ornl

ORNL/TM-7112

OAK
RIDGE
NATIONAL
LABORATORY

UNION
CARBIDE

A Comparison of the Results of Several Heat Transfer Computer Codes When Applied to a Hypothetical Nuclear Waste Repository

H. C. Claiborne
R. S. Wagner
R. A. Just

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A04; Microfiche A01

DISCLAIMER

ORNL/TM-7112

Contract No. W-7405-eng-26

CHEMICAL TECHNOLOGY DIVISION

A COMPARISON OF THE RESULTS OF SEVERAL HEAT TRANSFER COMPUTER CODES WHEN APPLIED TO A HYPOTHETICAL NUCLEAR WASTE REPOSITORY

H. C. Claiborne
R. S. Wagner*
R. A. Just*

Date Published - December 1979

---

*UCC-ND Engineering.

This report was prepared by Oak Ridge National Laboratory under Contract No. W-7405-eng-26 with the Department of Energy. The project was administered by the Office of Nuclear Waste Isolation, Battelle Memorial Institute.

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

CONTENTS

A COMPARISON OF THE RESULTS OF SEVERAL HEAT TRANSFER COMPUTER CODES WHEN APPLIED TO A HYPOTHETICAL NUCLEAR WASTE REPOSITORY

H. C. Claiborne
R. S. Wagner
R. A. Just

ABSTRACT

A direct comparison of transient thermal calculations was made with the heat transfer codes HEATING5, THAC-SIP-3D, ADINAT, SINDA, TRUMP, and TRANCO for a hypothetical nuclear waste repository. With the exception of TRUMP and SINDA (actually closer to the earlier CINDA3G version), the other codes agreed to within ±5% for the temperature rises as a function of time. The TRUMP results agreed within ±5% up to about 50 years, where the maximum temperature occurs, and then began an oscillatory behavior with up to 25% deviations at longer times. This could have resulted from time steps that were too large or from some unknown system problems.

The available version of the SINDA code was not compatible with the IBM compiler without using an alternative method for handling a variable thermal conductivity. The results were about 40% low, but a reasonable agreement was obtained by assuming a uniform thermal conductivity; however, a programming error was later discovered in the alternative method. Some work is required on the IBM version to make it compatible with the system and still use the recommended method of handling variable thermal conductivity. TRANCO can only be run as a 2-D model, and TRUMP and CINDA apparently required longer running times and did not agree in the 2-D case; therefore, only HEATING5, THAC-SIP-3D, and ADINAT were used for the 3-D model calculations.

The codes agreed within ±5%; at distances of about 1 ft from the waste canister edge, temperature rises were also close to that predicted by the 3-D model.

---

1. INTRODUCTION

The design and analysis of a nuclear waste repository in deep geologic media represents a large computational problem in three dimensions with time dependency. The calculated temperature distributions and histories represent the driving force or input to calculations involving rock mechanics or the thermomechanical response, hydrology,

and waste canister--material rock interactions in the presence of any ambient water at the repository horizon.

Thermal analysis codes (e.g., HEATING5)[1] that have been used extensively in repository calculations have been documented in formal reports that are readily available. However, it was felt that additional documentation was desirable for use in any future licensing hearings concerning a nuclear waste repository. Consequently, it was decided that a direct comparison of the results of calculations with various conduction heat transfer codes be made on simplified models of a hypothetical nuclear waste repository.

The heat transfer codes selected for this study were HEATING5,[1] THAC-SIP-3D,[2] ADINAT,[3] SINDA,[4] TRUMP,[5] and TRANCO.[6]

Time limitations prevented the ANSYS code[7] from being considered for inclusion in this study. However, a recently published comparison of HEATING5 and ANSYS results was made for a conceptual design of a waste repository in salt.[8] The capabilities of ANSYS include: static and dynamic stress analysis with plastic creep, swelling, and both small and large deflections; steady-state and transient heat transfer; and steady-state fluid flow. The matrix displacement method of analysis uses a finite element technique. The results of the two codes are in very good agreement except at the canister-salt interface where HEATING5 produced temperatures that averaged about 25°F higher than ANSYS. This was explained by the use of a different model of the source term. The ANSYS model assumed a uniform volumetric heat generation rate, whereas the HEATING5 model expressed the source as a flux into the surrounding salt. Just a few inches away and beyond, the differences due to the source term were negligible.

The ADINAT and TRANCO codes use the finite element technique of solution and were specifically written to generate the temperature distributions as input to a stress analysis code. TRANCO has been widely used in the rock mechanics calculations involved in nuclear waste repositories. The other four codes use the finite difference method. The SINDA code, or its earlier version, CINDA3G, has been widely used in the aerospace industry.

All of these codes have the capability of solving the time-dependent equation for heat conduction with variable thermal properties; namely,

$$\nabla \cdot k \nabla T + S = c\rho \frac{\partial T}{\partial t} , \qquad (1)$$

where

∇ = geometric operator for gradient

k = thermal conductivity

T = temperature

S = heat source

c = heat capacity

ρ = density

t = time.

A brief description of the techniques used by each code and their capabilities is given in the following section. In the remainder of the report, the two- and three-dimensional models that were used to simulate a repository are described, and detailed comparisons of the calculational results of the codes are made in the form of tables and graphs.

2. DESCRIPTION OF COMPUTER CODES

2.1 HEATING5

HEATING5[1] is the latest version of the "The HEATING Program," where HEATING is an acronym for heat engineering and transfer in nine geometries. HEATING was originally developed by Liguori and Stephenson[9] from Fowler and Volk's generalized heat conduction code, GHT.[10]

HEATING5 is an improved version of HEATING3[11] that has the added capability of solving transient problems involving materials that undergo a change of phase. The HEATING5 program is designed to solve steady state and/or transient heat conduction problems in one-, two-, or three-dimensional Cartesian or cylindrical coordinates or one-dimensional spherical coordinates. The thermal conductivity, density, and specific heat may be both spatially and temperature dependent. The thermal conductivity may be anisotropic. The thermal conductivity, density, and heat capacity can also be time dependent if they are defined in user-

supplied subroutines. Materials may undergo a change of phase. Heat generation rates may be dependent on time, temperature, and position and boundary temperatures may be time dependent. The boundary conditions, which may be surface-to-boundary or surface-to-surface, may be fixed temperatures or any combination of prescribed heat flux, forced convection, natural convection, and radiation. The boundary condition parameters may be time and/or temperature dependent. The mesh spacing can be variable along each axis. The code is designed to allow a maximum of 100 regions, 50 materials, and 50 boundary conditions. The maximum number of lattice points can be easily adjusted to fit the problem and the computer storage requirements. A newer version, labeled "5A," removes these limitations but has not yet been documented. The storage requirements on an IBM 360 computer range from approximately 250K bytes for one lattice point to 1256K bytes for 6000 lattice points.

The point-successive overrelaxation iterative method and a modification of the "Aitken δ^2 extrapolation process" are used to solve the finite difference equations, which approximate the partial differential equations for a steady-state problem.

The transient problems may be solved by using one of the following finite difference schemes.[12] The first is the classical explicit procedure (CEP) which involves the first forward difference with respect to time and is thus stable only when the time step is smaller than the stability criterion. A modification to the CEP is the second scheme, and it requires the temperature distribution at two times to calculate the temperatures at the new time level. The technique is stable for a time step of any size. The third procedure, which is written quite generally, actually contains several implicit techniques that are stable for a time step of any size. One can use the Crank-Nicolson heat balance equations, the classical implicit procedure (CIP) or backwards Euler heat balance equations, or a linear combination of the two. The resulting system of equations is solved by point successive overrelaxation iteration. Techniques have been included in the code to approximate the optimum acceleration parameter for problems involving constant thermal parameters as well as those whose effective thermal conductances and capacitances

vary with time or temperature. The size of the time step may be varied as a function of the maximum temperature change or the maximum percent of relative change in temperature throughout the model.

2.2 THAC-SIP-3D

THAC-SIP-3D[2] is a transient heat analysis code designed to use the strongly implicit procedure[13,14] (SIP) to calculate the temperature distributions for problems that can be modeled in the three-dimensional (3-D) Cartesian coordinate system. This code was developed to reduce the computing time required for large 3-D heat conduction problems associated with nuclear waste repositories. The SIP is highly competitive and, in many cases, is superior to other existing techniques for solving systems of equations arising in the finite difference solution of multi-dimensional heat conduction problems. The method is accurate and faster than most other techniques, especially for problems involving large systems of equations. Its rate of convergence is not strongly dependent on the nature of the coefficient matrix of the system of equations to be solved, thus making the method superior to other techniques for most cases involving nonlinear problems.

In THAC-SIP-3D, the thermal conductivity, density, specific heat, and heat generation may be dependent on position, temperature, and time. The thermal conductivity may be anisotropic. A model may have boundary conditions on the external surfaces which may be adiabatic, a prescribed temperature, or any combination of prescribed heat flux, forced convection, natural convection, and radiation. The boundary temperatures may vary with time, and other boundary condition parameters may be dependent on position, temperature, and time. The mesh spacing may be variable along each axis. The code will allow a model with a maximum of 200 fine lattice planes along each axis, 100 regions, 100 materials, and 25 boundary conditions. The maximum number of nodes can be easily adjusted to fit the problem and the computer storage requirements. The amount of storage required by the code on an IBM 360 or 370 computer varies from approximately 240K bytes for one node to 1500K bytes for 5000 nodes.

The code uses a finite difference scheme that can range from the Crank-Nicolson procedure to the CIP[12] (backwards Euler) to generate the system of equations solved by the SIP to obtain the transient temperature distribution. The size of the time step may be varied as a function of the maximum temperature change or the maximum percent of relative change in temperature throughout the model.

2.3 ADINAT

ADINAT, a proprietary computer code of Massachusetts Institute of Technology, was designed for the automatic dynamic incremental nonlinear analysis of temperatures using finite element analysis. It is a heat transfer program that is compatible with the stress analysis program ADINA[15] and can be used to input the temperature distributions to that program. ADINAT can address heat transfer and field problems in essentially any geometry system or combinations of systems. The orthotropic thermal conductivity may be a function of position or temperature. The heat capacity may be a function of position. The source terms may be a function of position or time. The boundary conditions include prescribed temperature, forced and natural convection, radiation, and prescribed heat flux.

Basically, ADINAT uses different algorithms to solve each of the four cases: linear steady state, nonlinear steady state, linear transient and nonlinear transient. The resulting system of linear equations is solved using a Gaussian elimination method. The system of equations for transient problems is formed using a fully implicit scheme for a single run, and the time step is constant throughout the computations. It can be varied using the restart feature.

To minimize core requirements, ADINAT allocates storage dynamically during different phases of the computation and makes use of overlay features. The maximum size of a model is a function of the available in-core and out-of-core storage. Problems can be restarted at preselected time steps. One feature allows the data to be run through the program to check it without actually solving for the temperatures. ADINAT was written in FORTRAN IV and was developed to run directly on IBM, CDC, and UNIVAC computers.

There is a single degree of freedom (the temperature) associated with each node. One-dimensional (1-D) conduction elements are two-node members allowed arbitrary orientation in the global X, Y, Z system. Two-dimensional (2-D) conduction elements are 4- to 8-node isoparametric quadilaterals which must be input in the global Y-Z plane. These elements can be used in planar or axisymmetric solids. Three-dimensional conduction elements are 8- to 21-node isoparametric or subparametric curvilinear hexahedra. Although the finite element model can approximate the problem as closely as desired, the mesh description can be quite time consuming for all but the simplest of cases. However, several mesh generators are available which can be used to generate this information for many complex geometries, especially for 2-D problems.

2.4 SINDA

The systems improved numerical differencing analyzer (SINDA) program,[4] developed through the efforts of the NASA Manned Spacecraft Center, is a computer software system which possesses capabilities that make it well suited for solving lumped parameter representations of physical problems governed by diffusion-type equations. This code is an expanded and refined version of CINDA3G.[14] A recent version of SINDA is referred to as SINDA-8. CINDA3G was originally written for a Univac computer and has since been generalized to IBM systems, as well as others. In this situation, distribution as well as revision are frequently not well controlled; for example, it is possible to obtain documentation that does not match a user's version of the same code. This can cause severe problems for the user, which is the case for the Oak Ridge National Laboratory (ORNL) version of the program (CINDA3G). For example, none of the documentation presently available mentions the need for a blank card as the first input card to the IBM version.

The special devices peculiar to CINDA also cause a problem when this program is used. For example, it creates preprocessor programs to deal with variable thermal conductivity. This technique is particularly inefficient for salt repository problems and prevents compilation of the CINDA-generated subroutines in the Computer Sciences Division and ORNL

system. It was necessary to bypass such procedures before the program could be executed, because the available documentation gives no other alternative to this preprocessor technique.

To compound the user's problem, the CINDA3G program is extremely versatile and, therefore, inherently difficult to use, particularly with respect to large-scale repository problems. To reduce the complexity and the effort of dealing with input data sets, it is often necessary to write special input generation programs to provide the data on a simpler and less-flexible basis. Changes in repository design in this approach often require that a new generator be written for each case.

The SINDA or CINDA3G system consists of an extensive library composed of many FORTRAN subroutines, which perform various commonly needed actions, and a preprocessor, which accepts instructions from the user and converts them to a FORTRAN computer code that solves the problem. The input to the preprocessor consists of references to members of the SINDA library as well as actual FORTRAN statements. In particular, SINDA can address steady state or transient heat transfer problems defined in essentially any geometry system or combinations of geometry systems. The thermal conductance, thermal capacitance, and source terms may be spatially, temperature, and/or time dependent, and there are many routines available in the library to evaluate these parameters. A heat transfer model may include change of state and mass flow calculations. Boundary conditions include prescribed temperature, forced and natural convection, radiation, and prescribed heat flux, and these parameters can vary in essentially any manner.

The SINDA library contains a large assortment of subroutines that solve the thermal network using different methods. Steady state techniques include block iteration, successive point iteration with extrapolation, and an accelerated successive point iteration method for radiation dominated problems. Each of these techniques can employ relaxation. Transient techniques include the explicit forward differencing method with variations, the explicit exponential prediction method, the duFort-Frankel explicit method, the Crank-Nicholson scheme, the fully implicit procedure, and an implicit method that falls between Crank-Nicholson and fully implicit.

SINDA uses a dynamic storage feature to mimimize core requirements for a given problem. The system provides for storing, retrieving, and editing input data, contains a restart option, and has a multiple case feature to facilitate parametric analyses. A wide assortment of output features are available to assist the user in checking out data and in interpreting results.

Nodes are used for the geometrical description of a SINDA model. The user must supply the thermal capacitance, the initial temperature, the source at each node or group of similar nodes, and the thermal conductance and pair of nodes involved for each connector or group of similar connectors.

The SINDA package is a powerful system, but the user must have a working knowledge of techniques for modeling thermal systems using resistor-capacitor network representations as well as a thorough knowledge of programming techniques using the SINDA and FORTRAN languages. Other concepts in computing are essential to grasping the routine use of SINDA. Due to the generality of SINDA, the input data can be quite tedious for all but the simplest of problems.

2.5 TRUMP

The Lawrence Livermore Laboratory computer code TRUMP[5] (version AE74 3/5) is a general, nonlinear parabolic, partial differential equation solver designed to address problems involving flow in various kinds of potential fields such as heat flow in temperature fields, mass flow in pressure fields, and current in electrical and magnetic fields. Simultaneously, it will solve two additional equations representing, in thermal problems, heat production by decomposition of two reactants having rate constants with a general Arrhenius temperature dependence. Steady state and transient flow in one, two, or three dimensions are considered in geometrical configurations having simple or complex shapes and structures. Material properties, source and sink strengths, boundary conditions, and other problem parameters may vary with spatial position, time, or the primary dependent variable (temperature, pressure, field strength). External sources or sinks, coupled to the system by means of

specified boundary conditions, may vary with time. Certain problem parameters at one spatial location may be made to depend on the value of the dependent variable at another spatial location. Initial conditions may vary with spatial position.

The boundary condition parameters may be time and/or temperature dependent. A material-properties library is available for the TRUMP code, which contains thermal properties data in the gram-centimeter-second-calories-°C, the kilogram-meter-second-joule-°C, and the kilogram-meter-second-joule-K systems.

Among the criteria that may be specified for ending a problem are upper and lower limits on the size of primary dependent variable, upper limits on the problem time, the number of time steps, or the computer time, and attainment of steady state. Solutions may be obtained by use of explicit or implicit difference equations, or by an optimized combination of both. The steady state solution is obtained by stepping through the transient. The time-step size may be computed by TRUMP as a function of the temperature change over the time interval.

All arrays whose lengths are a function of the number of parameters in the input data are variably dimensioned to allow the user to tailor the core requirements for each problem. TRUMP contains restart options and has features that allow multiple cases to facilitate parametric analyses. Several output features are available to assist the user in checking out data and in interpreting results.

Nodes are also used for the geometric description of a TRUMP model. In addition to supplying all of the thermal property data, the user must supply the length, width, and height of each node, and the length, width, and height of each thermal connector and the nodes defining the connector. This information only needs to be supplied once for each group of nodes or connectors, which are identical. Preparation of the input data can be quite tedious for all but the simplest of cases. However, several mesh generators are available to produce this information.

2.6 TRANCO

TRANCO[6,17] is a transient heat conduction code that uses the finite element method in the solution of the heat conduction equation. It has been used extensively to determine the temperature distributions for input to codes used in the analysis of the thermomechanical response of geologic media containing a nuclear waste repository. The name of the code has recently been changed to SPECTROM-41[18] with the addition of minor improvements.

Arbitrarily shaped and plane boundary value problems in heat transfer may be solved with TRANCO by utilizing eight-noded isoparametric elements and six-noded subparametric triangular elements, or a combination of both. Only 2-D X-Y and axially symmetric R-Z thermal problems can be solved for the following specific conditions:

1. Boundary conditions
   a. Constant temperature
   b. Constant flux
   c. Convective
   d. Adiabatic
2. Time dependent or constant internal heat generation
3. Anisotropic thermal conductivities.

In addition, an initial spatially dependent temperature field may be specified; otherwise, zero initial temperatures are assumed throughout the body under consideration.

In the solution, a variational principle is applied to the heat conduction equation. The first variation yields a set of linear matrix equations that the program solves for temperature by a direct solution technique.

The direct solution technique consists of a step-by-step integration procedure through time. The technique is very efficient for constant material properties because the global conductivity matrix is upper triangularized (in the usual Gauss-Doolittle fashion) during the first time step, and it is used throughout the total specified time interval with updates on the flux and temperature. The efficiency of the calculation, however, is reduced considerably when the material properties are a function of temperature.

3. CALCULATIONAL MODELS

A simplified model of a hypothetical high-level waste repository was developed in both 2- and 3-D coordinates for the test calculations. The model represents a repository of room and pillar construction having the room floor at a depth of 2000 ft below the surface. Sixty-foot wide pillars separate very long rooms that are 25 ft high and 20 ft wide. Thermal loading was assumed to be 187 kW/acre, with waste canisters buried in ∿16-in.-diam vertical holes in the room floor. However, because of the problem of fitting boundaries to coordinate surfaces, the model assumes that the hole is a square area with 16-in. sides. For the 2-D model this is represented by an infinitely long, 16-in.-thick slab. The waste was assumed to be uniformly distributed within the hole for a length of 10 ft and to extend to within 8 ft of the floor. A concrete plug was assumed to exist between the top of the waste canister and the floor level. The model further assumes that soil or rock constant thermal properties exist from the surface to a depth of 1000 ft and from 3000 to 10,000 ft. Salt occupies the space between 1000 ft and 3000 ft. The temperatures were maintained at 60 and 160°F at the surface and the 10,000-ft-deep bottom of the model, respectively, with the initial temperature varying linearly between the vertical extremities of each zone.

Radiant heat transfer was considered from the floor to the ceiling of the room with the latter being modeled as a void (i.e., conduction and convection heat transfer were neglected). It was estimated that the black-body view factor was 0.35 and the emissivity of the salt and concrete were 0.9, which produces a gray-body shape factor of 0.325.

The ADINAT and the THAC-SIP-3D codes do not permit modeling internal radiative heat transfer. Consequently, when these codes were used, the radiative heat transfer was modeled by an approximately equivalent heat conduction model. This was done by assuming that the room air was replaced by a hypothetical material of very low heat capacity, and its thermal conductivity was determined as follows:

$$q_{radiation} = F\sigma(T_1^4 - T_2^4)$$

$$q_{conduction} = k\frac{\partial T}{\partial X} \simeq k\frac{\Delta T}{\Delta X} = k\frac{T_1 - T_2}{\Delta X} ,$$

where

σ = Stefan-Boltzman constant

F = effective view factor.

Therefore, if $q_{conduction}$ is to be equivalent to $q_{radiation}$

$$F\sigma(T_1^4 - T_2^4) = k\frac{T_1 - T_2}{\Delta X} .$$

Performing the indicated algebraic operations yields

$$k = \frac{\Delta X \ F\sigma \ (T_1^2 + T_2^2) \ (T_1 + T_2) \ (T_1 - T_2)}{(T_1 - T_2)} .$$

Replacing T_1 and T_2 by an average temperature T gives

$$k = 4\Delta X \ F\sigma \ T^3.$$

For ΔX = 25 ft and T = 660°R, the equivalent conductivity for the hypothetical material is

$$k = 16 \ \frac{Btu}{hr - ft - °R} .$$

3.1 Two-Dimensional Model

The two-dimensional model uses Cartesian coordinates with symmetry about the vertical axis (z-axis) with respect to the x-axis and is infinite in extent in the other horizontal direction (y-axis). For an infinite array of the previously described room and pillar arrangement, adiabatic boundary occurs at x = 0 (canister center) and x = 40 (pillar midplane). A schematic representation of the model is shown in Fig. 1. The grid lines, which run the length and width of the model are shown in Table 1. The intersections of the grid lines are the nodes (640) at which the temperatures are calculated.

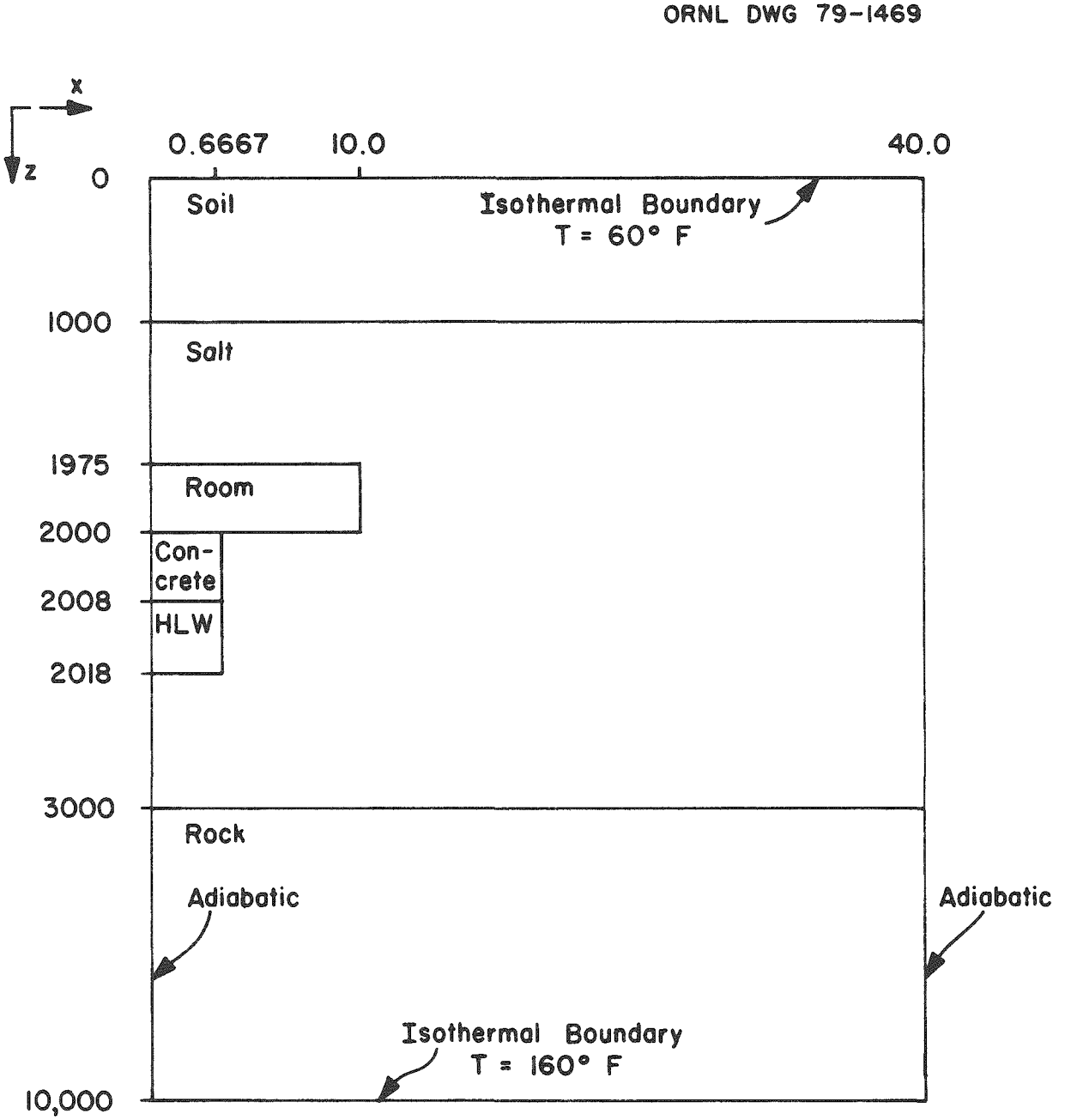

Fig. 1. Two-dimensional x-y cartesian model of hypothetical high level waste repository.

Table 1. Grid lines for the two-dimensional model

Horizontal coordinate x (ft)		Vertical coordinate z (ft)				
0		0	1925	2018	2500	9,000
0.6667	25	100	1950	2024	2625	10,000
2		200	1975	2030	2750	
4	30	500	1987.5	2040	2875	
6		1000	2000	2050	3000	
8	35	1500	2004	2075	4000	
10		1650	2008	2100	5000	
12.5	40	1800	2010	2150	6000	
15		1900	2012	2200	7000	
17.5			2014	2300	8000	
20			2016	2400		

3.2 Three-Dimensional Model

To reduce the size of the three-dimensional (3-D) problem, it was necessary to limit the number of grid lines as listed in Table 2 for a typical x-z plane. To save computer time, the comparison of the results was limited to times $\leq$400 years for the 3-D model. To generate the 3-D model, four x-z planes were incorporated into the 2-D model at y = 0, 0.6667, 1.5, and 4.0 ft. The planes at y = 0 and y = 4.0 ft are adiabatic boundaries and define the unit cell. [See Fig. 1 for a plan view (x-z plane) of the model.] The intersections of the grid lines generate 1720 nodes at which temperatures are calculated. Table 3 shows boundary coordinates for each region for z = 0 at the surface. With the canisters on an 8-ft pitch, each canister is considered to have a 2.75-kW heat

Table 2. Grid lines for the three-dimensional model

Horizontal coordinate x (ft)	Vertical coordinate z (ft)		y (ft)
0.0	0.0	2,024.0	0.0
0.6667	100.0	2,030.0	0.6667
2.0	200.0	2,040.0	1.5
4.0	500.0	2,050.0	4.0
6.0	1000.0	2,075.0	
8.0	1500.0	2,100.0	
10.0	1650.0	2,150.0	
15.0	1800.0	2,200.0	
25.0	1900.0	2,300.0	
40.0	1925.0	2,400.0	
	1950.0	2,500.0	
	1975.0	2,625.0	
	1987.5	2,750.0	
	2000.0	2,875.0	
	2004.0	3,000.0	
	2008.0	4,000.0	
	2010.0	5,000.0	
	2012.0	6,000.0	
	2014.0	7,000.0	
	2016.0	8,000.0	
	2018.0	9,000.0	
		10,000.0	

Table 3. Coordinate boundaries for each region

Material	Lower x	Upper x	Lower y	Upper y	Lower z	Upper z
Soil	0.0	40.0	0.0	4.0	0.0	1,000.0
Salt	0.0	40.0	0.0	4.0	1000.0	1,975.0
Air	0.0	10.0	0.0	4.0	1975.0	2,000.0
Salt	10.0	40.0	0.0	4.0	1975.0	2,000.0
Concrete	0.0	0.6667	0.0	0.6667	2000.0	2,008.0
HLW	0.0	0.6667	0.0	0.6667	2008.0	2,018.0
Salt	0.6667	40.0	0.0	4.0	2000.0	2,018.0
Salt	0.0	40.0	0.0	4.0	2018.0	3,000.0
Rock	0.0	40.0	0.0	4.0	3000.0	10,000.0
Salt	0.0	0.6667	0.6667	4.0	2000.0	2,018.0

source distributed uniformly throughout the 16-in.-sq by 10-ft-long volume of the hole in the floor. This produces 187 kW/acre.

