Contents

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 system requirements in terms of hardware and software, 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 is applicable to direct simulation of inertia-free (Stokes) flow, laminar flow and transitional flow just before the flow becomes turbulent. For fully developed turbulent flow, turbulence models are provided for the prediction of the mean (time-averaged) flow behavior; see Section 2.6.

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; see Section 2.2. 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 ϕ_i, i=1,2,…,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,

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

the continuity (mass conservation) equation in D_F,

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

the energy equation in $D_F \cup D_S$,

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

and a convective-diffusive equation for each passive scalar i, i=1,2,…,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; i.e., the momentum equation, in which the nonlinear term $(\mathbf{u} \cdot \nabla)\mathbf{u}$ is set to zero, and the continuity equation.
(6) Steady Navier-Stokes flow; i.e., the momentum equation, in which and the transient term $\partial_t \mathbf{u}$ is set to zero, and the continuity equation.
(7) Steady Stokes flow; 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.
(8) Transient conduction heat transfer; i.e., the energy equation only.
(9) 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 1,2,3,4,5,8), 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 the geometry is time-dependent or if turbulence model is used. In addition, stress formulation must be employed 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 $\partial \overline{D}'_F$ (note $\partial D_F = \partial D'_F \cup \partial \overline{D}'_F$). In addition, $\partial D_s, \partial D'_S$ and $\partial \overline{D}'_s$ are analogously defined. These distinct portions of the domain boundary are illustrated in Fig. 2.1. The restrictions on the domain for NEKTON are itemized below.

- 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 seperate 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.

Fully general three-dimensional meshes generated by other softwares packages can be input to PRENEK as import meshes. Currently, meshes generated by PRE-BFC and I-DEAS can be accepted by PRENEK.

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 $\mathbf{w}$ 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 a 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 responsibity of the user to decide when a mesh would 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 $\mathbf{u}$ are specified; or it can be *all Neumann* if all traction components $\mathbf{t} = [-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] \cdot \mathbf{n}$, where $\mathbf{I}$ is the identity tensor, $\mathbf{n}$ 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 Dirchlet, all Neumann and mixed Dirichhlet/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.

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 composes 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 continuty equations, then only the common boundary between the fluid and the solid (i.e., all or portion of $\partial\overline{D}'_F$ in Fig. 2.1) can be defined as the melting front. In this case, segments on $\partial\overline{D}'_F$ that belong to the dynamic melting/freezing 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 tempertaure 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.

- $\overline{\beta}$, the effective thermal expansion coefficient, is assumed time-independent and constant.

- $\hat{\mathbf{g}}$, the unit vector in the direction of the acceleration of gravity, is taken to be time-independent and constant.

- $\mathbf{f}(t)$, 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, $\overline{\beta}$ would be the Rayleigh or Grashof number, wheareas in dimensional terms it represents the product of the coefficient of thermal expansion and the acceleration of gravity.

2.6 Turbulence Model

Currently, an algebraic turbulence model based on the mixing length theory is available for the simulation of fully developed turbulent flows. In this case an effective viscosity, μ_E, defined as $\mu_E = \mu + \mu_T$ is used instead of the molecular viscosity μ in Equation (2.1). In the above definition, μ_T is the turbulent (or eddy) viscosity which accounts for the Reynolds stresses associated with turbulent flow. In the algebraic turbulence model, the turbulent viscosity is computed as

$$\mu_T \; = \; \rho \, \ell^2 \, |J| \tag{2.5}$$

where ρ is the fluid density, ℓ is the mixing length, and J^2 is a quadratic function of the mean-velocity gradients. Note that the mixing length is an input parameter dependent upon the type of flow under consideration. In NEKON, the mixing length is expressed as the product of a characteristic length scale and an intensity factor. The characteristic length scale is determined automatically based on the flow geometry. The distribution of the intensity factor is obtained from the solution of a Poisson equation, for which the *maximum* value of the intensity factor in the entire domain is input by the user (the value for circular pipes is used as default).

In the turbulent flow calculation, it is assumed that near solid boundaries, the thickness of the viscous sublayer (within which the flow is laminar) and the buffer layer (within which the viscous stresses and the Reynolds stresses are of the same order of magnitude) is negligble. Consequently, the no-slip conditions usually imposed on walls are replaced by the following shear boundary conditions: zero normal velocity and nonzero tangential shear computed based on a logarithmic velocity profile. Thus, wall boundaries should be interpreted as the boundaries at which the logarithmic layer terminates. In addition, tangential velocity is in general *nonzero* on all wall boundaries. We remark that the above law-of-the-wall formulation alleviates the need to resolve extremely steep velocity gradients across the viscous sublayers and allows the use of relatively coarse meshes for high-Reynolds-number turbulent flow simulation.

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 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 hexahedral elements, of which each face corresponds to a nondegenerate quadrangle, see Figure 3.2. 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.

Elemental Mesh

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 $0 < \theta < 180°$ (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 generated in PRENEK as 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. Note that this is an *artificial* constraint of PRENEK, which can be removed with import meshes, see Section 1.2.1.

- **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).

- **E4**: In axisymmetric analysis, all elements must lie in the positive y halfplane. For elements which have one face bounded the axis of symmetry, this face must be input as the first face; that is, both Corner I and Corner II of the element must lie on the axis. It is illegal for any axisymmetric element to have only one corner lying on the axis of symmetry. These last two requirements are due to the use of different types of polynomial expansion in the axial and radial directions.

Mesh Structure As regards how elements may be placed together, the fundamental rule 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, (1) and (3), 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 are expanded in terms of $(N-1)^{th}$ 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.

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 with higher K or N.

3.2 Temporal Discretization

In transient simulations NEKTON employs both explicit and implicit time-integration techniques, and the solution is updated to the new time-level using various combinations of multistep and multistage schemes. In this section we discuss primarily issues related to accuracy and stability.

Convective Terms The convective terms $(\mathbf{u}\cdot\nabla)\mathbf{u}$, $(\mathbf{u}\cdot\nabla)T$, $(\mathbf{u}\cdot\nabla)\phi_i$ 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 $\mathcal{O}(\Delta t^3)$, 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.*

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

$$\Delta t < C \times MIN_{D_F}\{\frac{\Delta x}{\mid u_x \mid}, \frac{\Delta y}{\mid u_y \mid}, \frac{\Delta z}{\mid u_z \mid}\}\,, \tag{3.1}$$

where C is the Courant number, Δx,Δy,Δz 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_F} 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, $\rho\mathbf{f}$, $-\rho\beta\hat{g}T$, 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.

Diffusion 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 $\mathcal{O}(\Delta t^Q)$. 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.

Method of Characteristics 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 (method of characteristics) is available for the user; this method can potentially yield significant speed-up, see Section 3.4.

Additional Stability Limit 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:

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

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 $MIN_{\partial D_{st}}$ refers to the minimum over the surface tension boundary.

In this case, the order P is automatically set equal to the order Q for the Backward Differentiation scheme for maximum stability. For other type of moving boundary problems, the theoretical basis is difficient. 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).

Splitting Scheme 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 formulation, and the temporal accuracy is $\mathcal{O}(\Delta t)$.

3.3 Solution Methods

In order to solve the fully discrete set of equations NEKTON employs different solution strategies depending on the specific problem, available memory, and computer architecture. If the problem to be solved is a steady two-dimensional problem in fixed geometry with no turbulence model, NEKTON uses a direct solver to solve the discrete equations, *provided* that the computer which is being used has enough memory for such an approach. Otherwise, a time stepping approach combined with iterative solution procedures are used.

A direct solver requires the construction and storage of the matrices representing the discrete spatial terms, in contrast to an iterative solver which only requires a way to evaluate the action of the the discrete spatial operators on some vectors. While a direct approach is well suited for smaller (two-dimensional) problems, such an approach becomes too expensive in terms of memory and computational cost for large (three-dimensional) problems, and thus necessitates an iterative approach.

Steady-state Solver For a steady two-dimensional problem, for which a direct solution approach is applicable, the discrete equations will often be nonlinear (e.g., for the Navier-Stokes equations or for natural convection problems). In this case the equations are linearlized and convergence is achived via a Newton iteration. The user can specify the initial state to linearize around (e.g., by using the restart option); if no initial conditions are specified, NEKTON will first solve (directly) the steady linear problem (e.g., the steady Stokes solution), and then proceed by linearizing around this state.

The computationally most expensive part in the direct solver is the LU-factorization (Gaussian elimination) associated with each linearization. After each completed LU-factorization NEKTON will therefore proceed with back-substitutions as far as possible, and will only perform a new linearization if this is required for convergence.

