

Contents

1	Introduction	3
2	Simulation Capabilities	5
2.1	Governing Equations	5
2.2	Geometry	8
2.2.1	Convention and Restrictions	8
2.2.2	Moving Geometry	9
2.3	Boundary Conditions	10
2.3.1	Fluid Velocity	11
2.3.2	Temperature	13
2.3.3	Passive scalar	13
2.3.4	Internal Boundary Conditions	13
2.4	Initial Conditions	15
2.5	Material Properties and Forcing Functions	15
3	Numerical Methods	18
3.1	Spatial Discretization	18
3.2	Temporal Discretization	25
3.3	Solution Methods	27
3.4	Method Tradeoffs	29
4	Introductory NEKTON Usage	31
4.1	Site-Specific Considerations	31
4.1.1	Operating Systems	32
4.1.2	Hardware	33

4.2	Sample Sessions	38
4.2.1	Notation	38
4.2.2	Using the Mouse Buttons	39
4.2.3	Problem 1: Steady Two-Dimensional Heat transfer	40
4.2.4	Problem 2: An Unsteady Two-Dimensional Navier-Stokes Problem with Natural Convection	65
4.3	Tips for Saving Time	88
4.3.1	Reducing the Setup Time Required by PRENEK	88
4.3.2	Reducing the CPU Time Required by NEKTON	90
5	PRENEK	93
5.1	Structure	93
5.1.1	Menu Structure	93
5.1.2	File Structure	95
5.2	BUILD Menu	96
5.3	BOUNDARY CONDITION Menu	97
5.4	Using the GLOBAL REFINE submenu	98
5.4.1	Two-Dimensional Mesh refinement	100
5.4.2	Extensions to Three-dimensional Meshes	106
5.4.3	Boundary Conditions and Split elements	108
5.5	Using the Variable Properties Menu	108
5.6	Using the Fortran Function Option	109
5.7	Special moving boundary features	110
6	NEKTON	113
6.1	Command Procedures	113
6.2	File Structure	114
7	POSTNEK	116
7.1	POSTNEK Menu Structure	116
7.2	Hardcopy	118
8	The Menu System	119
8.1	PRENEK Menus	120

8.2	POSTNEK Menus	167
9	Troubleshooting NEKTON	188
9.1	Preprocessor Difficulties	189
9.2	NEKTON Solver Difficulties	190
9.3	Postprocessor Difficulties	191
A	Restrictions in Capabilities and Numerical Methods	193
A.1	Restrictions in Physical Capabilities	194
A.2	Restrictions in Numerical Methods	195

Chapter 1

Introduction

NEKTON is a computer code for the simulation of steady and unsteady incompressible fluid flow and heat transfer, as well as optional convective-diffusive passive scalar quantities. The computational domain can be either stationary or moving (free surface, moving boundary, fluid layers, Stefan). If both the fluid and heat transfer simulation capabilities are used, they can be coupled via forced and/or natural convection heat transfer. A conjugate heat transfer, solid and fluid problem can also be solved. The package consists of three parts: PRENEK, NEKTON, and POSTNEK.

PRENEK is an interactive menu-driven program in which the user inputs the necessary geometrical (mesh), physical (material properties, boundary conditions, initial conditions), and numerical (discretization, solver, i/o) parameters to completely specify the computational problem. NEKTON is the computational engine that actually performs the numerical integration of the Navier-Stokes and energy equations, while POSTNEK is an interactive graphical package which allows the user to inspect and analyze the results of a NEKTON simulation. NEKTON is based on the spectral element method, a high-order finite element technique for solution of partial differential equations.

This documentation gives detailed instructions for use of NEKTON for solving a large variety of complex fluid flow and heat transfer problems. Although some background in fluid mechanics and heat transfer is required, no assumptions are made as to proficiency in numerical analysis. In particular, although we do briefly describe the general structure of the spectral element method in Section 3,

no details of the underlying numerical algorithms are given, except to the extent that such knowledge is necessary for efficient application of NEKTON or correct interpretation of diagnostics.

In Chapter 2, the capabilities of the NEKTON computational package are discussed both in terms of the physical problem statement and the associated set of governing partial differential equations. The parameter values, geometries, boundary conditions, and initial conditions that NEKTON currently accomodates are described in detail. Chapter 3 presents an overview of those parts of the numerical methods which the user must appreciate to use NEKTON effectively. A discussion of method tradeoffs is also given. Chapter 4 guides the user through the process of running NEKTON, beginning with a discussion of the site- specific hardware and software environment, and proceeding with two examples. These introductory sessions begin with the problem definition in PRENEK, and proceed through to the analysis of the NEKTON simulation results using POSTNEK. The purpose of these sample sessions is to illustrate the overall construction and workings of the NEKTON package, and to give the user a general framework from which to interpret the detailed command descriptions that is given in subsequent chapters. The chapter ends with some tips for saving time. Chapters 5, 6, and 7 give a discussion of PRENEK, NEKTON, and POSTNEK, respectively; these sections describe the capabilities, structure, commands, menus, file organization, and usage strategies of each of these programs. Chapter 8 defines in detail each of the menus and menu items in PRENEK and POSTNEK. These menus are organized alphabetically for quick reference. An index of the individual menu items with page reference is also included at the end of the documentation. Chapter 9 is a troubleshooting guide, in which potential problems with NEKTON usage is discussed alphabetically according to the symptoms associated with these problems. This section is intended to provide a means of quick diagnosis for solution of these potential difficulties. Lastly, an appendix summarizes the restrictions in capabilities and numerical methods in NEKTON. This appendix, together with Chapter 8, are meant to be the the reference portion of the documentation.

Chapter 2

Simulation Capabilities

In this chapter we present the capabilities of NEKTON by presenting the set of governing equations, together with boundary conditions, initial conditions, and material property/forcing function parameters that the program can solve numerically.

2.1 Governing Equations

NEKTON solves the unsteady incompressible two-dimensional, axisymmetric, or three-dimensional Navier-Stokes (fluid flow) equations with forced or natural convection heat transfer in both stationary (fixed) or time-dependent geometry.

The solution variables are the fluid velocity $\mathbf{u} = (u_x, u_y, u_z)$, the pressure p , the temperature T , up to nine independent passive scalar fields $\phi_i, i=1,2,\dots,9$ and, for time-dependent geometry, the mesh velocity $\mathbf{w} = (w_x, w_y, w_z)$; all of the above field variables are functions of space $\mathbf{x} = (x, y, z)$ and time t in domains D_F and/or D_S defined in Fig. 2.1. The governing equations used are :

the momentum equations in D_F ,

$$\rho(\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}) = -\nabla p + \nabla \cdot [\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] - \rho \bar{\beta} T \hat{\mathbf{g}} + \rho \mathbf{f} , \quad (2.1)$$

the continuity (mass conservation) equation in D_F ,

$$\nabla \cdot \mathbf{u} = 0 , \quad (2.2)$$

the energy equation in $D_F \cup D_S$,

$$\rho c_p [\partial_t T + (\mathbf{u} \cdot \nabla)T] = \nabla \cdot (k \nabla T) + q_{vol} , \quad (2.3)$$

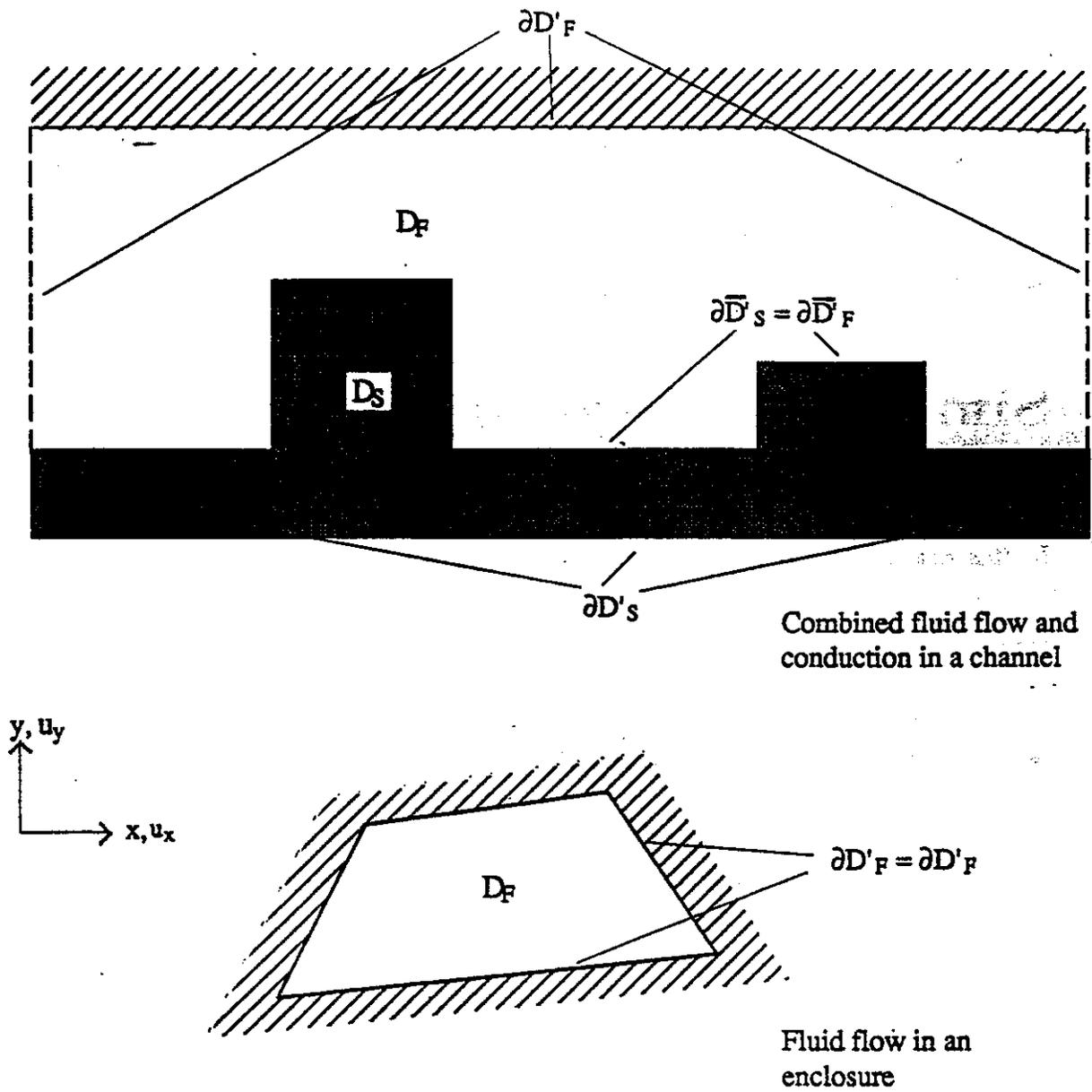


Figure 2.1 Illustration of the domain and boundary definitions used in this document

- Notation: D_F - Fluid domain
 D_S - Solid domain
 $\partial D'_F(\partial D'_S)$ - The part of $D_F(D_S)$ boundary not shared by $D_S(D_F)$
 $\partial \bar{D}'_F(\partial \bar{D}'_S)$ - The part of $D_F(D_S)$ boundary shared by $D_S(D_F)$
 ∂D_F - The entire fluid domain boundary, $\partial D'_F \cup \partial \bar{D}'_F$

and a convective-diffusive equation for each passive scalar i , $i=1,2,\dots,9$ in $D_F \cup D_S$ or in D_F ,

$$(\rho c_p)_\phi [\partial_t \phi_i + (\mathbf{u} \cdot \nabla) \phi_i] = \nabla \cdot (k_\phi \nabla \phi_i) + (q_{vol})_\phi. \quad (2.4)$$

The above governing equations are subject to boundary conditions and initial conditions described in the following sections.

It should be noted that NEKTON can solve not only the above fully coupled system, but also various subsets of these equations. Namely, NEKTON recognizes the following subsets.

- (1) Fluid flow, heat transfer, and an arbitrary subset of passive scalar fields; i.e., the momentum, continuity, energy and convective-diffusive equations for which $i < 9$.
- (2) Fluid flow and heat transfer only; i.e., the momentum, continuity and energy equations only.
- (3) Fluid flow only; i.e., the momentum and continuity equations only.
- (4) Fluid flow with the nonstress formulation; i.e., the momentum equation, in which the term $(\nabla \mathbf{u})^T$ in the viscous contribution is set to zero, the continuity equation and with or without the energy and convective-diffusive equations.
- (5) Unsteady Stokes flow only; i.e., the momentum equation, in which and the nonlinear term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ is set to zero, and the continuity equation.
- (6) Steady Stokes flow only; i.e., the momentum equation, in which and the nonlinear term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ and the transient term $\partial_t \mathbf{u}$ are set to zero, and the continuity equation.
- (7) Transient conduction heat transfer; i.e., the energy equation only.
- (8) Steady conduction heat transfer; i.e., only the energy equation in which and the nonlinear term $(\mathbf{u} \cdot \nabla) T$ and the transient term $\partial_t T$ are set to zero.

Thus any combination of the equation characteristics given in the SET EQUATION menu is permissible with the following restrictions. First, the equation must be set to unsteady if it is time-dependent (i.e., options 1,2,3,4,5,7), or if there is advection (i.e., options 1,2,3,4), or if there are materials properties which are functions

of time. For these cases, the steady-state (if it exists) is found as stable evolution of the initial-value-problem. Secondly, stress formulation must be selected if there are traction boundary conditions applied on any fluid boundary, or if any mixed velocity/traction boundaries, such as symmetry and outflow/n, are not aligned with either one of the Cartesian x, y or z axes. Using the minimum set of equations possible for a given application clearly saves both in user time (PRENEK automatically screens out irrelevant aspects of the problem definition) and computational time (NEKTON solves only those equations specified).

2.2 Geometry

2.2.1 Convention and Restrictions

The domain in which the fluid flow/heat transfer problem is solved consists of two distinct subdomains. The first subdomain is that part of the region occupied by fluid, denoted D_F , while the second subdomain is that part of the region occupied by a solid, denoted D_S . These two subdomains are depicted in Fig. 2.1 for several different geometries. The entire domain is denoted as $D = D_F \cup D_S$. The fluid problem is solved in the domain D_F , while the temperature in the energy equation is solved in the entire domain; the passive scalars can be solved in either the fluid or the entire domain.

We denote the entire boundary of D_F as ∂D_F , that part of the boundary of D_F which is not shared by D_S as $\partial D'_F$, and that part of the boundary of D_F which is shared by D_S as $\partial \bar{D}'_F$ (note $\partial D_F = \partial D'_F \cup \partial \bar{D}'_F$). In addition, ∂D_s , $\partial D'_s$ and $\partial \bar{D}'_s$ are analogously defined. These distinct portions of the domain boundary are illustrated in Fig. 2.1. The restrictions on the domain for NEKTON are as follow.

- The domain $D = D_F \cup D_S$ must correspond either to a planar (Cartesian) two-dimensional geometry, or to the cross-section of an axisymmetric region specified by revolution of the cross-section about a specified axis, or by a (Cartesian) three-dimensional geometry.
- For two-dimensional and axisymmetric geometries, the boundaries of both

subdomains, ∂D_F and ∂D_S , must be representable as (or at least approximated by) the union of straight line segments, splines, or circular arcs.

- NEKTON can interpret a two-dimensional image as either a planar Cartesian geometry, or the cross-section of an axisymmetric body. In the case of the latter, it is assumed that the y -direction is the radial direction, that is, the axis of revolution is at $y=0$. Although an axisymmetric geometry is, in fact, three-dimensional, NEKTON can assume that the field variables are also axisymmetric (that is, do not depend on azimuth, but only y , that is, radius, x , and t), thus reducing the relevant equations to "two-dimensional" form.
- In the geometry generation using PRENEK, a three-dimensional geometry must correspond to a series of layers of planar two-dimensional geometries. Each of those layers must adhere to the rules governing planar two-dimensional geometries. In addition, it is assumed that the parallel planes which separate the layers of elements are oriented parallel to the x - y plane, and that the first such plane is located at $z = 0$. Subsequent planes are located at arbitrarily increasing values of z . Note that this is a PRENEK restriction as NEKTON can readily accept general three-dimensional geometries.

2.2.2 Moving Geometry

If the imposed boundary conditions allow for motion of the boundary during the solution period (for example, moving walls, free-surfaces, melting fronts, fluid layers), then the geometry of the computational domain is automatically considered in NEKTON as being time-dependent.

For time-dependent geometry problems, a mesh velocity w is defined at each collocation point of the computational domain (mesh) to characterize the deformation of the mesh. In the solution of the mesh velocity, the value of the mesh velocity at the moving boundaries are first computed using appropriate kinematic conditions (for free-surfaces, moving walls and fluid layers) or dynamic conditions (for melting fronts). On all other external boundaries, the normal mesh velocity on the boundary is always set to zero. In the tangential direction, either a zero tangential velocity condition or a zero tangential traction condition is imposed;

this selection is automatically performed by NEKTON based on the fluid and/or thermal boundary conditions specified on the boundary. However, under special circumstances the user may want to override the defaults set by NEKTON, this is described in the PRENEK manual in Section 5.7. If the zero tangential mesh velocity is imposed, then the mesh is fixed in space; if the zero traction condition is imposed, then the mesh can slide along the tangential directions on the boundary. The resulting boundary-value-problem for the mesh velocity is solved in NEKTON using an elastostatic solver, with the Poisson ratio typically set to zero. The new mesh geometry is then computed by integrating the mesh velocity explicitly in time and updating the nodal coordinates of the collocation points.

Note that the number of macro-elements, the order of the macro-elements and the topology of the mesh remain *unchanged* even though the geometry is time-dependent. The use of an arbitrary-Lagrangian-Eulerian description in NEKTON ensures that the moving fronts are tracked with the minimum amount of mesh distortion; in addition, the elastostatic mesh solver can handle moderately large mesh distortion. However, it is the responsibility of the user to decide when will a mesh become "too deformed" and thus requires remeshing. The execution of the program will terminate when the mesh becomes unacceptable, that is, a one-to-one mapping between the physical coordinates and the isoparametric local coordinates for any macro-element no longer exists.

2.3 Boundary Conditions

The boundary conditions for the governing equations given in the previous section are described in the following. Note that if the boundary conditions (for any field variable) are nonzero, the inhomogeneities can either be defined as constant, or as fortran functions of appropriate parameters such as space, time, temperature, etc. In this case, the user is responsible for using relevant variables in all fortran function definitions.

2.3.1 Fluid Velocity

Two types of boundary conditions are applicable to the fluid velocity : essential (Dirichlet) boundary condition in which the velocity is specified; natural (Neumann) boundary condition in which the traction is specified. For segments that constitute the boundary ∂D_F (refer to Fig. 2.1), one of these two types of boundary conditions must be assigned to each component of the fluid velocity. The fluid boundary condition can be *all Dirichlet* if all velocity components of u are specified; or it can be *all Neumann* if all traction components $t = [-pI + \mu(\nabla u + (\nabla u)^T)] \cdot n$, where I is the identity tensor, n is the unit normal and μ is the dynamic viscosity, are specified; or it can be *mixed Dirichlet/Neumann* if Dirichlet and Neumann conditions are selected for different velocity components. Examples for all Dirichlet, all Neumann and mixed Dirichlet/Neumann boundaries are wall, free-surface and symmetry, respectively. If the nonstress formulation is selected, then traction is not defined on the boundary. In this case, any Neumann boundary condition imposed must be homogeneous; i.e., equal to zero. In addition, mixed Dirichlet/Neumann boundaries must be aligned with one of the Cartesian axes.

For flow geometry which consists of a periodic repetition of a particular geometric unit, the periodic boundary conditions can be imposed, as illustrated in Fig. 2.2. For a fully-developed flow in such a configuration, one can effect great computational efficiencies by considering the problem in a single geometric unit (here taken to be of length L), and requiring periodicity of the field variables. NEKTON requires that the pairs of sides (or faces, in the case of a three-dimensional mesh) identified as periodic be identical (i.e., that the geometry be periodic).

For an axisymmetric flow geometry, the axis boundary condition is provided for boundary segments that lie entirely on the axis of symmetry. This is essentially a symmetry (mixed Dirichlet/Neumann) boundary condition in which the normal velocity and the tangential traction are set to zero.

For free-surface boundary segments, the inhomogeneous traction boundary conditions involve both the surface tension coefficient σ and the mean curvature of the free surface.

Periodic channel

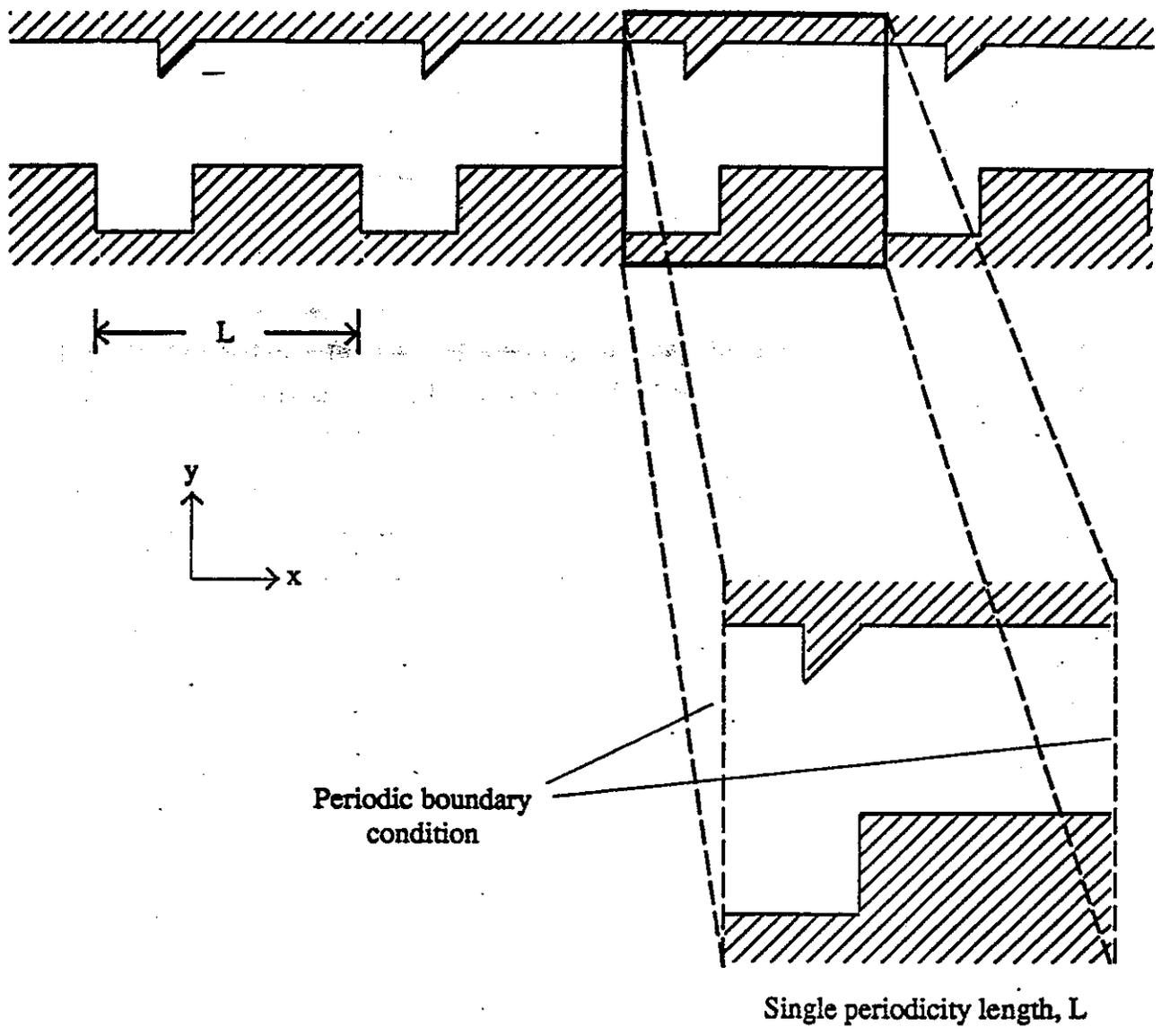


Figure 2.2 Example of a domain consisting of a periodic repetition of a particular geometric unit

2.3.2 Temperature

The three types of boundary conditions applicable to the temperature are: essential (Dirichlet) boundary condition in which the temperature is specified; natural (Neumann) boundary condition in which the heat flux is specified; and mixed (Robin) boundary condition in which the heat flux is dependent on the temperature on the boundary. For segments that constitute the boundary $\partial D'_F \cup \partial D'_S$ (refer to Fig. 2.1), one of the above three types of boundary conditions must be assigned to the temperature.

The two types of Robin boundary condition for temperature are : convection boundary conditions for which the heat flux into the domain depends on the heat transfer coefficient h_c and the difference between the environmental temperature T_∞ and the surface temperature; and radiation boundary conditions for which the heat flux into the domain depends on the Stefan-Boltzmann constant/view-factor product h_{rad} and the difference between the fourth power of the environmental temperature T_∞ and the fourth power of the surface temperature.

2.3.3 Passive scalar

The boundary conditions for the passive scalar fields are analogous to those used for the temperature field. Thus, the temperature boundary condition menu will reappear for each passive scalar field so that the user can specify an independent set of boundary conditions for each passive scalar field. The terminology and restrictions of the temperature equations are retained for the passive scalars, so that it is the responsibility of the user to convert the notation of the passive scalar parameters to their thermal analogues. For example, in the context of mass transfer, the user should recognize that the values specified for temperature and heat flux will represent concentration and mass flux, respectively.

2.3.4 Internal Boundary Conditions

In the spatial discretization, the entire computational domain is subdivided into macro-elements, the boundary segments shared by any two of these macro-elements in D_F and D_S are denoted as internal boundaries. For fluid flow analysis

with a single-fluid system or heat transfer analysis without change-of-phase, internal boundary conditions are irrelevant as the corresponding field variables on these segments are part of the solution. However, for a multi-fluid system and for heat transfer analysis with change-of-phase, special conditions are required at particular internal boundaries, as described in the following.

For a fluid system composed of multiple immiscible fluids, the boundary (and hence the identity) of each fluid must be tracked, and a jump in the normal traction exists at the fluid-fluid interface if the surface tension coefficient is nonzero. For this purpose, the interface between any two fluids of different identity must be defined as a special type of internal boundary, namely, a fluid layer; and the associated surface tension coefficient also needs to be specified.

In a heat transfer analysis with change-of-phase, NEKTON assumes that both phases exist at the start of the solution, and that all solid-liquid interfaces are specified as special internal boundaries, namely, the melting fronts. If the fluid flow problem is considered, i.e., the energy equation is solved in conjunction with the momentum and continuity equations, then only the common boundary between the fluid and the solid (i.e., all or portion of $\partial\bar{D}'_F$ in Fig. 2.1) can be defined as the melting front. In this case, segments on $\partial\bar{D}'_F$ that belong to a solid-liquid interface need to be specified by the user. NEKTON always assumes that the density of the two phases are the same (i.e., no Stefan flow); therefore at the melting front, the boundary condition for the fluid velocity is the same as that for a stationary wall, that is, all velocity components are zero. If no fluid flow is considered, i.e., only the energy equation is solved, then any internal boundary can be defined as a melting front. The temperature boundary condition at the melting front corresponds to a Dirichlet condition; that is, the entire segment maintains a constant temperature equal to the user-specified melting temperature T_{melt} throughout the solution. In addition, the volumetric latent heat of fusion ρL for the two phases, which is also assumed to be constant, should be specified.

2.4 Initial Conditions

For time-dependent problems NEKTON allows the user to choose among the following types of initial conditions for the velocity, temperature and passive scalars:

- Zero initial conditions: default; if nothing is specified.
- Fortran function: This option allows the user to specify the initial condition as a fortran function, e.g., as a function of x , y and z .
- Presolv: For a temperature problem the presolv option gives the steady conduction solution as initial condition for the temperature. For a fluid problem this option *can* give the steady Stokes solution as the initial condition for the velocity *provided* that the classical splitting scheme is *not* used.
- Restart: this option allows the user to read in results from an earlier simulation, and use these as initial conditions.

A tabulated summary of the compatibility of these initial condition options with various other solution strategies/parameters is given in the appendix.

2.5 Material Properties and Forcing Functions

The following restrictions are imposed on the material property and forcing function parameters in the governing equations and initial and boundary conditions described in the above sections (here and in what follows time-independent implies no variation in time, whereas constant refers to uniform in space).

- ρ , the density, is taken to be time-independent and constant; however, in a multi-fluid system different fluids can have different value of constant density.
- μ , the dynamic viscosity can vary arbitrarily in time and space; it can also be a function of temperature (if the energy equation is included) and strainrate invariants (if the stress formulation is selected).

- σ , the surface-tension coefficient can vary arbitrarily in time and space; it can also be a function of temperature and passive scalars.
- $\bar{\beta}$, the effective thermal expansion coefficient, is assumed time-independent and constant.
- \hat{g} , the unit vector in the direction of the acceleration of gravity, is taken to be time-independent and constant.
- $f(t)$, the body force per unit mass term can vary with time, space, temperature and passive scalars.
- ρc_p , the volumetric specific heat, can vary arbitrarily with time, space and temperature.
- ρL , the volumetric latent heat of fusion at a front, is taken to be time-independent and constant; however, different constants can be assigned to different fronts.
- k , the thermal conductivity, can vary with time, space and temperature.
- q_{vol} , the volumetric heat generation, can vary with time, space and temperature.
- h_c , the convection heat transfer coefficient, can vary with time, space and temperature.
- h_{rad} , the Stefan-Boltzmann constant/view-factor product, can vary with time, space and temperature.
- T_∞ , the environmental temperature, can vary with time and space.
- T_{melt} , the melting temperature at a front, is taken with time and space; however, different melting temperature can be assigned to different fronts.

In the solution of the governing equations together with the boundary and initial conditions, NEKTON treats the above parameters as pure numerical values; their physical significance depends on the user's choice of units. The system of units used is arbitrary (MKS, English, CGS, etc.). However, the system chosen

must be used consistently throughout. For instance, if the equations and geometry have been non-dimensionalized, the μ/ρ in the fluid momentum equation is in fact the inverse Reynolds number, whereas if the equations are dimensional, μ/ρ represents the kinematic viscosity with dimensions of *length*²/*time*. Similarly, for a natural convection problem, if the equations are non-dimensionalized, $\bar{\beta}$ would be the Rayleigh or Grashof number, whereas in dimensional terms it represents the product of the coefficient of thermal expansion and the acceleration of gravity.

Chapter 3

Numerical Methods

In this chapter we discuss the numerical methods used by NEKTON to solve the incompressible Navier-Stokes equations (2.1)-(2.2), and the energy equation (2.3). A basic knowledge about the algorithms could help the user achieve a more cost effective solution.

3.1 Spatial Discretization

The spatial discretization in NEKTON is based on the spectral element method, a high-order finite element method for partial differential equations. Here, the computational domain is represented as a set of (disjoint) macro-elements, with the solution, data and geometry being approximated by high-order polynomial expansions within each macro-element. This approach offers high-order (spectral) accuracy, while maintaining the geometric flexibility of low-order finite-element techniques. In the rest of this section we discuss the key features of the spectral element method, emphasizing the mesh generation.

Domain Decomposition. For two-dimensional domains, the isoparametric spectral element spatial discretization proceeds by first breaking the two-dimensional domain up into (non-degenerate) quadrangles, which we denote as "macro-" or "spectral" elements. An example of such a decomposition is shown in Fig. 3.1a. Three-dimensional domains are broken up into "bricks", in which the two horizontal parallel faces (top and bottom) are nondegenerate quadrangles, see Figure 3.2.

Elemental Mesh. Within each element, a local Cartesian mesh is constructed corresponding to a $N \times N(\times N)$ tensor-product Gauss-Lobatto Legendre collocation points, as shown in Fig 3.1b. The Gauss-Lobatto points are clustered near elemental boundaries, and are chosen because of their accurate approximation, interpolation and quadrature properties. The Δx , Δy , Δz notation used in this manual refers to the distance between the collocation points.

Element and Continuity Requirements. The requirements that NEKTON places on individual spectral elements are:

(E1) Two-dimensional spectral elements must be quadrangles, as shown in Fig 3.3a, with the interior angle at each of the four corners, θ , in the range $0 < \theta < 180^\circ$ (strict inequality). This is to ensure unique elemental transformations (non-zero Jacobians). Elements that do not satisfy this condition, and are therefore not permitted, are shown in Fig. 3.3b.

(E2) Three-dimensional elements are defined by two quadrangles on different (parallel) planes which make up the "top" and "bottom" faces of the element. These quadrangles are subject to the rules of the previous paragraph. The four "side" faces of the spectral element are defined by the four surfaces which connect each of the four respective sides of the two quadrangles. Points on the "side" faces are interpolated linearly from corresponding points on "top" and "bottom" edges, see Fig. 3.2.

(E3) The number of Gauss-Lobatto collocation points in the two or three directions within each spectral element (see Fig. 3.1) must be the same, and will be denoted by N . We remark that, whereas (E1) is a *fundamental* restriction, (E3) is an *artificial* constraint of NEKTON (conforming elements).