3.3 Thermal Properties and Heat Source Terms

The thermal properties of the materials that were used in the calculations are shown in Table 4.

The unit volume, heat source term as a function of time that was used in the 2-D calculation is shown in Table 5. This is equivalent to 2.75 kW (the power produced in one canister) distributed uniformly throughout the 16 in. x 8 x 10 ft source region. The corresponding source terms for the 3-D calculations are a factor of 6 higher because the source volume is only 16 x 16 in. x 10 ft.

For times between the tabulated values, linear interpolation was used in the calculations. These heat generation rates are essentially

Table 4. Thermal properties of materials

Material	Temp (°F)	Thermal conductivity, (Btu/hr•ft•°F)	Density, (lb_m/ft^3)	Heat capacity, (Btu/lb_m°F)
Soil		1.0	120.0	0.2
Concrete		0.5	125.0	0.2
Salt	32	3.53	135.0	0.2
	122	2.90		
	212	2.43		
	302	2.08		
	392	1.80		
	482	1.60		
	572	1.44		
	662	1.33		
	752	1.20		
Glass		0.3	200.0	0.22
Rock		1.0	125.0	0.2

Table 5. Heat generation rate of a 187-kW/acre[a] thermal loading for a two-dimensional model

Time[b]	F(t)[c]	Time[b]	F(t)[c]	Time[b]	F(t)[c]	Time[b]	F(t)[c]
0	88.08	15	53.36	30	36.78	180	1.871
1	83.25	16	52.00	40	28.95	190	1.652
2	79.32	17	50.70	50	22.89	240	1.054
3	76.03	18	49.43	60	18.18	290	0.830
4	73.18	19	48.21	70	14.49	390	0.658
5	70.67	20	47.02	80	11.60	490	0.563
6	68.40	21	45.86	90	9.332	590	0.489
7	66.32	22	44.74	100	7.548	690	0.427
8	64.39	23	43.64	110	6.144	790	0.374
9	62.58	24	42.58	120	5.036	890	0.330
10	60.87	25	41.55	130	4.162	990	0.291
11	59.24	26	40.54	140	3.471	1990	0.115
12	57.68	27	39.57	150	2.924	2990	0.076
13	56.18	28	38.61	160	2.490	3990	0.065
14	54.73	29	37.67	170	2.145	4990	0.059

[a] Based on PWR waste buried 10 years after reprocessing.

[b] Time after burial in years.

[c] Tabular heat generation function used in codes, Btu/hr·ft^3.

linear for small time changes with a slight concavity upwards when plotted on semilog paper, and this procedure leads to a slight overestimate of the heat generation rates. However, since this is a comparison study, the error is of no significance.

4. BASIC ERRORS IN ANALYSIS

Several types of errors can arise when any code is used; these include:[19]

Modeling errors. Occur from the use of inaccurate material properties, inaccurate initial and boundary conditions, other approximations used in modeling the real system, and interpolation errors in evaluating tabulated functions.

Spatial truncation errors. Occur from the subdivision of the system into discrete volume elements, or nodes, for which average values of spatially dependent variables must be estimated, and for which inaccuracies in volumes, areas, and distances may arise.

Time truncation errors. In transient calculations these can result from the use of discrete time steps for which average values of time-dependent variables must be estimated.

Temperature truncation errors. Occur from the discrete temperature changes that occur in each node in each time step, for which average values of temperature-dependent variables must be estimated.

Convergence errors. Occur from the use of an iterative method of solving the heat-transport difference equations.

Arithmetic truncation errors. Occur from accumulation of round-off errors and from the loss of significant figures that result when numerical values of widely differing magnitudes are added to, or numerical values of similar magnitude are subtracted from, each other.

In an attempt to minimize differences that can occur due to these errors, identical nodal locations were used in all the codes for the 2-D calculations. This was not possible in the 3-D calculations. Although this does not eliminate modeling and spatial truncation as a source of error, it does produce identical modeling and spatial truncation errors in the results of the different computer programs for the 2-D calculations.

Time and temperature truncation errors and differences in convergence criteria are the primary identified sources of differences between the codes.

HEATING5 allowed the use of a geometrically increasing time step, but ADINAT, SINDA, and TRANCO (SPECTROM41) used fixed time steps over each major time interval. The time step could be changed at the end of each major time interval at the user's choice. In TRUMP, the time step can only be indirectly controlled and essentially is determined by the code to satisfy convergence criteria. The use of different time steps between codes introduced some differences from the evaluation of the time-dependent heat generation function.

The methodology for evaluation of temperature-dependent properties and, hence, the solution of the nonlinear heat transfer equation, varies in the type of iterative scheme used at each time step.

The differences in convergence criteria are also sources of possible differences in this analysis. Each of the codes has its own established methods of determining convergence for the solution technique used at each time step, and this is not believed to be a large source of error.

5. CALCULATED RESULTS AND EVALUATION FOR THE TWO-DIMENSIONAL MODEL

The HEATING5 results were chosen as the reference for comparison of the results.

In Figs. 2-9, the temperature predicted by each code is plotted as a function of time using a nonlinear time axis as shown (except CINDA results, which are discussed later). The true time scale is shown in Table 6. The physical location of the nodes plotted can be determined from Table 7 and are shown in Fig. 1. Examination of the first set of plots and Figs. 10-17 leads to the following observations:

1. The maximum temperatures predicted by each code occur at the same printout time.
2. Typically, the largest separation, or error, is seen at the time the maximum temperature is seen; the exception is TRUMP, which begins an oscillatory behavior at that time.

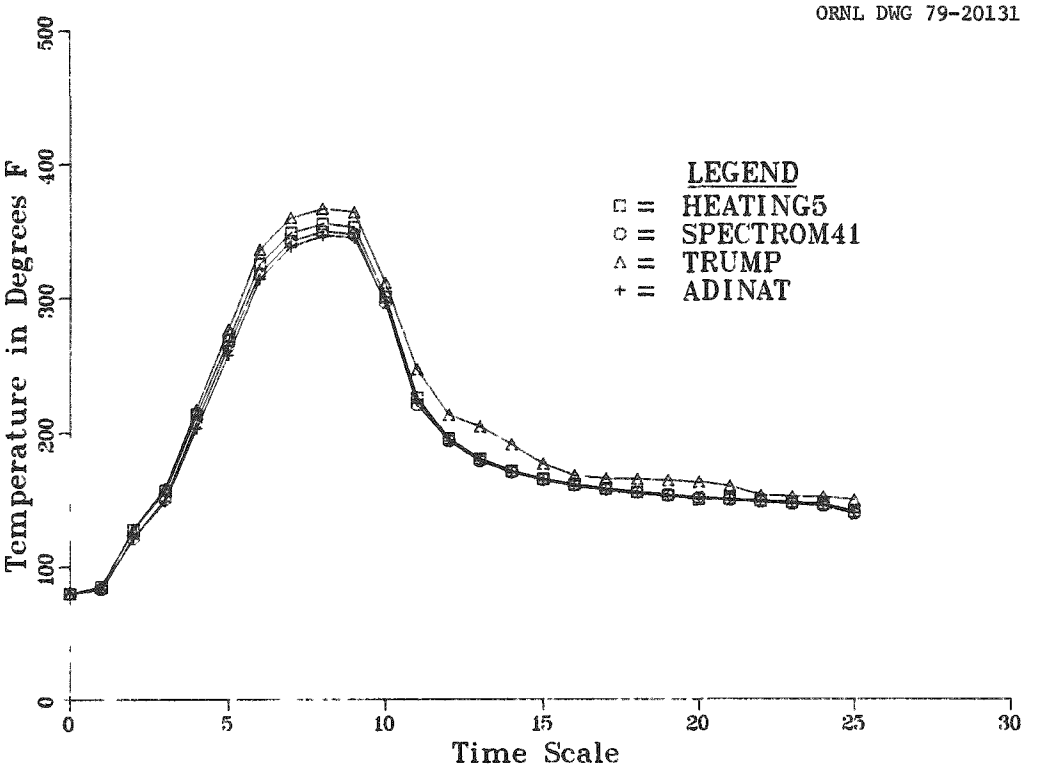

Fig. 2. Temperature vs time for location 1.

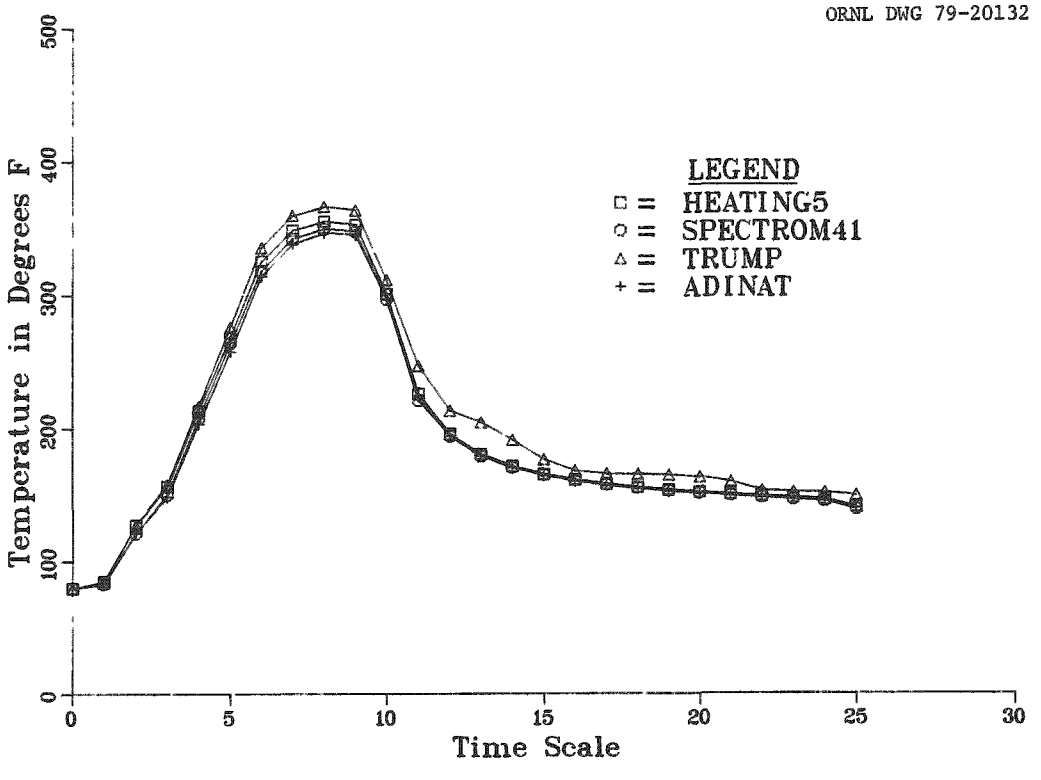

Fig. 3. Temperature vs time for location 2.

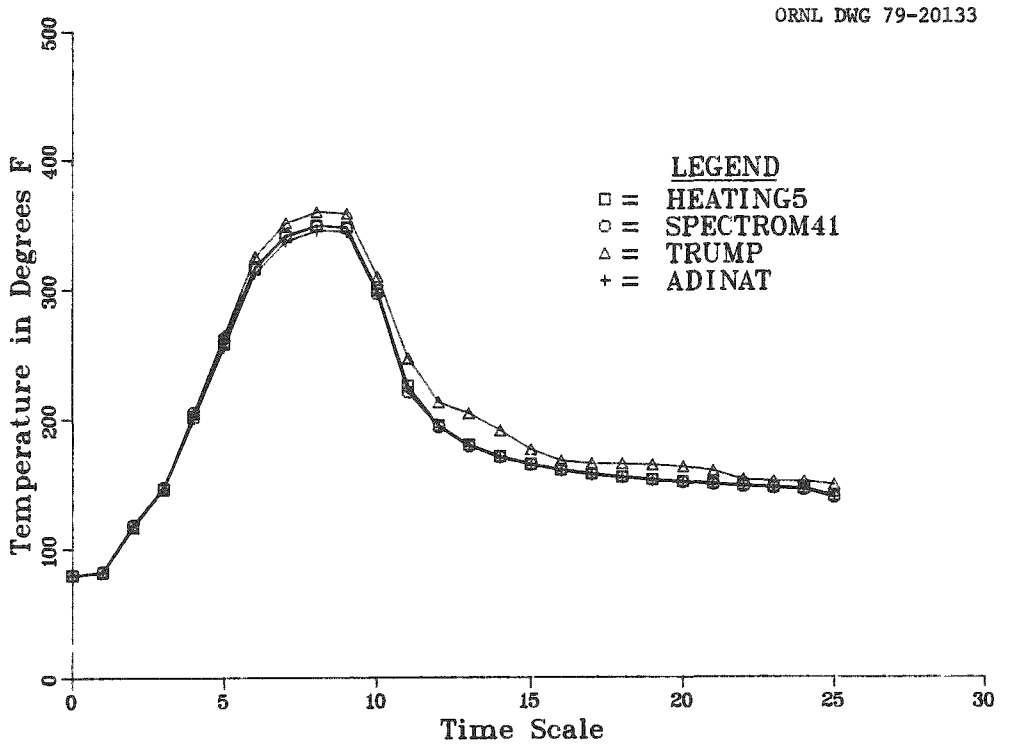

Fig. 4. Temperature vs time for location 3.

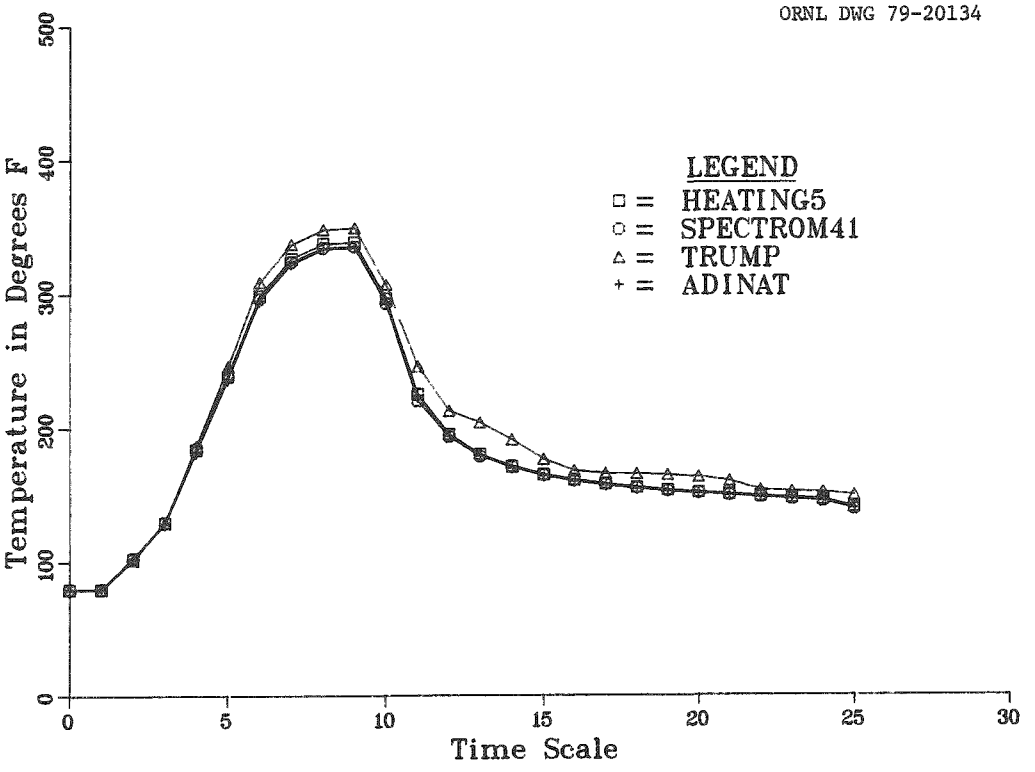

Fig. 5. Temperature vs time for location 4.

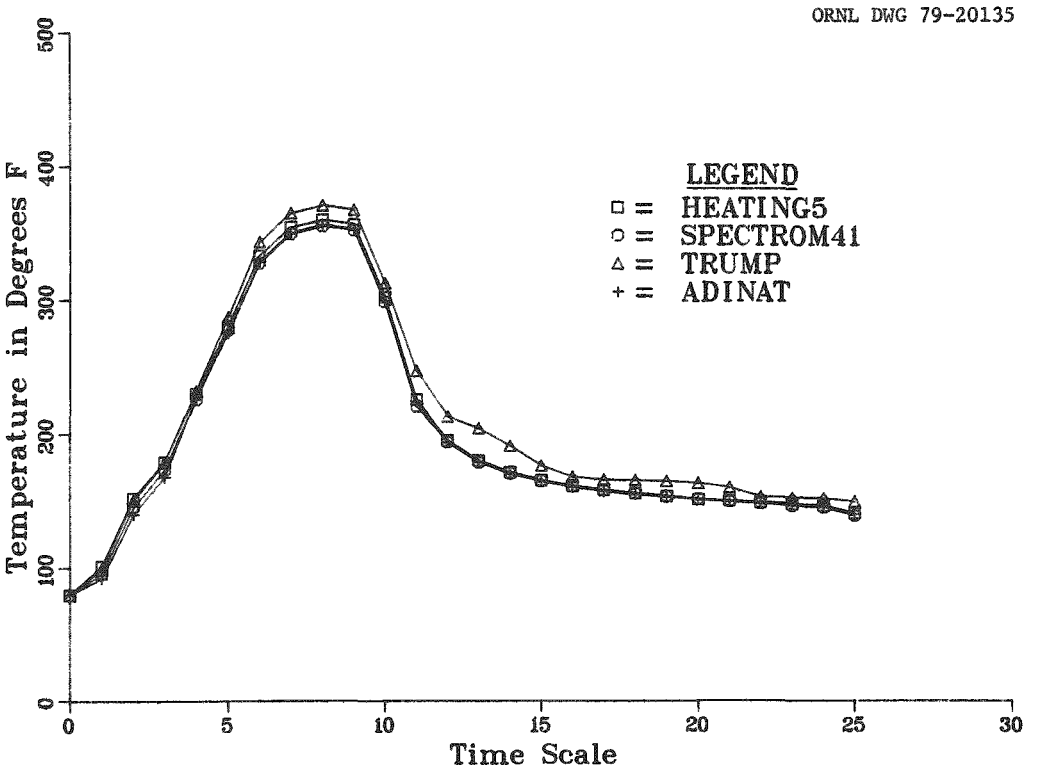

Fig. 6. Temperature vs time for location 5.

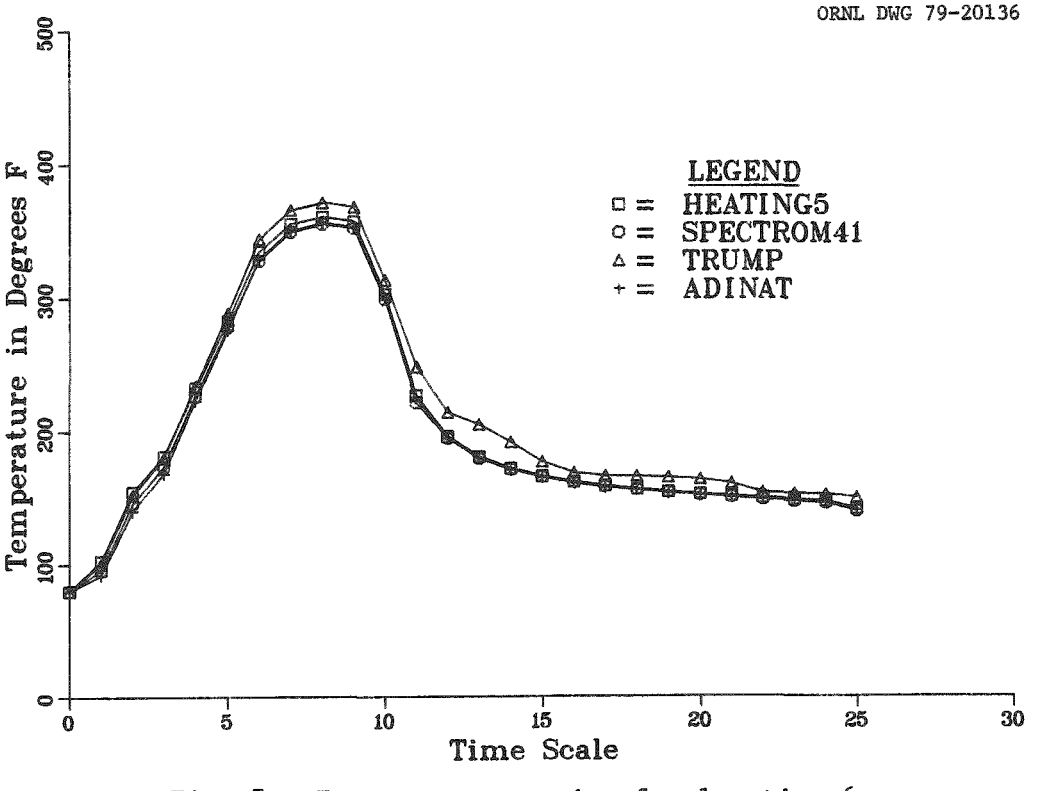

Fig. 7. Temperature vs time for location 6.

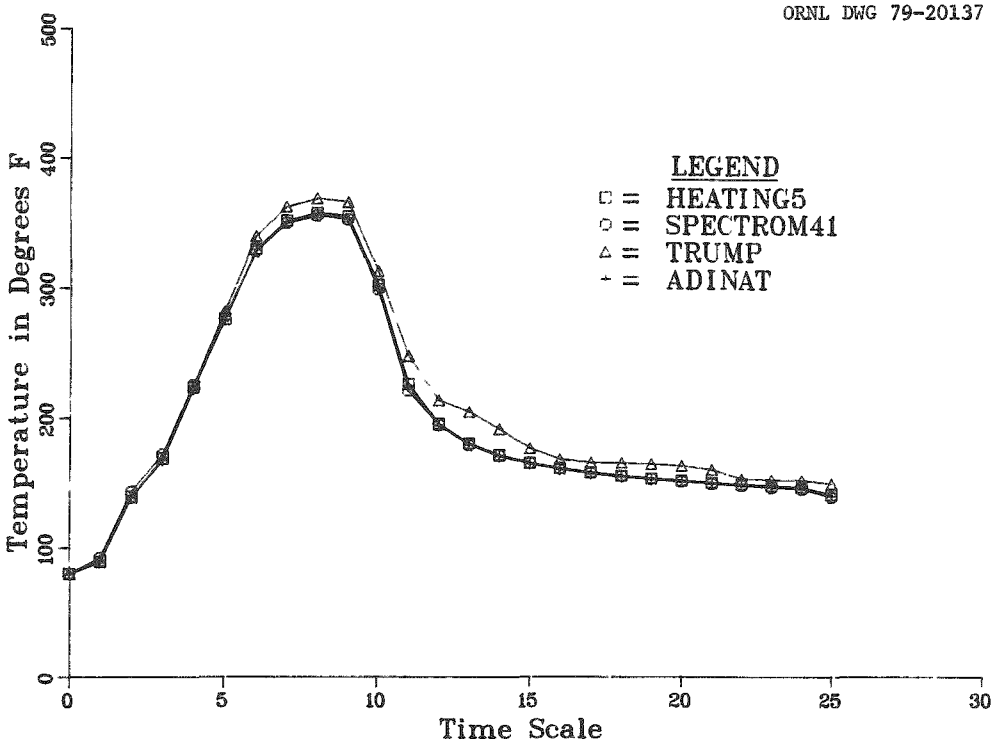

Fig. 8. Temperature vs time for location 7.

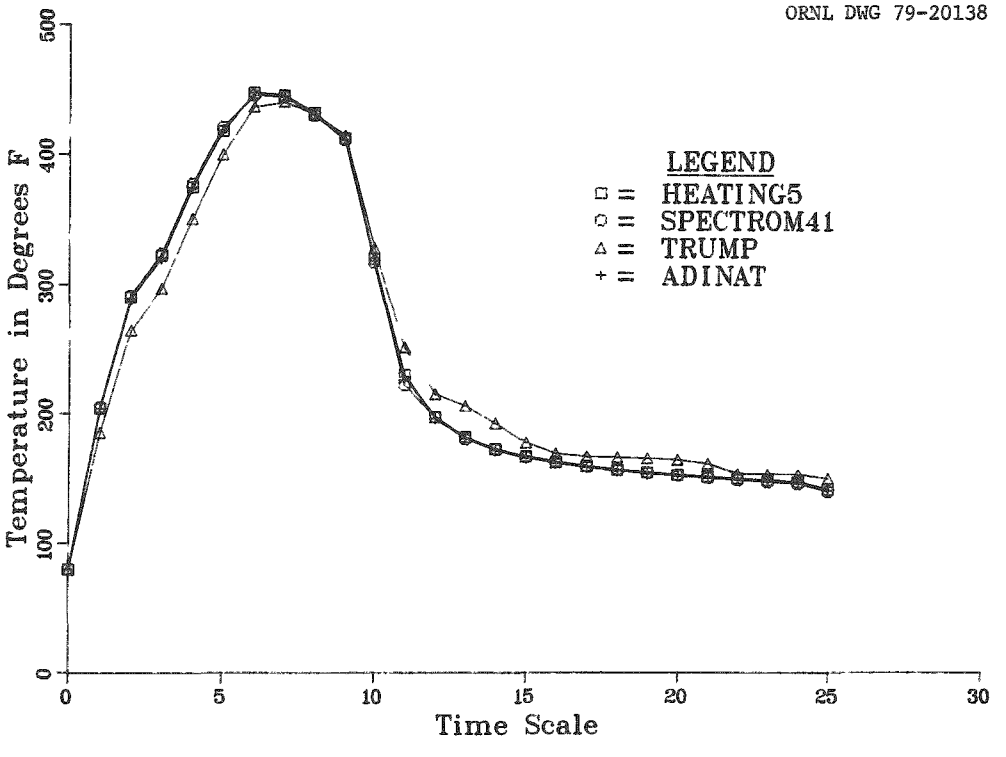

Fig. 9. Temperature vs time for location 8.

Table 6. Scale and real-time equivalence

Time scale	Output time (years)
0	0.0
1	0.1
2	1.0
3	2.0
4	5.0
5	10.0
6	20.0
7	30.0
8	40.0
9	50.0
10	100.0
11	200.0
12	300.0
13	400.0
14	500.0
15	600.0
16	700.0
17	800.0
18	900.0
19	1000.0
20	1100.0
21	1200.0
22	1300.0
23	1400.0
24	1500.0
25	2000.0

Table 7. Node locations

Location No.	Physical coordinates in model x (ft)	z (ft)
1	0.0	1975
2	2.0	1975
3	10.0	1975
4	40.0	1975
5	0.0	2000
6	2.0	2000
7	10.0	2000
8	0.6667	2104

ORNL DWG 79-20139

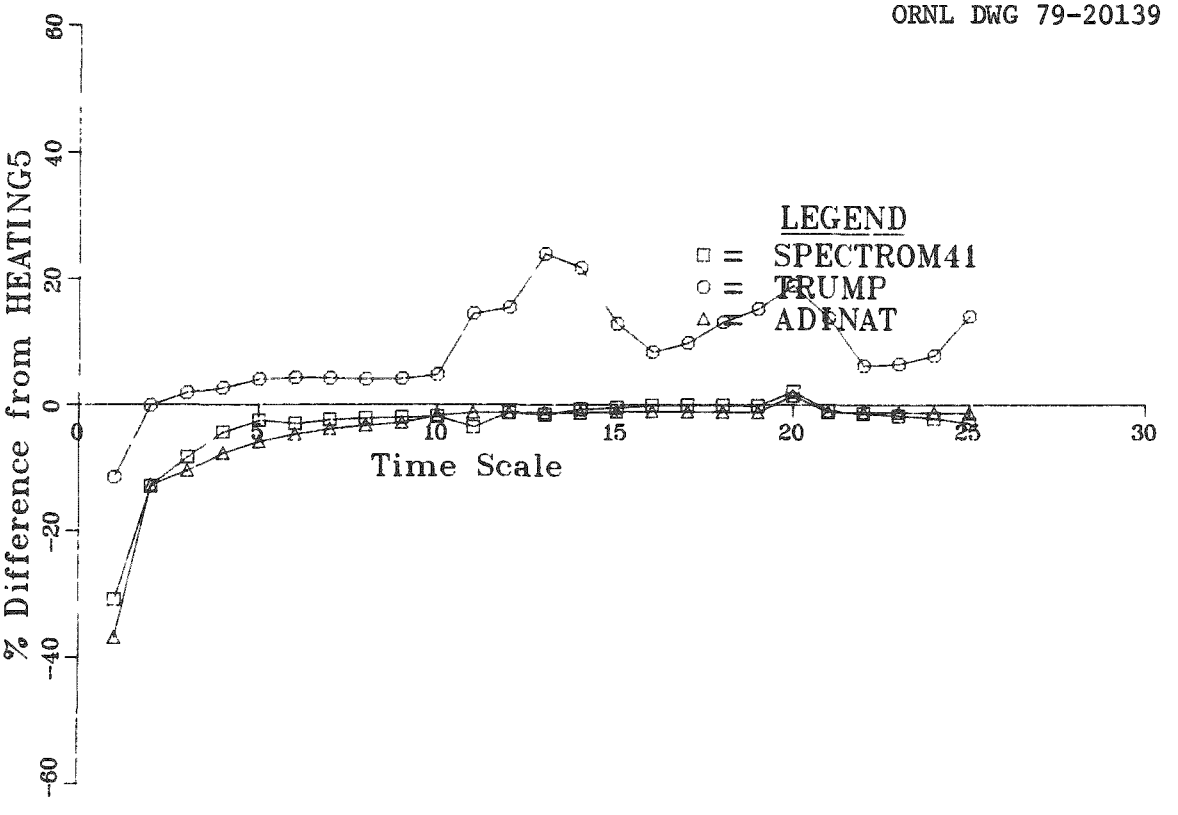

Fig. 10. Temperature differences for location 1 (see Table 6).