Steady-state Acceleration For steady problems where the direct solution strategy is not applicable, e.g, for all three-dimensional problems, the governing equations are solved in the time-dependent evolution form given in (2.1)-(2.4). If a steady state state solution exists, and the user has specified a steady calculation, NEKTON automatically obtains the solution as the time-asymptotic result of a transient simulation, see Section 3.2. In addition, the steady state solver applies an extrapolation procedure in order to accelerate the convergence to steady state.

Time-dependent Calculations For transient simulations the solution of the fully discrete set of equations at each time step is characterized by the use of iterative solvers and tensor-product sum-factorization techniques; no direct solvers are used in this case.

Convective Terms 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.

Diffusion Terms 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. Note that in the iterative solution procedures the global or elemental matrices are never formed explicitly, an essential feature for solving large (three-dimensional) problems. The storage requirement in NEKTON is in this case minimal; it is on the order of the number of collocation points in the mesh.

The fully discretized unsteady Navier-Stokes equations require, at each time step, the solution of one elliptic system for the pressure, and one for each velocity component. In the classical splitting scheme, the pressure operator is a standard discrete Laplace operator. In contrast, a discretization based on consistent approximation spaces (polynomial degree N for velocity and polynomial degree $N-2$ for the pressure) results in a non-standard elliptic pressure operator which is inverted using a new domain decompostion solver. While the latter approach is slower, it is generally more accurate and robust; in NEKTON the user has a choice of which of the two schemes to use, see Section 3.4.

Parallelism The data structure used in NEKTON also allows the code to run efficiently on modern architectures, in particular on distributed memory parallel machines. The loosly 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*. Note that in a parallel environment NEKTON only applies iterative solvers. This is mainly because a direct solution approach, such as Gaussian elimination, is cross-coupled and less effective at exploiting parallel computer architectures.

Advantages The motivation behind spectral element (rather than low-order finite element) 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 classical splitting scheme.
- the non-stress (default) formulation
- the stress formulation

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.

Split Formulation 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 $Re > 25$.

Non-stress Formulation The non-stress formulation uses consistent approximation spaces for the velocity and pressure, and gives accurate results for the entire range of Reyolds 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.

Stress Formulation 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 rigourous of the three, and requires slightly more solution time than the non-stress formulation.

Courant Number As discussed in Section 3.2, NEKTON treats the convective terms 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 $C = 0.4$.

Method of Characteristics Bigger savings can be achieved for time-dependent 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 System Requirements

The user should read the appropriate (Unix or VMS) sections from the Operating Systems section. The user or system manager can refer to the sections on hardware if he is considering modifying his computing environment for NEKTON usage. The user should read the section on Graphical Software if he is not familiar with the usage of X Windows or Motif applications.

Operating Systems:

- UNIX
- VMS

Hardware:

- X Windows Workstations
- Cray
- Parallel Processors

- Printers
- Networks

Graphical Software:

- X Windows
- Motif

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. PRENEK preserves one backup version of the master problem definition file by appending to it the character "˜" (e.g. test.rea is copied to test.rea˜), but in general, 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 {*nekb 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 NEKTON user interface, consisting of PRENEK and POSTNEK, requires a display compatable with the X Windows standard. For the setup and postprocessing of two dimensional problems, a black and white display is adequate, but a color display provides enhanced plotting capability and enhanced visualization of mesh and boundary condition input. For three dimensional problems, a color display is strongly recommended.

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. An image of the drawing window of PRENEK or plotting window of POSTNEK can be produced in the PostScript format by pushing the DUMP SCREEN button in the menu window. An alternate possibility is to use a hardware compatible hard copy unit which will accept a hardware 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.

4.1.3 Graphics Software

X Windows

X Windows is a graphical system which resides on a layer above the operating system. X Windows gives the capability to use multiple windows across a network and to interact with an application program in a standard format independent of hardware, and to a large extent, operating systems.

The graphical interface of NEKTON, the PRENEK and POSTNEK programs, are such applications. They can be can be invoked on an X Windows workstation by typing ”pre” or ”post” from a standard text window. The interface will open two subwindows, a menu window near the top of the screen and a drawing/plotting window below. The interface can display menus, text, and graphics and other output simultaneously through different windows. However, the interface can accept keyboard input only through the window to which the keyboard is currently attached. The keyboard can be attached to a window by moving the cursor to within the window. Some invocations of the X Windows window manager, notably DecWindows, additionally require a mouse click inside that window to attach the keyboard. Thus, if PRENEK or POSTNEK prompts for text input, but the text entered does not appear, the problem can be solved by attaching the keyboard to the window from which the prompt is displayed.

X Windows applications can be run across a network. This enables the interface to be run on one workstation, but displayed on the screen of another workstation or on an X Windows terminal. As an example, suppose a user is sitting in front of a workstation called MyOffice which has has a window opened which is connected to a remote Unix machine called Remote. He wishes to run POSTNEK on Remote, but display the results on MyOffice. On the window he types:

```
setenv DISPLAY MyOffice:0
post
```

All interaction with the user (mouse, keyboard, and display) which follow will occur on MyOffice, as if POSTNEK were running locally on MyOffice.

There are 3 mouse buttons on an X Windows workstation. The leftmost button corresponds is the default button used to invoke menus, key in numbers into the keypad, and enter coordinate points in the drawing area. The rightmost button can also sometimes be used (as described later in the manual) to enter coordinate points into the drawing area which "latch" to previously entered points. The center button is used to display the cursor coordinates.

The X Windows system allows windows to be resized and moved arbitrarily. The NEKTON interface can scale the drawing area appropriately for drawing windows of arbitrary size and aspect ratio. This, together with the zoom function, can be very helpful when building and modifying large meshes. The drawing window can be enlarged to use the screen real estate to its full advantage, and reduced in size when the screen needs to be shared with other windows. The windows can also be temporarily reduced from a full window to an icon to clear the screen for other uses. The method for changing window size is dependent on the local window manger; usually it is accomplished by pulling the top right corner of the window with the mouse until the desired size is obtained.

Motif

Motif is a graphical system which resides on a layer above X Windows. There are three parts to the Motif package: the Motif window manager, the Motif interface compiler, and the Motif library. NEKTON makes use of only the Motif library, which is part of the executable programs PRENEK and POSTNEK. Motif is becoming very popular as a standard for layout and organization of graphical user interfaces. The principle Motif features used by the NEKTON interface are push buttons, toggle buttons, scrollbars, text input windows, and pulldown menus.

Push buttons are labels surrounded by boxes that bound the labels and give a three dimensional appearance. Pushing the button is accomplished by pushing and releasing the left mouse button while the cursor is inside the box. Pushing a button causes some action in the interface.

Toggle buttons are similar to push buttons, except that rather than performing a single action, it permanently changes a state until the toggle button is pushed again. A toggle button is used for the NEKTON help utility. When the help toggle button is pushed, it changes the state of the menu operation so that subsequent choices from menus give a help message rather than invoke the menu function. The help function is turned off by a second push to the help toggle button.

Text input windows are used to input strings of text into the interface and can be used as an alternative to the calculator push buttons to enter numbers into the keypad.

The interaction of the mouse and keyboard for text input is similar to that of a MacIntosh. A single mouse click can be used to locate the cursor to edit a text string. A double mouse click can be used to automatically delete the current text and replace it with the keyboard input. Trailing and preceeding blanks are automatically stripped.

The NEKTON user interface uses scrollbars to scroll through the messages which appear in the left part of the menu window and to scroll through zoomed mesh and output in the drawing/plotting window. The scrollbar can be dragged by holding the mouse button down with the cursor on the scrollbar and moving the cursor. Drawing area contents can be scrolled and clipped in real time, but hidden areas are redisplayed only when the mouse button in released. This refresh requires redrawing of the screen, so for large or complex meshes the mouse button should released only when the final scroll location is reached. A scrollbar can alternately be moved a fixed increment by pushing the arrow buttons on either of its ends. The convention used is that pulling a scrollbar down results in the contents of the window moving upward. This can be made more intuitive using the analogy of viewing the drawing area through a pair of binoculars; moving the binoculars (scrollbars) downward moves the viewing port (window) upward past a stationary background (drawing area).