As regards how elements may be placed together, the fundamental rule rule

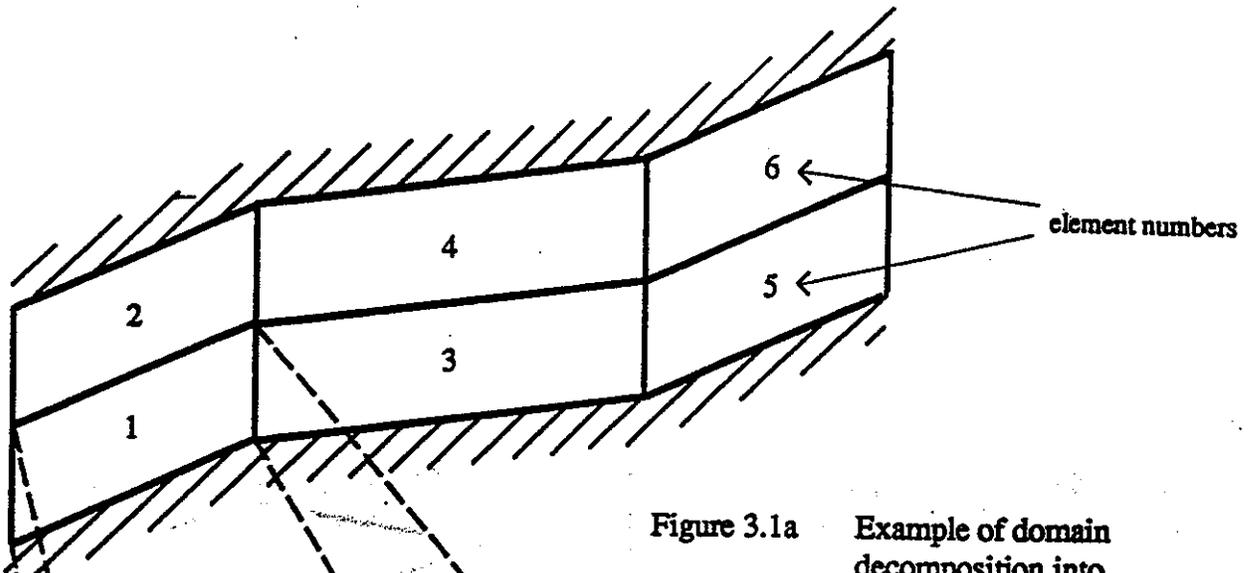


Figure 3.1a Example of domain decomposition into quadrilateral spectral elements

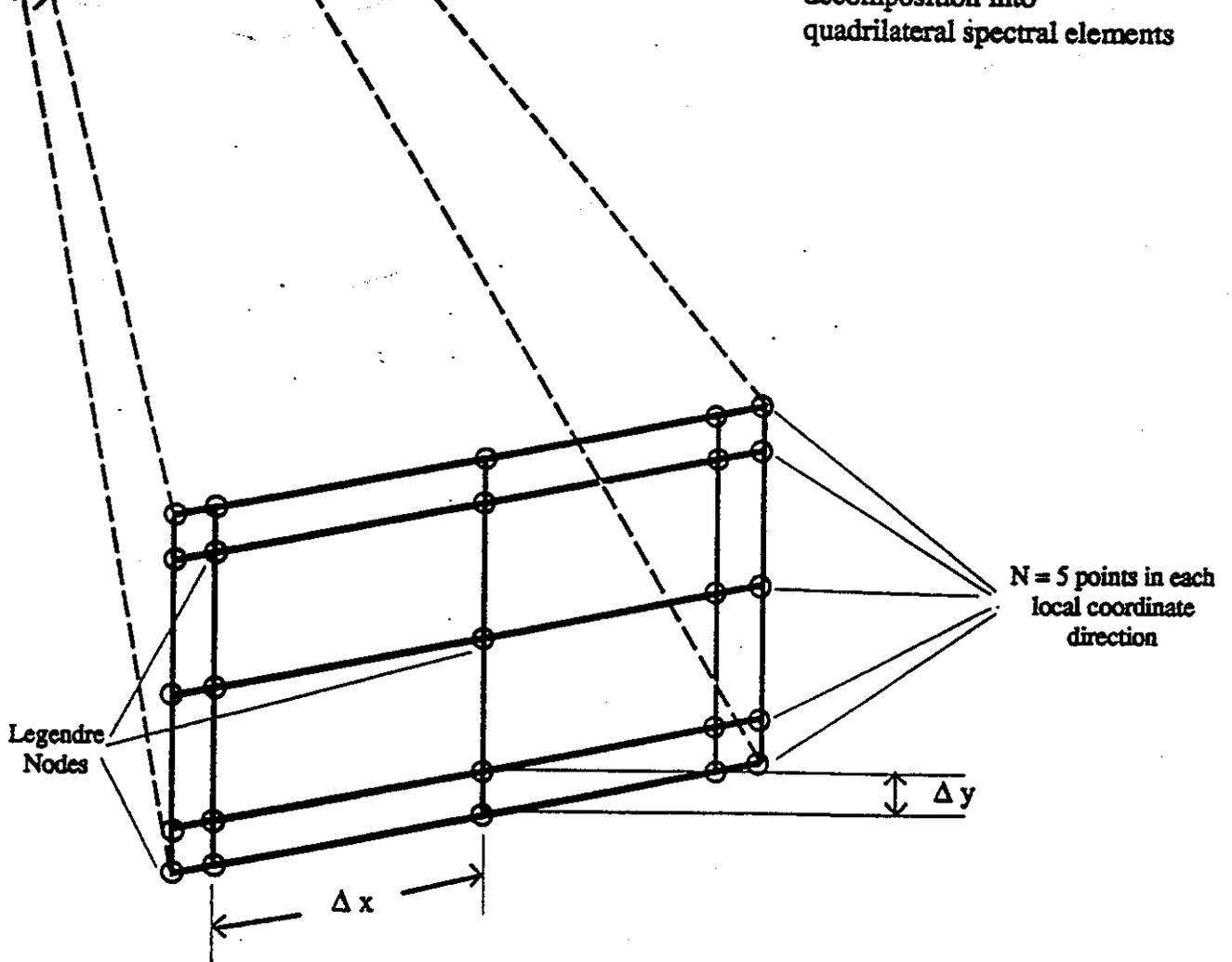


Figure 3.1b Legendre mesh local to a spectral element

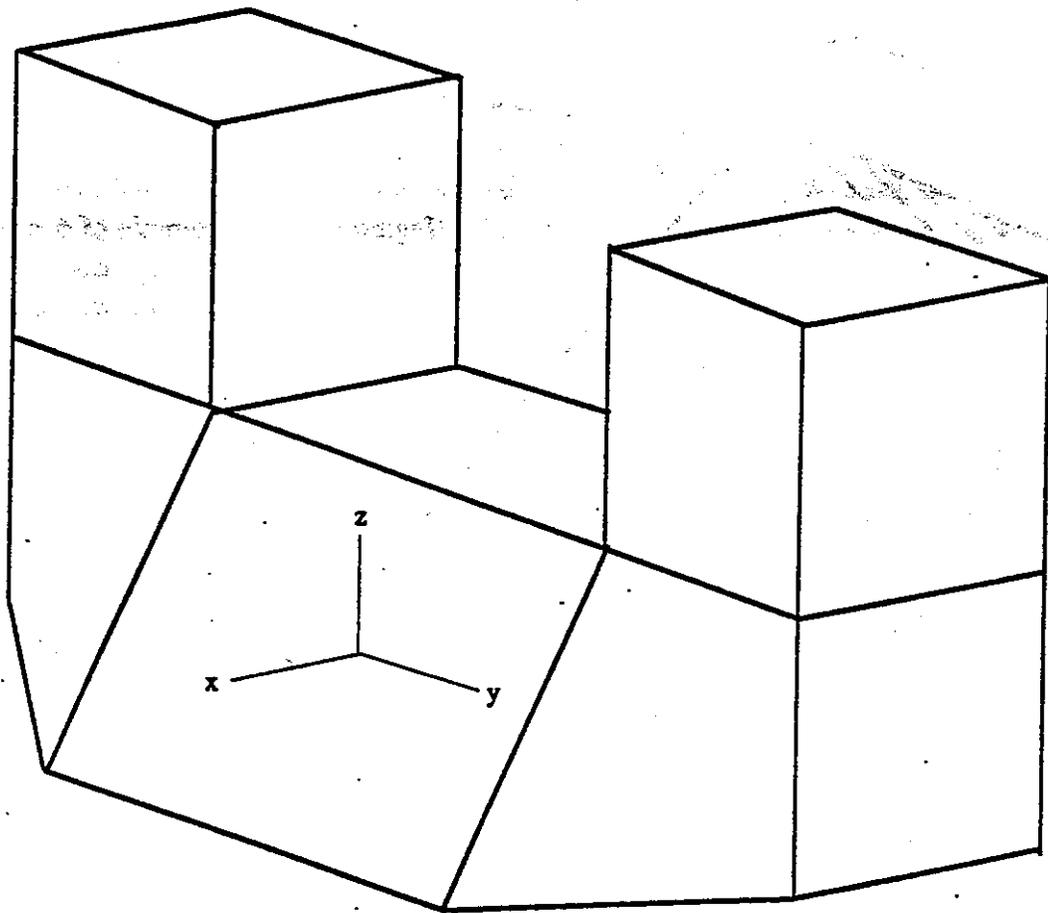


Figure 3.2 Example of domain decomposition into three-dimensional hexahedral spectral elements

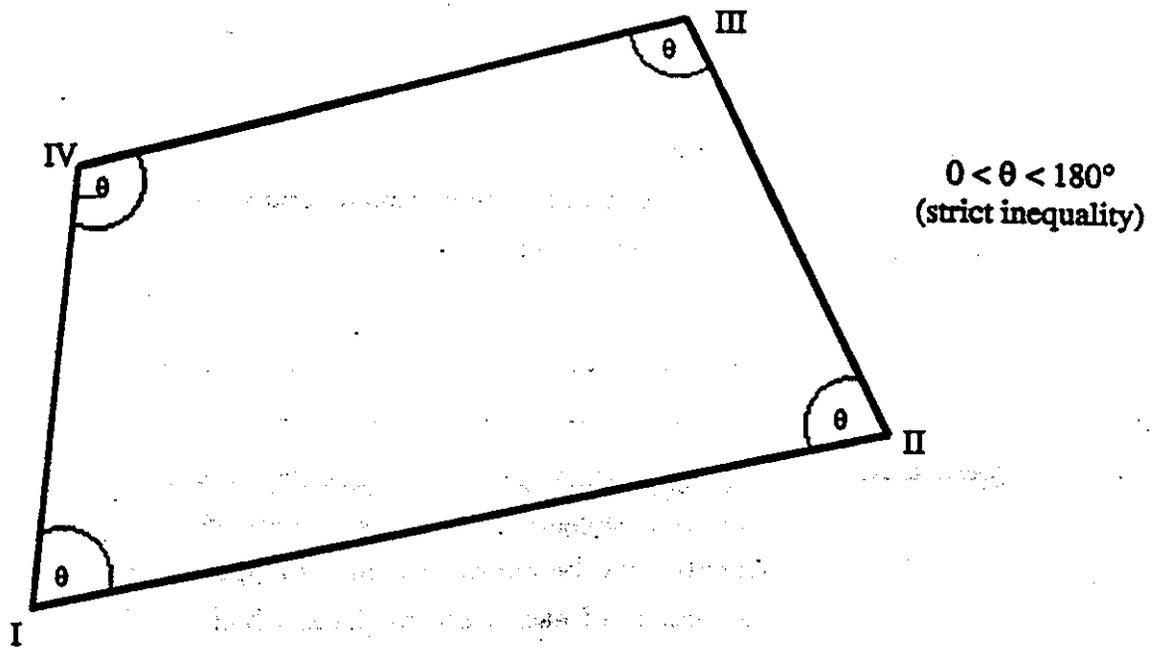


Figure 3.3a Illustration of an admissible element

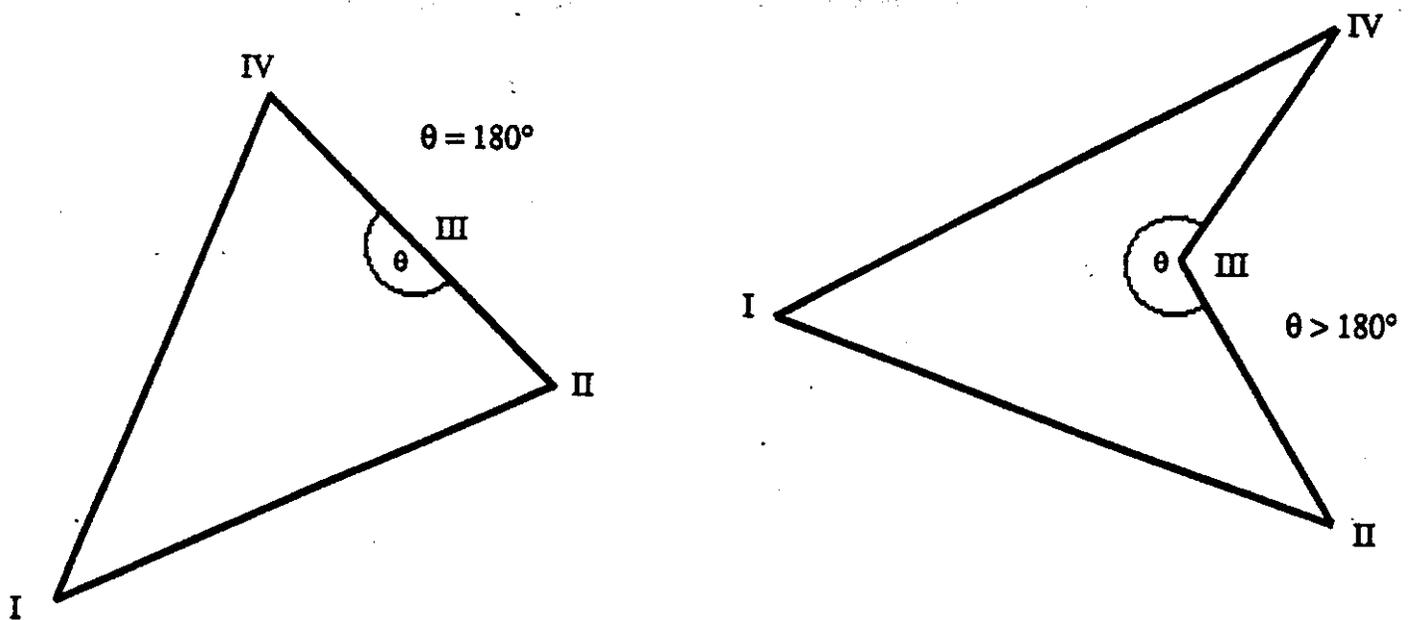


Figure 3.3b Illustration of inadmissible elements

dictated by the form of the governing equations is that the solution and geometry must be continuous across elemental boundaries. Given the restrictions on individual elements, (E1)-(E3), this continuity requirement translates into the following rules of connection for assemblies of elements:

(A1) No part of one element may lie in the interior of another element. Fig. 3.4a gives an example of correct and incorrect element placement.

(A2) Two elements may be connected only by spatial coincidence of two neighboring vertices of each element (from which it follows that they each share an entire common side or face). Examples of correct and incorrect assembly are given in Fig. 3.4b.

(A3) To discretize a domain, D , elements must cover the entire domain. That is, the only element sides which are not connected to other element sides (per (A2)) must lie on the domain boundary, ∂D .

(A4) The number of Gauss-Lobatto collocation points, N , in each spatial direction within each element is the same for all elements. This follows from the functional and geometrical conformity (continuity) requirement and the restriction (E3). It should be noted that (E3) and (A4) are automatically taken care of by NEKTON.

Function Interpolation. Similar to standard finite element techniques, within each element the dependent variables (v, p, T), are expanded in terms of $(N-1)^{\text{th}}$ order tensor-product (polynomial) Lagrangian interpolants through the Gauss-Lobatto Legendre collocation points. For instance, the temperature is expanded as $T = \sum_i T_i h_i$, where the h_i are Lagrangian interpolants (that is, polynomials which are unity at the i^{th} Gauss-Lobatto node and zero at all others), and the T_i are nodal values of the temperature.

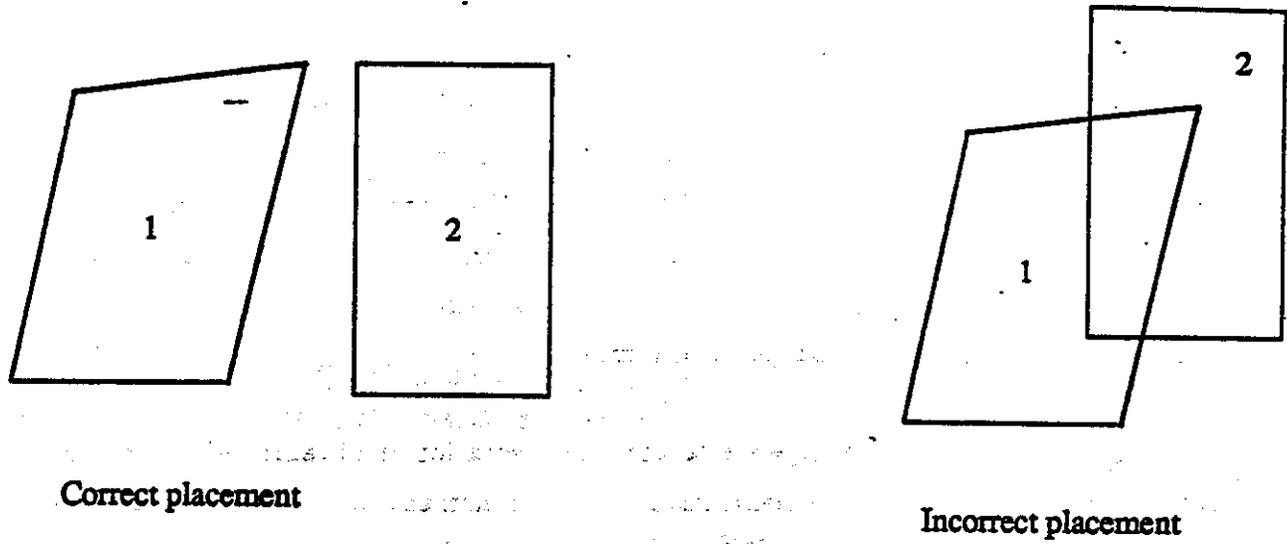


Figure 3.4a Illustration of incorrect relative placement of spectral elements

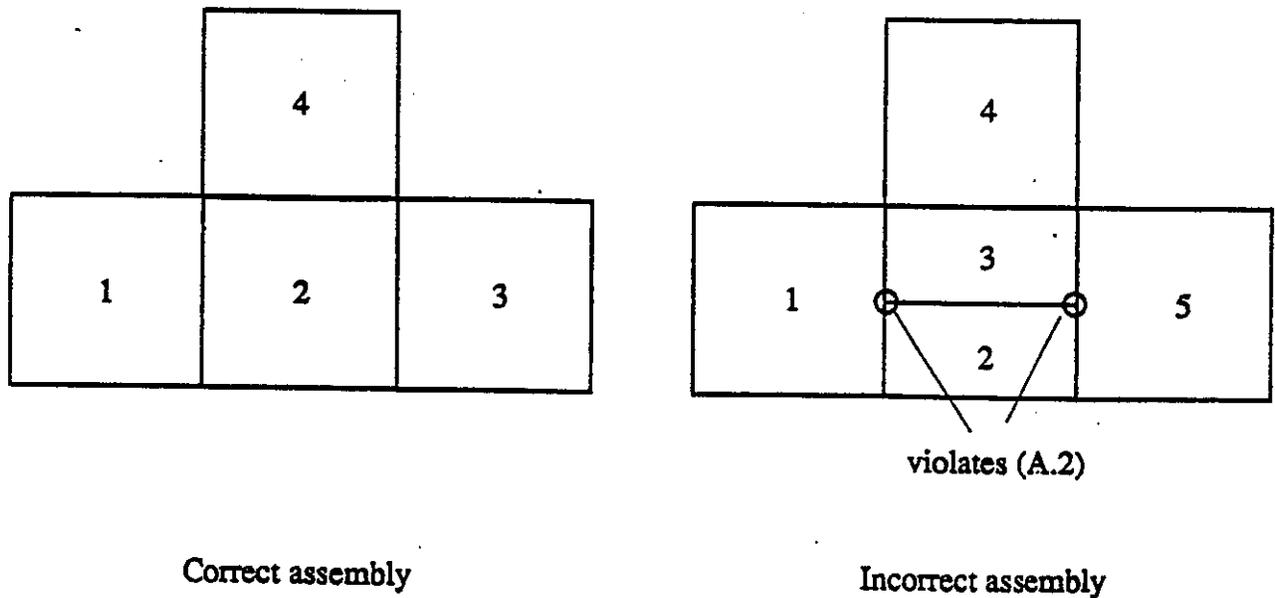


Figure 3.4b Illustration of incorrect assembly of spectral elements

Generation of Discrete Equations. The semi-discrete equations (discrete in space) are generated using weighted-residual (variational, Galerkin) techniques, corresponding to inserting the assumed forms for the dependent variables (e.g., $T = \sum_i T_i/h_i$) into the governing equations, and requiring that the resulting residual vanish in some integral, weighted sense. Note that, since the basis functions chosen are Lagrangian *interpolants*, the numerical variables that are solved for (e.g., the T_i), correspond to the values of the function of interest (e.g., the temperature) at the collocation points of the mesh.

Convergence. Convergence to the exact solution is obtained either by increasing the number of macro-elements, K , ("finite-element"), in which case the error decreases algebraically like K^{-N} , or by increasing the order of the interpolants, N , in elements of fixed identity ("spectral"), in which case the error decreases exponentially for smooth solutions, $e^{-\alpha N}$. As always, any solution obtained numerically should be verified to be converged (mesh-independent) by repeating the calculation on a finer mesh.

3.2 Temporal Discretization

With the exception of steady state conduction and steady Stokes flow, all simulations are *time-dependent*, that is, the governing equations are solved in the time-dependent evolution form given in (2.1)-(2.4). If a steady state solution exists to the problem, it will be achieved by NEKTON as the time-asymptotic result of a transient calculation. The solution is updated to the new time-level using various combinations of multistep and multistage schemes.

In all the transient simulations, the convective terms $(\mathbf{u} \cdot \nabla)\mathbf{u}$, $(\mathbf{u} \cdot \nabla)T$, $(\mathbf{u} \cdot \nabla)\phi$; in (2.1), (2.3) and (2.4) are treated explicitly, that is, only data from previous time steps are used. The standard time-stepping procedure in NEKTON uses a third-order Adams-Bashforth multistep scheme, and the *temporal accuracy* associated with these terms are therefore $O(\Delta t^3)$, where Δt is the time step. The reason for the explicit treatment of the convective terms is to avoid the solution of a nonlinear, nonsymmetric system of algebraic equations at each time level. However, due to this explicit treatment, a restriction on the time step, Δt , must be enforced

in order to obtain *stability*.

The stability restriction (Courant condition) can be written in the form

$$\Delta t < C \min_{D_p} \left\{ \frac{\Delta x}{|u_x|}, \frac{\Delta y}{|u_y|}, \frac{\Delta z}{|u_z|} \right\}. \quad (3.1)$$

where C is the Courant number, $\Delta x, \Delta y, \Delta z$ are the x, y, z distances between the spatial collocation points (see Fig. 3.1b), u_x, u_y, u_z are the x, y, z velocities, respectively, and \min_{D_p} refers to the minimum over the entire flow field. NEKTON will always ensure that the Courant condition (3.1) is satisfied; the Courant number is usually a default value set by NEKTON, however, the user can easily modify this.

The (prescribed) volumetric forcing functions, f , $-\rho\beta\hat{g}T$, q_{vol} , $(q_{vol})_\phi$, in (2.1), (2.3) and (2.4) are also treated using a third-order explicit Adams-Bashforth time-stepping procedure, however, the stability restriction (3.1) does not apply to these terms.

The diffusion terms, $\nabla \cdot [\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)]$, $\nabla \cdot k \nabla T$, $\nabla \cdot k_\phi \nabla \phi_i$, in (2.1), (2.3), (2.4), and the pressure/incompressibility condition in (2.1)-(2.2), are always treated implicitly using a Q^{th} -order Backward Differentiation multistep scheme. In NEKTON the user can specify the order to be $Q = 1, 2, 3$, resulting in a temporal accuracy $O(\Delta t^Q)$. Due to the implicit treatment, there is *no* stability restriction associated with these terms, however, a set of linear algebraic equations must be solved at each time level. This can be done efficiently using iterative solvers as described in Section 3.3.

Remark 1. In general, the stability determining convective terms (explicit), and the work-intensive pressure/diffusion terms (implicit) do not coincide in a traditional semi-implicit treatment of the unsteady Navier-Stokes equations. In order to address this issue, a new high-order operator splitting method has been implemented in the latest version of NEKTON; this method can potentially yield significant speed-up, see Section 3.4. Although this scheme can be interpreted as the method of characteristics, it is in fact a particular example of a scheme generated from a more general methodology based on operator integration factor methods.

Remark 2. NEKTON also provides the option of integrating the unsteady Navier-Stokes equations using the classical splitting method (fractional step

method). This scheme cannot be applied to stress boundary conditions, and the temporal accuracy is $O(\Delta t)$.

Remark 3. For problems with time-dependent domains, the geometry is updated explicitly, using a P^{th} -order Adams-Bashforth multistep scheme, $P = 1, 2, 3$. For maximum stability the order for P depends on the type of moving boundary problem. For example, when surface tension effects are included as in free-surface or multi-fluid problems, an additional stability criterion (to the CFL condition) is required:

- The stability restriction used in the NEKTON for surface tension effects is:

$$\Delta t < \text{MIN}_{\partial D_{st}} \left[\frac{\rho}{\sigma} \left(\frac{\Delta s}{\pi} \right)^3 \right]^{1/2}, \quad (3.2)$$

where ρ is the fluid density (minimum density in a multi-fluid system), Δs is the curvilinear distance between the spatial collocation points on the surface tension boundary ∂D_{st} , σ is the surface tension coefficient, and $\text{MIN}_{\partial D_{st}}$ refers to the minimum over the surface tension boundary.

In this case, the order P should be the same as the order Q for the Backward Differentiation scheme for maximum stability. For other type of moving boundary problems, the theoretical basis is deficient. On an empirical basis, it is recommended that $P = 3$ be used for problems in which the motion of the boundary is prescribed (i.e., moving walls), and $P = 1$ be used for heat transfer analysis with change-of-phase (i.e., Stefan problem).

3.3 Solution Methods

The solution of the fully discrete set of equations are characterized by the use of iterative solvers and tensor-product sum-factorization techniques. For the *explicitly* treated convective terms, there is no matrix inversion, but only matrix-vector product evaluations. This is done efficiently, both in terms of operation count and memory requirement, by using sum-factorization techniques on an element-by-element basis.

The *implicitly* treated diffusion and pressure terms result in a large linear system of equations, which are solved using iterative solution procedures. The basic

solver in NEKTON is an elliptic Helmholtz solver, appropriate for inverting the diffusion operator. This solver is based on conjugate gradient iteration, in which the computationally intensive part in each iteration is dominated by a single matrix-vector product evaluation. As for the explicit terms, this evaluation is performed using sum-factorization techniques on an element-by-element basis. Note that the global or elemental matrices are never formed explicitly, an essential feature for solving large (three-dimensional) problems. The storage requirements in NEKTON is minimal; it is on the order of the number of collocation points in the mesh.

Remark 4. The classical splitting scheme for solving the unsteady Navier-Stokes equations only involves one elliptic solve for the pressure and one for each of the velocity components. In contrast, a discretization based on consistent approximation spaces (polynomial degree N for velocity and polynomial degree $N - 2$ for the pressure) results in nested elliptic solves for the pressure (Uzawa algorithm). While the latter approach is slower, it is generally more accurate; in NEKTON the user has a choice of which of the two schemes to use, see Section 3.4.

Remark 5. NEKTON has the capability of using two different elliptic solvers. The first (default) is based on conjugate gradient iteration, while the second one is based on an intra-element multigrid procedure. The multigrid technique is particularly attractive for large problems, as the associated convergence rate is almost resolution independent. The earlier multigrid solver is currently being upgraded in order to remove some of the former limitations, and this version of NEKTON relies solely on the conjugate gradient based solver.

The data structure used in NEKTON also allows the code to run efficiently on modern architectures, in particular on distributed memory parallel machines. The loosely coupled spectral elements can readily take advantage of *medium-grained parallelism* by distributing the elements on different processors, while each individual (tightly coupled) elemental matrix-vector product evaluation can be *vectorized*. This is in contrast to direct solution techniques such as Gaussian elimination, which are cross-coupled and less effective at exploiting parallel computer architecture.

The motivation behind spectral element (rather than low-order finite ele-

ment) methods is that the increased coupling introduced by high-order methods increases the work only *algebraically*, whereas the increase in accuracy is *exponential*. These arguments only apply for problems with smooth solutions. However, with the exception of boundary-induced singularities, incompressible flow and heat transfer problems will generally satisfy this requirement.

3.4 Method Tradeoffs

NEKTON uses several different algorithms for solving problems in fluid mechanics. The selection of the algorithm most appropriate for a given problem can greatly affect the speed and accuracy with which that problem can be solved. In this section we outline and compare the capabilities, relative speeds, and accuracies associated with each solution algorithm used by NEKTON, and the discussion complements Sections 3.1-3.3.

The solution algorithm has two parts, the problem formulation, in which the partial differential equations comprising the Navier-Stokes equations are converted into a set of discrete equations, and the solution, a numerical procedure in which the set of discrete equations are solved simultaneously. NEKTON uses three formulations, the non-stress (default) formulation, the stress formulation, and the classical splitting scheme. These formulations convert the Navier-Stokes equations into a series of discrete elliptic problems, which are solved iteratively using an elliptic solver; in NEKTON this solver is based on conjugate gradient iteration.

The split formulation gives the fastest solution for a any given time-dependent flow problem. Its primary limitation is that it incurs splitting errors which are of order $\nu\Delta t$. Splitting errors are significant where the viscosity is high and the time step is large, as in creeping or Stokes flows. For high Reynolds number problems, the viscosity is low and the time step is small due to stability restrictions (Courant condition), so that generally the splitting errors are small. The split formulation is recommended for flows in which $Re > 25$.

The non-stress formulation uses consistent approximation spaces for the velocity and pressure, and gives accurate results for the entire range of Reynolds numbers. It can be used for flows ranging from Stokes flows (where $Re=0$) through transition flows. It is slower than the split formulation by a factor of approximately

four.

The stress formulation is a variant of the non-stress formulation in which the three (or two) velocity components are solved for simultaneously, and all components of the stress tensor are retained. This formulation allows for additional stress-related boundary conditions: imposed stress, zero stress, and moving boundaries, which can be defined in terms of local (tangential and normal) coordinate components. This formulation is the most general and rigorous of the three, but also the most cpu-intensive. It is slower than the non-stress algorithm by a factor of approximately two.

As discussed in Section 3.2, NEKTON solves any convection-diffusion problem as a transient problem, with the convective terms treated explicitly. For convection dominated flows ($Re \gg 1$) this explicit treatment can give very severe time step restrictions due to the Courant condition (3.1). For the standard semi-implicit time-stepping strategy, NEKTON uses a default Courant number of $C = 0.25$. However, if the flow problem has prescribed velocity on the whole (fluid) domain boundary, the user can save some time by increasing the Courant number to about $C = 0.4$.

Bigger savings can be achieved for convection-diffusion problems if the method of characteristics is used, in which the convective terms are decoupled from the pressure and viscous treatment. Hence, the time step for the (expensive) integration of the viscous term, together with the imposition of the incompressibility constraint, can be set much larger than the (restrictive) time step for the integration of the explicitly treated convective terms. If the work is dominated by the pressure solver, an increase in the Courant number (measured using the bigger time step) from $C = 0.25$ to $C = 2.0$ can sometimes give a factor 8 in speed-up. The limitations using the method of characteristics are typically related to problems with steep (marginally resolved) normal boundary layers, and if the prescribed inflow boundary conditions do not correspond to fully developed flows.

For a summary of the restrictions in NEKTON as regards numerical methods, we refer to Appendix A.

Chapter 4

Introductory NEKTON Usage

This chapter deals with the practical aspects of running simulations with NEKTON. It first covers the hardware and software issues specific to individual sites. Next, two sample problems are solved. The chapter then covers the relative advantages of the different numerical algorithms available in NEKTON and when each is most appropriate.

Experience has shown that the optimal learning strategy for NEKTON usage is to delay a detailed description of PRENEK and POSTNEK until the next chapter. We will first run sample sessions and discuss strategies of using the NEKTON package. It is recommended that after this initial exposure to NEKTON, the user then read the next chapters which describe PRENEK and POSTNEK in detail.

4.1 Site-Specific Considerations

The user should read the appropriate sections from the following list to familiarize himself with his local computing environment before beginning the sample problems.

Operating Systems:

- UNIX
- VMS

Hardware:

- X-Windows Workstations
- Sun Workstations
- Tektronix Terminals
- Color VAX Workstations
- Cray
- Parallel Processors
- Printers
- Networks

4.1.1 Operating Systems

The UNIX Operating System

When running NEKTON in a UNIX environment, the user should be aware of several issues. First, UNIX is case-sensitive. All file names, session names, and commands (such as `pre`, `nek`, and `post`) must be typed using the correct case. Lower case is used for the NEKTON commands and, for simplicity, is recommended for all user-defined session names. Secondly, the UNIX file system supports only a single version of a given file name. Thus, if a new file is opened which has the same name as an existing file, the existing file will be deleted. For this reason, the user should be careful not to use the same name for two distinct sessions. Batch jobs are submitted in UNIX by putting an ampersand (&) at the end of an interactive command, or by typing `{nek session}` rather than `{nek session}`. This procedure is sometimes used to free a terminal or window for other uses while a long NEKTON job is running. Finally, unnecessary compilations can be avoided typing `nek session run` rather than `nek session`.

The VMS Operating System

When running NEKTON in a VMS environment, the user should be aware of several issues. First, VMS is not case-sensitive. All file names, session names, and commands (such as *pre*, *nek*, and *post*) can be either upper or lower case. Thus, session *TEST* will be the same as *test* or *Test*. Secondly, the VMS file system supports many versions of a given file name. To each filename is appended a version number. Thus, if a new file is opened which has the same name as an existing file, it will be given a higher version number and the old file will be buried, and will remain on the disk until a *PURGE* or *DELETE* command is used. Finally, the command files in the VMS operating system support several options. A batch job can be submitted by typing *nekb session* rather than *nek session*. Several switches can be appended to the "nek" command to avoid unnecessary compiles or alter the way the executable file is assembled. The command *nek session run* will override the default compilation process and run an existing executable. The command *nek session compile* will force the entire code to be recompiled before running the session.

4.1.2 Hardware

X-Windows Workstations

The X Windows standard developed at MIT is rapidly becoming the de facto graphical standard for new workstations and new software interfaces. X Windows has two principal advantages over other standards currently available. The first is that it is popular and available on essentially all new workstations. The second advantage is that it provides a means by which graphical information can be transmitted across the network. This allows images generated on one workstation to be seen on a window on another workstation (which can be different brand of workstation). It also allows for a single cpu to support several "dumb" X Windows terminals. The graphical interface of NEKTON, *prenek* and *postnek*, can be invoked on an X Windows workstation by typing "pre" or "post" from a standard text window. The interface will open a graphics window, and all I/O (both graphical and text) will be channeled through the graphics window. Thus, the user should keep

the cursor within the border of the graphics window, so that the keyboard remains attached to the NEKTON graphics window. With DecWindows, a initial click of the mouse button within the window is required to attach the keyboard. There are 3 mouse buttons on an X Windows workstation. The leftmost button corresponds to the yellow button referred to in the manual, the rightmost corresponds to the blue button, and the center corresponds to the white button. Color-coded tape can be installed on the buttons as a mnemonic aid.

Sun Workstations (Without X Windows)

The Sun workstation uses a UNIX operating system. On top of this operating system resides the Suntools windowing environment. This windowing system enables many windows, or virtual terminals, to run simultaneously. This can be quite convenient; in one window a user might edit files, in another he might send and receive mail, and in another, he might be running NEKTON. Before running NEKTON the user can type "suntools" to open this windowing environment in a generic fashion or "nektools" to open this windowing environment with windows tailored to using NEKTON.

The graphical user interface of NEKTON (PPRENEK and POSTNEK) uses a special type of window called a graphics tool. The commands pre and post should be typed only from within a graphics tool. A graphics tool is split into two subwindows, a text subwindow and a graphics subwindow. All graphical communication (color output or mouse input) is done from inside the graphics subwindow. All text communication (fortran read and writes) is done from inside the text subwindow. When typing in text in response to a prompt from PPRENEK or POSTNEK the cursor MUST BE IN THE TEXT SUBWINDOW, otherwise the text will not be received by the program. When a numerical response is expected from the keypad in the graphics subwindow, the border of the keypad will be highlighted as a prompt.

There are 3 mouse buttons on the Sun workstation. The leftmost button corresponds to the yellow button referred to in the manual, the rightmost corresponds to the blue button, and the center corresponds to the white button. Color-coded tape, supplied with the NEKTON manual, should be installed on the mouse buttons as a mnemonic aid.

Tektronix Terminals

NEKTON supports the Tektronix 4107, 4109, 4207, and 4209 terminals. All four have the same command set, and differ only in size and in that the 4200 series supports a mouse as well as a graphics tablet. A number of emulation programs for color Macintoshes and color IBM-compatible PCs exist with varying degrees of compatibility. Some of these packages have been successfully used in conjunction with NEKTON, however, their use is not supported by Nektonics.

When the terminals are setup according to the NEKTON installation sheet, they default to a VT100 edit mode. This enables use of a screen oriented editor such as EDT or EMACS. PRENEK or POSTNEK put the terminals into the tek graphics mode at the beginning and return the terminals to edit mode upon exit. If the user hits ^C in the middle of a PRENEK or POSTNEK session, the terminal will remain in tek mode. To return it to its default VT100 mode (and to clear the screen of any stray graphics), the user should hit the *SETUP* key from the top row of keys, then should type the text *RESET* followed by a carriage return. The terminal will spend the next 30 seconds clearing all memory and returning to the VT100 mode.

These terminals have the capability of true hardware pan and zoom. Any of the images displayed in PRENEK or POSTNEK can readily be blown up for more precise mouse inputs or more detailed examination of the output. Pan and zoom are invoked by first pushing the *MENU* key from the top row of keys, then by pushing the *F2* key. The size and location of the new display region are defined by the joydisk at the top left corner of the keyboard. The newly defined region is displayed by pushing the *F3* (view) key. The joydisk can be used to scroll to recover text which has been scrolled above the 8-line text display on the bottom of the screen.

The recommended input device for the Tektronix terminals is the 4957 graphics tablet. Connected to this tablet is a four-button color-coded mouse. The alternate three-button mouse available on the 4200 series should have color-coded yellow tape installed on the leftmost button and blue tape on the rightmost button. Use of this three-button mouse requires the file *DEFAULTS.NEK* in NEKTON (or the user's home directory) to have the following line: *mouse tek3button*, see also the

Installation Instructions..

Color VAX Workstations

The use of NEKTON on a VAX/GPX color workstation requires the presence of VAX GKS software package. The GKS software is displayed through a separate graphics window. The commands used to invoke PRENEK and POSTNEK on GKS are PREGKS and POSTGKS rather than PRE and POST. Upon issuing one of these commands from a VT220 Terminal window, the window is resized to a tall, thin geometry. After this resizing, the user should move this window to the right edge of the workstation screen so that the graphics window can occupy the remainder of the screen.

The GPX workstation has a three-button mouse. The leftmost mouse button corresponds to the yellow button, and the rightmost corresponds to the blue button. To make an entry corresponding to the yellow button, only the yellow button is pushed. However, use of the blue button actually requires a two-step procedure. To make an entry, first the rightmost (blue) button is pushed, then the leftmost (yellow) button is pushed.

Cray

Nekton runs on machines supporting UNIX, which includes the Cray series of machines. Cray machines run UNICOS, which is the Cray-supported version of UNIX. The commands to run nekton on UNIX machines are *nek session* for an interactive run, and *nekb session* for a batch run.