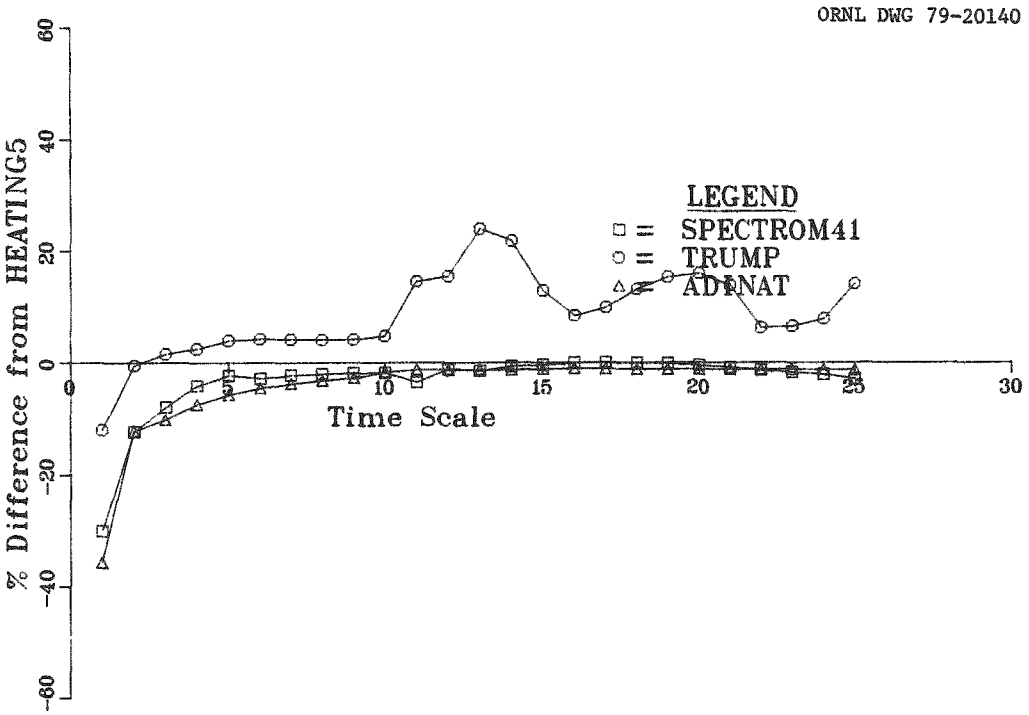

Fig. 11. Temperature differences for location 2 (see Table 6).

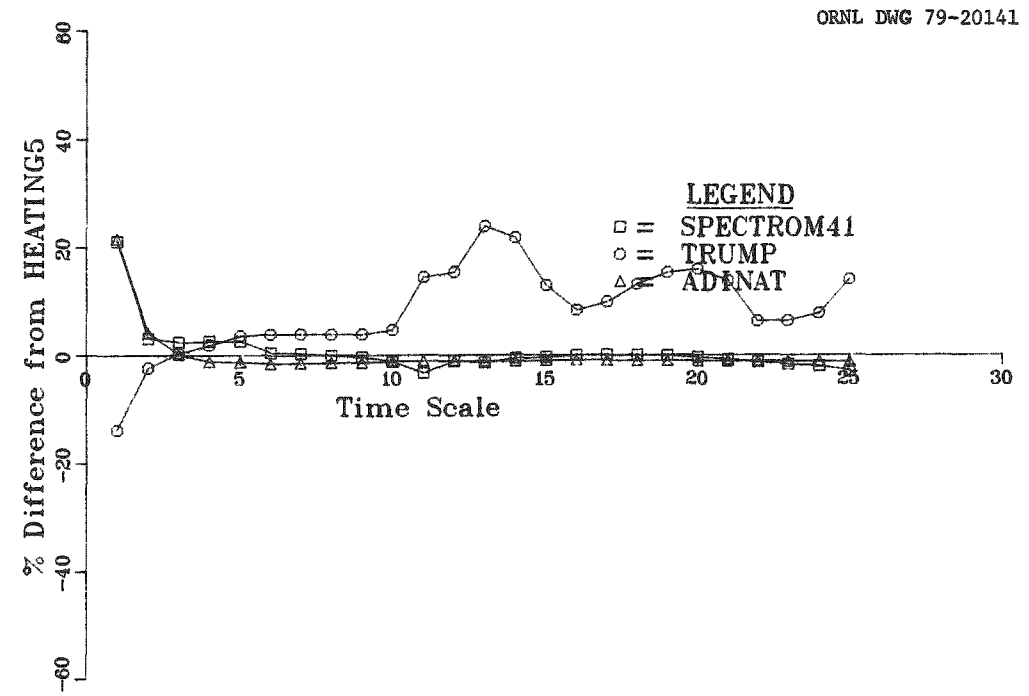

Fig. 12. Temperature differences for location 3 (see Table 6).

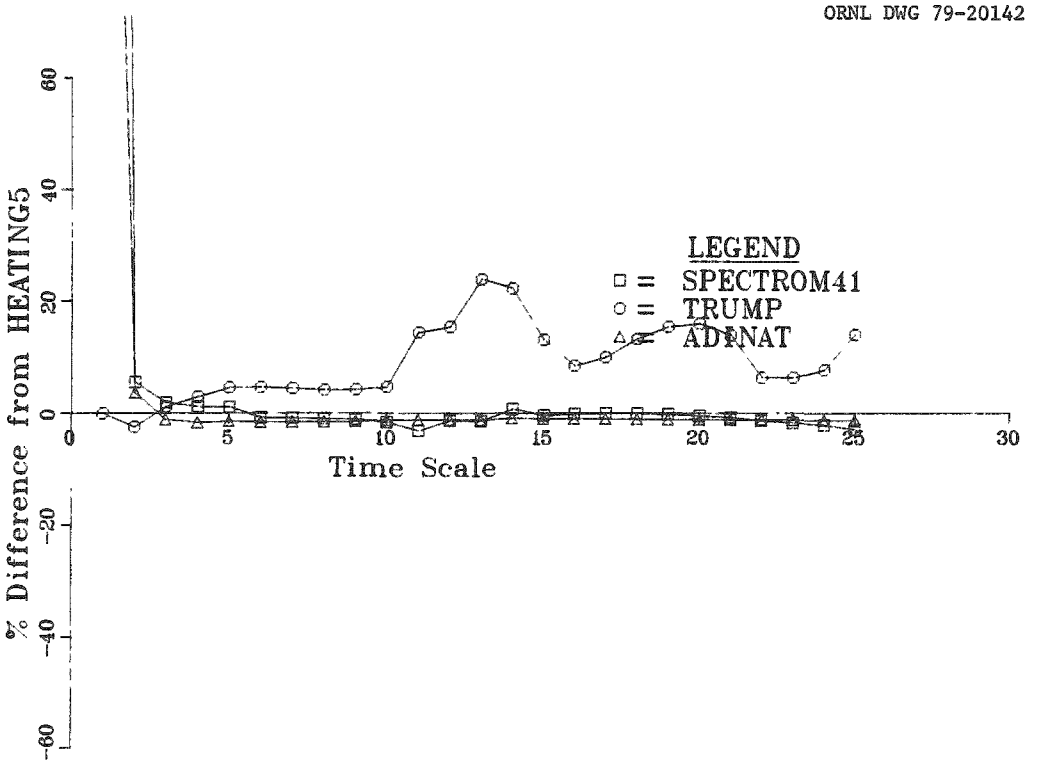

Fig. 13. Temperature differences for location 4 (see Table 6).

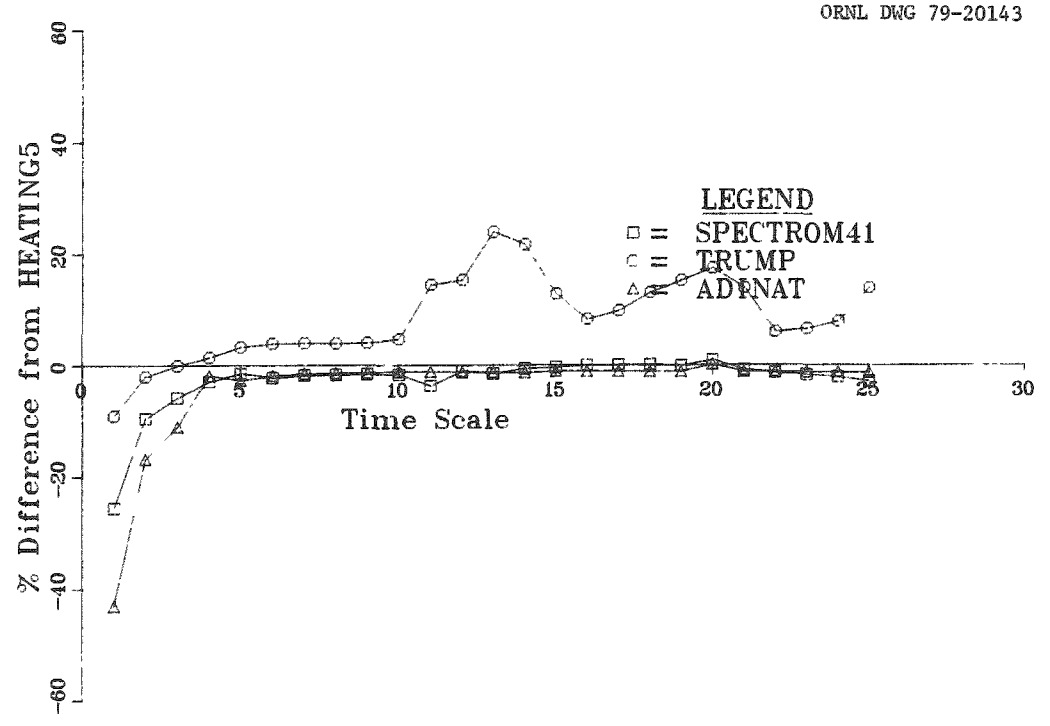

Fig. 14. Temperature differences for location 5 (see Table 6).

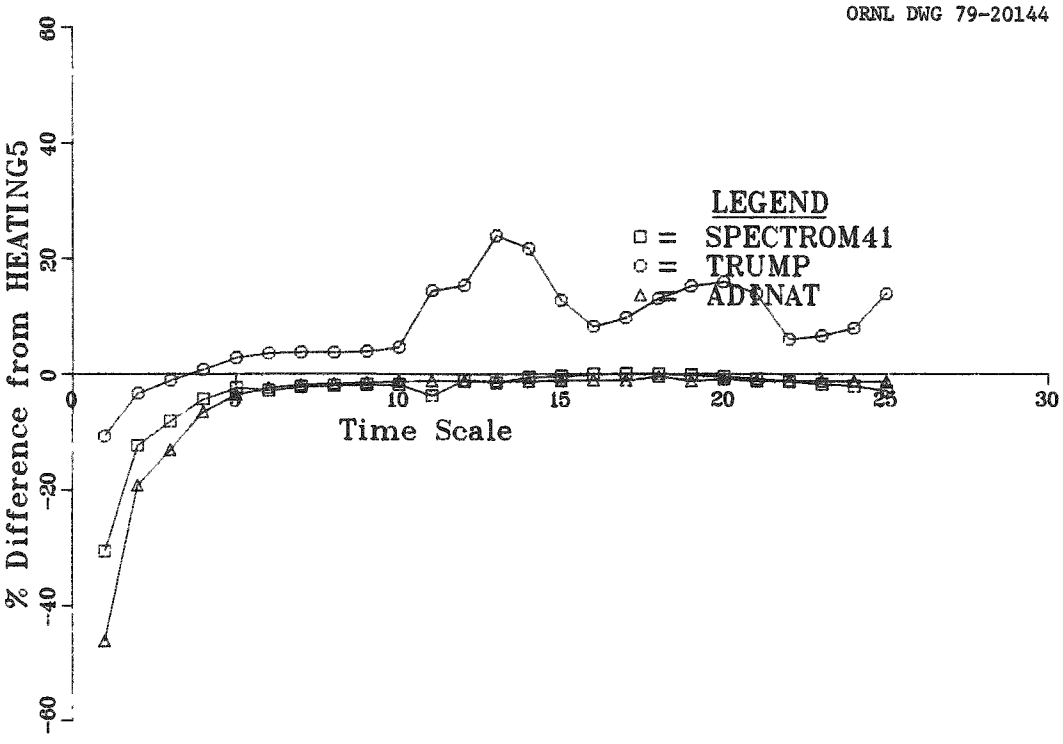

Fig. 15. Temperature differences for location 6 (see Table 6).

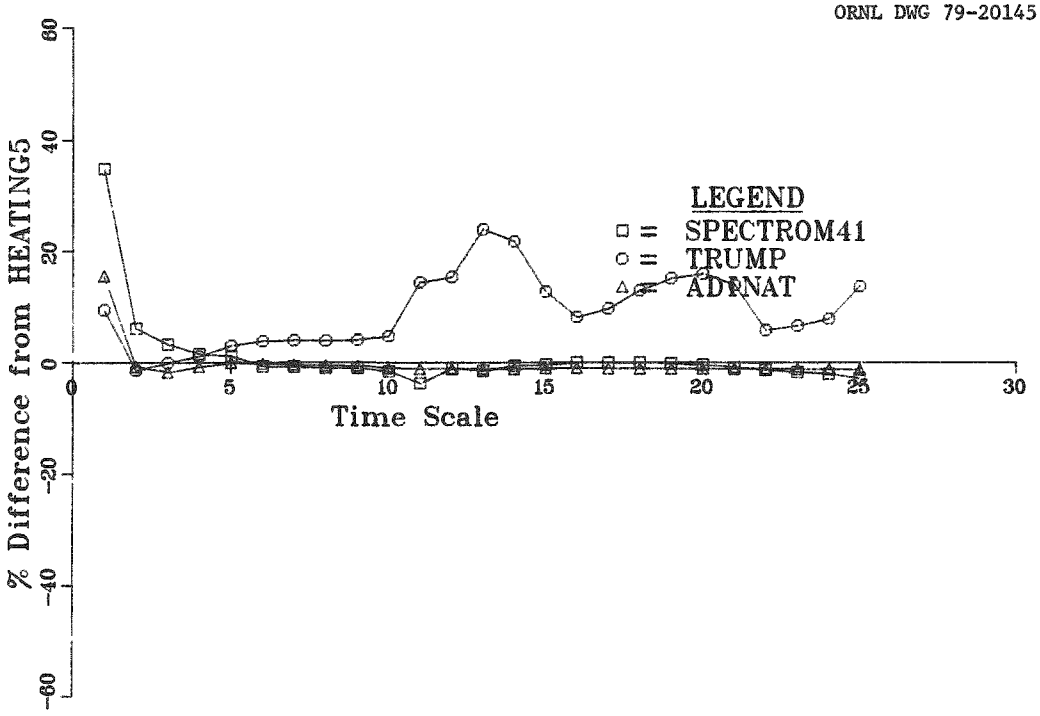

Fig. 16. Temperature differences for location 7 (see Table 6).

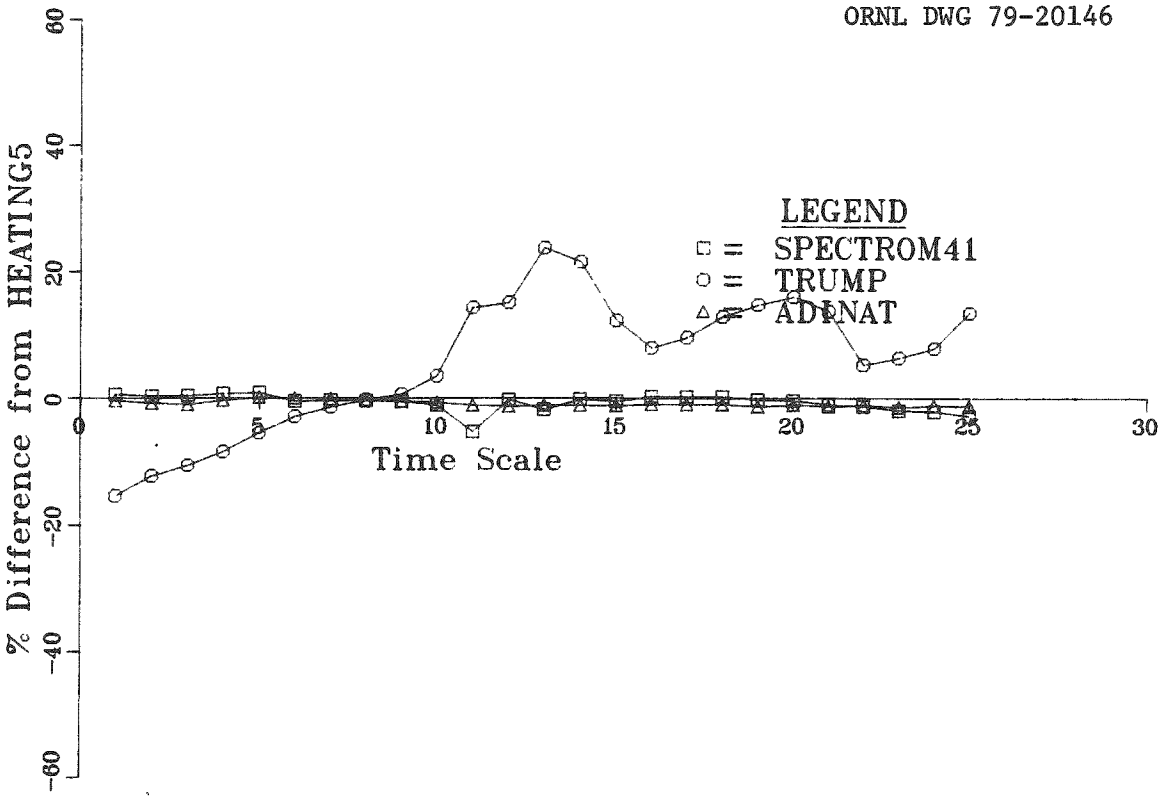

Fig. 17. Temperature differences for location 8 (see Table 6).

3. TRUMP predicts a higher temperature farther from the source and a lower temperature closer to the source than do the other codes.

The ORNL version of the SINDA code could not be run through the IBM compiler without using an alternative method for handling variable thermal conductivity than that recommended. On close examination of the code listing, it became apparent that the version was closer to the original CINDA3G even though the listing had the title of SINDA. Preliminary calculations with the amended version gave results that were lower by 40% than the other codes. Computations with a constant salt conductivity of 2.08 Btu/hr•ft•°F provide results that are about 20% too high in the early time frames but came into agreement at the peak temperature. Computations beyond that point were not made.

After this work was completed, it was discovered that an error had been made in programming the alternative method for handling a variable conductivity. Consequently, some work on the IBM version will probably

be required before the repository models can be adequately treated with the CINDA code. The CDC version (i.e., CINDA3G, except for minor modifications to permit use of the local system) used at the Sandia Laboratories is in operation and is apparently giving good results.

In the second set of plots (Figs. 10-17) the temperatures are referred to the HEATING5 results by a percentage difference function for the temperature rise above the initial ambient that is defined by:

$$\text{Percentage difference} = \frac{T_{PRED} - T_{HEAT}}{T_{HEAT} - T_{INIT}} \times 100,$$

where

T_{PRED} = temperature at a particular time predicted by a code other than HEATING5,

T_{HEAT} = temperature, at the particular time, predicted by HEATING5,

T_{INIT} = initial temperature used as input to all codes.

This percentage difference function is a measure of the error of the temperature rise with respect to the HEATING5 result at a given location and is independent of the temperature units used.

The large differences in the percentage functions of Figs. 10-17, seen in the first few output times, are caused by a very small demoninator (i.e., the temperature is close to the initial temperature and difference errors are relatively large). Once this initial situation is passed, the curves indicate a maximum difference in the codes of under ±5% except for the TRUMP results. The TRUMP curve agrees within 5% up to year 50 where the maximum temperature is seen; it then begins some form of oscillatory behavior, with a maximum difference of 25% at year 300.

The differences observed with TRUMP are apparently caused by an overestimation of the thermal diffusivity. The exact cause of this overestimation is unknown. It is possible that the nodes in TRUMP were not centered in the control volumes as is usual in other codes, but, to match the mesh, were located in the upper left-hand corner of the control volume.

The oscillatory behavior of the differences seems to indicate a problem with convergence in the latter part of the problem. This may

possibly be overcome by a tightening of the convergence criteria in the latter part of the transient.

6. CALCULATED RESULTS AND EVALUATION OF THE THREE-DIMENSIONAL MODEL

As in the 2-D comparison, the HEATING5 results were shown as a reference for comparison. Because of the long running times and poor agreement with the other codes for the 2-D model, the 3-D model was not run with TRUMP and CINDA.

Figures 18-49 show the temperatures predicted by each code as a function of time using a nonlinear time axis. The true time scale is shown in Table 6. The physical location of the nodes plotted can be determined from Table 7 and located in Fig. 1.

Examination of Figs. 18-49 leads to the following observations:

1. the peak temperature predicted by each code occurs at the same printout time;
2. the curves tend to grow together with time.

Figures 50-73 show the percentage difference function that was previously defined. As in the 2-D comparison, the large differences seen initially are due to the relatively small size of the denominator. Once this condition is passed, the curves agree to within a ±5% difference that decreases with time.

Although it is not the intent of this report to compare the 2- and 3-D results, it is worth noting that they agree to within 5% of each other external to the source. The differences inside the source region are attributable to the 2-D source being a smeared source and the 3-D source being a concentrated one. When the distance from the surface of the source reaches >1 ft there is very little difference in the two models.

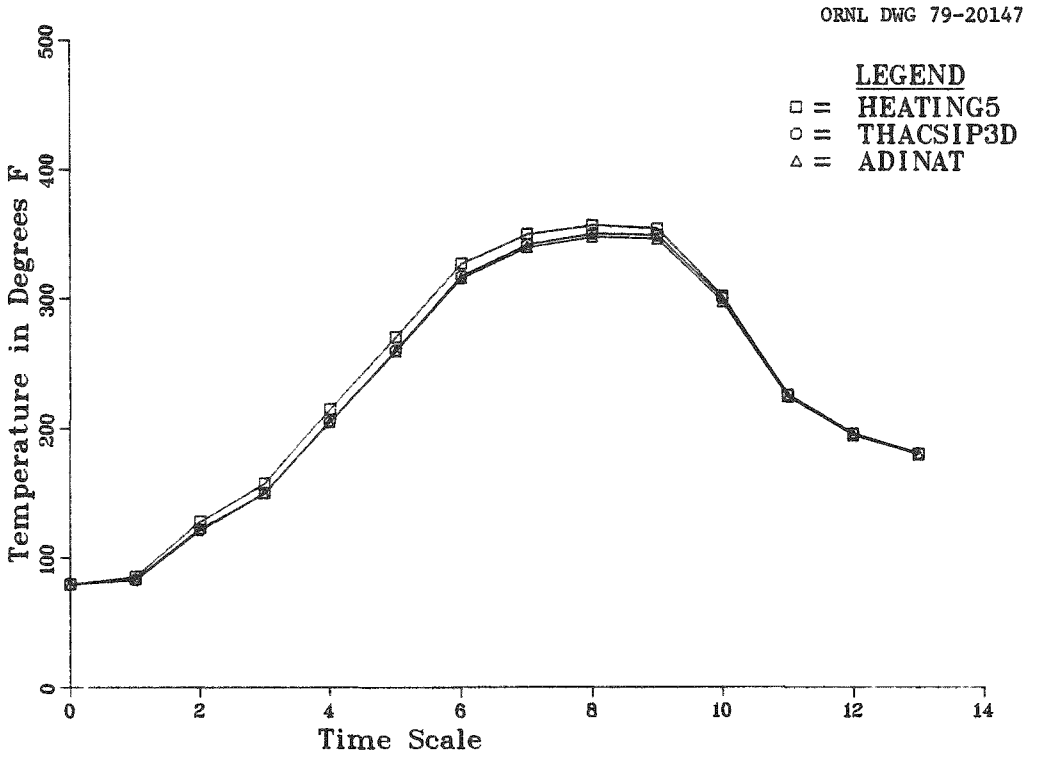

Fig. 18. Temperature vs time for location 1, plane 1 (see Table 6).

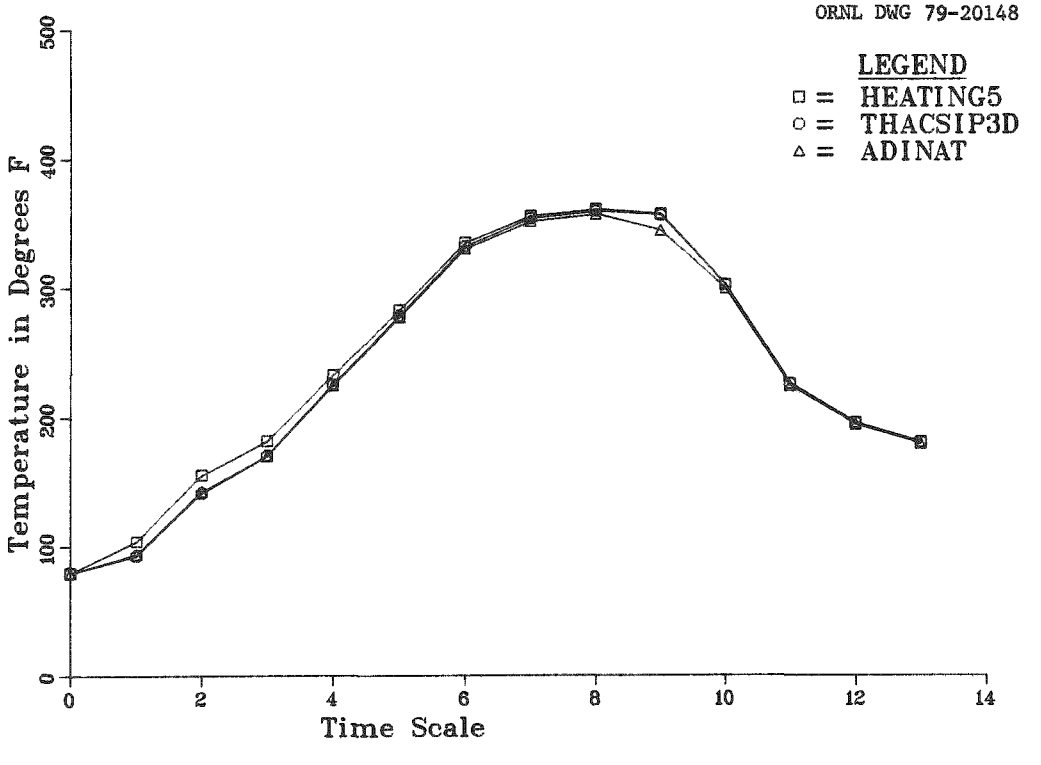

Fig. 19. Temperature vs time for location 2, plane 1 (see Table 6).