Pulldown menus are attached to a main menu bar at the top of the menu window in POSTNEK. On this menu bar are a row of cascade buttons. Attached to each cascade button is a pulldown menu, which appears when the cascade button is pushed. On the pulldown menu are more buttons which are either normal push buttons, which invoke some function, or cascade buttons, which generate additional pulldown submenus. An rightward arrow appears on buttons which are cascade buttons to inform the user that pushing the button will cause a pulldown to appear at its right. A function is not actually invoked until the user releases the mouse button with the cursor on the push button. No action is taken if the button is released with the cursor away from the menu, so it is very easy to abort without doing anything. Thus, a pulldown menu system gives the user the opportunity to scan through and explore the entire menu structure without actually invoking any function.

For users who have interest in Motif beyond NEKTON usage, a complete description of Motif usage is in the Motif User's Guide and Motif Style Guide. A comprehensive description of the progamming calls and conventions is in the Motif Programmer's Guide and Motif Reference Manual. For users who are interested in programming their own Motif applications, the source code and demo problems are available at a nominal cost. All of these can be obtained from the Open Software Foundation at 11 Cambridge Center, Cambridge, Ma. 02142, (617)621-8700.

4.2 Sample Sessions

To demonstrate the operation of the NEKTON 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 editting, knowledge of simple Fortran, and familiarity with file structure and operating system commands is helpful.

4.2.1 Notation

In the sample sessions described below, 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 leftmost button (yellow button). Entries enclosed within curly brackets refer to text strings that are to be typed in by the user followed by a carriage return (or selecting the [OK] menu button with the mouse). For example, {problem1} means that the user should type in the the eight characters in the string "problem1" and then type a carriage return.

Entries of the form (M/*LETTERS number.letter*) refer to mouse inputs of points in the mesh drawing area. Here *LETTERS* refer to which mouse button is to be pushed, the leftmost button (LB), the middle button (MB), or the rightmost button (RB). *Number.letter* refers to a location marked on a figure, 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, **Enter Session Name:** 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 in the menu area. For example, [**READ PARAMETER**] corresponds to the READ PARAMETER menu, which is the first menu displayed by PRENEK. In POSTNEK there is only one (horizontal) menu bar, denoted in the following as [**... POSTNEK ...**], and the individual menu items are selected using pulldown menus. A complete description 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 leftmost button (LB), middle button (MB), and rightmost button (RB) in PRENEK and POSTNEK is summarized below.

PRENEK

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

General purpose: used for all mouse inputs except those described below.

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

RB - 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.

POSTNEK

LB - Used for all mouse inputs.

MB- Not used.

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

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, respectively. The top and bottom boundaries are insulated.

PRENEK

The session begins with the user typing ”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.

Enter Session Name:

{problem1} ! See Fig. 4.2.

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 postprocessing. For future bookkeeping purposes, all files related to this session will contain the prefix *problem1.* Note that the case of these user inputs is important with UNIX and irrelevant with VMS.

PRENEK now presents the first menu, the READ PARAMETER menu as in Fig. 4.3. This figure, and all screen display figures in this manual are screen dumps from a DECstation onto a POSTSCRIPT printer. A more complete description of a particular menu can be found 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]

[EQUATION TYPE]
Now we set the switches illustrated in Fig. 4.4, 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 in the comment area.
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]
CURRENT SETTING: STEADY TWO DIMENSIONAL STOKES
[FLUID FLOW]
CURRENT SETTING: STEADY TWO DIMENSIONAL
[HEAT TRANSFER]
CURRENT SETTING: STEADY TWO DIMENSIONAL HEAT TRAN
[ACCEPT CURRENT SWITCHES] ! We accept the current switch settings.

[CONVECTION]
Here, we get the option of choosing steady heat transfer with or without convection, see Fig. 4.5. The default is conduction heat transfer, and we accept the current switches.
[ACCEPT CURRENT SWITCHES]

[INPUT VALUES]
The input parameters are split into two screens, and are initially assigned default values. We can either accept these default settings, or alter one or more of them. In the first menu, illustrated in Fig. 4.6, we modify the parameter CONDUCT (the thermal conductivity) from its default value of 1.0 to 2.0. We do this by first positioning the mouse input in the area where the numerical value of CONDUCT is displayed, we then delete the 1 using the delete key from the keyboard, and finally type in the value 2. In general, the keyboard is used to input new parameter values.

The parameter NORDER, the number of nodes in each spatial direction in each macro-element, is left unchanged, and we proceed to the next set of parameters. [OK]

[INPUT VALUES]
Here we accept the default settings, see Fig. 4.7. 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, and we will retain the default convergence criterion for the iterative solvers.
[OK] ! This completes the setting of the input parameters.

Note that we set far fewer parameters than those described in the INPUT VALUES menu in Chapter 8; in particular, 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 INPUT VALUES menu.

[CENTRAL]
The CENTRAL menu (Fig. 4.8) 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 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. Normalized 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. 4.9. 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.
[ACCEPT SCALE FACTORS]

[**BUILD**]
In the build menu, we will first enter the quadrilateral element in Fig. 4.10. *PRENEK requires that the corners be entered in counter-clockwise order.*
[ADD ELEMENT]
Corner I >
(M/LB 4.10.a)
Corner II >
(M/LB 4.10.b)
Corner III >
(M/LB 4.10.c)
Corner IV >
(M/LB 4.10.d)

A gray rectangular macro-element should appear with the number "1" in the center similar to the element in Fig. 4.10. If the element does not look like that in Fig. 4.10, 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 material property menu.
[ACCEPT MESH]

[**MATERIAL PROPERTY**] ! See Fig. 4.11.
[ACCEPT CONST PROP] ! We will accept the constant material properties specified in the input parameter menu.

[**MIDWAY BREAK**] ! See Fig. 4.12.
[SET BCs] ! We won't stop for a break.

[BOUNDARY CONDITION]
Here the flashing side at the bottom of element 1 in Fig. 4.13 prompts us to select a boundary condition from the menu.
[INSULATED]

The side quits flashing, returns to its original color, and the character "I" is drawn near the side's center signifying that the requirement for a boundary condition has been satisfied (see Fig. 4.13). The BOUNDARY CONDITION 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. 4.14.
[CONSTANT]

A keypad now appears in the menu area, see Fig. 4.15, signalling that it is waiting for input. (KP/1)
1
(KP/R)
[INSULATED]

We next 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. 4.16. If we are satisfied with the boundary conditions we proceed to the options menu.
[ACCEPT BOUNDARY CONDITIONS]

[OPTIONS]
In Fig. 4.17 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.
Session problem1
Compiling user-defined source code ! The file *problem1.user* can contain fortran statements which describe material properties, boundary conditions, initial conditions, and forcing functions. In this simulation we used constants rather than fortran statements, but the routines in *problem1.user* must nonetheless be combined with other routines to make the file *problem1.f*. This file is then compiled and linked with NEKTON.
Linking with N52D
Start of simulation
Temperature/Passive scalar solution
8 Helmholtz solver: 0.35027E-1 0.5236E-1
Simulation successfully completed
End of problem1 simulation

This message states that there was a successful compilation and linking of the session, that 8 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 .3502E-1, and that the simulation ended successfully without incident. The exact 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
problem1 ! Default session name.
[OK]
Here we accept the default session name. The mesh is read from the file *problem1.rea* and is brought up along with the POSTNEK menu bar, see Fig. 4.18.

[**... POSTNEK ...**]
[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, and a color bar quantifies the values of the temperatures along the isotherm lines that separate the color levels.
[CLEAR] ! Clears the plotting area.

We will next modify the plot format to get a profile plot of the temperature along a line. This can be done by selecting from a set of pulldown-menus.
[**... POSTNEK ...**]
[SET PLOT FORMAT]
[X-Y PLOT]
[PROFILE] ! See Fig. 4.19.
During the selection process, the leftmost button is pressed down, and the selection becomes active once this button is dropped.

We next select a location for the profile.
[**... POSTNEK ...**]
[SET LOCATION]
[LINE]
Enter 1st Endpoint:
(M/LB 4.20.a)
Enter 2nd Endpoint:
(M/LB 4.20.b) ! See Fig. 4.20.

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 inside the menu area. PRENEK interprets this as a request to reject the graphical mouse input and choose keypad input.
[... POSTNEK ...]
[PLOT]

A profile of the temperature vs. distance along the line segment is plotted on the screen as shown in Fig. 4.21. 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 in the comment area.

Note the procedure by which we obtained our plot: We began at the POSTNEK menu bar, went into submenus (pulldown menus) 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 a plot. This system allows for the most flexibility in displaying results.

We now move on to a more complex example.
[EXIT]
[CONFIRM] ! The user is asked to confirm that the postprocessing session is over.

