An Energy Balance Climate Model Which Accounts For Changes In Heat Storage Of The Earth

Submitted By Eric J. Keyes

In Partial Fulfillment Of The Masters Of Science Degree In Hydrology

New Mexico Institute Of Mining And Technology

TABLE OF CONTENTS

LIST OF FIGURES

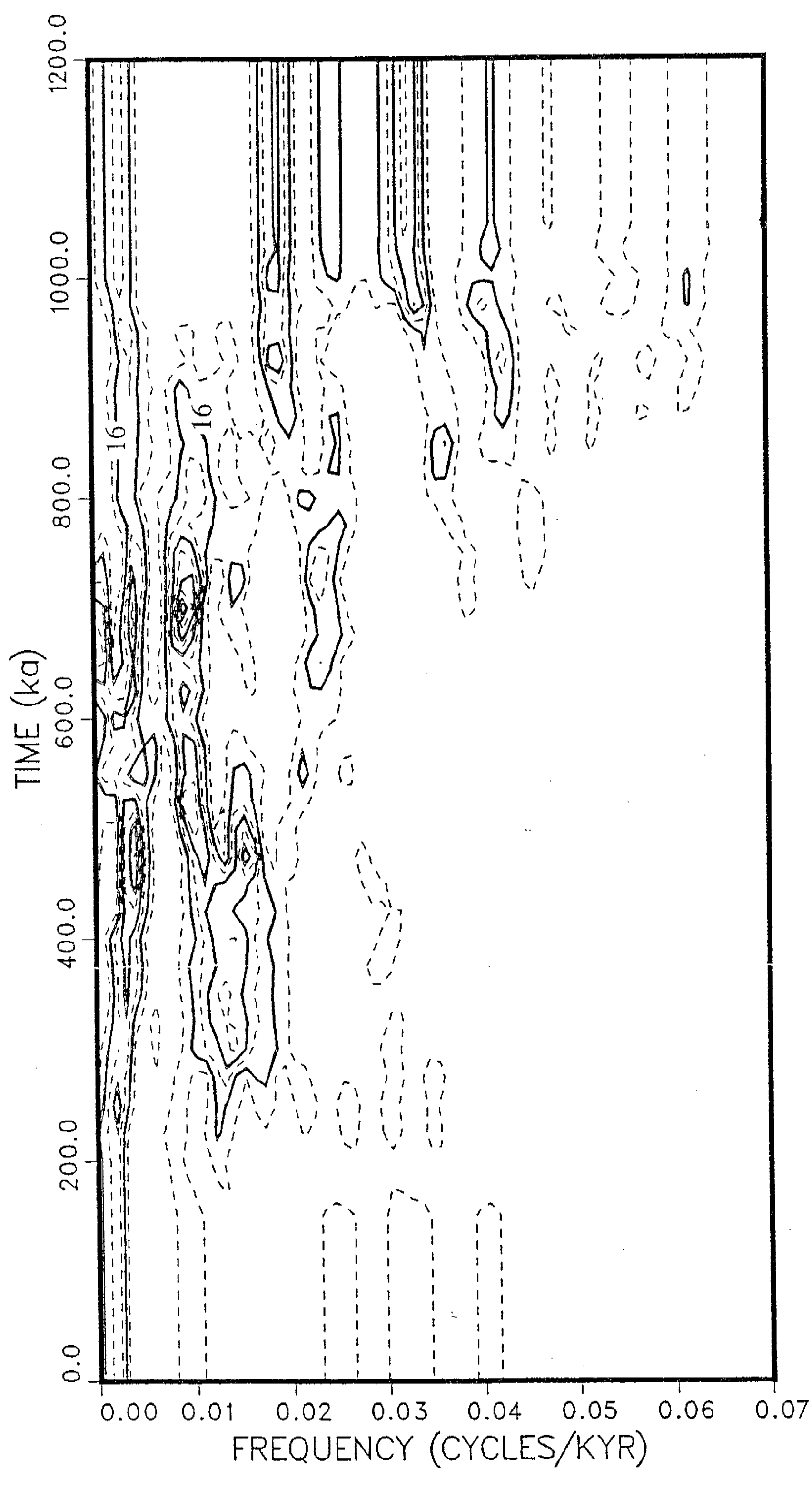
Introduction

Energy balance climate models have proved a useful method for simulation and physical understanding of earth temperature distributions. Early nontransient models by Budyko (1969) and Sellers (1969) accurately reproduced current average yearly surface temperatures. Sellers (1973) obtained enviable results with a simple transient model run for a one year period. Short term transient energy balance models may be successful despite either ignoring change in heat storage of the earth or examining it only to a depth of a meter or less. For models simulating longer time periods it is subject to conjecture whether change in heat storage for the deep earth will significantly effect surface temperatures. The energy balance model presented here, while not designed to precisely imitate historical temperatures, demonstrates the relationship between deep earth heat storage and surface temperature oscillations.

Spectral analyses of past climate have demonstrated a pronounced one hundred year cycle. Fluctuations in the earth 's orbital parameters give rise to dominant frequencies of 19, 23, and 41 thousand years. For a climate which responds linearly with solar insolation variations, the one hundred thousand year cycle cannot be justified. Snieder (1985), however, has shown that in a nonlinear system the nonlinearity can give rise to strong one hundred thousand year component. Phillips et al. (1990) used water and isotopic mass balance in a transient numerical energy balance model to simulate 1.2 Million years of lake level history at Searles Lake, California. A spectral analysis of his results (figure 1) reveals periodicities which cannot be accounted for by simple linear solar insolation forcing, in particular, a strong 400 kyr component. If changes in deep earth heat storage proves to be a substantial contributor of nonlinear feedbacks in the energy budget, a comparison with this established reconstruction of climate history would be prompted.

figure 1.

SPECTRAL DENSITY CONTOUR PLOT , SEARLES LAKE



The Model

Energy Balance

The energy balance equation used for the model is

$$R = A + G_E \qquad (1)$$

where,

R : Net radiation at surface of earth [kcal/cm^2 month]

A : Gain or loss of heat as a result of the atmosphere and hydrosphere circulation [kcal/cm^2 month]. Negative for gain of heat.

G_E: Change in heat storage of subsurface earth [kcal/cm^2 month]. Negative for loss of heat from the earth.

The net radiation term is expanded to its incoming and outgoing terms

$$R = Q (1 - \alpha) - I$$

where,

Q : solar radiation coming to the outer boundary of the atmosphere [kcal/cm^2 month]

α : albedo [−]

I : outgoing radiation [kcal/cm^2 month] .

The circulation term, A, and the outgoing radiation term, I, are related to a given latitudinal surface temperature using empirical relationships presented by Budyko (1968) . For outgoing radiation

$$I = a + B T - (a_1 + B_1 T) n \qquad (2)$$

where,

T : latitudinal surface temperature [°C]

n : fraction of cloudiness [−]

a : coefficient of value 14.0 [−]

B : coefficient of value 0.14 [−]

a_1 : coefficient of value 3.0 [−]

B_1 : coefficient of value 0.10 [−] .

For horizontal heat transfer in the atmosphere and hydrosphere

$$A = \beta (T - T_P) \tag{3}$$

where ,

T_P : mean earth temperature [°C]

β : coefficient of value 0.235 [kcal/cm^2 month °C] .

Equations (1), (2), and (3) are combined to form

$$T = \frac{Q(1-\alpha) - G_E - a + a_1 + \beta T_p}{\beta + B - B_1 n} \quad . \tag{4}$$

T_P is found from

$$T_p = \frac{Q_p\,(1-\alpha_p) - a + a_1 n}{B - B_1 n} \quad .$$

Q_p and α_p are average planetary values of radiation and albedo. In order to account for the variations of latitude belt surface area, the sampled latitude values of solar insolation and albedo were weighted with the cosine of the latitude when average planetary values were calculated.

The empirical Budyko equations, alone, cannot replicate the magnitude of historic surface temperature oscillations. Amplitude magnification of Budyko generated surface temperature was included in the model. For each sampled latitude, mean surface temperatures obtained from an exclusion of G_E simulation were read into the fortran model from the data file AVGT.DAT. These mean surface temperatures were calculated from a one hundred thousand year interval. The differences between the model generated temperatures and these mean values were multiplied by an amplification factor and added to the mean values, $T_{GE_{amplified}} = (T_{GE} - AVGT) * amplification\ \ factor + AVGT$. The amplified surface temperatures were used as the forcing function in G_E calculations. Amplification for all simulations was a factor of ten, producing fluctuations on the order of ten to twenty degrees Celsius.

Solar Insolation

Variations of temporal and latitudinal solar radiation over the simulations were calculated using an algorithm from Berger (1979). The algorithm calculates the energy available at any given latitude, with the assumption of a perfectly transparent atmosphere. Berger also assumes a constant solar constant of 1353 W/m^2. Long term variations of the earth 's eccentricity, obliquity, and longitude of the perihelion are taken into account. The trigonometric series expansions

$$\varepsilon = \varepsilon^* + \sum_{i=1}^{i=47} A_i \cos(f_i\ t + \delta_i\ )$$

$$e\ \sin\omega = \sum_{i=1}^{i=47} P_i \sin(\alpha_i\ t + \zeta_i\ )$$

$$e\ \cos\omega = \sum_{i=1}^{i=47} P_i \cos(\alpha_i\ t + \zeta_i\ )$$

are utilized to find values for the eccentricity e, the perihelion ω, and the obliquity ε. In the expansions above, ε^* is 23.320556°, t=0 refers to 1950 A.D., t is negative for times prior to 1950 A.D. The sets of forty−seven coefficients to the right of the summation signs are read into the model code from data files. The files are A_i 's from OAMP.DAT, f_i 's from OMNRT.DAT, δ_i 's from OPHASE.DAT, P_i 's from EAMP.DAT, α_i 's from EMNRT.DAT, and ζ_i 's from EPHASE.DAT. These files along with an independent code to calculate solar insolation given time and latitude, MILANK.FOR, are included in Appendix I.

Once the orbital parameters for a given time are known, the daily solar insolation for a given latitude and season can be calculated. The classical formulas for the daily insolation, W, are

1) for latitudes where there is no sunset

$$|\phi| \geq \frac{\pi}{2} - |\delta|$$

$$\phi > 0 \text{ if } \delta > 0$$

$$\phi < 0 \text{ if } \delta < 0$$

$$W = \frac{86.4 S_o}{\varrho^2} \sin\phi \sin\delta$$

2) for latitudes where there is no sunrise

$$|\phi| \geq \frac{\pi}{2} - |\delta|$$

$$\phi < 0 \text{ if } \delta > 0$$

$$\phi > 0 \text{ if } \delta < 0$$

$$W = 0$$

3) for latitudes where there is daily sunset and sunrise

$$-\left(\frac{\pi}{2} - |\delta|\right) < \phi < \frac{\pi}{2} - |\delta|$$

$$W = \frac{86.4 S_o}{\pi \varrho^2}(H_o \sin\phi \sin\delta + \cos\phi \cos\delta \sin H_o) \quad .$$

H_O is the absolute value of the hour angle at sunrise and sunset and is given by

$$cosH_o = -\tan\phi\tan\delta \quad .$$

All the angles which locate the earth on its orbit are taken as being constant over the whole day. The declination is related to the true longitude of the sun by

$$\sin\delta = \sin\varepsilon\sin\lambda$$

The normalized earth 's sun distance is given by

$$\varrho = \frac{r}{a} = \frac{1 - e^2}{1 + e\cos\nu} \quad .$$

Definitions for parameters in the above equations are

ϕ : latitude,

δ : declination of the sun,

ϱ : earth−sun distance r measured in units of the semimajor axis a,

H : hour angle of the sun during the day,

ν : true anomaly: positional angle of the earth on its orbit, counted counter clockwise from the perihelion,

M : mean anomaly: positional angle of a "mean" earth rotating around the sun with a constant angular speed equal to $2\pi/T$ and counted counterclock wise from the perihelion,

λ : true longitude of the earth is counted counterclockwise from the vernal equinox and is related to ν through $\lambda=\nu+\omega$. As in this formula, ω is mea sured from the vernal equinox,

λ_m : mean longitude associated with the mean earth is related to M:$\lambda_m=M+\omega$.

In the model, latitudes were sampled from 0° to 90° with 15° intervals. For each sample latitude, the solar insolation values were calculated at twenty−four equal subdivisions of each year. The solar insolation values were then averaged over the year, for each sample latitude, and used in the energy balance calculations..

