Purdue University
Purdue e-Pubs

Department of Electrical and Computer Engineering Technical Reports

Department of Electrical and Computer Engineering

3-1-1989

SEQUAL 2.1 User's Manual

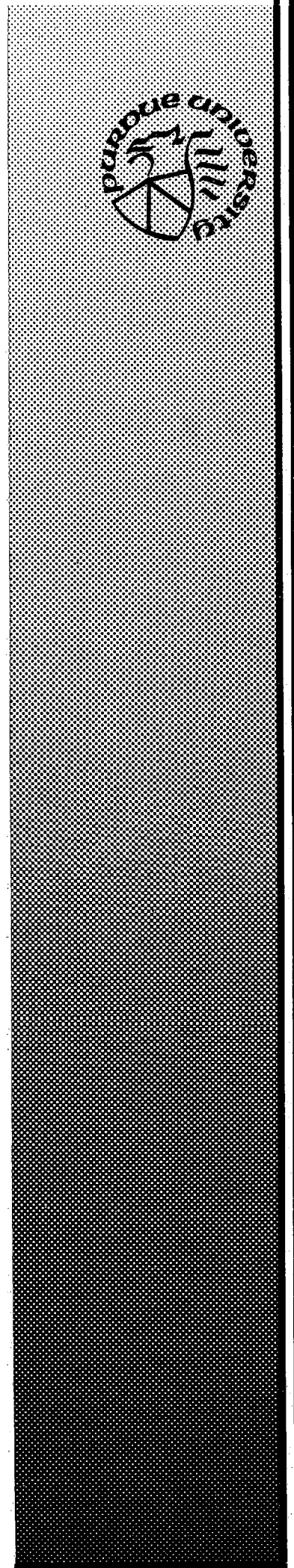
M. J. McLennan
Purdue University

S. Datta
Purdue University

Follow this and additional works at: https://docs.lib.purdue.edu/ecetr

McLennan, M. J. and Datta, S., "SEQUAL 2.1 User's Manual" (1989). *Department of Electrical and Computer Engineering Technical Reports.* Paper 649.
https://docs.lib.purdue.edu/ecetr/649

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.



SEQUAL 2.1
User's Manual

M. J. McLennan
S. Datta

TR-EE 89-17
March, 1989

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

Preface

SEQUAL is a device simulation program which performs a quantum mechanical analysis of electron transport. The user need not be an expert in quantum mechanics to obtain useful results from this program, although an understanding of the subject is an invaluable aid in the interpretation of output. For this reason, the theoretical formulation of the analysis is presented in some detail, following a brief introduction. The knowledgeable (or anxious) user may turn directly to the discussions of input and output in Chapters 3 and 4. The reader is cautioned, however, not to skip Chaper 2 entirely. At the heart of many a program "error" lies a misinterpretation of output.

© Copyright, 1989 by Purdue Research Foundation, West Lafayette, Indiana 47907. All Rights Reserved. Unless permission is granted, this material shall not be copied, reproduced or coded for reproduction by any electrical, mechanical or chemical processes, or combinations thereof, now known or later developed.

Table of Contents

List of Tables

List of Figures

1

Introduction

SEQUAL is a device simulation program, computing Semiconductor Electrostatics by QUantum AnaLysis. Given the characteristics of a particular device, SEQUAL will compute the electron density and the current density, using a quantum mechanical description of electrons. To enhance the versatility of this program, it was designed to be a post-processor for classical simulation programs; as such, it does not calculate any material parameters, so it can be used with any material system. Instead, SEQUAL relies upon the output from classical analysis programs such as FISH1D, PUPHS, or SEDAN, to provide the description of a device. A direct comparison of classical and quantum mechanical results is then possible.

Often (for devices which merit the use of this program) the difference between classical and quantum mechanical results is significant. In such cases, the difference in electron density places the classical description of the electrostatic potential in doubt. The quantum mechanical distribution of space-charge demands a solution for the electrostatic potential which is self-consistent. When a self-consistent solution is desired, SEQUAL allows for iteration between calculations of the electron density and the electrostatic potential. Hence, SEQUAL can provide a correct solution of quantum mechanical electrostatics.

Figure 1.1 depicts the input/output structure of SEQUAL. A numerical description of the device, obtained from the output of a classical analysis program, is the primary source of input. In addition, SEQUAL accepts a list of commands called the "input deck," which allows for selection of program options. A summary of the analysis is written to standard output, in a form well suited to printing. SEQUAL does not produce any plotted output, since undoubtedly the user has a program favored for this purpose. Instead, SEQUAL can generate output files in a variety of formats, to interface with many plotting programs. Quantities available for writing to output files include the input device description, the electron density, the transmission coefficient, and the squared magnitude of the electron wavefunction, output as a surface in both position and energy spaces.

The following chapters are intended not only to describe the operation of SEQUAL, but also to provide some understanding of the calculations performed. In

Chapter 2, the theoretical formulation of the analysis is discussed. Some of the difficulties encountered in a numerical implementation of the theory are noted, and an approach is outlined, which is used by SEQUAL to insure accurate results. Chapter 3 details the input structure of SEQUAL, specifying both the syntax of the input deck and the format of the device description. Program output is discussed in Chapter 4, and an example calculation is presented in Chapter 5. Finally, Chapter 6 examines some practical considerations, for the implementation and use of SEQUAL.

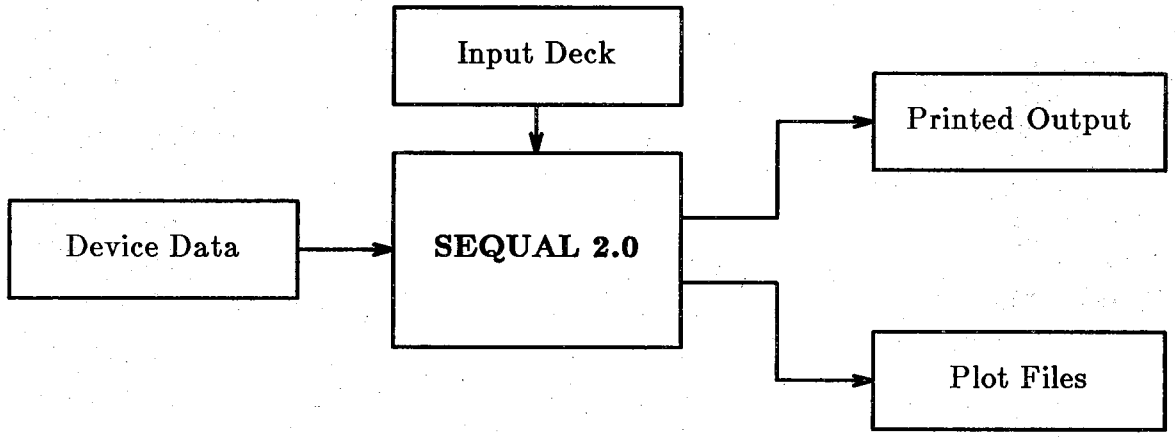


Figure 1.1: Structure of input/output in SEQUAL

2

Theoretical Formulation

The computations performed in SEQUAL are based upon a number of simplifying assumptions:

- Electrons are assumed to be majority carriers; holes are completely ignored in the analysis.
- Profiles of doping density and material composition assumed to vary in only one dimension (hereafter referred to as the "longitudinal direction").
- Effects of the crystal potential are parameterized by an effective mass, which is constant in each material region, and which changes abruptly at a material interface.
- Solutions reflect steady-state conditions, and are independent of time.
- The energy dispersion relation is assumed to be parabolic.
- Electron-electron interactions are neglected.
- Transport is assumed to be ballistic; overall device dimensions are assumed to be sufficiently short, so that scattering can be neglected.

A pictorial representation of the analysis is presented in Figure 2.1(a). Contacts, assumed to be in local thermodynamic equilibrium, inject electrons into a device, with a spectrum of wavevectors $\vec{k}$. These electrons interact with the potential profile E_C; some are transmitted across the structure, and the remainder are reflected back to the injecting contact. Contacts are assumed to supply a continuum of electron energies, and these energies are referred to as "propagating states." If inelastic scattering were included in the model, some of the injected electrons could scatter into states of lower energy. In particular, some electrons could scatter into states which could not be populated by either contact. Such energies, which fall below the conduction-band

edge of either contact, are referred to as "bound states." Solutions of the Schrödinger equation for both propagating states and bound states are described below. Because the primary importance of SEQUAL is its application to transport problems, the solution for propagating states is considered in more detail.

2.1. Propagating Electronic States

2.1.1. Solution of the Schrödinger Equation

Quantum mechanically, an electron is represented by a wavefunction $\psi_{\vec{k}}(\vec{r})$, which is obtained by solving the (time-independent) Schrödinger equation:

$$\frac{\partial}{\partial z}\left(\frac{m_c^*}{m^*(z)}\frac{\partial}{\partial z}\psi_{\vec{k}}(z)\right)+\frac{2m_c^*}{\hbar^2}\left[E_z+E_t\left(1-\frac{m_c^*}{m^*(z)}\right)-E_C(z)\right]\psi_{\vec{k}}(z) = 0 \quad (2.1)$$

where

$$E = \frac{\hbar^2 k_z^2}{2m_c^*}$$

is the energy of the injected electron in the direction of propagation, and

$$E_t = \frac{\hbar^2}{2m_c^*}(k_x^2 + k_y^2)$$

is the energy in transverse directions. In the form presented above, the Schrödinger equation accounts for spatial variations in the effective mass $m^*(z)$, relative to the effective mass of the injecting contact (m_c^*). The conduction-band profile $E_C(z)$ is assumed to be known for a particular device.

An arbitrary potential profile can be described by segmenting a device into a number of tiny intervals, as shown in Figure 2.1(b). Each interval is delineated by points in position space called "nodes." Within each interval, the potential is approximated by its average value. Of course, intervals should be sufficiently small to accurately represent a potential profile. In an region of constant potential, the solution of the Schrödinger equation is some linear combination of plane-waves; hence, the wavefunction for an entire device is deduced by matching boundary conditions between plane-wave solutions in each interval.

In this respect, the solution of the Schrödinger equation is equivalent to a transmission line analysis. Each interval of constant potential acts like a small

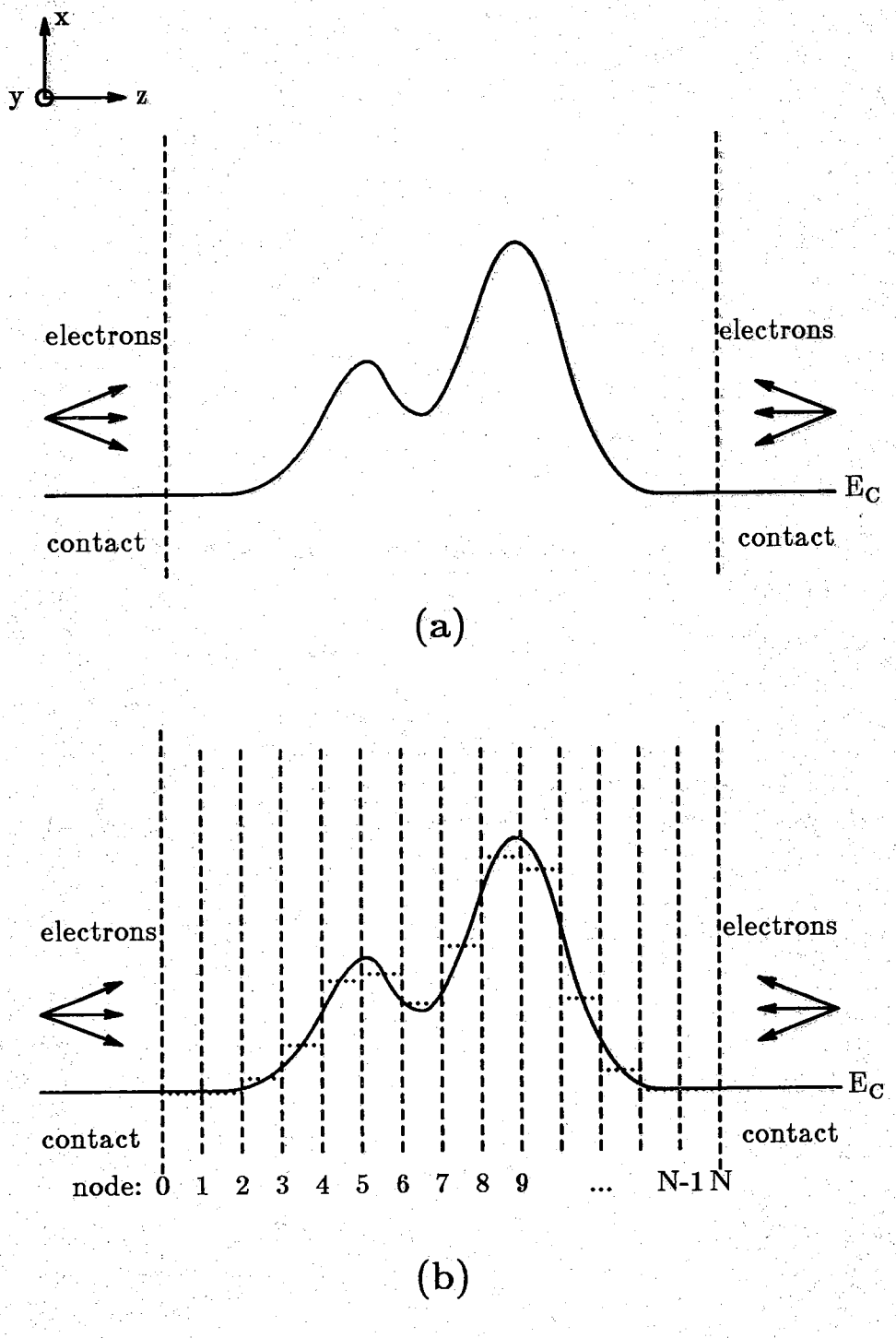


Figure 2.1: (a) Electrons are injected from contacts into a device with an arbitrary potential profile; and (b) any potential profile can be represented by a series of tiny intervals, in which the potential is approximately constant.