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

PRENEK

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.

<u>Enter Session Name:</u>
{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 UNIX system, PRENEK will append the old filenames with a “~” character, allowing for one duplicate SESSIONNAME before

destroying any data. In a VMS operating system, this will result in the old files being "buried" under new files that have higher version numbers. However, either of these situations requires careful file bookkeeping. To avoid confusion, duplicate SESSIONNAMEs should be avoided.

PRENEK now 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 *problem1.rea* 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.
<u>**Enter Session Name:**</u>
{problem1} ! We will read the parameters from the file *problem1.rea*.

Control skips the EQUATION TYPE and INPUT VALUES 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 *problem1*, but instead we choose to change the equation type and alter some of the input parameters.
[ALTER PARAMETERS]

[**EQUATION TYPE**] ! See Fig. 4.22.
<u>**CURRECT SETTING: STEADY TWO DIMENSIONAL HEAT TRANSFER**</u>
[UNSTEADY]

The menu item "UNSTEADY N" is chosen and the menu reappears with the item toggled to "UNSTEADY Y".
<u>**CURRECT SETTING: UNSTEADY TWO DIMENSIONAL HEAT TRANSFER**</u>

[FLUID FLOW]

This flag turns on the fluid flow calculation in a manner similar to the unsteady flag.
CURRECT SETTING: UNSTEADY TWO DIMENSIONAL STOKES AND HEAT TRANSFER
[ADVECTION]

The advection flag is turned on, i.e., we will solve a Navier- Stokes problem rather than a Stokes problem.
CURRECT SETTING: UNSTEADY TWO DIMENSIONAL NAVIER-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).

We now accept the current settings, which is to solve an unsteady two- dimensional Navier-Stokes problem with convective heat transfer, using the split formulation.
[ACCEPT CURRENT SWITCHES]

Next, 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. 4.23.
[**CONVECTION**]
[ACCEPT CURRENT SWITCHES]

Lastly, we have a choice as regards the treatment of the advection term, Fig. 4.24. We choose the standard semi-implicit scheme (see Section 3.2), and proceed to the INPUT VALUES menu.
[**CHARACTERISTICS**]
[ACCEPT CURRENT SWITCHES]

[**INPUT VALUES**] ! See Fig. 4.25.

Here we modify the default values using the keybord as in *problem1*. The molecular viscosity, VISCOS, is set to 0.01, and the thermal conductivity, CONDUCT, is set to 0.01. We keep the default value for ρC_p to 1.0, the order of each element, NORDER, to 5, and the order of the temporal scheme, TORDER, to 1.

In the the second screen of the INPUT VALUES menu, Fig. 4.26,

we set NSTEPS, the number of time steps, DT, the value of the time step, and IOSTEP, the interval between data dumps. We will turn on natural convection by giving a nonzero value for BETAG (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, and we will keep the default values for all other parameters.

[**CENTRAL**]
We are now back to the CENTRAL menu. Assuming we are satisfied with the current status of the equation type and input 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 Session Name:
problem1

We are prompted for the name of the session from which to read the mesh and subsequent data. The default filename is the same as the filename from which we just read the input parameters. This is what we want here, so we accept the current status.
[OK]

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

We now have the opportunity to either modify or accept the mesh, material 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 currrent 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.

[**ACCEPT/REVIEW**]
[REVIEW/MODIFY]

This puts us into the BUILD menu. All the 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. 4.27.

We start with the ZIPPER command. This command splits one or more elements.
[ZIPPER]
Push yellow button to enter element to be split:
(M/LB 4.27.a)
Enter a side of this element that is to be split:
(M/LB 4.27.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 split factor, s, to be 0.5.
Enter Value:
(KP/0)
0
(KP/.)
0.
(KP/5)
0.5
(KP/R)

We next split the two elements vertically.
[**GLOBAL REFINE**]
[ZIPPER]
Push yellow button to enter element to be split:
(M/LB 4.27.c)
Enter a side of this element that is to be split:
(M/LB 4.27.d)

We again split them evenly.
Enter Value:
(KP/0)
0
(KP/.)
0.
(KP/5)

0.5
(KP/R)

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]

[**BUILD**]
[CURVE SIDES] ! We will next curve two of the sides on the mesh boundary.

[**CURVE SIDES**] ! See Fig. 4.28.
[MAKE CIRCLE]
Enter element side:
(M/LB 4.28.a)
Enter circle radius for side of element *
>0 for convex element, <0 for concave
(KP/0)
0
(KP/.)
0.
(KP/7)
0.7
(KP/R)

The result should look like that in Fig. 4.28. Continuing onto the next side,
[MAKE CIRCLE]
Enter element side:
(M/LB 4.28.b)
Enter circle radius for side of element *
>0 for convex element, <0 for concave
(KP/-0.7R)

The negative radius corresponds to a concave curve, see Fig. 4.29.
[**CURVE SIDES**]
[BUILD MENU]

[**BUILD**]
[ACCEPT MESH] ! The mesh is completed.

PRENEK next brings us to the material property menu.
[**MATERIAL PROPERTY**]
[ACCEPT CONST PROP] ! We accept the current status.

[**MIDWAY BREAK**]
[SET BCs]

PRENEK now requests entry of boundary conditions. Had we kept the equation type from 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.**]
[ACCEPT BOUNDARY CONDITIONS]
[**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]
[INSULATED]
[TEMP]
[**FORTRAN FUNCTION**]
[CONSTANT]

(KP/1)
1
(KP/R)

To avoid keying in the temperature boundary condition again, we use the rightmost button (blue button) to copy the boundary condition from the previously entered side.
(M/RB) ! Place cursor close the side where the $T = 1$ was set.
[INSULATED]
[INSULATED]
[TEMP]
[**FORTRAN FUNCTION**]
[CONSTANT]
(KP/0)
0
(KP/R)
(M/B) ! Place cursor close the side where the $T = 0$ was set.
[**MODIFY B.C.**]
[ACCEPT BOUNDARY CONDITIONS]

[**OPTIONS**]
We will define two history points, see Fig. 4.30.
[HISTORY]

[**HISTORY**]
[ENTER POINT]
(M/LB 4.30.a)
[ENTER POINT]
(M/LB 4.30.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
$

NEKTON

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:
problem2 ! Default session name.
[OK] ! We accept the suggested session *problem2* for post-processsing.

[... POSTNEK ...]
The horizontal POSTNEK menu bar is displayed, and the mesh is displayed in the plotting area 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. 4.31.

We now change the quantity to be plotted from velocity to temperature, and request a plot.
[SET QUANTITY]
[TEMPERATURE] ! Use pulldown menu.
[CLEAR] ! Clear the plotting area.
[PLOT]

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 ...]
[SET ATTRIBUTE]
[SET RESOLUTION]
[COLOR FILL] ! Use pulldown menus.
# Bands/Element
10 ! Default value is 10.
(KP/DD25R) ! Replace 10 with 25.
[PLOT]

The temperature (a two-dimensional scalar quantity) can also be displayed using the contour plot.
[... POSTNEK ...]
[SET PLOT FORMAT]
[SURFACE PLOT]
[SCALAR PLOT]
[CONTOUR LINES] ! Use pulldown menus.
[PLOT] ! See Fig. 4.32.

We set the time for the next plot to be at the first dump, at an earlier earlier time of about 5.5.
[... POSTNEK ...]
[SET TIME]
(M/LB 4.31.a)
[PLOT] ! See Fig. 4.33.

The difference between the two temperature plots is due the diffusion and convection of heat from the hot right side (Temperature=1.0) throughout the domain whose initial temperature was 0.0.

We next plot the temperature at a point as a function of time. To do this, we change the plot format.
[... POSTNEK ...]
[SET PLOT FORMAT]
[X-Y PLOT]
[HISTORY] ! Use pulldown menus.
[PLOT]

After the screen clears itself, we are presented with a time history of temperature, see Fig. 4.34, 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.
[CONTINUE]

This calculation began with default (zero) initial conditions. Had we wished 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 ...]
[EXIT]
[CONFIRM] ! Confirm the end of post-processing session.

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 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 preprocesor. 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 characteristic 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 execuable 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 philisophy 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 macroelemental 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 transient is avoided by starting with a conduction solution. Hence, the bouyant forces driving the natural convection begin immediately from the initial temperature field.

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 (this strategy is automatically accelerated if the user has requested a steady solution). 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 will typically 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. If a steady solution has been requested, the ususal time stepping procedure is automatically accelerated using extrapolation techniques.