Albedo

Albedos, for each latitudinal increment, were chosen based on present planetary climate conditions. For latitudes between 0° and 60° an average albedo of 0.32 was chosen. For latitudes between 60° and 70° the albedo was increased linearly from 0.32 to 0.50. For latitudes from 70° to 90° the albedo was increased linearly from 0.50 to 0.74. Constant albedo values, independent of surface temperature, were used to lend higher stability to the model. Energy balance models with transient albedos often cover the earth with an expanded icecap after a few thousand years (Schneider, Gal–Chen, 1973). A transient albedo function may be accounted for in future adjustments to the model.

Change in solid earth heat storage

The change in solid earth heat storage, G_E, if written in integral form is

$$G_E = C \int_0^d \frac{\partial T_E}{\partial t} dz \qquad (5)$$

where,

C : Heat capacity of solid earth [kcal/cm^3 °C]

d : depth in solid earth of lower boundary condition [cm]

T_E : Temperature at depth , z , for time , t , in subsurface earth [°C]

t : time [months]

z : depth [cm] .

Subsurface temperature distributions are determined using the heat flow continuity equation. For an isotropic, homogeneous body without interior heat sources or sinks, the one dimensional partial differential equation for transient heat conduction in the subsurface earth is

$$D_T \frac{\partial^2 T_E}{\partial z^2} = \frac{\partial T_E}{\partial t} \qquad (6)$$

where, D_T is the thermal diffusivity constant [cm^2/month]. In the model, this equation is evaluated numerically using an implicit finite difference approach. This approach generates the system of equations

$$\begin{bmatrix} 1+\lambda & -\lambda/2 & 0 & 0 & \cdots & & 0 \\ -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & \cdots & & 0 \\ 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & 0 & \cdots & 0 \\ 0 & 0 & -\lambda/2 & 1+\lambda & -\lambda/2 & 0 \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & 0 & -\lambda/2 & 1+\lambda & -\lambda/2 \\ 0 & 0 & \cdots & & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} T_{E_1}^{\,j+1} \\ T_{E_2}^{\,j+1} \\ T_{E_3}^{\,j+1} \\ \vdots \\ \vdots \\ T_{E_m}^{\,j+1} \end{bmatrix} =$$

$$\begin{bmatrix} (1-\lambda)T_{E_1}^{\,j} + \lambda/2\,(T_{E_2}^{\,j} + T_{E_0}^{\,j} + T_{E_0}^{\,j+1}) \\ (1-\lambda)T_{E_2}^{\,j} + \lambda/2(T_{E_1}^{\,j} + T_{E_3}^{\,j}) \\ (1-\lambda)T_{E_3}^{\,j} + \lambda/2\,(T_{E_2}^{\,j} + T_{E_4}^{\,j}) \\ \vdots \\ (1-\lambda)T_{E_{m-1}}^{\,j} + \lambda/2\,(T_{E_{m-2}}^{\,j} + T_{E_m}^{\,j}) \\ (.03\ ^\circ\mathrm{C/m})\Delta z \end{bmatrix} . \qquad (7)$$

Above, $\lambda = D_T \Delta t/(\Delta z)^2$, the superscript j+1 is the present time step, and the superscript j is the previous time step. The subscript on T_E is the depth node position. T_{E0} represents surface temperature. Boundary conditions were included in the system of equations. For each time step, the boundary condition at the surface of the earth was determined from equation (4). The G_E determined from the previous time iteration was used in the equation (4) calculation. The lower boundary condition was the geothermal gradient. A thermal gradient of .03 °C/meter was used (Lapidus).

The system of equations were solved using a Gauss–Seidal iterative algorithm. Whenever possible, this algorithm makes use of newly computed values of T_{Ei} to obtain greater efficiency in convergence to the solution. The calculation

$$T_{E_i} = \frac{-\sum_{l=1}^{l=i-1} a_{il}\, T_{E_l} - \sum_{l=i+1}^{l=m} a_{il}\, PT_{E_l} + b_i}{a_{ii}}$$

is preformed until $T_{Ei} - PT_{Ei}$ is below a prescribed tolerance level. PT_{Ei} is depth temperature at node i from the previous iteration. The tolerance level used in the model was 10^{-6} °C. The terms a_{il} and b_i are matrix positions in the matrixes on the left and right sides of equation (7), respectively. At the start of each Gauss–Siedal subroutine call, the initial values of PT_{Ei}, were depth node temperatures from the previous time step.

Numerical Differentiation

For equation (5), the derivatives of subsurface temperature with respect to time for a given depth were evaluated using a three point formula with Lagrange coefficient polynomials. Subroutine SUBSURF uses the equation

$$\left(\frac{dT}{dt}\right)_{at\ t} = -\frac{1}{\Delta t}\left[-\frac{3}{2}T_{at\ t} + 2T_{at\ t-\Delta t} - \frac{1}{2}T_{at\ t-2\Delta t}\right]$$

to calculate these derivatives. The three point formula lent greater accuracy to the model than would a two point formula.

Numerical Integration

The integration of these derivatives with respect to depth was accomplished using a Simpson Composite numerical integration scheme. This method subdivides the interval [a, b]into n subintervals and uses Simpson 's rule on each pair of consecutive subintervals, leading to improved precision. Subroutine SUBSURF uses the equation

$$\int_a^b f(x)dx = \sum_{j=1}^{j=m} \frac{\Delta x}{3}[f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})]$$

to accomplish the Simpson Composite numerical integration. In this equation, m=n/2, where n is the number of depth nodes.

All calculations in the model were done in double precision. The fortran code representing the model is in Appendix II.

Model Verification

To check the model, a heat transfer boundary value problem with known analytic solution was utilized. The problem that was tested was

$$D_T \frac{\partial^2 T_E}{\partial z^2} = \frac{\partial T_E}{\partial t}, \quad 0 < z < d, \ \ 0 < t$$

$$T_E(0,t) = T_0 \ , \quad 0 < t$$

$$T_E(d,t) = T_1 \ , \quad 0 < t$$

$$T_E(z,0) = T_1 \ , \quad 0 < z < d \ .$$

The analytic solution is $T_E(z,t) = T_0 + (T_1 - T_0)\frac{z}{d} + \sum_{n=1}^{n=\infty} b_n \sin\lambda_n z \ e^{-\lambda_n{}^2 D_T t}$

where, $b_n = \frac{2}{n\pi}(T_1 - T_0)$and $\lambda_n = \frac{n\pi}{d}$. After differentiating the analytic solution for T_E with respect to t and integrating with respect to z, the analytic solution for change in subsurface heat storage was obtained,

$$G_E = C \int_0^d \frac{\partial T_E}{\partial t} dz = C \sum_{n=0}^{n=\infty} \lambda_n \, D_T \, b_n \, (cos\lambda_n d \; - 1) \; e^{-\lambda_n^2 D_T t} \quad .$$

A comparison with the numerical model output yielded identical solutions.

Results

The numerical model was used to simulate the effect of inclusion of earth heat storage into the earth surface energy balance calculation. Omission of earth heat storage in the energy balance model is equivalent to an earth thermal diffusivity of zero. Deviations in surface temperature from this D_T=0 case were observed over a range of plausible solid earth thermal diffusivities. A range of reasonable values for heat capacity and thermal diffusivity of the earth were obtained from Saltzman(1985), Oliver and Fairbridge(1987), and James(1989). Values for some common earth materials are shown in table 1.

table 1. Thermal parameters of various earth materials
(after Oliver and Fairbridge)

material	heat capacity cal/cm^3 °C	thermal diffusivity cm^2/sec
ice	0.45	0.012
dry sand	0.3	0.0013
wet soil	0.4	0.01
still water	1.0	0.0015
stirred water	1.0	50 variable and estimated

Simulations were performed for eight values of thermal diffusivity. These values ranged from 2 x 10^{-6} m^2/sec, a best estimate of solid earth thermal diffusivity, to 200 x 10^{-6} m^2/sec. For diffusivity values in this range, a depth node spacing of approximately one hundred meters produced results nearly identical to depth node spacings of either ten meters or one meter. The greater node spacing was chosen for its rapidity of calculation. Time node spacing of fifty years yielded stable output.

Figure 2 demonstrates the lag and magnitude of the change in heat storage oscillations for the total depth d=1000 meters case. For comparison purposes, the $D_T=0$ average planetary temperature case is illustrated in the upper plot. G_E is not an exact linear transformation of ΔT_P Visually, however, the differences between the curves are indistinguishable when a second y−axis is added to the lower plot. The magnitude of these G_E peaks translates to a maximum temperature difference of about five hundredth of a degree Centigrade when compared to the exclusion of heat storage, $D_T=0$, case. As shown in figure 3, the peaks asymptotically approach a limiting value as the thermal diffusivity increases. This limiting value is dependant on the depth to the lower boundary condition, d. For d=3000 meters, the energy ratio approaches a much larger limiting value. This is also shown in figure 3. At the lower end of the diffusivity range, the differences between the energy ratios are minor.

Figure 4 and table 3 reveal the lag structure of G_E with variations in thermal diffusivity and depth, d. Again, for D_T's near the best estimate end of the axis, there is little variation with changes in effective penetration depth. Differences in the lag structure and energy ratio curves may be due to energy balance contributions from the constant energy flux at the lower boundary condition.

The contribution of changes in solid earth heat storage into the surface energy balance appear to be insignificant when thermal diffusivity values near 2×10^{-6} m^2/sec are taken as average earth values. To determine the effects on surface temperature oscillations if changes in solid earth heat storage were a significant factor, the model was executed using a thermal diffusivity value of .005 m^2/second. This value is the estimated thermal diffusivity of stirred water. In order to maintain stable calculations, depth node spacing of 3000 meters was required. The total depth was 27,000 meters. The time node spacing was 50 years. Figures 5 and 6 reveal the output of the model calculation. The time range was set for a period of 1.2 million years for comparison with figure 1. The output signal with inclusion of G_E is a dampening of the noninclusive signal. Lags between the two temperature outputs are less than 2 kyr. E−fold response time for a step input is given by $t=d^2/D_T$ (James, 1989) . A value of 4.62 kyr is obtained using the above parameters.

Comparison spectral plots are in figure 7 and figure 8. Figure 7 is created from spectral data from the exclusion of the G_E term case. Figure 8 is created from spectral data from the inclusion of G_E case. The contour plot data files were formed using moving 200,000 year subintervals spaced 40,000 years apart. The spectral calculations were preformed using a Lomb periodogram. The fortran code which created the data files is PLATE.F. It is included in appendix III.

Spectral contour plots reveal only minor contrasts between the two cases. There is a slight elongation in the spectral mound near frequency .052 cycles/kyr (period of 19,200 years) and the time 400 to 500 ka. There is also some change in the low power mounds near frequency .01 cycles/kyr (period of 100,000 years). Inclusion of G_E into the energy balance facilitates the loss of the mound near the time 520 to 690 ka. The inclusion of G_E also reshapes the mound from 930 to 1080ka. Overall however, it appears that as the uniform thermal diffusivity chosen for the earth is increased and the lag between G_E and Q_P goes to zero, the two temperature cases become nearly linearly related.

Also of note, in figures 7 and 8, are the absence of spectral mounds corresponding to orbital forcing due to variations of the obliquity of the ecliptic. Orbital forcing of this variety should generate frequencies near .0243 cycles/kyr (periods of 41,000 years). The averaging process, which weights lower latitude solar insolation values more heavily, was thought to dampen out these frequencies. Figures 9 and 10, spectral contour plots of surface temperature values simulated at a latitude of 75°, demonstrate activity in the obliquity frequency area. As in the average surface temperature spectral plots, inclusion of G_E into the energy balance equation does not significantly alter the frequency pattern of the exclusion of G_E case.