section of transmission line with a constant impedance. At the junction between two transmission lines, voltage and current must be continuous. Similarly, proper solutions of the Schrödinger equation require continuity of:

$$\psi_{\vec{k}}(z) \quad \text{and} \quad \frac{1}{m^*(z)}\frac{\partial}{\partial z}\psi_{\vec{k}}(z)$$

at the boundaries between each interval of constant potential. Just as a voltage plane-wave would be reflected by a mismatch in impedance, the electron wavefunction is reflected by changes in the conduction-band profile. When interpreting the results of an analysis, it is useful to keep this analogy at the back of one's mind: Many quantum mechanical effects (*e.g.*, tunneling, resonance states, *etc.*) can be understood in the general context of wave phenomena.

2.1.2. Calculation of Electron Density and Current Density

Because each contact injects electrons into a device, the electron density can be resolved into two components:

$$n(z) = n^{l\to r}(z) + n^{r\to l}(z) \tag{2.2}$$

Electrons injected from the left contact (node 0) are labeled $l\to r$; those injected from the right contact (node N) are labeled $r\to l$. Each component is obtained by integrating the squared-magnitude of the wavefunction over the entire spectrum of wavevectors:

$$n^{l\to r}(z) = \int_0^\infty \frac{dk_z}{2\pi} \left| \psi_{\vec{k}}^{l\to r}(z) \right|^2 \sigma^{l\to r}(k_z) \tag{2.3}$$

$$n^{r\to l}(z) = \int_0^\infty \frac{dk_z}{2\pi} \left| \psi_{\vec{k}}^{r\to l}(z) \right|^2 \sigma^{r\to l}(k_z) \tag{2.4}$$

where the factors $\sigma(k_z)$ represent the integration over transverse wavevectors k_x and k_y:

$$\sigma^{l\to r}(k_z) = \frac{m_c^* k_B T}{\pi\hbar^2} \ln\left[1 + \exp\left[(E_F - E_C(0) - \frac{\hbar^2 k_z^2}{2m_c^*}) / k_B T \right]\right]$$

$$\sigma^{r\to l}(k_z) = \frac{m_c^* k_B T}{\pi\hbar^2} \ln\left[1 + \exp\left[(E_F - E_C(N) - \frac{\hbar^2 k_z^2}{2m_c^*}) / k_B T \right]\right]$$

Here, E_F is the Fermi-level in the injecting contact; m_c^* is the effective mass in the injecting contact; $E_C(0)$ and $E_C(N)$ are the conduction band energies at nodes 0 and N

(left and right contacts, respectively); T is the temperature; and k_z is the incident wavevector, in the direction of propagation.

Notice that, in the Schrödinger equation (2.1), the wavefunction $\psi_{\vec{k}}(\vec{r})$ is dependent on the transverse energy E_t. It was assumed in the derivation of (2.3) and (2.4) that this dependence is weak, so that the wavefunction is approximately constant over all transverse wavevectors. The wavefunction is then evaluated at a single, representative E_t, and removed from the transverse integration. Hence, the integration over transverse wavevectors was performed analytically, and the result is embodied in the factors $\sigma(k_z)$. Normally, the representative transverse energy is assumed to be the thermal average energy, k_BT. Although this assumption is reasonable in many cases, it is strictly an approximation. A rigorous calculation would require an evaluation of the wavefunction at all (longitudinal and transverse) wavevectors. Such a calculation would increase execution time tremendously, to the point where computations are no longer tractable.

Current density for electrons can also be resolved into two components, due to the two, oppositely flowing streams of electrons:

$$J = J^{l\to r} - J^{r\to l} \tag{2.5}$$

where

$$J^{l\to r} = \frac{-q\hbar}{m_c^*} \int_0^\infty \frac{dk_z}{2\pi}\, k_z\, T^{l\to r}(k_z)\, \sigma^{l\to r}(k_z) \tag{2.6}$$

$$J^{r\to l} = \frac{-q\hbar}{m_c^*} \int_0^\infty \frac{dk_z}{2\pi}\, k_z\, T^{r\to l}(k_z)\, \sigma^{r\to l}(k_z) \tag{2.7}$$

The functions $T^{l\to r}(k_z)$ and $T^{r\to l}(k_z)$ are the transmission coefficients from left to right, and from right to left, respectively. Each represents the fraction of injected electrons which propagate across the entire device. Each is proportional to the squared-magnitude of the wavefunction, at the proper end of the device:

$$T^{l\to r}(k_z) \propto |\, \psi_{k_z}^{l\to r}(N)\,|^2 \qquad T^{r\to l}(k_z) \propto |\, \psi_{k_z}^{r\to l}(0)\,|^2$$

In essence, therefore, both the electron density and the current density are determined by integrating the squared-magnitude of the wavefunction.

2.1.3. Numerical Integration Concerns

By using the formulas presented above, calculation of the electron density and current density is straightforward: A grid of wavevectors k_z is chosen for the injection of electrons, and the squared-magnitude of the wavefunction is integrated over that grid. The remaining difficulty lies in determining a suitable grid of wavevectors. Naively, one might assume that a uniform mesh (if sufficiently dense) would provide adequate resolution; this is not the case. Consider, for example, electrons experiencing a transmission resonance. For certain device geometries, it is possible to achieve a large transmission of electrons at particular (resonant) wavevectors. In this case, the wavefunction peaks sharply at resonance, and is nearly zero at all other wavevectors. If resonances are sufficiently sharp, they could be missed entirely by a uniform mesh. Even if a series of successively finer meshes were used, there is no guarantee that the wavefunction would be properly resolved. Aside from being an inefficient solution, therefore, the use of a uniform mesh is a little like playing Russian Roulette.

To insure proper resolution of the wavefunction, the following approach is used in SEQUAL. For each direction of propagation, electrons are injected at uniform intervals in k_z-space. In each interval of k_z-space, the squared-magnitude of the wavefunction is examined for a local maximum. If a maximum is found to exist somewhere in the interval, then the exact wavevector corresponding to the maximum is calculated. The process by which a maximum is detected and located in k_z-space is somewhat complicated, and is unimportant for this discussion. By isolating wavefunction maxima, it is assured that all features of the wavefunction will be resolved clearly.

The integrations required by equations (2.3-2.6) could be performed by a variety of numerical techniques. In SEQUAL, Gauss-Legendre quadrature was chosen, because it offers high accuracy with a comparatively small number of wavefunction evaluations. The "order" of the integration process refers to number of mesh points in an interval; mesh points are chosen by mapping zeroes of Legendre polynomials into a given interval. An estimate of the error obtained for a particular interval is calculated by comparing the results from different orders of integration.

An overview of the entire process of injection and integration is as follows. A uniform grid of wavevectors is used to isolate maxima in the squared-magnitude of the wavefunction. The wavevectors corresponding to maxima define a grid of integration intervals, in which the wavefunction is smooth and well-behaved. Within each integration interval, successive orders of Gauss-Legendre quadrature are applied, until the accuracy of the result is acceptable. In principle, the process of isolating maxima in k_z-space could be performed at each node. Realistically, however, it need only be

performed at those nodes for which the wavefunction varies most rapidly in k_z-space. By default, SEQUAL performs isolation of maxima for the first and last nodes (nodes 0 and N) of a device. Because of "standing-wave" patterns produced by reflection, the wavefunction varies most rapidly at the contact nodes. Other nodes for which rapid variations are expected (*e.g.*, resonance states) can be specified at the time of execution.

2.2. Bound Electronic States

In contrast to the propagating electronic states considered in the preceding sections, bound states cannot be populated by either contact. In a real device, electrons injected from the contacts could inelastically scatter into states of lower energy; however, inelastic scattering has been neglected in SEQUAL, thereby removing the link between propagating states and bound states. In many cases, the density of electrons residing in bound states can be significant. This contribution is particularly important if a self-consistent solution of the electrostatic potential is required. For this reason, SEQUAL allows for the consideration of bound states.

Bound state energies are determined by solving for the eigenvalues of the Schrödinger equation (2.1), written in finite-difference form. It is assumed that the wavefunction vanishes at both ends of the device; hence, the spatial location of bound states should be sufficiently removed from the contacts. The actual solution of the eigensystem is performed by an International Mathematical and Statistical Library (IMSL) routine *eigrf*. Given the eigenfunctions $\psi_i(z)$ and the energy eigenvalues ϵ_i, the electron density for each state is determined by:

$$n_i(z) = \frac{m^*(z) k_B T}{\hbar^2 \pi} \left| \psi_i(z) \right|^2 \ln\left[1 + \exp\left[(E_F - \epsilon_i)/k_B T\right]\right] \tag{2.8}$$

where E_F is the Fermi-level, which is assumed to be constant across the entire device. This result was obtained by assuming that states are populated according to Fermi-Dirac statistics, an assumption which is valid only in equilibrium. For each bound state, a two-dimensional electron density can be calculated by integrating the electron density $n_i(z)$ over the length of the device:

$$n_i^{2-D} = \frac{k_B T}{\hbar^2 \pi} \ln\left[1 + \exp\left[(E_F - \epsilon_i)/k_B T\right]\right] \int_0^{z_N} dz\, m^*(z) \left| \psi_i(z) \right|^2 \tag{2.9}$$

Of course, a finite-difference formulation with nodes 0 to N in position-space will have $N+1$ eigenvalues; of these, only the states which cannot be populated by either contact are considered to be bound states. That is, states are "bound" if the energy

eigenvalue ϵ_i is below the conduction-band edge of either contact. The total (three-dimensional) density of bound-state electrons, therefore, is the sum of contributions from each bound state:

$$n_b(z) = \sum_i n_i(z) \tag{2.10}$$

where the index i is restricted according to:

$$\{\, i \mid \epsilon_i < E_C(0),\ \epsilon_i < E_C(N) \,\}$$

2.3. Iteration for Self-Consistency

In the preceding sections, a method was described for the calculation of electron density, assuming that the conduction-band profile was known. In general, the conduction-band profile is determined by two components: a static component $\Delta E_C(z)$, describing the offset of the band edge in different material layers, and the electrostatic potential $\Phi(z)$:

$$E_C(z) = \Delta E_C(z) - q\Phi(z)$$

Of course, the electrostatic potential can be determined, given the electron density, by solving the Poisson equation:

$$\frac{d}{dz}\left(\kappa_r(z)\,\epsilon_0\,\frac{d}{dz}\Phi(z)\right) = -q\left[N_D^+(z) - n(z)\right] \tag{2.10}$$

where $\kappa_r(z)$ is the (position-dependent) relative dielectric constant, and $N_D^+(z)$ is the density of ionized donors.

For a proper analysis of any device, the solutions for both the electron density and the electrostatic potential should be self-consistent. In SEQUAL, a fully self-consistent solution can be obtained by iterating between the solutions for each quantity; an overview of the iteration process is presented in Figure 2.2. After an initial guess of the electrostatic potential has been input, the electron density $n^0(z)$ is calculated. If iteration is not allowed, the process terminates, yielding a solution of electron density for the given potential profile. If iteration is specified, however, a better guess for the electrostatic potential is determined, by solving the Poisson equation. The corresponding electron density is then calculated, and the process is continued. Iteration is terminated when the calculations converge to a certain number of significant figures, or when the iteration counter i reaches some maximum value.