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 10-15 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 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. There are four major steps to creating the data in PRENEK :

Step 1. Input the equation type that describes the physics of the problem to be solved, together with associated physical parameters (material properties) and numerical parameters (discretization and solver specifications).

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

Step 3. Input the boundary conditions.

Step 4. Input the initial conditions (for transient analysis) and other option specifications.

In Step 1, first the session name associated with the problem is assigned. Next the type of analysis is defined by setting the equation type and parameter values using the SET EQUATION TYPE and INPUT VALUES menus. New sets of input can be entered interactively or data from a previous session can be reused.

Between Step 1 and Step 2 is the CENTRAL menu, in which options are provided to show the input parameters, or to return to Step 1 to modify the equation type and/or numerical parameters, or to continue on to Step 2.

In Step 2, the mesh is either input interactively or read in from a previously created file generated either by PRENEK or other software packages. If the mesh is read in from a file generated by PRENEK, then the boundary conditions are also read and reused, provided that the equation type of the old session is the same as the current one. If the mesh is imported from a file generated by other software packages, then the mesh building process is assumed complete and no change of the mesh geometry is allowed.

Between Step 2 and Step 3 are the MATERIAL PROPERTY MENU in which different sets of properties are defined, and a MIDWAY BREAK MENU which allows an exit of PRENEK with the saving of all input data.

In Step 3, the boundary conditions for all fields (fluid velocity, temperature, passive scalar 1,...,etc.) are defined.

In Step 4, the initial conditions and other option specifications are input. Upon exit of Step 4, data files pertained to the current session are written, as described in the following.

5.1.2 File Structure

PRENEK generates data files that are later used by NEKTON and POSTNEK. 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 interactively input the geometry and the 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>
.
.

Possible user replies to each prompt are

[M/LB -] ! 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/RB -] ! 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.

Note that the current position of the mouse cross hair in user coordinates is given at the upper left corner of the screen whenever the cross hair is inside the grid area.

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

5.3 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 refining 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.

5.3.1 Two-Dimensional Mesh refinement

SPIDER WEB

The SPIDER WEB command splits a single element into 5 elements, see Fig. 5.1. 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. 5.1. 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. 5.2. 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.

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. 5.3 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. 5.4. 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. 5.4a. When this is done, PRENEK responds with a warning. This is because the mesh in Fig. 5.4b 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. 5.4c.

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 similiar 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.

ZIPPER

The ZIPPER command splits one or more elements into two, see Fig. 5.5. 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" intiated 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.

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.

If the mesh is read in from a file generated previously using PRENEK, then boundary conditions associated with the problem of the previous session can also be transfer to the current session. When an element is 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.

5.3.2 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.

5.4 Using the Set Variable Properties Menu

The SET VARIABLE PROP 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 VARIABLE QUANTITIES" menu.

Fluid Flow Only

Fluid elements with different material properties are assigned group number:
$0, -1, -2, ..., -5$, for which the order must be followed without skipping. Group 0 is the group for which the properties (density and viscosity) default to the constant values entered in the INPUT VALUES menu (Step 1 in Section 5.1.1). Alternatively, properties which are fortran functions of other variables (see Section 5.5) can be defined for this group. Properties for nonzero groups can be defined as either constant or fortran functions. Note that the density must be *constant* within the same identity of fluid, even though different constants can be assigned to different identities of fluid when there are more than one fluid component. In a multi-fluid system, NEKTON requires that all spectral elements associated with a fluid component belong to the same unique material property group.

Heat Transfer Only

Conduction elements with different material properties are assigned group number: $0, 1, 2, ..., 10$, for which the order must be followed without skipping. Group 0 is the group for which the properties (conductivity, specific heat and internal heat source) default to the constant values entered in the INPUT VALUES menus (Step 1 in Section 5.1.1). Alternatively, properties which are fortran functions of other variables (see Section 5.5) can be defined for this group. Properties for nonzero groups can be defined as either constant or fortran functions. For heat transfer with phase change all spectral elements associated with the liquid phase must be assigned to group 0.

Fluid Flow/Heat Transfer with only Fluid Phase

The convention is exactly the same as the "Fluid Flow Only" case, except that the required fluid properties now include conductivity, specific heat and internal heat source.

Fluid Flow/Heat Transfer with both Fluid and Solid Phases

Fluid elements with different material properties are assigned group number:
$0, -1, -2, ..., -5$ and solid elements with different material properties are assigned group number: $1, 2, ..., 10$, for which the order must be followed without skipping. Group 0 is the default group for the fluid elements for which the properties (density, viscosity, conductivity, specific heat and internal heat source) default to the constant values entered in the INPUT VALUES menus (Step 1 in Section 5.1.1). Alternatively, properties which are fortran functions of other variables (see Section 5.5) can be defined for this group. Any nonzero group properties can be defined as either constant or fortran function of other variables, see Section 5.5. For heat transfer with phase change all spectral elements associated with the liquid phase must be assigned to group 0.

5.5 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 x 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
PA	ambient pressure
SI2,SI3	second and third strain-rate invariants
FFX,FFY,FFZ	forcing function, x, y, and z components
TEMP	temperature
PS1,PS2...	passive scalar number 1, 2, ...
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
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:

$$SI2 = e_{xx}e_{yy} + e_{yy}e_{zz} + e_{zz}e_{xx} - e_{xy}^2 - e_{yz}^2 - e_{xz}^2 \tag{5.1}$$

$$SI3 = e_{xx}e_{yy}e_{zz} + 2e_{xy}e_{yz}e_{xz} - e_{xx}e_{yz}^2 - e_{yy}e_{xz}^2 - e_{zz}e_{xy}^2 \tag{5.2}$$

5.6 BOUNDARY CONDITION Menu

After the MATERIAL PROP 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 changing the color of the side. 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, the “mouse keypad” will appear in the manual area. 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. 4.15):

[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 velocity and traction boundary conditions whose components are normal and/or tangential to the boundary, 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. 5.6. Note that the normal vector $\mathbf{n}$ 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 $\mathbf{t}_1$ is obtained using : $\mathbf{t}_1 = \mathbf{t}_2 \times \mathbf{n}$. 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 $\mathbf{t}_2$ is obtained using: $\mathbf{t}_2 = \mathbf{n} \times \mathbf{t}_1$.

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.

5.7 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 Table 5.1. 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.

For heat transfer problem with change-of-phase, the default values for various boundaries are shown in Table 5.2. 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.

For special modeling purposes, these defaults can be overidden 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 macro-element, 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 colume 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	7	≤ 100	2-D
Not Split	5	≤ 100	2-D
Not Split	7	≤ 100	2-D
Not Split	5	≤ 40	3-D
Not Split	7	≤ 40	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 sesssionname*} 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 POSTNEK:

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 using 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 are 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.

sessionname.lgl contains data for converting from local to global numbering of the spectral element collocation points.

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

The structure of POSTNEK centers around a Motif menu bar appearing at the top of the menu window. To make a plot using POSTNEK, the user chooses the PLOT item from the menu bar. To change aspects of the plot, the chooses from one of the items in the pulldown menus and submenus. These pulldown menus and submenus alter the values of of the parameters controlling various aspects of the plot. The function of each top level pulldown is described below.

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 SET 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 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, and surface from which to plot data. All three locations can be simultaneously specified; the current setting of the PLOT FORMAT will determine which is plotted.

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

The SET SCREEN menu governs aspects of the visual layout of the screen and of the quantities plotted on the screen, including zoom, orientation, and whether axes and internal element boundaries are displayed.

7.2 Hardcopy

Plots generated by the user in a POSTNEK session with the Dump Screen button are dumped inta a file sessionname.plt using the PostScript standard.

To obtain a hardcopy of these plots the user types:

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

On Unix systems, the command can be issued during the postprocessing session; all screen dumps up to that time will be printed. On VMS systems, the dumps can be printed only after exiting POSTNEK.

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 form the menu is listed along with its definition and usage.

These menus use a mouse 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 a box above the menu items. For pulldown menus, the menu name is the name of the menu's parent cascade button (the button pushed that caused the pulldown menu to appear). An item is chosen from the menu by pushing and releasing the leftmost 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.
REVIEW/MODIFY

Menu function: The menu appears after data (either mesh or boundary condition) is read from a file previously generated by PRENEK. It enables the user to either review and/or modify the data, or to accept the data and skip over the modification process.

ACCEPT MESH

Accept the mesh data.

ACCEPT B.C.

Accept the boundary condition data.

REVIEW/MODIFY

Activate appropriate menus for review/modification.