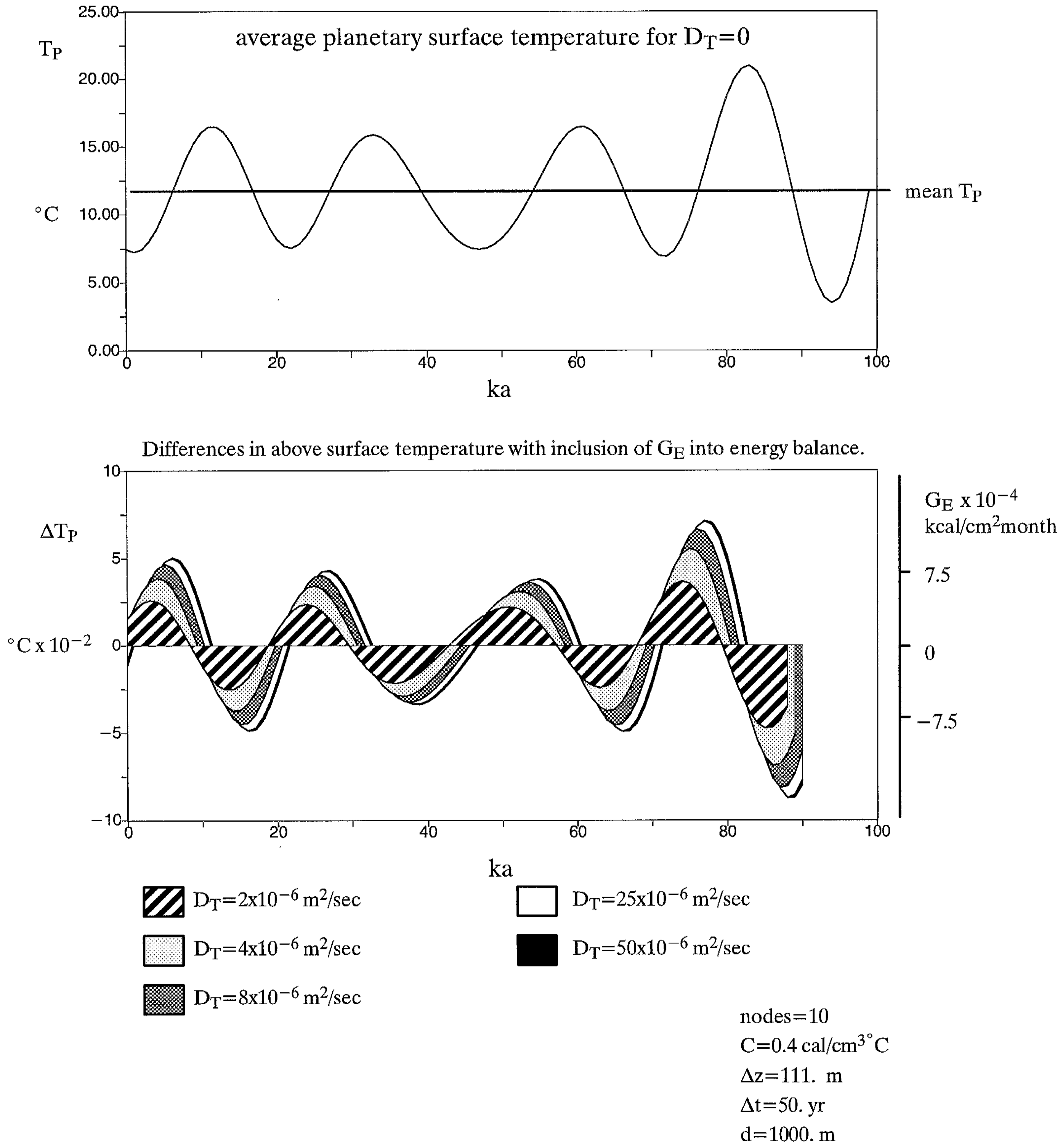


Figure 2. Effect of varying earth thermal diffusivities on subsurface energy contribution.

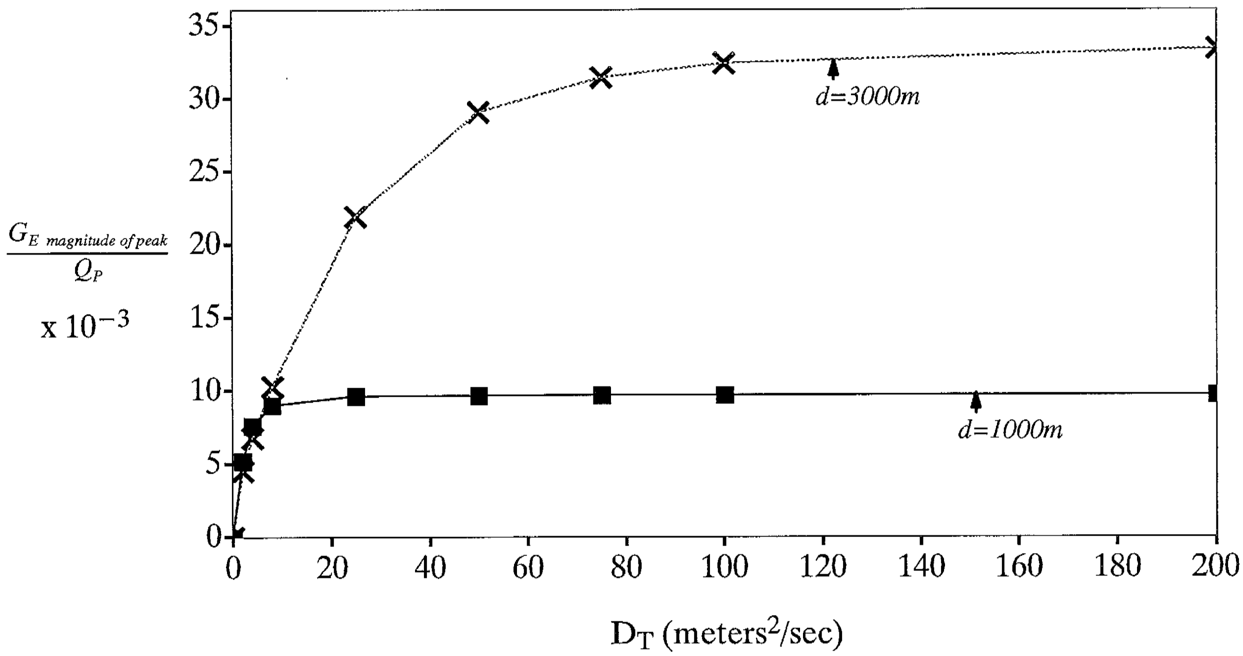


Figure 3. Sensitivity plot of change in energy ratio with variation of thermal diffusivity.

Table 2. variation of average ratio with thermal diffusivity.

	d=1000 m	d=3000 m
D_T (m^2/sec)	ratio x 10^{-3}	ratio x 10^{-3}
2 x 10^{-6}	5.199	4.518
4 x 10^{-6}	7.604	6.775
8 x 10^{-6}	9.015	10.276
25 x 10^{-6}	9.6125	21.888
50 x 10^{-6}	9.666	29.066
75 x 10^{-6}	9.675	31.441
100 x 10^{-6}	9.677	32.406
200 x 10^{-6}	9.681	33.381

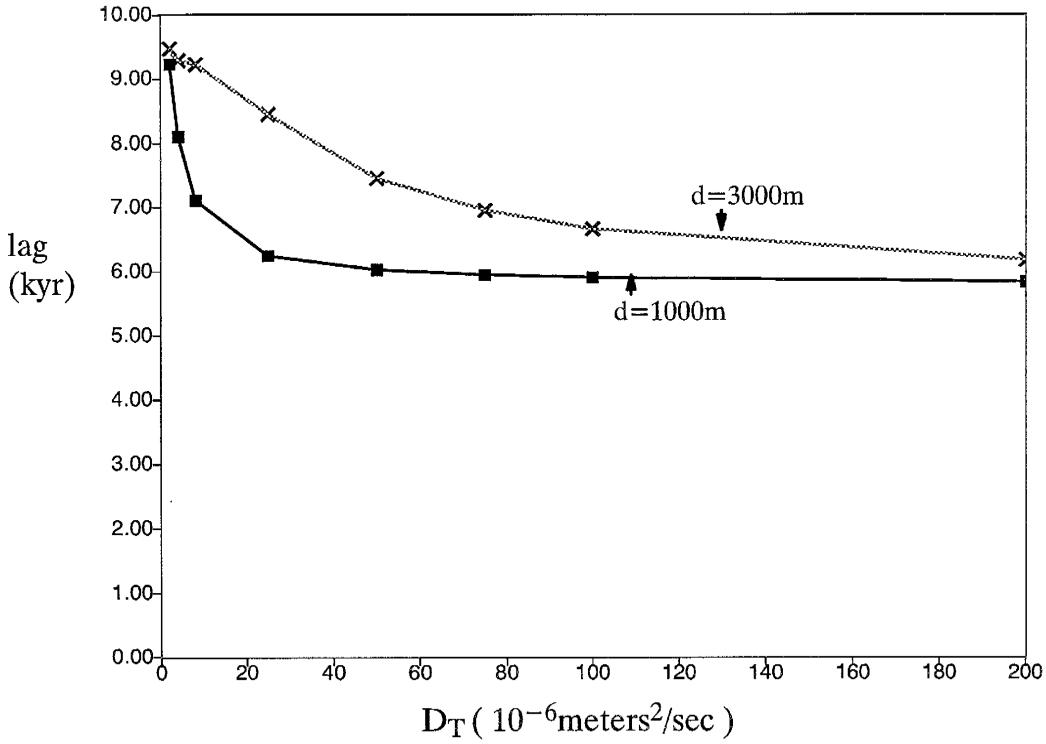


Figure 4. Sensitivity of lag with changes in thermal diffusivity.

Table 3. variation of average lag with thermal diffusivity.

	d=1000 m	d=3000 m
D_T (m^2/sec)	avg. lag (kyr)	avg. lag (kyr)
2×10^{-6}	9.225	9.47
4×10^{-6}	8.100	9.29
8×10^{-6}	7.100	9.225
25×10^{-6}	6.240	8.45
50×10^{-6}	6.025	7.45
75×10^{-6}	5.950	6.95
100×10^{-6}	5.906	6.66
200×10^{-6}	5.837	6.18

figure 5. Comparison of average surface temperature over 1200 kyr.

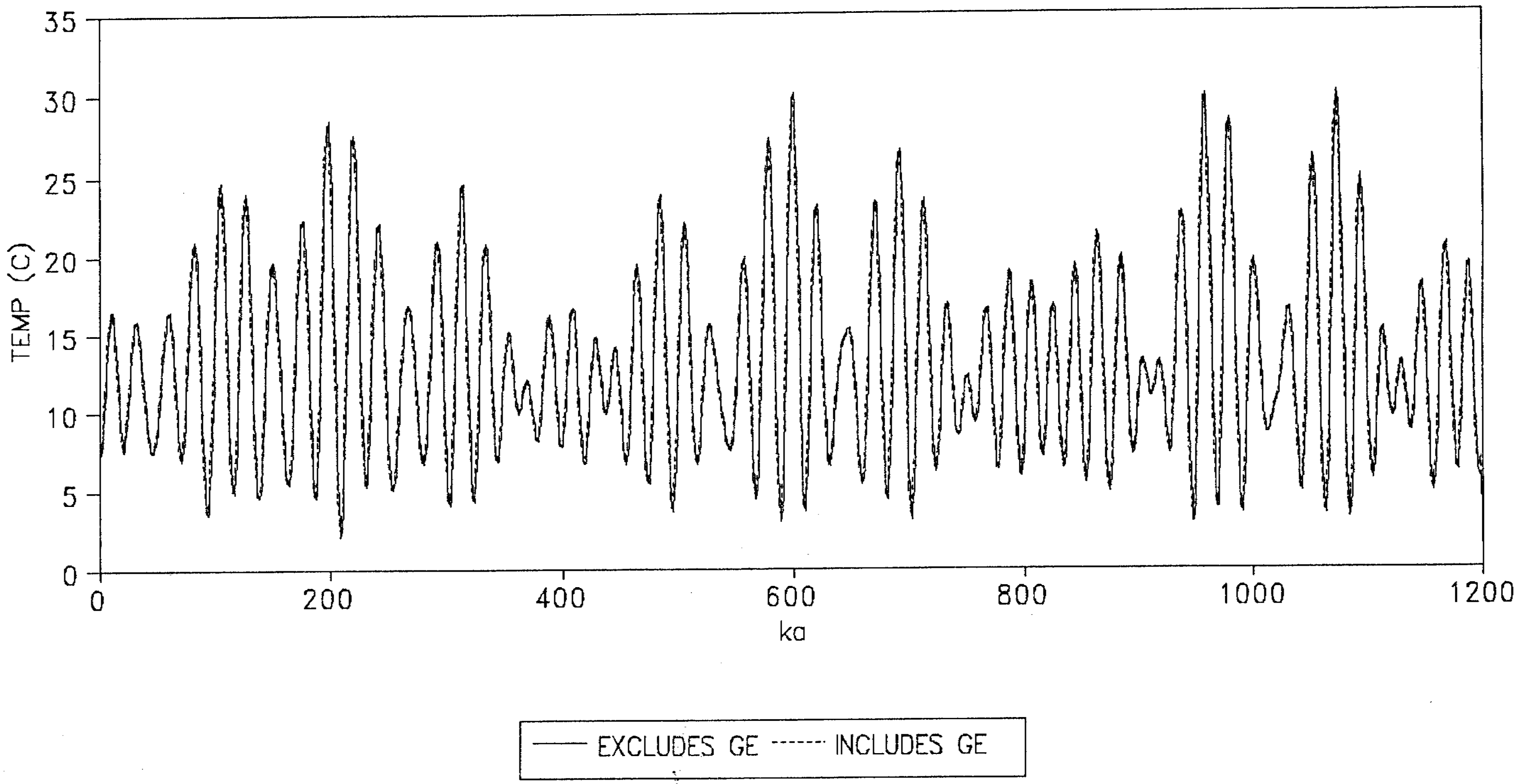


figure 6. Comparison of average surface temperature over 200 kyr.