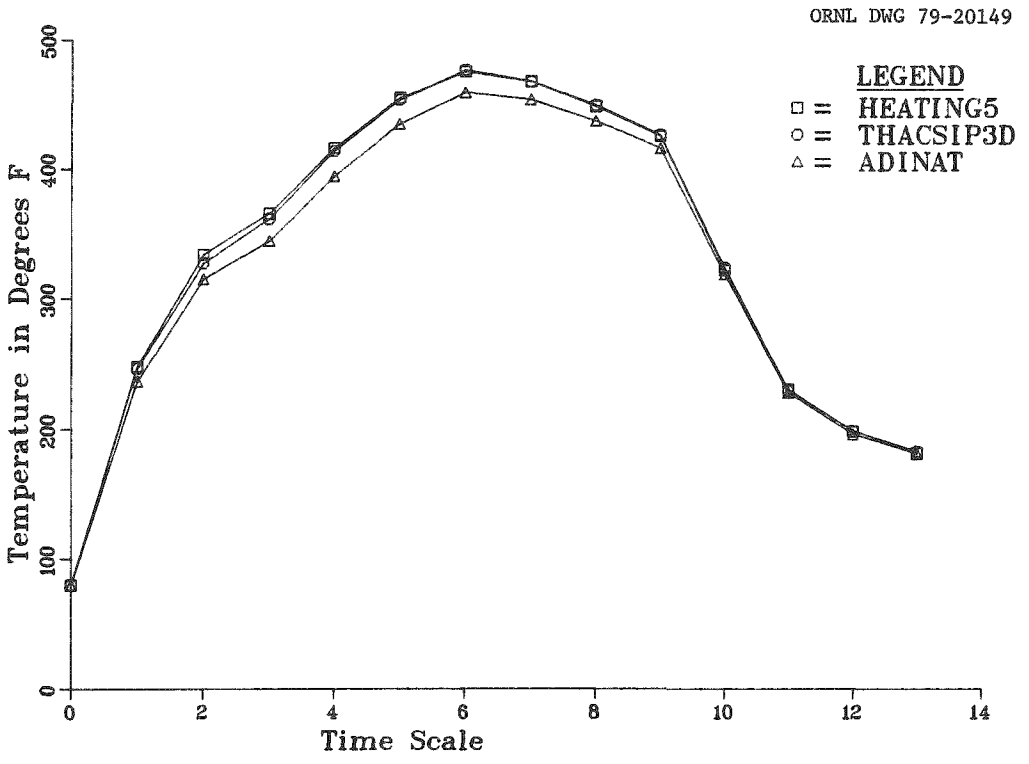

Fig. 20. Temperature vs time for location 3, plane 1 (see Table 6).

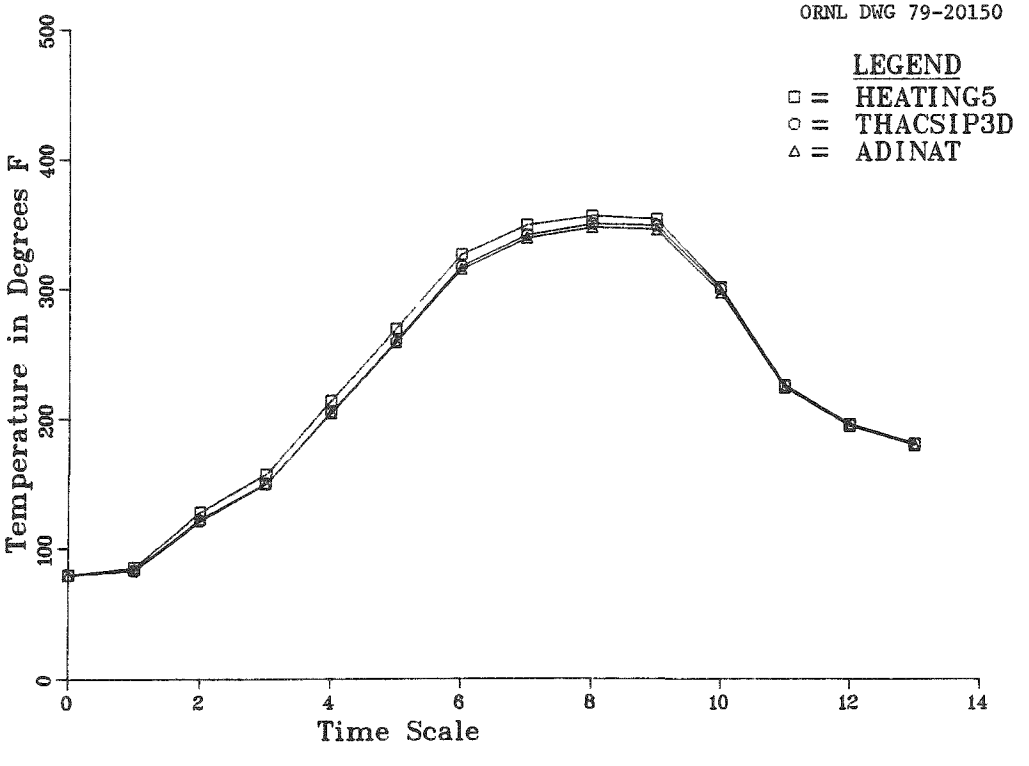

Fig. 21. Temperature vs time for location 4, plane 1 (see Table 6).

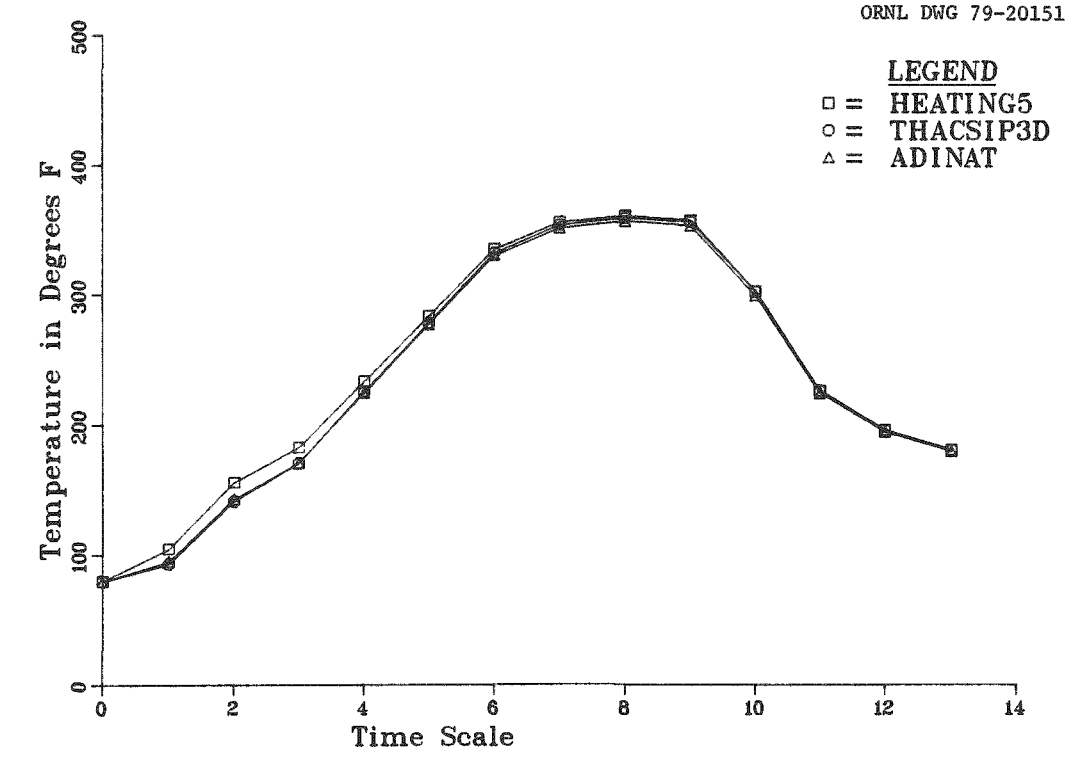

Fig. 22. Temperature vs time for location 5, plane 1 (see Table 6).

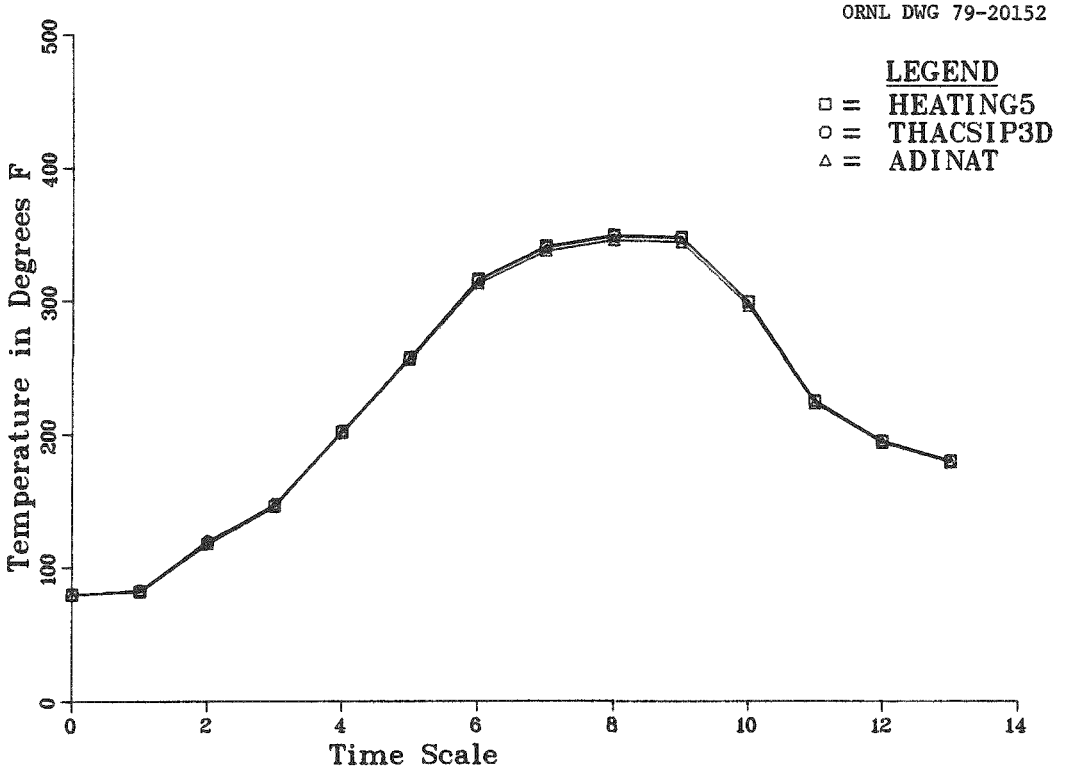

Fig. 23. Temperature vs time for location 6, plane 1 (see Table 6).

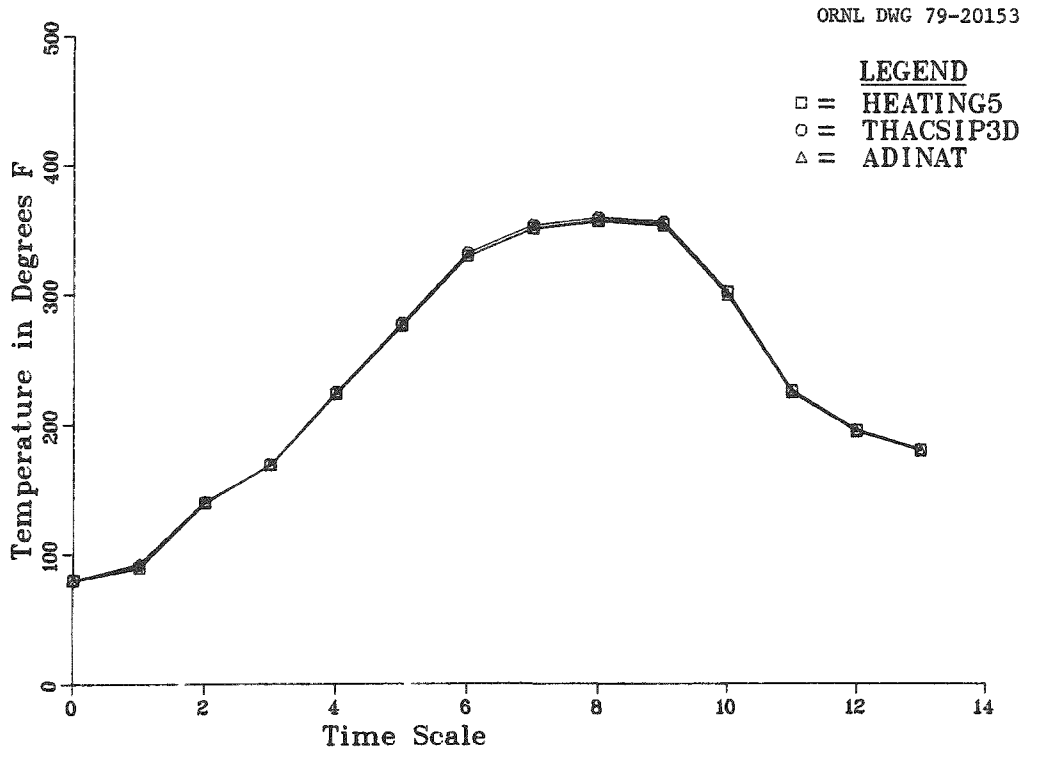

Fig. 24. Temperature vs time for location 7, plane 1 (see Table 6).

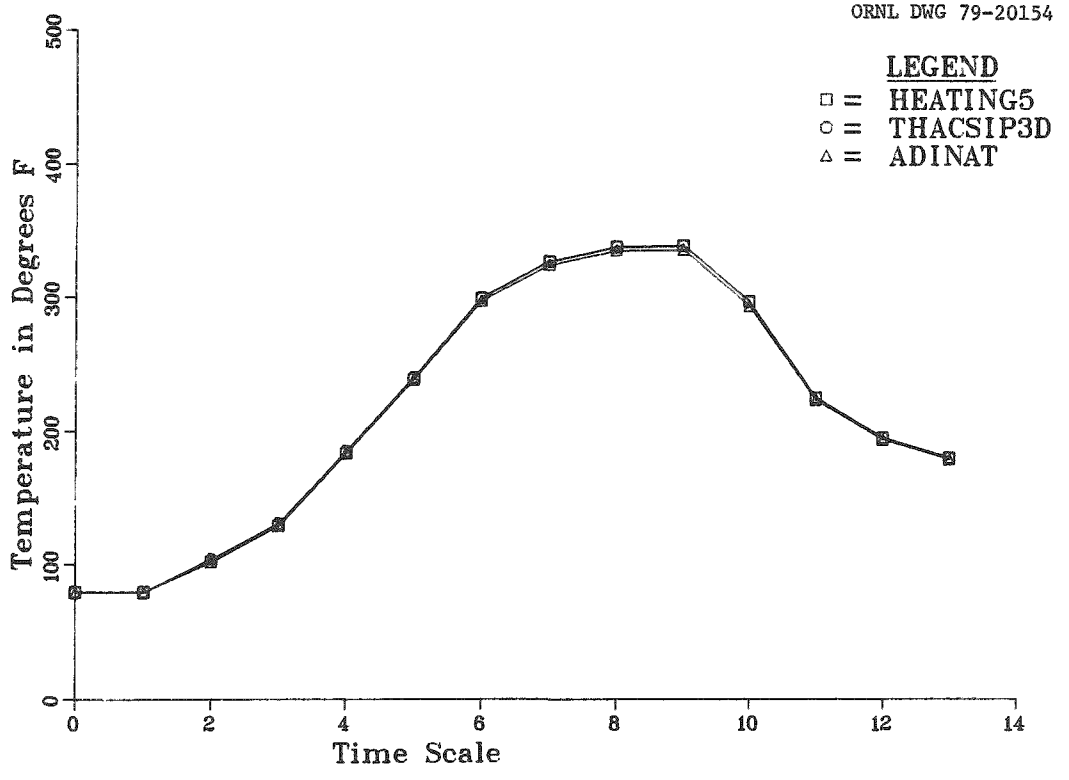

Fig. 25. Temperature vs time for location 8, plane 1 (see Table 6).

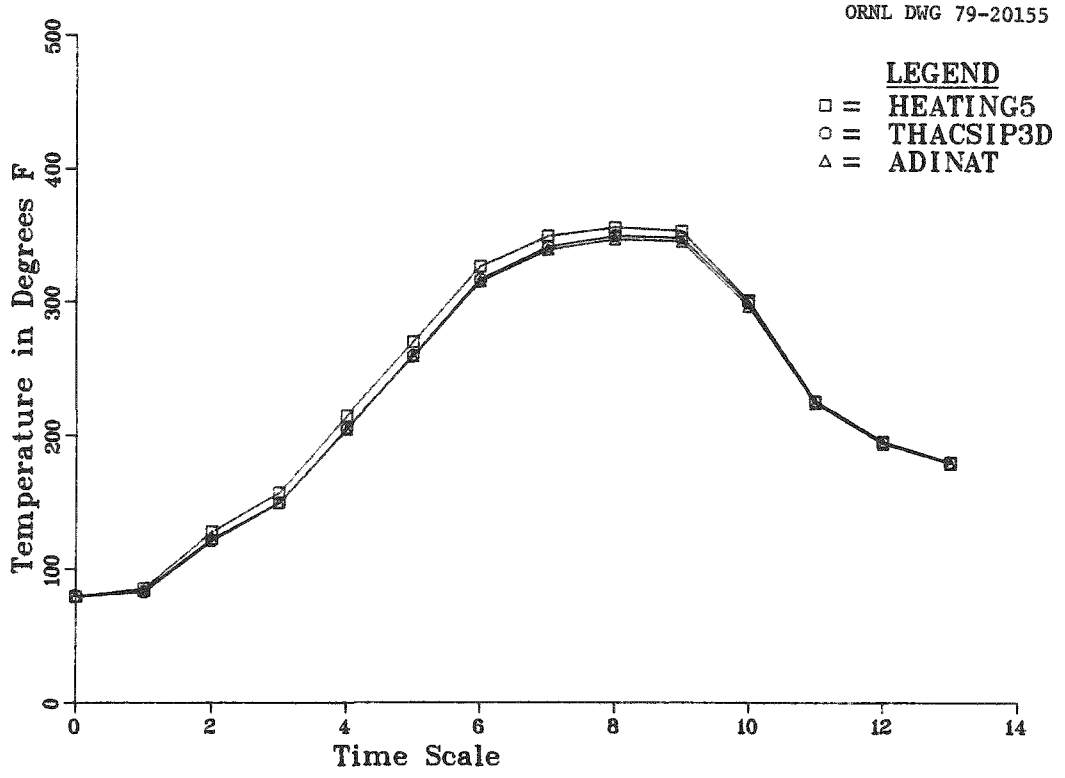

Fig. 26. Temperature vs time for location 1, plane 2 (see Table 6).

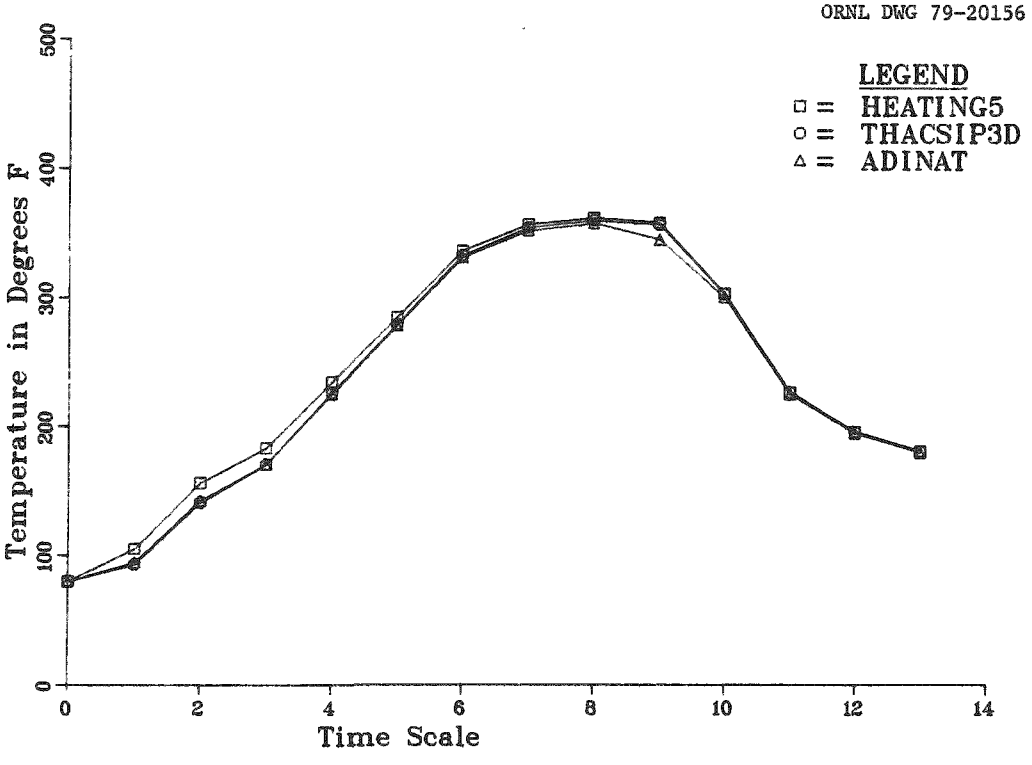

Fig. 27. Temperature vs time for location 2, plane 2 (see Table 6).

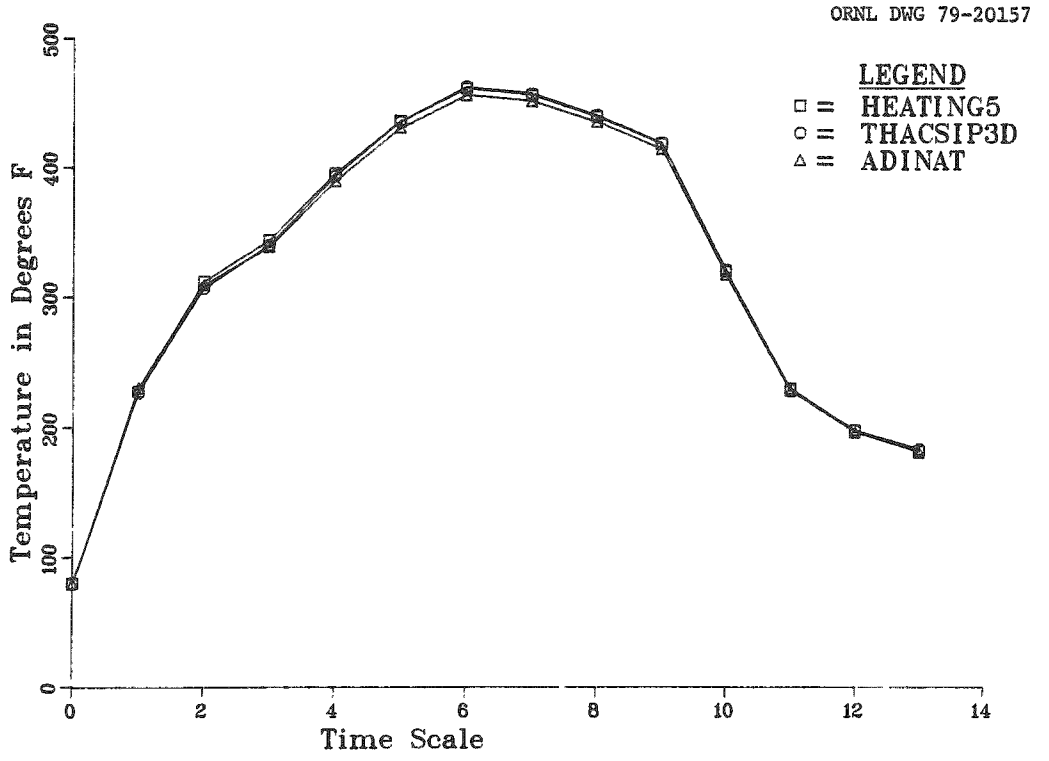

Fig. 28. Temperature vs time for location 3, plane 2 (see Table 6).

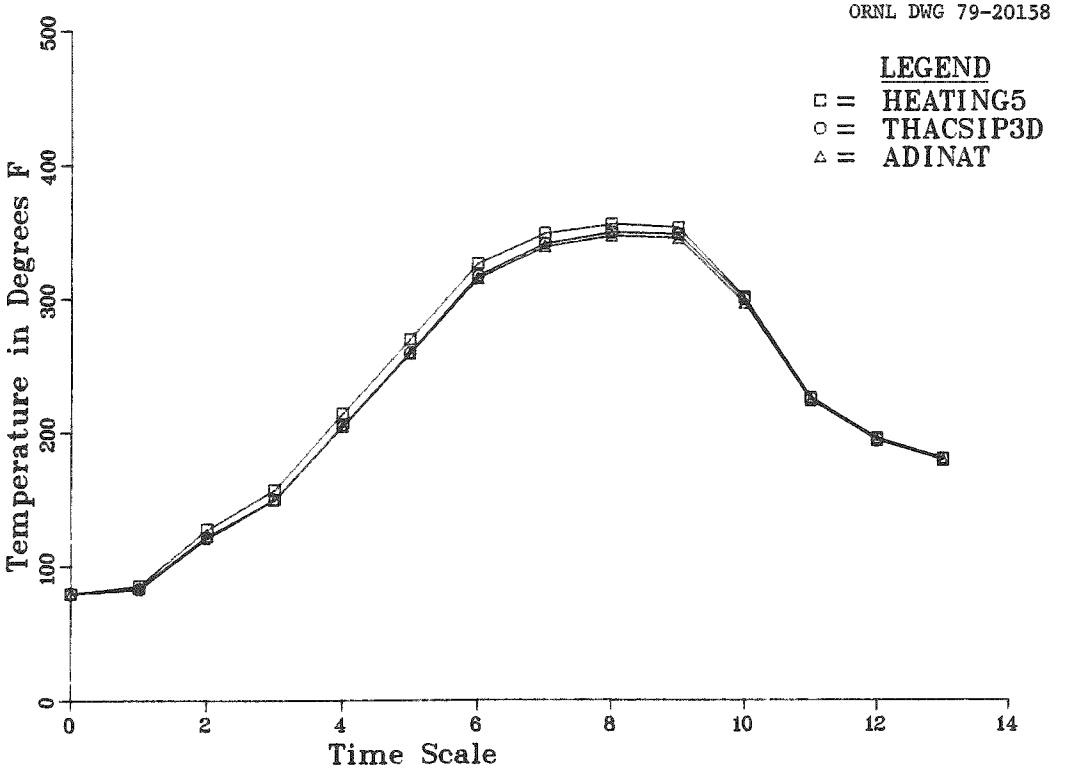

Fig. 29. Temperature vs time for location 4, plane 2 (see Table 6).

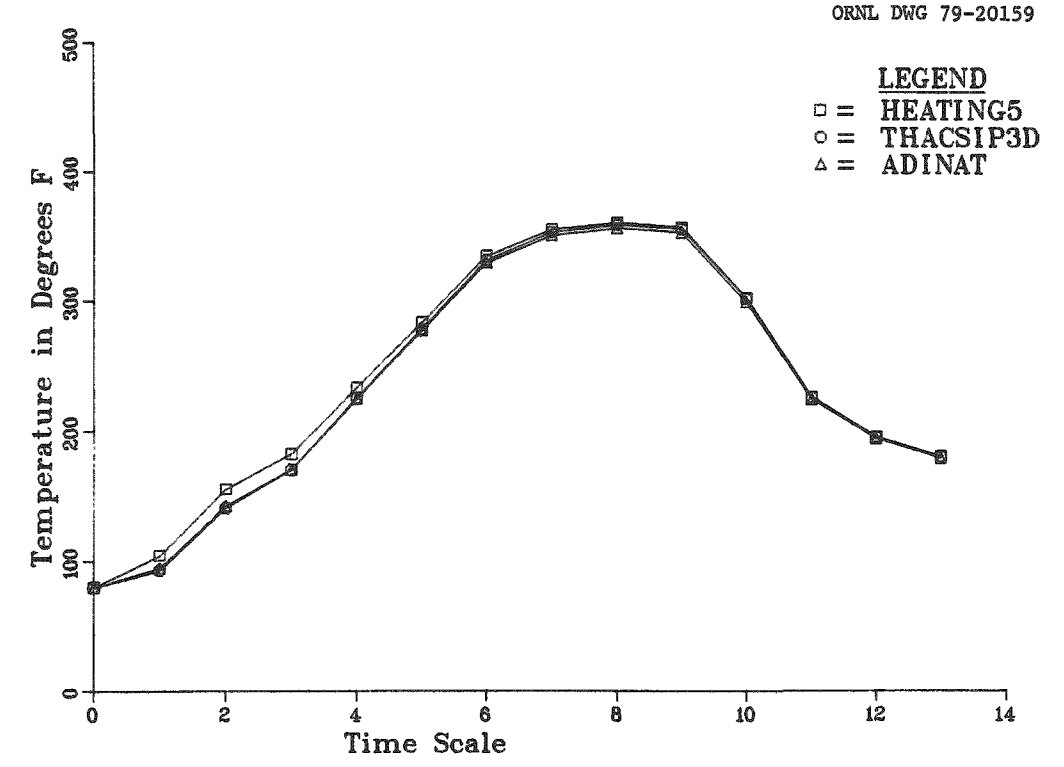

Fig. 30. Temperature vs time for location 5, plane 2 (see Table 6).