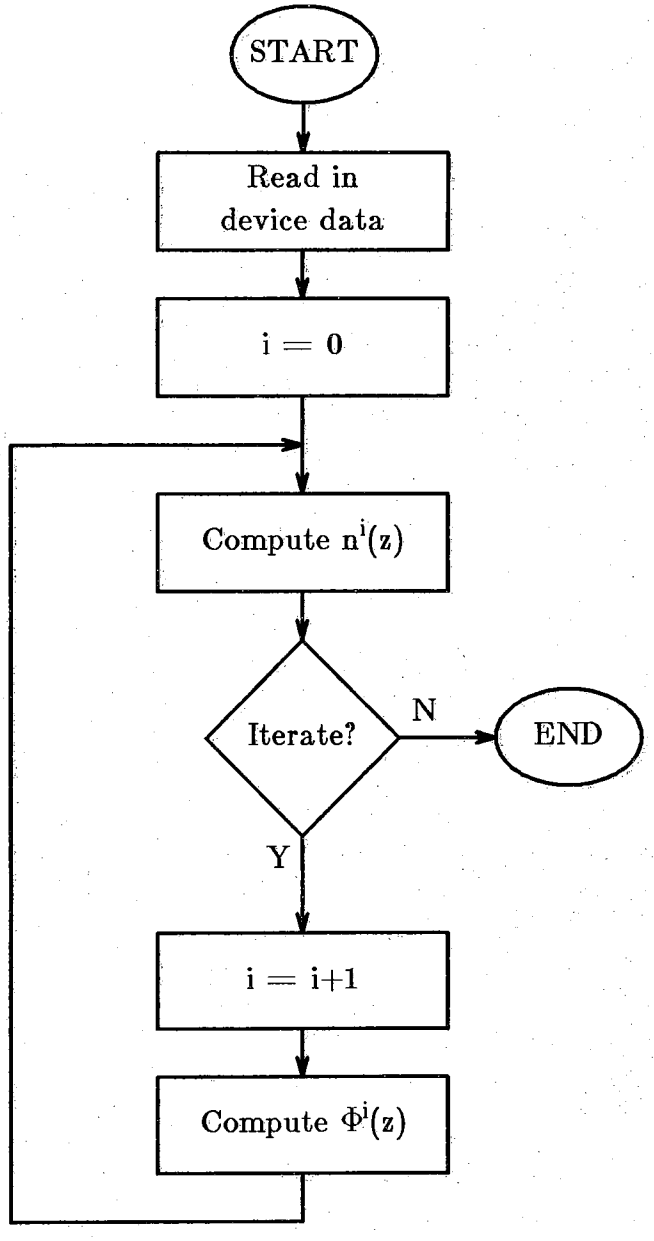


Figure 2.2: Flowchart for a self-consistent solution

3

Program Input

SEQUAL accepts two kinds of input: the device description, and the input deck (see Figure 1.1). Output from a classical analysis program constitutes the device description, a file containing columns of numbers which characterize the device in one dimension. One column is a list of "nodes," or positions in space; all other columns list the value of a quantity (*i.e.*, the conduction band energy, the effective mass, *etc.*) at each node. The second source of input, the input deck, is a list of commands which control the operation of SEQUAL. These commands specify the format of input and output files, modify the default values of input variables, and select a variety of program options. Descriptions of both types of input are presented in detail, below.

3.1. Device Data

The numerical representation of a device is obtained primarily from a file containing several columns of floating-point data; each column represents a different input quantity. Data can be stored in either ASCII or (Fortran 77) binary format. Files containing the necessary information are typically generated as output from a classical analysis program. To provide a flexible interface for many different programs, the quantity associated with each column can be specified in the input deck; therefore, data columns can appear in any order. Moreover, the units of input quantities can be specified, if they differ from the default units in SEQUAL. (For details of the input deck, see section 3.2, below.)

Possible input quantities are listed in Table 3.1. Notice that it is not necessary to have all six quantities specified in the device description file. If it is more convenient, profiles of the donor doping density, the effective mass, and the dielectric constant can be specified in the input deck. Because these quantities are usually constant over large regions (*e.g.*, in a particular material layer), it is doubtful that all classical analysis programs would provide their value at each node. If tabular output of these quantities is unavailable, specifying the values in the input deck is a more convenient approach than modifying output to include the extra columns. Notice also that unless a self-consistent solution is required, profiles of the electrostatic potential

"Device Data" Input Quantities		
Quantity	**Units**	**Necessary for Program Execution**
position-space grid	cm	•
conduction-band profile	eV	•
electrostatic potential	V	‡
ionized donor doping density	cm^{-3}	†
effective mass	(m_0)	†
dielectric constant	(ϵ_0)	†, ‡

† May be specified alternatively in the input deck.
‡ Necessary only for a self-consistent solution.

Table 3.1: Summary of possible input quantities for device data input file.

Input quantities represented by the columns below are in the default order:

position	conduction band profile	electrostatic potential	ionized donor doping density	effective mass	dielectric constant
0.000000e+00	0.000000	0.000000	2.000000e+18	0.0670	12.8464
1.000000e-07	0.000000	0.000000	2.000000e+18	0.0670	12.8464
2.000000e-07	0.000000	0.000000	2.000000e+18	0.0670	12.8464
3.000000e-07	0.000000	0.000000	2.000000e+18	0.0670	12.8464
4.000000e-07	0.000000	0.000000	2.000000e+18	0.0670	12.8464
4.990000e-07	0.000000	0.000000	2.000000e+18	0.0670	12.8464
5.000000e-07	0.365000	0.000000	2.000000e+18	0.08953	11.4705
6.000000e-07	0.365000	0.000000	2.000000e+18	0.08953	11.4705
7.000000e-07	0.365000	0.000000	2.000000e+18	0.08953	11.4705
8.000000e-07	0.365000	0.000000	2.000000e+18	0.08953	11.4705
9.000000e-07	0.365000	0.000000	2.000000e+18	0.08953	11.4705
9.990000e-07	0.365000	0.000000	2.000000e+18	0.08953	11.4705
1.000000e-06	0.000000	0.000000	2.000000e+18	0.0670	12.8464
1.100000e-06	0.000000	0.000000	2.000000e+18	0.0670	12.8464
1.200000e-06	0.000000	0.000000	2.000000e+18	0.0670	12.8464
1.300000e-06	0.000000	0.000000	2.000000e+18	0.0670	12.8464
1.400000e-06	0.000000	0.000000	2.000000e+18	0.0670	12.8464
1.500000e-06	0.000000	0.000000	2.000000e+18	0.0670	12.8464

Figure 3.1: Example listing of a device-description input file in ASCII format

and the dielectric constant are not required. These two quantities are used only in the solution of Poisson's equation, and are therefore unnecessary if iteration is not allowed. Commonly, the device description file will contain only the position-space grid and the conduction-band profile; all other quantities will be specified in the input deck, or will be unnecessary.

From the above discussion, it would seem that the conduction-band profile must always appear in the data description file. This is not exactly the case. The conduction-band profile is composed of two parts:

$$E_C(z) = \Delta E_C(z) - qV(z)$$

where $V(z)$ is the electrostatic potential, and $\Delta E_C(z)$ specifies the offset of the conduction band in different material layers. For a device fabricated with a single material, $\Delta E_C(z)$ is zero (or some arbitrary constant) everywhere. The alignment of energy bands for differing materials, however, produces some offset in the conduction-band edge, described by $\Delta E_C(z)$. An alternative to specifying the conduction-band profile, therefore, is specifying both the electrostatic potential $V(z)$ and the conduction-band offset $\Delta E_C(z)$. In fact, SEQUAL requires a knowledge of all three quantities for a self-consistent solution, so at least two must be specified at the start of execution. In a self-consistent solution, the conduction-band profile is calculated at the beginning of each iteration by adding contributions from the (constant) offset potential and the (updated) electrostatic potential.

An example of a simple device description file in ASCII format appears Figure 3.1. The device which it represents is composed of three material layers--two layers of GaAs sandwiched around a layer of AlGaAs. Each layer is 5.0×10^{-7} cm (50 Å) thick, so the overall device length is 1.5×10^{-6} cm (150 Å). The GaAs layers have an electron effective mass of $m^* = 0.067m_0$, and a dielectric constant of $\epsilon = 12.8464\epsilon_0$. The AlGaAs layer has an electron effective mass of $m^* = 0.08953m_0$, and a dielectric constant of $\epsilon = 11.4705\epsilon_0$. An offset of 0.365 eV in the conduction-band edge of AlGaAs can be seen clearly in the conduction-band profile, since the electrostatic potential was assumed to be zero. Note that the nodes on either side of a material interface are closely spaced, so that the change in the conduction-band edge at the interface is abrupt. Finally, the entire device was assumed to have a ionized donor density of 2×10^{18} cm^{-3}.

3.2. Input Deck

In days of old, when "small" computers fit neatly in a warehouse, and their operators spoke a language no one else understood, the preferred method of input was the punched card. Each line of input was recorded on a card; each file was created by stacking cards in order. Although card-punch machines have become obsolete, their associated terminology has not. The following description of SEQUAL's input "deck" is but one example of jargon refusing to die.

In SEQUAL, the user controls program operation via the input deck, a short file of commands read from standard input. Essentially, the input deck is a list of assignment statements, specifying particular values for SEQUAL's input parameters. Each input parameter is represented by a key word, and logically related "keys" are grouped together on a command line or "card." Figure 3.2 illustrates the input deck syntax. Each card begins with the card name, and is followed by a list of assignments to various keys. A single card can be continued on any number of lines by specifying the continuation character "+" in place of the card name. Note that space is not allowed around the equal sign in an assignment to a key, although anywhere else the use of white space (including spaces, tabs, and commas) is encouraged, to promote legibility. Blank lines and comment lines are also allowed. A comment line is defined as any line beginning with a character which is neither alphabetic nor the "+" continuation character. Notice from Figure 3.2 that a single key can be assigned multiple values (as many as 10 values), by separating each value with a slash; white space is not allowed between values and slashes. A multiple-value assignment can also be continued on any number of lines, as shown in Figure 3.2.

The use of multiply-valued keys will become clear as the details of each input card are explained. Consider, however, the following card, which specifies the electron effective mass in a number of material layers:

```
matter  nodes=5/11/17  emass=0.067/0.08953/0.067
```

This is the input deck specification of effective mass, for the example considered in Figure 3.1. The key *nodes* is assigned the ending nodes for three different material layers. The first layer (GaAs) is defined by nodes 0-5; the middle layer (AlGaAs), by nodes 6-11; and the final layer (GaAs), by nodes 12-17. The key *emass* is assigned the electron effective mass in each of the three material regions. The other use of multiply-valued keys is illustrated by the following card, which specifies the temperature of a device:

```
device  temp=77.0/300.0  bias=0.0
```

Since two temperatures are specified for the key *temp*, two separate calculations will

Typical Input Deck Card:

cardname *key1=value* *key2=value*

Continuation of a Single Card:

cardname	*key1=value*
+	*key2=value*
+	*key3=value*

Assigning Multiple Values to a Single Key:

cardname *key1=value1/value2/value3*

Continuation of Assignment of Multiple Values:

cardname	*key1=value1/value2/value3*
+	*key1=value4/value5*
+	*key1=value6/value7*

Comment Line:

? If the first character on a line is neither alphabetic
? nor a "+" the rest of the line is ignored.

Figure 3.2: Syntax of cards in the input deck

be performed--one at 77 ° K, and the other at 300 ° K. Notice that although two temperatures are specified, only one bias is given. Both calculations, therefore, are performed at the stated bias. To perform one calculation at 77 ° K with zero bias, and another at 300 ° K with a bias of 0.1 V, one would use:

```
device  temp=77.0/300.0  bias=0.0/0.1
```

In general, every key is allowed to have multiple values. Certain keys (such as *temp* and *bias*) direct SEQUAL to perform several different calculations; other keys (such as *nodes* and *emass* considered above) specify information which remains the same for all calculations. Given an input deck, SEQUAL first extracts all of the data from "global information" keys. Remaining keys ("control" keys) are then examined to determine the number of calculations to be performed. The first value of each control key is used in the first calculation; the second value, in the second calculation, and so on. When a list of values for a particular key has been exhausted, the last value of that key is used in any subsequent calculations. The following card,

```
device  temp=77.0/200.0/300.0  bias=0.0
```

is equivalent to

```
device  temp=77.0/200.0/300.0  bias=0.0/0.0/0.0
```

although it demands less typing.