Cray also continues to support the older COS operating system; the procedures for running NEKTON on a Cray under COS are different than from all other platforms. Generally there is "front-end" computer which handles interface to COS. The procedures for running NEKTON depend on the front end hardware and software. These procedures are established during the installation of NEKTON. Consult your Cray system manager for the detailed NEKTON commands.

Parallel Processors

At every stage of the development process, NEKTON has been designed to function on the high-performance distributed memory parallel processors which are currently available. NEKTON achieves very high performance on these architectures because the heterogeneous nature of the spectral element discretization (tight coupling within each high-order element, weak coupling between elements); this type of spatial discretization naturally exploits data locality. Extensive tests and numerical simulations have established that this is an economical approach to achieving supercomputer performance. The current version of NEKTON has been implemented on the Intel iPSC/2-VX hypercubes as well as the four processor CRAY-2/256.

Plotters

NEKTON supports the PostScript standard and the QMS Quik command standard. The plotting device is chosen by editing the file \$HOME/.nekdefaults in UNIX or defaults.nek in VMS. A sample of this file is in the NEKTON directory. This file should be copied to the user's working directory in UNIX systems. For further details we refer to the Installation Instructions.

When POSTNEK is being used to generate hard copies, it is best to hit the CLEAR to clear the screen before making surface plots (arrow plots, fishnet plots, or streamlines). An alternate possibility is to use a hardware compatible hard copy unit which will accept a screen dump.

Networks

Often the optimal usage of NEKTON involves running PRENEK and POSTNEK on a color workstation used as a front-end and running NEKTON on a larger mainframe or parallel processor. This installation requires the existence of a network for file transfer between the workstation and the mainframe and a command procedure to properly handle this file transfer. The details of this command procedure will depend on the operating systems of the front end and mainframe and on the network connecting them. To implement such a procedure it is recommended

to contact your Nektonics representative for details concerning the command procedures and the licensing arrangements.

4.2 Sample Sessions

To demonstrate the operation of the NEKTON 2.5 simulation package, two sample sessions will now be presented. Emphasis here is not on exploring all possible options of the simulation package, but rather on describing the framework of the package and showing how the various components fit together to solve two particular problems. A more detailed description of the options available at each stage can be found in the menu descriptions in Sections 5,7 and 8.

It is assumed that the user is somewhat familiar with the computer hosting NEKTON. Although most of the user's work is within the confines of PRENEK, NEKTON, and POSTNEK, there is occasion where file editing, knowledge of simple Fortran, and familiarity with file structure and operating system commands is helpful.

4.2.1 Notation

In the sample sessions the following notation is used: Entries enclosed within square brackets refer to menu items to be selected by the user. For example, [TYPE IN NEW PARAMETERS] means that the user is to move the mouse so that the screen cross hairs are within the menu box containing that phrase and then push the yellow mouse button (the leftmost button on an X Windows workstation). Entries enclosed within curly brackets refer to text strings that are to be typed in by the user followed by a carriage return. For example, {problem1} means that the user should type in the the eight characters in the string "problem1" and then type a carriage return. As another example, {} means that the user should simply hit a carriage return.

Entries of the form (M/LETTER number.letter) refer to mouse inputs of points in the mesh drawing area. Here LETTER refers to which mouse button is to be pushed, B,Y,W, or G (for the colors on the buttons). Number.letter refers to a location marked on a figure, which indicates the location where the cross hairs

should be when the mouse button is pushed; *number* refers to the figure number, and *letter* to the particular reference on that figure.

Entries of the form (KP/*character*) refer to mouse inputs of points in the keypad area. Here *character* refers to which keypad key the cursor should be on when the mouse button is to be pushed. The "R" key terminates the entry (as a return key). The "E" key is used for scientific notation (e.g., 1.0E3). The "D" key deletes a keystroke entered by mistake.

Entries in boldface and underlined correspond to prompts and messages from the computer that appear on the screen. For example, Choose a Name for This Session: corresponds to the first prompt generated by PRENEK. Non-essential computer-generated text that appears on the screen will sometimes be omitted from this documentation for the purpose of compactness and readability.

Entries in boldface and enclosed within square brackets refer to the names of menus that the computer displays on the screen. The menu itself is composed of a list of items, each of which is inside a blue rectangular box. The name of the menu appear at the top left corner of the screen whenever the menu is displayed. For example, [READ PARAMETER] corresponds to the READ PARAMETER menu, which is the first menu displayed by PRENEK. The location of the menu within the menu structure scheme can be found in Fig. 5.1. Full descriptions of the menus with their contents are arranged alphabetically in Section 8.

Any prose that does not fall into one of the above five categories can be assumed to be comments; these will not appear on the terminal screen, nor should the user type them on the keyboard. If such a comment follows immediately after a user input, an exclamation point will separate the input from the description to further emphasize that the text is a comment.

4.2.2 Using the Mouse Buttons

The use of the YELLOW, WHITE, BLUE, and GREEN mouse buttons in both PRENEK and POSTNEK is summarized below. Note that on an X Windows terminal, the leftmost button correspond to the yellow button, the rightmost to the blue button, and the middle one to the white button.

PRENEK

YELLOW - BUILD menu: specifies element corner position to be current location of cross hairs; used to create new corners.

General purpose: used for all graphics tablet inputs except those described below

WHITE - BUILD menu: returns current cross hairs location in Cartesian coordinates.

BLUE - BUILD menu: specifies element corner position to be that of closest previously-defined corner; used to connect elements.

- BOUNDARY CONDITION menu: specifies a side whose boundary condition is to be copied to a new side.

GREEN - Not used.

POSTNEK

YELLOW - Used for all graphics tablet inputs.

WHITE - Not used.

BLUE - Latches line endpoints (e.g., profile) to nearest grid intersection.

GREEN - Not used.

4.2.3 Problem 1: Steady Two-Dimensional Heat transfer

Problem Statement

The purpose of this sample session is to familiarize the user with the basics of the interactive user interface, the PRENEK and POSTNEK modules. A simple physical problem is chosen with a simple geometry. In particular, the problem of steady heat transfer in a square box is considered, as shown in Fig. 4.1. We will use equation (2.3) with the convective velocity terms and the time dependent terms set to zero. The length and width of the box is $L = 1.2$ and it is centered at the origin. The thermal conductivity described in Section 2.2 is set to $k = 2$. The boundary conditions are that the temperature at the left and right faces are $T = 0.0$ and 1.0 ,

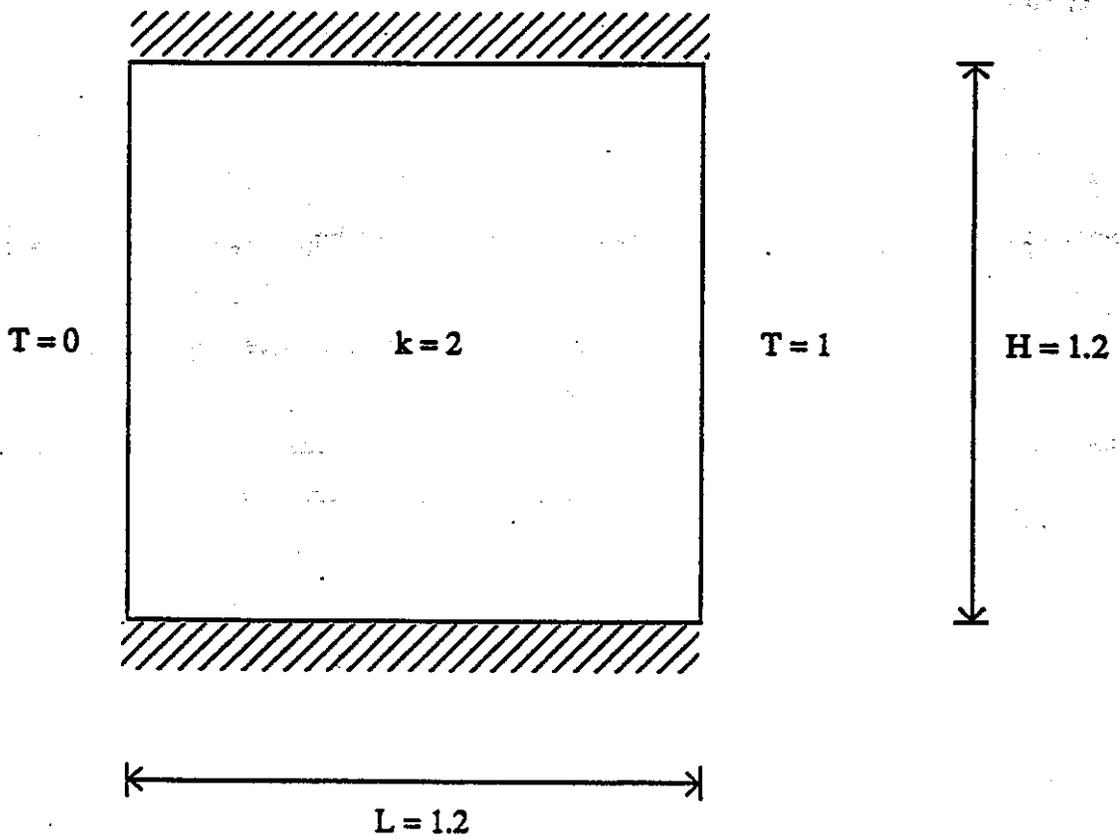


Figure 4.1 Steady two-dimensional conduction heat transfer in a square box. The top and bottom sides are insulated. The thermal conductivity $k = 2$.

respectively. The top and bottom boundaries are insulated.

PRENEK

The session begins with the user typing (from within a graphics tool or graphics terminal) "pre" to the operating system prompt;

```
$ {pre}
```

This puts the user inside the PRENEK module, which is confirmed by the programs inquiry for a SESSIONNAME label.

Choose a Name for This Session: {problem1} ! The user chooses the name "problem1" for this session. This label will be used for all subsequent stages of the modeling process (preprocessing, solution, and postprocessing). This label also appears on the screen during preprocessing and postprocessing. For future book-keeping purposes, all files related to this session will contain the prefix "problem1". The case of these user inputs is important with UNIX and irrelevant with VMS.

PRENEK now clears the screen, draws the keypad and presents the first menu, the READ PARAMETER menu as in Fig. 4.2. This figure, and all screen display figures in this manual are screen dumps from a Tektronix terminal onto an Epson printer. The appearance on a Sun workstation will differ from that in these figures principally in that the graphics and text will be on separate subwindows, whereas the Tektronix screen has overlapping text and graphics windows. Note that in Fig. 4.2 the menu name appears at the left top corner of the screen. From this, we can reference a complete description of a given menu from the alphabetical listings in Chapter 8.

[READ PARAMETER]

Here the user is given the option to read parameters and equation type specifications from a previous session, or to type in new parameters. Reading parameters is convenient when the user has specified the desired parameters in a previous session; the user can read in those values and avoid entering them manually. Here, however, we wish to demonstrate how parameters are entered, so the response is

[TYPE IN NEW PARAMETERS]

[SET EQUATION TYPE] ! Now we set the switches illustrated in Fig. 4.3

READ PARAMETER

NEKTON

TYPE IN NEW PARAMETERS

READ PREVIOUS PARAMETERS

7	8	9	-
4	5	6	0
1	2	3	E
0	.	R	

Choose a Name for This Session:
problem1

Beginning Session problem1

Figure 4.2

which define the type of equation we will solve. The default configuration of the switches appears in this menu and the equation type resulting from this combination of switch settings is described by the text

CURRENT SETTING: UNSTEADY TWO DIMENSIONAL STOKES

The desired equation type for problem1 is steady two-dimensional heat transfer. In order to obtain this we toggle the switches in the following way

[UNSTEADY]

[SET EQUATION TYPE]

CURRENT SETTING: STEADY TWO DIMENSIONAL STOKES

[FLUID FLOW]

[SET EQUATION TYPE]

CURRENT SETTING: STEADY TWO DIMENSIONAL

[HEAT TRANSFER]

[SET EQUATION TYPE]

CURRENT SETTING: STEADY TWO DIMENSIONAL HEAT TRANSFER

We now can accept these switch settings and move to the SET PARAMETER menu

[ACCEPT CURRENT SWITCHES]

[SET PARAMETER]

REQUIRED PARAMETERS

The parameters are split into two screens. The first screen, illustrated in Fig. 4.4, contains the "required" parameters. We are prompted for a value for the parameter CONDUCT (the thermal conductivity) by the flashing border around that parameter. The default value of 1.0000 is displayed next to the parameter. Since we wish the thermal conductivity to be 2.0, we respond by typing in the value {2.0}.

In general, the keyboard is used to input new parameter values. However, on a Sun workstation, the value of 2.0 is entered with the mouse by hitting the leftmost mouse button while the mouse is over the keypad boxes containing "2", ".", "0", "R". The keypad "R" means "return" and the "E" is for exponent, e.g., 1.0E3. As a prompt, the border of the keypad becomes highlighted when the keypad is expecting input from the user. When this prompt does not appear, the value should be typed in via the keyboard.

SET EQUATION TYPE

NEKTON

ACCEPT CURRENT SWITCHES	
DIMENSION	2
UNSTEADY	N
FLUID FLOW	N
HEAT TRANSFER	Y
ADDL PASSIVE SCALARS	0
MULTIGRID SOLUTION	N

7	8	9	-
4	5	6	D
1	2	3	E
0	.		R

using mouse to toggle logical switches on and off.
CURRENT SETTING: UNSTEADY TWO DIMENSIONAL STOKES
CURRENT SETTING: STEADY TWO DIMENSIONAL STOKES
CURRENT SETTING: STEADY TWO DIMENSIONAL
CURRENT SETTING: STEADY TWO DIMENSIONAL HEAT TRANSFER

Figure 4.3

SET PARAMETER

NEKTON

CONDUCT	2.00000
NORDER	5.00000

7	8	9	-
4	5	6	D
1	2	3	E
0	.		R

Type in new value or <cr> to keep current (default) value
REQUIRED PARAMETERS
2.0

Figure 4.4

PRENEK echoes the new value next to the CONDUCT parameter to confirm that the value has changed to 2.0 and we are prompted for the next parameter, $\overline{\text{NORDER}}$ (see Fig. 4.4). We wish to retain the default value of 5 as the number of collocation points (nodes) in each direction in each macro-element. We respond

{

On a Sun, we hit the "R" box on the keypad.

[SET PARAMETERS]

OPTIONAL PARAMETERS

Next we set the optional parameters of Fig. 4.5. The heat input per volume (or area), QVOL, should remain zero. The GRID, which we will use in building the mesh, should remain 0.05. We will retain the default convergence tolerance for the iterative scheme, HELMHOLTZ. Thus, we accept all three defaults by entering the inputs

{

{

{

which completes the SET PARAMETER menu.

Note that we set far fewer parameters than those described in the SET PARAMETER menu in Chapter 8. Note in particular that all items relating to time dependence and fluid flow were not displayed. PRENEK automatically determines which parameters are relevant to the equation type specified, and screens out the irrelevant parameters from the SET PARAMETER menu.

[CENTRAL]

The CENTRAL menu (Fig. 4.6) allows control of PRENEK to branch to several different places. We could, for example, choose to ALTER PARAMETERS (i.e., return to the beginning of the SET EQUATION TYPE menu). This would be done to correct a mistake or to modify parameter data read in from a previous session. Alternately, we could BUILD FROM FILE, i.e., read in geometry data from a previous session. Since we wish to demonstrate how to build a mesh, we choose

[BUILD INTERACTIVELY]

[SCALE FACTOR]

[USE NORMALIZED] ! Specifies that normalized coordinates will be used. Nor-

SET PARAMETER

NEKTON

QUAL	0.000000E+00
GRID	0.500000E-01
HELMHOLTZ	0.100000E-04

7	8	9	-
4	5	6	0
1	2	3	E
0	.	R	

OPTIONAL PARAMETERS

Figure 4.5

CENTRAL

ALTER PARAMETERS
SHOW PARAMETERS
BUILD INTERACTIVELY
BUILD FROM FILE
IMPORT UNIVERSAL FILE
ABORT (SAVING PARAMETERS)

7	8	9	-
4	5	6	D
1	2	3	E
0	.	R	

Figure 4.6

malized coordinates are those in which the origin is at the screen center and the range in x and y is from -1 to 1, Fig. 4.7. Alternately, the mouse could have been used to define a set of scale factors which determine the coordinate mapping from the problem coordinates to the screen dimensions.

[BUILD]

In the build menu, we will first enter the quadrilateral element in Fig. 4.8. *PRENEK* requires that the corners be entered in counter-clockwise order.

[ADD ELEMENT]

Corner I >

(M/Y 4.8.a)

Corner II >

(M/Y 4.8.b)

Corner III >

(M/Y 4.8.c)

Corner IV >

(M/Y 4.8.d)

A gray rectangular macro-element should appear with the number "1" in the center similar to the element in Fig. 4.8. If the element does not look like that in Fig. 4.8, the user should use the [DELETE ELEMENT] menu item, and then go to the [ADD ELEMENT] point in the procedure immediately above.

After successful input of the element, we exit the BUILD menu and proceed to the variable property menu.

[END ELEMENTS]

[VARIABLE PROPERTY] ! See Fig. 4.9.

[ACCEPT MATL,QVOL] ! We will accept the (constant) material properties specified in the parameter menu.

[MIDWAY BREAK] ! See Fig. 4.10.

[SET BCs] ! We won't stop for a break

[BOUNDARY CONDITION]

Here the flashing side at the bottom of element 1 in Fig. 4.11 prompts us to select a boundary condition from the menu.

[INSULATED]

The side quits flashing, returns to its original color, and the character "I"

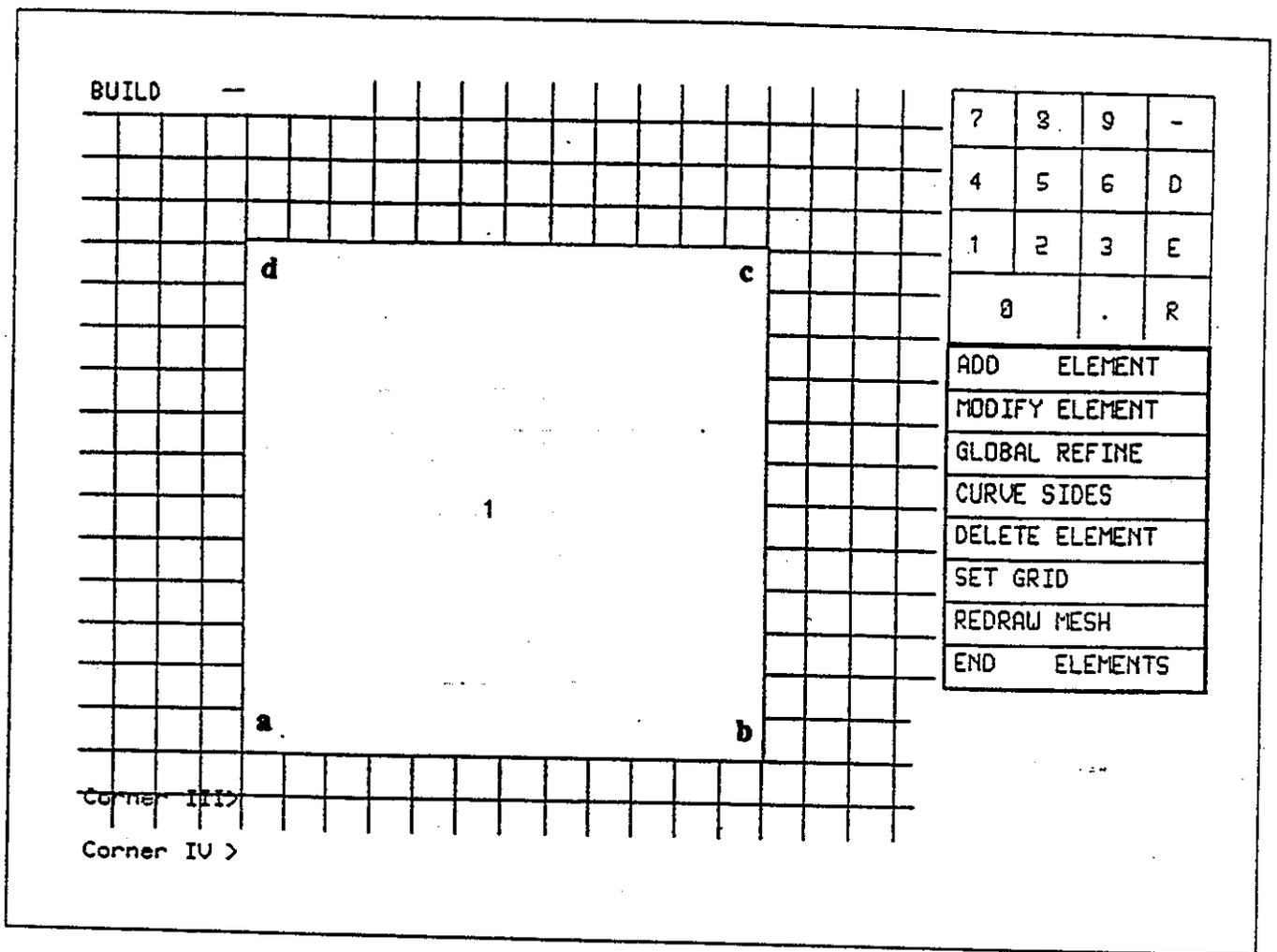


Figure 4.8

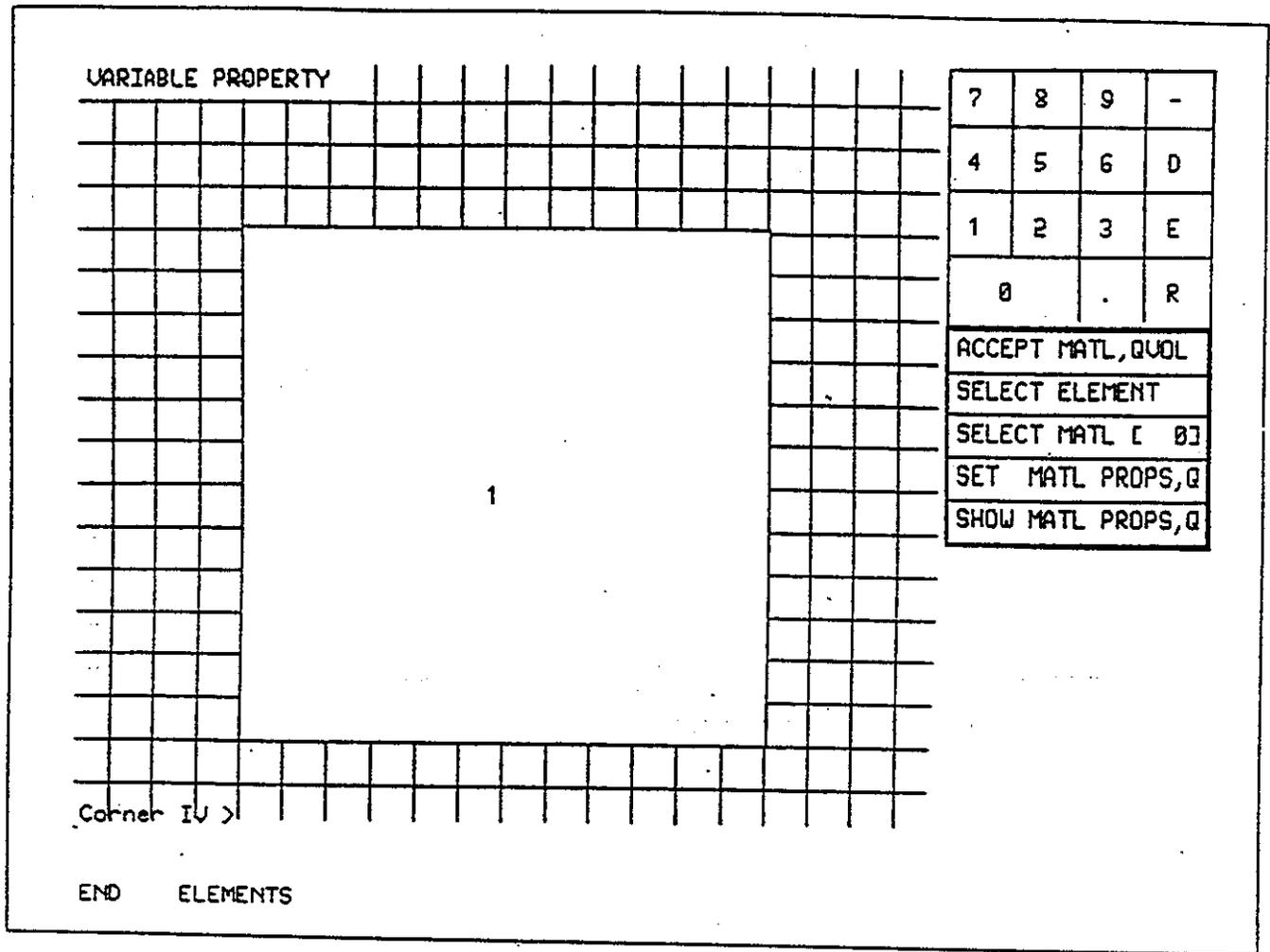


Figure 4.9

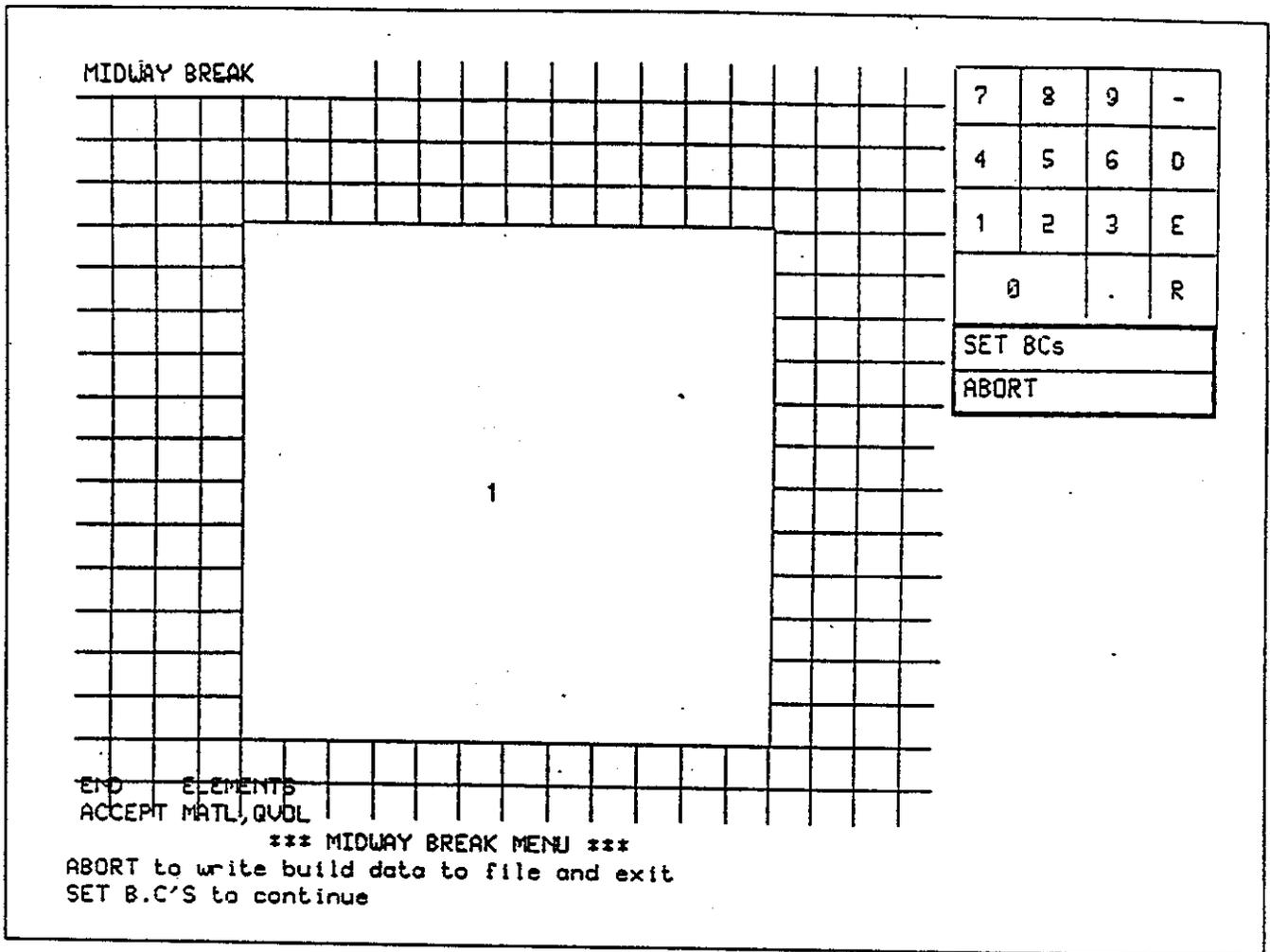


Figure 4.10

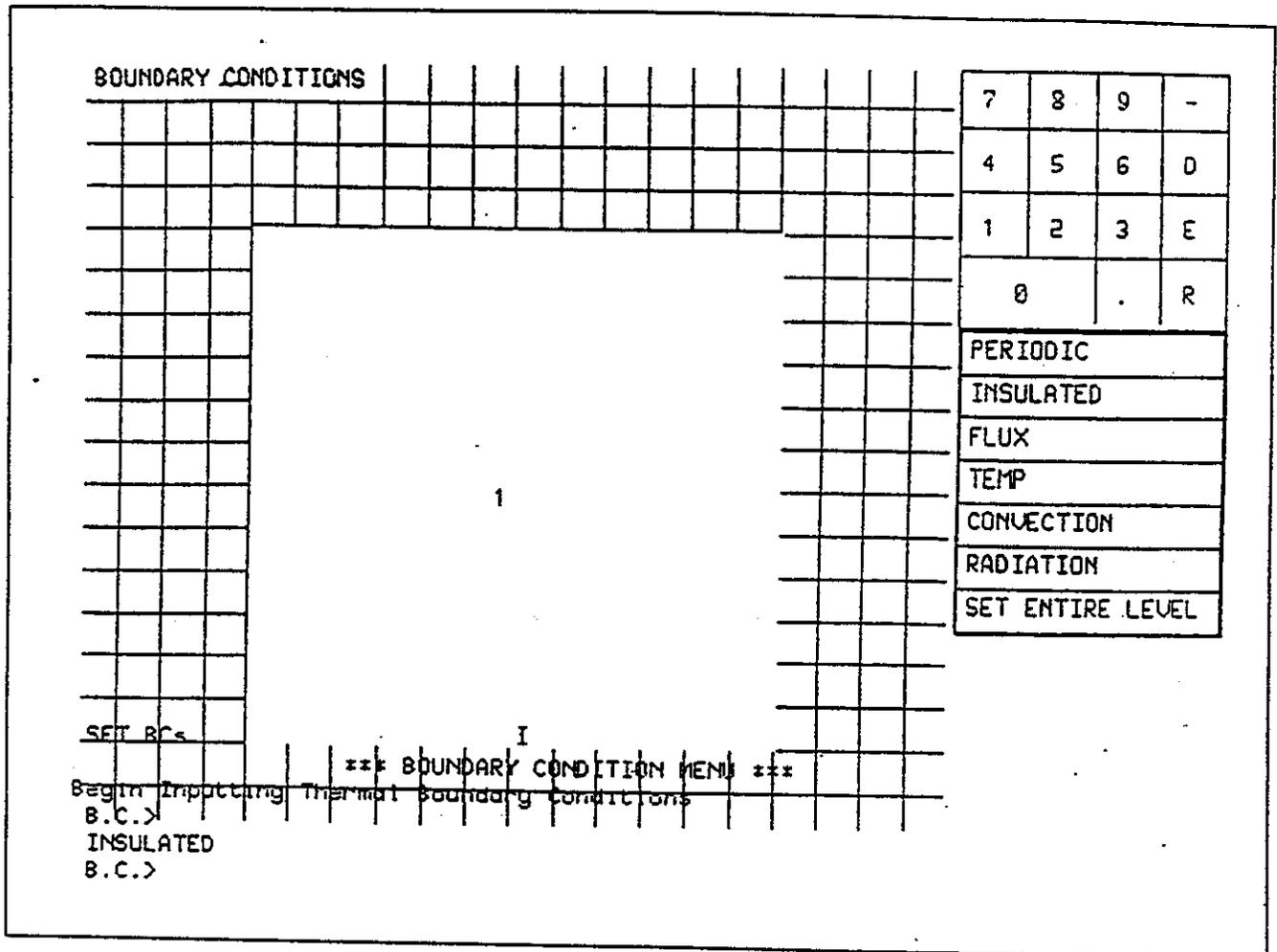


Figure 4.11

is drawn near the side's center signifying that the requirement for a boundary condition has been satisfied (see Fig. 4.11). The BOUNDARY CONDITIONS menu proceeds to prompt us to select a boundary condition for each of the remaining sides.

We specify a temperature of one for the right boundary of the domain.

[TEMP]

[FORTRAN FUNCTION] ! See Fig. 4.12.

[CONSTANT]

The border of the keypad becomes highlighted, signalling that the keypad is waiting for input.

(KP/1)

1

(KP/R)

[INSULATED]

We specify a temperature of zero for the left boundary of the domain.

[TEMP]

[FORTRAN FUNCTION]

[CONSTANT]

(KP/0)

0

(KP/R)

[MODIFY B.C.]

When all the boundary conditions are specified, we have the opportunity to modify them, see Fig. 4.13. If we are satisfied with the boundary conditions we proceed to the options menu.

[END LEVEL]

[OPTIONS]

In Fig. 4.14 we see the internal mesh in the element, the element number (1) at the element center and a one-character symbol for the boundary condition at the center of each of the 4 elemental boundary sides.

We do not want to invoke any of the OPTIONS from the last PRENEK menu, so we exit PRENEK.

[EXIT]

Exiting problem1 Session

§

NEKTON

Having generated all the necessary geometrical, physical, and numerical parameters required to perform the flow simulation, the user now accesses NEKTON.

§ {nek problem1} ! Recall that problem1 is the SESSIONNAME (i.e., simulation name) associated with this particular problem. Problem1 enables NEKTON to know that the data generated by PRENEK that defines this problem is stored in the file problem1.rea. Similarly, the data files generated by NEKTON will be called problem1.fld and problem1.his so that POSTNEK will be able to know where the output data is stored. The case of filenames is machine-dependent; in the VAX/VMS convention VMS automatically converts filenames to upper case.

FORTRAN STOP

Compiling custom routines in file problem1.for ! The file problem1.user can contain fortran statements which describe boundary conditions, initial conditions, and forcing functions. In this simulation we used constants rather than fortran statements to specify boundary conditions, but the routines in problem1.user must nonetheless be combined with other routines to make problem1.for, compiled, and linked to NEKTON. The messages given by a UNIX operating system will be similar in content but different in detail than the following VMS message:

Linking with NekObject:N52D

Initialization successfully completed

START OF SIMULATION

9 Helmholtz solver: 0.2097E-5

Simulation successfully completed

End of nekton PROBLEM1 simulation

This message states that there was a successful compilation and linking of the session, that 9 iterations inside the Helmholtz equation solver (which is the backbone of both the fluid and heat transfer solvers) were needed to bring the residual to .2097E-5, and that the simulation ended successfully without incident. The exact

[SET PLOT FORMAT]

[SET PLOT FORMAT]

The name of the menu brought up by POSTNEK appears to echo the menu item chosen by the user; this is because the submenu names in POSTNEK are the same as the item in the parent menu which invoked it.

[X-Y PLOT]

[X-Y PLOT]

[PROFILE]

[POSTNEK MAIN]

[SET LOCATION]

[SET LOCATION]

[LINE]

Enter 1st Endpoint:

(M/Y 4.15.a)

Enter 2nd Endpoint:

(M/Y 4.15.b) ! See Fig. 4.15.

Note that these points could have been defined quantitatively with the keypad rather than visually with the mouse. This is true for all mouse coordinate inputs. To switch from mouse inputs to coordinate inputs, push the mouse button with the cursor outside the graphics area and inside the menu area. PRENEK interprets this as a request to reject the graphical mouse input and choose keypad input.

[POSTNEK MAIN]

[PLOT]

A profile of the temperature vs. distance along the line segment is plotted on the screen and in Fig. 4.16. With each profile plot, the integral of the plotted quantity with respect to distance along the profile line is also calculated, and the result is displayed.

Note the procedure by which we obtained our plot: We began at the POSTNEK MAIN menu, went into submenus to alter various aspects of the plot, and finally displayed the plot using the [PLOT] command. This is the method by which all plots are constructed by POSTNEK. We begin with default plot aspects, and go into individual submenus to independently change aspects of plot. This system

number of iterations will vary with the release of NEKTON and need not match exactly. The final line of the message states that NEKTON has returned control to the operating system.

\$

POSTNEK

Having generated all the necessary data (field and history) in NEKTON, the user now enters POSTNEK to inspect the output.