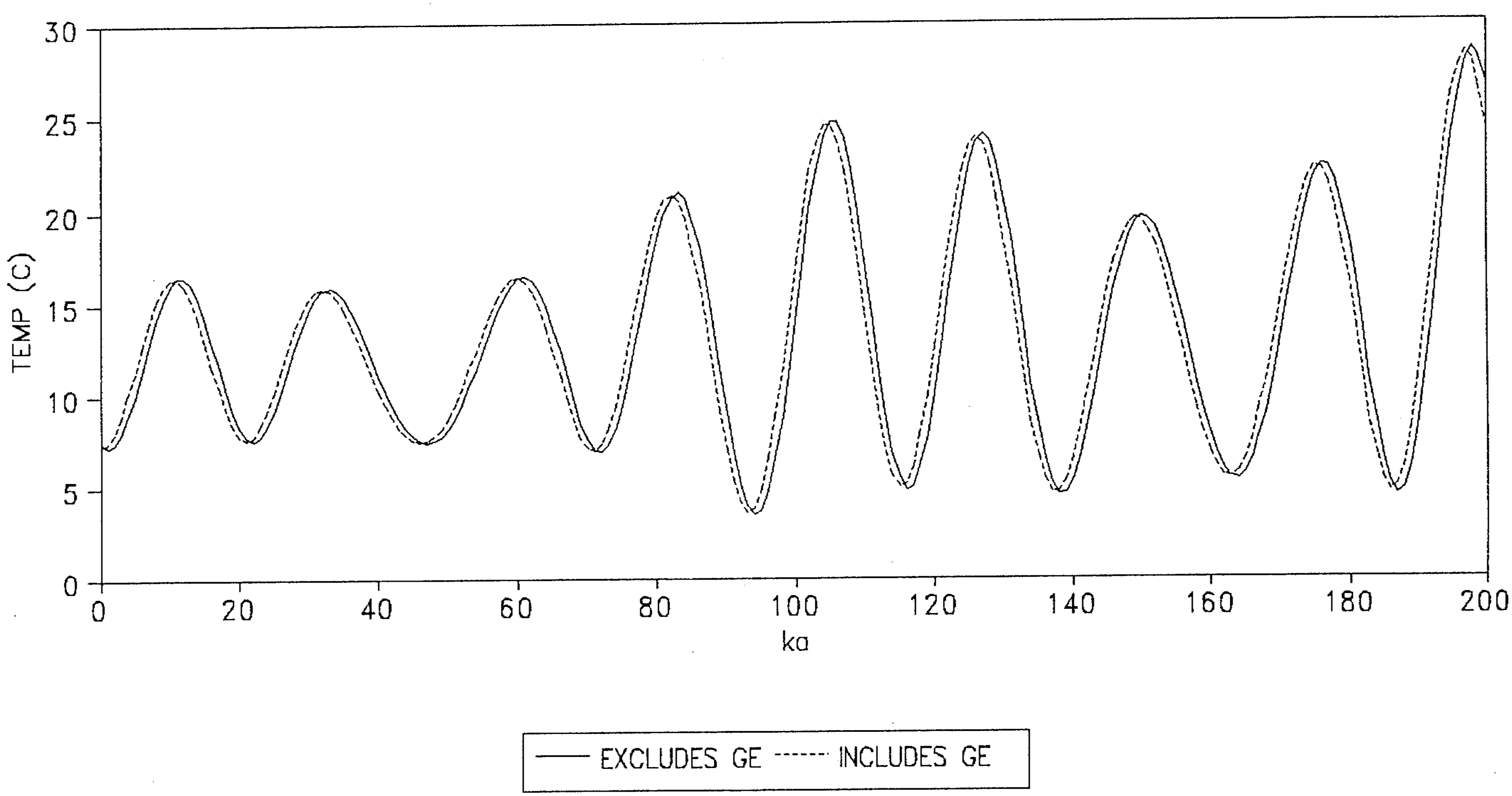


figure 7. Spectral density contour plot of average earth surface temperature, excluding G_E.

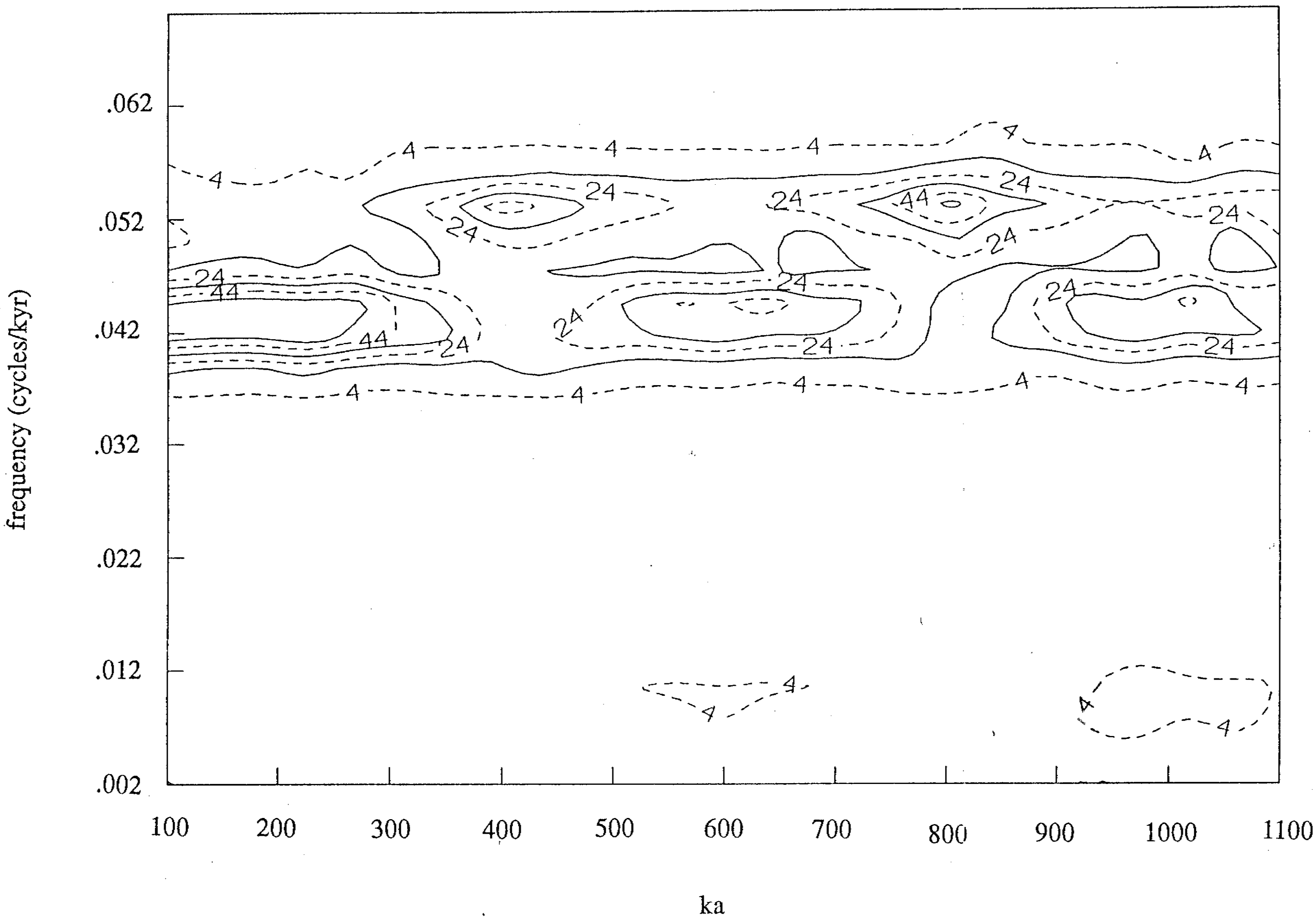
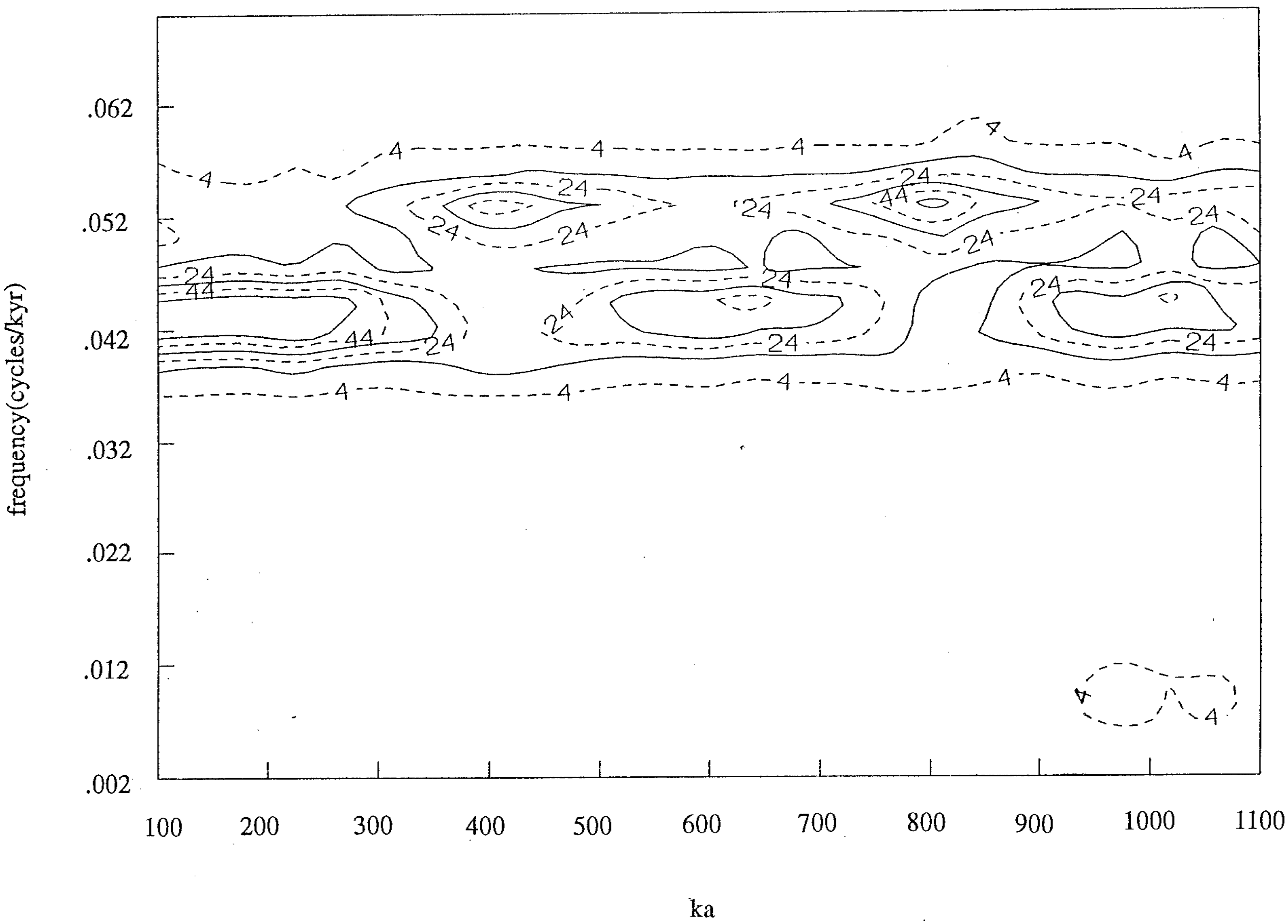


figure 8. Spectral density contour plot of average earth surface temperature, including G_E.