Hence, the use of multiply-value keys can greatly increase the power of the input deck in SEQUAL. Global information keys can be used to specify those device parameters which are not available in tabular form. Control keys can be used to direct SEQUAL to perform a number of similar calculations in a single program execution. The remainder of this chapter is devoted to a detailed explanation of each program key. Further examples of program operation will be presented in Chapter 5.

input — specify format of device description file

Default State:

input file=seq.in format=zevdmk ascii=true

file

The name of the device description file is specified by the *file* key of the *input* card. Since *file* is a control key, several different device descriptions can be processed in a single execution of SEQUAL, by assigning *file* to each of the file names. Note that, in SEQUAL, a file name (or the value assigned to any key, for that matter) is limited to 15 characters.

format

To enhance the flexibility of SEQUAL's interface with the output of other programs, data columns in the device description file can appear in any order. The order is simply specified in the *format* key of the *input* card. The quantity associated with each column of the data description file is represented by a single character, as follows:

Character	Quantity
z	Position-space grid
e	Conduction-band profile
v	Electrostatic potential
d	Ionized donor doping density
m	Electron effective mass
k	Relative dielectric constant
?	(column is ignored)

Notice that it is possible for a device description file to contain extraneous columns, which do not correspond to any input quantities. These columns, referred to by a question mark in the format specification, are read and ignored. For example, suppose a classical analysis program outputs data in tabular form, with the columns: position-space grid, electron density, hole density, ionized donor density, electric field, and conduction-band profile. Of these quantities, only the position-space

grid, the ionized donor density, and the conduction-band profile are desired for input. The format specification for such an input file would be:

```
input   format=z??d?e
```

Note that space is not allowed between the different characters in a format specification.

ascii

A device description file can be stored in either ASCII or (Fortran 77) binary format. The storage type is specified with the *ascii* key of the *input* card:

Value	Implication
true	File is in ASCII format
false	File is in (Fortran 77) binary format

Files in ASCII format contain floating-point data represented by ASCII characters. This is the normal method of output, since data can be listed easily. Files in binary format contain floating-point data represented in the computer's internal format. Acceptable binary-format files must have been produced by a Fortran 77 program, and quantities written must have been single-precision (real), floating-point numbers.† Files in binary format have the advantage of demanding less storage space than equivalent files in ASCII format.

† Of course, if the precision of floating-point variables is somehow doubled by the compiler, then input quantities will be expected to be double precision.

scale

change the units of input quantities in the device description file

---

Default State:

scale cm=1.0 ev=1.0 v=1.0 cm**-3=1.0

If quantities in the device description file are not in the units required by SEQUAL, they can be scaled automatically by specifying an appropriate scale factor. Only values in the device description file are converted. Values in the input deck are assumed to have the appropriate units.

cm

The length scale of the position-space grid is altered by the *cm* key. For example, if the position of each node is in units of microns (1 μm = 10^{-6} m), the appropriate scale factor would be:

```
scale cm=1.0e4
```

ev

The energy scale of the conduction-band profile is altered by the *ev* key. For example, if the conduction-band edge at each node is in units of meV (1 meV = 10^{-3} eV), the appropriate scale factor would be:

```
scale ev=1.0e3
```

v

The scale of the electrostatic potential is altered by the *v* key. For example, if the potential at each node is in units of mV (1 mV = 10^{-3} V), the appropriate scale factor would be:

```
scale v=1.0e3
```

cm-3**

The scale of the donor doping density is altered by the *cm**-3* key. For example, if the doping density at each node is in units of m^{-3}, the appropriate scale factor would be:

```
scale cm**-3=1.0e6
```

matter specify material layers and properties

Default State:

matter

(In the default state, profiles of material parameters are specified in the device description file.)

Normally, material properties are specified in the device description file. When a tabular listing of material properties is difficult to obtain, however, the *matter* card provides an alternative method of input. As many as ten different material layers can be defined, each with differing material properties.

nodes In the data description file, a single line of input (in other words, a single position in space) constitutes a "node." Nodes are referred to by number, starting from zero at the beginning of the file. Material layers, therefore, are defined in the *matter* card by specifying the nodes which are endpoints of material regions. For example, the device description file presented in Figure 3.1 contains 18 lines; the range of node numbers, then, is 0 to 17. To define a single material region with an effective mass of 0.067, the *matter* card would read:

```
matter  nodes=17  emass=0.067
```

To define three material layers (as in the example of Figure 3.1) the matter card would read:

```
matter  nodes=5/11/17  emass=0.067/0.08953/0.067
```

Note an equal number of values must be assigned to *nodes*, *delec*, *emass*, and *krel*.

delec The conduction-band offset ΔE_C can be specified for each material layer defined by *nodes*. The offset is understood as the conduction-band discontinuity at a material interface, and is specified in electron-volts. Note that, for an abrupt interface, the nodes on either side of the interface should be closely spaced.

emass The electron effective mass can also be specified for each material layer defined by *nodes*. The value is input as a dimensionless quantity, in terms of the free electron mass, m_0.

krel The relative dielectric constant can also be specified for each material layer defined by *nodes*. The value is input as a dimensionless quantity, in terms of the dielectric constant of free space, ϵ_0.

doping specify the ionized donor doping density

Default State:

```
doping   ni=1.79e6
```

(In the default state, the doping-density profile is specified in the data description file.)

nodes Just as the *nodes* key of the *matter* card defines different material regions, the *nodes* key of the *doping* card defines regions with a constant density of ionized donors. Unless a self-consistent solution is desired, the donor density is needed only at the contacts (first and last nodes), to determine the Fermi-level of injected electrons. In this case, the specification could be as simple as:

```
doping   nodes=0/19/20   nd+=5.0e17/0/5.0e17
```

for a device with nodes 0 to 20, with an ionized donor density of 5.0×10^{17} cm^{-3} in each contact. Because the donor density at interior nodes is irrelevant (unless a self-consistent solution is required), the example above could be equally expressed as:

```
doping   nodes=20   nd+=5.0e17
```

For specifying the contact doping, or for some crude approximations to real doping profiles, the *doping* card provides a convenient method of input.

nd+ The density of ionized donors (in units of cm^{-3}) is assigned to *nd+* for each region defined by the *nodes* key. Notice that SEQUAL requires the density of *ionized* donors. Partial ionization of donors, therefore, must be calculated by the classical analysis program contributing input.

ni If the ionized donor density in either device contact (*i.e.*, the first and last nodes) is identically zero, then the intrinsic electron density assigned to *ni* is assumed, for the calculation of the Fermi-level. SEQUAL assumes that charge neutrality exists in the contacts,

between electrons and ionized donors; therefore, the ionized donor density in the contacts, unless it is identically zero, is used to determine the Fermi-level for injection of electrons. Notice that if an extremely small (but non-zero) donor density is specified, a Fermi-level will be deduced to correspond to this concentration. This provides a trick for specifying the electron concentration in the case of a *p*-*n* junction. Holes are completely ignored in SEQUAL, and hence, there is no way of specifying a doping density for acceptors. In order to obtain the proper electron concentration, the "ionized donor density" on the *p*-side of the junction should be set equal to the minority carrier (electron) concentration. If, instead, it were set to zero, the value assigned to *ni* would be assumed for the minority carrier (electron) concentration.

device specify macroscopic device parameters

Default State:

device temp=300.0 area=1.0 bias=0.0

temp The device temperature (in °K) for a calculation is assigned to the *temp* key.

area The cross-sectional area (in cm^2) of a device is assigned to the *area* key. It is used as a multiplicative constant in the calculation of total current. The default area is 1 cm^2, so that the total current (in A) is the same as the current density (in A/cm^2).

bias An applied bias can be added to the device, in addition to any existing bias in the output from a classical analysis program. The amount of bias (in V), assigned to the *bias* key, is applied as a linear potential drop across the lightly-doped (middle) region of the device. For a self-consistent solution, several iterations should be allowed after the application of bias, so that the electrostatic potential will return to the self-consistent state. If the internal nodes of the device are no more lightly doped than the contacts, the bias is applied as a linear drop across the entire device. In this case, obtaining convergence for self-consistent calculations might be difficult. Note that it is possible to have a device description file in which bias is built into the electrostatic potential and the conduction-band profile. In some cases, it may be more convenient (or accurate) to apply bias in the classical analysis program, and use SEQUAL to analyze the resulting potential profile.

solve — specify parameters controlling the analysis

Default State:

solve prec=3 itmax=9 states=prop inject=both

prec The number of significant figures desired for important quantities can be assigned to the *prec* key of the *solve* card. "Important" quantities include the electron density, the current density, and the Fermi-level in each contact. Since SEQUAL will struggle admirably (at the expense of CPU time) to achieve whatever precision is specified, the value should be kept within reasonable limits.

itmax For a self-consistent analysis, SEQUAL solves iteratively for the electrostatic potential and the electron density (see Figure 2.2). Convergence is achieved when the number of significant figures in the current density (between iterations) settles down to the number requested. To avoid excessive use of CPU time in obtaining convergence, SEQUAL terminates iteration when the number of iterations exceeds the value assigned to *itmax*. Therefore, iteration can be suppressed by specifying zero as the maximum number of iterations. If results are written to output files (see description of the *output* card), iteration can be continued at any point, by using the output of a previous run as the input device description file (see Figure 4.2).

states Both propagating and bound electronic states can be considered in the solution of the Schrödinger equation, according to the value assigned to *states*:

Value	Implication
prop	Consider propagating electronic states
bound	Consider bound electronic states
all	Consider propagating and bound states

The default is to consider only the propagating states, since bound states do not contribute to current density. Furthermore, the

population of bound states is correct only for devices in equilibrium (see section 2.2). For small deviations from equilibrium, however, the bound-state result obtained is a reasonable approximation to the correct solution. The consideration of bound states is particularly important for a self-consistent solution. Because the electrostatic potential is determined from the electron density, the bound-state contribution (even if it is only approximate) can significantly alter the final result.

inject

In the solution of Schrödinger's equation for propagating states, electrons can be injected into the device from two contacts (see section 2.1.1). Each contact provides a separate contribution to both electron density and current density. The value assigned to *inject* determines which of the contributions will be calculated:

Value	Implication
l-to-r	Consider electrons propagating from left to right (inject from left contact)
r-to-l	Consider electrons propagating from right to left (inject from right contact)
both	Inject from both contacts

Because the vast majority of devices require an analysis with injection from both contacts, the default value is "both." In special cases, however, the contribution from one contact may be insignificant. Consider, for example, the calculation of electron current for a *p*-*n* junction. Injection from the *p*-type contact is unnecessary, since the current component would be negligible.

maxima specify parameters for the isolation of wavefunction maxima

Default State:

maxima prec=2 kscale=1.0

(In the default state, the first and last nodes are "watched" for wavefunction maxima; interior nodes are ignored.)

prec In the process of stepping through k_z-space, SEQUAL looks for maxima in the squared-magnitude of the wavefunction (see section 2.1.3). This insures both the proper integration of the wavefunction, and the proper resolution of the transmission coefficient. The number of significant figures requested for the calculations related to maxima isolation is assigned to the *prec* key of the *maxima* card. Normally, the default value demands sufficient precision. Particularly sharp transmission resonances, however, may require greater precision to be properly resolved. When a doubt arises, results from two different requested precisions should be compared.

kscale SEQUAL steps through wavevector-space for the purpose of isolating maxima in the electron wavefunction (see section 2.1.3). Intervals of k_z-space between successive maxima are then integrated, to determine the electron density and the current density. Normally, the default k_z-step is adequate for isolation of wavefunction maxima. The k_z-step can be scaled to any size, however, according to the value assigned to *kscale*. Notice that the size of the k_z-step (if it is sufficiently small) has no effect on integration of the wavefunction. Integration intervals are defined by local maxima in the wavefunction. A smaller k_z-step can provide better resolution in the output of quantities, such as the transmission coefficient and the wavefunction magnitude. A larger k_z-step should be used with extreme caution: If wavefunction maxima are improperly resolved, the calculations for electron density and current density will be in error.

watch The isolation of k_z-space maxima in the wavefunction could be accomplished at all position-space nodes. This would increase execution time tremendously, however, without adding to the accuracy of the calculation. It is necessary to isolate k_z-space maxima only at those nodes for which the wavefunction varies rapidly. By default, the first and last nodes of any device are "watched" for maxima in k_z-space (see section 2.1.3). Additional nodes can be considered, if the node numbers are assigned to the *watch* key.