\$

{post}

Enter Session Name (Default= problem1)

{ } ! A carriage return signals acceptance of the suggested default SESSIONNAME problem1.

The mesh is read from problem1.rea and is brought up along with the POSTNEK MAIN menu.

[POSTNEK MAIN]

[PLOT] ! We plot the result using all of the default plot characteristics. For our steady two-dimensional heat transfer problem, the default is a color fill temperature map. The result is a correct and colorful (if physically uninteresting) temperature plot. The colors represent evenly spaced temperature ranges.

Hit mouse button for menu

The POSTNEK MAIN menu is displaced by a color bar which quantifies the values of the temperatures along the isotherm lines that separate the color levels. The color bar remains there to be photographed or dumped to a hard copy unit. To recover the POSTNEK MAIN menu to continue the POSTNEK session, hit the mouse button with the cursor anywhere on the screen:

(M/Y) ! Cursor can be anywhere on the screen.

[POSTNEK MAIN]

We will next modify the plot format to get a profile plot of the temperature along a line.

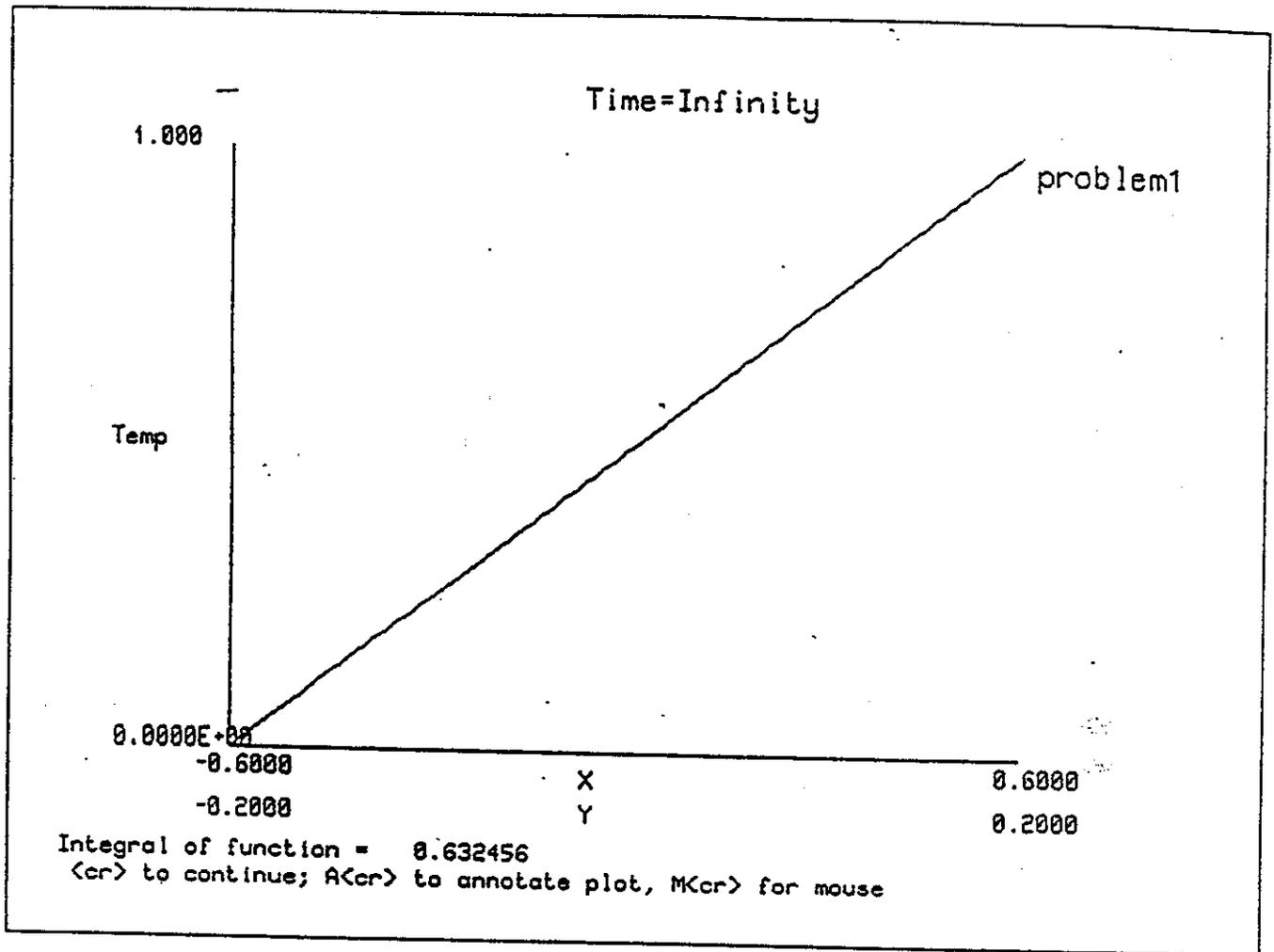


Figure 4.16

allows for the most flexibility in displaying results.

We now move on to a more complex example.

```
{  
[EXIT]
```

4.2.4 Problem 2: An Unsteady Two-Dimensional Navier-Stokes Problem with Natural Convection

PRENEK

The user should be logged into his workstation via a graphics window or through his graphics terminal. Again, the session begins with the user typing "pre" to the operating system prompt;

```
{pre}
```

This returns the user to the PRENEK program, which is confirmed by the programs inquiry for a SESSIONNAME label.

Choose a Name for This Session: {problem2} ! The user chooses the name "problem2" for this session. It is important that this name be different from that of the previous session. Creating a new session with the same name as an old session will cause new files to be created on top of the old files. In a VMS operating system, this will result in the old files being "buried" under new files that have higher version numbers. In a UNIX system, PRENEK will append the old filenames with a "~" character, allowing for one duplicate SESSIONNAME before destroying any data. However, either of these situations requires careful file bookkeeping. To avoid confusion, duplicate SESSIONNAMEs should be avoided.

PRENEK now clears the screen, draws the keypad and presents the first menu, the READ PARAMETER menu:

```
[READ PARAMETER]
```

This time we wish to demonstrate how to generate a new session from a read file-(the file with the extension ".rea") of an old session. This procedure can be used to make a set of parametric runs, to perform a mesh refinement procedure, or as part of an incremental problem definition. We will use session1 as a baseline from which to make modifications to the equation type, parameters, and mesh.

[READ PREVIOUS PARAMETERS]

This causes PRENEK to read in the equation type and numerical parameters, rather than to prompt the user for them via menus.

Enter name of previous session

We enter the SESSIONNAME problem1:

{problem1}

Control skips the SET EQUATION TYPE and SET PARAMETER menu and goes directly to the CENTRAL menu.

[CENTRAL]

From here we could continue onto the mesh generation using the equation type and parameter data of session1, but instead choose to change the equation type and some of the numerical parameters.

[ALTER PARAMETERS]

[SET EQUATION TYPE] ! See Fig. 4.17.

CURRECT SETTING: STEADY TWO DIMENSIONAL HEAT TRANSFER

[UNSTEADY]

The menu item "UNSTEADY N" is chosen and the menu reappears with the item toggled to "UNSTEADY Y".

[SET EQUATION TYPE]

CURRECT SETTING: UNSTEADY TWO DIMENSIONAL HEAT TRANSFER

[FLUID FLOW]

The flag that governs the fluid flow calculation is turned on in a manner similar to the unsteady flag.

[SET EQUATION TYPE]

CURRECT SETTING: UNSTEADY TWO DIMENSIONAL STOKES AND HEAT TRANSFER

[ADVECTION]

The advection flag is turned on, which makes a Navier- Stokes problem rather than a Stokes problem.

[SET EQUATION TYPE]

CURRECT SETTING: UNSTEADY TWO DIMENSIONAL NAVIER-

SET EQUATION TYPE

NEKTON

ACCEPT CURRENT SWITCHES	
DIMENSION	2
UNSTEADY	Y
FLUID FLOW	Y
ADVECTION	Y
STRESS FORMULATION	N
SPLIT FORMULATION	Y
TURBULENCE MODEL	N
HEAT TRANSFER	Y
ADOL PASSIVE SCALARS	0
MULTIGRID SOLUTION	N

7	8	9	-
4	5	6	D
1	2	3	E
0	.		R

CURRENT SETTING: UNSTEADY TWO DIMENSIONAL HEAT TRANSFER
CURRENT SETTING: UNSTEADY TWO DIMENSIONAL STOKES AND HEAT
CURRENT SETTING: UNSTEADY TWO DIMENSIONAL NAVIER-STOKES AN
Time stepping scheme set to First Order
CURRENT SETTING: UNSTEADY TWO DIMENSIONAL NAVIER-STOKES AN

Figure 4.17

STOKES AND HEAT TRANSFER

[SPLIT FORMULATION]

The split formulation is the faster Navier-Stokes algorithm in NEKTON, and it is appropriate to use for problems such as this, in which the Reynolds number is not too small, and no stress boundary conditions are applied (see Section 3.4).

[SET EQUATION TYPE]

[ACCEPT CURRENT SWITCHES]

We accept this equation type, which is unsteady two-dimensional Navier-Stokes with heat transfer solved with the split formulation.

[SET EQUATION TYPE]

[ACCEPT CURRENT SWITCHES]

Here we have the option of coupling or decoupling the heat transfer problem with the fluid problem via the convective term. We accept the default, in which the two equations are coupled, see Fig. 4.18.

[ACCEPT CURRENT SWITCHES]

Lastly, we have a choice as regards the treatment of the advection (convection) term, Fig. 4.19. We choose the standard semi-implicit scheme (see Section 3.2), and proceed to the SET PARAMETER menu

[ACCEPT CURRENT SWITCHES]

[SET PARAMETERS] ! See Fig. 4.20.

We set the value of the viscosity to 0.01

{.01}

We keep the default value of 1.0 for ρC_p .

}

Here the default value for the conductivity is 2.0, as set in problem1. We will change the conductivity to 0.01.

{0.01}

We will keep the default value of 5 for NORDER (the polynomial degree in each spatial direction within each element is equal to NORDER-1).

}

In the the second screen of the SET PARAMETER menu, Fig. 4.21, we will set NSTEPS, the number of time steps, DT, the value of the time step, and

SET EQUATION TYPE

NEKTON

ACCEPT CURRENT SWITCHES	
WITH CONVECTION	Y

7	8	9	-
4	5	6	D
1	2	3	E
0	.		R

CURRENT SETTING: UNSTEADY TWO DIMENSIONAL STOKES AND HEAT T
CURRENT SETTING: UNSTEADY TWO DIMENSIONAL NAVIER-STOKES AND
Time stepping scheme set to First Order
CURRENT SETTING: UNSTEADY TWO DIMENSIONAL NAVIER-STOKES AND
Convection in Temperature Field

Figure 4.18

SET EQUATION TYPE

NEKTON

ACCEPT CURRENT SWITCHES	
CHARACTERISTICS	N

7	8	9	-
4	5	6	D
1	2	3	E
0	.		R

CURRENT SETTING: UNSTEADY TWO DIMENSIONAL STOKES AND HEAT
CURRENT SETTING: UNSTEADY TWO DIMENSIONAL NAVIER-STOKES AN
Time stepping scheme set to First Order
CURRENT SETTING: UNSTEADY TWO DIMENSIONAL NAVIER-STOKES AN
Convection in Temperature Field

Figure 4.19

SET PARAMETER

NEKTON

VISCOS	0.100000E-01
RHOCF	1.00000
CONDUCT	0.100000E-01
NORDER	5.00000
TORDER	1.00000

7	8	9	-
4	5	6	0
1	2	3	E
0	.	R	

0.01

Figure 4.20

IOSTEP, the interval between data dumps. We will turn on natural convection by giving a nonzero value for βg , the product of the volumetric expansion coefficient times the gravitational constant. We will let NEKTON set the time step based on stability criterion by setting the time step to be zero. We will keep the default values for all other parameters.

```
{  
{0.1}  
{  
{  
{  
{20} ! 20 Time steps.  
{0.0} ! Set the time step to zero.  
{  
{10} ! We specify data dumps every 10 steps.  
{  
{  
{  
{  
{
```

[CENTRAL]

Which brings us back to the CENTRAL menu. Assuming we are satisfied with the current status of the equation type and parameters, we continue on to build the mesh. We want to begin with the mesh from problem1 as the base from which to construct our new mesh. We select

[BUILD FROM FILE]

Enter name of previous session

We are prompted for the name of the session from which to read the mesh and subsequent data. We type

```
{problem1}
```

to use the mesh from problem1.

In this example we read both the equation type and mesh from the same session. It is also permissible to read from two different sessions; one can "mix and match" to define a problem from two different data sources.

SET PARAMETER

NEKTON

DENSITY	1.00000
BETAG	0.100000
GTHETA	0.000000E+00
QUOL	0.000000E+00
FINTIME	0.000000E+00
NSTEPS	20.0000
DT	0.000000E+00
IOTIME	0.000000E+00
IOSTEP	10.0000
GRID	0.500000E-01
HELMHOLTZ	0.100000E-04
TOLREL	0.100000E-01
TOLABS	0.100000E-01
COURANT	0.250000

7	8	9	-
4	5	6	0
1	2	3	E
0	.		R

10

Figure 4.21

We now have the opportunity to either modify or accept the mesh, variable properties, boundary conditions, and options. After reading in the mesh, the menu for each of those four categories will appear. The top item in each menu will cause PRENEK to accept the current status of that category.

For this problem we will do modifications in three of the four categories to demonstrate modification techniques. The first category is the mesh (see Fig. 4.22).

[ACCEPT/REVIEW]

[REVIEW/MODIFY]

This puts us into the BUILD menu. All of capabilities of the BUILD menu are available to edit the mesh. Of particular usefulness is the GLOBAL REFINE submenu. This is typically used when preliminary results from an initial simulation with low resolution (few elements) suggest where more resolution is needed.

We first invoke the GLOBAL REFINE submenu.

[GLOBAL REFINE]

[GLOBAL REFINE] ! See Fig. 4.23.

We start with the ZIPPER command. This command splits one or more elements. We first invoke the ZIPPER command.

[ZIPPER]

Push yellow button to enter element to be split:

(M/Y 4.23.a)

Enter a side of this element that is to be split:

(M/Y 4.23.b)

The characters "a" and "b" appear on the top and bottom sides of the original single element. These characters straddle the split that is to be created. These characters also mark the orientation by which we can adjust the location of the split. We wish to put the split in the middle of the element. To accomplish this, we set the linear factor S to 0.5

Enter S, the location of the split:

(KP/0)

0 -

(KP/.)

0.

(KP/5)

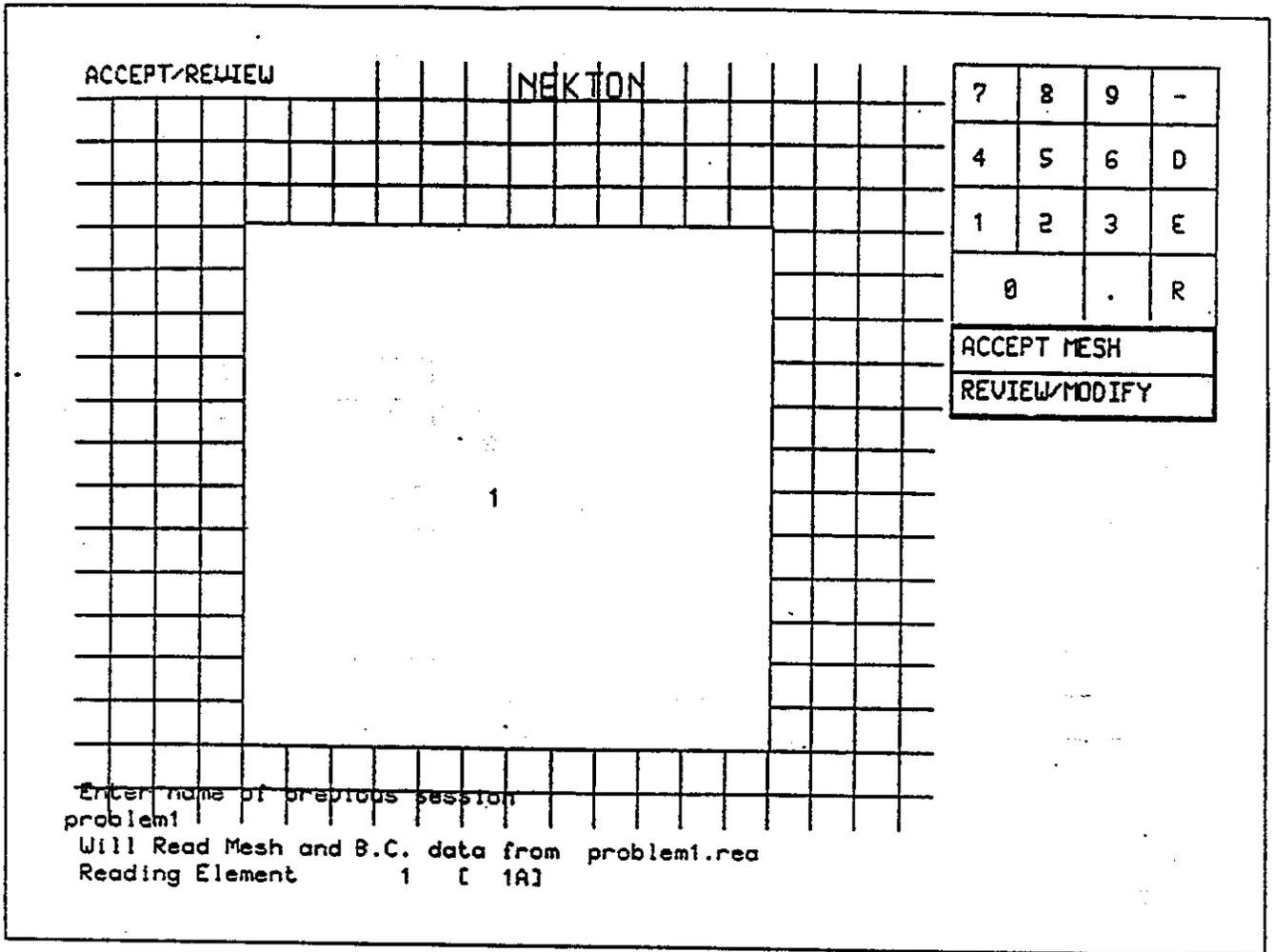


Figure 4.22

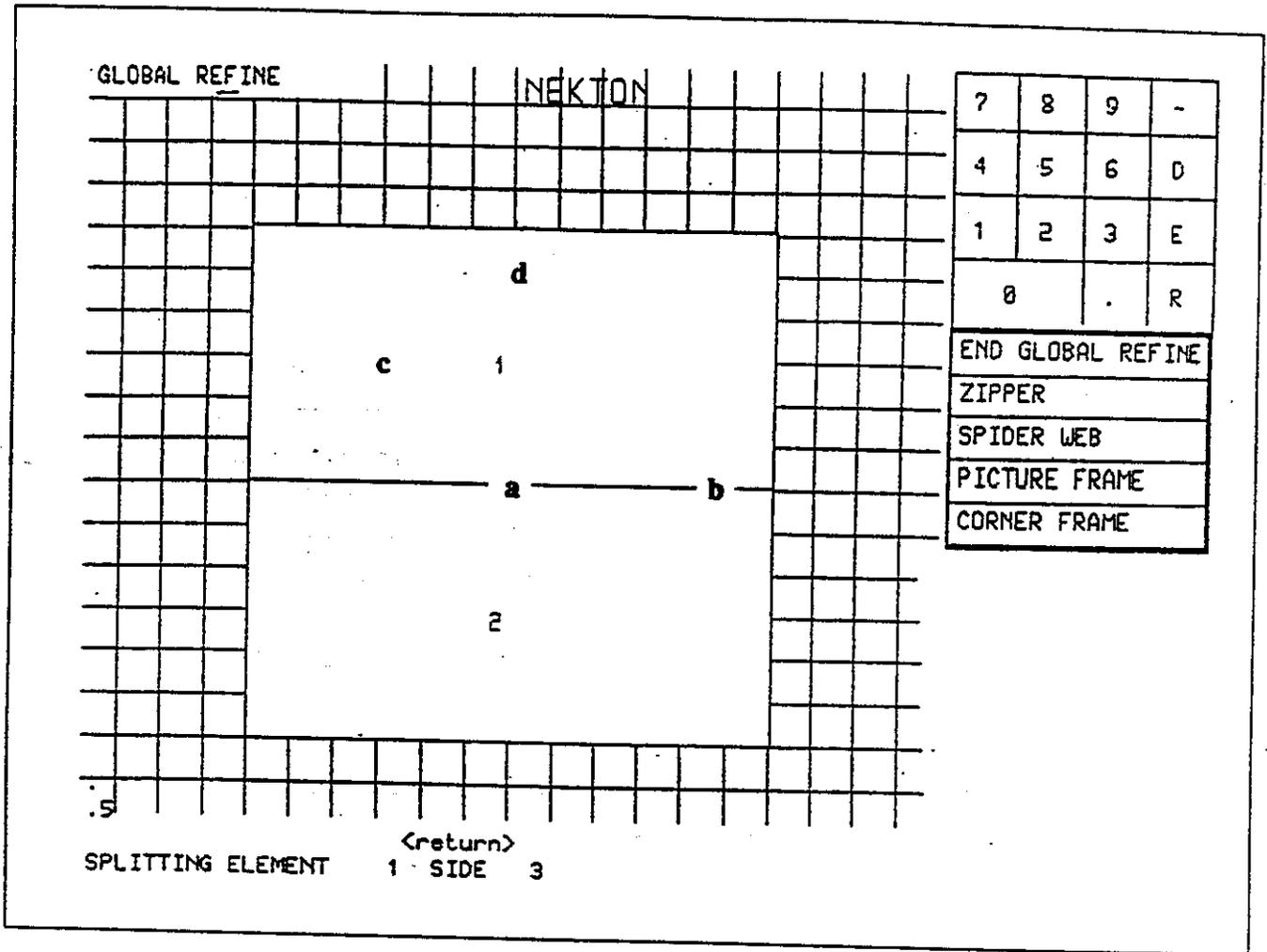


Figure 4.23

0.5

(KP/R)

<return>

We next split the two elements vertically.

[GLOBAL REFINE]

[ZIPPER]

Push yellow button to enter element to be split:

(M/Y 4.23.c)

Enter a side of this element that is to be split:

(M/Y 4.23.d)

We again split them evenly.

Enter S, the location of the split:

(KP/0)

0

(KP/.)

0.

(KP/5)

0.5

(KP/R)

<return>

Note that both elements 1 and 2 are split, even though we only specified that element 1 be split. The ZIPPER command operates such that elemental interfaces remain continuous. Thus, we can initiate a "crack" in a single element which will propagate throughout the mesh and terminate when it reaches a physical boundary. In a three-dimensional mesh, it will also propagate up and down levels.

As we have completed the global refinement of the mesh, we select

[END GLOBAL REFINE]

We will next curve two of the sides on the mesh boundary.

[BUILD]

[CURVE SIDES]

[CURVE SIDES] ! See Fig. 4.24.

[MAKE CIRCLE]

Enter element side:

(M/Y 4.24.a)

Enter Circle Radius

(KP/0)

0

(KP/.)

0.

(KP/7)

0.7

(KP/R)

<return>

The result should look like that in Fig. 4.24. Continuing onto the next side,

[CURVE SIDES]

[MAKE CIRCLE]

Enter element side:

(M/Y 4.24.b)

Enter Circle Radius

(KP/-)

=

(KP/0)

-0

(KP/.)

-0.

(KP/7)

-0.7

(KP/R)

<return>

The negative radius corresponds to a concave curve, see Fig. 4.25.

[CURVE SIDES] [BUILD MENU]

[BUILD]

As we have completed the editing of the mesh, we select

[END ELEMENTS]

PRENEK next brings us to the variable properties menu

[VARIABLE PROPERTY] ! We accept the current status.

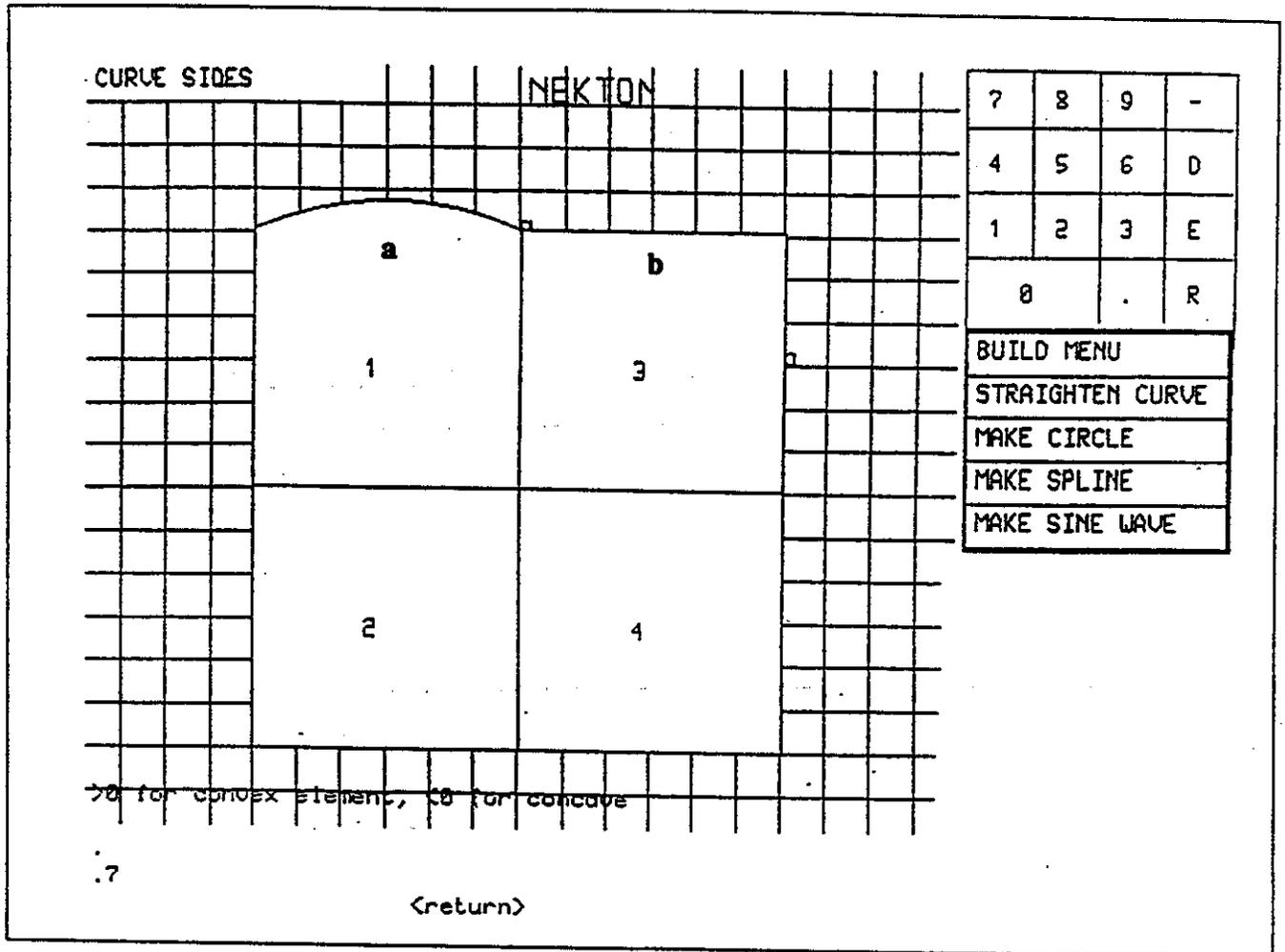


Figure 4.24

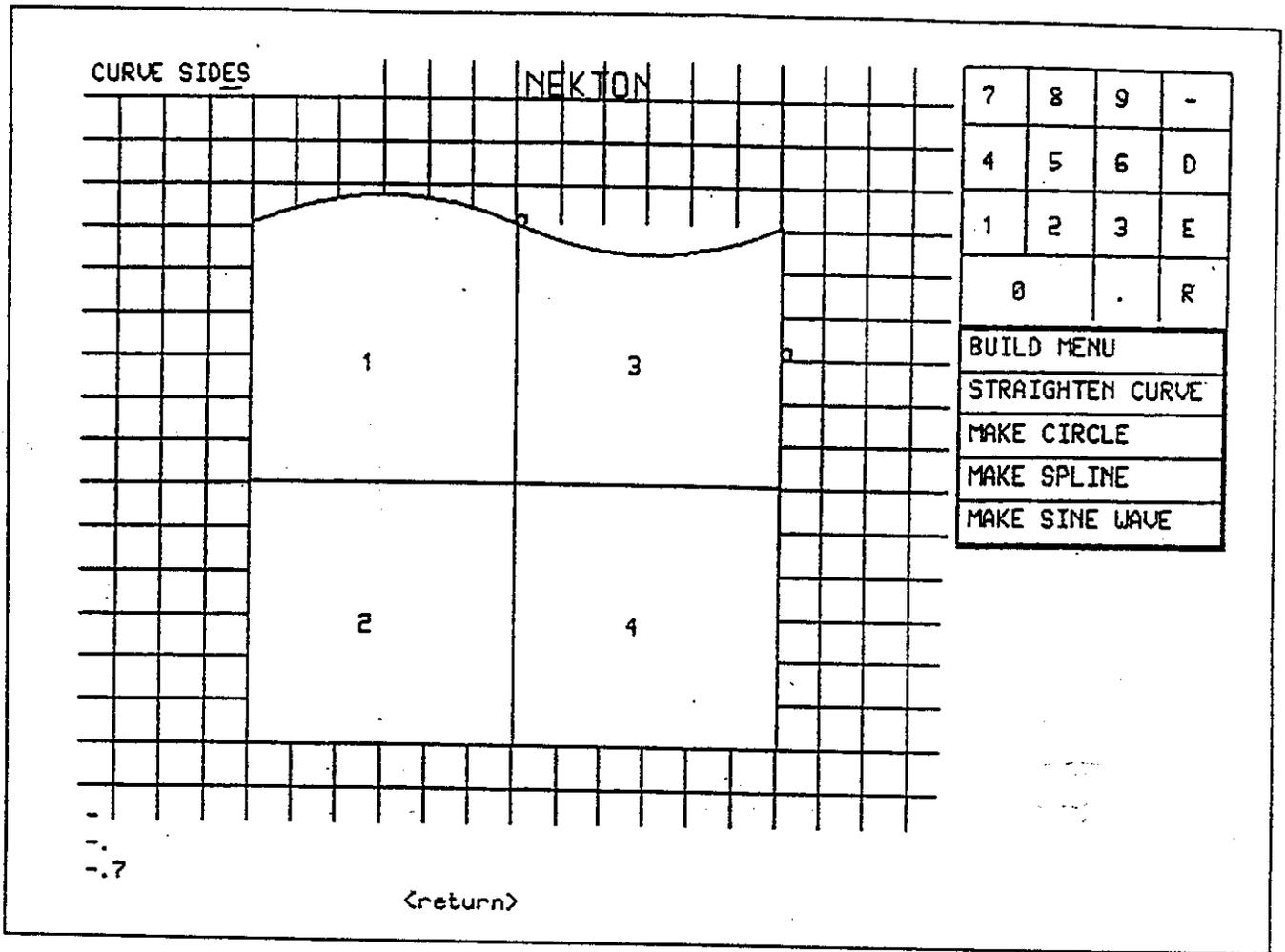


Figure 4.25

[ACCEPT MATL,QVOL]
[MIDWAY BREAK]
[SET BCs]

PRENEK now requests entry of boundary conditions. Had we kept the equation type from that which we read from in session problem1, the boundary conditions would have been retained through the mesh refinement process. This is a characteristic of mesh refinement: *The daughter sides created in mesh refinement will take on the boundary conditions of the parent side, except for periodic boundaries.* For periodic boundaries, PRENEK will not try to guess which of the daughter sides you wish to connect. Those daughter sides will be given blank boundary conditions. The BOUNDARY CONDITION menu will then insist that the user manually select boundary conditions for those sides.

In this example, however, we must enter the boundary conditions for our new equation. We begin with the fluid boundary conditions, all of which are walls (i.e., have zero normal and tangential velocities). We can set all the boundary conditions to walls in two mouse inputs:

[BOUNDARY CONDITIONS]
[SET ENTIRE LEVEL]
[WALL]

We next accept the fluid boundary conditions and begin with the heat transfer boundary conditions.

[MODIFY B.C.]
[END LEVEL]
[BOUNDARY CONDITIONS]

We will impose a temperature of 1 on the right boundary, a temperature of zero on the left boundary, and insulate the top and bottom boundaries.

[INSULATED]
[TEMP]
[FORTRAN FUNCTION]
[CONSTANT]
(KP/1)
1
(KP/R)

<return>

To avoid keying in the temperature boundary condition again, we use the *blue* button to copy the boundary condition from the previously entered side.

(M/B) ! Place cursor close the side where the $T = 1$ was set.

[INSULATED]

[INSULATED]

[TEMP]

[FORTRAN FUNCTION]

[CONSTANT]

(KP/0)

0

(KP/R)

<return>

(M/B) ! Place cursor close the side where the $T = 0$ was set.

[MODIFY B.C.]

[END LEVEL]

[OPTIONS]

We will define two history points, see Fig. 4.26.

[HISTORY]

[HISTORY]

[ENTER POINT]

(M/Y 4.26.a)

[ENTER POINT]

(M/Y 4.26.b)

The location of each history point will be marked by a small cross. We return up to the main OPTIONS menu and exit.

[MAIN MENU]

[OPTIONS]

[EXIT]

Exiting problem2 Session

\$

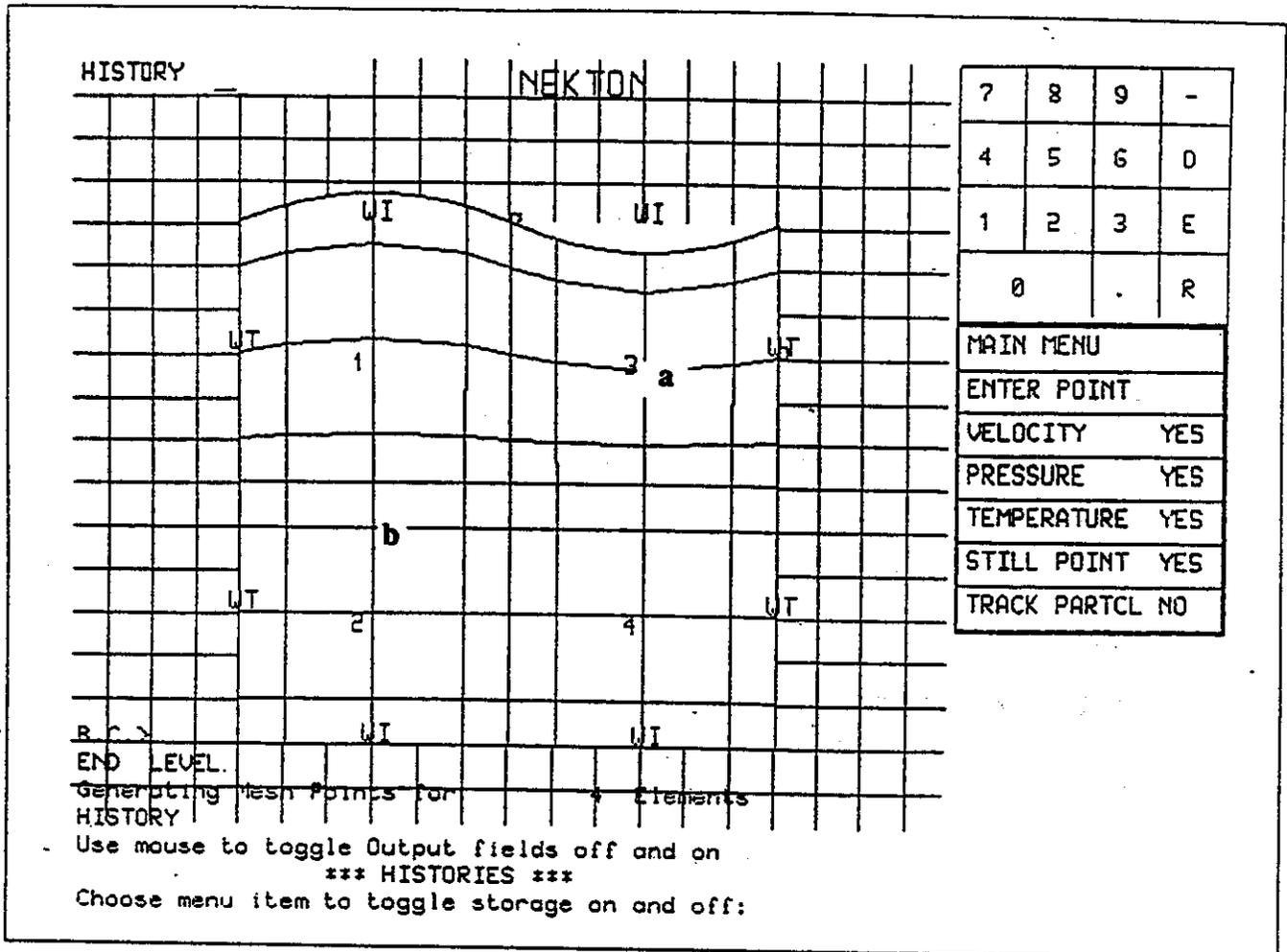


Figure 4.26

NEKTON

In the usual format, the problem is run through the NEKTON solver by typing

```
{nek problem2}
```

For this problem, time stepping information is given in addition to the compiling and linking information.