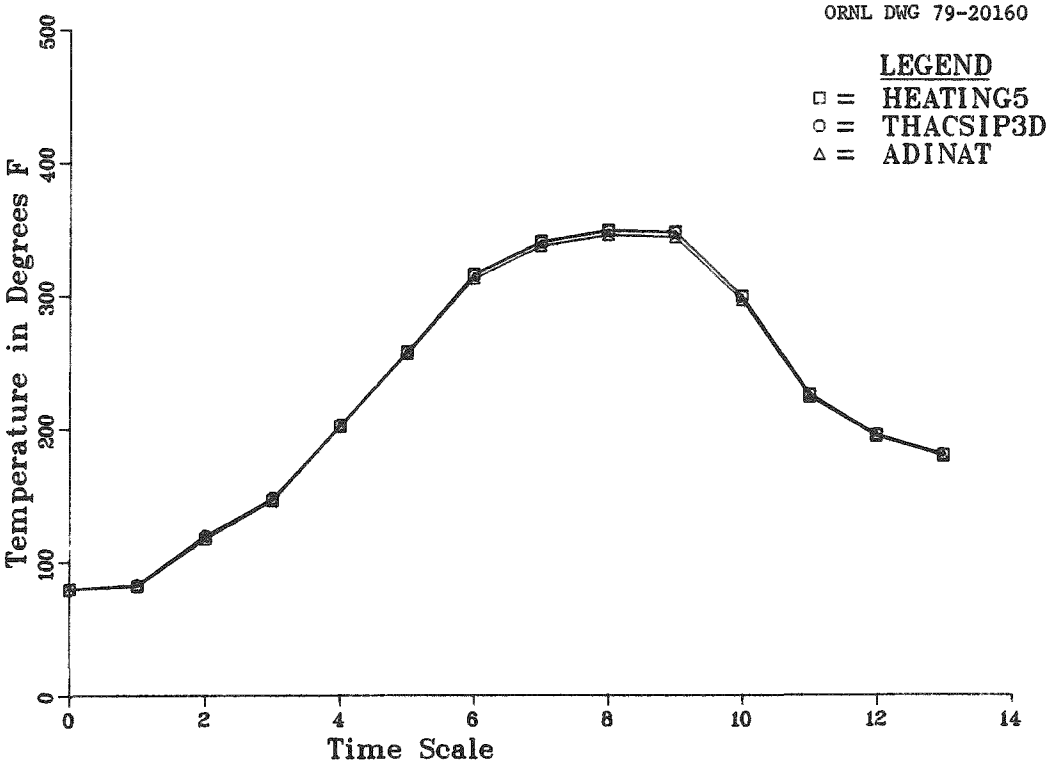

Fig. 31. Temperature vs time for location 6, plane 2 (see Table 6).

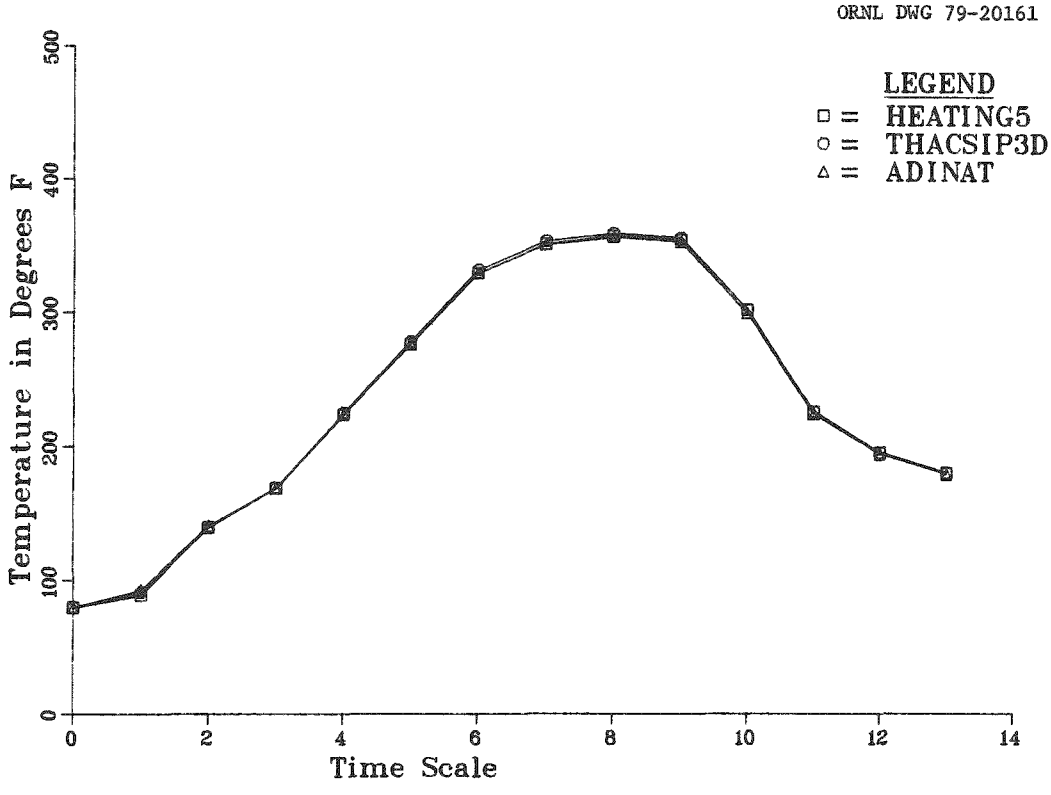

Fig. 32. Temperature vs time for location 7, plane 2 (see Table 6).

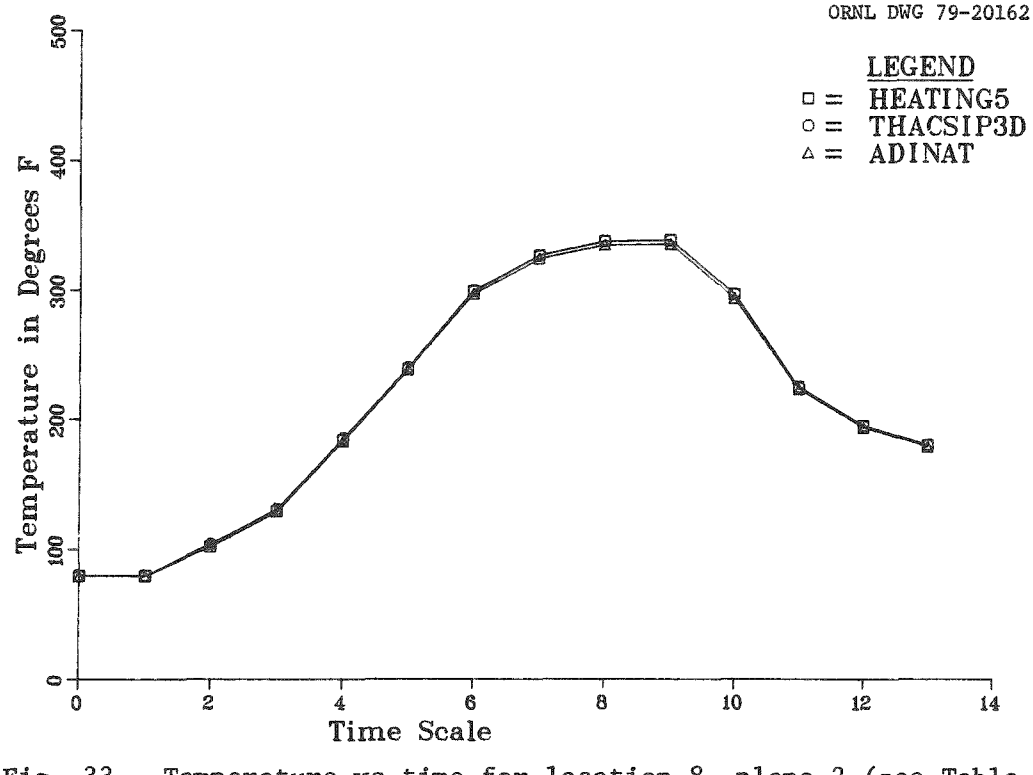

Fig. 33. Temperature vs time for location 8, plane 2 (see Table 6).

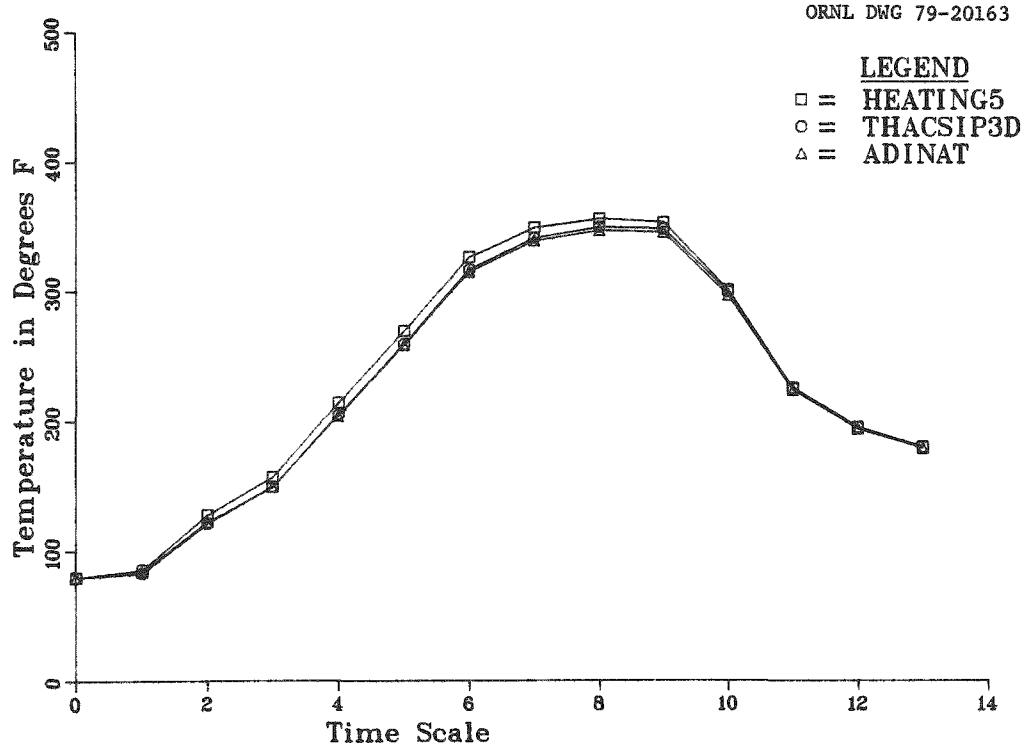

Fig. 34. Temperature vs time for location 1, plane 3 (see Table 6)

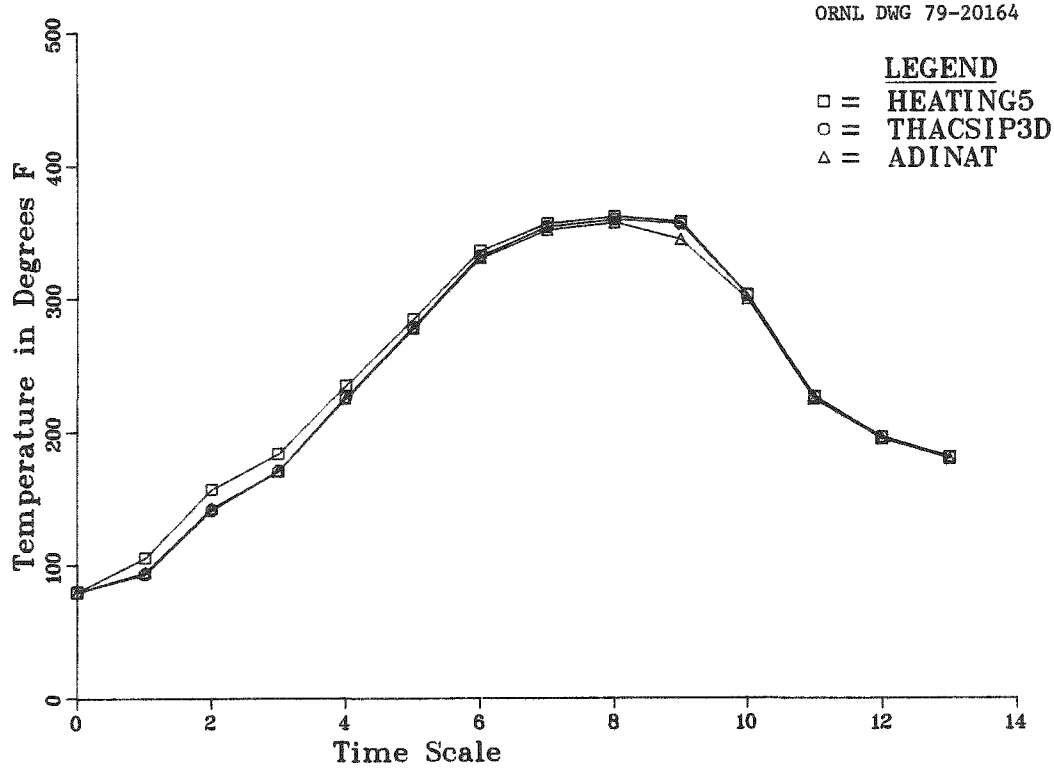

Fig. 35. Temperature vs time for location 2, plane 3 (see Table 6)

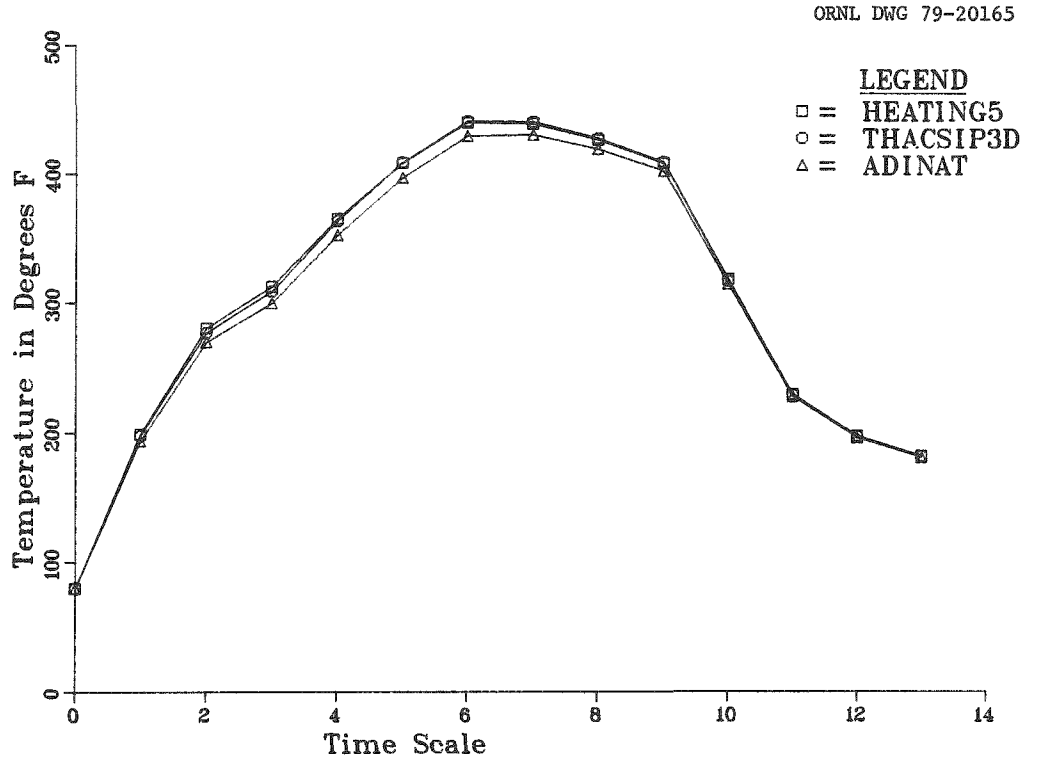

Fig. 36. Temperature vs time for location 3, plane 3 (see Table 6)

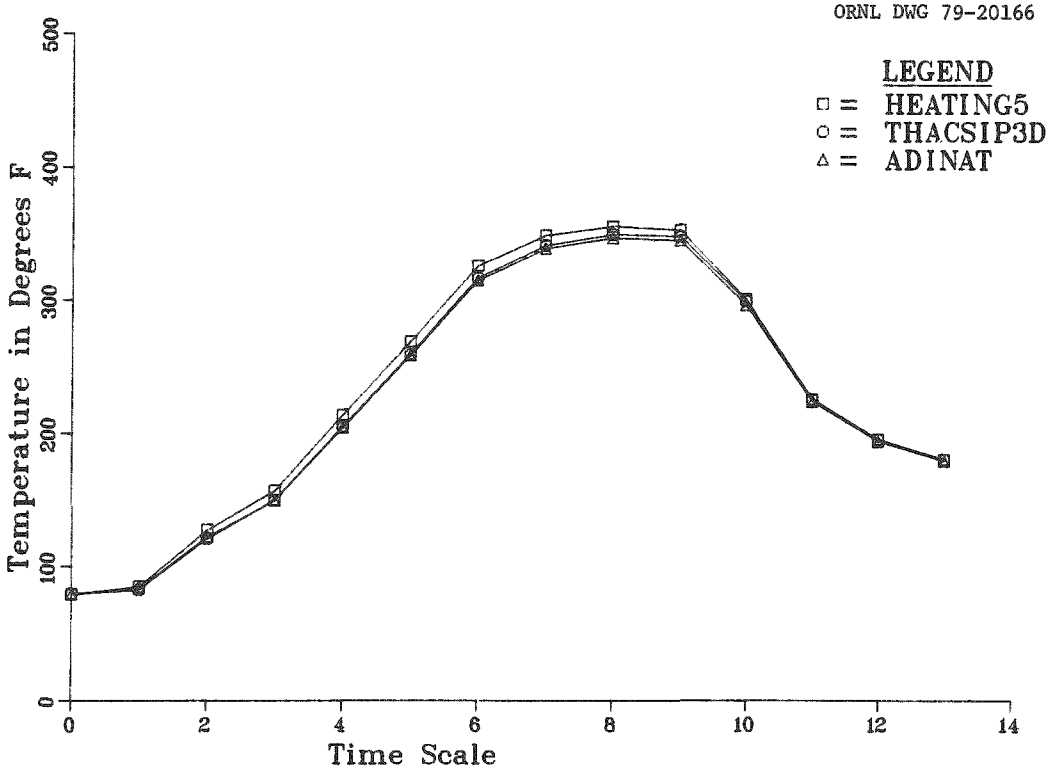

Fig. 37. Temperature vs time for location 4, plane 3 (see Table 6).

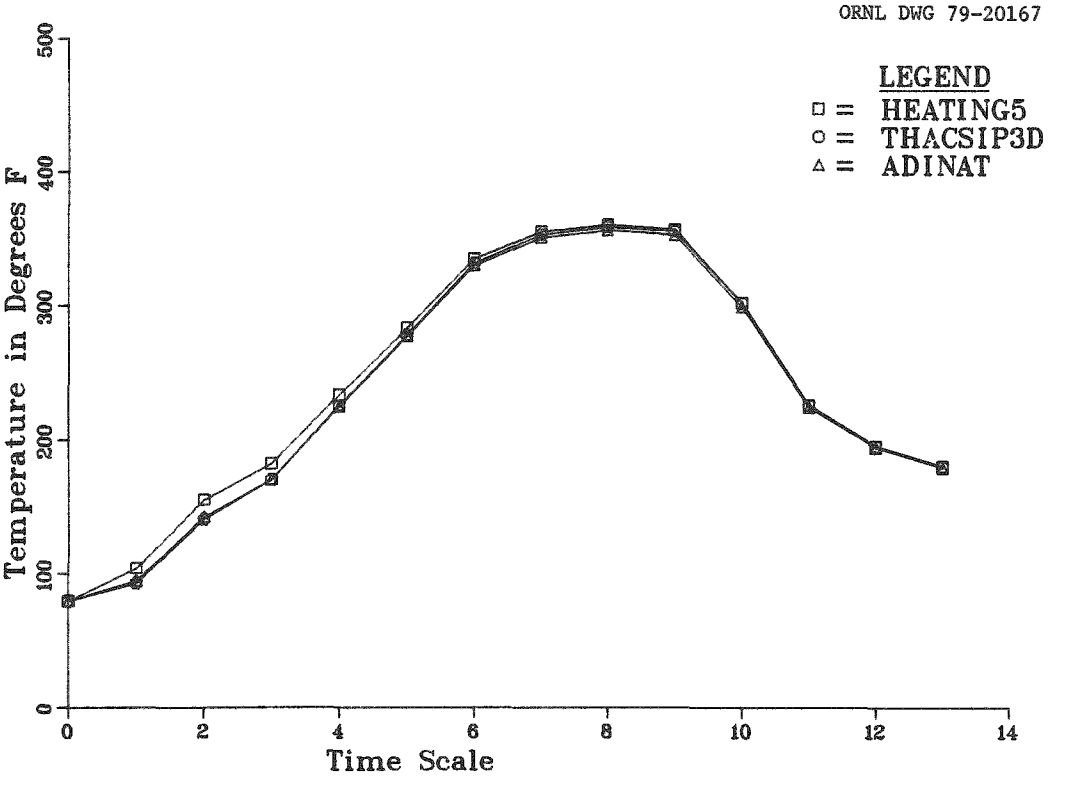

Fig. 38. Temperature vs time for location 5, plane 3 (see Table 6).

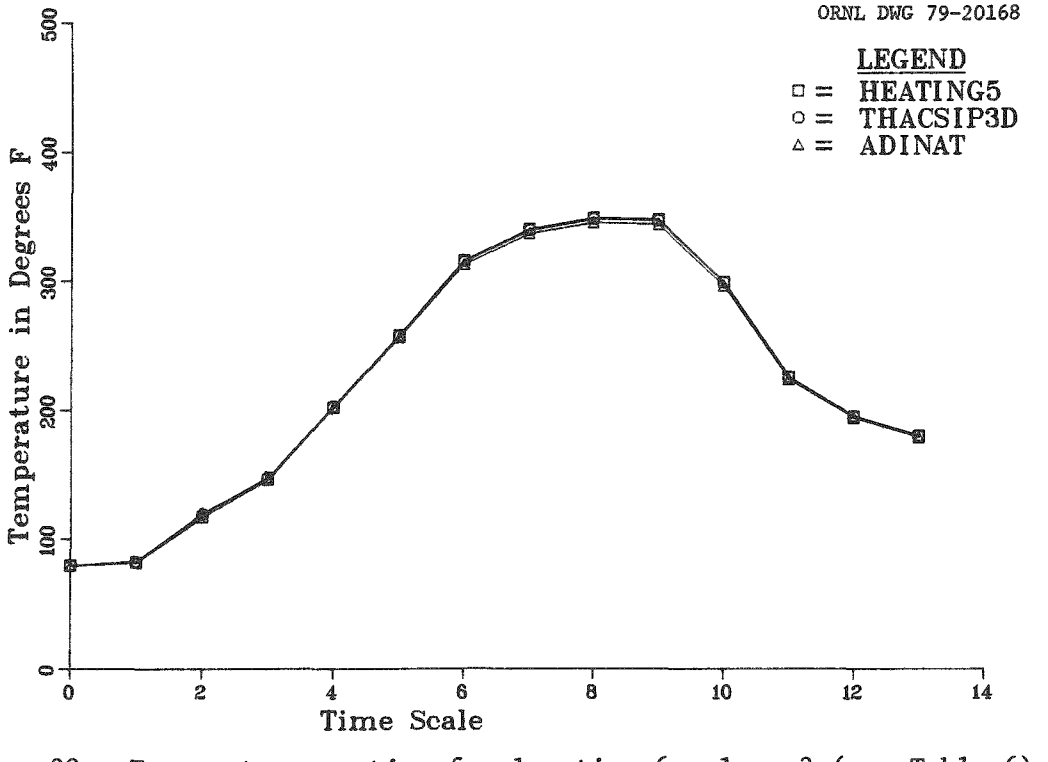

Fig. 39. Temperature vs time for location 6, plane 3 (see Table 6).

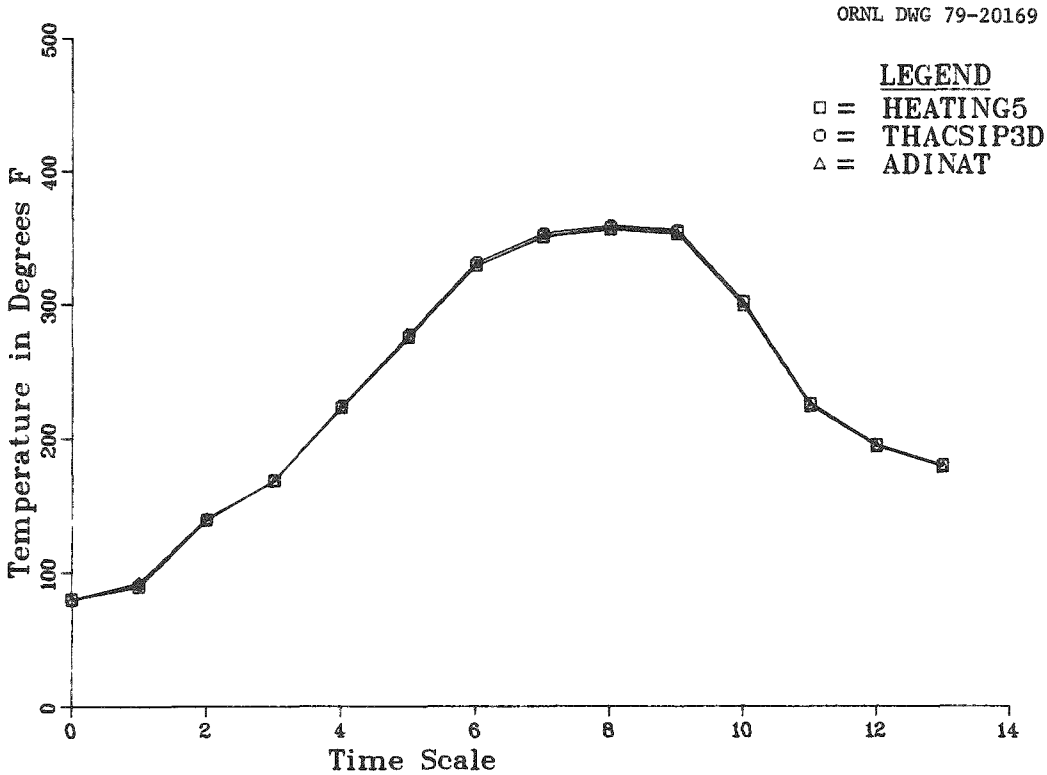

Fig. 40. Temperature vs time for location 7, plane 3 (see Table 6).

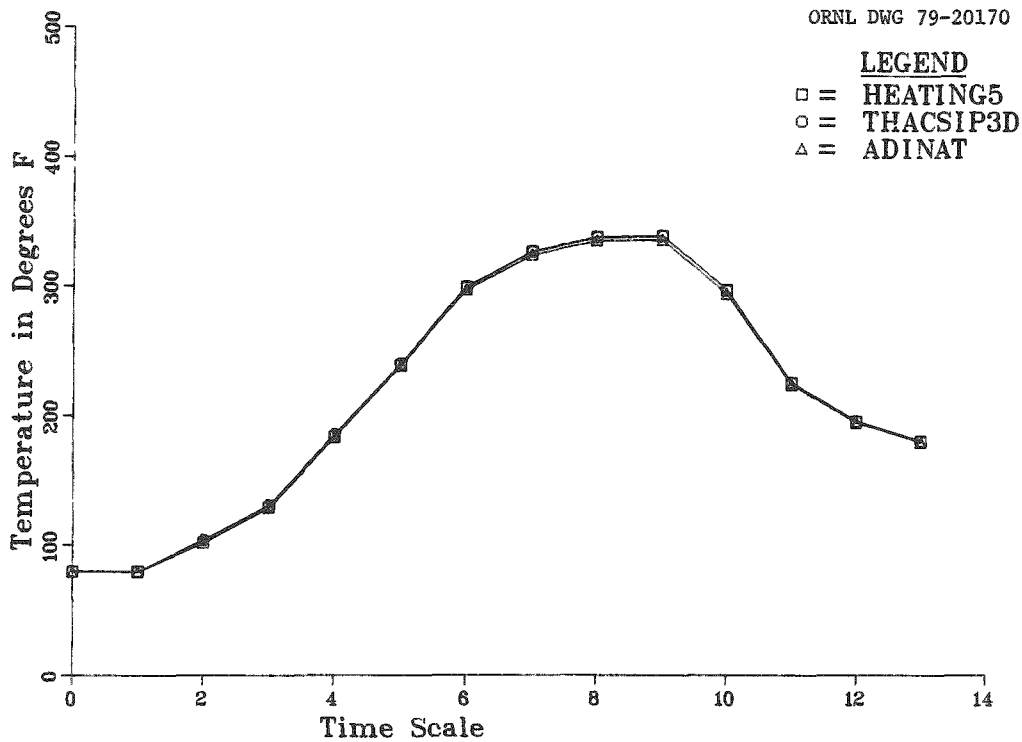

Fig. 41. Temperature vs time for location 8, plane 3 (see Table 6).

