to 255). Printers without this capability can still be used, but the program should be invoked with the “NOIBMSET” option (see Section 6.3). Unless instructed otherwise, the program formats tables for a 65-line page. By default tables are not broken across pages, and 5-line margins are inserted at top and bottom.

Graphs are saved from within the graphics module, and are formatted in a device-dependent manner. Only two device types are supported: HP 7475A Plotters (HPGL) and PostScript Printers (Level 1 PostScript). The user is prompted for a device type just before plotting the first graph. In both cases the program sends ASCII data containing plot instructions to the plotter device. As mentioned above, the device may be a port or disk file. When plotting to PostScript files, data for new plots is appended to the file, with each plot separated by a showpage command. If the HP plotter has been selected, each graph will be stored in a separate file (there is no chart advance command in HPGL). All files will have the same root name, but will have extensions .001, .002, .003, and so forth.

B.2 Installation

The distribution diskette contains ten files, including seven data files read by the program. Regardless of where the program is located, all seven data files must reside in the same directory. This may or may not be directory the from which the program is invoked. The simplest way to install the program on a hard disk is to create a single directory containing the program and data. Assuming the current drive is to hold the program, this can be accomplished with the following three DOS commands:

```
md ebmodel
cd ebmodel
copy a:*.*
```

The directory *ebmodel* will contain the files shown in Table 1. The first three files

File Name	Size	Contents
EBMODEL.EXE	490 KB	Executable Program
EBPVALS.BIN	12 KB	Standard Parameter Values + Menu Data
EBSVALS.BIN	15 KB	Standard Data Values
F77L.EER	11 KB	Runtime Error Messages
EBXYFONT.BIN	17 KB	Graphics Stroke Fonts
EBXYHELP.BIN	30 KB	Graphics Help File
EBXYPOST.INI	14 KB	Graphics PostScript Preamble
EBXY8514.FNT	10 KB	Graphics Hardware Font—8514 Adapter
READ.ME	1 KB	Installation Instructions—Not Used By Program
EBMODEL.PS	199 KB	This Manual, Formatted for a PostScript Printer

Table 1: Program and related files on distribution diskette.

are always required for execution. Others may be deleted to save disk space, depending on the machine configuration. For example, all of the EBXY files may be deleted if a graphics adapter is not installed. Likewise, the PostScript and 8514/A files may be deleted if those devices are not going to be used. The error file is not required, but is helpful if some fatal problem occurs, such as a hardware failure or disk full. The readme file is not needed at all during execution. The last file, *ebmodel.ps*, is a copy of the PostScript printer file used to print this document.

B.3 Startup and Configuration

Several optional parameters are available for fine-tuning the program to a particular hardware configuration. Configuration parameters are supplied on the command line each time the program is invoked; they are not stored in a file nor remembered from session to session. To configure a session, one types “*ebmodel*”, followed by no more than seven optional arguments:

```
EBMODEL -MONO -NOIBMSET -adapter -Pprinter -Lplotter -pagelength  
-margin -path
```

The first argument is typed exactly as shown, and is used to override default colors. It forces white text and lines on a black background, regardless of the type of adapter present. This is sometimes useful even for “color” adapters

attached to plasma or LCD displays, which typically do a poor job of rendering color.

The second argument is used when a printer cannot print extended characters (decimal 128 through 255). This will force translation of special characters used for table borders, exponents, etc., so that tables will print on any 7-bit ASCII device.

The third argument is a variable used to identify the type of display adapter present. The choices allowed are `-VGA`, `-EGA`, `-PGA`, `-8514/A` and `-ALPHA`. The program chooses the highest graphics resolution possible for the device specified. If `-ALPHA` is used, the program assumes a monochrome alphanumeric display is present, and no graphics output is attempted.

The third and fourth arguments specify printer and plotter ports. They are used to override the default devices (`lpt1:`, `lpt2:`). For example, if the plotter is connected to a serial port, one might use `-Lcom1:` to assign the plotter port. Filenames may be supplied as default “devices”.

The next two arguments are variables used to control pagination. The number of lines per page is given by *pagelength*. Any integer value is acceptable, but values below 25 disable page control and cause continuous printing with no top or bottom margin. When pagination is enabled, the program will not break a table across a page and skips *margin* lines at the top and bottom of every page. The default values are 65 and 5, which work reasonably well when printing 6 lines per inch on an 11 inch page. Both of these arguments are specified by `-n`, where *n* is an integer (e.g., `-65`, `-5`). The smaller value is always treated as *margin*. To disable pagination and margins, enter `-0 -0`. To disable margins, enter only `-0`.

The last argument is a path pointing to the location of the input data files. For example, if the data files are on the `c:` drive in a directory named `ebmodel`, then one would enter `ebmodel -c:\ebmodel`. This parameter only affects input file specifications; output files are always placed in the current directory on the current drive.

Note that if no command-line arguments are supplied, the program produces text and graphics for a color VGA system. Tables are printed using extended characters, and formfeeds are generated for a 65-line page with 1-inch margins.

Finally, without a path over-ride, data files are assumed to be in the current directory. Command line arguments can be entered in any order, and are not case sensitive.

Symbol	Meaning
$\alpha(\phi, t)$	surface albedo (prescribed values—no feedback)
$\alpha(T)$	surface albedo (calculated—with feedback)
α_{ij}	surface albedo for month i , zone j
α_j	annual insolation-weighted surface albedo for zone j
α_G	global insolation-weighted surface albedo
γ	slope of albedo feedback curve ($^{\circ}\text{C}^{-1}$)
δ	solar declination (radians)
ϵ	planetary emissivity
κ	eddy diffusion coefficient ($\text{W m}^{-1} \text{K}^{-1}$)
ν	true anomaly (radians)
ω	obliquity (radians)
ϕ	latitude (radians)
σ	stefan-boltzmann constant ($\text{W m}^{-2} \text{K}^{-4}$)
τ	atmospheric transmissivity
Ψ	atmospheric transmission
CO_2	atmospheric CO_2 concentration (ppm)
C	heat capacity ($\text{J m}^{-2} \text{K}^{-1}$)
d	instantaneous earth-sun distance (m)
\bar{d}	annual average earth-sun distance (m)
F	northward heat flux (W m^{-1})
H	hour angle of sunset (radians)
I	outgoing infrared radiation (W m^{-2})
m	optical air mass
M	mean anomaly (radians)
Q	solar radiation reaching the surface (W m^{-2})
Q_{abs}	solar radiation absorbed at the surface (W m^{-2})
Q_o	solar radiation on top of the atmosphere (W m^{-2})
\bar{Q}	mean daily incident solar radiation (W m^{-2})
S	solar constant (W m^{-2})
t	time (seconds)
T	surface temperature (K)
V	normalized earth-sun distance
Z	solar zenith angle (radians)

Table 2: Symbols used in the text. Units are shown in parentheses.