POSTNEK

Enter Session Name (Default= problem2)

{ } ! We accept the suggested session problem2 for post-processing.

```
[POSTNEK MAIN]
```

The mesh is displayed on the screen with the history points marked with small crosses.

```
[PLOT] ! We plot the default, which is the vector velocity field at the last data dump, see Fig. 4.27.
```

```
(M/Y) ! Cursor can be anywhere on the screen.
```

```
[POSTNEK MAIN]
```

We change the quantity to be plotted from velocity to temperature, and request a plot

```
[SET QUANTITY]
```

```
[SET QUANTITY]
```

```
[TEMPERATURE]
```

```
[POSTNEK MAIN]
```

```
[PLOT]
```

We set the time for subsequent plots to be at the first dump, at an earlier earlier time of about 7.

```
(M/Y) ! Cursor can be anywhere on the screen.
```

```
[SET TIME]
```

```
(M/Y 4.27.a)
```

```
[POSTNEK MAIN]
```

```
[PLOT]
```

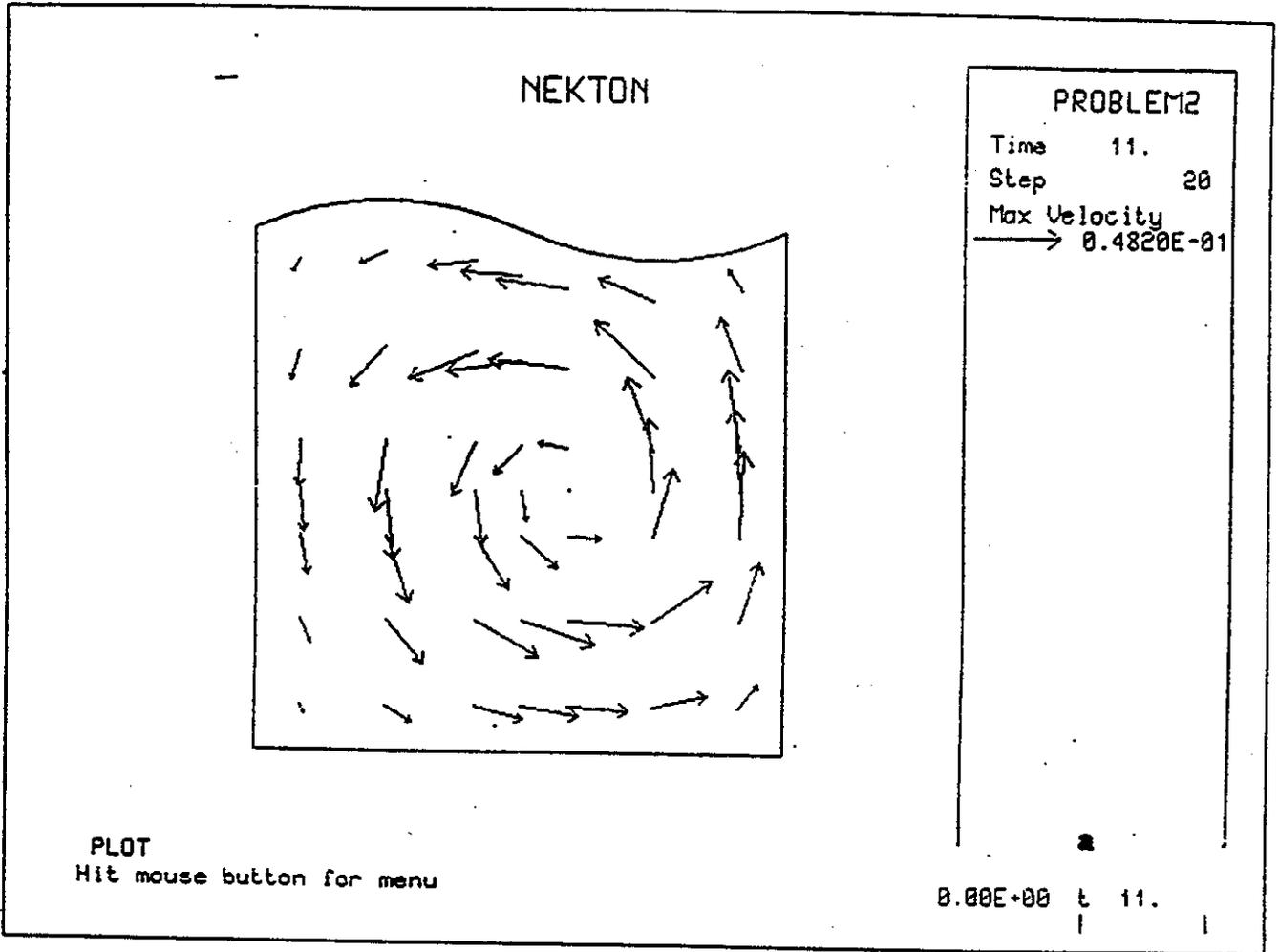


Figure 4.27

Note as POSTNEK paints the data from the earlier time over the data from the final time that the isotherms shift to the right; this corresponds to the diffusion of heat from the hot right side (Temperature=1.0) throughout the domain whose initial temperature was 0.0.

To increase the amount of detail from the coarse color fill in this default quick-look plot to a finer presentation-quality color fill, we go into the set attribute submenu.

```
{  
[POSTNEK MAIN]  
[SET ATTRIBUTE]  
[SET ATTRIBUTE]  
[SET RESOLUTION]  
[SET RESOLUTION]  
[COLOR FILL]  
{25}  
[POSTNEK MAIN]  
[PLOT]
```

We next plot the temperature at a point as a function of time. To do this, we change the PLOT FORMAT

(M/Y) ! Cursor can be anywhere on the screen.

```
[POSTNEK MAIN]  
[SET PLOT FORMAT]  
[SET PLOT FORMAT]  
[X-Y PLOT]  
[X-Y PLOT]  
[HISTORY]  
[POSTNEK MAIN]  
[PLOT]
```

After the screen clears itself, we are presented with a time history of temperature (see Fig. 4.28) at the default location, i.e., the location of the first history point. From this plot, we see that the temperature at this location is approaching steady state.

This calculation began with default (zero) initial conditions. Had we wished

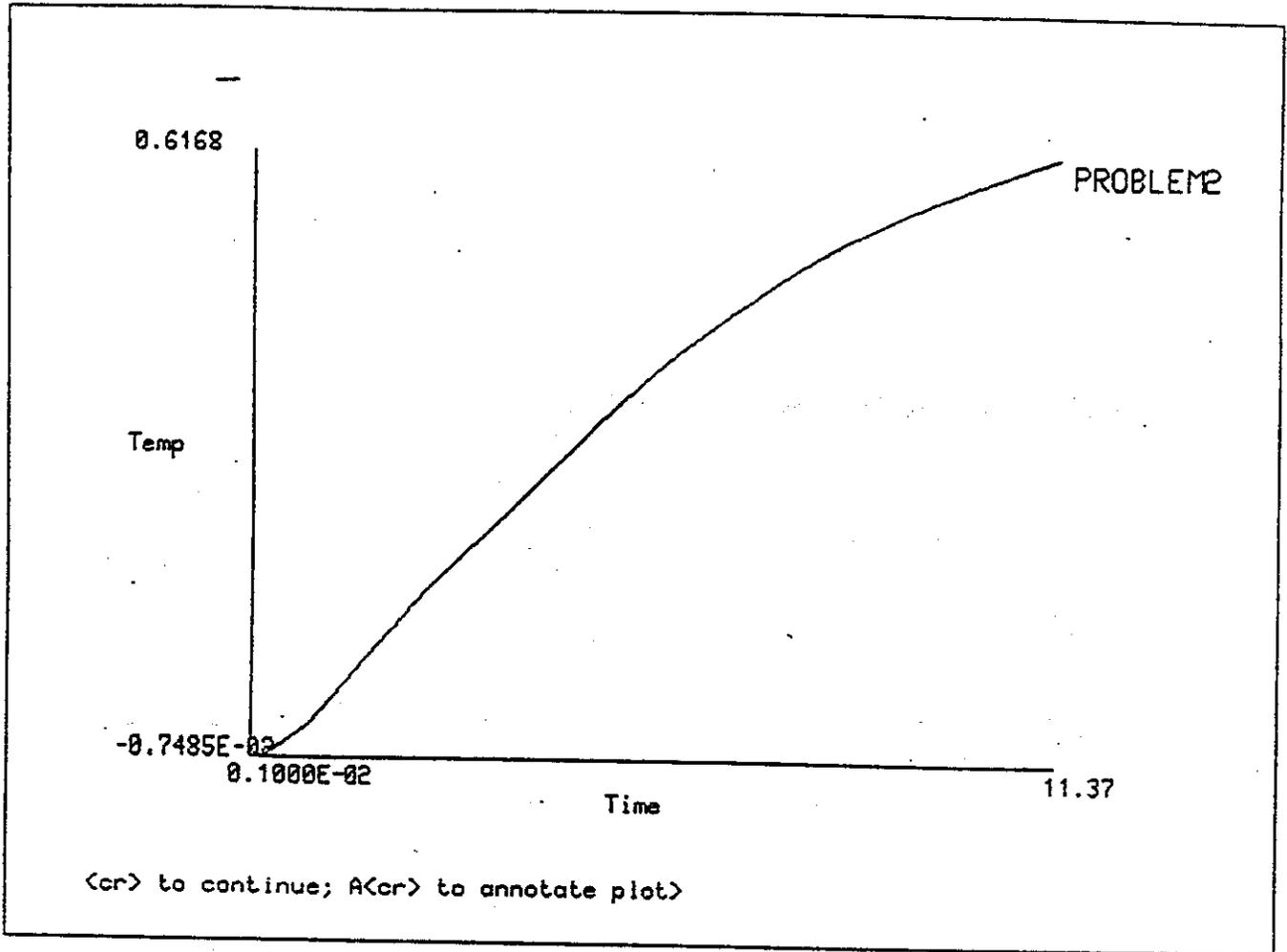


Figure 4.28

to accelerate the approach to steady state, we could have used the initial conditions menu in PRENEK to specify different initial conditions. As an exercise, the user may want to specify through a fortran function a linear initial temperature profile, or use the PRESOLV option in order to start with the steady conduction solution for the temperature. (see the INITIAL CONDITIONS menu in Chapter 8).

```
{  
[POSTNEK MAIN]  
[EXIT]
```

4.3 Tips for Saving Time

In this section we cover two topics related to generating faster results. We first cover ways in which the user can effectively and efficiently use his time to define problems and generate meshes in PRENEK. We next discuss methods of posing the problems that allow for efficient (in terms of computer time) solution by NEKTON.

4.3.1 Reducing the Setup Time Required by PRENEK

By this time the reader should have experience defining problems with PRENEK through the two example sessions. The user may have noticed that some operations (such as mesh refinement) in PRENEK are inherently irreversible, resulting in a unidirectional "flow" of the preprocessor. The user may also have noticed that a given problem definition can be recycled to define a new problem. A strategy for PRENEK usage in view of these two program characteristics is developed.

The strategy is as follows. When first beginning work on a given problem, the user should create a session with the minimum number of elements required to describe the geometry. This session should have all the necessary boundary conditions and initial conditions to describe the problem. This session should be saved as a baseline from which future sessions are created. At this point the baseline problem might be run through the NEKTON solver (possibly with a reduced Reynold's number), in order to quickly check the correctness of the boundary conditions, forcing functions, initial conditions, etc.

Next, the user should begin a new session, and read in the parameters and

mesh from the first session. After reading in the baseline mesh, PRENEK gives the user the opportunity to modify that mesh. Through the BUILD and GLOBAL REFINE-menus the user can perform quick refinement of the mesh in order provide sufficient resolution to accurately solve the problem. The mesh refinement process assigns the correct boundary conditions to the new elements in the refined mesh, so that no new boundary conditions need be input. History points are referenced by element number, so that definition of history points should be delayed until the refinement is done.

The strategy behind this approach is to view rough results early in the modelling process to recognize and correct potential errors in problem definition before substantial time is invested. Additionally, the automatic mesh refinement reduces the number of element entries and boundary condition specifications required. And finally, the existence of the minimal mesh allows for easy implementation of different mesh refinement strategies, should they be needed.

Another strategy for avoiding unnecessary work is to use the break point provided between the element definition and the boundary specification menus. Here one can abort the session and save the mesh to a file before setting the boundary conditions. Upon reading in the mesh in a new session, the boundary conditions can be added without endangering the work invested in mesh construction. If a gross mistake is made in setting boundary conditions, a "control C" can be used to scrap that session and the intact mesh can be read into another new session. This is particularly useful when the geometry is sufficiently complex so that the minimal baseline mesh that describes the geometry is still rather complex.

The usual method for modifying a NEKTON problem specification is to create an entirely new session by reading the old session into PRENEK and making the modifications. For a few simple modifications, such as changing a parameter such as viscosity, thermal conductivity, or number of time steps, there is a shortcut. These parameters can be changed by editing the "read" file, SESSIONNAME.REA, created by PRENEK, see Chapter 5. This shortcut can be used to modify parameters from the list at the top of the read file. These parameters include viscosity, thermal conductivity, number of time steps, etc. This shortcut should *not* be used to change the type of equation, the order of the polynomial, the number of elements, or fortran functions, because these changes would also need to be done consistently

in other files created by PRENEK (the files SESSIONNAME.CMN and SESSIONNAME.USER) If this shortcut is to be used where the results of the old session are of interest, a copy of the old files should be made so that the new results will not bury the old files or overwrite them with new data.

A second shortcut can be used to avoid an unnecessary compile and link. If the executable image exists for a given problem, that is, it has already been run except for parameter changes, the recompile can be avoided by typing `{nek session run}` instead of `{nek session}`.

4.3.2 Reducing the CPU Time Required by NEKTON

In this section we discuss the CPU times required by NEKTON for solution of various problems and suggest several ways of reducing this time. With a philosophy similar to that of the previous section, the CPU time used by NEKTON can be substantially reduced. Large production runs can be preceded by smaller checkout runs, and final accurate solutions can be preceded by rough initial tests.

The restart option in the initial conditions of PRENEK can invoke a useful feature of NEKTON for economizing CPU time. Restarts can be made from a coarse mesh to a finer mesh, where the macro-elemental mesh is the same but the NORDER is higher in the finer mesh. In this strategy, the recommended initial coarse mesh will typically have $NORDER = 5$. The solution produced with this mesh can be read in as initial conditions for a final simulation with a higher NORDER. The NORDER recommended for these accurate final solutions is 7 to 9. This strategy can be used effectively to give quick convergence to problems with high Reynolds numbers. The steady solution of a low- Reynolds number problem in which low resolution is used can be used as initial conditions for the solution of a high Reynolds number problem with higher resolution. In this way, part of the expensive and slow transient associated with high Reynolds number flows can be avoided.

For steady coupled conduction-convection problems the steady solution is independent of the thermal mass of the solid portion of the domain. The time to steady state can usually be reduced substantially by reducing the thermal mass ($RhoCp$) of the solid by a fixed factor (say, 10^4). Note, however, that reducing the thermal mass of the liquid will *not* result in the correct steady solution.

For steady natural convection problems the convergence to a steady solution can be accelerated by using the PRESOLV command, allowing the steady conduction solution as the initial condition for the temperature, and the steady Stokes solution as the initial condition for the velocity field. There is a long startup transient associated with the diffusion of heat in to an isothermal fluid. The natural convection will begin only after the diffusive process alters the temperature field. This long transient is avoided by starting with a conduction solution. The bouyant forces driving the natural convection begin immediately from the initial temperature field. This sets up the convective heat transfer, in which (especially for high Rayleigh number flows) the time scales are much lower than those in diffusive heat transfer.

NEKTON uses, for problems involving convection, a solution technique which is inherently unsteady. Steady solutions are calculated by running an unsteady problem for sufficient time for it to reach steady state. The two problems for which a steady solution technique can be used directly are conduction heat transfer and Stokes flow. The implications of the unsteady solution technique are important in the calculation of the CPU time that NEKTON will use for a given problem; the total run time will here be the CPU time per time step, summed over all time steps.

The CPU time per time step in a NEKTON calculation depends on several factors. First, it depends on the speed of the machine on which it runs. The number of MFLOPS achieved by NEKTON on various machines ranges from 0.1 MegaFlops on a MicroVax II, to 170 MegaFlops on a four-headed Cray-2. Second, the work per time step is not constant in a given problem. As a solution approaches steady-state, the work decreases because the method requires fewer iterations. Thus, any estimates will have considerable uncertainty. Third, the computational time also depends on the equation being solved. Steady conduction requires the least cpu time, followed by unsteady conduction, steady Stokes flow, unsteady Stokes flow, and finally, Navier-Stokes problems.

In all cases the CPU time depends on the number of elements, the order of the elements, and the number of time steps. Both the elemental resolution and the number of time steps required for a given problem can increase with increasing Re. The Reynolds number is a primary consideration in CPU time estimates.

The number of time steps required in the unsteady algorithms depends

strongly on the Reynolds (or Peclet) number where the time step is constrained by the Courant criterion (3.1). This constraint has the effect of limiting the distance that a particle of fluid can move in a single time step to on the order of a local mesh spacing. For a computational domain of characteristic length L , the number of steps required to convect the fluid throughout the domain will be $2L/\Delta x$ based on a convective time scale of L/U where U is a characteristic velocity. The time to steady state may also include a diffusive time scale, which will depend on the existence, location, and thickness of boundary layers. At worst case, the diffusive time scale would be of order (Re) times the convective time scale. The arrival at steady state should be verified by checking the temporal behavior of the flow using history points.

To the times discussed above should be added the overhead time required for compilation and linking. The time required to compile the entire 20,000 lines code is approximately 15 minutes on a workstation and about a minute for a supercomputer. For short runs and checkout runs, this time can be significant. To avoid this, a set of precompiled versions of the code exist for the most commonly used conditions, see Chapter 6.

Chapter 5

PRENEK

5.1 Structure

This chapter describes in detail the structure, commands, and menus available in PRENEK.

5.1.1 Menu Structure

PRENEK is a structured menu-driven pre-processor, allows for more flexibility than a single-menu system, and requires minimal knowledge on the part of the user. Using the flowchart of Fig. 5.1, we will describe the function of each menu within the context of the entire menu structure.

There are three principal steps to creating the data in PRENEK to run a NEKTON simulation:

Step 1. Input the equation type that describes the physics of the problem to be solved, together with associated physical parameter values (material properties). In addition, input numerical parameter values (discretization, solver, i/o specification).

Step 2. Input the spectral element mesh that describes the geometry of the problem to be solved.

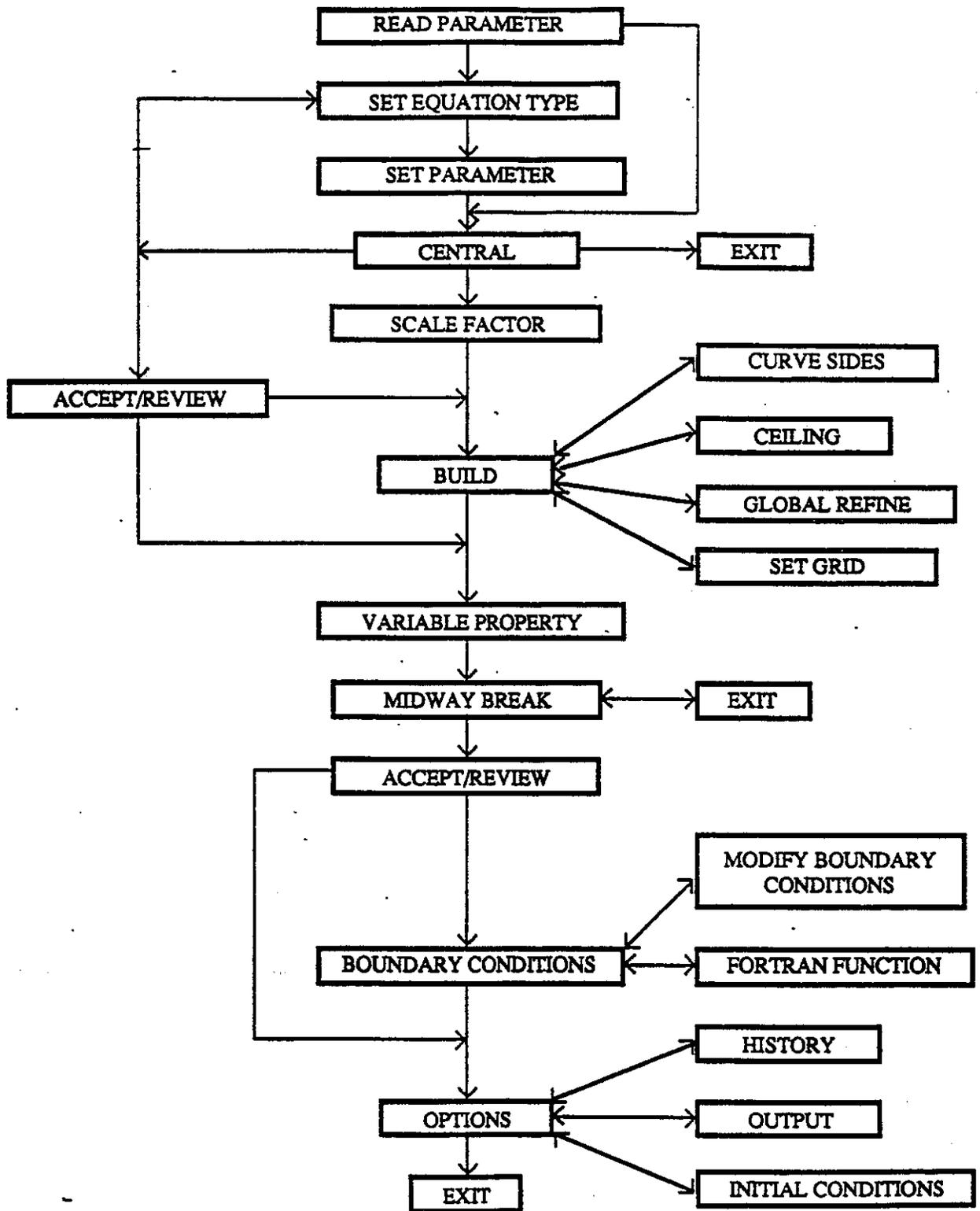


Figure 5.1 PRENEK Menu Structure

Step 3. Input the boundary conditions (plus initial conditions for time-dependent problems).

Additionally, there is a collection of optional specifications which can be applied to any given problem.

In Fig. 5.1, we outline the menu structure. Step 1 of the problem definition process is done immediately upon entering PRENEK. The equation type and parameters can be read in as an old data set using first menu, or they can be input interactively in the SET EQUATION TYPE and SET PARAMETERS menu.

Between Step 1 and Step 2 is the CENTRAL menu. From here there is the option to returning to Step 1, to modify or correct the equation type and/or numerical parameters. There is also the option to continue on to Steps 2 and 3.

Step 2, building the mesh, is done after Step 1 is completed. As in Step 1, it can either be read in from a previously created file or input interactively. If the mesh is read in from an existing file, the boundary conditions and accompanying options are read in as well.

Step 3, setting the boundary conditions, is initiated after the completion of Step 2.

5.1.2 File Structure

PRENEK generates data files that are later used by NEKTON and POST-NEK. Although the user can use the NEKTON package without being aware of what these files contain, problem modification (e.g., parametric studies, sensitivity analysis, etc.) can occasionally be expedited by knowledge of the files' contents. The filenames all begin with the SESSIONNAME of the simulation followed by an extension indicating their purpose:

SESSIONNAME.REA: contains a complete description of the problem as defined by all the user's inputs in PRENEK. It is used in the READ PARAMETERS and BUILD FROM FILE commands.

SESSIONNAME.USER: contains the Fortran programs generated by PRENEK from the "Fortran lines" input by the user in the BOUNDARY CONDITIONS, INITIAL CONDITIONS, VARIABLE PROPERTY, and OPTIONS (DRIVE FORCE) menus.

SESSIONNAME.CMN: contains the Fortran parameter statement that defines the array sizes to be used in the NEKTON solver.

5.2 BUILD Menu

The BUILD menu is used to input the geometry and corresponding spectral element discretization of a problem. The menu begins by prompting the user for scale information to relate the dimensions of the geometry appearing on the terminal screen to those of the actual problem. Once the "plotter to user" coordinate transformation has been established, the user begins to input the geometry spectral element by spectral element (i.e., macro-element by macro-element), following the procedures and rules described in Sections 3.2 and 4.3.

When adding elements, the BUILD menu prompt for each of the four corners of each spectral element:

Corner I>

Corner II>

Corner III>

Corner IV>

Corner I>

Corner II>

Possible user replies to each prompt are

- [M/W -] ! Returns current position of cross hairs (mouse) in user coordinates - used to determine screen locations.
- [M/Y -] ! Inputs the requested corner position as the current location of the mouse - used to input "new" corners.
- [KP/] ! Inputs the x and y coordinates using "mouse keypad".
- [M/B -] ! Inputs the requested corner position as being the same as that of the previously-defined corner nearest the current location of the mouse; used to input corners/elements connected to elements input previously.

Other options in the build menu allow for modification and correction of elemental corners, curving of sides, and global mesh refinement.

5.3 BOUNDARY CONDITION Menu

After the build menu, control is automatically passed to the BOUNDARY CONDITION menu. The user is first asked for velocity boundary conditions (on ∂D_F), followed by temperature boundary conditions (on ∂D_T), and finally for additional passive scalar boundary conditions. In each case, PRENEK prompts the user for the boundary condition on each spectral element face that is part of the physical boundary by "blinking" the element side in question on the Tektronix terminal, or changing the color of the side on the Sun or X workstation. Note : the BOUNDARY CONDITION menu is cognizant of the equation type, and will not ask the user for boundary conditions that are not relevant to the particular problem of interest.

Whenever a numerical value is required (e.g., coordinates, temperature, flux, or heat transfer coefficient), the "mouse keypad" located above the red menu area will become highlighted. This allows the user to input the coordinates or boundary conditions without having to cycle between the mouse/screen and the keyboard. The possible keypad entries are (see Fig. 4.12):

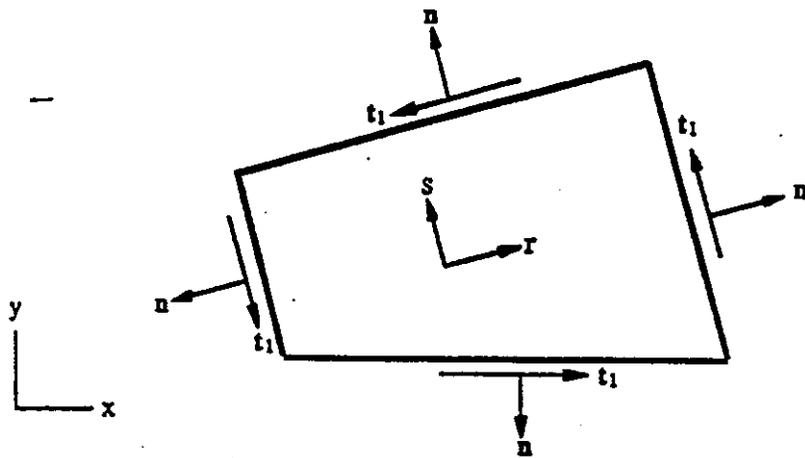
[KP/0,1,2,3,4,5,6,7,8,9] ! numerical values
 [KP/ -] ! negative
 [KP/ D] ! rubout (deletes last keypad entry)
 [KP/ E] ! exponential format
 [KP/ R] ! "carriage return" (terminates entry)
 [KP .] ! decimal point

Boundary conditions can also be specified as a fortran function, in which case a fortran statement is typed in on the keyboard. In order to facilitate imposition of various boundary conditions, a local (t_1, t_2, n) orthonormal coordinate system is defined on all macro-elemental boundaries. The convention used for this local system is illustrated in Fig. 5.2. Note that the normal vector n always points outward of the macro-element and (t_1, t_2, n) forms a right-handed system on all boundary surfaces. For two-dimensional geometry, t_2 is selected as the positive z-direction (solely for the definition of t_1) and the tangent vector t_1 is obtained using : $t_1 = t_2 \times n$. For three-dimensional geometry, t_1 is selected as one of the *two* macro-elemental isoparametric r, s or t coordinates that are tangent to the boundary surface. In choosing one of the tangential coordinates as t_1 , NEKTON uses a convention that a positive r, s or t direction is always selected and the right-handedness of (t_1, t_2, n) is maintained. Finally, the tangent vector t_2 is obtained using : $t_2 = n \times t_1$.

When the user has specified all boundary conditions, PRENEK closes the BOUNDARY CONDITION menu and displays the OPTIONS menu. From here, the user can set initial conditions, define the format and output quantities to be saved, or set a driving force.

5.4 Using the GLOBAL REFINE submenu

Refinement of both two- and three-dimensional meshes share a common set of 4 commands. These are the SPIDER WEB, ZIPPER, PICTURE FRAME, and CORNER FRAME commands. With these commands the user can easily refine a mesh or a portion of a mesh. In this section we illustrate the use of the mesh refin-



2-D: n always points outward; $t_2 \equiv z$; t_1, t_2, n are orthonormal

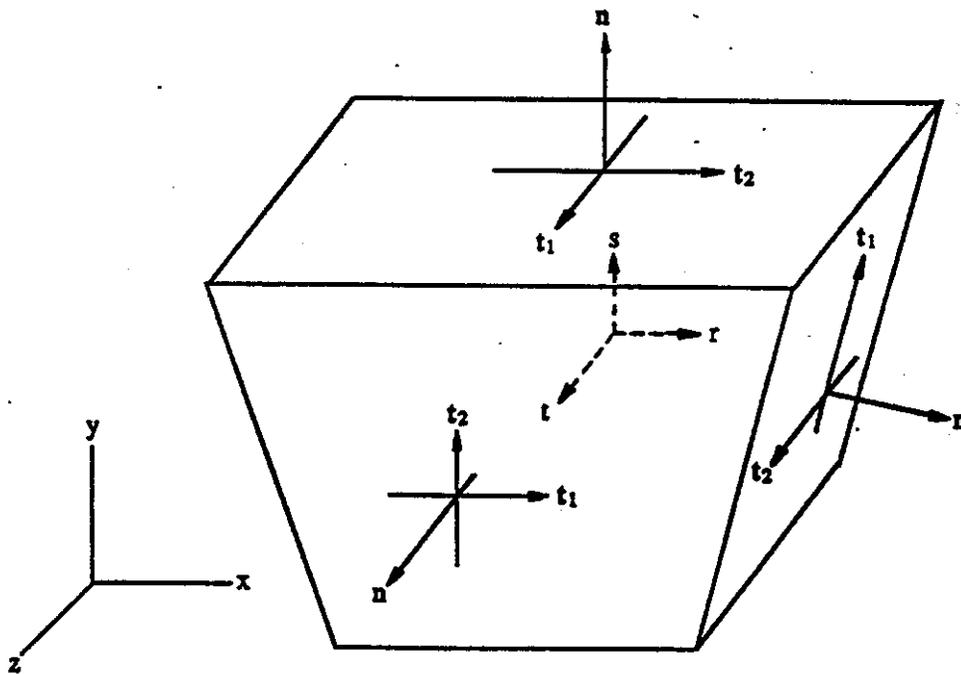


Figure 5.2

3-D: n always points outward; t_1 is selected as one of the positive $r, s,$ and t directions such that t_1, t_2, n form a right-handed system; $t_2 = n \times t_1$

ing commands in the refinement of two-dimensional meshes. In the next section, we show how these commands extend to three dimensional meshes introducing a fifth command, the SPLIT FLOOR, which exclusively applies to three dimensional meshes.

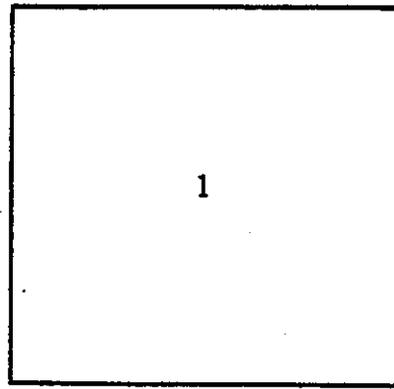
5.4.1 Two-Dimensional Mesh refinement

SPIDER WEB

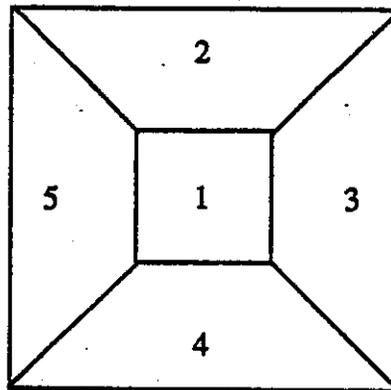
The SPIDER WEB command splits a single element into 5 elements, see Fig. 5.3. A new element is created inside the original quadrilateral element. The first new element is smaller but geometrically similar to the original element. A segment connects each corner of the new inside element to the associated corner in the outside quadrilateral element. The other 4 new elements are defined by the four regions bounded by the inside quadrilateral, the outside quadrilateral and the 4 connecting segments as shown in Fig. 5.3. The SPIDER WEB command prompts the user to enter the element to be split. PRENEK then splits the element into five elements.

CORNER FRAME

The CORNER FRAME command splits a single parent element into three daughter elements as in Fig. 5.4. The parent element should have two adjacent sides which are boundary sides, i.e. sides which do not border on other elements. Invoking CORNER FRAME command results in a prompt for the element to be divided. Next it asks for the inside corner, which is the corner shared by the two adjacent boundary sides. Finally it asks for the shrink factor, which determines the size of the first daughter element. This daughter element is at the inside corner and is geometrically similar to, but smaller than the original element, with the proportion defined by the shrink factor. The other two daughter elements are defined by the boundary of the first daughter element, the remaining segments of the original parent element, and the diagonal line segment connecting them.



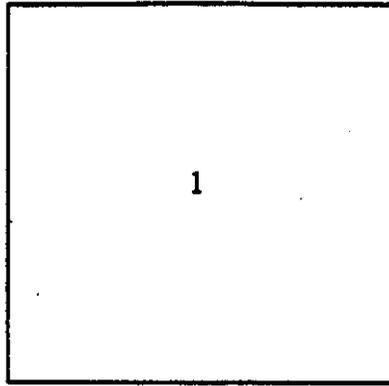
(a)



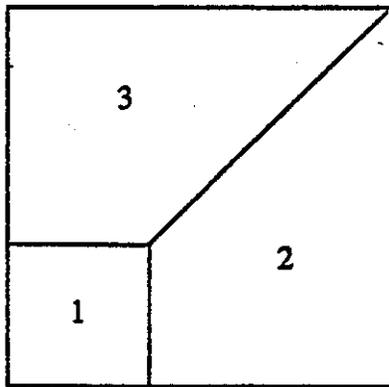
(b)

Figure 5.3 (a) Original single spectral element

(b) Mesh refinement using the SPIDER WEB command



(a)



(b)

Figure 5.4 (a) Original spectral element

(b) Mesh refinement using the CORNER FRAME command

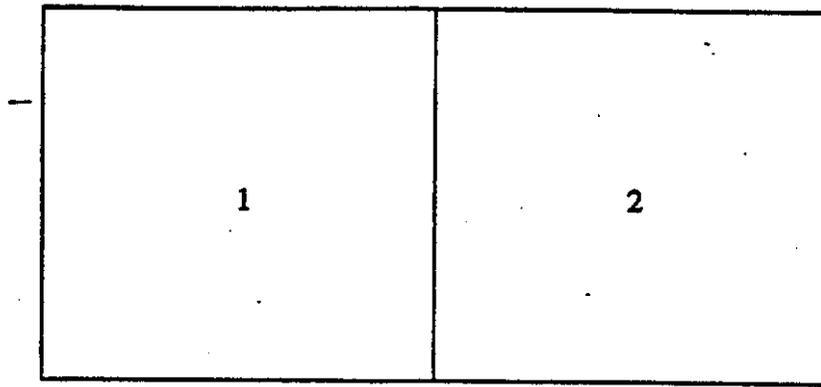
PICTURE FRAME

The PICTURE FRAME command refines the mesh in two adjacent elements simultaneously. Its purpose is to refine the elements near a corner (where finer mesh is usually needed). In Fig. 5.5 there are two adjacent elements in a "picture frame" configuration, that is, one in which two elements share a common overlapping side and in which each of the two elements has a boundary side next to the overlapping side. We refine the mesh near the corner common to both elements. The user first chooses the corner, then the shrink factor for the new elements. PRENEK splits the two elements into six elements.