integ — specify parameters for the integration of the wavefunction

Default State:

integ orders=2/4/6/8/10/12/16/20/24 kbt=10.0 et=1.0

orders For the calculation of electron density and current density, the magnitude of the wavefunction must be integrated in k_z-space (see section 2.1.2). Integration is performed using Gaussian quadrature with Legendre polynomials (see section 2.1.3). To achieve the precision requested (via *prec* of the *solve* card), successive orders of integration are applied to an integration interval, until the precision is obtained, or until the list of integration orders is exhausted. Orders available for the integration process are assigned to the *orders* key. In the default state, all available orders are assigned to *orders*. For cases in which the reduction of execution time is more important than the accuracy of results, the list of integration orders may be abbreviated. Indeed, a single integration order could be specified (*e.g.*, the highest order available), if an estimation of the solution accuracy is unnecessary.

kbt In theory, integration of the wavefunction should be performed over the range of all wavevectors from zero to infinity (see section 2.1.2); in practice, integration must be truncated at some large but finite wavevector. The point of truncation is determined in SEQUAL by considering the number of significant figures in integration results. Integration is performed up to some minimum wavevector. Beyond this, integration is continued (if necessary) to achieve the requested precision, for both the electron density and the current density. The minimum wavevector for integration corresponds to an energy which is some multiple of k_BT higher than the maximum energy in the conduction band. The value assigned to *kbt* specifies the multiple of k_BT.

et In the derivation of formulas for both the electron density and the current density, the wavefunction was assumed to be weakly dependent on the transverse energy E_t (see section 2.1.2). This allowed the wavefunction, evaluated at a representative transverse energy, to be

removed from the integration over transverse momenta. Hence, the integration over transverse momentum could be performed analytically. The particular transverse energy at which the wavefunction is evaluated is determined by the value assigned to *et* (in units of k_BT). A reasonable assumption for the representative transverse energy is the thermal average energy, k_BT.

title

specify a title for printed (standard) output

Default State:

title

(In the default state, no title appears on output pages.)

The *title* card is unique for two reasons: It is the only card in the input deck which does have any keys, and which cannot be continued on multiple lines. Any text following the card name is taken as the title of the execution. This title appears in the heading of each page of printed output.

print — specify the form of printed (standard) output

Default State:

print tcoeff=* format1=zevn format2=zdmk verbose=true

tcoeff For propagating electrons, a listing of the transmission coefficient versus incident energy can be obtained in printed output, according to the value assigned to *tcoeff*:

Value	Implication
l-to-r	Print the transmission coefficient for electrons propagating left-to-right
r-to-l	Print the transmission coefficient for electrons propagating right-to-left
both	Print transmission coefficients for both directions of propagation
*	Do not print transmission coefficients

For self-consistent solutions, a listing of the transmission coefficients is provided only for the final iteration.

format1

format2 A tabular listing of both input parameters and calculated results can be obtained in printed output. Two sections of output, each with a maximum of four columns, can be defined with the output format keys, *format1* and *format2*. The quantity associated with a particular column is represented by a single letter, as follows:

Character	Quantity
z	Position-space grid
e	Conduction-band profile
v	Electrostatic potential
n	Electron density
d	Ionized donor doping density
m	Electron effective mass
k	Relative dielectric constant
*	(No output in a section)

The definitions shown above, except for the addition of *n* and *, are identical to those presented in conjunction with input format. Unlike the input format specifier, *format1* or *format2* (or both) can be assigned to *, which causes an output section to be suppressed. For example, to obtain a single output section listing position, conduction-band energy, and electron density, the *print* card could be either of the following:

```
print    format1=zen  format2=*

print    format1=*  format2=zen
```

verbose

Because SEQUAL can provide so many sections of output in a single run, the user is allowed the option of suppressing superfluous output, with an assignment to *verbose*:

Value	Implication
true	All output sections are supplied
false	Superfluous output sections are suppressed

Although some of the "verbose" output may seem unnecessary, it is included to aid the user in understanding the problem description, and in evaluating the performance of SEQUAL.

output specify the form of (plotting) output files

Default State:

output file=seq data=* columns=3 ascii=true

For each calculation, data presented in printed output are also available for storage in files. Such files can be used as input to plotting programs, or as input (for further processing) to SEQUAL.

file The argument of the *file* key on the *output* card is used as a basis in forming all output file names. For each file created, an extension is added to the root file name specified, to identify both the type and format of stored data. (For an understanding of different output files and their respective extensions, see section 4.2.)

data Different types of data can be requested for output, according to a string of single-character keys assigned to *data*:

Character	Type of Data Stored
d	Device data (quantities *vs.* position)
b	Bound-state data
t	Transmission coefficient data
w	Surface plot of the wavefunction magnitude
*	(No data is stored)

In an assignment to *data*, the characters above may appear in any order. For example, the following two cards are equivalent:

```
output    data=dtw

output    data=wdt
```

Notice that space is not allowed between the characters.

columns For compatibility with many different plotting programs, data can be written to output files in single-column, paired-column, or multiple-column formats, according to the value assigned to *columns*:

Value	Implication
1	Single-column format
2	Paired-column format
≥ 3	Multiple-column format

ascii Output files can be stored in either ASCII format or (Fortran 77) binary format, according to the value assigned to *ascii* on the *output* card:

Value	Implication
true	File is in ASCII format
false	File is in (Fortran 77) binary format

Summary of Input/Output Keys			
Card	**Key**	**Value**	**Key Type †**
input	file	text (*filename*)	control
	format	(z, e, v, d, m, k, ?)	control
	ascii	logical	control
scale	cm	real	control
	ev	real	control
	v	real	control
	cm**-3	real	control
matter	nodes	integer	global information
	delec	real (eV)	global information
	emass	real (m_0)	global information
	krel	real (ϵ_0)	global information
doping	nodes	integer	global information
	nd+	real (cm^{-3})	global information
	ni	real (cm^{-3})	control
title	--	--	--
print	tcoeff	(l-to-r, r-to-l, both, *)	control
	format1	(z, e, v, d, m, k, n, *)	control
	format2	(z, e, v, d, m, k, n, *)	control
	verbose	logical	control
output	file	text (*filename*)	control
	data	(d, b, t, w, *)	control
	columns	integer	control
	ascii	logical	control

† For a explanation of "control" keys and "global information" keys, see section 3.2.

Table 3.2: Summary of keys which control the input/output of SEQUAL

Summary of Execution Keys			
Card	**Key**	**Value**	**Key Type †**
device	temp	real (°K)	control
	area	real (cm^2)	control
	bias	real (V)	control
solve	prec	integer	control
	itmax	integer	control
	states	(prop, bound, all)	control
	inject	(l-to-r, r-to-l, both)	control
maxima	prec	integer	control
	kscale	real	control
	watch	integer	global information
integ	orders	integer	global information
	kbt	real (K_BT)	control
	et	real (K_BT)	control

† For a explanation of "control" keys and "global information" keys, see section 3.2.

Table 3.3: Summary of keys which control the execution of SEQUAL

4

Program Output

SEQUAL provides two kinds of output: printed output, and output files. For each execution, a record of input data and output results is written to standard output. Because this record is formatted with Fortran line-printer codes (providing page ejection, *etc.*) it is intended to serve as a printed report of all calculations. Pages are numbered, and headings are printed at the top of each page. Output files, on the other hand, are not well suited to examination; rather, they are provided to serve as input for plotting programs. Because both forms of output can include a wide variety of different results, each form is presented in detail, in the following sections.

4.1. Printed Output

Figure 4.1 depicts the general form of printed output in SEQUAL. For each execution of the program, a number of different output sections could appear in the printed record. Each section will only appear, however, if it is necessary. Moreover, some of the output is verbose, and can be suppressed at the request of the user. In Figure 4.1, the sections grouped together in a dashed box present output for each calculation. Since SEQUAL can perform several calculations in a single program execution, these sections may appear several different times in the printed record. (See discussion of "control" keys in section 3.2.) Although the example output pages presented in Chapter 5 are easily worth a thousand words, a brief description of each section is given below:

Input Deck:
Printed output begins with an echo of the input deck. Syntax errors (if any are found) are pointed out under each offending line.

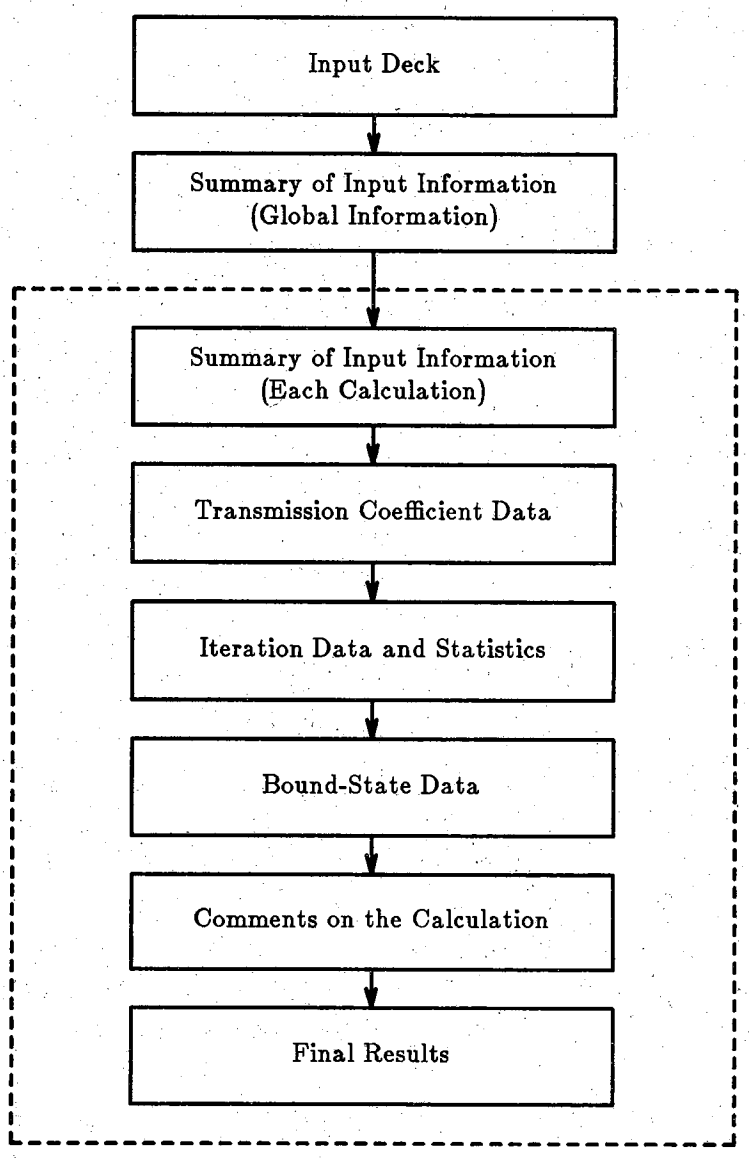


Figure 4.1: Diagram of printed output. Items in the dashed box are presented for each calculation (see discussion of "control" keys in section 3.2).

Summary of Input Information (Global Information):
Device data which has been specified on input cards *matter* or *doping* is summarized in a pictorial fashion. Because this information is global to all calculations, it appears only once, near the start of printed output.

Summary of Input Information (Each Calculation):
For each calculation, SEQUAL presents a page summarizing all important input parameters. Although this page is helpful in clarifying the details of a particular calculation, it is considered "verbose," and it can be suppressed.

Transmission Coefficient Data:
A table of the transmission coefficient versus energy can be requested for each direction of propagation. For iterative calculations, a table is generated only for the results of the final iteration.

Iteration Data and Statistics:
For iterative calculations, the degree of convergence obtained for each iteration is summarized in tabular form. In addition, statistics characterizing the propagating-state solution are presented. Because most of SEQUAL's execution time is spent in the analysis of propagating states, these statistics can be used to evaluate the overall performance of the program. Although propagating-state statistics are sometimes useful, they are considered "verbose," and can therefore be suppressed.

Bound-State Data:
For bound-state calculations, SEQUAL presents a table of bound-state energies and two-dimensional electron densities. If a propagating-state solution was also requested, a breakdown of the electron density is printed, showing the contributions due to propagating states and bound states.

Comments on the Calculation:
All warnings, cautions, and error messages encountered in a particular calculation are collected onto a single page. A brief explanation is presented for each problem encountered, along with suggestions for relieving the difficulty.

Final Results:
A section of final results begins by listing the current obtained for the bias across the device. Following this, device data is presented in two sections, as specified by *format1* and *format2* on the *print* card.