ADD/DELETE FACES Menu

END OBJECT
ADD FACE
DELETE FACE

Menu function: This menu is used in conjuction with the ADD OBJECT and MODIFY OBJECT options in the OBJECT menu. It enables the user to add or delete elemental edges (2D) or faces (3D) to and from an existing object.

END OBJECT
Return to the **OBJECT** menu.

ADD FACE
Add a face (or edge) to the selected object.

DELETE FACE
Delete a face (or edge) from the selected object.

BOUNDARY CONDITIONS

(Main Menu for Fluid Boundaries)

PERIODIC
WALL
VELOCITY
OUTFLOW
OUTFLOW/N
VELOCITY (LOCAL)
SYMMETRY
AXIS
STRESS B.C.s
ZOOM
SET ENTIRE LEVEL

(Additional B.C. Menu for Stress Formulation)

MAIN B.C. MENU
MOVING WALL
FREE SURFACE
MELTING FRONT
TRACTION
TRACTION (LOCAL)
SHEAR
SHEAR (LOCAL)

(Main Menu for Temperature Boundaries)

PERIODIC
INSULATED
FLUX
TEMP
CONVECTION
OUTFLOW
RADIATION
ZOOM
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 **STRESS B.C.s** submenu is only activated 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 right-most mouse button acts as a "repeat" button. Pushing the right-most 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". Note that duplication of boundary conditions using the right-most mouse button is not applicable for 3D import meshes.

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 $(u_x, u_y, u_z, p, T, \ldots)$ 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 in the next item.

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 nonsplit/nonstress formulation, the boundary conditions are

$$-p + \mu \frac{\partial u_n}{\partial n} = 0 \tag{8.1}$$

$$\frac{\partial u_t}{\partial n} = 0 \tag{8.2}$$

For the nonsplit/stress formulation, the conditions are

$$-p + 2\mu \frac{\partial u_n}{\partial n} = T_n \tag{8.3}$$

$$\frac{\partial u_t}{\partial n} + \frac{\partial u_n}{\partial t} = 0 \tag{8.4}$$

where T_n is the normal traction. It is assumed that under proper usage, the term $\mu \partial u_n / \partial n$ is negligible at the boundary and therefore the normal traction T_n corresponds to the negative ambient pressure (Pa), where Pa is input either as constant or as a fortran function. Note that if $\mu \partial u_n / \partial n$ is large compared with p due to inappropriate modelling, then the resulting pressure solution can differ significantly from the imposed ambient pressure.

For the split formulation, the boundary conditions are

$$p = 0 \tag{8.5}$$

$$\frac{\partial u_t}{\partial n} = 0 \tag{8.6}$$

For temperature and passive scalars the boundary condition is

$$\frac{\partial T}{\partial n} = 0 \tag{8.7}$$

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, in the case of the nonstress formulation, restricted 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 nonsplit/nonstress formulation, the conditions are

$$-p + \mu\frac{\partial u_n}{\partial n} = 0 \tag{8.8}$$

$$u_t = 0 \tag{8.9}$$

For the nonsplit/stress formulation, the conditions are

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

$$u_t = 0 \tag{8.11}$$

where T_n is the normal traction. It is assumed that under proper usage, the term $\mu\partial u_n/\partial n$ is negligible at the boundary and therefore the normal traction T_n corresponds to the negative ambient pressure (Pa), where Pa is input either as constant or as a fortran function. Note that if $\mu\partial u_n/\partial n$ is large compared with p due to inappropriate modelling, then the resulting pressure solution can differ significantly from the imposed ambient pressure.

For the split formulation, the boundary conditions are

$$p = 0 \tag{8.12}$$

$$u_t = 0 \tag{8.13}$$

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. 5.6.

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.

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.

ZOOM

The zoom option activates the **ZOOM** menu which enables the enlargement of the mesh picture with respect to the plotting frame.

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.

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 MOVING WALL option is available only if the moving geometry option is selected in the EQUATION TYPE menu.

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. 5.6. Note that the FREE SURFACE option is available only if the moving geometry option is selected in the EQUATION TYPE menu.

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 MELTING FRONT option is available only if both the moving geometry and the heat transfer options are selected in the EQUATION TYPE menu.

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. 5.6.

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. 5.6. Thus, for SHEAR/LOCAL boundaries, the input shear traction components are automatically assumed to be in the tangential $t_1 - t_2$ directions.

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

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

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 left-most mouse button signifies input of a corner which will be located at the nearest intersection of the "graph paper" screen grid lines. Hitting the right-most 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 mouse 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 left-most mouse button with the cursor in the menu area and the keypad will appear. The user then keys in the coordinates using the mouse.

If, in response to the menu prompt, a point is entered in the mesh area, the menu assumes that ADD ELEMENT was chosen and that the 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.

ACCEPT MESH

The ACCEPT MESH item should be selected when all mesh operations such as modifications, deletions, curvature, and refinements, are completed.

ADD ELEMENT

The ADD ELEMENT item is used to create a new element. A new element is defined by four corners input in a *COUNTERCLOCKWISE* 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. Provides 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.

ZOOM

The zoom option activates the **ZOOM** menu which enables the enlargement of the mesh picture with respect to the plotting frame.

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
IMPORT BFC CASE 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 **EQUATION TYPE** menu, which is followed by the **INPUT VALUES** 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.

SHOW PARAMETERS

Shows the numerical values of the parameters.

BUILD INTERACTIVELY

Activates the menus related to building a mesh. The mesh will be built interactively from “scratch” from user inputs.

BUILD FROM FILE

Upon the selection of this option PRENEK will prompt for the name of the session from whose file the mesh data is to be read in. Next the ACCEPT/REVIEW menu will be activated which allows for modification of the input 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 by the I-DEAS package. PRENEK assumes that the import mesh will require no modifications and the ACCEPT/REVIEW menu will be skipped.

IMPORT BFC CASE FILE

Use mesh generated by the PRE-BFC package. PRENEK assumes that the import mesh will require no modifications and the ACCEPT/REVIEW menu will be skipped.

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.

CHARACTERISTICS Menu

ACCEPT CURRENT SWITCHES
CHARACTERISTICS

Menu function: This menu enables the user to select treatment of the advection contribution using the method of characteristics.

ACCEPT CURRENT SWITCHES

Accept the switch below for using characteristics.

CHARACTERISTICS

The switch is toggled between Yes (Y) and No (N) each time a mouse click is applied inside the box.

CHOOSE ELEMENT Menu

UP LEVEL
DOWN LEVEL
CHOOSE ELEMENT
MAIN MENU

Menu function: Enables the user to assign to elements in *three-dimensional* meshes with multiple levels different material property set numbers.

UP LEVEL

Move up one level and draw the mesh for that level.

DOWN LEVEL

Move 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 **SET VARIABLE PROPERTY** menu.

CONVECTION Menu

ACCEPT CURRENT SWITCHES
WITH CONVECTION
TEMPERATURE MESH

Menu function: This menu enables the user to include or exclude convection contributions in the heat or passive-scalar transport equation.

ACCEPT CURRENT SWITCHES

Accept the switches below.

WITH CONVECTION

The switch is toggled between Yes (Y) and No (N) each time a mouse click is applied inside the box.

TEMPERATURE MESH

This option is only applicable to passive scalars which can be defined on either the fluid or the temperature mesh. The switch is toggled between TEMPERATURE MESH and FLUID MESH each time a mouse click is applied inside the box.

CURVE SIDES Menu

BUILD MENU
STRAIGHTEN CURVE
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 CURVE

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 (bulging outward) and a negative radius corresponding to a concave arc (bulging inward). For sides which are 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

The sine wave must be in the X-Y plane given in the following form: $Y = Y_o + A\sin(\alpha X)$, where Y_o is the off-set distance along the Y-axis, A is the amplitude and α is the wave number. MAKE SINE WAVE prompts for the element side and the above three parameters.

EQUATION TYPE Menu

ACCEPT CURRENT SWITCHES
DIMENSION
UNSTEADY
FLUID FLOW
MOVING GEOMETRY
ADVECTION
STRESS FORMULATION
SPLIT FORMULATION
TURBULENCE MODEL
HEAT TRANSFER
ADDL PASSIVE SCALARS

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 click 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 **EQUATION TYPE** menu and move on to the **INPUT VALUES** menu.

ACCEPT CURRENT SWITCHES

Exit the **EQUATION TYPE** menu and proceed to the **INPUT VALUES** menu.

DIMENSION

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

UNSTEADY

Values: T or F.

FLUID FLOW

Values: T or F.