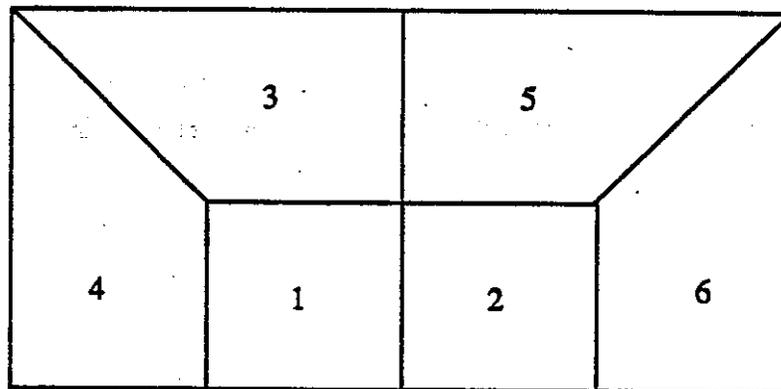
An alternative configuration for using the PICTURE FRAME command is in Fig. 5.6. Refinement of this configuration requires two commands, the PICTURE FRAME command and CORNER FRAME command, both needed to ensure proper elemental connectivity. The first step is to divide two of the three elements near the corner with the PICTURE FRAME command as in Fig. 5.6a. When this is done, PRENEK responds with a warning. This is because the mesh in Fig. 5.6b is illegal; element 3 cannot properly connect to elements 2 and 7. In the second step of this procedure, the CORNER FRAME commands corrects this condition by splitting element 3 in such a way that the resulting daughter elements 3 and 9 connect to elements 2 and 7 respectively, see Fig. 5.6c.

Implementation of this procedure is as follows. First, we invoke the PICTURE FRAME command. The inside corner common to the three elements to be split is chosen with the mouse. We choose a shrink factor of 0.3. Next we invoke the CORNER FRAME command. Responding to the PRENEK prompts, first element 3 is chosen, then the inside corner of element 3 (which touches elements 1 and 2) is chosen. We must choose the same shrink factor (0.3) as we chose in the PICTURE FRAME command to insure proper connectivity.

The result of this procedure is that we have split three parent elements into nine daughter elements. Three of the daughter elements are geometrically similar to (but smaller than) the three parent elements. The procedure can be repeated recursively on the three new elements to give essentially unlimited local resolution by crowding elements near the corner of interest.



(a)



(b)

Figure 5.5 (a) Original two spectral elements

(b) Mesh refinement using the PICTURE FRAME command

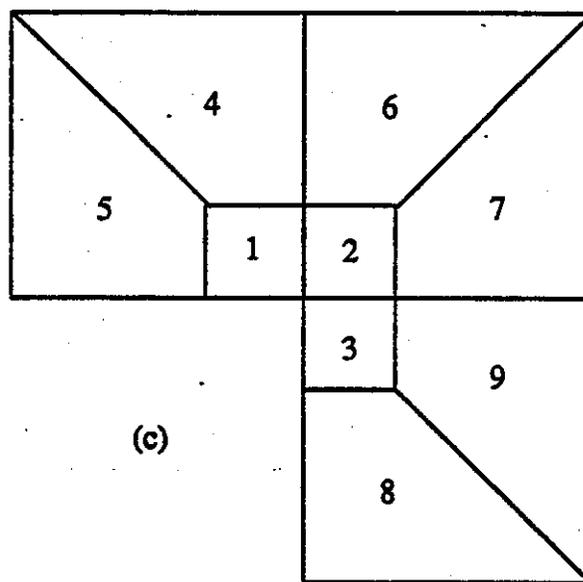
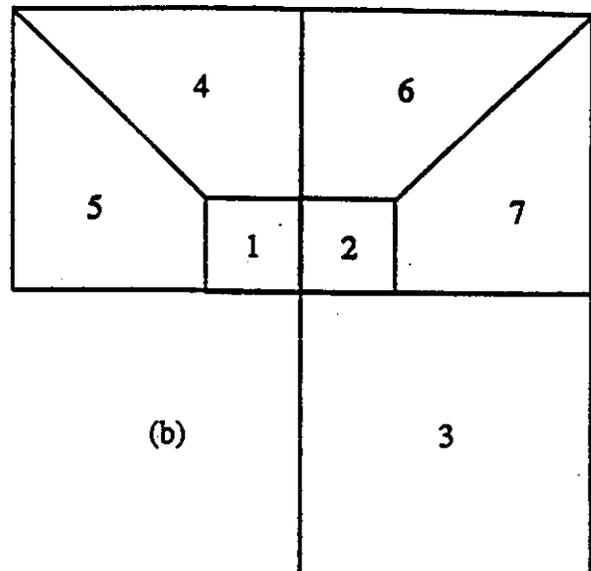
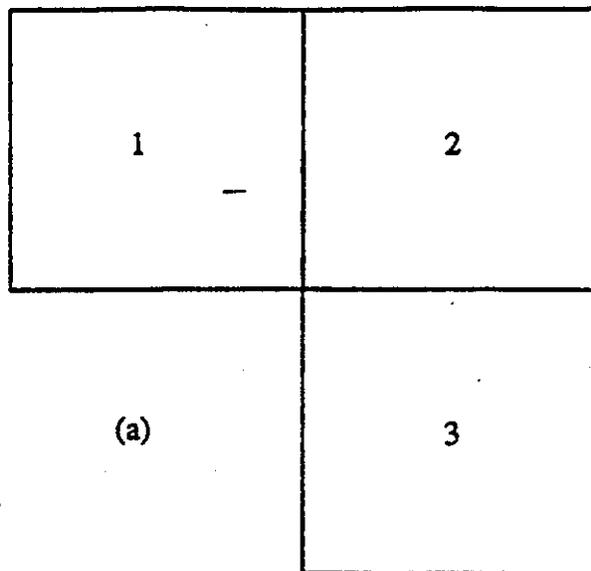


Figure 5.6 (a) Original 3 spectral elements
 (b) Mesh refinement using the PICTURE FRAME command
 (c) Mesh refinement using the CORNER FRAME command

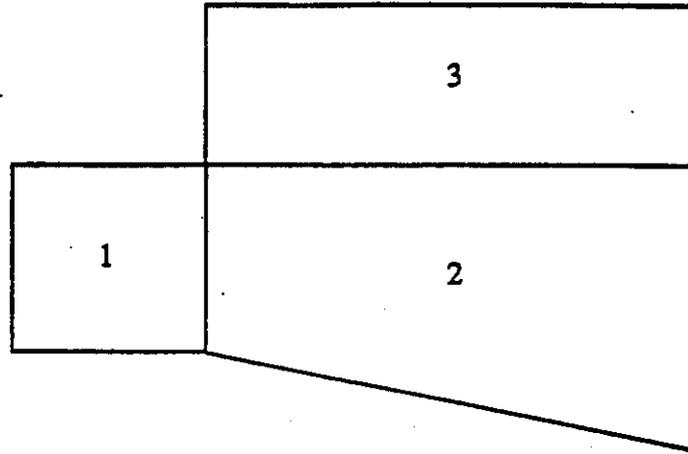
ZIPPER

The ZIPPER command splits one or more elements into two, see Fig. 5.7. The number of elements split depends on the elemental configuration as follows. The continuity requirement for adjacent elemental faces implies that splitting an elemental face requires also splitting the overlapping face of an adjacent element. Thus, a "crack" initiated in one element will propagate throughout the macro-elemental mesh in all directions until it reaches a boundary, i.e., a face which overlaps no other face. Thus, one can quickly enact global mesh refinement by initiating a split in a well-chosen location in a single macro-element.

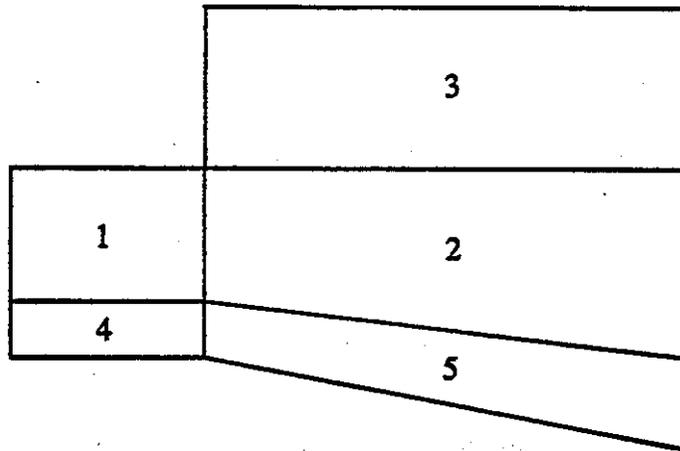
5.4.2 Extensions to Three-dimensional Meshes

All of the commands for two-dimensional mesh refinement readily extend to three-dimensional meshes. When an element in a given level of a three-dimensional mesh is split, the elements on that level are split according to the rules for 2-d meshes. That is, both the floor and ceiling of the three-dimensional element is split. Additionally, the "crack" in the elements propagate to adjacent floors to ensure proper elemental connectivity.

The global modify menu contains one additional capability for the refinement of three dimensional meshes, the SPLIT FLOOR command. According to the rules of construction for three dimensional meshes, the three-dimensional mesh is composed of layers of two-dimensional levels, each of which has a fixed, constant height. Therefore, only one type of refinement is possible in the third (z) direction, which is to split a given floor into two floors. The SPLIT FLOOR command results in a prompt for level>. The user enters the level (or floor, to use the terminology from the apartment building analogy). PRENEK next prompts for the proportion of the split, that is how far up the wall in the chosen level to make the split. The user enters this split factor, where $0.0 < splitfactor < 1.0$. A split factor closer to 1.0 corresponds to a split closer to the ceiling; a split factor closer to 0.0 corresponds to a split closer to the floor.



(a)



(b)

Figure 5.7 (a) Original 3 spectral elements

(b) Mesh refinement using the ZIPPER command applied to element no. 1

5.4.3 Boundary Conditions and Split elements

When an element has been split, the two daughter sides created by the splitting of a parent side take on the boundary conditions of that parent side. Thus, the boundary conditions need not be re-entered after mesh refinement. The single exception to this rule is with sides that have periodic boundary conditions. PRENEK will not attempt to guess which daughter sides are connected periodically. The boundary condition menu will insist that the boundary conditions from these periodic sides be entered by the user.

CURVED Sides

When an element is split, the old sides retain their curvature. Newly created internal sides are straight with the following exception: A newly created side between two sides that are circular arcs will be a circular arc whose radius is the weighted average of the radii of the original circular arcs. This enables easy generation of concentric circular meshes.

5.5 Using the Variable Properties Menu

The VARIABLE PROPERTY menu controls not only the materials properties, but also the volumetric heat source and the fluid/solid domain definition. A more accurate and descriptive name would be "MATERIALS, HEAT SOURCE, OTHER SPATIALLY DEPENDENT QUANTITIES" menu.

The fluid/solid domain definition is controlled by the materials groups. Fluid elements can be in group 0 only. Group 0 is the default group. Elements that are solid elements in a coupled fluid-solid problem are in groups one and above.

All elements in a given materials group have common materials properties (which can depend on space or temperature). The default group number is zero. If no group is specified for an element, it will fall into group zero. When new groups are defined, they must begin with group one and continue through two, three, etc. without skipping any groups.

Once the fluid/solid domain definition has been completed, the user can specify the associated materials properties. Within each group the density, viscosity,

thermal conductivity etc. together with the volumetric heat source can be specified to be either a constant or a fortran function. If the fortran function option is chosen, the properties can be defined as a function of space and other physical quantities.

For multi-fluid systems, NEKTON requires that all spectral elements associated with a fluid component belong to the same unique material property group. For heat transfer with phase change all spectral elements associated with the liquid phase must be assigned to group zero.

5.6 Using the Fortran Function Option

In setting the boundary conditions, material properties, and (volumetric) forcing functions, the user can define these as being function of spatial coordinates, time, and various physical quantities. The keyboard is used to type in a single fortran statement for each condition. In order to ensure appropriate communication between PRENEK and NEKTON, the user must conform to the following convention as regards choice of variable names in these fortran statements:

VARIABLE NAME : Definition

TIME	time
X,Y,Z	x , y , and z coordinates
UX,UY,UZ	velocity, x , y , and z components
UN,U1,U2	velocity, normal and tangential components
TRX,TRY,TRZ	traction, x , y , and z components
TRN,TR1,TR2	traction, normal and tangential components
SI2,SI3	second and third strain-rate invariants
FFX,FFY,FFZ	forcing function, x , y , and z components
TEMP	temperature
FLUX	heat flux
HC	heat transfer coefficient (convection heat transfer)
HRAD	$\mathcal{F} \sigma$, the viewfactor, \mathcal{F} , times the Stefan Boltzmann's constant, σ (radiation heat transfer)

TINF	T_∞ (radiation heat transfer)
QVOL	volumetric heat source
PS1,PS2...	passive scalar number 1, 2, ...
DENSITY	density
RHOCP	ρC_p (density times heat capacity)
VISCOS	μ , fluid viscosity
CONDUCT	k , thermal conductivity

Note that the strain rate invariants are defined as follows:

$$SI2 = e_{xx}e_{yy} + e_{yy}e_{xx} + e_{zz}e_{zz} - e_{xy}^2 - e_{yx}^2 - e_{zz}^2 \quad (5.1)$$

$$SI3 = e_{xx}e_{yy}e_{zz} + 2e_{xy}e_{yz}e_{zx} - e_{zz}e_{yx}^2 - e_{yy}e_{zz}^2 - e_{xx}e_{xy}^2 \quad (5.2)$$

5.7 Special moving boundary features

As described in Ch. 2, the mesh velocity needs to be defined at all external boundaries. For external boundaries which are moving boundaries (moving walls and free surfaces), the mesh is free to deform in both the normal and tangential directions on those boundaries. For external boundary which are not defined as moving boundaries, the mesh velocity normal to the boundary is always zero. Based on the fluid/thermal boundary conditions, NEKTON selects either a zero tangential velocity condition (i.e., fixed mesh), or a zero tangential traction condition (i.e., sliding mesh) on the boundary. The default values for various types of fluid boundaries are shown in the following table. Note that these settings are valid if heat transfer analysis is not included, or if heat transfer analysis is also considered, but no melting fronts (i.e., no change-of-phase) are present.

	FREE	SLIDING	FIXED
AXIS		✓	
FREE SURFACE	✓		
MOVING WALL	✓		
OUTFLOW		✓	
OUTFLOW/N		✓	
PERIODIC	✓		
SHEAR		✓	
SHEAR (LOCAL)		✓	
SYMMETRY		✓	
TRACTION		✓	
TRACTION (LOCAL)		✓	
VELOCITY			✓
VELOCITY (LOCAL)			✓
WALL			✓

For heat transfer problem with change-of-phase, the default values for various boundaries are shown in the following table. Note that in this case, even if the fluid flow problem is considered, the defaults given in the above table are non-applicable *except* for moving wall and free surface, which are always free.

	FREE	SLIDING	FIXED
CONVECTION		✓	
FLUX		✓	
INSULATED		✓	
MELTING FRONT	✓		
OUTFLOW		✓	
PERIODIC	✓		
RADIATION		✓	
TEMP			✓

For special modeling purposes, these defaults can be overridden by the user by editing the SESSION.REA file generated by PRENEK as described in the following. Under "FLUID BOUNDARY CONDITIONS" and "THERMAL BOUNDARY CONDITIONS" are data for the boundary conditions for each face of each macroelement, the format for which is:

FORMAT (1X,A3,2I3,5G14.7)

where the character name (A3) specifies the type of boundary and the two subsequent integers (2I3) indicates the element number and face number, respectively. By placing a "+" character on column 1 before the character name, the sliding mesh condition is enforced; and by placing a "-" character, the fixed mesh condition is enforced. Note that the above options are not applicable to boundaries which are defined as free.

Chapter 6

NEKTON

From the user perspective, there is relatively little to describe concerning the computational engine. NEKTON takes the geometrical, physical, and numerical data from the PRENEK-prepared files, Section 5.1.2, performs the simulation and generates output files that contain the simulation results at the temporal and spatial points specified by the user. The numerical methods used in NEKTON are described in Chapter 3.

6.1 Command Procedures

In order to access NEKTON the user types one of the following commands:

{nek sessionname}

{nek sessionname run}

{nekb sessionname}

{nekb sessionname run}

The first, *{nek sessionname}*, runs the simulation "sessionname" interactively. In order to avoid recompiling the whole code for every run, a set of precompiled codes exists for the most commonly used conditions. These precompiled codes will be used automatically if the user selects his run parameters to one of the following entries:

Formulation	NORDER	NEL	Dimensions
Split	5	≤ 100	2-D
Split	9	≤ 100	2-D
Not Split	5	≤ 100	2-D
Not Split	7	≤ 100	2-D
Not Split	5	≤ 100	3-D

In this case a precompiled code will be linked together with the (compiled) user-defined subroutines in `SESSIONNAME.USER`, see Section 5.1.2. Here, (NORDER-1) is the order (polynomial degree) of each spectral element, while NEL is the number of spectral elements.

The second command, `{nek sessionname run}`, runs NEKTON interactively without any compilation or linking. This command is appropriate to use if the code has already been compiled and linked, and the user only wants to change certain parameters (e.g, the viscosity), or if the user wants to continue an earlier simulation (see RESTART option).

The last two commands, `{nekb sessionname}` and `{nekb sessionname run}`, do the same thing as `{nek sessionname}` and `{nek sessionname run}`, but printout of comments etc. are directed to a log-file instead of to the screen.

Once the simulation has begun, no further input is required on the part of the user.

The code itself, including the precompiled codes listed above, requires about 6 MBytes of memory. The storage (in words) used in a given problem is approximately the number of grid points times 250. The user should therefore have at least 10 MBytes of free disk space before installing and running NEKTON.

6.2 File Structure

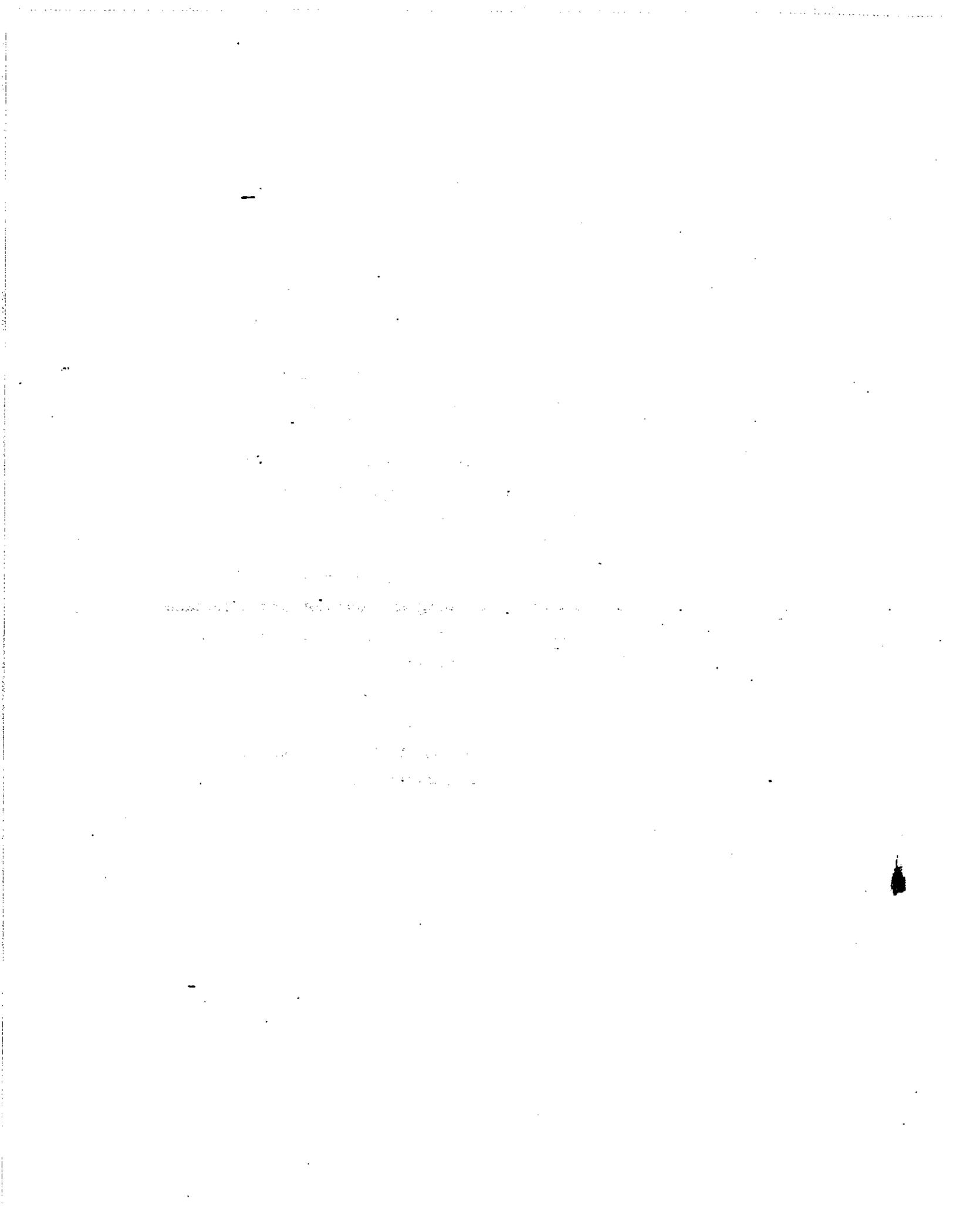
NEKTON generates the following output files for subsequent use by POST-NEK:

SESSIONNAME.HIS contains the history data (continuous in time, discrete in space) requested by the user in the HISTORY menu in PRENEK.

SESSIONNAME.FLD contains the field data (continuous in space, discrete in time) requested by the user via the IOTIME/IOSTEP parameters and the OUTPUT menu in PRENEK. The data in **SESSIONNAME.FLD** is used not only by POSTNEK, but also by PRENEK in generating initial conditions and restarts in the INITCOND menu.

SESSIONNAME.LOG is a log-file generated when invoking NEKTON by a {*nekb ...*} command; in this case printouts from NEKTON to the screen is directed to the log-file.

SESSIONNAME.SCH is a scratch file generated by NEKTON. This file is saved at the end of the program execution and it contains history and integral quantity data for every time step. Before normal termination of NEKTON, data is extracted from this scratch file and put into **SESSIONNAME.HIS** for postprocessing. If abnormal termination occurs, this file can be renamed as **SESSIONNAME.HIS** and accessed by POSTNEK for a display of time-history or integral quantity data up to the time of program termination.



Chapter 7

POSTNEK

This chapter describes the structure of the POSTNEK post-processor and suggested plotting strategies. It briefly describes the function of individual menus and menu items. For a detailed description of each menu and menu item see Chapter 8.

7.1 POSTNEK Menu Structure

Fig. 7.1 shows the easy-to-use POSTNEK menu structure. To make a plot using POSTNEK, the user chooses the PLOT item from the top menu. The user changes the characteristics of the plot by entering the appropriate submenu and altering the value of one of the parameters controlling the plot.

The SET PLOT FORMAT menu determines whether the plot will be the value at a point, along a line, or on a surface. Three-dimensional data is examined by specifying a two-dimensional plane on which to plot. Submenus of the PLOT FORMAT menu determine what type of line or surface plot will be plotted. Available surface plots include arrows for vector fields and color fill or fishnet grid for scalar fields. Line plots are simple x-y plots, with the user having the option of plotting a quantity vs. time or a quantity vs. space. An option for plotting streamlines is also available.

The SET PLOT ATTRIBUTE menu is used to specify details about the format of the plot, such as resolution.

The SET LOCATION menu determines the point, line segment, or surface from which to plot data. All three locations can be simultaneously specified.

*Become
memorize
the outside
each build
the locatio.
them your I
you discover*

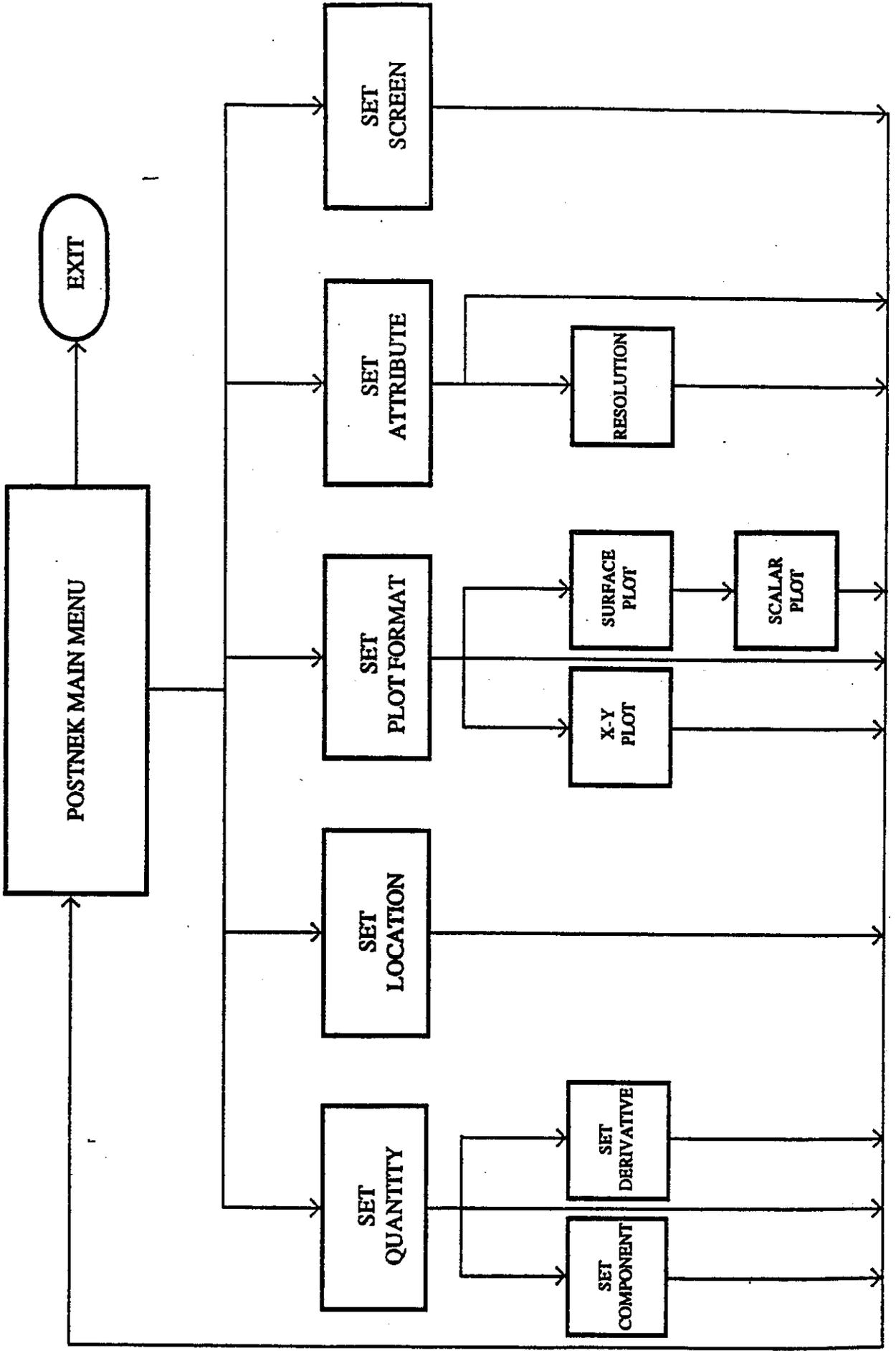


Figure 7.1 POSTNEK Menu Structure

current setting of the PLOT FORMAT will determine which is plotted.

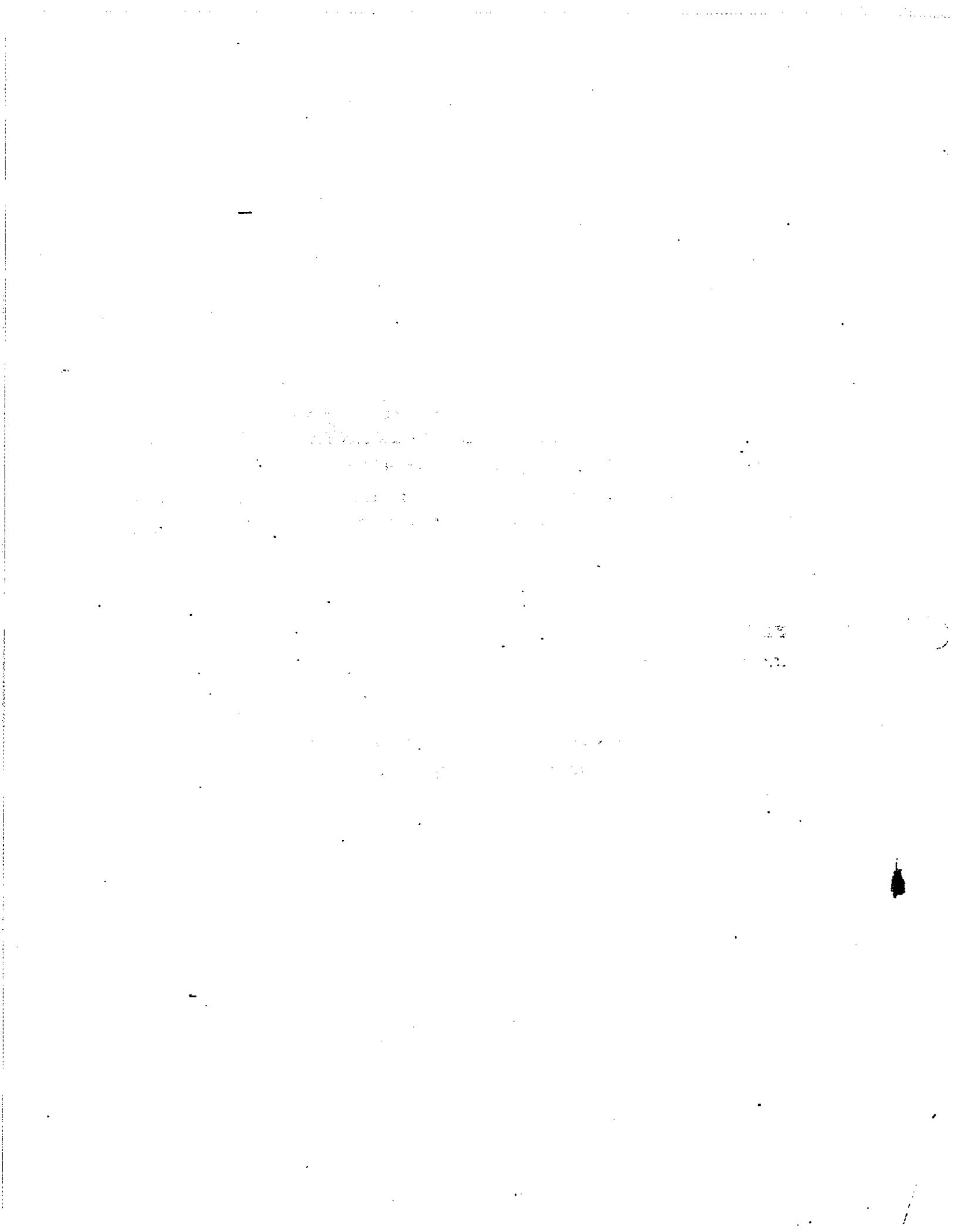
The SET QUANTITY menu determines whether velocity, temperature, etc. will be plotted.

7.2 Hardcopy

All plots generated by the user in a POSTNEK session SESSIONNAME are stored in a file SESSIONNAME.PLT. NEKTON supports the *PostScript* standard and the *QMS Quik* graphics standard. The graphics standard is selected by editing the file NEKTON:DEFAULTS.NEK (in VMS), or in the file `~/nekdefaults` (in UNIX) (the first two lines of `~/nekdefaults` are reserved for the nekton directory and hardware specification), see the Installation Instructions. This file should also contain the line *PRINTER QMS* or *PRINTER POSTSCRIPT*. To obtain a hardcopy of these plots the user types:

```
{PRINT SESSIONNAME.PLT} ! from VMS  
{lpr sessionname.plt}      ! from UNIX
```

For those using a Tektronix terminal, a screen dump can be made by pushing the *SCOPY* key. Tektronix supports a large number of hardcopy units both color and black-and-white.



Chapter 8

The Menu System

In this chapter we describe the menu system that forms the structure from which **PRENEK** and **POSTNEK** operate. This section describes the function of the individual menus, which are arranged alphabetically for **PRENEK** and **POSTNEK**, respectively. Following the menu description, each item from the menu is listed along with its definition and usage.

These menus use a mouse or graphics tablet to input data. The menus are composed of two or more **ITEMS** from which to choose. The items that comprise a given menu can vary with the equation type defined in the beginning of the **PRENEK** session. In this section, all possible items for each menu are described.

When running **PRENEK** or **POSTNEK**, the menu name appears in the top left corner of the screen for reference. The menu itself appears in the screen center or screen right edge. An item is chosen from the menu by pushing the yellow mouse button with the screen cursor inside the menu box corresponding to the desired item.

8.1 PRENEK Menus

ACCEPT/REVIEW Menu

ACCEPT MESH

ACCEPT B.C.'s

REVIEW/MODIFY

Menu function: This menu appears in two places: before inputting mesh and boundary conditions. As the name suggests, it allows the user to review and/or modify data read in from an earlier session, or to accept the data and skip over the modification process.

ACCEPT MESH

Continue in PRENEK without changing the mesh data.

ACCEPT B.C.'s

Continue in PRENEK without changing the boundary conditions.

REVIEW/MODIFY

Modify and/or review the data (mesh or b.c.'s).

BOUNDARY CONDITIONS

(Main Menu for Fluid Boundaries)

- PERIODIC
- WALL
- VELOCITY
- OUTFLOW
- OUTFLOW/N
- VELOCITY (LOCAL)
- SYMMETRY
- MOVING WALL
- MELTING FRONT
- AXIS
- STRESS B.C.s
- SET ENTIRE LEVEL

(Additional Traction B.C. Menu for Fluid Boundaries)

- MAIN B.C. MENU
- SHEAR
- SHEAR (LOCAL)
- FREE SURFACE
- TRACTION
- TRACTION (LOCAL)

(Main Menu for Temperature Boundaries)

- PERIODIC
- INSULATED
- FLUX
- TEMP
- CONVECTION
- OUTFLOW
- RADIATION
- SET ENTIRE LEVEL

Menu function: The BOUNDARY CONDITIONS menu sets boundary conditions for each external boundary sides, i.e., each macro-elemental side which does not overlap another macro-elemental side. All other boundary sides shared by two macro-elements correspond to internal boundaries. For an inhomogeneous medium such as immiscible fluid layers in a multifluid system, or solid-liquid phases in a heat transfer analysis with change-of-phase, special internal boundaries need to be defined; see MODIFY BOUNDARY CONDITIONS menu. Note that the additional traction boundary conditions menu is only applicable if the stress formulation is selected.

PRENEK first checks each side to determine if it is an external boundary side. Each external boundary side for which boundary conditions have not been specified is successively highlighted. The user can then assign a boundary condition to the highlighted side in two ways. The first way is to choose a boundary condition from the menu list provided. The second way is to copy an existing boundary condition from a side whose boundary condition has already been specified. Here, as in the build menu, the blue mouse button acts as a "repeat" button. Pushing the blue button with the cursor near a side with a boundary condition copies that boundary condition to the highlighted side. This is particularly useful when setting boundary conditions that involve typing in a fortran statement or numerical values since the data need only be entered once.

For three-dimensional problems; the boundary conditions are set one level at a time. The side on the floor mesh is highlighted as well as the face on the isometric view and the level on the elevator icon. The order by which boundary conditions are set on a given level is the following: the "wall faces" are set first, followed by the "floor faces", and finally, the "ceiling faces".

Description of Individual Menu Items :

PERIODIC

Periodic boundary conditions are used to mathematically connect two macro-elemental faces that do not physically coincide. The mathematical relationship between two sides that are connected periodically is the same as that between two internal sides

that do coincide. Periodic boundary conditions can be used, for example, to connect the outflow of a problem to its inflow. The resultant simulation corresponds to a physical geometry and flow solution that repeats itself infinitely (or can be approximated as such) and therefore requires simulation of only one periodicity unit.