4.2. Output Files

In addition to the printed record, SEQUAL can create a number of data files for the storage of results. Because these files contain raw data, they are well suited for use with plotting programs. Of course, different plotting programs require different formats for input data. Some programs demand x-axis data and y-axis data in separate files; some require a paired listing of x and y values. To accommodate the majority of programs, SEQUAL can output data in three formats, according to the value assigned to *columns* on the *output* card. Data can be written in single-column, paired-column, or multiple-column formats. Obviously, many data files will be created by the single-column or paired-column option; files are distinguished by an extension added to the specified file name. Extensions were chosen to indicate at a glance the contents of a particular data file. For instance, the files ending in ".z" contain a single column, listing the position-space grid; files ending in ".zv" contain two columns of data--the position-space grid and the electrostatic potential. A complete listing of extensions is presented in Tables 4.1-4.4.

Table 4.4 shows that, for output files describing the wavefunction surface, the format is independent of the number of columns specified. Files ending in ".wlr-z" or ".wrl-z" contain a list of position-space nodes; similarly, files ending in ".wlr-e" or ".wrl-e" contain a list of energy-space nodes. The squared-magnitude of the wavefunction, weighted by $\sigma/2\pi$, is listed in files ending with ".wlr-m" or ".wrl-m" extensions. Data appear sequentially: For each energy, the quantity is listed across the entire device, from the first node to the last. Plotting programs reading the data, therefore, will read a matrix of values. In the process of reading data, the index for position-space entries will vary more rapidly.

Output files in multiple-column format can also be used as input to SEQUAL. Consider, for instance, the computation of a self-consistent current-voltage characteristic. Using a multiple-valued assignment to the *bias* key, the user can instruct SEQUAL to calculate current at a number of different biases. For a self-consistent calculation, a bias is applied to an input potential, and solutions for the electron density and electrostatic potential are performed iteratively. One would expect the self-consistent potential obtained for one bias to be quite close to the solution for the

next bias. Rather than apply biases to the same input file, the clever user will feed output results back in, as input.

In Figure 4.2, an example input deck is shown, to perform the feedback. The file *classical* contains output from a classical analysis program, with data in the default column-format *zevdmk*. A self-consistent calculation is performed at zero bias, and the results are stored in the file *out1.z...k*. This file is then used as the device description for the next calculation. Because the output from SEQUAL includes three columns describing electron density (propagating, bound, and total), the input format becomes *zev???dmk* for the three calculations with feedback. For each calculation, an additional bias of 0.1 V is applied to the potential profile. Hence, current is calculated for biases of 0 V, 0.1 V, 0.2 V, and 0.3 V.

```
              **** I/O Feedback:
              *      Results of one calculation are stored in output
              *      files, and read back as input.
              ****

input         file=classical/out0.z...k/out1.z...k/out2.z...k
  +           format=zevdmk/zev???dmk

device        bias=0.00/0.10/0.10/0.10
solve         itmax=15

output        file=out0/out1/out2/out3  data=d  columns=3
```

Figure 4.2: Input deck for feedback of output files

Summary of Output Files (*data*=d)		
columns=	File Extension	Quantity
1	.z	position
	.e	conduction-band energy
	.v	electrostatic potential
	.p	propagating electron density
	.b	bound-state electron density
	.n	total electron density
	.d	ionized donor density
	.m	effective mass
	.k	relative dielectric constant
2	.ze	position conduction-band energy
	.zv	position electrostatic potential
	.zp	position propagating electron density
	.zb	position bound-state electron density
	.zn	position total electron density
	.zd	position ionized donor density
	.zm	position effective mass
	.zk	position relative dielectric constant
≥ 3	.z...k	position conduction-band energy electrostatic potential propagating electron density bound-state electron density total electron density ionized donor density effective mass relative dielectric constant

Table 4.1: Summary of output files created for storage of device data

Summary of Output Files (*data*=t)		
columns=	File Extension	Quantity
1	.tlr-e	energy (propagation left-to-right)
	.tlr-c	transmission coefficient (propagation left-to-right)
	.trl-e	energy (propagation right-to-left)
	.trl-c	transmission coefficient (propagation right-to-left)
2, ≥3	.tlr-ec	energy transmission coefficient (propagation left-to-right)
	.trl-ec	energy transmission coefficient (propagation right-to-left)

Table 4.2: Summary of output files created for storage of transmission coefficient data

Summary of Output Files (*data*=b)		
columns=	File Extension	Quantity
1	.bs-e	bound-state energy
	.bs-n	two-dimension electron density
2, ≥3	.bs-en	bound-state energy two-dimensional electron density

Table 4.3: Summary of output files created for storage of bound-state data

Summary of Output Files (*data*=w)		
columns=	**File Extension**	**Quantity**
1, 2, $\geq$3	.wlr-z	position (propagation left-to-right)
	.wlr-e	energy (propagation left-to-right)
	.wlr-m	$\lvert\psi^{l\rightarrow r}\rvert^2\ \sigma^{l\rightarrow r}\ /\ 2\pi$
	.wrl-z	position (propagation right-to-left)
	.wrl-e	energy (propagation right-to-left)
	.wrl-m	$\lvert\psi^{r\rightarrow l}\rvert^2\ \sigma^{r\rightarrow l}\ /\ 2\pi$

Table 4.4: Summary of output files created for storage of wavefunction surface plots

5

Example Calculations

To illustrate some of the abilities of SEQUAL, example calculations are presented in this chapter, for a resonant tunneling device fabricated by Ray, *et. al.* † The device, pictured in Figure 5.1, is composed of two $Al_{0.45}Ga_{0.55}As$ barriers sandwiched around a GaAs potential well.† Classically, electrons injected from the contacts at energies below the top of the barrier are completely reflected; quantum mechanically, however, it is possible for electrons to be transmitted. Electrons, after tunneling through one potential barrier, can experience multiple reflections inside the quantum well, before tunneling through the other barrier. Because of the wave-nature of electrons, these multiple reflections can constructively interfere, producing a large transmission across the entire device. In summary, electrons injected at particular "resonant" energies will experience unity transmission; those injected at energies off-resonance will be strongly reflected. Resonant energies are often referred to as "quasi-bound states," since in the process of multiple reflection, electrons are effectively bound to the well.

SEQUAL can be used to graphically illustrate the resonance condition described above. Figure 5.2 presents a surface plot of the wavefunction versus position and incident electron energy. The quantity $\log_{10}(|\psi^{r\to l}(z)|^2 \, \sigma^{r\to l}(k_z) \, / \, 2\pi)$ plotted can be interpreted loosely as the probability of finding an electron at a particular position, or at a particular energy. It is plotted on a logarithmic scale, so that important features can be seen clearly. Electrons, injected from the right-hand contact, are propagating from right to left. At low energies, electrons are strongly reflected, and the wavefunction exhibits a pattern of standing waves, near the right-hand contact. At the resonant energy, however, the wavefunction peaks sharply within the GaAs-well, and a ridge of transmitted electrons can be seen extending to the left contact. Note that the wavefunction peak is localized to the GaAs well in position-space, and it is extremely narrow in energy-space.

---

† S. Ray, P. Ruden, V. Sokolov, R. Kolbas, T. Boonstra, and J. Williams, "Resonant Tunneling Transport at 300 K in GaAs-AlGaAs Quantum Wells Grown by Metalorganic Chemical Vapor Deposition," *Applied Physics Letters*, 48(24), pp. 1666-1668, 1986.

The wavefunction surface plot presented in Figure 5.2 was generated from the output of SEQUAL. A position-space grid, for the structure shown in Figure 5.1, was created and stored in the file *rtd*. Positions were written in angstrom units (1 Å $= 10^{-8}$ cm), and the electrostatic potential was taken as zero everywhere. The input deck, along with the rest of printed output, is shown in Figure 5.3. To insure proper resolution of the wavefunction, an additional node in the middle of the quantum well was specified for "watching." Injection of electrons was specified as right-to-left. Data files used for the wavefunction surface plot were obtained by specifying "*data*=w" on the *output* card. Notice that the electron density, shown in the section of final results, is smaller at the left-hand contact (near z=0). This is expected, because the electron density listed is not the total electron density; rather, it is the component due to electrons injected only from the right-hand contact.

SEQUAL can also be used to determine the current-voltage characteristic for this device, by assigning a number of voltages to the *bias* key. In Figure 5.4, an input deck is shown, which has been modified for this purpose. Notice that separate output-file names are specified for each bias, so that files are not overwritten. For the purposes of this example, a self-consistent solution was not necessary; hence, the maximum number of iterations was set to zero. As an initial guess, biases were specified in increments of 0.05 V; the input deck was then modified again, to include a few additional biases. Values of current, taken from the listings in the sections of final results, were stored in a separate file, and plotted. The resulting current-voltage characteristic appears in Figure 5.5.

Negative differential resistance, apparent in Figure 5.5, is a characteristic feature of resonant tunneling devices. For small biases, the dominant current component is supplied by electrons tunneling through the first quasi-bound state, from the right-hand contact. As larger biases are applied across the device, the quasi-bound state is lowered in energy, with respect to the conduction-band edge in the right-hand contact. More electrons are available for tunneling at energies near the conduction-band edge in the contact. Therefore, current increases to a maximum value. When the quasi-bound state is pulled below the range of injection energies of the contact, however, current is abruptly reduced. This effect is depicted graphically in Figure 5.6, which shows the conduction-band profile of the resonant tunneling device, at the bias of maximum current. Notice that the quasi-bound state energy is quite close to the conduction-band edge in the right-hand contact. Any additional bias lowers the quasi-bound state below the range of (right-hand) injection energies, and current is abruptly cut-off.

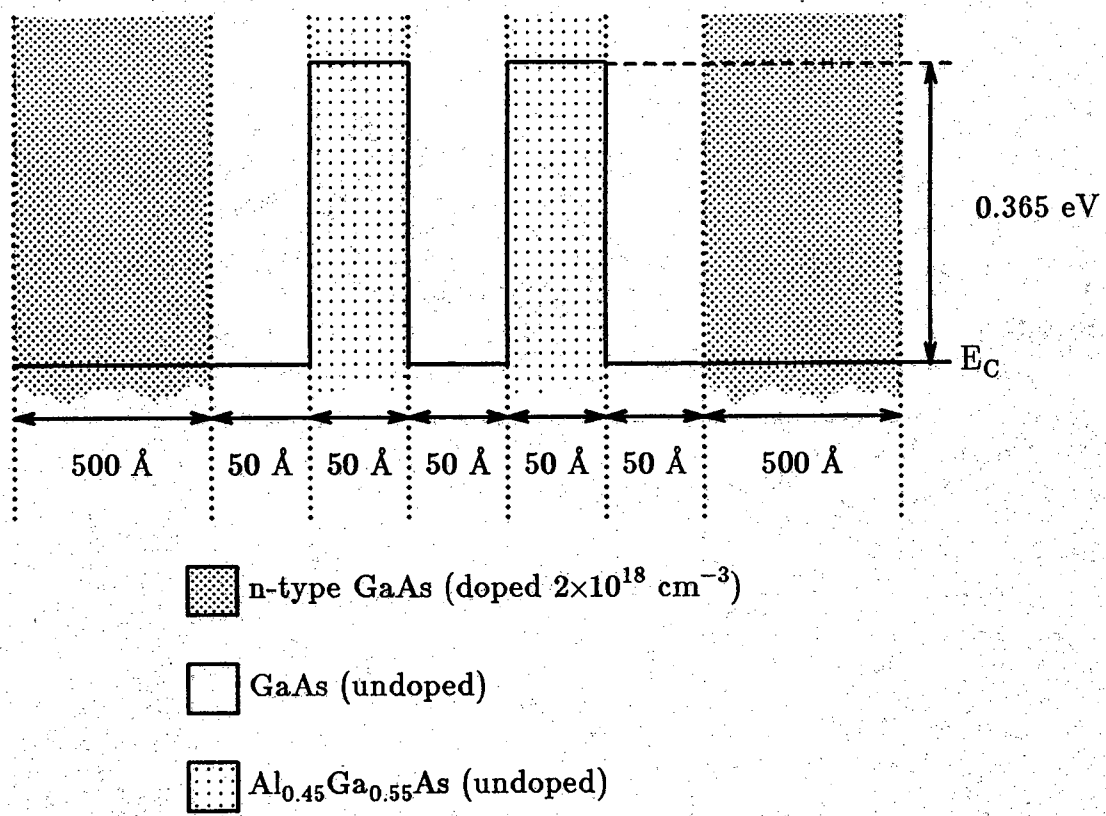


Figure 5.1: Structure of the resonant tunneling device examined in subsequent calculations