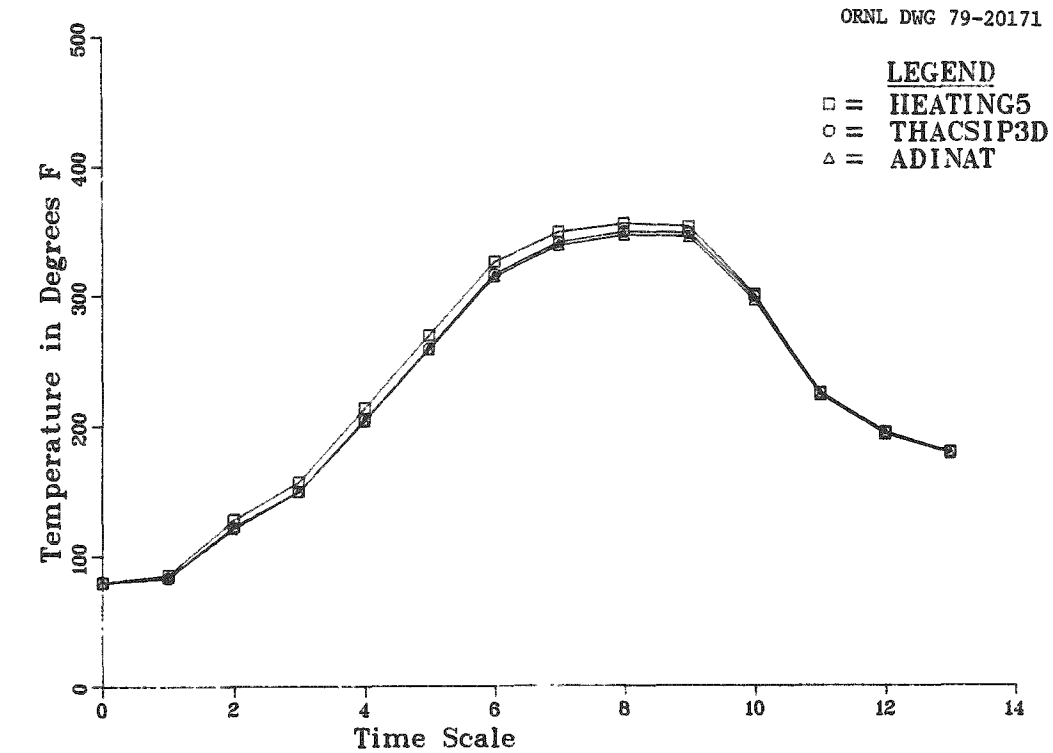

Fig. 42. Temperature vs time for location 1, plane 4 (see Table 6).

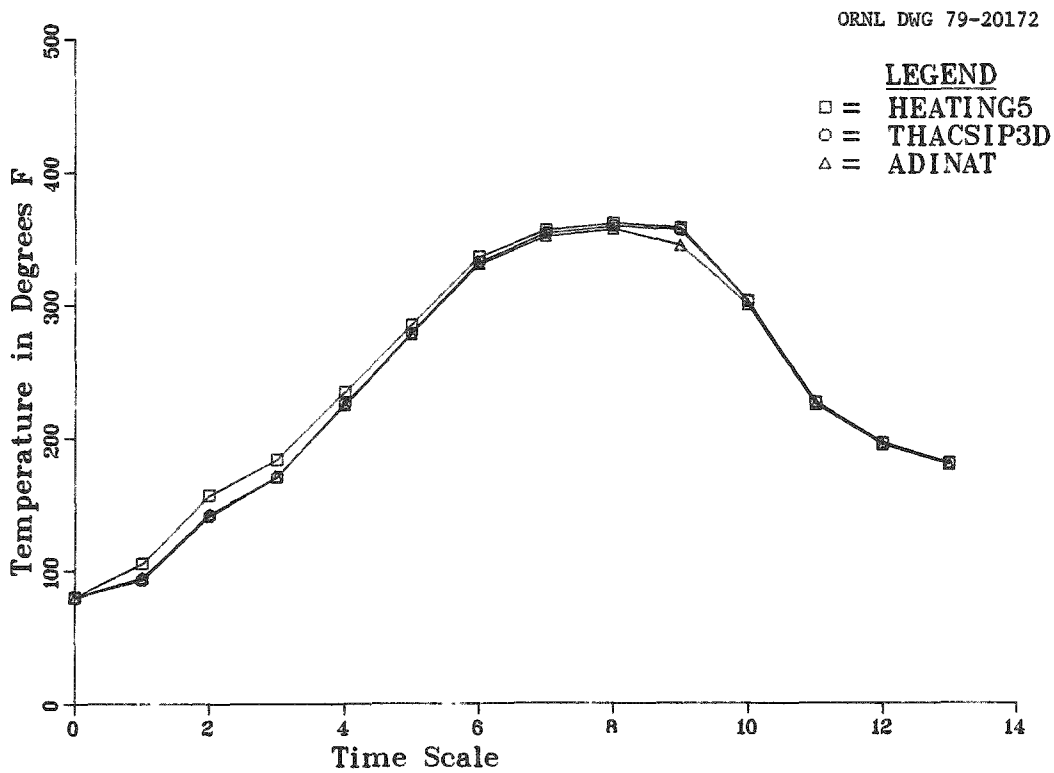

Fig. 43. Temperature vs time for location 2, plane 4 (see Table 6).

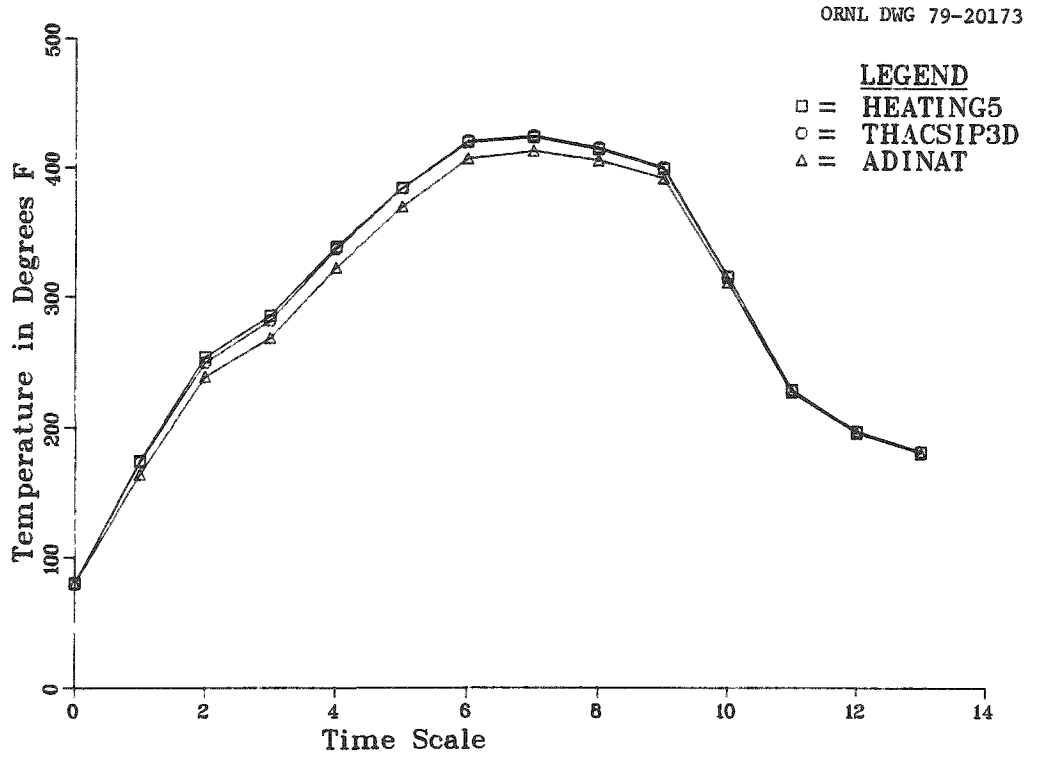

Fig. 44. Temperature vs time for location 3, plane 4 (see Table 6).

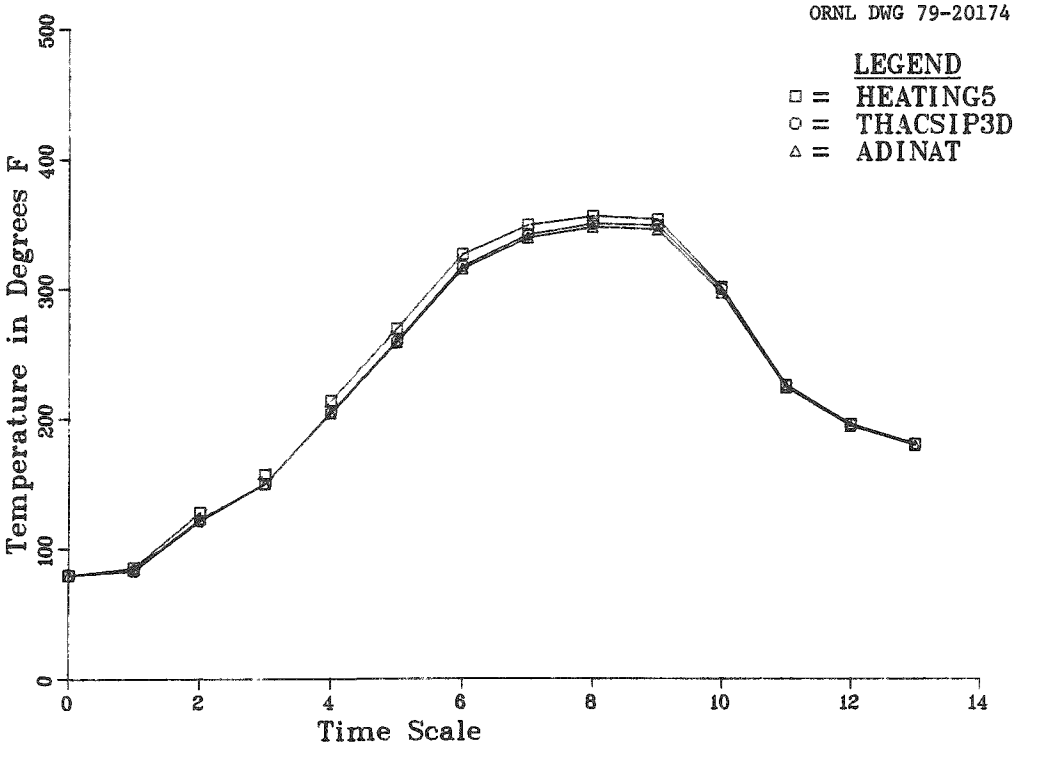

Fig. 45. Temperature vs time for location 4, plane 4 (see Table 6).

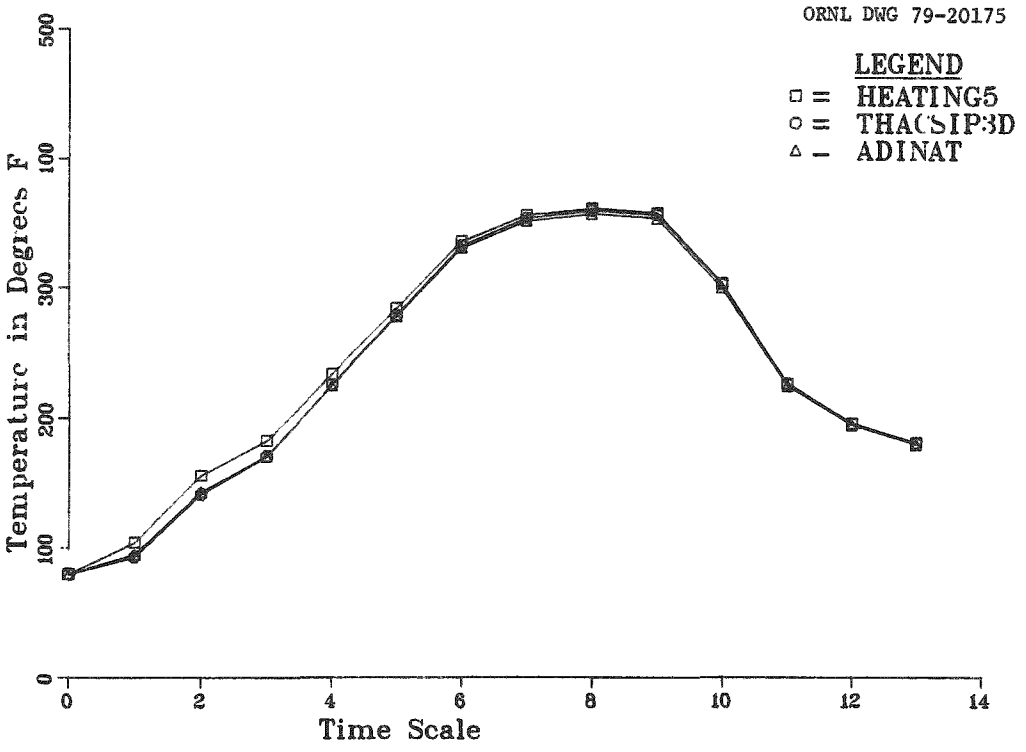

Fig. 46. Temperature vs time for location 5, plane 4 (see Table 6).

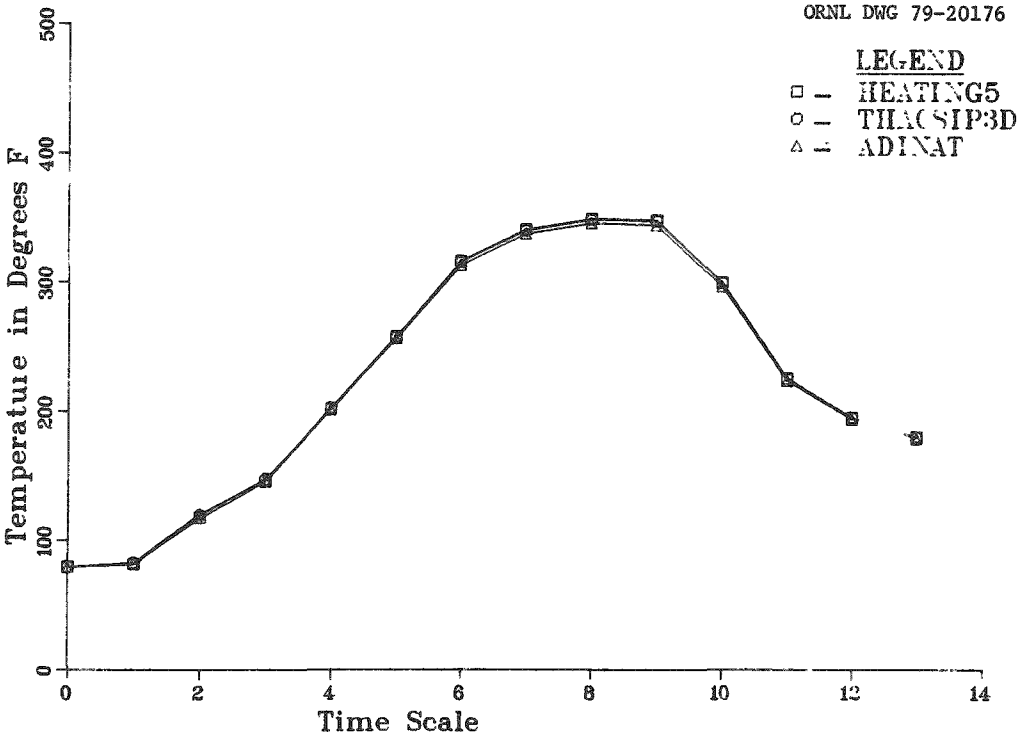

Fig. 47. Temperature vs time for location 6, plane 4 (see Table 6).

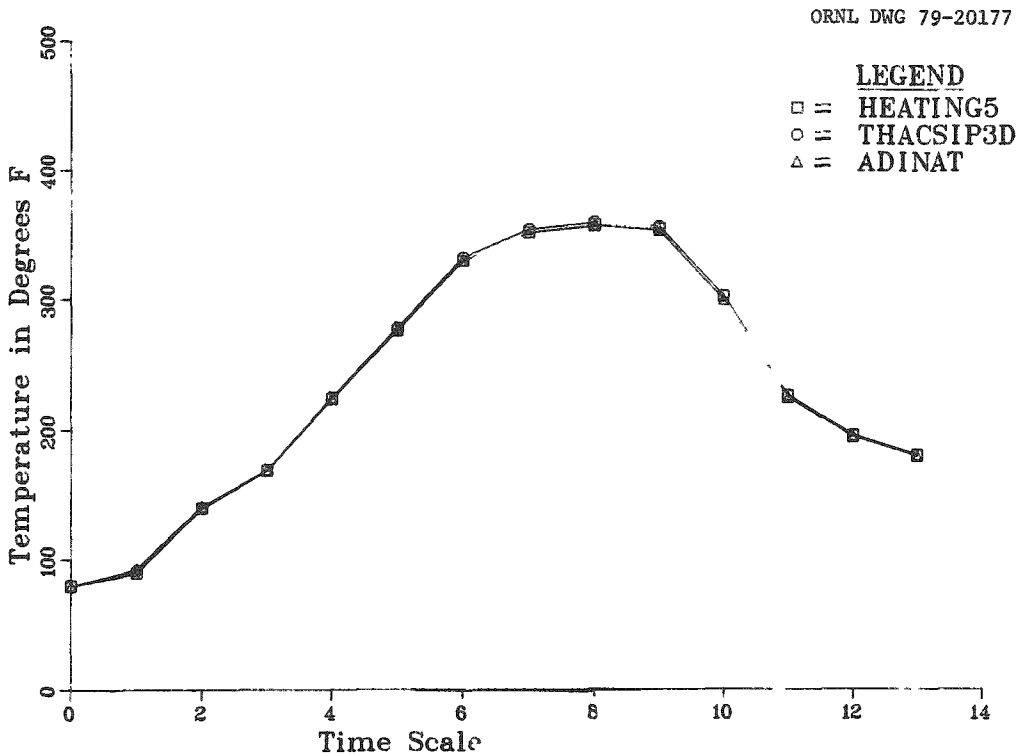

Fig. 48. Temperature vs time for location 7, plane 4 (see Table 6).

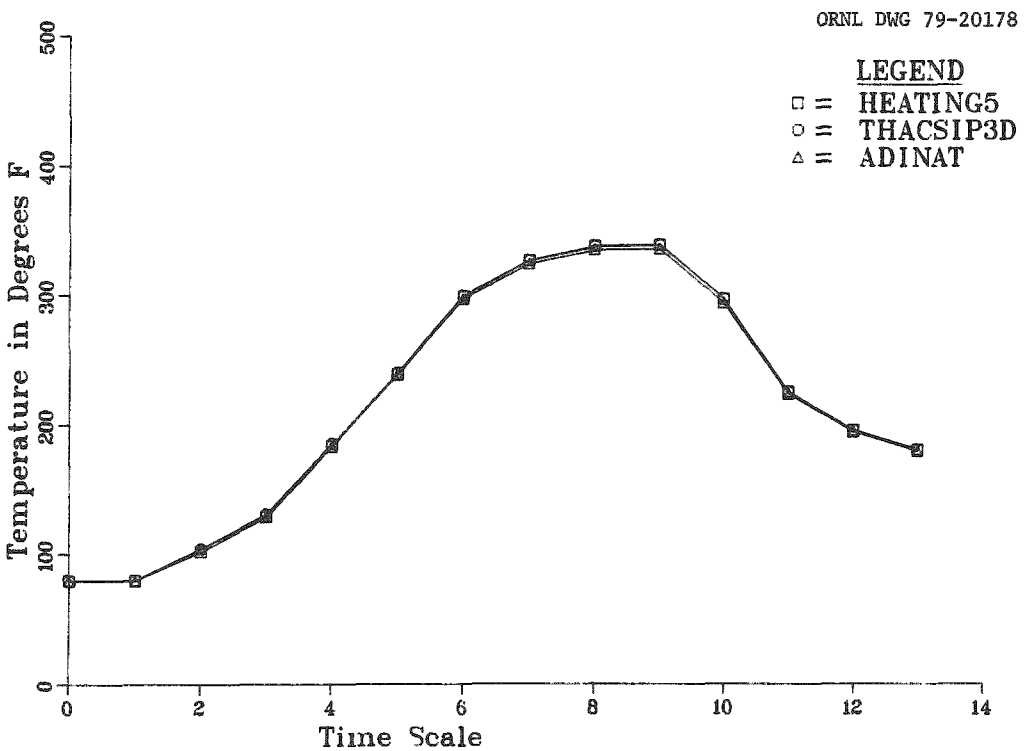

Fig. 49. Temperature vs time for location 8, plane 4 (see Table 6).

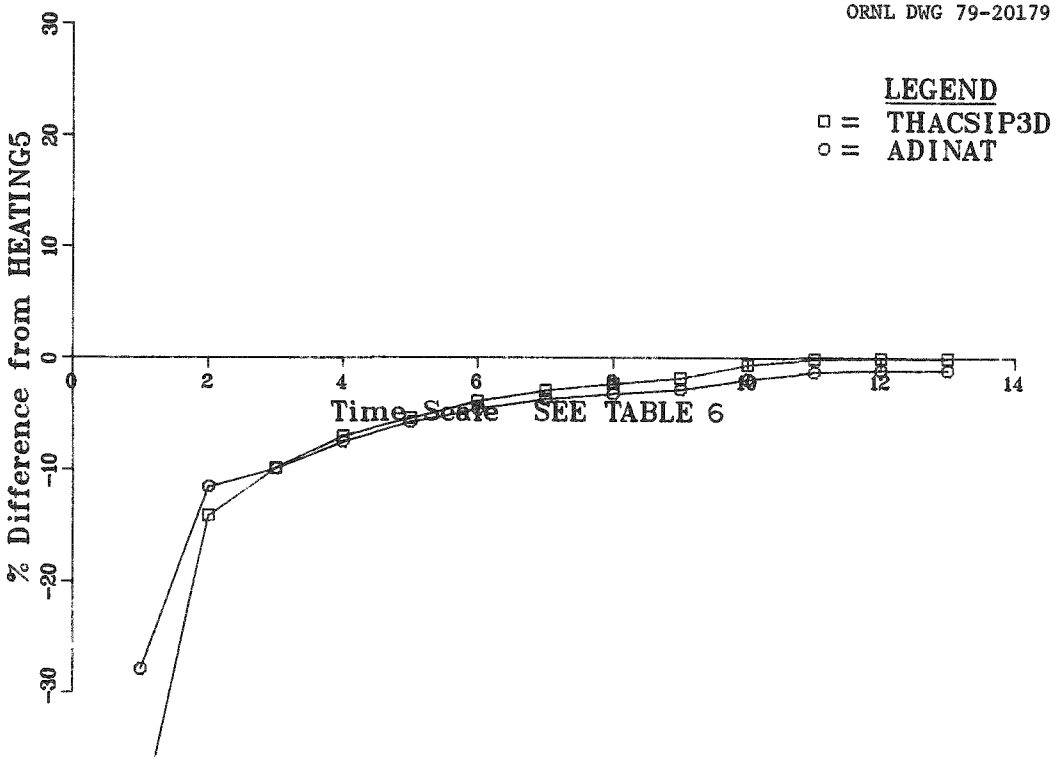

Fig. 50. Temperature vs time for location 1, plane 1.

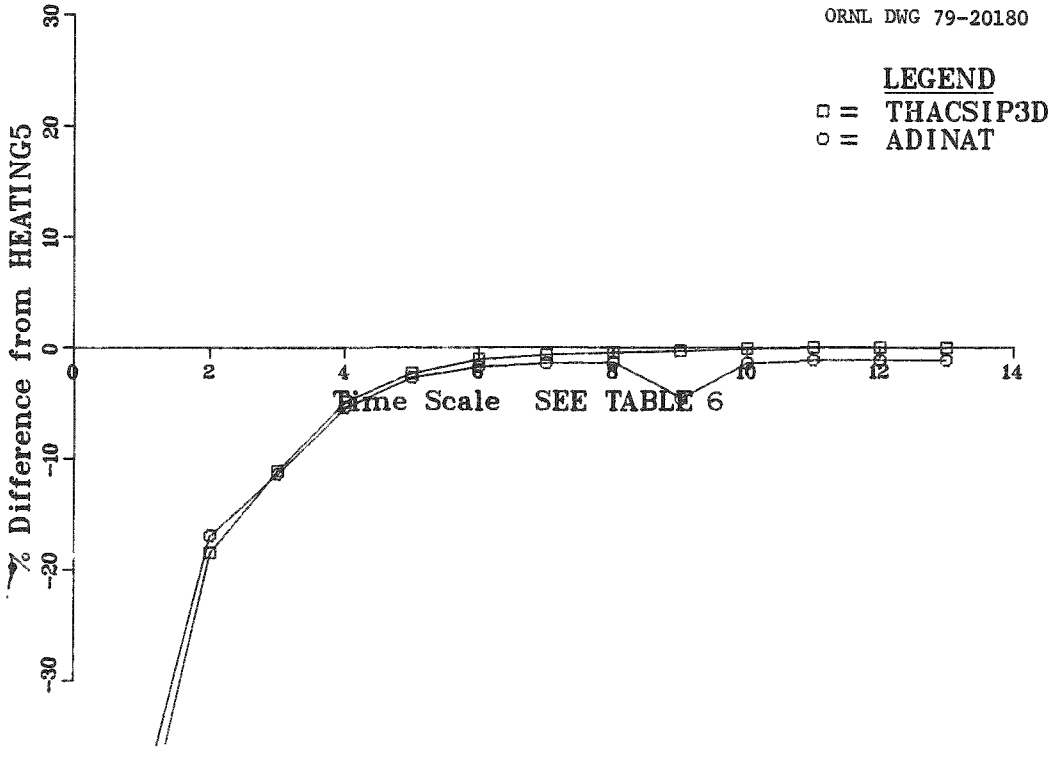

Fig. 51. Temperature vs time for location 2, plane 1.

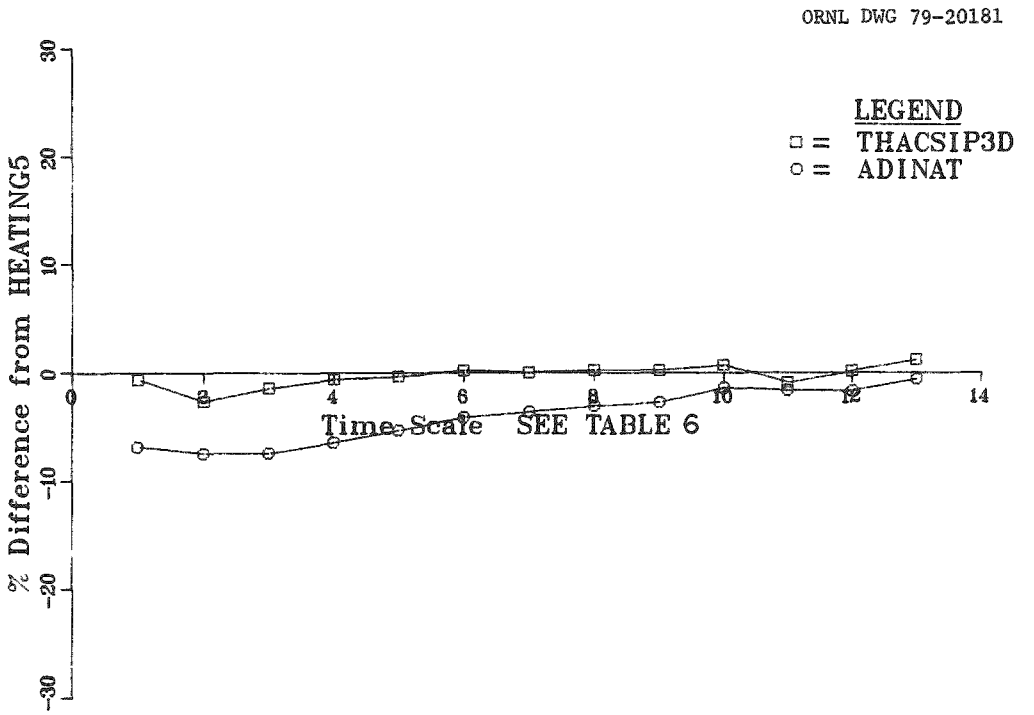

Fig. 52. Temperature vs time for location 3, plane 1.