Setting two elemental sides to be periodic imposes the condition on the solution that all specified quantities ($u_x, u_y, u_z, p, T, \dots$) are *equal* along the entire periodic sides of the two elements, (e.g., $u_x(x, y, t) = u_x(x + L_p, y, t)$ for all times t , where the sides at (x, y) and $(x + L_p, y)$ are specified as periodic). This makes sense only if the sides have equal lengths, the same curvature and the same orientation. It is the responsibility of the user to choose sets of periodic sides that satisfy these imposed restraints.

WALL

The WALL boundary condition on the fluid imposes the boundary condition that all velocity components on the boundary are zero (i.e., the no-slip condition).

VELOCITY

The VELOCITY boundary condition allows the user to set each velocity component (with respect to the Cartesian X-Y-Z system) on the boundary equal to a constant or user-specified fortran function.

OUTFLOW

The OUTFLOW boundary condition is one of the two available models that can be used to simulate the fluid leaving the computational domain; the other one being OUTFLOW/N (not available to temperature and passive scalars) which is described below.

For the OUTFLOW boundary condition, the assumptions used differ for different formulations and solution strategies. In the descriptions given below, u_n and u_t are the normal and tangential velocities, respectively, at the boundary; and $\partial/\partial n$ and $\partial/\partial t$ denote spatial derivative in the normal and tangential directions, respectively.

For the Uzawa/nonstress formulation, the boundary conditions are

$$-p + \mu \frac{\partial u_n}{\partial n} = 0 \quad (8.1)$$

$$\frac{\partial u_t}{\partial n} = 0 \quad (8.2)$$

For the Uzawa/stress formulation, the boundary conditions are

$$-p + 2\mu \frac{\partial u_n}{\partial n} = 0 \quad (8.3)$$

$$\frac{\partial u_t}{\partial n} + \frac{\partial u_n}{\partial t} = 0 \quad (8.4)$$

For the split formulation, the boundary conditions are

$$p = 0 \quad (8.5)$$

$$\frac{\partial u_t}{\partial n} = 0 \quad (8.6)$$

For temperature and passive scalars the boundary condition is

$$\frac{\partial T}{\partial n} = 0 \quad (8.7)$$

OUTFLOW/N

The OUTFLOW/N boundary condition is one of the two available models that can be used to simulate the fluid leaving the computational domain; the other one being OUTFLOW which is described above. The use of the OUTFLOW/N boundary conditions is restricted for nonstress formulation to boundaries which are both straight and aligned with either the Cartesian X, Y or Z axes. In addition, OUTFLOW/N is not applicable to temperature and other passive scalars.

For the OUTFLOW/N boundary condition, the assumptions used differ for different formulations and solution strategies. In the descriptions given below, u_n and u_t are the normal and tangential velocities, respectively, at the boundary; and $\partial/\partial n$ and $\partial/\partial t$ denote spatial derivative in the normal and tangential directions, respectively.

For the Uzawa/nonstress formulation, the conditions are

$$-p + \mu \frac{\partial u_n}{\partial n} = 0 \quad (8.8)$$

$$u_t = 0 \quad (8.9)$$

For the Uzawa/stress formulation, the conditions are

$$-p + 2\mu \frac{\partial u_n}{\partial n} = T_n \quad (8.10)$$

$$u_t = 0 \quad (8.11)$$

where T_n is either a constant or a fortran function not involving spatial coordinates.

For the split formulation, the boundary conditions are

$$p = 0 \quad (8.12)$$

$$u_t = 0 \quad (8.13)$$

VELOCITY (LOCAL)

The VELOCITY (LOCAL) boundary condition allows the user to set the velocity components on the boundary with respect to a local $n - t_1 - t_2$ coordinate system equal to constant or user-specified fortran functions. The definition of the local system is given in Fig. 5.2.

SYMMETRY

The SYMMETRY boundary conditions set the normal velocity and the shear traction on the boundary to zero. For nonstress formulation, the use of the SYMMETRY conditions is restricted to boundaries which are both straight and aligned with either the Cartesian X, Y or Z axes.

MOVING WALL

The MOVING WALL boundary conditions imposes at the boundary the no-slip condition that all fluid velocity components at the wall are equal to the corresponding wall velocity components which are prescribed fortran functions. Note that the presence of MOVING WALL boundaries automatically sets the computational mesh to be time-dependent, and therefore the method of characteristics can not be used with this option.

MELTING FRONT

The MELTING FRONT boundary corresponds to solid-liquid interfaces in a heat transfer analysis with fluid flow and change-of-phase. Since the density of the solid

and the liquid must be *equal* in such analysis (Stefan flow is not allowed), the MELTING FRONT boundary condition on the fluid imposes the no-slip boundary condition that all velocity components on the boundary are zero. Note that the presence of MELTING FRONT boundaries automatically sets the computational mesh to be time-dependent.

AXIS

For problems in axisymmetric geometry, AXIS specifies that the boundary lies on the axis of symmetry, (i.e. at $y = 0$). Note that under almost all circumstances, elemental boundaries which lie entirely on the axis of symmetry should be given the AXIS boundary condition.

STRESS B.C.s

For analysis using the stress formulation, additional traction-type boundary conditions can be imposed. These are given in a separate menu and are activated by selecting the STRESS B.C.s option.

SET ENTIRE LEVEL

SET ENTIRE LEVEL sets all remaining unspecified boundary conditions on the current level to a given boundary condition using a single command. SET ENTIRE LEVEL results in a prompt for the type of boundary condition to which to be specified for the entire remaining level. The user responds with a choice from the menu or by choosing to copy an existing boundary condition using the blue ("repeat") button.

MAIN B.C. MENU

Return to the main fluid boundary conditions menu.

SHEAR

The SHEAR boundary conditions set the normal velocity on the boundary to zero. In addition, nonzero shear traction components defined with respect to the Cartesian X, Y, Z system are also allowed which can either be constant or fortran functions. Note that since SHEAR boundaries are, in general, either curved or not aligned with

the Cartesian axes, the user is responsible for entering the correct Cartesian shear traction components such that the resultant traction vector is tangent to the boundary.

SHEAR/LOCAL

The SHEAR/LOCAL boundary conditions is analogous to the SHEAR boundary conditions in setting the normal velocity on the boundary to zero. However, for SHEAR/LOCAL boundaries the shear traction components are defined with respect to a local $n - t_1 - t_2$ coordinate system defined in Fig. 5.2. Thus, for SHEAR/LOCAL boundaries, the input shear traction components are automatically assumed to be in the tangential $t_1 - t_2$ directions.

FREE SURFACE

The FREE SURFACE boundary conditions is applicable for liquid-gas interfaces which are time-dependent and free to move in two- or three-dimensional space. At FREE SURFACE boundaries, surface-tension coefficients are defined which can either be constant or fortran functions of other parameters, including temperature to simulate Marangoni effects. In addition to the traction resulted from surface-tension/curvature products, normal and shear traction components which are either constant or fortran functions can also be applied to simulate nonzero ambient pressure and additional shear effects, respectively; these components are defined with respect to a local $n - t_1 - t_2$ coordinate system defined in Fig. 5.2. Note that the presence of FREE SURFACE boundaries automatically sets the computational mesh to be time-dependent.

TRACTION

The TRACTION boundary condition allows the user to set each traction component (with respect to the Cartesian X-Y-Z system) on the boundary equal to a constant or user-specified fortran function.

TRACTION (LOCAL)

The TRACTION (LOCAL) boundary condition allows the user to set the traction components on the boundary with respect to a local $n - t_1 - t_2$ coordinate system

equal to constant or user-specified fortran functions. The definition of the local system is given in Fig. 5.2.

INSULATED

The INSULATED boundary condition corresponds to zero flux normal to the boundary.

FLUX

The FLUX boundary condition corresponds to user-specified flux normal to the boundary; the flux can either be constant or a fortran function.

TEMP

The TEMP boundary condition corresponds to user-specified temperature at the boundary; the temperature can either be constant or a fortran function.

CONVECTION

The CONVECTION boundary condition corresponds to user-specified convection coefficient and environmental temperature (at infinity), which can both be constant or fortran functions.

RADIATION

The RADIATION boundary condition corresponds to user-specified Stefan-Boltzmann constant/view factor product and environmental temperature (at infinity), which can both be constant or fortran functions.

BUILD Menu

ADD ELEMENT
MODIFY ELEMENT
GLOBAL REFINE
UP LEVEL
DOWN LEVEL
CURVE SIDES
DELETE ELEMENT
CEILING
REDRAW ISOMETRIC
REDRAW MESH
SET GRID
END ELEMENTS

Menu function: The build menu is used to construct a two- or three- dimensional mesh. This mesh defines spectral elements, i.e., macro-elements, which in turn describe the geometry of the problem being simulated. The mesh definition is interactive, enabling the user edit the mesh to correct mistakes, move elements, or refine the mesh.

Element corner points can be entered in two ways. The first, and most convenient, method is to use the mouse. Hitting the yellow mouse button signifies input of a corner which will be located at the nearest intersection of the "graph paper" screen grid lines. Hitting the blue mouse button signifies input of a corner that coincides with the nearest corner of a pre-existing element. This "latching" onto corners ensures that elements meant to be connected will, in fact, be so connected.

The second method of entering element corners is to enter the numerical coordinates with the screen keypad. This is used where the point to be entered is not at a screen grid intersection. To switch modes from mouse input to keypad input, hit the mouse button with the cursor off the mesh input area, (i.e., in the red menu area). The border of the keypad becomes highlighted, signifying that the keypad is

waiting for input. The user then keys in the coordinates using the mouse.

If, in response to the menu prompt, a point is entered outside the menu area (in the mesh area), the menu assumes that ADD ELEMENT was chosen and that the first point of that element is the current cursor position. This feature is meant to avoid the necessity of returning to the menu each time a new element is being created when many elements are to be created.

ADD ELEMENT

The ADD ELEMENT item is used to create a new element. A new element is defined by four corners input in a *COUNTER-CLOCKWISE* order. These define a quadrilateral which is the basic spectral element. For three-dimensional elements, the element is an orthogonal quadrilateral prism; the quadrilateral "base" defined with the mouse is given vertical sides and a quadrilateral top that has the same x,y coordinates as the base but a higher constant value of z

MODIFY ELEMENT

The MODIFY ELEMENT item modifies one or more elements by moving one corner. If more than one element share a corner with the same coordinates, all of those corners are moved. This includes elements on different levels of a three-dimensional mesh that share a corner, (e.g., the "ceiling" of element 1A will be modified with the "floor" of element 2A). If in a three-dimensional mesh the modified element is on the top level, (i.e., no elements currently exist on higher levels), then the element's ceiling will be modified along with its floor.

GLOBAL REFINE

The GLOBAL REFINE item invokes a submenu that can split an element or group of elements. See the GLOBAL REFINE menu description for more detail.

UP LEVEL

Applicable for 3-d meshes only. The UP LEVEL item moves the mesh generator (and the elevator icon) up a level in the mesh. If it is a new level, (i.e., if no elements yet exist on this level) it prompts for the height of the new level. If elements already

exist on this level, it draws the mesh representing the "floors" of the elements on the level.

DOWN LEVEL

Applicable for 3-d meshes only. The DOWN LEVEL item moves the mesh generator (and the elevator icon) down a level in the mesh. It then draws the mesh representing the "floors" of the elements on that level.

CURVE SIDES

Invokes a submenu in order to distort straight edges into curved sides. See the CURVED SIDES menu description for more details.

DELETE ELEMENT

Deletes an element from the current floor.

CEILING

Applicable for 3-d meshes only. Moves the mesh generator to the ceiling of the current floor. This is used only when one wants to alter the ceiling independently of the floor. It invokes a submenu which allows the user to curve or modify the edges of an element's ceiling. See the CEILING submenu for more detail.

REDRAW ISOMETRIC

Applicable for 3-d meshes only. The isometric can sometimes require a complete redraw to restore its accurate representation of the mesh after certain modifications or deletions of elements. Since this can be a time-consuming process, a complete redraw is done only at the specific request of the user via the REDRAW ISOMETRIC item.

REDRAW MESH

Redraws the mesh to restore a "clean" and accurate representation of the mesh.

SET GRID

Modify or eliminate the background grid used for latching the mouse input. Pro-

vides a means of changing grid resolution, and supports a polar grid for building elements with curved sides quickly. See the SET GRID submenu for more detail.

END ELEMENTS

This menu item is chosen when all modifications, deletions, curvature, and refinements are completed. Upon input of END ELEMENTS, PRENEK moves out of the BUILD menu and into the VARIABLE PROPERTY menu.

CEILING Menu

OFF CEILING

MODIFY ELEMENT

CURVE SIDES

Menu function: Alter (i.e., **MODIFY** or **CURVE**) the ceiling of a three-dimensional element independently of the floor. The display of the ceiling menu implies that the mesh generator is operating on the ceiling of the current floor, and that any modifications will affect the ceiling *only*. If there is an element on the next level immediately above the ceiling, **PRENEK** will, of course, modify the floor of that element to retain connectivity.

OFF CEILING

Exit the **CEILING** menu, return to the **BUILD** menu, and change the mesh generator to the floor.

MODIFY ELEMENT

Invoke the **MODIFY ELEMENT** command with changes affecting the top (ceiling) faces of elements on the current floor. See the **MODIFY ELEMENT** item in the **BUILD** menu description for more detail.

CURVE SIDES

Invoke the **CURVE SIDES** menu.

CENTRAL Menu

ALTER PARAMETERS

SHOW PARAMETERS

BUILD INTERACTIVELY

BUILD FROM FILE

IMPORT UNIVERSAL FILE

ABORT (SAVING PARAMETERS)

Menu function: This menu is the central branch point of PRENEK. It appears after the parameters are set and before the mesh is built. From this menu, the user can return to the parameter and equation menus to modify parameters and/or equation types, or can proceed ahead to build a mesh either from "scratch" or starting with a previously built mesh stored in an existing file.

ALTER PARAMETERS

Causes PRENEK to return to the **SET EQUATION TYPE** menu, which is followed by the **SET PARAMETER** menu. The default values of the equation type and parameters are the current values, (i.e., those set in the current session either by reading from an existing file or input by the user). Thus, modification of one parameter consists of typing in the new value of the parameter when its menu box is highlighted and hitting a carriage return for each of the other parameters.

SHOW PARAMETERS

Shows the numerical values of the parameters.

BUILD INTERACTIVELY

Sends the user to the menus related to building a mesh. The mesh will be built interactively from "scratch" from user inputs.

BUILD FROM FILE

This menu item also sends PRENEK to the menus related to building a mesh. Before building the mesh, PRENEK will prompt for the name of the session containing

the desired mesh. The mesh extracted by PRENEK from the session specified will serve as a template (i.e., departure point) for building the new mesh. Note: The name of the session containing the template must be *DIFFERENT* from the current session to avoid confusion and file access problems. It is therefore recommended that names of different sessions be distinct.

IMPORT UNIVERSAL FILE

Use mesh generated with a different mesh generator (preprocessor). PRENEK is currently compatible with IDEAS.

ABORT (SAVING PARAMETERS)

This command enables the user to exit PRENEK at this point and save the parameters and equation type settings. These can be read in the READ PARAMETERS menu during any future PRENEK session.

CHOOSE ELEMENT Menu

UP LEVEL

DOWN LEVEL

CHOOSE ELEMENT

MAIN MENU

Menu function: Enables the user to define conduction elements in *three-dimensional* meshes with multiple levels.

UP LEVEL

Move the material group specification up one level and draw the mesh for that level.

DOWN LEVEL

Move the material group specification down one level and draw the mesh for that level.

CHOOSE ELEMENT

Specify an element for the currently-selected material group.

MAIN MENU

Return to the **VARIABLE PROPERTY** menu.

CURVE SIDES Menu

BUILD MENU

STRAIGHTEN CIRCLE

MAKE CIRCLE

MAKE SPLINE

MAKE SINE WAVE

Menu function: This menu individually modifies the sides of quadrilateral elements originally entered in the build menu. It thereby makes the elements conform to curved geometries. Unlike the **MODIFY ELEMENT** command, this menu distorts an element's sides and not its corners. Thus, the endpoints of a side, (i.e., corners of an element) are not moved in the **CURVE SIDES MENU**. Elemental continuity is maintained throughout all curving operations. For three-dimensional geometries, the quadrilateral elemental faces (top or bottom) are modified at the level currently set in the **BUILD** menu.

BUILD MENU

Returns the control to the **BUILD** menu.

STRAIGHTEN CIRCLE

Changes a curved side to a straight side.

MAKE CIRCLE

MAKE CIRCLE first prompts for the element side and radius. It changes the shape of the side to a circular arc while retaining the locations of the side's two endpoints. The two fixed endpoints, along with the radius, completely define the circular arc. By convention, we define a positive radius corresponding to a convex arc and a negative radius corresponding to a concave arc. Where the side is shared by two elements, this convention is referenced to the element that is marked by an asterisk (*).

MAKE SPLINE

MAKE SPLINE prompts for the element side and for two auxiliary points. It changes the shape of the side to a cubic polynomial (i.e., cardinal spline). The polynomial has endpoints which remain fixed at the location of the side endpoints. Specifying the slopes at these two endpoints completes the specification of the cubic polynomial. The slope at each endpoint is set to be equal to that of the line joining the other endpoint and the respective auxiliary point.

MAKE SINE WAVE

Not yet implemented.

FORTRAN FUNCTION Menu

CONSTANT

FORTRAN FUNCTION

Menu function: This menu appears after a boundary condition is chosen, and determines the mode in which it is specified.

CONSTANT

Prompts the user for input of a constant value using the keypad.

FORTRAN FUNCTION

Prompts the user for input of a one-line fortran statement to specify the boundary condition.

GLOBAL REFINE Menu

END GLOBAL REFINE

ZIPPER

SPIDER WEB

PICTURE FRAME

CORNER FRAME

SPLIT FLOOR

Menu function: Allows the user to split one or more elements in order to refine a mesh. It can greatly reduce the amount of time required by a user to enter a mesh. See also Section 5.4

END GLOBAL REFINE

Returns to BUILD menu.

ZIPPER

Initiate a crack in a specified element that propagates to adjacent elements and beyond, see Fig. 5.7.

SPIDER WEB

Split one element into five elements, see Fig. 5.3. An element geometrically similar to the original element, but smaller, and is created in the center of the original element. Four new sides connect the corners of the original element and the new element. These four sides, the sides of the original element, and the sides of the center element define the other four newly created elements.

PICTURE FRAME

Split two elements into six elements, see Fig. 5.5. One call to PICTURE FRAME is equivalent to two calls to CORNER FRAME, Fig. 5.4, but requires less user input.

CORNER FRAME

Split one element into three elements, see Fig. 5.4.

SPLIT FLOOR

Splits all elements on the current floor with a plane normal to the z axis.

HISTORY Menu

MAIN MENU
ENTER POINT
TEMPERATURE
VELOCITY
PRESSURE

Menu function: Specify spatial coordinates and variables to save at each time step. Note that spatial locations are restricted to collocation points. NEKTON will save data from the collocation point nearest to coordinates specified.

MAIN MENU

Return to **OPTIONS** menu

ENTER POINT

Specify a new history point that has the variables currently displayed in the menu:

TEMPERATURE
VELOCITY
PRESSURE

Toggle on or off the variables to be included in the next **ENTER POINT** command.

INITIAL CONDITION Menu

MAIN MENU

DEFAULT=0

FUNCTION

PRESOLV

RESTART

Menu function: Set (optional) initial conditions. The default is zero initial condition for each field variable.

MAIN MENU

Return to **OPTIONS** menu.

DEFAULT=0

Default (zero) initial conditions.

FUNCTION

Specify initial condition for a field variable as a fortran function.

PRESOLV

For a heat transfer problem this option gives the steady conduction solution as an initial condition for the temperature. For a flow problem this option gives the steady Stokes solution as an initial condition for the velocity field *as long as* the classical splitting scheme is not used.

RESTART

The user can here read in the results from an earlier simulation as initial conditions by giving the name of the previous session. If no additional information is given, the program will restart all field variables (temperature, velocity, geometry) from the last complete dump in the restart file. The user can also request restart from selected field variables, and from more than one restart file.

MODIFY BOUNDARY CONDITIONS Menu

MODIFY BC

SHOW BC

INTERNAL BC

END LEVEL

Menu function: This menu appears after the **BOUNDARY CONDITIONS** menu. It allows the user to change, review, or accept the boundary conditions generated for the current level. In addition, special internal boundaries can also be defined.

MODIFY BC

Change previously input boundary conditions.

SHOW BC

Display previously input boundary conditions.

INTERNAL BC

This option is applicable only with the stress formulation. In a multi-fluid system, immiscible fluid layers are defined by specifying the interfaces between different fluids as internal boundaries. In a heat transfer analysis with change-of-phase but without fluid flow, all freezing/melting fronts are defined as internal boundaries. Note that in a heat transfer analysis with fluid flow, the solid-liquid interfaces are *automatically* defined when the mesh is input. For multi-fluid systems, NEKTON requires that all spectral elements associated with a fluid component belong to the same unique material property group. For heat transfer with phase change all spectral elements associated with the liquid phase must be assigned to group zero.

END LEVEL

Accept all boundary condition data generated for the current level.

MIDWAY BREAK Menu

SET BCs

ABORT

Menu function: This menu appears between the **BUILD** menu and the **BOUND** menu. It allows the user to exit PRENEK and save the mesh geometry without inputting boundary conditions. It is at roughly the halfway point in PRENEK and a convenient place to temporarily stop work to be continued later. Thus, it is called the **MIDWAY BREAK** menu.

SET BCs

Continue in PRENEK to the next menu, the **BOUNDARY CONDITION** menu.

ABORT

Exit PRENEK, saving the equation type, parameter, and mesh data.

OBJECT Menu

MAIN MENU

DELETE OBJECT

MODIFY OBJECT

ADD OBJECT

Menu function: Allows the user to add, delete or modify an object. An object is a surface defined as a contiguous set of edges/faces of already defined 2d/3d spectral elements. Currently objects are used *only* in conjunction with integral quantities. The INTEGRAL QUANTITY in the OPTIONS menu should be entered after the objects are defined.

MAIN MENU

Return to OPTIONS menu.

DELETE OBJECT

Delete an object.

MODIFY OBJECT

Modify an object (surface) by adding or deleting edges/faces. See the MODIFY OBJECT submenu.

ADD OBJECT

Assign a name to the new object, and go to to the MODIFY OBJECT submenu.

MODIFY OBJECT Menu

END OBJECT

ADD FACE

DELETE FACE

Menu function: Modify an object by adding or deleting edges/faces to the list defining the surface.

END OBJECT

Return to the **OBJECT** menu.

ADD FACE

Add a face (or edge) to the list defining the object.

DELETE FACE

Delete a face (or edge) from the list defining the object.

OPTIONS Menu

EXIT
OUTPUT
DRIVE FORCE
HISTORY
INITIAL COND
INTEGRAL QUANTITY
OBJECT

Menu function: This menu contains a collection of optional features within NEKTON such as specification of output data, initial conditions, and driving force.

EXIT

Exit PRENEK, saving the current problem definition.

OUTPUT

Specify which variables (velocity, temperature, pressure, coordinates...) are to be dumped at the time or step intervals specified by the parameters IOTIME or IOSTEP, respectively. The default is all variables relevant to the problem type.

DRIVE FORCE

Specify a volumetric driving force for a fluid problem (e.g. constant pressure gradient or fortran function).

HISTORY

Specify one or more points in space at which to save output data continuously (at each time step) in time. Output data can be x , y , and z components of velocity and spatial coordinates, pressure, temperature, and any *one* passive scalar, and are selected by the user in the SET QUANTITY menu. Note that in a fluid flow/moving geometry problem, if the user wants to compute the time history of the velocity and coordinates at the same physical location, the current version of

NEKTON requires that two history points be defined, one for the velocity and one for the coordinates.

INITIAL COND

Specify initial conditions as default, fortran function, steady linear solution (pre-solv), or as data from a previous simulation (restart). The default values are zero temperature and velocity.

INTEGRAL QUANTITY

Specify an integral quantity. An integral goes over the surface of an object defined by the user in the **OBJECT** menu. The integrals are computed at every time step, and are part of the time history response in NEKTON. After invoking this menu item, a list of the names of the previously defined objects are displayed, and the user can select objects for which integral quantities are to be computed. In a fluid flow problem integral quantities are the x , y , z components of the total force exerted by the fluid on the entire surface of the object. In a heat transfer analysis the integral quantity is the heat transfer rate across the entire surface of the object. NEKTON automatically computes the applicable quantities based on the equation type.

OBJECT

Add, delete or modify an object, see the **OBJECT** menu for more details.

OUTPUT Menu

MAIN MENU
COORDINATES
VELOCITY
TEMPERATURE
PASSIVE SCALAR

Menu function: One form of data saving is to dump entire fields of data at a user specified time. This is controlled by the output menu. Data variables to be included in field dumps can be specified by toggling them on or off via the menu. For example, the item "coordinates yes" appears on every menu the first time it is invoked. The meaning is that the quantity "coordinates" will be dumped. If the user selects that item with the mouse, the item will toggle to "coordinates no", meaning that the status has changed so that currently the coordinates will *not* be dumped. When the field specifications are satisfactory, the MAIN MENU item returns to the OPTIONS menu. Depending on the type of simulation, other quantities will be available for selection. If there are passive scalars in the simulation, the user can choose to save none of the quantities, or save a single passive scalar quantity. When satisfied with the status of the output variables, the user selects MAIN MENU to exit the OUTPUT menu.

MAIN MENU

Accept current switch settings and return to OPTIONS menu.

COORDINATES
VELOCITY
TEMPERATURE

Toggle specification of output fields on or off.

PASSIVE SCALAR

Select the number of the passive scalar field to save. Due to storage restrictions, no more than three passive scalar fields can be saved.

READ PARAMETER Menu

TYPE IN NEW PATRAMETERS

READ PREVIOUS PARAMETERS

Menu function: The **READ PARAMETER** menu is the first menu encountered upon entering PRENEK. From this menu the user decides whether to read the equation type and its associated parameters from an existing file or to interactively enter new values.

TYPE IN NEW PATRAMETERS

Proceed to the **SET EQUATION** in order to set the equation type type with associated parameters.

READ PREVIOUS PARAMETERS

Read in the equation type and numerical parameters from a previous session (the user is prompted for the name of the previous session). This menu item causes control to skip the **SET EQUATION** and **SET PARAMETER** menus and go directly to the **CENTRAL** menu. If modification of the parameters is desired, the user can return to the **SET EQUATION** and **SET PARAMETERS** menus from the **CENTRAL** menu.

SCALE FACTOR Menu

MOUSE

TYPE

USE NORMALIZED

Menu function: This menu appears immediately after the BUILD INTERACTIVELY command is specified. It determines how the scale factors are to be defined. The scale factors define the mapping between screen coordinates and the coordinates associated with the geometry of the problem to be simulated.

MOUSE

The mouse will be used to define distances on the screen which correspond to distances in the physical problem, see remark below.

TYPE

The scale factors will be typed in directly, see remark below.

USE NORMALIZED

Normalized coordinates will be used, (i.e., the origin will be located at the center of the mesh grid portion of the screen, with the domain in x and y between -1 and 1).

Remark: The scale factors in x and y can be different. However, care must be used in interpreting arrow plots in such a case. POSTNEK uses the same scaling of the velocity arrow for x and y velocities, so that the direction of the arrows may not correspond to the direction of geometrical features.

SET EQUATION TYPE Menu

ACCEPT CURRENT SWITCHES
DIMENSION
UNSTEADY
FLUID FLOW
ADVECTION
FORCED CONVECTION
CHARACTERISTICS
STRESS FORMULATION
SPLIT FORMULATION
TURBULENCE MODEL
HEAT TRANSFER
ADDL PASSIVE SCALARS
MULTIGRID SOLUTION

Menu function: This menu gives a list of switches whose settings define the type of physical problem to be solved. The name of the equation switch is displayed as a menu item in the left portion of the menu box and its current default value in the right. The values can be binary (True or False) or integer. To modify the value of an item on the list, use the mouse to move the cursor to that item. Each press of the mouse button will toggle the value of the switch setting. When the desired combination of switch settings has been arrived at, pressing the mouse button at the **ACCEPT CURRENT SWITCHES** will cause the program to exit the **SET EQUATION** menu and move on to the **SET PARAMETER** menu.

ACCEPT CURRENT SWITCHES

Exit the **SET EQUATION** menu and proceed to the **SET PARAMETER** menu.

DIMENSION

Values: 2, A (axisymmetric), or 3.

UNSTEADY

Values: T or F.

FLUID FLOW

Values: T or F.

ADVECTION

Values: T or F. Determines whether fluid problem will be Stokes (ADVECTION=F) or Navier-Stokes (ADVECTION=T).

FORCED CONVECTION

Values: T or F. If the problem combines fluid flow and heat transfer, the user can choose if it is convective heat transfer. If only heat transfer has been chosen, this option will allow the user to define a convective velocity field through the use of the INITIAL CONDITION menu, in which the velocity field can be specified as a fortran function, or read in as data from an earlier simulation.

CHARACTERISTICS

Values: T or F. For convection-diffusion problems, the user can choose between the method of characteristics and the standard (default) semi-implicit time-stepping procedure. For a discussion of these timestepping procedures we refer to Sections 3.2, 4.4 and 4.5.

STRESS FORMULATION

Values: T or F. Determines whether the stress formulation of the fluid flow equations will be used.

SPLIT FORMULATION

Values: T or F. Determines whether the Navier-Stokes equations will be solved using the classical first-order splitting scheme, see Section 3.2.

TURBULENCE MODEL

Values: T or F. Determines if $k-\epsilon$ turbulence model will be used.

HEAT TRANSFER

Values: T or F.

ADDL PASSIVE SCALARS

Values: 0 through 9. This enables passive scalars (in addition to the temperature scalar) to be calculated. For ease in bookkeeping, additional passive scalar fields are only allowed when heat transfer is already turned on.

MULTIGRID SOLUTION

Values: T or F. Determines whether to use multigrid as the (iterative) elliptic solver. The default is F, implying that the elliptic solver is based on conjugate gradient iteration.

SET GRID Menu

MAIN MENU
CARTESIAN
POLAR
DX = dx
DY = dy
DIV n
dR = dr
Xo = xo
Yo = yo

Menu function: Change the grid settings and resolution used for latching the mouse during the build procedure. This feature is particularly useful when building complex meshes with elements varying widely in size or meshes having numerous curved elements. Cartesian (default) or polar grids may be used separately or together. When POLAR is set (a "T" will be displayed), new elements entered will automatically have curved sides for any side having vertices at the same radii with respect to the current origin of the polar grid. Note that the automatic curving procedure is not invoked when using MODIFY ELEMENT.

OFF CEILING

Exit the SET GRID menu, return to the BUILD menu, and redraw the grid and mesh if a change in the grid has been requested.

CARTESIAN

Toggle the Cartesian grid from its current state (T/F). If neither Cartesian and Polar are set, there will be no grid latching, other than the standard BLUE button feature which latches to the nearest element vertex.

POLAR

Toggle the polar grid from its current state (T/F). Upon first invocation of the POLAR option, the user is prompted for the number of divisions in the azimuthal direction (the minimum will not be less than 3), and the value to be used for the

distance between grid lines in the radial direction, in physical dimensions.

Change the number of radial "spokes" for the polar grid from its current value of "n".

Modify the resolution of the polar grid in the radial direction such that the radial spacing in physical dimensions is dr.

Modify the resolution of the Cartesian grid such that the spacing between grid lines is dx, or dy (in physical dimensions). The current implementation only supports grids which have equal spacing in X and Y (in screen coordinates), such that changing either DX or DY forces a change in the other.

Move the origin of the polar grid to (xo,yo), thereby allowing easy generation of multi cylinder meshes, etc. Future revisions will provide this flexibility for the Cartesian grid as well, although this is not as critical as for the polar grids.

SET PARAMETER Menu

VISCOS-
RHOCP
CONDUCT
NORDER
TORDER

DENSITY
BETAG
GTHETA
QVOL
FINTIME
NSTEPS
DT
IOTIME
IOSTEP
GRID
DIVERGENCE
HELMHOLTZ
TOLREL
TOLABS
COURANT

Menu function: The **SET PARAMETER** menu displays the subset of the above parameters that pertain to the equation type currently specified. The name of the parameter is displayed as a menu item in the left portion of the menu box and its current default numerical value in the right. To retain the default value as displayed, hit a carriage return. To modify the numerical value, type in (with the keyboard) the new value followed by a carriage return. Each item in the menu is sequentially highlighted in this manner. If the parameters was read in at the **READ PARAMETER** menu, the default values displayed will be those read from the data file. Thus, to make a series of parametric runs, the user can read in the baseline

parameters, type in the parameter to be modified, and hit carriage return for all the other parameters.

VISCOS

The viscosity.

RHOCP

The product of density and specific heat.

CONDUCT

The thermal conductivity.

NORDER

The order of the polynomial expansion plus one, which is constant in all elements and in all of the two or three spatial directions, see Section 3.1. For example, setting norder to 5 in a three-dimensional simulation gives 5x5x5 points in x, y, and z, respectively, in each macro-element comprising the mesh. NORDER must be between 4 and 15 for fluid problems, and between 2 and 15 for heat transfer/passive scalar problems. An initial simulation is usually run with NORDER left at its default value of 5, and subsequent simulations can be made more accurate by increasing NORDER, usually in increments of 2.

TORDER

In a transient simulation, TORDER is the (temporal) order of the multi-step scheme used for the implicit treatment of the diffusion/pressure terms in a fluid problem, or for the implicit treatment of the thermal diffusion a heat transfer problem, see Section 3.2. Note that for the classical splitting scheme, the temporal order is always equal to one.

DENSITY

The density of the fluid or solid.

BETAG

The product of the volumetric expansion of the fluid with temperature, β , and the gravitational constant, g .

GTHETA

The gravitational acceleration angle in the x - y plane as measured counter-clockwise from the negative y axis (GTHETA=0 is "down"). Note that in order to include the hydrostatic effect, the user still needs to include gravity in the forcing function f .

QVOL

A constant volumetric heat source.

FINTIME

The (physical) finishing time of the simulation. The time stepping will cease after this time has been reached.

NSTEPS

The number of time steps in the simulation. This is an alternative to FINTIME for specifying the finishing time of the simulation.

DT

The value of the time step in the simulation. If DT is set to zero, NEKTON automatically (and dynamically) calculates the time step based on the stability requirements imposed on the time step by the courant condition (see section 3.2), or by the stability condition for free surface problems. If DT is set to a nonzero value and the courant condition/free surface condition is violated, NEKTON will reduce the time step sufficiently to ensure (temporal) stability. Thus, a nonzero DT can be used to define a maximum time step; the actual time step will be either the value of DT specified or a lower value determined by the stability criteria. If DT is set to zero in an unsteady simulation where no stability requirements exist (for flows without advection, e.g., Stokes flow, or unsteady conduction heat transfer), NEKTON will (arbitrarily) set DT=0.001 in order to proceed with the simulation. In these cases it is therefore strongly recommended that the user specifies a non-zero DT.

IOTIME

The interval in units of time between outputs of field dumps. If both IOTIME and IOSTEP are set to zero (the default values), NEKTON will output a single dump of the final values of the field variables after the last step of the simulation.

IOSTEP

The interval in number of steps between outputs of field dumps.

GRID

Spacing of the grid lines in the "graph paper" background in PRENEK, measured in screen heights. This "graph paper" is used when building a mesh. Mouse input of elemental corners are directed to the nearest intersection of these grid lines. The user should adjust this GRID coarseness together with the scale factors for maximum convenience in generating a mesh. Where elemental corners in a mesh must be entered at locations not at an intersection of two grid lines, the user can hit the mouse button when the cursor is in the menu area. This will change the mode by which elemental corners are entered from mouse input to numerical input. In this mode, precise numerical values of the corner coordinates can then be entered with the keypad.

DIVERGENCE

The convergence criteria for the outer loop (Uzawa, see Section 3.3) in the iterative solution of the steady Stokes equations. The user can directly specify the allowable divergence in the flow field.

HELMHOLTZ

The tolerance for the iterative solution of the Helmholtz equation. This stop criterion applies to the inner elliptic solves in the iterative solution of the steady Stokes equations, and in the iterative solution of the temperature equation.

TOLREL

The tolerance criterion for velocity errors (relative to the solution velocity) due to

incomplete iteration in solving unsteady fluid problems. NEKTON uses this parameter together with the length scales of the problem to calculate eigenvalues, which are then used to convert this physical criterion into numerical convergence criteria for the iterative loops inside the NEKTON solvers. This ensures optimal efficiency in the NEKTON solvers; the iteration count within each loop is adjusted separately in order to meet the overall accuracy criterion.