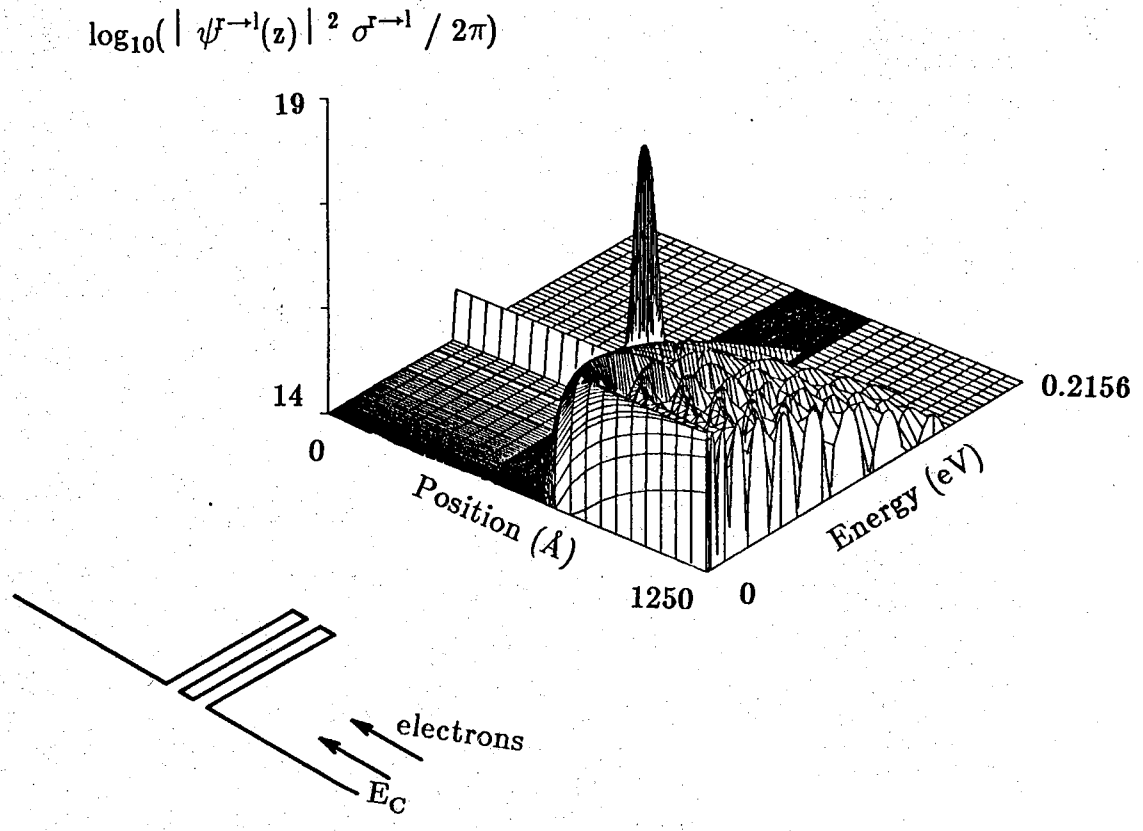


Figure 5.2: The wavefunction magnitude, weighted by the transverse integration $\sigma^{r\to l}(k_z)$, for electrons injected from the right contact of the resonant tunneling device in Figure 5.1

```
------- XXXX  XXXXX   XXXX   XX  XX   XXXX   XX -----------------------------
      XX     XX     XX  XX  XX  XX  XX  XX  XX                      SEQUAL 2.0
----- XXX   XXXX   XX  XX  XX  XX  XXXXXX  XX ------------- Purdue University
       XX  XX     XX  X   XX  XX  XX  XX  XX                       August 1987
-- XXXX   XXXXX   XXX X   XXXXX  XX  XX  XXXXXX -----------------------------


SEQUAL:  input deck
-------------------------------------------------------------------------------

        ************************************************************
        **  a resonant tunneling device fabricated by ray et. al.  **
        **    (applied physics letters, 48(24), p. 1666, 1986) is  **
        **    simulated.  a plot of the wavefunction surface in    **
        **    position-space and energy space is obtained, for     **
        **    electron injection from the right contact.           **
        ************************************************************

title   resonant tunneling device (appl. phys. lett., 48(24), p. 1666, 1986)


        -- input file "rtd" contains positions in angstroms --

input   file=rtd format=zv
scale   cm=1.0e8

matter  nodes=21/32/43/54/76
      + delec=0.0/0.365/0.0/0.365/0.0
      + emass=0.067/0.08953/0.067/0.08953/0.067

doping  nodes=10/65/76  nd+=2.e18/0./2.e18

device  temp=300.0  area=9.e-6
solve   itmax=0  prec=3  inject=r-to-l


        -- to insure proper resolution of the wavefunction,
        --  watch a node in the well-region (node 38)

maxima  watch=38

print   tcoeff=*  format1=zen  format2=*  verbose=true
output  file=rtd  data=w
-------------------------------------------------------------------------------

Executing a total of  1 calculation(s).
```

Figure 5.3: Example of printed output

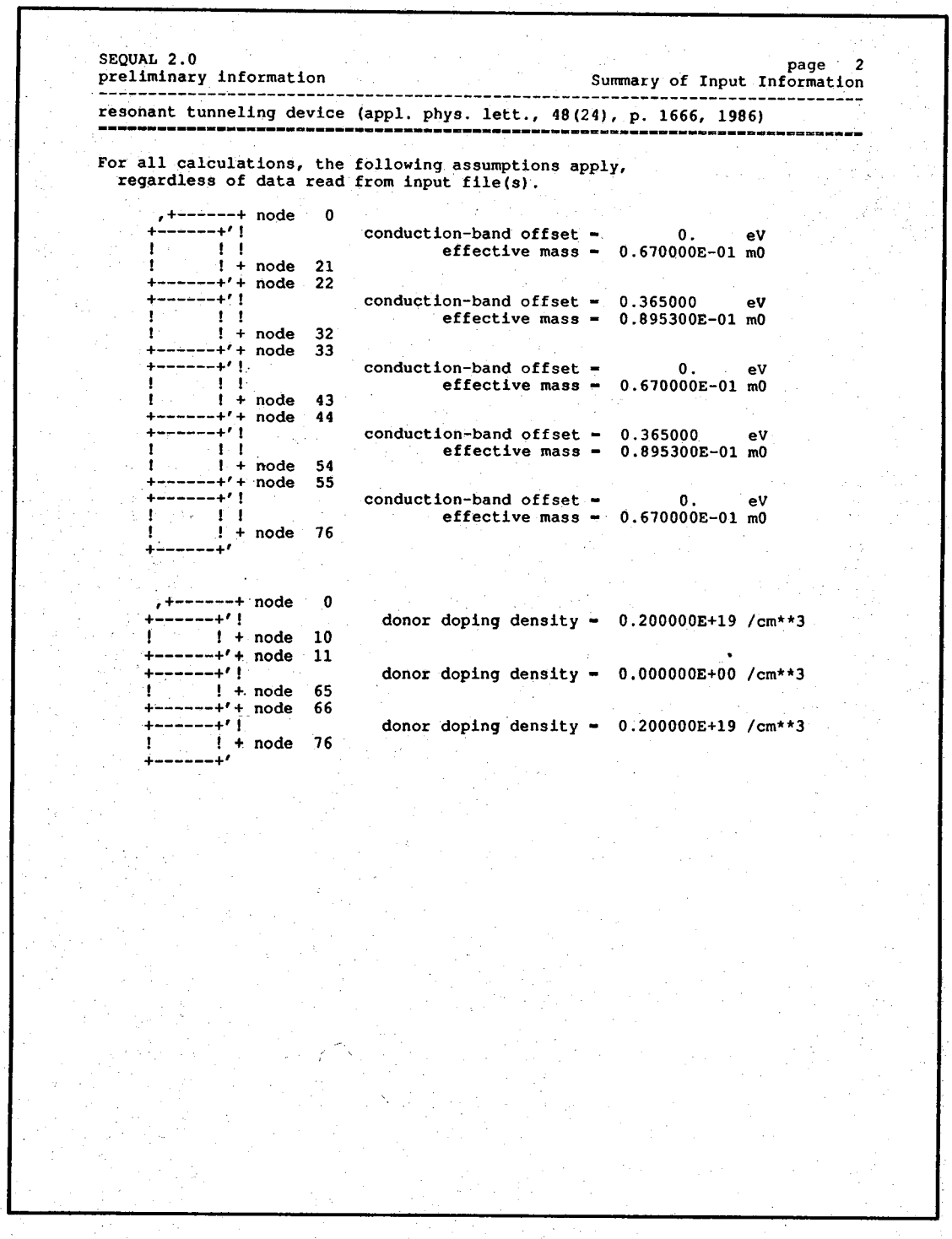


Figure 5.3: Example of printed output (continued)

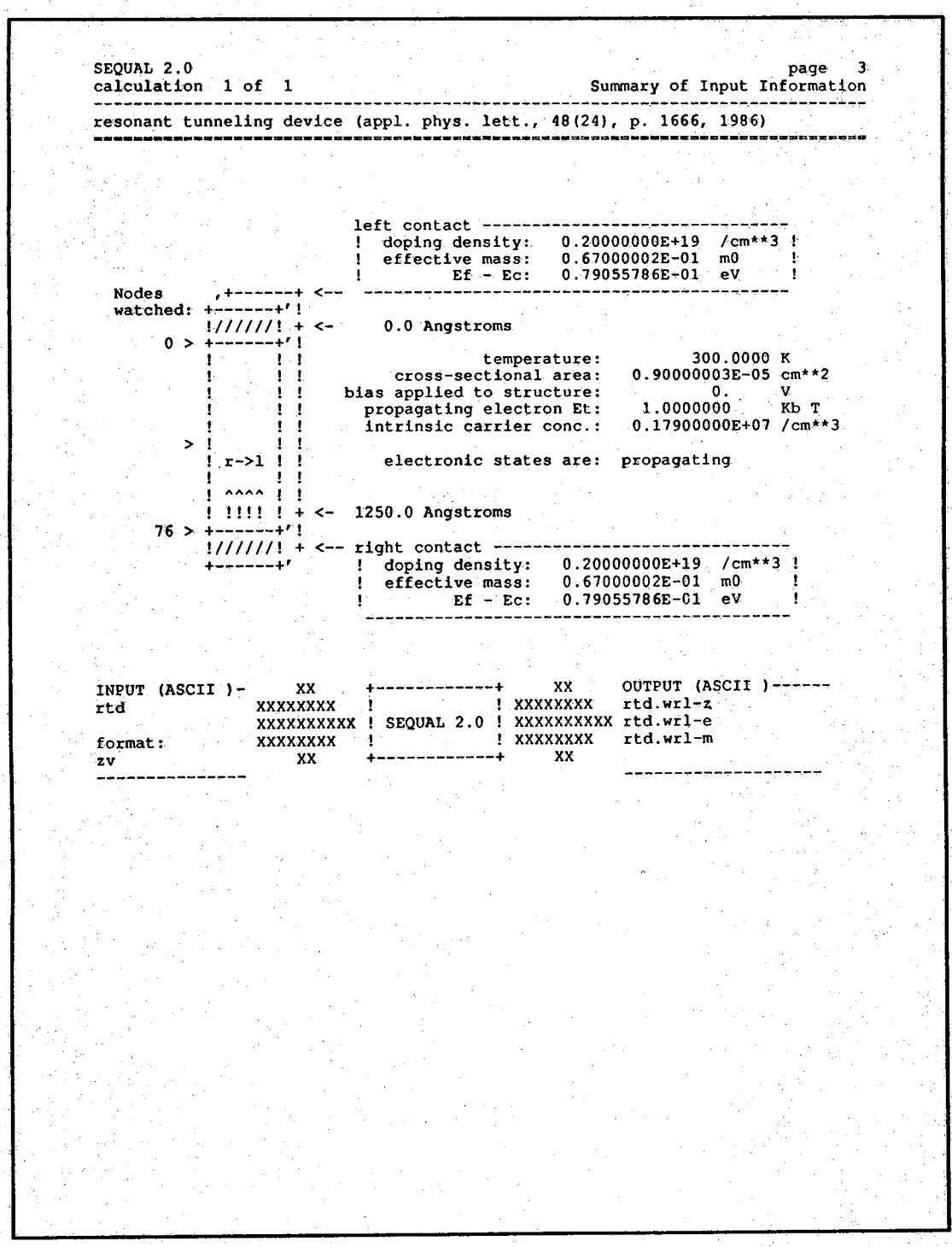
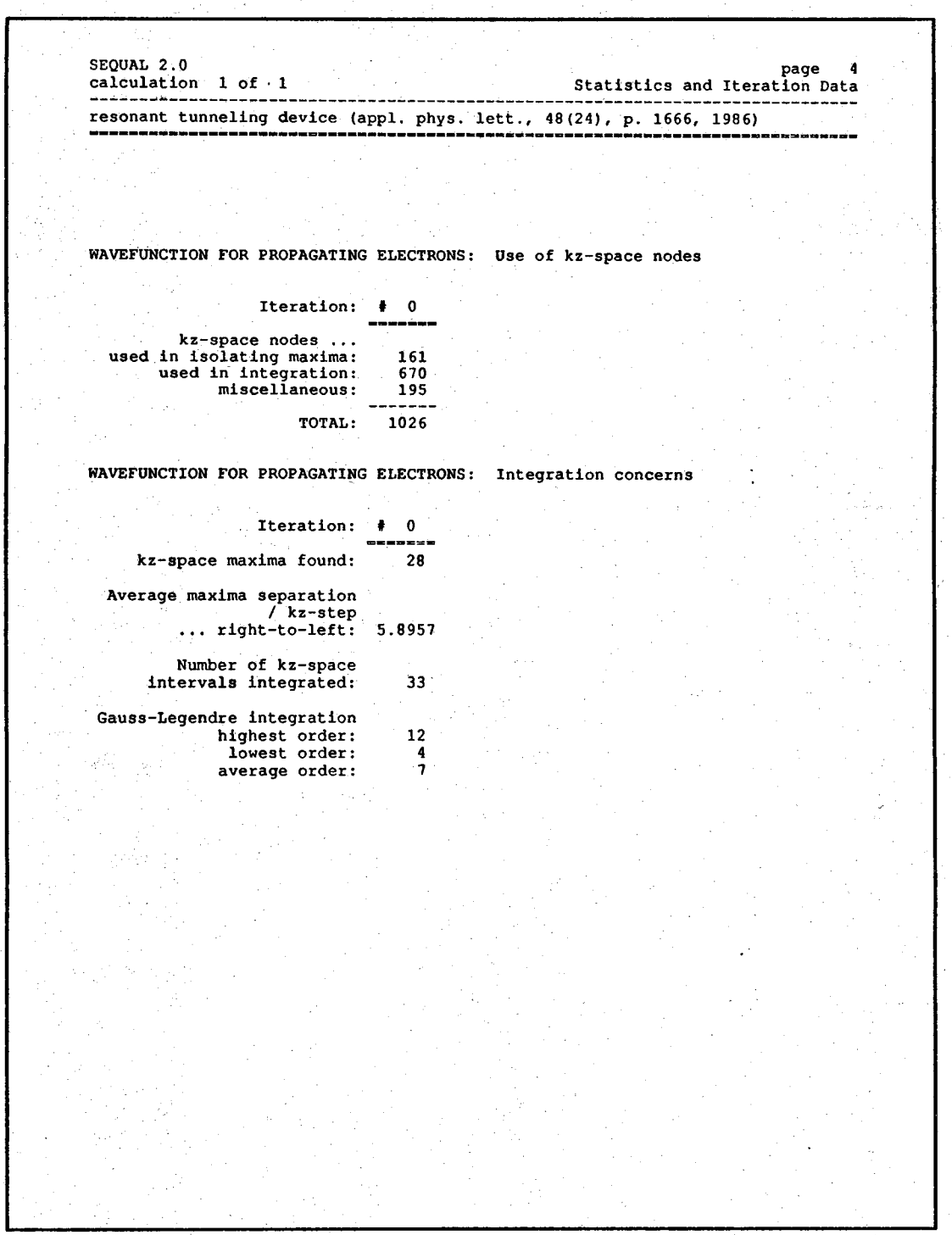