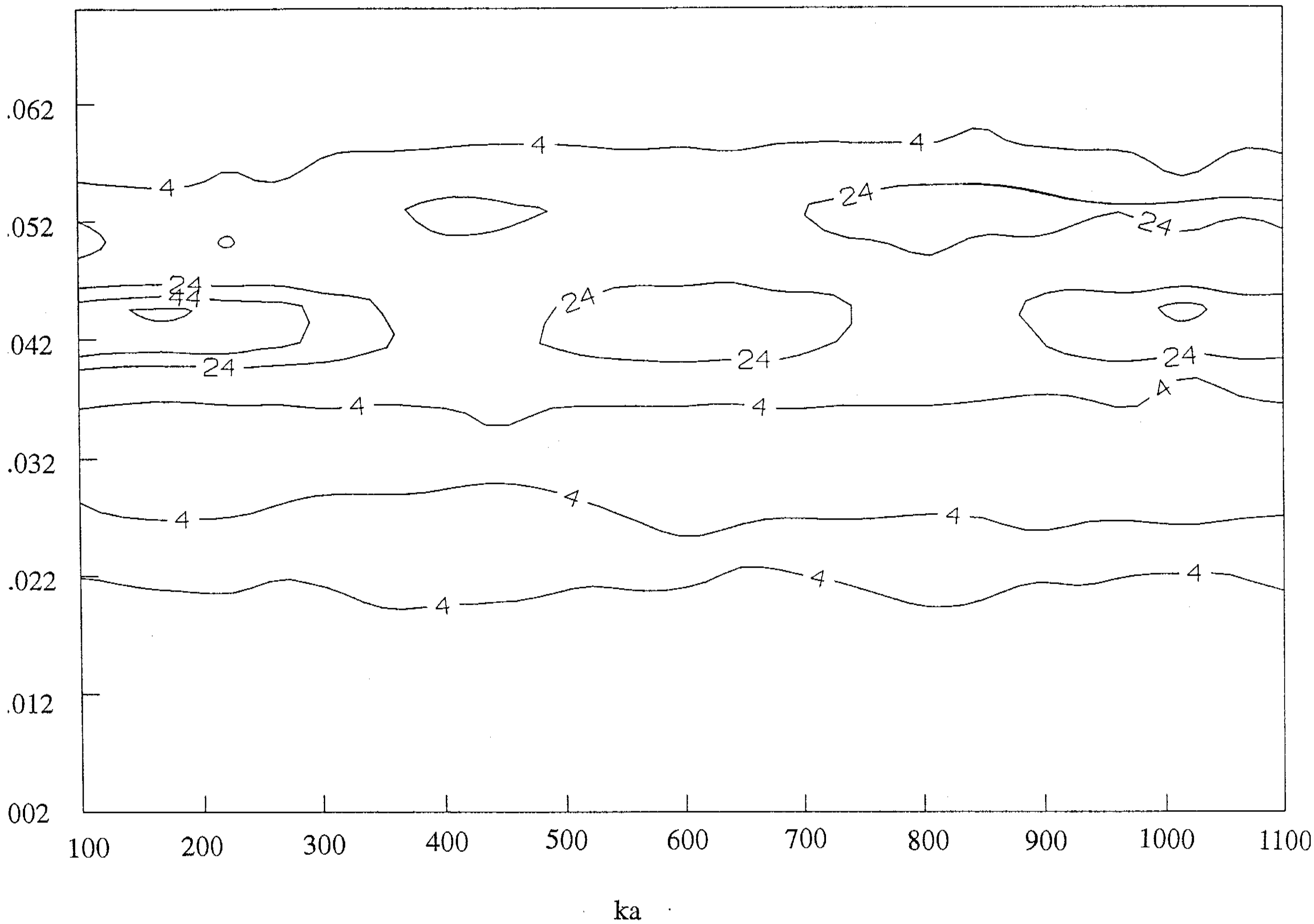


figure 9. Spectral density contour plot of earth surface temperature at 75° latitude, excluding G_E .

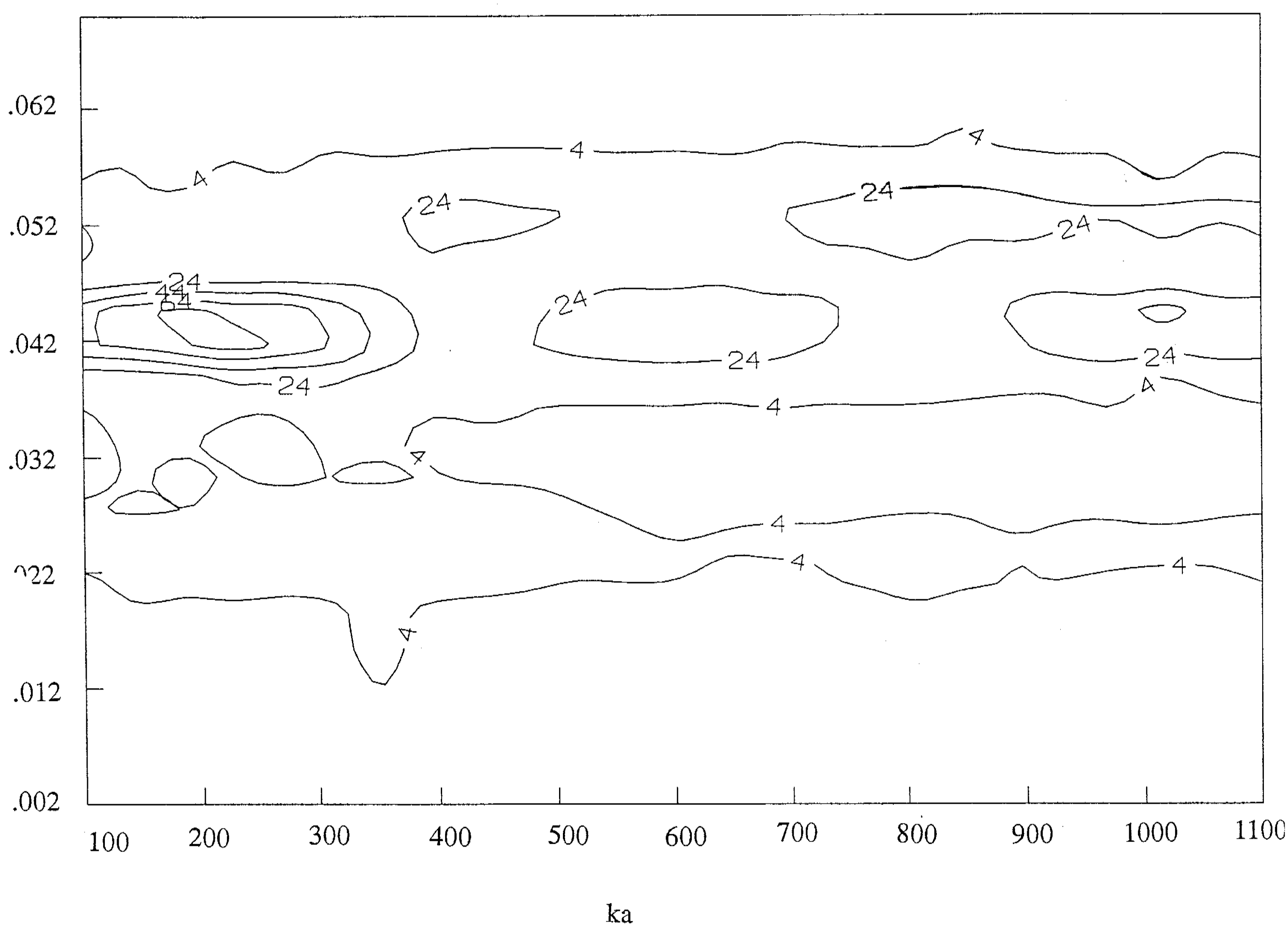


figure 10. Spectral density contour plot of earth surface temperature at 75° latitude, including $\bar{G}_E$.

Conclusions

The model demonstrates that changes in solid earth heat storage represent a very small portion of the total energy budget at the surface of the earth. The choice of effective penetration depth effects maximum energy ratio. For values of thermal diffusivity on the order of 10^{-6} m^2/sec, however, the solid earth heat storage contribution represents a Budyko temperature deviation of only .05°C.

Forcing the thermal diffusivity to magnitudes large enough to produce significant temperature deviations, energy contributions from changes in earth heat storage no longer lag far enough behind the solar insolation contribution to produce nonlinearity in the system. More complex thermal couplings would undoubtable occur if separate land and ocean masses were discretized in the model. Future versions of the model will also require simulation of ice cap expansion, and the consequent effect on albedo.

BIBLIOGRAPHY

Berger, Andre L. "Long–Term Variations of Daily Insolations and Quaternary Climatic Changes," *Journal of Atmospheric Sciences,* 35:2362–2367. December, 1978.

Berger, Andre L. "Milankovitch Theory and Climate," *Reviews of Geophysics*, 26:624–657. November, 1988.

Budyko, M. I. "The Effect of Solar Radiation Variations on the Climate of the Earth," *Tellus,* XXI (1969), 5:611–619

Budyko, M. I. *Climatic Changes.* American Geophysical Union, 1977.

Burden, Richard L. and J. Douglas Faires. *Numerical Analysis.* 3rd ed. Boston: Prindle, Weber & Schmidt, 1985.

Eagleton, Peter S. *Dynamic Hydrology.* New York: McGraw Hill, 1970.

James, David E. ed. *The Encyclopedia of Solid Earth Geophysics.* New York: Van Nostrand Reinhold Company, 1989.

Lapidus, Dorothy Farris. *Dictionary of Geology and Geophysics.* New York: Facts on File Publications, 1987.

MacDonald, Gordon J. "Spectral Analysis of Time Series Generated By Nonlinear Processes," *Reviews of Geophysics*, 27:449–469. November 1989.

North, Gerald R. *et al.* "Energy Balance Climate Models," *Reviews of Geophysics and Space Physics*, 19:91–121. February 1981.

Oliver, John E. and Rhodes W. Fairbridge eds. *The Encyclopedia of Climatology*. New York: Van Nostrand Reinhold Company, 1987.

Phillips, Fred M. *et al.* "1.2 Million Years of Lake Level History at Searles Lake, U.S.A.," *WRRI Report*, January 1991.

Powers, David L. *Boundary Value Problems*. 2nd ed. Academic Press, 1979.

Saltzman, Barry. " Paleoclimatic Modeling, " *Paleoclimatic Analysis and Modeling.* Ed. A.D. Hecht. Wiley, 1985.

Schneider, Stephan H. and Tzvi Gal–Chen. " Numerical Experiments of Climate Stability, " *Journal of Geophysical Research,* 78:6182–6194. September 20, 1973.

Shumway, Robert H. *Applied Statistical Time Series Analysis*. Englewood Cliffs, New Jersey: Prentice Hall, 1988.

Sellers, William D. "A Global Climate Model Based on the Energy Balance of the Earth–Atmosphere System, " *Journal of Applied Meteorology,* 8:392–400. June, 1969.

Sellers, William D. "A New Global Climate Model, " *Journal of Applied Meteorology*, 12:241–254. March, 1973.

Snieder, Roelof K. "The Origin of the 100,000 Year Cycle in a Simple Ice Age Model, " *Journal of Geophysical Research*, 90:5661–5664. June 20,1985.

Wang, Herbert F. and Mary P. Anderson. *Introduction to Groundwater Modeling, Finite Difference and Finite Element Methods.* New York: W.H. Freeman and Company, 1982.