MOVING GEOMETRY

Values: T or F. Determines whether or not the geometry time-dependent.

ADVECTION

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

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: MI (mixing length) or N (none). Determines if the algebraic 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.

FLUID FORCING FUNCTIONS Menu

SPECIFY BODY FORCE
NO BODY FORCE

Menu function: This menu appears after the DRIVE FORCE option is selected in the OPTION menu.

SPECIFY BODY FORCE

This option enables the user to enter the Cartesian components of the body force per unit *mass* applied to the fluid as fortran functions. Note that the program will collocate the input body force vector with the fluid density, and the resulting force per unit *volume* vector is used in the computation, as inicated in Equation (2.1).

NO BODY FORCE

In this case the fluid body force is set to zero.

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 5.3

END GLOBAL REFINE

Returns to **BUILD** menu.

ZIPPER

Initiate a crack in a specified element that propagates to adjacent elements and beyond, see Fig. 5.5.

SPIDER WEB

Split one element into five elements, see Fig. 5.1. 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. 5.3. One call to PICTURE FRAME is equivalent to two calls to CORNER FRAME, Fig. 5.2, but requires less user input.

CORNER FRAME

Split one element into three elements, see Fig. 5.2.

SPLIT FLOOR

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

HISTORY Menu

MAIN MENU

ENTER POINT

UP LEVEL

DOWN LEVEL

VELOCITY

PRESSURE

TEMPERATURE

PASSIVE SCALAR

Menu function: Specify spatial coordinates and other field variables to be saved at each time step such that time-history plots can be made in POSTNEK. 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 as YES in the menu:

VELOCITY

PRESSURE

TEMPERATURE

PASSIVE SCALAR

Toggle on or off the variables to be included in the next ENTER POINT command. Note that only one passive scalar can be saved.

INITIAL CONDITION Menu

MAIN MENU
DEFAULT=0
FUNCTION
PRESOLVE
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.

PRESOLVE

Activates the **PRESOLVE** Menu. 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

Activates the **RESTART** Menu. The user can then enter session names of earlier simulations from which selected field variables will be reinstated as initial conditions.

INITIAL TIME Menu

DEFAULT INITIAL TIME

SET INITIAL TIME

Menu function: This menu is used in conjunction with the restart option described in the OPTION menu. The restart will always be continued from the last solution dump and this menu enables the reset of the solution time for the restart run.

DEFAULT INITIAL TIME

The solution time in the last dump is reinstated as initial time in the restart run.

SET INITIAL TIME

A new initial time is selected for the restart run.

INPUT VALUES Menu

VISCOS
RHOCP
CONDUCT
NORDER
TORDER
MIXLFAC

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

Menu function: The **INPUT VALUES** menu display 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 modify the numerical value, move the mouse screen cursor to the appropriate box. By clicking the mouse left-most button once, the existing numerical value is retained as an menu item, which can then be modified using the keyboard. If the mouse left-most button is clicked twice, the existing numerical value is nullified and a new value can be entered using the keyboard. If the parameters from a previous session is selected in the **READ PARAMETER** menu, the default values displayed will be those read from the old session data file.

If the default or read in parameters are to be reinstated after modifications have been made, click the left-most mouse button in the CANCEL box to start over again. When all the parameter values in the menu are properly set, click the left-most mouse botton in the OK box to exit.

VISCOS

The dynamic 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 5 × 5 × 5 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.

MIXLFAC

The maximum mixing length intensity factor used in conjunction with the algebraic turbulence model.

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 accelaration 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 $\mathbf{f}$.

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.

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 1the initial velocity in the unsteady simulation is zero, achieving a relative error of less than 1velocity (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.

INTEGRAL Menu

Name of objects defined

MAIN MENU

Menu function: Integral quantities such as lift/drag and heat transfer rate can be computed for any surface object at each time step. This menu enables the user to select the objects (which must have been defined already) for which integral quantities are computed.

Name of objects defined

The name assigned to each previously defined object appears in the menu as a separate menu item. The user chooses the objects for which integral quantities are computed by selecting the appropriate object items. Each selected object will have the character "I" preceeding its name.

MAIN MENU

Return to the OPTION menu.

MATERIAL PROPERTY Menu

ACCEPT CONST PROPS
SET VARIABLE PROPS

Menu function: This menu allows the user to define different sets of properties which can be constant or fortran functions of other variables.

ACCEPT CONST PROPS

This is the default case in which all elements will have constant properties whose values are input in the INPUT VALUES menu.

SET VARIABLE PROPS

This activates the SET VARIABLE PROPS menu in which all variable properties are defined, and all elements are assigned a property set number.

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.

MODIFY B.C. Menu

ACCEPT BOUNDARY CONDITIONS
MODIFY B.C.
SHOW B.C.
INTERNAL B.C.
ZOOM

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.

ACCEPT BOUNDARY CONDITIONS

Input of boundary conditions for the current field (velocity or temperature or passive scalars) at the current level is complete.

MODIFY B.C.

Change previously input boundary conditions.

SHOW B.C.

Display previously input boundary conditions.

INTERNAL B.C.

This option is applicable only when moving geometry is specified in the EQUATION TYPE menu.

In a multi-fluid system, immiscible fluid layers are defined by specifying the interfaces between different fluids as internal boundaries. NEKTON requires that all spectral elements associated with a fluid component belong to the same unique material property group.

Note that in a heat transfer analysis with fluid flow, the solid-liquid interfaces are *automatically* defined when the mesh is input. In a heat transfer analysis with change-of-phase but without fluid flow, all freezing/melting fronts are defined as internal boundaries. For heat transfer with phase change all spectral elements associated with the liquid phase must be assigned to group zero.

ZOOM

The zoom option activates the **ZOOM** menu which enables the enlargement of the mesh picture with respect to the plotting frame.

OBJECT Menu

MAIN MENU
DELETE OBJECT
MODIFY OBJECT
ADD OBJECT
UP LEVEL
DOWN LEVEL

Menu function: This menu enables 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.

UP LEVEL

Use in 3D geometry to move up to the next higher level.

DOWN LEVEL

Use in 3D geometry to move down to the next lower level.

OPTIONS Menu

EXIT
OUTPUT
DRIVE FORCE
HISTORY
INITIAL COND
INTEGRAL QUANTITY
OBJECT

Menu function: This menu contains a collection of optional features within **NEKTON** susch 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, in the **INPUT VALUES** Menu. The default is all variables relevant to the problem type.

DRIVE FORCE

Activates the **FLUID FORCING FUNCTION** Menu which enables the user to specify a driving force for a fluid problem (e.g. constant pressure gradient or fortran function).

HISTORY

Activates the **HISTORY** Menu which enables the user to specify one or more points in space at which to save output data at each time step. Note that in a fluid flow/moving gemetry 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

Activates the **INITIAL CONDITIONS** Menu which enables the user to specify initial conditions using the default, fortran function, presolve or restart options.

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

Activates the **OBJECT** Menu which enables the user to add, delete or modify an object.

OUTPUT Menu

MAIN MENU
COORDINATES
VELOCITY
TEMPERATURE
PS

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 staus 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 satisfies 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.

PS

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

PRESOLVE Menu

ACCEPT CURRENT STATUS
VELOCITY
TEMPERATURE
PASSIVE SCALAR NO.

Menu function: This menu is activated when the presolve option is selected in the INITIAL CONDITIONS menu. Appropriate field variables such as velocity and temperature are listed as menu items. The user selects the fields for which presolve is to be performed by toggling the switch in the appropriate boxes to YES or NO.

ACCEPT CURRENT STATUS

Accept current switch settings and return to **INITIAL CONDITIONS** menu.

VELOCITY
TEMPERATURE
PASSIVE SCALAR NO.

Toggle specification of presolve fields on or off.

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 **EQUATION TYPE** 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 **EQUATION TYPE** and **INPUT VALUES** menus and go directly to the **CENTRAL** menu. If modification of the parameters is desired, the user can return to the **EQUATION TYPE** and **INPUT VALUES** menus from the **CENTRAL** menu.

RESTART Menu

ACCEPT CURRENT STATUS

ADD A FILE

DELETE A FILE

INITIAL TIME

Menu function: This menu is activated when the restart option is selected in the INITIAL CONDITIONS menu. Field variables such as velocity and temperature from previous analyses can be reinstated as initial conditions, and the files (generated by NEKTON under old session names) from which field variables are to be read should be entered in this menu. The user can also reset the initial time.