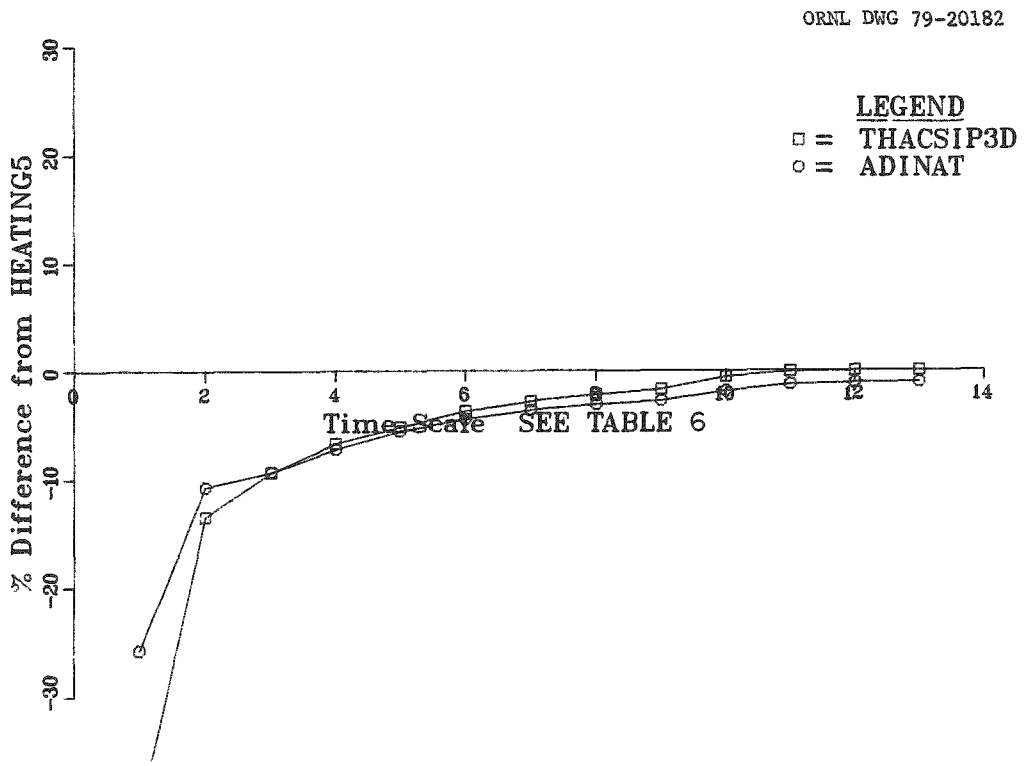

Fig. 53. Temperature vs time for location 4, plane 1.

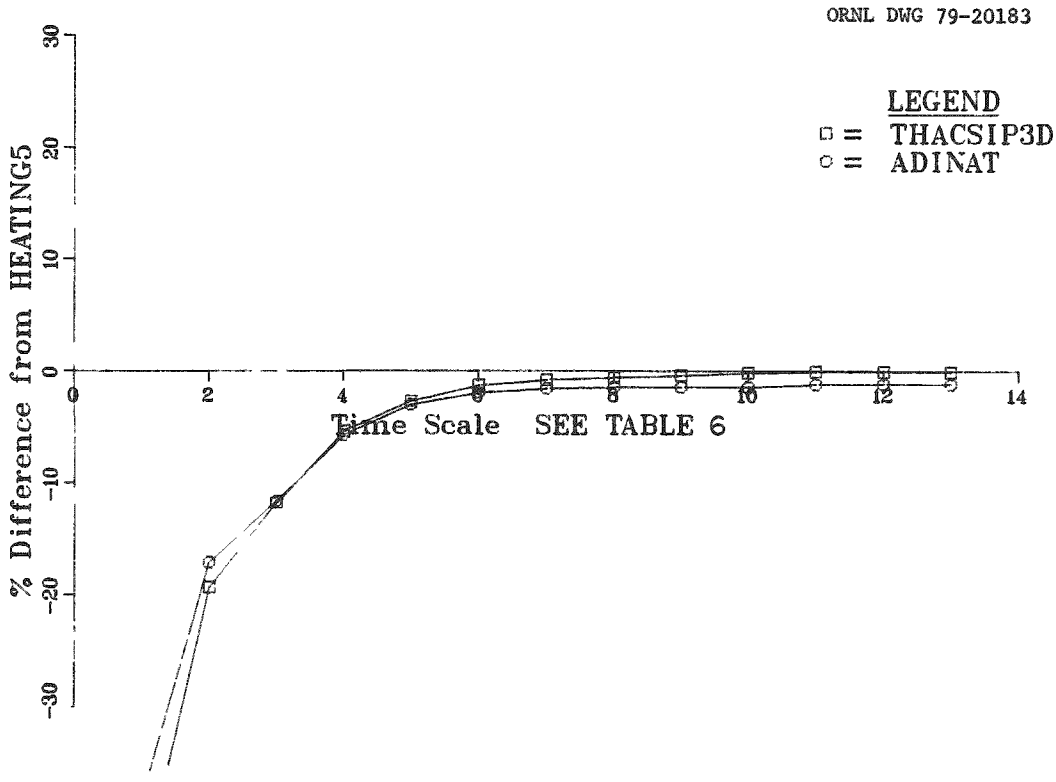

Fig. 54. Temperature vs time for location 5, plane 1.

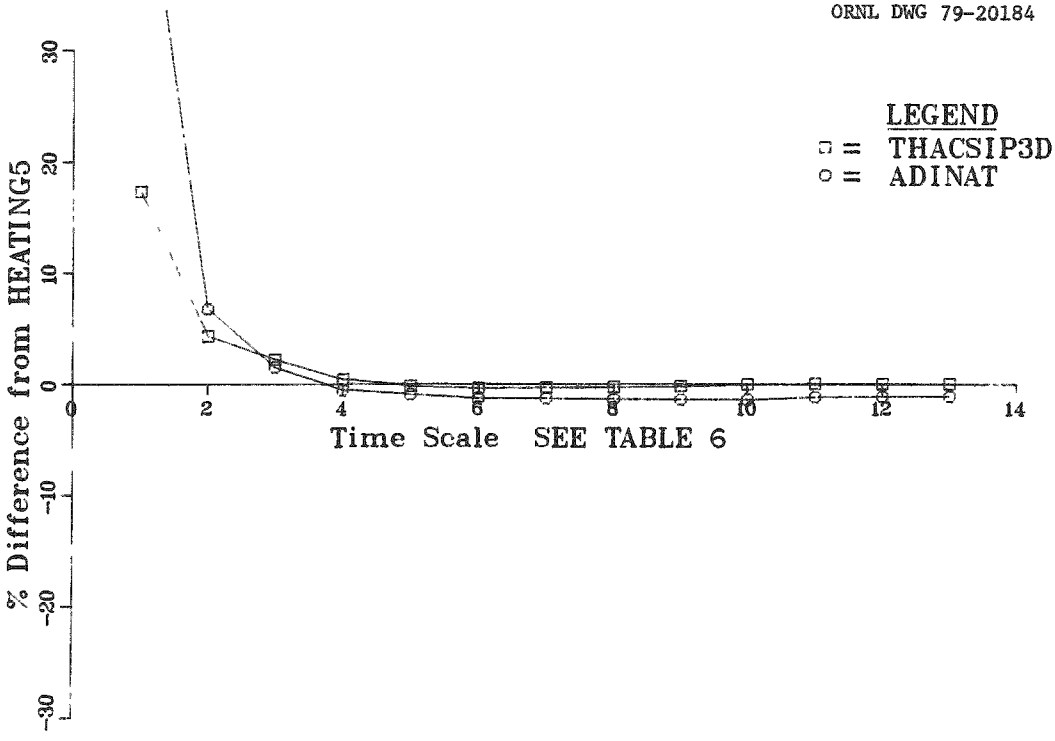

Fig. 55. Temperature vs time for location 6, plane 1.

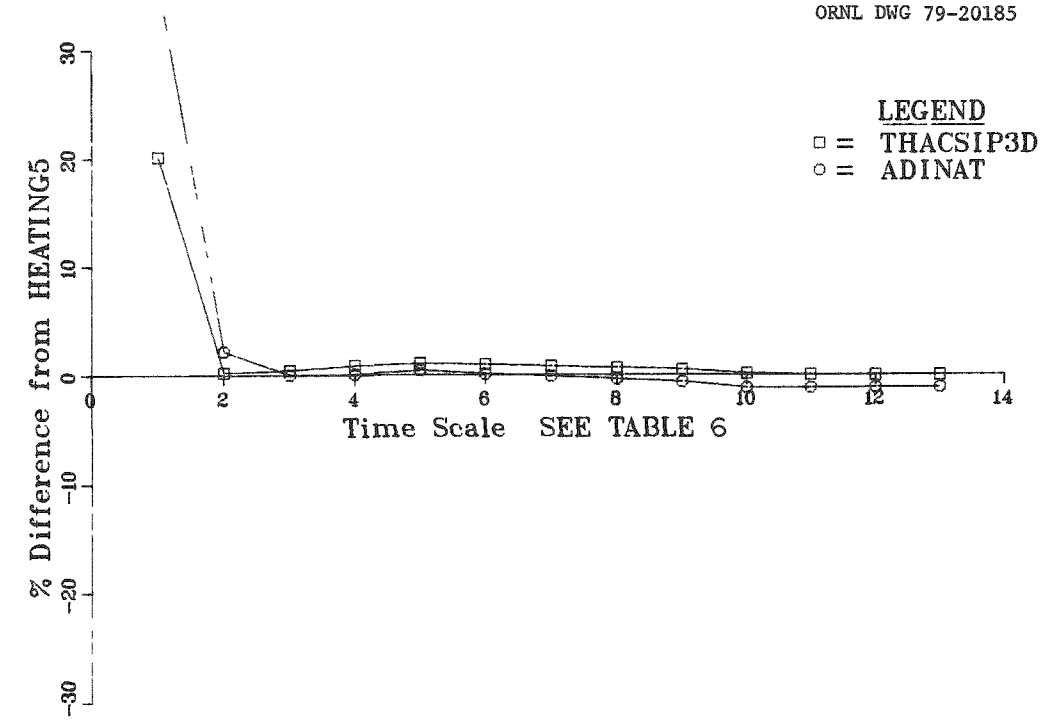

Fig. 56. Temperature vs time for location 7, plane 1.

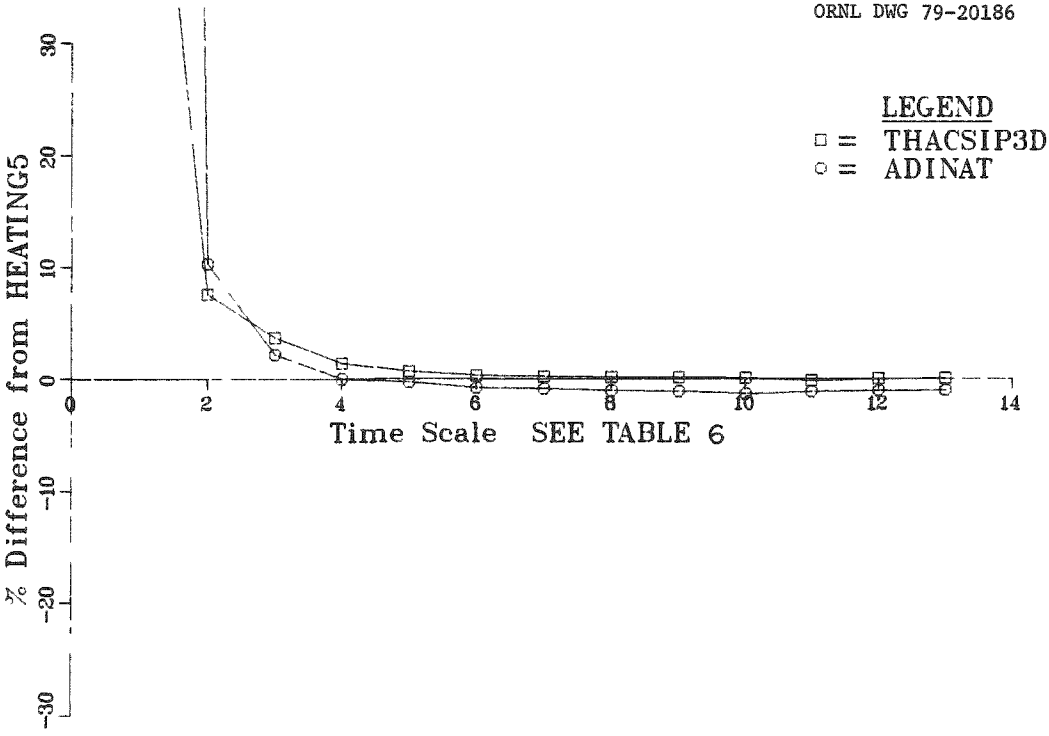

Fig. 57. Temperature vs time for location 8, plane 1.

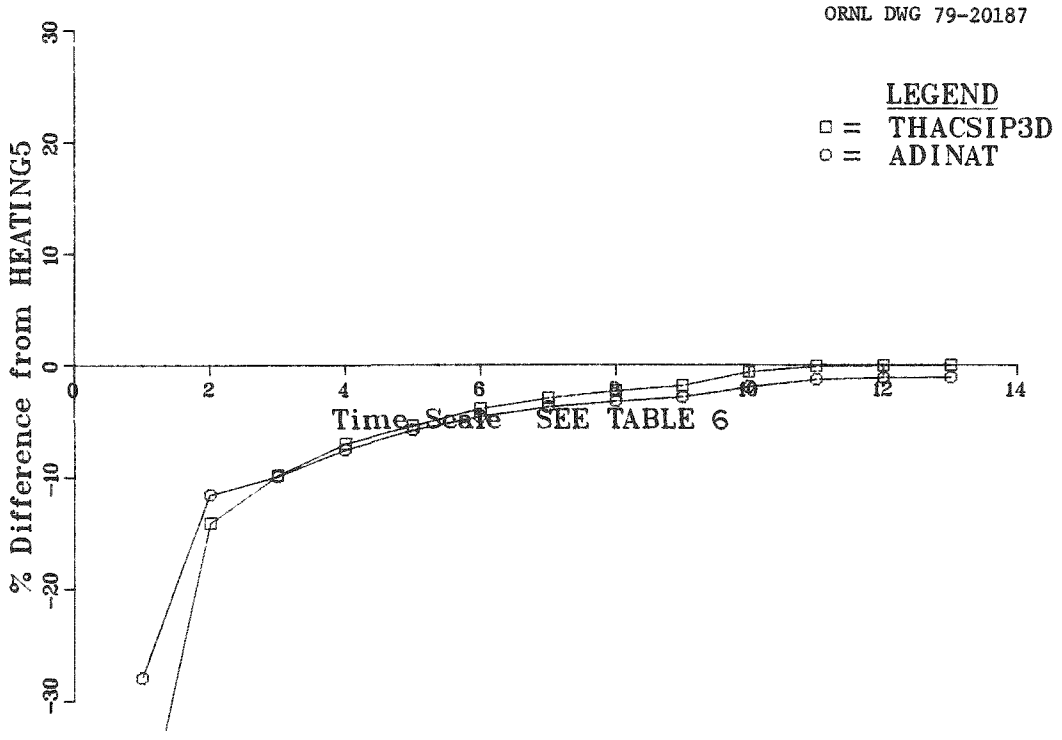

Fig. 58. Temperature vs time for location 1, plane 3.

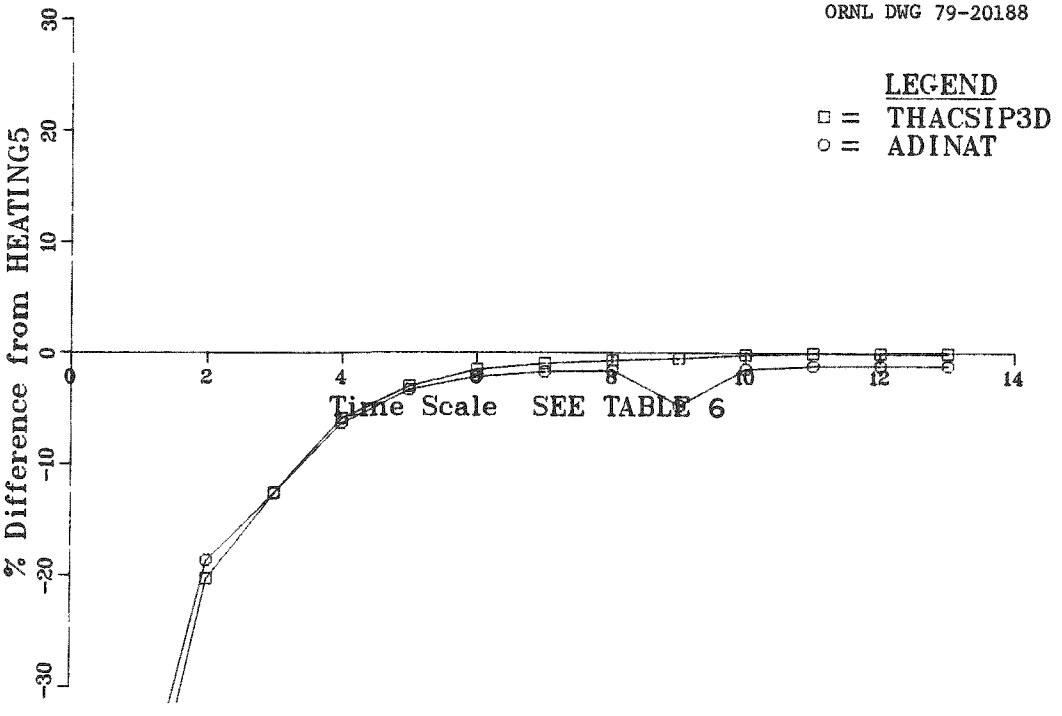

Fig. 59. Temperature vs time for location 2, plane 3.

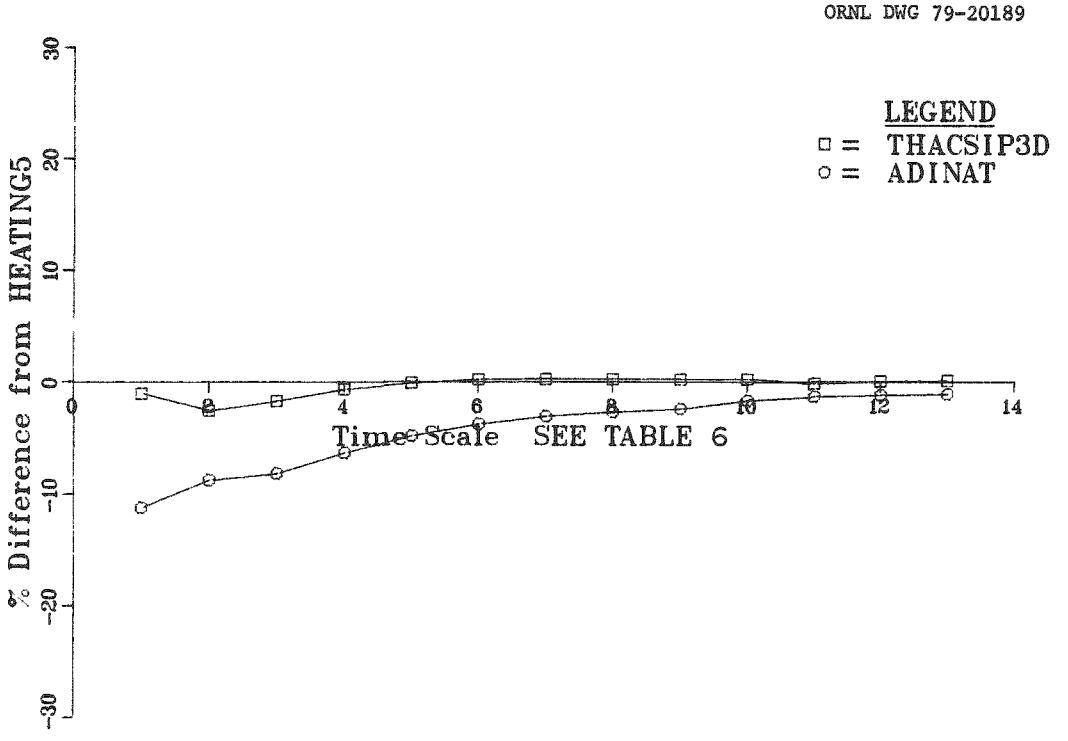

Fig. 60. Temperature vs time for location 3, plane 3.

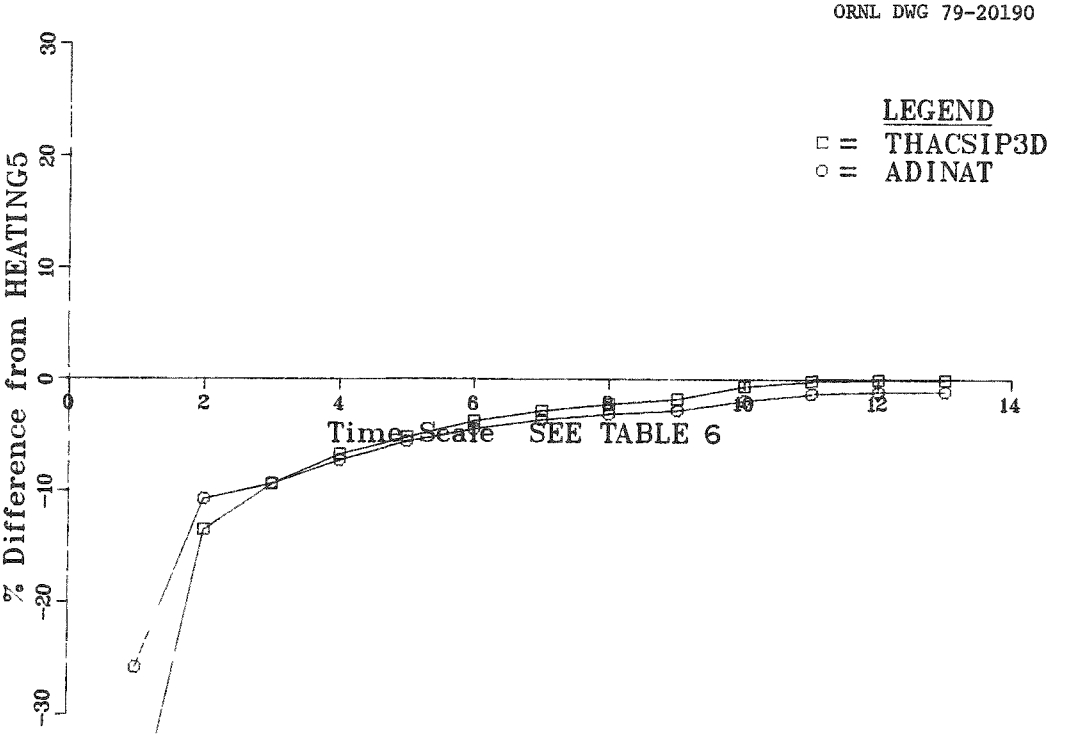

Fig. 61. Temperature vs time for location 4, plane 3.

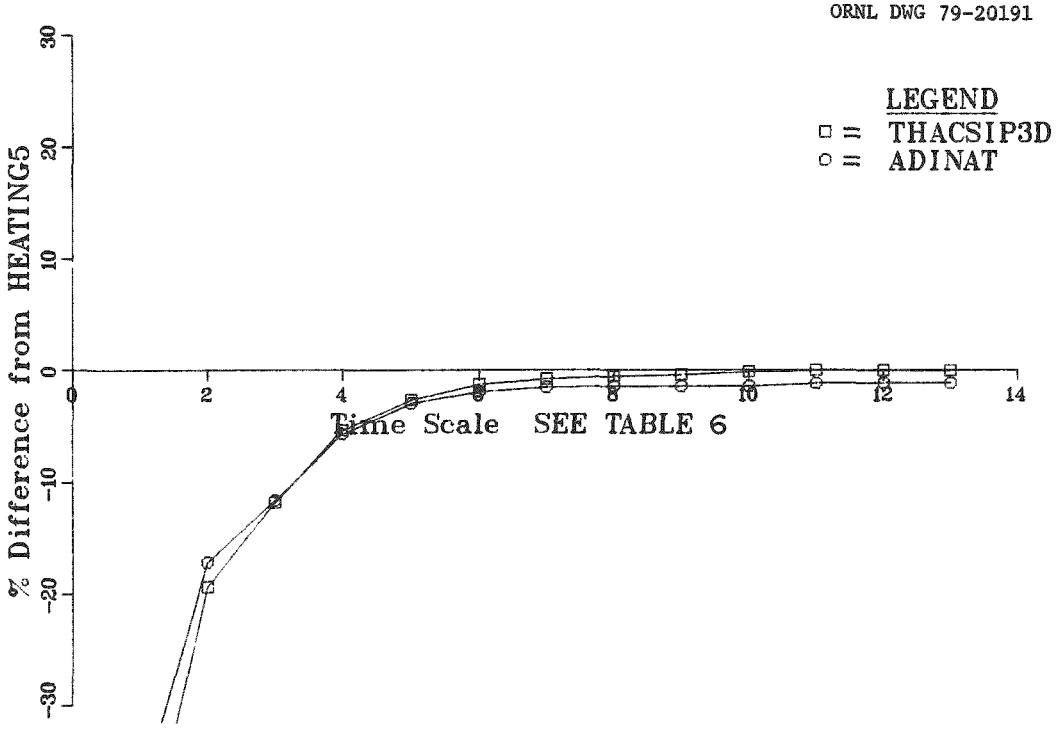

Fig. 62. Temperature vs time for location 5, plane 3.

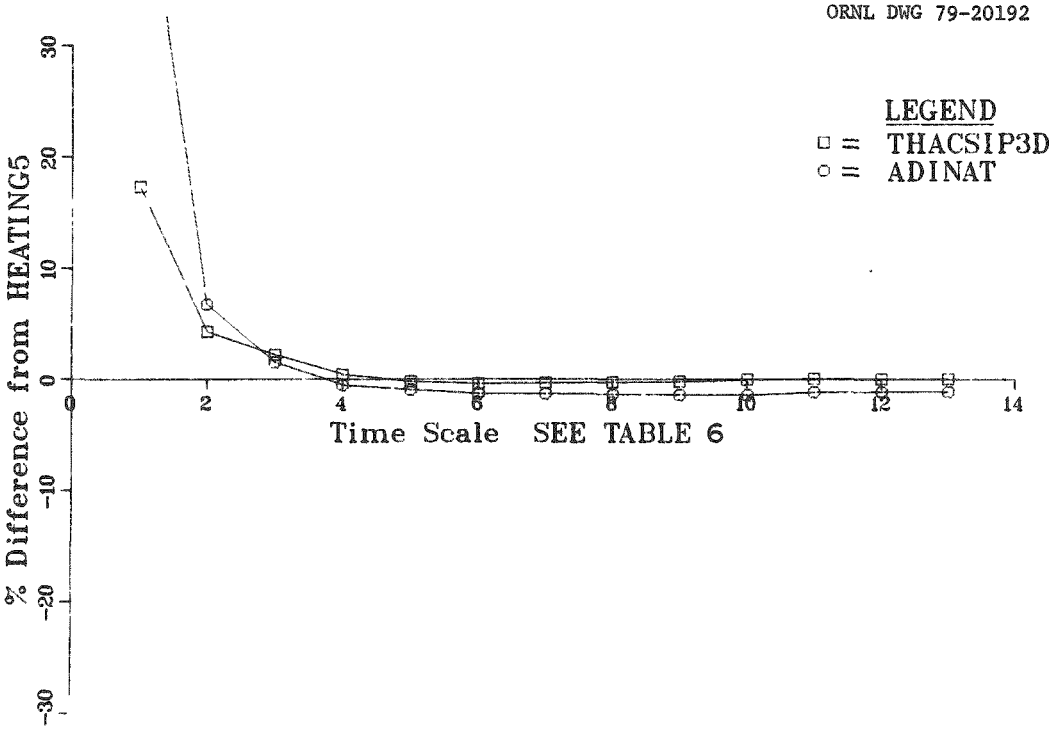

Fig. 63. Temperature vs time for location 6, plane 3.

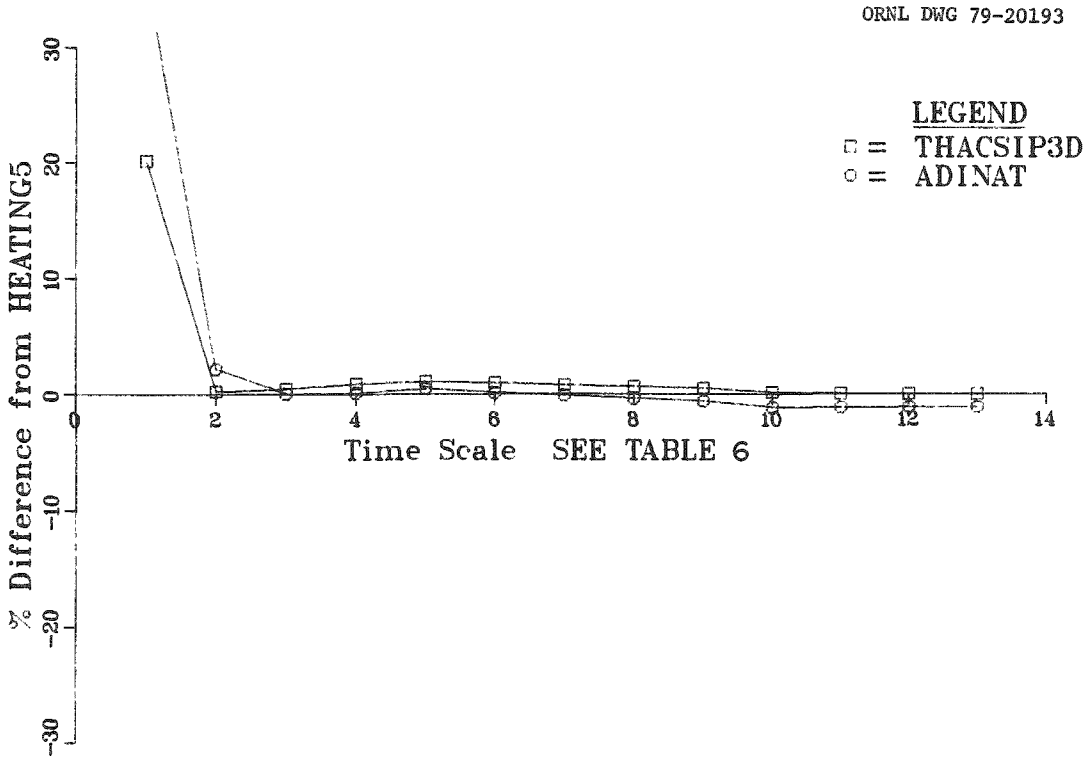

Fig. 64. Temperature vs time for location 7, plane 3.