APPENDIX I

```
C      PROGRAM MILANK.F

***********************************************************************
*CALCULATES SOLAR INSOLATION, USING AN ALGORITHM FROM BERGER, A. L.,*
*'LONG-TERM VARIATIONS OF DAILY INSOLATION AND QUATERNARY CLIMATE   *
*CHANGES,' JOURNAL OF THE ATMOSPHERIC SCIENCES, 35:2362-2367.       *
***********************************************************************

        DOUBLE PRECISION sumob, pi, sumc1, sumc2, obliq, obconst,
     +     long, eccen, oamp(50), omnrt(50), ophase(50),
     +     eamp(50), emnrt(50), ephase(50), const1, const2, solrconst,
     +     phi, delta, sq, one, solrins,rho,insconst,
     +     Ho,two,sqrt,LAMD,lam,nu

       parameter (obconst= 23.320556, pi = 3.141592653, const1 = 3600.,
     +           const2 = 180., solrconst=1.95, insconst=1440., sq=2.,
     +           one=1., two = 2., sqrt = 0.5)

** All of the solar insolation values a calculated for the month **
** corresponding to the parameter 'lam' .  Lam=0 corresponds **
** to approx March 21.  The mid month values are defined by a **
** change in 'lam' of thirty(30) degrees, i.e., april=30,may=60. **

** read coeff. into arrays **
** These coeff. are used to calculate the earths' wobbles and tilts, **
** or more specifically, its eccentricity, obliquity, and longitude **
** of the perihelion. **

      open(unit=61, status = 'old', file = 'oamp.dat')
      read (61,*) (oamp(i), i = 1,47)
      write(6,*)' read ',i-1,' points from oamp'
      close(unit=61)

      open(unit=62, status = 'old',file = 'omnrt.dat')
      read (62,*)(omnrt(j), j = 1,47)
      write(6,*)' read ',j-1,' points from omnrt'
      close(unit=62)

      open(unit=63, status = 'old',file = 'ophase.dat')
      read (63,*)(ophase(k), k = 1,47)
      write(6,*)' read ',k-1,' points from ophase'
      close(unit=63)

      open(unit=64, status = 'old',file = 'eamp.dat')
      read (64,*)(eamp(i), i = 1,46)
      write(6,*)' read ',i-1,' points from eamp'
      close(unit=64)

      open(unit=65, status = 'old',file = 'emnrt.dat')
      read (65,*)(emnrt(j), j = 1,46)
      write(6,*)' read ',j-1,' points from emnrt'
      close(unit=65)

      open(unit=66, status = 'old',file = 'ephase.dat')
      read (66,*)(ephase(k), k = 1,46)
      write(6,*)' read ',k-1,' points from ephase'
      close(unit=66)

        WRITE(6,*)'ALL INPUT DATA READ'
```

```
      WRITE(6,*)'ENTER VALUE FOR STARTING TIME'
      WRITE(6,*)'-10000. MEANS TEN THOUSAND YEARS AGO '
      WRITE(6,*)
      READ(5,*)TIME
      WRITE(6,*)

      WRITE(6,*)'ENTER VALUE FOR LATITUDE IN DEGREES'
      WRITE(6,*)
      READ(5,*)PHID
      WRITE(6,*)

      WRITE(6,*)'ENTER VALUE FOR SEASON'
      WRITE(6,*)
      WRITE(6,*)'MARCH 21=0., JUNE 21=90., SEPTEMBER 21=180., ETC.'
      WRITE(6,*)
      READ(5,*)LAMD
      WRITE(6,*)

** Convert to radians **

      phi = phiD*(pi/const2)
      lam = lamD*(pi/const2)

c.................................................................
        sumc1 = 0.
        sumc2 = 0.
        sumob = 0.

        do 200 j = 1,47

** Mid-month obliquity for a given LATITUDE AND TIME OF YEAR **

           sumob = sumob + (oamp(j)/const1)*(cos(((omnrt(j)*dble(time))
     +                           /const1 + ophase(j))*(pi/const2)))

** sumc1 & sumc2 are used to calculate eccentricity and longitude of the **
** perihelion based on a moving vernal equinox **

           sumc1 = sumc1 + eamp(j)*sin(((emnrt(j)*dble(time))/const1
     +                     + ephase(j))*(pi/const2))

           sumc2 = sumc2 + eamp(j)*cos(((emnrt(j)*dble(time))/const1
     +                 + ephase(j))*(pi/const2))

 200        continue

        obliq = sumob + obconst
        long = (atan2(sumc1,sumc2))*(const2/pi)
        eccen = sumc1/sin(atan2(sumc1,sumc2))

** PARAMETERS USED TO CALCULATE MID-MONTH SOLAR INSOLATION **

        delta = asin((sin(lam))*sin(obliq*(pi/const2)))

        nu = lam-((long+const2)*(pi/const2))

        rho = (one-(eccen**sq))/
     +          (one+(eccen*cos(nu)))
```

```
** MID-MONTH SOLAR INSOLATION IN CAL/CM^2-DAY **

      IF(ABS(PHI).GE.(PI/TWO-abs(DELTA)))THEN

      if(((phi.gt.(0.)).and.(delta.gt.(0.))).or.((phi.lt.(0.)).
     +   and.(delta.lt.(0.))))then

        SOLRINS=      ((insconst*solrconst)/(rho**sq))
     +       *sin(phi)*sin(delta)
      end if

      if(((phi.lt.(0.)).and.(delta.gt.(0.))).or.((phi.gt.(0.)).
     +   and.(delta.lt.(0.))))then

        SOLRINS = DBLE(0.)
      end if

      ELSE

       Ho = acos(-tan(phi)*tan(delta))

       solrins = ((insconst*solrconst)/(pi*(rho)**sq))
     +     *(Ho*sin(phi)*sin(delta)+cos(phi)*cos(delta)*sin(Ho))
      END IF

      WRITE(6,*)'SOLAR INSOLATION = ',SOLRINS,' CAL/CM^2 DAY'
      END
```

EAMP.DAT	EMNRT.DAT	EPHASE.DAT
0.018608	54.646484	32.01
0.0162752	57.785370	197.18
-0.0130066	68.296539	311.69
0.0098883	67.659821	323.59
-0.003367	67.286011	282.76
0.0033308	55.638351	90.58
-0.002354	68.670349	352.52
0.0014002	76.656036	131.83
0.001007	56.798447	157.53
0.000857	66.649292	294.66
0.0006499	53.504456	118.25
0.000599	67.023102	335.48
0.000378	68.933258	299.80
-0.000337	56.630219	149.16
0.0003334	86.256454	283.91
0.0003334	23.036499	320.11
0.0002916	89.395340	89.08
0.0002916	26.175385	125.27
0.000276	69.307068	340.62
-0.000233	99.906509	203.60
-0.000233	36.686569	239.79
0.000182	67.864838	155.48
0.0001772	99.269791	215.49
0.0001772	36.049850	251.68
-0.0001740	56.625275	130.23
-0.000124	68.856720	214.05
0.0001153	87.266983	312.84
0.0001153	22.025970	291.17
0.0001008	90.405869	118.01
0.0001008	25.164856	96.34
0.0000912	78.818680	160.31
0.0000912	30.474274	83.70
-0.0000806	100.917038	232.53
-0.0000806	35.676025	210.86
0.0000798	81.957565	325.48
0.0000798	33.613159	248.87
-0.0000638	92.468735	80.00
-0.0000638	44.124329	3.39
0.0000612	100.280319	244.42
0.0000612	35.039322	222.75
-0.0000603	98.895981	174.67
-0.0000603	35.676025	210.86
0.0000597	87.248322	342.48
0.0000597	24.028381	18.68
0.0000559	86.630264	324.73
0.0000559	22.662689	279.28

OAMP.DAT	OMNRT.DAT	OPHASE.DAT
-2462.22	31.60997	251.9
-857.32	32.620499	280.83
-629.32	24.172195	128.3
-414.28	31.983780	292.72
-311.76	44.828339	15.37
308.94	30.973251	263.79
-162.55	43.668243	308.42
-116.11	32.246689	240.
101.12	30.599442	222.97
-67.69	42.68132	268.78
24.91	43.836456	316.79
22.58	47.439438	319.6
-21.16	63.219955	143.8
-15.65	64.230484	172.73
15.39	1.01053	28.93
14.67	7.437771	123.59
-11.73	55.782181	20.2
10.27	.373813	40.82
6.49	13.218362	123.47
5.85	62.583237	155.69
-5.49	63.593765	184.62
-5.43	76.438309	267.27
5.16	45.815262	55.01
5.08	8.448301	152.52
-4.07	56.792709	49.13
3.72	49.747849	204.66
3.40	12.058272	56.52
-2.83	75.278214	200.32
-2.66	65.241013	201.66
-2.57	64.604294	213.55
-2.47	1.647247	17.03
2.46	7.811584	164.41
2.25	12.207832	94.54
-2.08	63.856659	131.91
-1.97	56.155991	61.03
-1.88	77.448837	296.2
-1.85	6.801054	135.48
1.82	62.209412	114.87
1.76	20.656128	247.06
-1.54	48.344406	256.61
1.47	55.145462	32.1
-1.46	69.000534	143.68
1.42	11.071350	16.87
-1.18	74.291306	160.68
1.18	11.047742	27.59
-1.13	.636717	348.1
1.09	12.844549	82.64