After a file name is entered, a menu will be activated in which the file name appears as the top item, and all appropriate field variables such as velocity and temperature are listed as menu items. The user selects the fields for which restart is to be performed by toggling the switch in the appropriate boxes to YES or NO.

ACCEPT CURRENT STATUS

Accept current switch settings and return to **INITIAL CONDITIONS** menu.

ADD A FILE

Define a new file for restart.

DELETE A FILE

Delete a specific file which must have been defined previously.

INITIAL TIME

Activate the INITIAL TIME menu for resetting the solution time.

SCALE FACTOR Menu

ACCEPT SCALE FACTORS
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.

ACCEPT SCALE FACTORS

Accepts the input scale factors and exit the menu. This should be selected after the scale factors are defined using one of the three options.

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 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 separetely 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.

# DIV n	Change the number of radial ”spokes”

for the polar grid from its current value of ”n”.

dR = dr

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

DX = dx

DY = dy

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.

Xo = xo

Yo = yo

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 VARIABLE PROPERTY Menu

ACCEPT MATL PROPS
SELECT ELEMENT
SELECT MATL [0]
SET MATL PROPS
SHOW MATL PROPS

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. 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 **INPUT VALUES** menu.

ACCEPT MATL PROPS

Accept the current status of the materials properties and their assignment to appropriate elements and continue on to the next menu.

SELECT ELEMENT

Specify that an element belongs to the material group whose number currently appears in the SELECT MATL box.

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 the SELECT ELEMENT command, the SET MATL PROPS command, or the

SHOW 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 PROPS

Set the values of the currently-selected material group.

SHOW MATL PROPS

Display the values of the currently-selected material group.

ZOOM Menu

DEFINE RECTANGLE
ZOOM OUT
ZOOM X 2
ZOOM / 2
KEYPAD

Menu function: The zoom menu allows the user to ajust the size of the mesh with respect to the plotting screen. This option facilitates both mesh generation and boundary condition input when the element is very small.

DEFINE RECTANGLE

The user defines a new frame by entering with the mouse two points which are the opposing corners of the diagonal of the frame.

ZOOM OUT

Return the plotting frame to the original preset dimensions (zoom factor equals 1).

ZOOM X 2

Double the current zoom factor.

ZOOM / 2

Half the current zoom factor, subject to a minimum zoom fator of 1.0 (the plotting frame can not be reduced to a size smaller than the original preset dimensions).

KEYPAD

Enter the zoom factor using the mouse keypad.

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.

MAIN Menu

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

Menu function: This menu (menu bar) controls all other menus in **POSTNEK**. The user goes into submenus (pulldown menus) 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 selects [PLOT] to display the result.

EXIT

Exits **POSTNEK**

CLEAR

Clears all plot data from the screen and redraws a wire frame representation of the spectral element mmesh. 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

Determines which variable (velocity, temperature, pressure) is to be plotted.

SET LOCATION

Sets the spatial location of the plot.

SET PLOT FORMAT

Sets 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

Modifies one of the plot attributes, e.g. resolution.

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.

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

Set the plot format of the scalar plot to CONTOUR LINES, in which the scalar value of the quantity is represented by iso-lines. Currently this option only works for two-dimensional problems.

SET ATTRIBUTE Menu

SET RESOLUTION
SET ANNOTATION
SET COLORS
SET ARROW SIZE
SET FISHNET HEIGHT

Menu function: Adjust 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 ARROW SIZE

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 firshnet height is the constant of proportion.

SET COMPONENT Menu

SPEED
X
Y
Z
NORMAL TO PLANE

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

SPEED

Plot the magnitude of the vector quantity.

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.

SET LOCATION Menu

POINT
LINE
X-PLANE
Y-PLANE
Z-PLANE
ALL WALLS
ALL BOUNDARIES
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 appropiate for the current PLOT FORMAT.

POINT

Specify a point. This will be followed by a prompt for the coordinates of the point, which are entered either using the leftmost 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.

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. Applicable 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.

HISTORY POINT

Specify a point/quantity. This menu option is primarily designed for moving geometry problems, 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 a surface plot.

VALUE AT POINT

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

X-Y PLOT

Determine which type of $x - y$ plot format to use.

SURFACE PLOT

Determine which type of surface plot format to use.

SET QUANTITY Menu

DERIVATIVE
VELOCITY
PRESSURE
DIVERGENCE
STREAMFUNCTION
VORTICITY
TEMPERATURE

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.

DERIVATIVE

Set subsequent plots to be a spatial derivative (d/dx, d/dy, d/dz) of quantities, rather than the quantities themselves.

VELOCITY
PRESSURE
DIVERGENCE
STREAMFUNCTION
VORTICITY
TEMPERATURE

Choose quantity to plot.

SET RESOLUTION 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 will be displayed in the menu area. The user can keep this default value, or type in an integer using the displayed keypad.

FISHNET GRID
Modify the number of lines per element in the fishnet gird plots.

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

CONTOUR LINES
Modify the number of contour lines.

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

SET DRAW AXIS
GRID LATCH
SET VIEW
ZOOM
ELEMENT BOUNDARIES

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

SET DRAW AXIS

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

GRID LATCH

Toggle the grid latching feature when entering x-y coordinates with the mouse. The default is that the leftmost 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 rightmost button (vertex latching) will remain unchanged.

SET VIEW

Set 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.

ZOOM

Zoom the plotting area. See description in Section 8.1 (PRENEK Menus).

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 will be displayed. This feature is frequently usefull when generating a hardcopy display of streamlines past an object or any cases where it is desirable to show only the physical object.

STREAMLINE Menu

MAIN
NEW STREAMLINE
READ FROM FILE
OPTIONAL SETTINGS
REPLOT

Menu function: Plot one or more streamlines 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 *sessionname.str* (where *sessionname* is the current session name), the keyboard, or the mouse (for 2D only).

MAIN

Return to the POSTNEK menu bar (MAIN menu).

NEW STREAMLINE

Plot one or more streamlines. Before starting plotting, input parameters for the streamline integration is selected. The user can choose between default values, or set the parameters manually. In the former case, the user is requested to select a starting point for a new streamline, using the leftmost mouse button. Selecting a point outside the domain will cause POSTNEK to return to the STREAMLINE menu.

If the streamline parameters are set manually, the user will be prompted 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.

READ FROM FILE

Plot the streamlines based on input from the file *sessionname.str* generated prior to commencing the post-processing session. The necessary streamline information to be stored in this file is the same as for manual input of the streamline parameters (see menu item NEW STREAMLINE above). One line is stored per streamline to be plotted.

OPTIONAL SETTINGS

Set optional parameters for streamline integration. The user can choose between default values, or set them manually. The user can also establish a plane to plot the streamlines for 3D geometries; a plotting plane is selected by giving the coordinates for three points in the plane.

REPLOT

Within the same post-processing session, and upon subsequent reentry into the streamline plotting menu, the user can rapidly redraw all the streamlines which have been previously generated. This is particularly useful for 3D calculations where the user 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.

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
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

Plot streamlines corresponding to the currently selected time. The location of the streamlines are set after selecting the PLOT option from the POSTNEK menu bar (MAIN menu).

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 after selecting the PLOT option from the POSTNEK menu bar (MAIN menu). 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 only works for 2D problems.

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
- 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:

- 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 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 8 mbyte or less) the program must be recompiled with smaller arrays according to the installation procedure.

Valloc Failed. See "Not enough memory"

9.2 NEKTON Solver Difficulties

Compiler Errors. If NEKTON aborts during the "Compiling User-defined source code" 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 SESSIONNAME.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, wsmax, 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.

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^o of 90^o). 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 Postprecessor Difficulties

Not enough memory. This message appears on a 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 8 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, moving boundary, fluid layers, variable ρ and μ, and turbulence model. 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 moderatly 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 responsibity 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)
(A) Nonstress, Split ($\mathbf{P}_N/\mathbf{P}_N$)		X	X			X	X
(B) Nonstress, Uzawa ($\mathbf{P}_N/\mathbf{P}_{N-2}$)	X		X				X
(C) Stress, Uzawa ($\mathbf{P}_N/\mathbf{P}_{N-2}$)	X	X					
(D) Characteristics					X		
(E) Moving geometry				X		X	
(F) Steady state solver	X				X		(1)
(G) 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 **EQUATION TYPE** Menu.

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

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

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

(E) Moving geometry is set by selecting moving geometry as "Y" in the **EQUATION TYPE** Menu.

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

(G) Variable fluid properties are set in the **SET 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 (except for steady, laminar, two-dimensional problems in fixed geometries).

[

Index

]