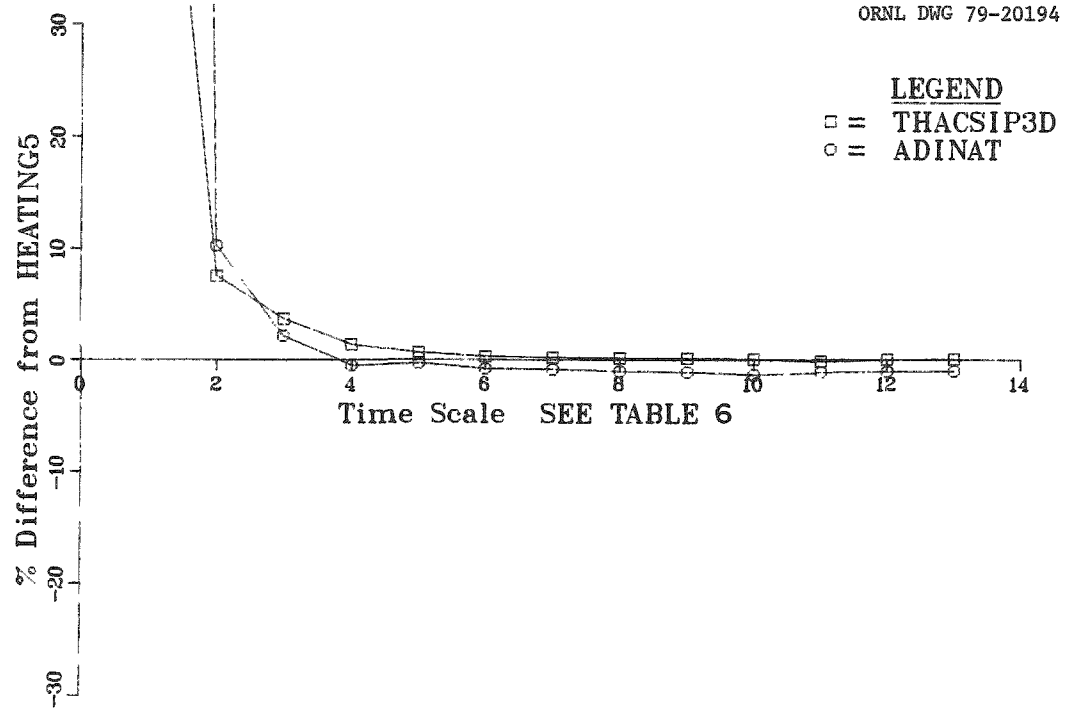

Fig. 65. Temperature vs time for location 8, plane 3.

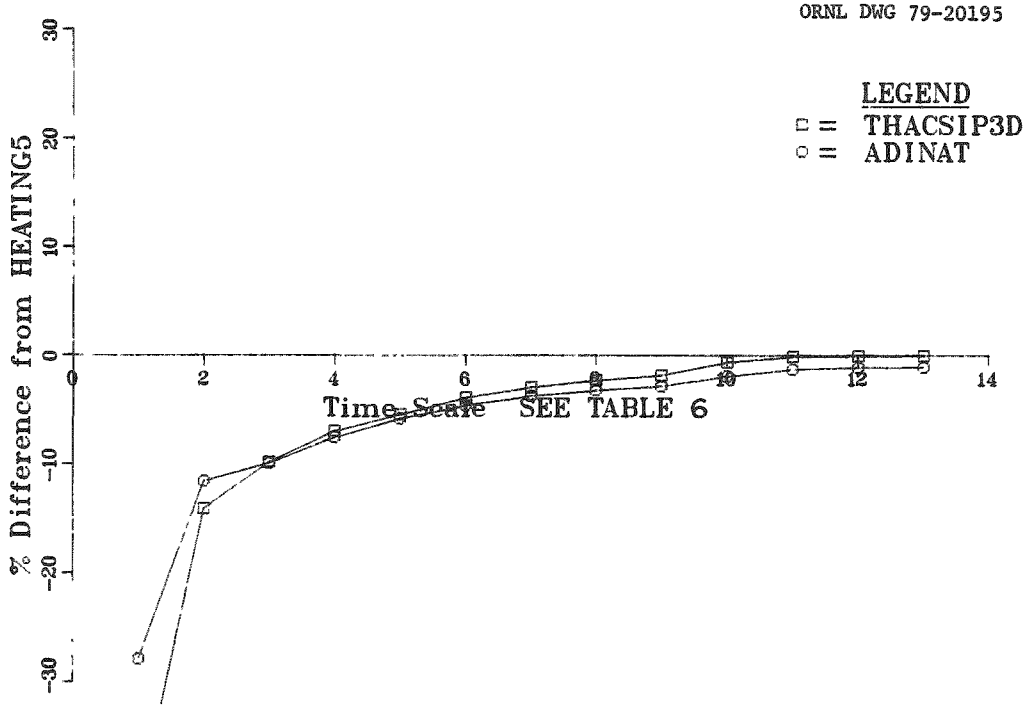

Fig. 66. Temperature vs time for location 1, plane 4.

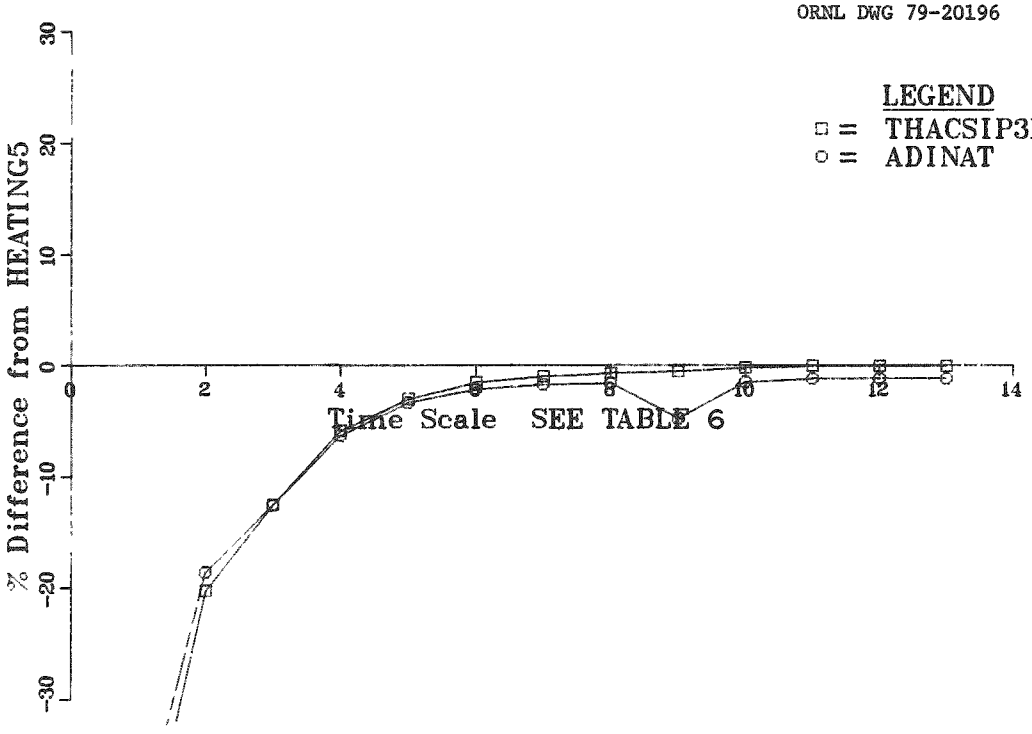

Fig. 67. Temperature vs time for location 2, plane 4.

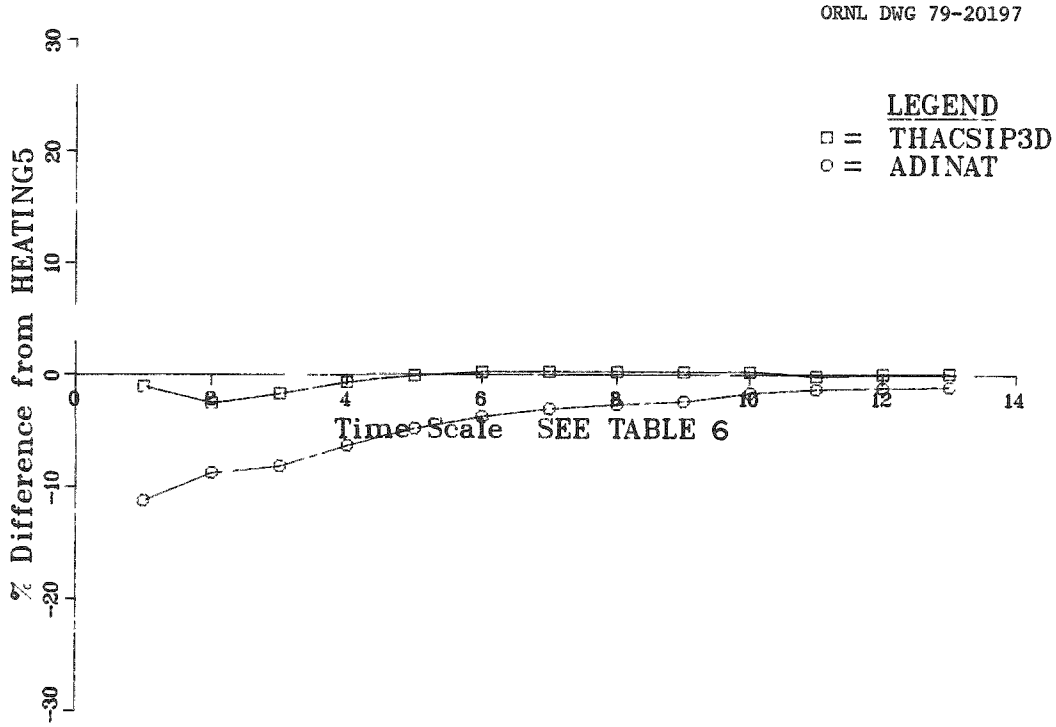

Fig. 68. Temperature vs time for location 3, plane 4.

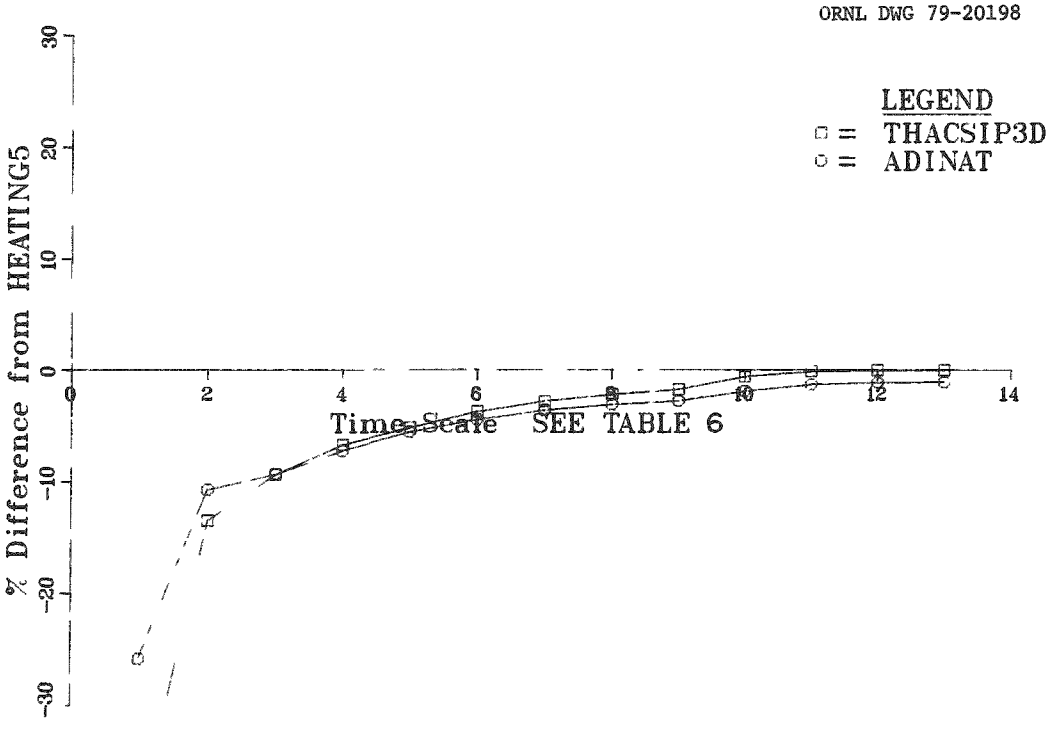

Fig. 69. Temperature vs time for location 4, plane 4.

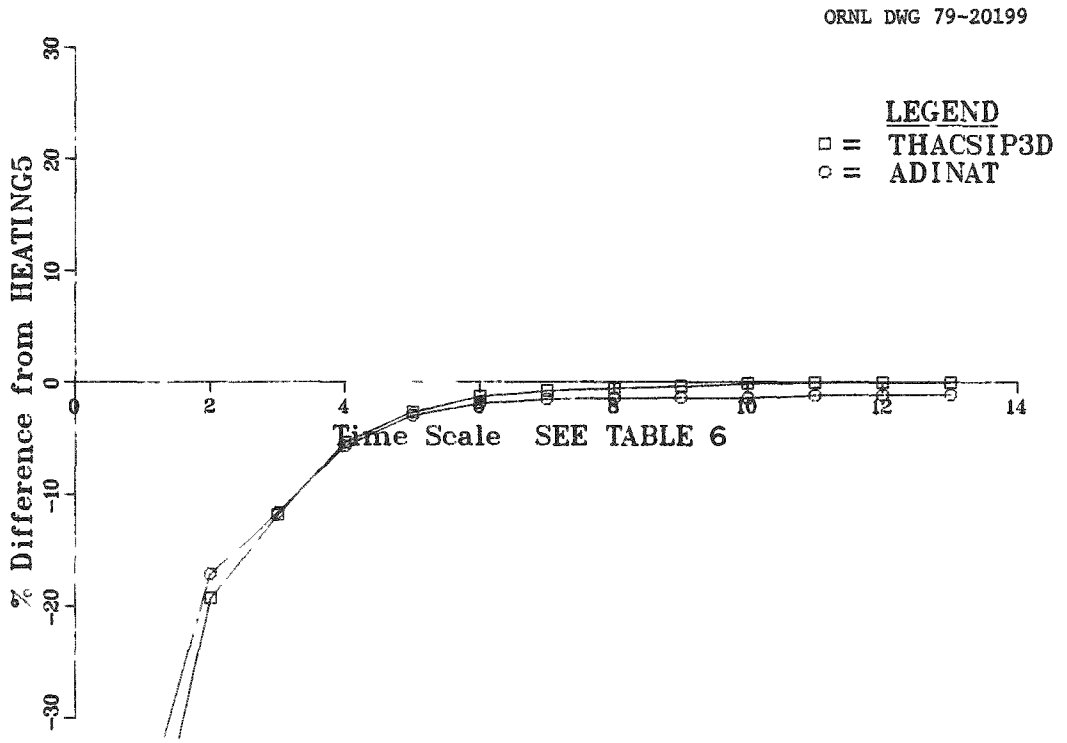

Fig. 70. Temperature vs time for location 5, plane 4.

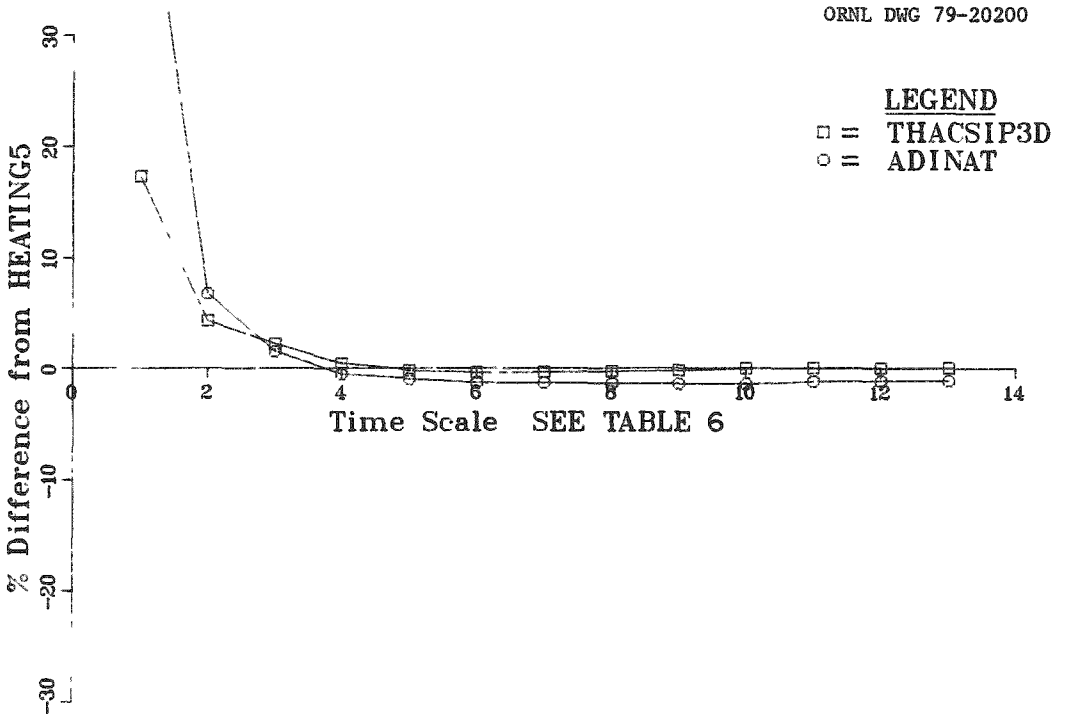

Fig. 71. Temperature vs time for location 6, plane 4.

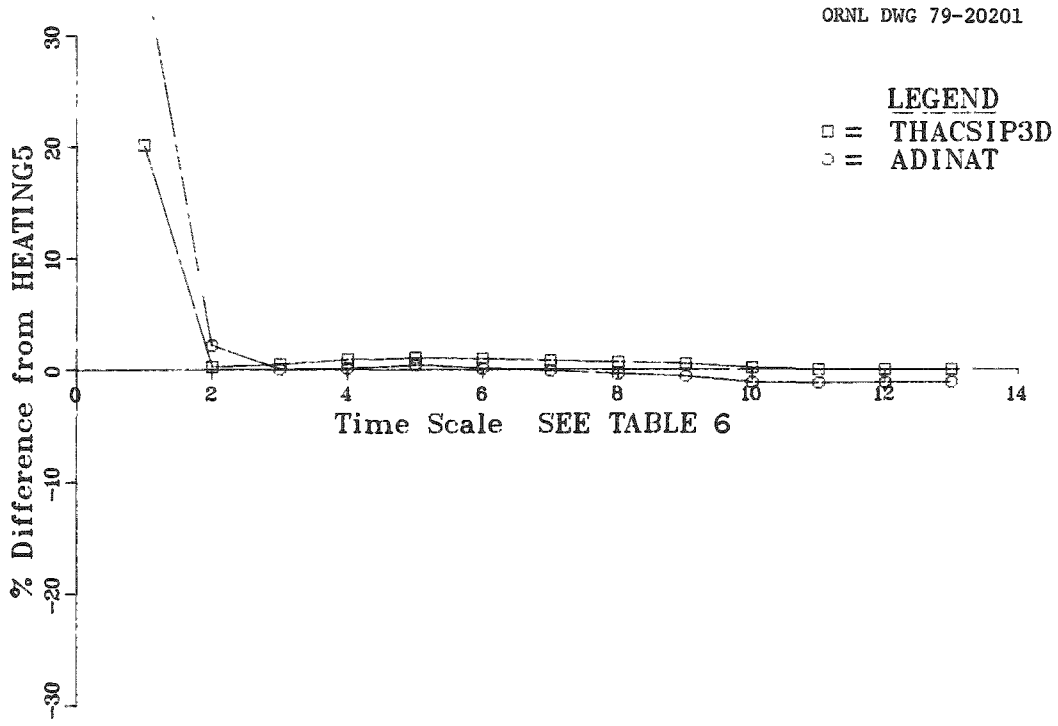

Fig. 72. Temperature vs time for location 7, plane 4.

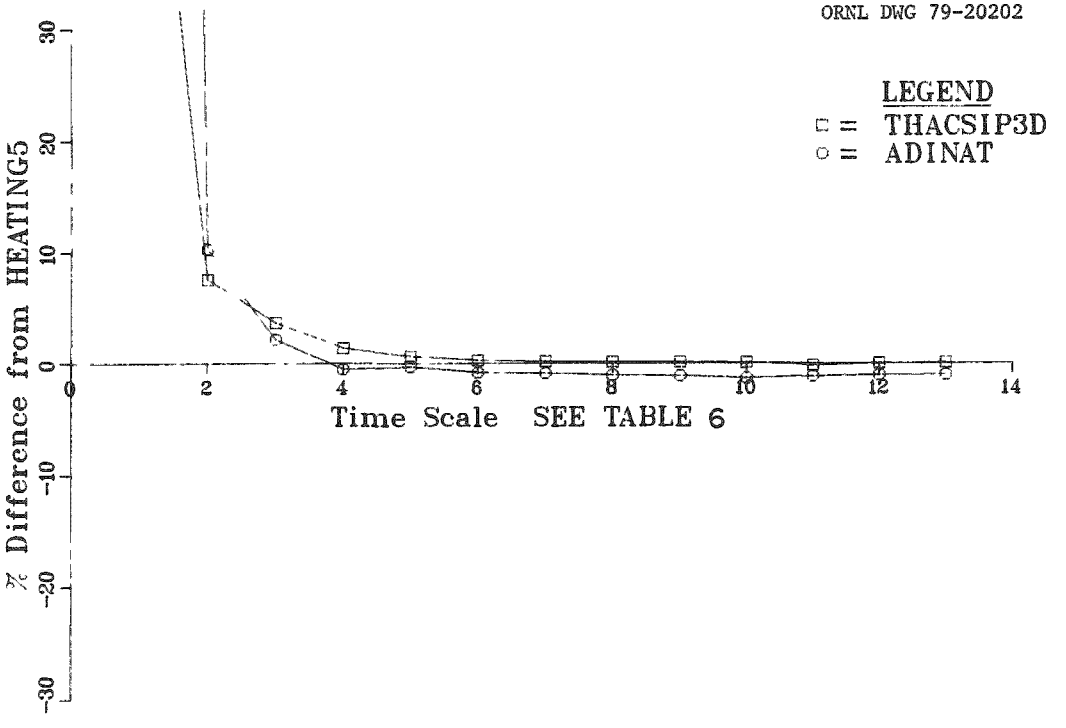

Fig. 73. Temperature vs time for location 8, plane 4.

7. ACKNOWLEDGMENTS

We are grateful to W. D. Turner and S. K. Iskander of the Union Carbide Nuclear Division Computer Sciences Division (UCC-ND, CSD) for their criticisms and assistance.

The calculations were made by:

K. W. Childs, UCC-ND, CSD, ADINAT;

J. L. Ratigan, RE/SPEC Inc., TRANCO (SPECTRUM-41);

R. F. Graham, Science Applications, Inc., SINDA.

8. REFERENCES

1. W. D. Turner, D. C. Elrod, and I. I. Siman-Tov, HEATING5 - An IBM 360 Heat Conduction Program, ORNL/CSD/TM-15 (March 1977).

2. W. D. Turner, THAC-SIP-3D - A Three-Dimensional, Transient Heat Analysis Code Using the Strongly Implicit Procedure, K/CSD/TM-24 (September 1978).

3. K. J. Bathe, ADINAT - A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis of Temperatures, MIT-82448-5 (May 1977).

4. J. P. Smith, Systems Improved Numerical Differencing Analyzer User's Manual, NASA-CR-134271 (TRW 14690-H001-R0-00) (April 1971).

5. A. L. Edwards, TRUMP: A Computer Program for Transient and Steady-State Temperature Distributions and Multidimensional Systems, UCRL-14754 Rev3 (September 1972).

6. G. D. Callahan, Documentation of the Heat Transfer Code TRANCO, RSI-0037, ORNL/SUB-4269/15 (August 1975).

7. Swanson Analysis Systems, Inc., ANSYS-Engineering Analysis Computer Program, Version 3 (1978).

8. C. T. Hsu, A Comparison of a Finite Element and a Finite Difference Computer Code in Heat Transfer Calculations, ASME Paper No. 79-PVP-63, presented at the Pressure Vessel and Piping Conference, San Francisco, June 25-29, 1979.

9. R. R. Liguori and J. W. Stephenson, The HEATING Program, ASTRA 417-5.0 (January 1961).

10. T. B. Fowler and E. R. Volk, Generalized Heat Conduction Code for the IBM-704 Computer, ORNL-2734 (October 1959).

11. W. D. Turner and M. Siman-Tov, HEATING3 - An IBM 360 Heat Conduction Program, ORNL/TM-3208 (February 1971).

12. R. D. Richtmyer and K. W. Morton, "Difference Methods for Initial-Value Problems," Interscience Publishers, New York, London, and Sydney, 1967.

13. H. L. Stone, Iterative Solution of Implicit Approximations of Multi-dimensional Partial Differential Equations, SIAM J. Numer. Anal. 5(3), 530-58 (1968).

14. H. W. Weinstein, H. L. Stone, and T. V. Kwan, Iterative Procedure for Solution of Systems of Parabolic and Elliptic Equations in Three Dimensions, Ind. Eng. Chem. Fundam. 8 (2), 281-27 (1969).

15. K. J. Bathe, ADINA - A Finite Element Program for Automatic Dynamic Nonlinear Analysis, MIT-82448-1 (September 1975; rev. May 1977).

16. D. R. Lewis, J. D. Gaski, and L. R. Thompson, Chrysler Improved Numerical Differencing Analyzer - Third Generation, TN-AP-67-287 (October 1967).

17. G. D. Callahan and A. F. Fossum, Data Input Manual for RSI/TRANCO: A Finite Element Heat Conduction Computer Program, RSI-0049, Y/OWI/SUB-77/22303/1 (February 1977).

18. J. L. Ratigan, RE/SPEC Inc., private communication.

19. Taken from ref. 5.

INTERNAL DISTRIBUTION

1. J. O. Blomeke
2. K. W. Childs
3-22. H. C. Claiborne
23. R. W. Glass
24. V. A. Jacobs
25. R. A. Just
26. G. H. Llewellyn
27. T. W. Pickel
28. R. S. Wagner
29-30. Laboratory Records
31. Laboratory Records - RC
32. Central Research Library
33. Y-12 Technical Library
Document Reference Section
34. ORNL Patent Office

EXTERNAL DISTRIBUTION

Battelle Memorial Institute, Office of Nuclear Waste Isolation, 505 King Avenue, Columbus, OH 43201
35. W. A. Carbiener
36-40. N. E. Carter
41-45. G. E. Raines
46. ONWI Library

NUKEM GmbH, Postfach 110080, D-6450 Hanau 11, Federal Republic of Germany
47. R. Schönfeld

Physikalisch-Technische Bundesanstalt, D3300 Braunschweig, Bundesalle 100, Federal Republic of Germany
48. F. Oesterle

RE/SPEC Inc., P. O. Box 725, Rapid City, SD 57709
49. W. C. McClain
50. J. L. Ratigan

Sandia Laboratories, P.O. Box 5800, Albuquerque, NM 87185
51. J. R. Wayland--5413

Science Applications, Inc., P.O. Box 843, Oak Ridge, TN 37830
52. R. F. Graham
53. L. D. Rickertsen

U.S. Department of Energy, Office of Nuclear Waste Management, Washington, DC 20545

54. C. R. Cooley
55. W. Eister

U.S. Department of Energy, Oak Ridge Operations, Radioactive Waste Management Division, Oak Ridge, TN 37830

56-57. D. E. Large

U.S. Department of Energy, Richland Operations Office-Columbus, 505 King Avenue, Columbus, OH 43201

58. J. O. Neff

59. Assistant Manager, Energy Research and Development, DOE-ORO, Oak Ridge, TN 37830

60-86. Technical Information Center, DOE-ORO, Oak Ridge, TN 37830