APPENDIX II

```
C     PROGRAM MODEL.F

C*******************************************************************
C  GENERATES TRANSIENT SOLAR INSOLATION VALUES AT EARTH'S EQUATOR . *
C USES THESE SOLAR INSOLATION VALUES TO DRIVE A SIMPLE BUDYKO        *
C MODEL WHICH DETERMINES TEMPERATURE VALUES AT A GIVEN LATITUDE.ALBEDO *
C VALUES ARE INVARIANT OF TEMPERATURE . FEEDBACK FROM SOLID EARTH    *
C HEAT STORAGE IN THE ENERGY BALANCE IS ACCOUNTED FOR VIA A          *
C SUBSURFACE IMPLICIT FINITE DIFFERENCE APPROXIMATION .              *
C*******************************************************************

      DOUBLE PRECISION sumob, pi, sumc1, sumc2, obliq, obconst,
     +   long, eccen, oamp(50), omnrt(50), ophase(50),ESINW50,
     +   eamp(50), emnrt(50), ephase(50), const1, const2, solrconst,
     +   phi, delta, sq, one, solrins,rho,GRAD(7,1000),insconst,
     +   Ho,two,trop, sqrt,TS(7),Q(7),ALBEDO(7),LAMD,lam,C,nu,
     +   DELTAZ,TEMP(7,1000),TLAST(7,1000),T2AGO(7,1000),DT,
     +   LAMDINT,OBL50,BETA,A1,A2,B1,B2,GE(7),
     +   TP,QP,SUMLATQ,SUMCOS,SUMGE,AVG_GE,LASTGE(7),
     +   TNOGE(7),TPGE,AMP,TAMP(7),AVGT(7),TSAMP(7),SUMTAMP,
     +   SUMTSAMP,AVG_TAMP,AVG_TSAMP,deltat,weight,sumweight

      INTEGER NODES

      parameter (obconst= 23.320556, pi = 3.141592653, const1 = 3600.,
     +         const2 = 180., solrconst=1.95, insconst=1440., sq=2.,
     +         one=1., two = 2., sqrt = 0.5,
     +         trop = 525948.761666,obl50=23.44631,BETA=.235,
     +         esinw50 = .016109170339,A1=14.0,A2=3.0,B1=.14,B2=.10)

** All of the solar insolation values a calculated for the month **
** corresponding to the parameter 'lam' .  Lam=0 corresponds **
** to approx March 21.  The mid month values are defined by a **
C  change in 'lam' of thirty(30) degrees, i.e., april=30,may=60. **

** read coeff. into arrays **
** These coeff. are used to calculate the earths' wobbles and tilts, **
** or more specifically, its eccentricity, obliquity, and longitude **
** of the perihelion. **

      open(unit=60, status = 'old', file = 'avgt.dat')
      read (60,*) (AVGT(i), i = 1,7)
      write(6,*)' read ',i-1,' points from AVGT'
      close(unit=60)

      open(unit=61, status = 'old', file = 'oamp.dat')
      read (61,*) (oamp(i), i = 1,47)
      write(6,*)' read ',i-1,' points from oamp'
      close(unit=61)

      open(unit=62, status = 'old',file = 'omnrt.dat')
      read (62,*)(omnrt(j), j = 1,47)
      write(6,*)' read ',j-1,' points from omnrt'
      close(unit=62)

      open(unit=63, status = 'old',file = 'ophase.dat')
      read (63,*)(ophase(k), k = 1,47)
      write(6,*)' read ',k-1,' points from ophase'
      close(unit=63)
```

```
      open(unit=64, status = 'old',file = 'eamp.dat')
      read (64,*)(eamp(i), i = 1,46)
      write(6,*)' read ',i-1,' points from eamp'
      close(unit=64)

      open(unit=65, status = 'old',file = 'emnrt.dat')
      read (65,*)(emnrt(j), j = 1,46)
      write(6,*)' read ',j-1,' points from emnrt'
      close(unit=65)

      open(unit=66, status = 'old',file = 'ephase.dat')
      read (66,*)(ephase(k), k = 1,46)
      write(6,*)' read ',k-1,' points from ephase'
      close(unit=66)

        WRITE(6,*)'ALL INPUT DATA READ'

c       open(unit=67, status = 'unknown',file = 'Q_INV.RES')
c       open(unit=70, status = 'unknown',file = 'qp.b7')
c       open(unit=71, status = 'unknown',file = 'avg_ge.b7')
c       open(unit=72, status = 'unknown',file = 'tp_qp.sim10')
c       open(unit=73, status = 'unknown',file = 't_inv.sim10')
      open(unit=74, status = 'unknown',file = 'avg_tamp.end')
      open(unit=79, status = 'unknown',file = 'avg_tsamp.end')
c       open(unit=80, status = 'unknown',file = 'tamp.sim10')
c       open(unit=81, status = 'unknown',file = 't_diff.test1')

**KEYBOARD ENTERED VALUES**

        WRITE(6,*)'ENTER VALUE FOR CLOUDINESS'
        READ(5,*)CLOUD
        WRITE(6,*)

        WRITE(6,*)'ENTER NUMBER OF DEPTH NODES ,MUST BE EVEN'
        READ(5,*)NODES
        WRITE(6,*)

        WRITE(6,*)'ENTER VALUE FOR THERMAL DIFFUSIVITY IN M^2/SEC'
        READ(5,*)DT
*CONVERT DT FROM M^2/SEC TO CM^2/MONTH
        DT=DT*10000.*3600.*24.*365./12.
        WRITE(6,*)

        WRITE(6,*)'ENTER VALUE FOR NODE SPACING IN METERS'
        READ(5,*)DELTAZ
*CONVERT FROM METERS TO CENTIMETERS
        DELTAZ=DELTAZ*100.
        WRITE(6,*)

        WRITE(6,*)'ENTER VALUE FOR DELTA T IN YEARS'
        READ(5,*)DELTAT
*CONVERT FROM YEARS TO MONTHS
        DELTAT=12.*DELTAT
        WRITE(6,*)

        WRITE(6,*)'ENTER LATITUDE INCREMENTS IN DEGREES'
        READ(5,*)PHIDINT
        WRITE(6,*)

        WRITE(6,*)'ENTER VALUE FOR HEAT CAPACITY OF EARTH IN CAL/CM^3-C'
```

```
      READ(5,*)C
* CONVERT C FROM CAL/CM^3-C TO KCAL/CM^3-C
      C=.001*C
      WRITE(6,*)

      WRITE(6,*)'ENTER VALUE FOR GE TOLERANCE IN KCAL/CM^2 MONTH'
      READ(5,*)GETOL
      WRITE(6,*)

      WRITE(6,*)'ENTER VALUE FOR SEASONAL INCREMENT IN DEGREES'
      WRITE(6,*)'ONE MONTH EQUALS 30 DEGREES'
      READ(5,*)LAMDINT
      WRITE(6,*)

      WRITE(6,*)'ENTER VALUE FOR STARTING TIME'
      WRITE(6,*)'-10000. MEANS TEN THOUSAND YEARS AGO '
      WRITE(6,*)
      READ(5,*)START
      WRITE(6,*)

      WRITE(6,*)'ENTER VALUE FOR TEMPERATURE AMPLIFICATION'
      READ(5,*)AMP

      WRITE(6,*)'ENTER FINAL TIME'
      READ(5,*)TFINAL
**TIME LOOP**

      DO 40 TIME=START,TFINAL,DELTAT/12.


**CALCULATE AVERAGE YEARLY TEMPERATURE FOR THE EARTH**

      IF(TIME.EQ.START)THEN
       TP=6.718
       tpge=tp
      ELSE
        I=0
        SUMGE=0.
        SUMLATQ=0.
        SUMCOS=0.
        SUMTAMP=0.
        SUMTSAMP=0.
       DO 41 PHID=0.,90.,PHIDINT
        I=I+1
        phi = phiD*(pi/const2)
        SUMTAMP=SUMTAMP + TAMP(I)*COS(PHI)
        SUMTSAMP=SUMTSAMP + TSAMP(I)*COS(PHI)
        SUMGE=SUMGE + GE(I)*COS(PHI)
        SUMLATQ=SUMLATQ + Q(I)*COS(PHI)
        SUMCOS=SUMCOS + COS(PHI)
 41    CONTINUE
       AVG_GE=SUMGE/SUMCOS
       QP=SUMLATQ/SUMCOS
       AVG_TAMP=SUMTAMP/SUMCOS
       AVG_TSAMP=SUMTSAMP/SUMCOS
c     IF(MOD(NINT(-TIME),100).EQ.0)WRITE(70,*)-time,QP
       TP=(QP*(1.-P_ALB)-A1+A2*CLOUD)/(BETA+B1-B2*CLOUD)
       TPGE=(QP*(1.-P_ALB)-AVG_GE-A1+A2*CLOUD)/(BETA+B1-B2*CLOUD)
      END IF
```

```
c       if(mod(-time,50.).eq.0.)WRITE(72,*)-TIME,tp,qp
        IF(MOD(-TIME,100.).EQ.0.)WRITE(6,*)-TIME,AVG_TAMP,AVG_TSAMP,
     +       avg_ge
        if(mod(-time,100.).eq.0.)WRITE(74,*)-TIME,AVG_TAMP
        if(mod(-time,100.).eq.0.)WRITE(79,*)-TIME,AVG_TSAMP
        if(mod(-time,100.).eq.0.)WRITE(71,*)-TIME,AVG_GE

**LOOP FOR LATITUDES**
        I=0
        DO 50 PHID=0.,90.,PHIDINT
        I=I+1

**LOOP FOR SEASONAL VARIATIONS**

          COUNT=0.
          SUMQ=0.
          SUMWEIGHT=0.
          DO 60 lamD= 0.,(360.-LAMDINT),LAMDINT

**WEIGHTING WITH SEASON**

c         IF(LAMD.LT.90.)WEIGHT= .5 + LAMD/180.
c         IF(LAMD.GE.90.)WEIGHT= 1.5-LAMD/180.

        COUNT=COUNT+1.

**ALBEDO VALUES THAT DON'T VARY WITH TIME**

        IF(ABS(PHID).LE.(60.))THEN
          ALBEDO(I)=.32
        END IF
        IF((ABS(PHID).GT.(60.)).AND.(ABS(PHID).LE.(70.)))THEN
          ALBEDO(I)=.32+(phid-60.)*.018
        END IF
        IF(ABS(PHID).GT.(70.))THEN
          ALBEDO(I)=.50+(phid-70.)*.012
        END IF
        P_ALB=.334

** Convert to radians **

       phi = phiD*(pi/const2)
       lam = lamD*(pi/const2)

c...............................................................
         sumc1 = 0.
         sumc2 = 0.
         sumob = 0.

         do 200 j = 1,47

** Mid-month obliquity for a given LATITUDE AND TIME OF YEAR **

            sumob = sumob + (oamp(j)/const1)*(cos(((omnrt(j)*dble(time))
     +                            /const1 + ophase(j))*(pi/const2)))

** sumc1 & sumc2 are used to calculate eccentricity and longitude of the **
** perihelion based on a moving vernal equinox **

            sumc1 = sumc1 + eamp(j)*sin(((emnrt(j)*dble(time))/const1
```

```
     +                    + ephase(j))*(pi/const2))

           sumc2 = sumc2 + eamp(j)*cos(((emnrt(j)*dble(time))/const1
     +              + ephase(j))*(pi/const2))

 ?00      continue

          obliq = sumob + obconst
          long = (atan2(sumc1,sumc2))*(const2/pi)
          eccen = sumc1/sin(atan2(sumc1,sumc2))

** PARAMETERS USED TO CALCULATE MID-MONTH SOLAR INSOLATION **

          delta = asin((sin(lam))*sin(obliq*(pi/const2)))

          nu = lam-((long+const2)*(pi/const2))

          rho = (one-(eccen**sq))/
     +          (one+(eccen*cos(nu)))

** MID-MONTH SOLAR INSOLATION IN CAL/CM^2-DAY **

         IF(ABS(PHI).GE.(PI/TWO-abs(DELTA)))THEN

         if(((phi.gt.(0.)).and.(delta.gt.(0.))).or.((phi.lt.(0.)).
     +      and.(delta.lt.(0.))))then

          SOLRINS=      ((insconst*solrconst)/(rho**sq))
     +        *sin(phi)*sin(delta)
         end if

         if(((phi.lt.(0.)).and.(delta.gt.(0.))).or.((phi.gt.(0.)).
     +      and.(delta.lt.(0.))))then

          SOLRINS = DBLE(0.)
         end if

         ELSE

          Ho = acos(-tan(phi)*tan(delta))

          solrins = ((insconst*solrconst)/(pi*(rho)**sq))
     +        *(Ho*sin(phi)*sin(delta)+cos(phi)*cos(delta)*sin(Ho))
         END IF

**CONVERT SOLAR INSOLATION VALUES FOR EACH LATITUDE**
**IN LATITUDE LOOP TO kcal/cm^2 month AND SUM VALUES OVER
**EACH LATITUDE

         SUMQ=SUMQ + SOLRINS/DBLE(32.88)
c         SUMWEIGHT=SUMWEIGHT+WEIGHT

 60      CONTINUE

**CALCULATE AVERAGE YEARLY SOLAR INSOLATION VALUES FOR EACH
**LATITUDE

         Q(I)=SUMQ/(COUNT)
c        Q(I)=SUMQ/(SUMWEIGHT)
```

```
C       IF(MOD(NINT(-TIME),1000).EQ.0)WRITE(67,*)-TIME,Q(I)

**CALCULATE AVERAGE YEARLY TEMPERATURE FOR EACH LATITUDE INCREMENT
**USING AVERAGE YEARLY GE FROM PREVIOUS YEAR

        TNOGE(I)=(Q(I)*(DBLE(1.)-ALBEDO(I))-A1+A2*CLOUD+BETA*TP)/
     +      (BETA+B1-B2*CLOUD)

        TAMP(I)=(TNOGE(I)-AVGT(I))*AMP + AVGT(I)

        IF(ABS( GE(I)-LASTGE(I) ).LT.GETOL)THEN
        TS(I)=(Q(I)*(DBLE(1.)-ALBEDO(I))-GE(I)-A1+
     +      A2*CLOUD+BETA*TPGE)/(BETA+B1-B2*CLOUD)
        ELSE
        TS(I)=(Q(I)*(DBLE(1.)-ALBEDO(I))-A1+A2*CLOUD+BETA*TP)/
     +      (BETA+B1-B2*CLOUD)
        END IF

        TSAMP(I)=(TS(I)-AVGT(I))*AMP + AVGT(I)

C       WRITE(73,*)-TIME,TSAMP(I)
C       WRITE(80,*)-TIME,TAMP(I)
C       WRITE(81,*)-TIME,TAMP(I)-TSAMP(I)
C       WRITE(72,*)-TIME,GE(I)

**INITIAL SUBSURFACE TEMPERATURES WITH INCLUSION OF GEOTHERMAL GRADIENT
**INITIALIZE PREVIOUS SUBSURFACE TEMPERATURES EQUAL TO PRESENT VALUES
**SET BOTTOM NODE TEMPERATURE TO REMAIN CONSTANT

        IF(TIME.EQ.START)THEN
            TEMP(I,1)=TSAMP(I)
            TLAST(I,1)=TEMP(I,1)
           DO 10 J=2,NODES
            TEMP(I,J)=DBLE(J-1)*DELTAZ*DBLE(.0003) + TSAMP(I)
            TLAST(I,J)=TEMP(I,J)
 10        CONTINUE
        END IF

**REASSIGN AVERAGE YEARLY SUBSURFACE TEMPERATURE TO EARLIER TIME**

        DO 20 J=1,NODES
          T2AGO(I,J)=TLAST(I,J)
          TLAST(I,J)=TEMP(I,J)
 20     CONTINUE

 50     CONTINUE

**AFTER THE FIRST TIME ITERATION CAN CALCULATE CHANGE IN SUBSURFACE
**HEAT STORAGE , GE .

        IF(TIME.GT.START)THEN
                DO 17 I=1,7
                 LASTGE(I)=GE(I)
 17             CONTINUE
        CALL SUBSURF(NODES,C,GE,DELTAT,DELTAZ,T2AGO,TLAST,
     +              TEMP,GRAD)
        END IF

**CALCULATE SUBSURFACE TEMPERATURE DISTRIBUTIONS
```

```
         CALL DEPTH(NODES-1,DELTAZ,deltat,TEMP,TLAST,DT,TSAMP)

 40     CONTINUE
c       write(72,*)
c       write(73,*)
        write(74,*)
        write(71,*)
c       write(70,*)
c       write(67,*)
        write(79,*)
c       write(80,*)
c       write(81,*)
c       CLOSE(UNIT=67)
c       CLOSE(UNIT=70)
c       CLOSE(UNIT=71)
c      CLOSE(UNIT=72)
c      CLOSE(UNIT=73)
      CLOSE(UNIT=74)
      CLOSE(UNIT=79)
c      CLOSE(UNIT=80)
c      CLOSE(UNIT=81)
      STOP
      END

**DEPTH*******************************************************************
* SUBROUTINE USING                                                       *
*  IMPLICIT 1-D TRANSIENT FINITE DIFFERENCE CODE TO FIND TEMPERATURE *
* AT NODES BELOW SURFACE OF EARTH .                                      *
* SOLVES PARABOLIC PARTIAL DIFFERENTIAL EQUATION USING CRANK-            *
* NICOLSON ALGORITHM.                                                    *
*                                                                        *
*  NODES   : INTEGER, THE NUMBER OF DEPTH NODES . [-]                    *
   TEMP(NODES): TEMPERTURE AT NODE . [C]                                 *
   DELTAZ: DISTANCE BETWEEN NODES . [CM]                                 *
*  DELTAT: TIME INCREMENT . [MONTH]                                      *
*  DT: THERMAL DIFFUSIVITY . [CM^2/MONTH]                                *
*  TLAST(NODES): TEMPERATURE AT NODES THE PREVIOUS TIME INCREMENT.[C]*
*  TSURF: TEMPERATURE AT SURFACE NODE FOR THIS TIME INCREMENT.[C]       *
*  TBOT : CONSTANT TEMPERATURE AT BOTTOM NODE . [C]                      *
**************************************************************************

         SUBROUTINE DEPTH(M,H,deltat,TEMP,TLAST,DT,TSURF)

        DOUBLE PRECISION TEMP(1:7,0:M),H,deltat,DT,
     +      TSURF(7),TLAST(1:7,0:M),LAMDA,A(1000,1000),
     +      XO(7,1000),B(1000),X(1000)

        TOTALD=H*REAL(M)
        LAMDA=DT*deltat/(H*H)
        TOL=.000001
        OMEGA=DBLE(1.24)

        DO 3 ILAT=1,7
           DO 4 I=0,M
              XO(ILAT,I)=TLAST(ILAT,I)
              X(I)=0.
 4         CONTINUE
 3      CONTINUE

        DO 60 ILAT=1,7
```

```
          B(1)=(1.-LAMDA)*TLAST(ILAT,1)+ (LAMDA/2.)*(TLAST(ILAT,2)
     +          +TLAST(ILAT,0)+TSURF(ILAT))
          B(M)=.0003*H
          A(1,1)=1.+LAMDA
          A(1,2)=-LAMDA/2.
          A(M,M-1)=-1.
          A(M,M)=1.

          DO 5 I=2,M-1
             B(I)=(1.-LAMDA)*TLAST(ILAT,I)+(LAMDA/2.)*
     +            (TLAST(ILAT,I-1)+TLAST(ILAT,I+1))
             A(I,I-1)=-LAMDA/2.
             A(I,I)=1.+LAMDA
             A(I,I+1)=-LAMDA/2.
 5        CONTINUE

          DO 7 L=1,1000
           BIGDIF=0.

           DO 6 I=1,M
             X(I)= (1.-OMEGA)*XO(ILAT,I)+OMEGA*( B(I)-A(I,I-1)*X(I-1)
     +              -A(I,I+1)*XO(ILAT,I+1) )/A(I,I)

             IF(ABS(X(I)-XO(ILAT,I)).GT.BIGDIF)THEN
              BIGDIF=ABS(X(I)-XO(ILAT,I))
             END IF
  6        CONTINUE
           IF(BIGDIF.LT.TOL)GO TO 50

           DO 8 I=1,M
             XO(ILAT,I)=X(I)
  8        CONTINUE

  7        CONTINUE
           WRITE(6,*)'EXCEEDED MAXIMUM ITERATIONS'

 50       CONTINUE

       DO 500 I=1,M
          TEMP(ILAT,I)=X(I)
 500   CONTINUE

 60    CONTINUE

       END

*SUBSURF***************************************************************
* SUBROUTINE TO DETERMINE CHANGE IN HEAT STORAGE OF EARTH OVER ONE   *
* TIME INTERVAL . DERIVATIVES ARE FOUND USING THREE TEMPERTURE       *
* POINTS AND LAGRANGE COEFFICIENT POLYNOMIALS . THE INTEGRATION IS   *
* PREFORMED USING A SIMPSON COMPOSITE NUMERICAL INTEGRATION SCHEME .*
*                                                                    *
*  C    :HEAT CAPACITY OF EARTH . [KCAL/CM^3-K]                      *
*  TOTALD : TOTAL DEPTH FROM SURFACE TO LOWER CONSTANT TEMPERATURE   *
*           BOUNDARY . [CM]                                          *
*  GE   :CHANGE IN HEAT STORAGE OF EARTH . [KCAL/CM^2-MONTH]         *
*  GRAD   : TEMPERATURE GRADIENT WRT TIME FOR EACH NODE . [C/MONTH]  *
**********************************************************************

       SUBROUTINE SUBSURF(NODES,C,GE,DELTAT,DELTAZ,T2AGO,TLAST,
```

```
     +                TEMP,GRAD)

        DOUBLE PRECISION C,GE(7),DELTAT,DELTAZ,T2AGO(7,NODES),
     +     TLAST(7,NODES),TEMP(7,NODES),GRAD(7,NODES),TOTALD,
     +     ZI0,ZI1,ZI2,ENDA,ENDB,H

        INTEGER NODES

        TOTALD=DBLE(NODES-1)*DELTAZ

C***FIND TEMPERATURE GRADIENT WRT TIME******************************
C**FIRST NODE (SURFACE TEMPERATURE) GRADIENTS ARE NOT TAKEN INTO ACCOUNT
C*******************************************************************

        DO 11 I=1,7
        DO 10 J=2,NODES
          GRAD(I,J)=(-DBLE(1.)/DELTAT)*(-DBLE(1.5)*TEMP(I,J)+
     +             DBLE(2.) *TLAST(I,J)-DBLE(.5)*T2AGO(I,J))
 10     CONTINUE
 11     CONTINUE

C***INTEGRATE TEMPERATURE GRADIENT WRT DEPTH************************

        ENDA=DELTAZ
        ENDB=TOTALD
        M=(NODES-2)/2
        H=(ENDB-ENDA)/DBLE(2*M)
        DO 21 I=1,7
        ZI0=GRAD(I,2)+GRAD(I,NODES)
        ZI1=DBLE(0.)
        ZI2=DBLE(0.)
        DO 20 J=2,NODES-2
        IF(MOD(J,2).EQ.0)ZI2=ZI2+GRAD(I,J+1)
        IF(MOD(J,2).NE.0)ZI1=ZI1+GRAD(I,J+1)
 20     CONTINUE
        GE(I)=C*H*(ZI0 +DBLE(2.)*ZI2 + DBLE(4.)*ZI1)/DBLE(3.)
 21     CONTINUE
        END
```