TOLABS

This absolute tolerance is used in conjunction with TOLREL described above. NEKTON quits iterating when EITHER the absolute tolerance OR the relative tolerance criterion is satisfied. The strategy for use of these two convergence criteria is illustrated in the following example. Suppose we run an unsteady simulation in which the magnitude of the final expected velocities is about 10. Also suppose that we are interested in accuracy of 1% the initial velocity in the unsteady simulation is zero, achieving a relative error of less than 1% velocity (zero) is impossible, and the code would iterate forever without satisfying this relative criteria. In this case we set an absolute tolerance to TOLABS=0.01, say; this allows an absolute error of 0.01 in the velocity. Whenever the velocity (norm) is greater than TOLABS (e.g. 0.01), the relative criterion TOLREL will be used. The criterion TOLABS therefore typically applies to the initial time step(s) in order to avoid start-up problems.

COURANT

The Courant number, see Section 3.2. Applicable when solving a problem with *convection* (e.g. Navier-Stokes, forced convection heat transfer, or moving geometry). The convection term is treated *explicitly*, and the time step must be restricted in order to maintain *stability*. The maximum allowable time-step is proportional to the COURANT number; the default value is 0.25 for the standard semi-implicit scheme, and 1.0 if the method of characteristics is used. In the latter case the user can often achieve further speed-up by increasing the Courant number to about 2 or so.

VARIABLE PROPERTY Menu

ACCEPT-MATL PROPS

SELECT ELEMENT

SELECT MATL [0]

SET MATL, Q

SHOW MATL, Q

Menu function: This menu is used to specify variable properties, e.g. viscosity, thermal conductivity, and volumetric heat source, see also Section 5.5. Properties can be elementally dependent (i.e., different elements can be made of different materials), temperature dependent, time dependent, spatially varying etc. Additionally, individual elements can be specified as solid (or conduction) elements. Conduction elements can be used when a simulation involves both a fluid and solid phase, such as in the thermal management of electronic components via forced convection. Conduction elements belong to the "solid" domain, (e.g. those in an electronic chip). The "fluid" domain will be a subdomain of the "solid" domain. The "solid" domain can be made up of multiple materials each having independent thermal properties. The elements within each materials group are color-coded to easily distinguish the different materials groups. Conduction elements can also be used when a simulation involves only a solid phase, (i.e., no fluid flow). Here the only purpose in defining conduction elements is to give those elements thermal properties different from the values specified in the **SET PARAMETER** menu.

ACCEPT MATL PROPS

Accept the current status of the materials properties and continue on to the next menu. The default materials properties are those that were set in the **SET PARAMETER** menu.

SELECT ELEMENT

Specify that an element belongs to the material group currently specified by the **SELECT MATL** command.

SELECT MATL [0]

Specify the material group. The value of the currently selected material group is displayed within square brackets immediately after the "SELECT MATL" text in the menu. The material group should be set to the desired value before using the SELECT ELEMENT command, the SET MATL PROPS command, or the SHOW MATL PROPS command.

Elements are, by default, in material group 0. The default properties of material group 0 are those specified in the SET PARAMETER menu. These default properties can be modified with the SET MATL PROPS command.

Positive material groups represent *solid* materials. Elements in positive groups are therefore conduction elements.

Negative material groups represent (immiscible) fluid materials. Since the properties are discrete across elemental boundaries, it makes sense to use these only if the fluids are not allowed to mix.

SET MATL, Q

Set the values of the currently-selected material group.

SHOW MATL, Q

Display the values of the currently-selected material group.

8.2 POSTNEK Menus

ARROW DENSITY Menu

HIGH —

MEDIUM

LOW

Menu function: Determine how closely the arrows will be spaced in a vector plot.

HIGH

Highest resolution; arrows will be plotted at each collocation point on the plotting plane inside each element.

MEDIUM

Medium resolution; arrows will be plotted at *every other* collocation point on the plotting plane inside each element.

LOW

Lowest resolution; one arrow will be plotted at the center of each element on the plotting plane.

SET LOCATION

Calls a submenu to set the spatial location of the plot.

SET PLOT FORMAT

Calls a submenu to set the manner in which the variable will be plotted. Options include point values, $x - y$ plots, streamlines, and surface plots.

SET TIME

If the data are dumped at more than one time slice, SET TIME allows the user to select the time slice to be used in subsequent plots.

SET ATTRIBUTE

Calls a submenu to modify one of the plot attributes, e.g. resolution.

SET VIEW

Sets the perspective from which a three-dimensional object is viewed. SET VIEW prompts for the altitude (in degrees above the $x - y$ plane), and the azimuth (in degrees counterclockwise from the x -axis) of the observer's eye. SET VIEW can also be used to set the perspective when plotting two-dimensional data using the FISHNET GRID format. In this case the z -axis represents the scalar value of the quantity plotted.

DISPLAY OBJECT

Draws a shaded solid representation of the spectral elemental mesh from the current view angle. The external boundaries are opaque and shaded according to their orientation to illustrate the three-dimensional shape of the object. The object is drawn using software z -buffering: the rearmost faces are drawn first and then covered by faces in the foreground. The object is painted on the screen in this manner so that the user can observe the entire three-dimensional object (including hidden lines) as it is built. Periodic "boundaries" remain transparent to emphasize that they are not true external boundaries, but are connected mathematically to other internal faces.

MAIN Menu

EXIT --
CLEAR
PLOT
SET QUANTITY
SET LOCATION
SET PLOT FORMAT
SET TIME
SET ATTRIBUTE
SET VIEW
SET SCREEN
DISPLAY OBJECT

Menu function: This menu controls all other menus in POSTNEK. The user goes into submenus of the main menu to alter the various aspects of the plot. All subsequent plots retain the altered aspects until they are altered again. When the alternations are complete, the user returns to the main menu to plot the result.

EXIT

Exits POSTNEK

CLEAR

Clears all plot data from the screen and redraws a wire frame representation of the spectral element mesh. CLEAR also inputs a page feed into the plot file so that a hard copy will be drawn on a fresh sheet.

PLOT

Plots the currently set quantity in the current format with the current attributes.

SET QUANTITY

Calls the submenu to determine which variable (velocity, temperature, pressure) is to be plotted.

SET ATTRIBUTE Menu

MAIN MENU

SET RESOLUTION

SET ANNOTATION

SET COLORS

SET THRESHOLD LEVEL

SET ARROW SIZE

SET FISHNET HEIGHT

SET HARD COPY

SET DRAW AXIS

Menu function: Adjust attributes.

MAIN MENU

Return to POSTNEK MAIN menu without altering any attributes.

SET RESOLUTION

Invoke the bf SET RESOLUTION submenu to adjust the resolution of one of the the plot formats.

SET ANNOTATION

Annotate the current plot with a label at a user specified location.

SET COLORS

Set the threshold at which the color levels change in color fill and fishnet grid plots. The user can adjust each of the thirteen break points between the fourteen color levels to emphasize a particular range of values.

SET THRESHOLD LEVEL

Plot all colors between a specified lower and upper threshold level.

SET ARROW SIZE

SCALAR PLOT Menu

MAIN MENU

FISHNET GRID

COLOR FILL

CONTOUR LINES

Menu function: Set the format for subsequent surface plots of scalar quantities.

MAIN MENU

Return to POSTNEK MAIN menu without changing plot format for surface plots.

FISHNET GRID

Set the format of the scalar surface plots to FISHNET GRID. The scalar value of the quantity on the plot surface is represented by the distance from the fishnet grid lines to the plot surface. The scalar value is also (redundantly) represented by the color of the grid lines.

COLOR FILL

Set the format of the scalar surface plots to COLOR FILL, in which the scalar value of the quantity on the plot surface is represented by the color at the surface.

CONTOUR LINES

Not yet implemented.

Set the arrow size on vector arrow plots.

SET FISHNET HEIGHT

Set the height of the fishnet grid on scalar plots. In a fishnet grid plot, the grid lines representing the mean value of the quantity are plotted at the level of the plotting plane. The distance of the grid lines from the plotting plane is proportional to the value of the quantity minus its mean value. The fishnet height is the constant of proportion.

SET HARD COPY

Not yet implemented.

SET DRAW AXIS

Determine whether the axis should be included in the subsequent plots.

SET COMPONENT Menu

X	—
Y	
Z	
NORMAL TO PLANE	
SPEED	

Menu function: Specify which component of a vector quantity is to be plotted when a plot format calls for a scalar plot.

X
Y
Z

Specify x , y , or z component of a vector to be plotted.

NORMAL TO PLANE

For plots on a specified x , y or z plane, the component of the vector quantity plotted is normal to the specified plane.

SPEED

Plot the magnitude of the vector quantity.

SET LOCATION Menu

MAIN MENU
X-PLANE
Y-PLANE
Z-PLANE
ALL WALLS
ALL BOUNDARIES
POINT
LINE
SET OF LINES
ELEMENT
HISTORY POINT

Menu function: Set the location for subsequent plots. The location for point, line and surface plots are set independently from one another. Thus, the user can specify an X-PLANE and a LINE simultaneously; the resulting plot will use the location appropriate for the current PLOT FORMAT.

MAIN MENU

Return to MAIN menu with the current location(s).

X-PLANE

Specify the x-plane for surface plots. Applicable for 3-D problems only.

Y-PLANE

Specify the y-plane for surface plots. Applicable for 3-D problems only.

Z-PLANE

Specify the z-plane for surface plots. Applicable for 3-D problems only.

ALL WALLS

Specify all elemental faces with wall boundary condition for surface plots. Applica-

ble for 3-D problems only.

ALL BOUNDARIES

Specify all elemental faces with real physical boundary conditions (no periodic or elemental boundaries) for surface plots. Applicable for 3-D problems only.

POINT

Specify a point. This will be followed by a prompt for the coordinates of the point, which are entered either using the yellow mouse button or typed in from the keyboard. In the latter case the user must first hit the mouse button with the cross hair in the red menu area. This menu item has two different applications. First, it can be used to in conjunction with the **VALUE AT A POINT**, **SET QUANTITY** and **PLOT** options, in which case the value of the user specified quantity will be displayed at the currently set time and at the currently set location (point). Second, it can be used in conjunction with the **HISTORY**, **SET QUANTITY** and **PLOT** options, in which case the user specified quantity will be displayed as a function of time at the history point which is closest to the specified point (see also **HISTORY POINT**).

LINE

Specify a line segment. This will be followed by a prompt for the coordinates of the two endpoints of the line segment.

SET OF LINES

Not yet implemented.

ELEMENT

Spelect element in which data is to be plotted. Applicable for two-dimensional elements only.

HISTORY POINT

Specify a point/quantity. This menu option only applies if the user is solving a moving geometry problem, and has requested the time history of both the velocity

and the coordinates at the same point. The current version of NEKTON will store this information as two separate history points, resulting in an ambiguity if the POINT menu item is invoked. The HISTORY POINT menu item allows the user to choose a history point (with associated quantity) from a list displayed in the menu area.

SET PLOT FORMAT Menu

VALUE AT POINT

X-Y PLOT

SURFACE PLOT

Menu function: Determine whether the quantity will be plotted as a value at a point, $x - y$ (line) plot or value at a point.

VALUE AT POINT

Set plot format to give numerical value of quantity at the currently specified point.

X-Y PLOT

Call **X-Y PLOT** menu to determine which type of $x - y$ plot format to use.

SURFACE PLOT

Call **SURFACE PLOT** menu to determine which type of surface plot format to use.

SET QUANTITY Menu

VELOCITY

PRESSURE

VORTICITY

TEMPERATURE

ERROR

DERIVATIVE

Menu function: Specify which quantity will be plotted. For vector quantities (i.e. velocity), the SET QUANTITY menu will be followed by the SET COMPONENT menu to specify the component for the scalar plots.

VELOCITY

PRESSURE

VORTICITY

TEMPERATURE

Choose quantity to plot.

ERROR

Not yet implemented.

DERIVATIVE

Call submenu to set subsequent plots to be a spatial derivative of quantities, rather than the quantities themselves.

SET RESOLUTION Menu

MAIN MENU

FISHNET GRID

COLOR FILL

CONTOUR LINES

PROFILE PONTS

ARROW DENSITY

CURVE SEGMENTS

Main function: Adjust the resolution of the data display in the different plot formats. The resolution is changed as follows: the current default resolution is displayed in brackets. Hit carriage return to keep the default, or type in an integer to change the value.

MAIN MENU

Return to POSTNEK MAIN menu with the current setting.

FISHNET GRID

Modify the number of lines per element in the fishnet grid plots.

COLOR FILL

Modify the number of bands per element in color fill plots.

CONTOUR LINES

Not yet implemented.

PROFILE PONTS

Modify N_p , the number of points plotted in the $x - y$ profile plot. The "curve" plotted is the sequence of line segments connecting the values at N_p equal intervals.

ARROW DENSITY

Invoke a submenu to set the arrow density for plotting vector fields.

CURVE SEGMENTS

Modify the number of points used to represent a curved segment of the geometry.

SET SCREEN Menu

MAIN MENU

ELEMENT BOUNDARIES

ROTATE VIEW

GRID LATCH

Menu function: Provide the user with control over various attributes of the screen.

MAIN MENU

Exit the SET SCREEN menu, return to the MAIN MENU.

ELEMENT BOUNDARIES

This feature toggles whether the spectral element boundaries will be displayed upon the next CLEAR operation or subsequent redraw of the mesh. When set to .FALSE., only element boundaries having WALL boundary conditions for fluid dynamics simulations, or TEMPERATURE boundary conditions for thermal problems, will be displayed. This feature is frequently useful when generating a hard-copy display of streamlines past an object or any cases where it is undesirable to show only the physical object.

ROTATE VIEW

This provides an extra degree of freedom for viewing three dimensional objects by taking an acyclic permutation of the X-Y-Z viewing coordinates. Some experimentation is recommended.

GRID LATCH

Toggle the grid latching feature when entering x-y coordinates with the mouse. The default is that the YELLOW button will latch to the original Cartesian grid that was present during the BUILD phase in PRENEK. Setting GRID LATCH to .FALSE. disables this function, and allows input resolution limited only by the input device and the user's steady hand. The function of the BLUE button (vertex latching) will remain unchanged.

SURFACE PLOT Menu

SCALAR PLOT

VECTOR PLOT

Menu function: Determine whether subsequent surface plots of vector quantities will plot the full vector quantity, or will plot one (scalar) component of the vector quantities.

SCALAR PLOT

For subsequent plots of vector quantities, plot the scalar component specified in **SET COMPONENT** menu. Invoke the **SCALAR PLOT** menu to choose the format of the scalar plot.

VECTOR PLOT

Plot vector plot of vector quantities.

X-Y PLOT Menu

HISTORY

INTEGRAL QUANTITY

STREAMLINE

PARTICLE PATH

PROFILE

SET OF PROFILES

Menu function: Set the format of the $x - y$ plot.

HISTORY

Plot the value of the given quantity at a point versus time.

INTEGRAL QUANTITY

Plot the value of an integral quantity (e.g., drag on a body or heat transfer across a surface) as a function of time. After invoking this menu item, the user can select an integral quantity from a list displayed in the menu area; this list correspond to the list of objects defined in PRENEK.

STREAMLINE

This option invokes a line by line streamline integration routine for 2D and 3D flow fields. It generates streamlines associated with the current (fixed time) velocity field by integrating along the path of motion via a 2nd order accurate time stepping scheme. Input is either from a *fname.str* (where *fname* is the current session name), the keyboard, or the mouse (for 2D only). To use the streamlines in a post-processing session, you must first select the following menu options:

X-Y PLOT

STREAMLINE

PLOT

When you select the PLOT option, POSTNEK begins to compute a large number of spatial derivatives - this can take a few minutes for large data files. You will

then be prompted for an integer input ranging from -1 to 2. Enter 0 if you have stored streamline information in a *.str file prior to commencing the postprocessing session. Otherwise, enter 1. When you wish to return to the main menu, enter -1 when you see this prompt.

If you enter 1 above, the next prompt will be for:

X, Y, (Z,)DT, NSTEPS, IO, ICOLOR:

where *X*, *Y*, and *Z* refer to the starting point of the streamline.

DT refers to the step size which should be taken between Lagrangian updates - enter 0 if you want to see a continuous line. In that case, *DT* will be chosen so that the integration moves the cursor a reasonable distance on the screen for each integration step, independent of the magnitude of the local velocity vector. If *DT* is negative, the integration will go in reverse, against the flow.

NSTEPS is the maximum number of steps the integration will carry out. If the streamline leaves the domain before step number *NSTEP*, the integration will terminate and you will be prompted for another input. Setting *NSTEPS* negative when *DT* is zero will make the streamline go in reverse. Setting *NSTEPS* to zero will allow you to go back to the previous prompt level and exit.

IO allows you to refine the step size. Setting *IO* to 1 is the standard. A more refined integration will be obtained by setting *IO* to 2,3,... thus cutting the step size by 2,3,... etc. Of course, this takes longer. The integration is 2nd order accurate so that setting *IO* to 2 will result in a factor of 4 increase in integration accuracy.

ICOLOR selects the color of the streamline: 0 is black - which will be visible on top of a color pressure or temperature plot; 1 is white. The

maximum is 15; 9 is green which nicely contrasts the element boundaries.

To exit the "new line" mode of the streamline plotting routine, enter 0 for *NSTEPS* (0,0,0,0,0,0,0,0 is the typical line entry). This will return you to the streamline options prompt. Entering -1 at this point will return the control to the MAIN MENU. Upon subsequent re-entry into the streamline plotting routine, you can rapidly redraw a the streamlines which have been previously been integrated. This is particularly useful for 3D calculations where you wish to view the streamlines from another angle. (Choosing the SET VIEW option allows the user to re-orient the domain.) A finite number of streamlines are stored for this purpose; in the case where the limits of the storage arrays are exceeded, the last computed streamline information overwrites that computed earlier.

The best results will be obtained if the user indulges in a little experimentation.

PARTICLE PATH

Plot the path of a particle whose initial location was specified by PRENEK. The path is governed by the local fluid velocity. Not yet implemented.

PROFILE

Plot the value of a variable versus distance along a line segment whose endpoints were defined in SET LOCATION or by the last entry in the SET OF PROFILES plotting mode..

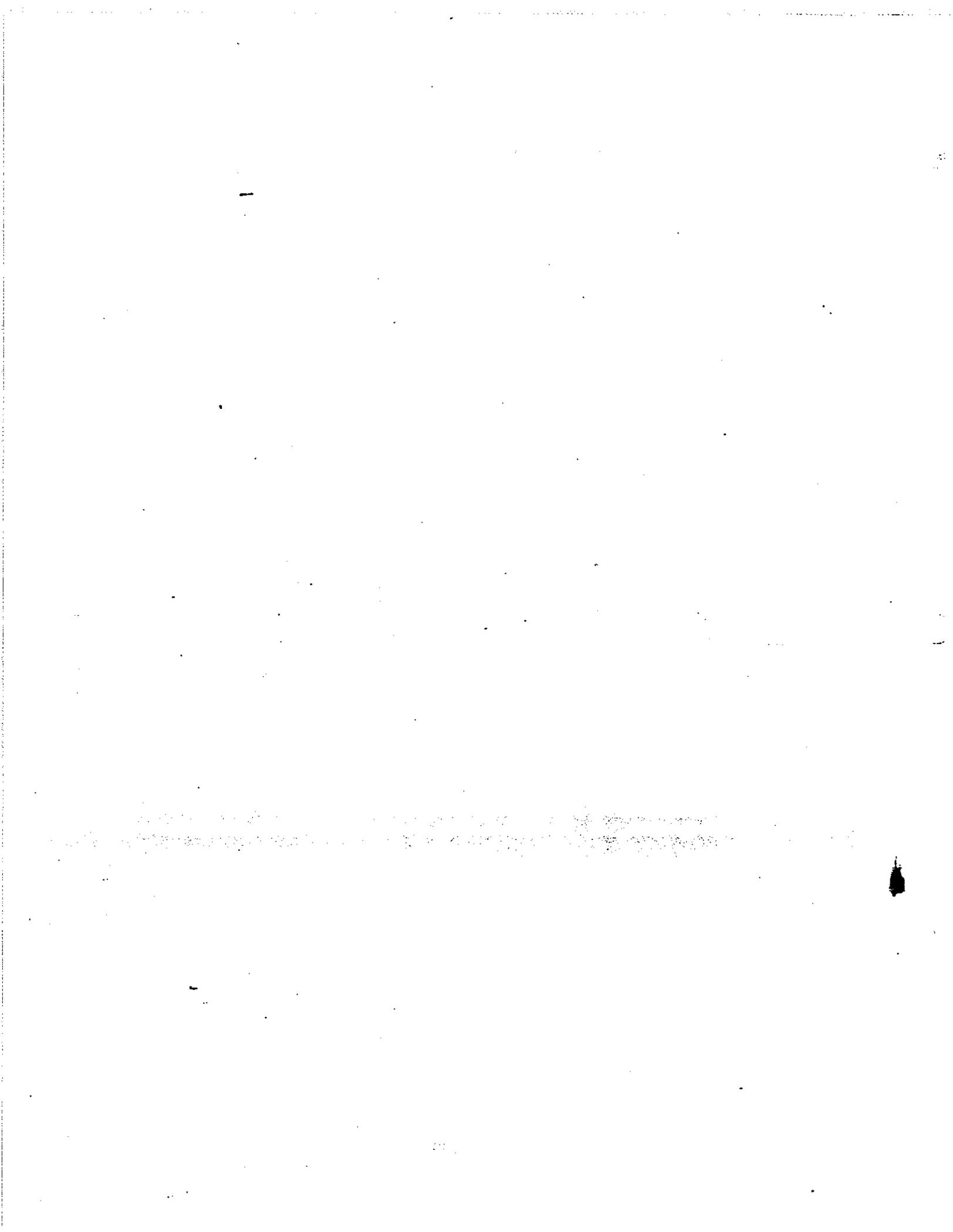
SET OF PROFILES

Plot multiple profiles on the computational domain (e.g. local boundary layer profiles). Line segments are entered interactively via the standard NEKTON protocol after selecting the PLOT option at the MAIN MENU level. Profile amplitude is set proportional to ARROW SIZE (see SET ATTRIBUTE menu option) and is scaled according to the minimum and maximum values in the current working field, except

when plotting VELOCITY with the NORMAL TO LINE component selected. In that case, the amplitude is scaled according to the minimum and maximum speeds in the computational domain.

To leave the SET OF PROFILES plotting mode, enter the same X-Y location for both endpoints of the line segment. The last line segment plotted is the default location for the PROFILE option (see above), so that a profile of particular interest can be readily viewed on an expanded and annotated scale.

SET OF PROFILES is not yet implemented in 3-D.



Chapter 9

Troubleshooting NEKTON

In this section the solution to a number of potential problems a user might encounter in running NEKTON are presented. These solutions are divided into the preprocessor, postprocessor, and solver sections below and organized alphabetically within each section according to the symptoms associated with each problem.

Preprocessor Difficulties:

- Boundary conditions requested on internal sides
- Not enough memory
- Tektronix terminal problems
- Valloc failed

NEKTON Solver Difficulties:

- Compiler errors
- Core dump
- Discontinuous derivatives
- Divergence in solution
- Excessive cpu times

- Insufficient virtual address space
- Linker errors
- No results files
- Premature exit
- Wiggles in solution

Postprocessor Difficulties:

- Hard copy problems
- Not enough memory
- Valloc failed

9.1 Preprocessor Difficulties

Boundary conditions requested on internal sides. Nekton requires that elemental sides to be connected in with one-to-one correspondence. Furthermore, two sides which connect adjacent elements must have common endpoints on BOTH ends of the sides. If these requirements are not met, PRENEK cannot connect the elements and will assume that the sides are on external boundary. This error should come to the user's attention when boundary conditions are requested on these sides. The solution to this problem is to read in the mesh in a new session and correct the disconnected sides.

Not enough memory. This message appears on a sun workstation when there is not sufficient memory to run PRENEK. Memory can be freed by closing all windows except the graphics tool used by PRENEK and logging out from any other processes. If there is still not enough memory (this should only on machines with 4 mbyte or less) the program must be recompiled with smaller arrays according to the installation procedure.

Tektronix terminal problems. The Tektronix terminal can become hung up for a variety of reasons. One solution for this problem is to hit the "Setup" key and type "resēt" followed by a carriage return. A ^C will get then get the user out of PRENEK.

Valloc Failed. See "Not enough memory"

9.2 NEKTON Solver Difficulties

Compiler Errors. If NEKTON aborts during the COMPILING AND LINKING stage, the problem is probably an incorrect Fortran statement in the user-input Fortran lines (from BOUNDARY CONDITION, INITCOND, and DRIVEF menus). This can be verified/corrected either in the SESSIONNAME.REA or SESSION-NAME.USER files associated with with the session. The lines must have correct Fortran syntax; the statement $UX=0.0$ is legal, whereas 0.0 is not.

Core Dump. For various reasons (memory allocation, etc.) a UNIX machine can give a core dump, resulting in a slow transfer of the memory contents into an unusable file. This can be avoided by inserting the line *limit coredumpsize 0* in your .login file.

Discontinuous Derivatives. See "Wiggles in Solution"

Divergence in solution. This problem can occur when the split formulation is used. The errors associated with splitting are of order $(\nu\Delta t)$, and can become large in problems with low Re. The solution is to switch to the default non-split Uzawa algorithm.

Excessive CPU times. There are a large number of reasons that the cpu time can become large. Section 4.4 outlines some of the techniques for reducing CPU time.

Insufficient virtual address space. See "Linker errors".

Linker errors. Linker errors can occur on a vax when the quota for virtual memory (page file quota) is set too small. The short fix is to include the "compile" switch: *nek session compile*.

The longer term fix is to have the system manager increase the VMS parameters *wsquo*, *wsmmax*, *virtualpagecnt*, or *pagefilequota*. Depending on job size, NEKTON can require working sets of up to a few thousand blocks and page piles of up to 35000 blocks.

Linker errors can occur on a Sun when linking PRENEK or POSTNEK when the SunCore libraries are missing. These are standard libraries that come with the operating system release which must be installed for the graphics routines to run.

No Results Files. When NEKTON exits before finishing, the partial results are stored in the files P1FLD and P1HIS. These should be moved to the files *session.fld* and *session.his*, respectively.

Premature Exit. See "No results files"

Wiggles in Solution. Elements should not be highly distorted (e.g., interior angles at the vertices should not be either very small or very large, but rather within 30° of 90°). Rectilinear elements give the best accuracy.

Variations in element size should be relatively smooth, not abrupt. Changes in size between adjacent elements should be limited to a factor of about 5.

At high Reynolds numbers, thin boundary layer and shear layers will form, and must be resolved by appropriate placement of elements and points.

9.3 Postprocessor Difficulties

Hard Copy Problems. No hard copy or hard copy in wrong format: There is a file *defaults.nek* in the NEKTON directory. There should be a line reading "printer postscript", "printer qms", or "printer none" depending on the type of printer at your site. In unix systems, this file must be copied to the user's working directory.

Missing frame on plots: The CLEAR command should be given immediately before the PLOT command when the plots require the mesh and frame on the hard copy.

Not enough memory. This message appears on a sun workstation when there is not sufficient memory to run POSTNEK. Memory can be freed by closing all windows except the graphics tool used by POSTNEK. If there is still not enough memory (this should only on machines with 4 mbyte or less) the program must be recompiled with smaller arrays according to the installation procedure.

Valloc Failed. See "Not enough memory"

Appendix A

Restrictions in Capabilities and Numerical Methods

In this appendix we summarize the restrictions in NEKTON as regards physical capabilities (Section A.1) and numerical methods (Section A.2). The restrictions in Section A.2 are presented in the form of a table, and should be interpreted in the following way. For a specific problem, if the user wants to combine certain features in NEKTON, and does not find any comments in the table, the convention will be that this a valid combination. Otherwise, if the table indicates an X for a combination, it means that this combination is illegal (either of fundamental reasons, or because NEKTON does not accomodate this feature). If the table indicates a number, it means that the combination is conditionally valid, and the condition is given below the table.

A.1 Restrictions in Physical Capabilities

In principle NEKTON can solve any fluid, heat transfer and passive scalar problem which can be described by the governing equations (2.1)-(2.4), in fixed or time-dependent domains, and subject to the most general types of boundary conditions. However, certain *fundamental* restrictions apply as regards *combinations* of features, and to what *extent* a feature can be used. In this section we summarize these restrictions which are partly imposed by the user, partly by PRENEK, and partly by the computational engine, NEKTON.

Nonstress Formulation. For fluid problems the nonstress formulation offers the most efficient solution strategy, however, only velocity boundary conditions (wall, inflow, outflow) and constant properties apply. The nonstress formulation should therefore *not* be combined with traction boundary conditions, free surface problems, fluid layers and variable ρ , μ . Furthermore, mixed boundary conditions (e.g., symmetry) can only be applied if the boundary aligns with one of the Cartesian x , y or z axes. If the nonstress formulation does not apply, the stress formulation should be used.

3-D Mesh Construction. In PRENEK, a three-dimensional mesh is constructed by adding layers of hexahedral spectral elements in the z -direction. The parallel planes which separate the layers are required to be parallel to the x - y plane, and the first such plane is located at $z = 0$. The edges of a three-dimensional spectral element can only be deformed in the x - y plane.

Moving Geometry. For time-dependent geometries NEKTON can handle moderately large mesh distortions. Note however that NEKTON does not provide any (dynamic) remeshing, and that the number of macro-elements, the order of the macro-elements and the topology of the mesh remain *unchanged* during a simulation. Hence, it is the responsibility of the user to decide when will a mesh become "too deformed" and thus requires remeshing. The execution of the program will terminate when the mesh becomes unacceptable.

A.2 Restrictions in Numerical Methods

	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)
(A) Nonstress, Split (P_N/P_N)		X	X			X		X
(B) Nonstress, Uzawa (P_N/P_{N-2})	X		X					X
(C) Stress, Uzawa (P_N/P_{N-2})	X	X						
(D) Characteristics					X	X		
(E) Moving geometry				X		X		
(F) Steady state solver	X			X	X		X	(1)
(G) Convection						X		
(H) Variable ρ, μ	X	X				(1)		

INPUT SETTING:

(A) Nonstress, Split formulation is set by selecting Stress Formulation as "N" and Split Formulation as "Y" in the SET EQUATION TYPE Menu.

(B) Nonstress formulation (default) is set by selecting Stress Formulation as "N" and Split Formulation as "N" in the SET EQUATION TYPE Menu.

(C) Stress formulation is set by selecting Stress Formulation as "Y" and Split Formulation as "N" in the SET EQUATION TYPE Menu.

(D) Characteristics is set by selecting Advection as "Y" and Characteristics as "Y" in the SET EQUATION TYPE Menu.

(E) Moving geometry is set by selecting Moving Wall, Melting Front or Free Surface in the BOUNDARY CONDITIONS Menu, or by selecting Internal BC in the MODIFY BOUNDARY CONDITIONS Menu.

(F) Steady state solver is set by selecting Unsteady as "N" in the SET EQUATION TYPE Menu.

(G) Convection is set by selecting Advection as "Y" (fluid problem), or With Convection as "Y" (heat transfer problem) in the SET EQUATION TYPE Menu.

(H) Variable fluid properties are set in the VARIABLE PROPERTY Menu.

NOTES:

(1) Only if the variable properties depend on spatial coordinates, and not time or any physical variable (e.g., velocity or temperature). If a property is a function of the physical variables, the (nonlinear) problem must be solved as a time dependent problem.

Index

ABORT 147
ABORT (SAVING PARAMETERS) 137
ACCEPT B.C.'s 123
ACCEPT CURRENT SWITCHES 157
ACCEPT MATL PROPS 167
ACCEPT MESH 123
ADD ELEMENT 132
ADD FACE 149
ADD OBJECT 150
ADDL PASSIVE SCALARS 157
ADVECTION 157
ALL BOUNDARIES 177
ALL WALLS 177
ALTER PARAMETERS 137
ARROW DENSITY 182
AXIS 124
BETAG 162
BUILD FROM FILE 137
BUILD INTERACTIVELY 137
BUILD MENU 140
CARTESIAN 160
CEILING 132
CHARACTERISTICS 158
CHARACTERISTICS 157
CHOOSE ELEMENT 139
CLEAR 171
COLOR FILL 173, 182
CONDUCT 162
CONSTANT 142
CONTOUR LINES 173, 182
CONVECTION 124
COORDINATES 153
CORNER FRAME 143
COURANT 162
CURVE SEGMENTS 182
CURVE SIDES 132, 136
DEFAULT=0 146
DELETE ELEMENT 132
DELETE FACE 149
DELETE OBJECT 150
DENSITY 162
DERIVATIVE 181
DIMENSION 157
DISPLAY OBJECT 171
DIV 160
DIVERGENCE 162
DOWN LEVEL 132, 139
dR 160
DRIVE FORCE 151
DT 162
DX 160
DY 160
ELEMENT 177
ELEMENT BOUNDARIES 184
END ELEMENTS 132
END GLOBAL REFINE 143
END LEVEL 148
END OBJECT 149
ENTER POINT 145
ERROR 181
EXIT 151, 171
FINTIME 162
FISHNET GRID 173, 182
FLUID FLOW 157

FLUX 124
 FORCED CONVECTION 157
 FORTRAN FUNCTION 142
 FREE SURFACE 124
 FUNCTION 146
 GLOBAL REFINE 132
 GRID 162
 GRID LATCH 184
 GTHETA 162
 HEAT TRANSFER 157
 HELMHOLTZ 162
 HIGH 170
 HISTORY 151, 186
 HISTORY POINT 177
 IMPORT UNIVERSAL FILE 137
 INITIAL COND 151
 INSULATED 124
 INTEGRAL QUANTITY 151, 186
 INTERNAL BC 148
 IOSTEP 162
 IOTIME 162
 LINE 177
 LOW 170
 MAIN MENU 139, 145, 146, 150, 153, 160,
 173, 174, 177, 182, 184
 MAIN B.C. MENU 124
 MAKE CIRCLE 140
 MAKE SINE WAVE 140
 MAKE SPLINE 140
 MEDIUM 170
 MELTING FRONT 124
 MODIFY BC 148
 MODIFY ELEMENT 132, 136
 MODIFY OBJECT 150
 MOUSE 156
 MOVING WALL 124
 MULTIGRID SOLUTION 157
 NORDER 162
 NORMAL TO PLANE 176
 NSTEPS 162
 OBJECT 151
 OFF CEILING 136
 OUTFLOW 124
 OUTFLOW/N 124
 OUTPUT 151
 PARTICLE PATH 186
 PASSIVE SCALAR 153
 PERIODIC 124
 PERIODIC 124
 PICTURE FRAME 143
 PLOT 171
 POINT 177
 POLAR 160
 PRESOLV 146
 PRESSURE 145, 181
 PROFILE 186
 PROFILE POINTS 182
 QUANTITY 171
 QVOL 162
 RADIATION 124
 READ PREVIOUS PARAMETERS 155
 REDRAW ISOMETRIC 132
 REDRAW MESH 132
 RESTART 146
 REVIEW/MODIFY 123
 RHOC 162
 ROTATE VIEW 182
 SCALAR PLOT 185
 SELECT ELEMENT 167
 SELECT MATL [0] 167
 SET ANNOTATION 174
 SET ARROW SIZE 174
 SET ATTRIBUTE 171
 SET BCs 147
 SET COLORS 174
 SET DRAW AXIS 174
 SET ENTIRE LEVEL 124
 SET FISHNET HEIGHT 174

SET GRID 132
SET HARD COPY 174
SET LOCATION 171
SET MATL PROPS, Q 167
SET OF LINES 177
SET OF PROFILES 186
SET PLOT FORMAT 171
SET RESOLUTION 174
SET SCREEN 171
SET TIME 171
SET THRESHOLD LEVEL 174
SET VIEW 171
SHEAR 124
SHEAR (LOCAL) 124
SHOW BC 148
SHOW MATL, Q 167
SHOW PARAMETERS 137
SPEED 176
SPIDER WEB 143
SPLIT FLOOR 143
SPLIT FORMULATION 157
STRAIGHTEN CURVE 140
STREAMLINE 186
STRESS B.C.s 124
STRESS FORMULATION 157
SURFACE PLOT 180
SYMMETRY 124
TEMP 124
TEMPERATURE 145, 153, 181
TOLABS 162
TOLREL 162
TORDER 162
TRACTION 124
TRACTION (LOCAL) 124
TURBULENCE MODEL 157
TYPE 156
TYPE IN NEW PARAMETERS 155
UNSTEADY 157
UP LEVEL 132, 139

USE NORMALIZED 156
VALUE AT POINT 180
VECTOR PLOT 185
VELOCITY 124, 145, 153, 181
VELOCITY (LOCAL) 124
VISCOS 162
VORTICITY 181
WALL 124
X 176
X-PLANE 177
X-Y PLOT 180
Xo 160
Y 176
Y-PLANE 177
Yo 160
Z 176
Z-PLANE 177
ZIPPER 143