APPENDIX III

```
C     PROGRAM PLATE.F
C
C         CREATE A DATA FILE TO USE FOR A SPECTRAL CONTOUR PLOT SIMILAR
C         TO THE ONE SEEN IN PALEOCEANOGRAPHY,AUGUST 1990,PG.519.
C
C.....................................................................
      DIMENSION T(10000),X(10000),TB(1000),XB(1000)

      CHARACTER*20INFILE,OUTFILE1,OUTFILE2

      WRITE(6,*)'ENTER NAME OF INPUT FILE'
      READ(5,'(A)')INFILE
      WRITE(6,*)'ENTER NAME OF OUTPUT FILE 1 , SPECTUM DATA'
      WRITE(6,*)
      READ(5,'(A)')OUTFILE1
      WRITE(6,*)'ENTER NAME OF OUTPUT FILE 2 , MEAN AND VARIANCE'
      WRITE(6,*)
      READ(5,'(A)')OUTFILE2

      OPEN(UNIT=21,FILE=INFILE,STATUS='OLD')
      OPEN(UNIT=22,FILE=OUTFILE1,STATUS='UNKNOWN')
      OPEN(UNIT=23,FILE=OUTFILE2,STATUS='UNKNOWN')
C
C.....................................................
C
C
C     READ IN DATA
C
C.................................................................
        WRITE(6,*)'ENTER TOTAL # OF SPECTRUMS DESIRED <27>'
        READ(5,*)NSPECT
        WRITE(6,*)'ENTER INTEGER # FOR BAND WIDTH <200>'
        READ(5,*)NBAND
        WRITE(6,*)'ENTER LENGTH OF SHIFT INTERVAL IN KA'
        READ(5,*)SHIFT
        WRITE(6,*)'ENTER STARTING POINT'
        READ(5,*)START

      DO 50 I=1,10000
         READ(21,*,END=100)T(I),X(I)
         NCOUNT=NCOUNT + 1
 50   CONTINUE
 100  CONTINUE
        BAND=NBAND

        DO 20 J=1,NSPECT
C
C.........................................................
C        FIND MEAN FOR INTERVAL
C
C........................................................
        TMAX=START-(J-1)*SHIFT
        TMIN=TMAX-BAND
        WRITE(6,*)'TMAX=',TMAX,' TMIN=',TMIN
        IA=0
        DO 105 I=1,NCOUNT
        IF((T(I).LE.TMAX).AND.(T(I).GE.TMIN))THEN
        IA=IA+1
        TB(IA)=T(I)
        XB(IA)=X(I)
        ELSE
        END IF
 105    CONTINUE
          SUM=0.
        DO 106 I=1,IA
          SUM=SUM+XB(I)
 106    CONTINUE
        XIA=IA
        XMEAN=SUM/XIA
        WRITE(6,*)'MEAN=',XMEAN
```

```
C.....................................................................
C      FIND VARIANCE
C.................................................................
         SUM2=0.
       DO 150 I=1,IA
          SUM2 = SUM2 + (XB(I) - XMEAN)*(XB(I) - XMEAN)
 150   CONTINUE
       VARI = SUM2 /XIA
         WRITE(6,*)'VARIANCE=',VARI
         WRITE(23,*)VARI
C.................................................................
C
C      PUT INTO FREQUENCY LOOP
C
C.................................................................
       CUTOFF=.07
       INTER=5000
       XINTER=INTER
       DO 200 I=1,INTER
          XI=I
          FREQ = XI / XINTER
          IF(FREQ.GT.CUTOFF) GO TO 300
C...................................................................
C
C         CALL SUBROUTINE TO FIND TAU
C
C.................................................................
          CALL FINDTAU(FREQ,IA,TB,TAU)
C.............................................................
C
C         CALL SUBROUTINE TO FIND PERIODOGRAM
C
C...................................................................
          CALL PERGRAM(VARI,IA,XB,TB,XMEAN,FREQ,TAU,
     +PER)
          WRITE(22,*)FREQ,TMAX-(TMAX-TMIN)/2.,PER
 200      CONTINUE
 300      CONTINUE
         WRITE(6,*)J
         WRITE(23,*)J
  20      CONTINUE
         WRITE(22,*)
          END


       SUBROUTINE FINDTAU(FREQ,NCOUNT,T,TAU)
C.......................................................................
C      CODE TO FIND TAU NEEDED FOR THE LOMB PERIODOGRAM IN SUBROUTINE
C      PERGRAM .
C.......................................................................

       DIMENSION T(NCOUNT)
       REALPI = 3.1415926536
       TOP = 0.
       BOT = 0.
       DO 100 I=1,NCOUNT
          VAR = 4. * REALPI * FREQ
          TOP = TOP + SIN(VAR * T(I))
          BOT = BOT + COS(VAR * T(I))
 100   CONTINUE
       FRAC = TOP / BOT
       TAU = ATAN(FRAC)/VAR
       END


       SUBROUTINE PERGRAM(VARI,NCOUNT,X,T,XMEAN,FREQ,
     +TAU,PER)
```

```
C        CODE TO FIND THE SPECTRUM OF UNEVENLY SPACED DATA USING THE
C     LOMB-NORMALIZED PERIODOGRAM.  SEE REVIEW OF GEOPHYSICS,27,4,NOV.1989,
C     PG 459 .
C
C.........................................................................


      DIMENSION X(NCOUNT),T(NCOUNT)

      REALPI = 3.1414926536
      TOP1 = 0.
      BOT1 = 0.
      TOP2 = 0.
      BOT2 = 0.
      DO 100 I=1,NCOUNT
         TERM = 2.* REALPI * FREQ * (T(I) - TAU)
         TOP1 = TOP1 + (X(I) - XMEAN) * COS(TERM)
         BOT1 = BOT1 + COS(TERM)* COS(TERM)
         TOP2 = TOP2 + (X(I) - XMEAN) * SIN(TERM)
         BOT2 = BOT2 + SIN(TERM) * SIN(TERM)
  100 CONTINUE
      QUANT = TOP1 * TOP1 / BOT1 + TOP2 * TOP2 / BOT2
      PER = QUANT / (2. * VARI)
      END
```