Figure 5.3: Example of printed output (continued)

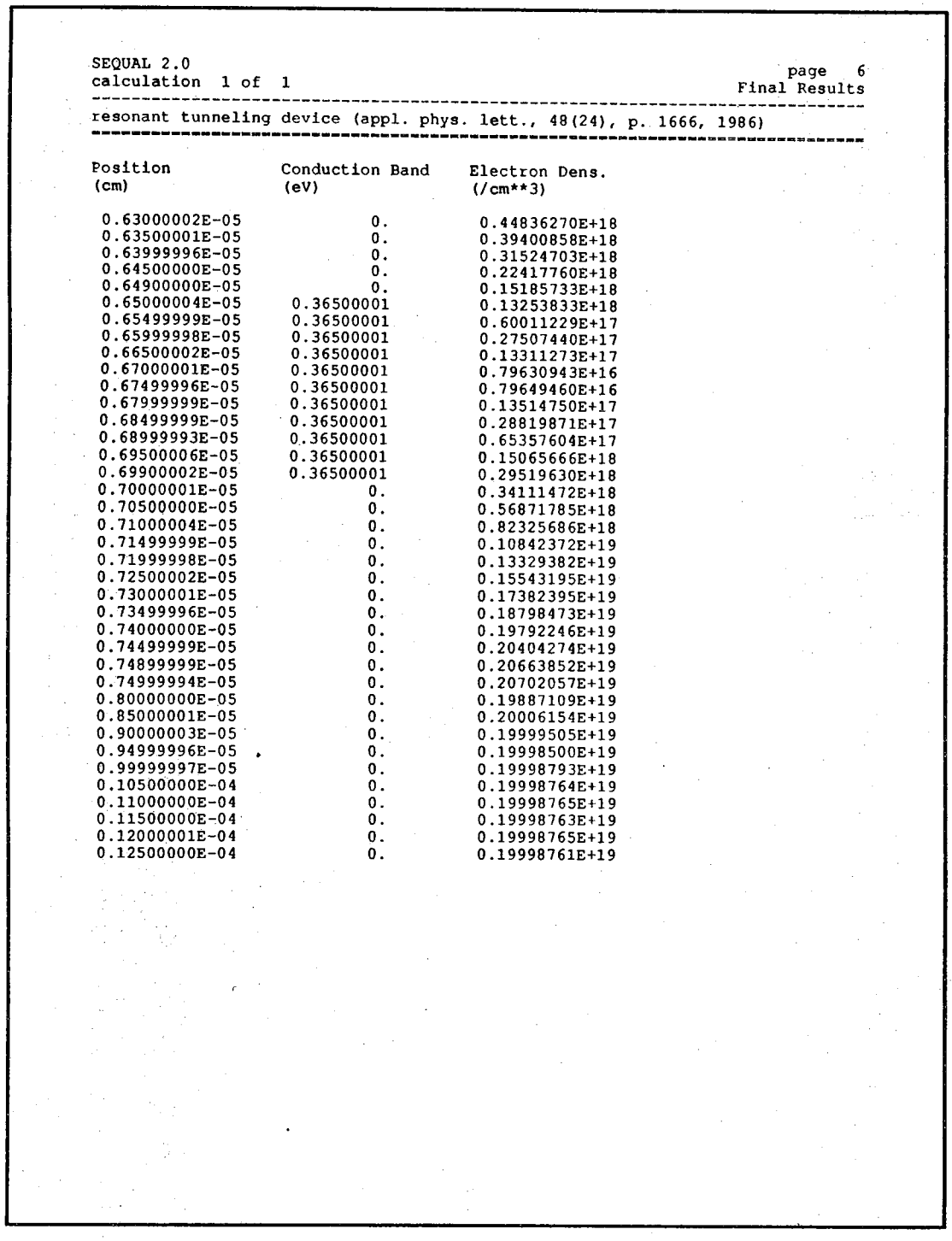


```
SEQUAL 2.0                                                    page   4
calculation  1 of  1                          Statistics and Iteration Data
-------------------------------------------------------------------------------
resonant tunneling device (appl. phys. lett., 48(24), p. 1666, 1986)
===============================================================================


WAVEFUNCTION FOR PROPAGATING ELECTRONS:  Use of kz-space nodes

                  Iteration:  #  0
                              =======
          kz-space nodes ...
  used in isolating maxima:      161
       used in integration:      670
             miscellaneous:      195
                              -------
                     TOTAL:     1026

WAVEFUNCTION FOR PROPAGATING ELECTRONS:  Integration concerns

                  Iteration:  #  0
                              =======
      kz-space maxima found:      28

   Average maxima separation
                  / kz-step
         ... right-to-left:  5.8957

         Number of kz-space
       intervals integrated:      33

  Gauss-Legendre integration
              highest order:      12
               lowest order:       4
              average order:       7
```

Figure 5.3: Example of printed output (continued)

```
SEQUAL 2.0                                                        page   5
calculation  1 of  1                                        Final Results
-------------------------------------------------------------------------------
resonant tunneling device (appl. phys. lett., 48(24), p. 1666, 1986)
===============================================================================


Current:                                  Voltage:
-------------------------------------     -------------------------------------
   LEFT-to-RIGHT:   0.000000E+00 A           In Input File:   0.000000E+00 V
   RIGHT-to-LEFT:  -0.137929E-01 A            Applied Bias:   0.000000E+00 V
                   --------------                            --------------
   Total Current:   0.137929E-01 A           Total Voltage:   0.000000E+00 V


Position            Conduction Band     Electron Dens.
(cm)                (eV)                (/cm**3)

        0.                      0.        0.13938995E+15
 0.50000000E-06                 0.        0.13938993E+15
 0.10000000E-05                 0.        0.13938994E+15
 0.15000002E-05                 0.        0.13938995E+15
 0.20000000E-05                 0.        0.13938994E+15
 0.24999999E-05                 0.        0.13938995E+15
 0.30000003E-05                 0.        0.13938995E+15
 0.35000000E-05                 0.        0.13938994E+15
 0.40000000E-05                 0.        0.13938995E+15
 0.45000002E-05                 0.        0.13938994E+15
 0.49900000E-05                 0.        0.13938994E+15
 0.49999999E-05                 0.        0.13938994E+15
 0.50499998E-05                 0.        0.13938994E+15
 0.51000002E-05                 0.        0.13938994E+15
 0.51500001E-05                 0.        0.13938995E+15
 0.52000000E-05                 0.        0.13938994E+15
 0.52500000E-05                 0.        0.13938994E+15
 0.52999999E-05                 0.        0.13938994E+15
 0.53499998E-05                 0.        0.13938994E+15
 0.54000002E-05                 0.        0.13938994E+15
 0.54500001E-05                 0.        0.13938994E+15
 0.54900001E-05                 0.        0.13938995E+15
 0.55000000E-05        0.36500001         0.13994561E+15
 0.55500000E-05        0.36500001         0.18066795E+15
 0.55999999E-05        0.36500001         0.31577197E+15
 0.56499998E-05        0.36500001         0.63506427E+15
 0.57000002E-05        0.36500001         0.13511151E+16
 0.57500001E-05        0.36500001         0.29412975E+16
 0.58000001E-05        0.36500001         0.64667889E+16
 0.58500000E-05        0.36500001         0.14281898E+17
 0.58999999E-05        0.36500001         0.31607775E+17
 0.59499998E-05        0.36500001         0.70023107E+17
 0.59899994E-05        0.36500001         0.13236001E+18
 0.60000007E-05                 0.        0.15169191E+18
 0.60500001E-05                 0.        0.24259597E+18
 0.61000001E-05                 0.        0.33233649E+18
 0.61500004E-05                 0.        0.40712321E+18
 0.61999999E-05                 0.        0.45546977E+18
 0.62499998E-05                 0.        0.46995659E+18
```

Figure 5.3: Example of printed output (continued)



SEQUAL 2.0 page 6
calculation 1 of 1 Final Results

resonant tunneling device (appl. phys. lett., 48(24), p. 1666, 1986)

Position (cm)	Conduction Band (eV)	Electron Dens. (/cm**3)
0.63000002E-05	0.	0.44836270E+18
0.63500001E-05	0.	0.39400858E+18
0.63999996E-05	0.	0.31524703E+18
0.64500000E-05	0.	0.22417760E+18
0.64900000E-05	0.	0.15185733E+18
0.65000004E-05	0.36500001	0.13253833E+18
0.65499999E-05	0.36500001	0.60011229E+17
0.65999998E-05	0.36500001	0.27507440E+17
0.66500002E-05	0.36500001	0.13311273E+17
0.67000001E-05	0.36500001	0.79630943E+16
0.67499996E-05	0.36500001	0.79649460E+16
0.67999999E-05	0.36500001	0.13514750E+17
0.68499999E-05	0.36500001	0.28819871E+17
0.68999993E-05	0.36500001	0.65357604E+17
0.69500006E-05	0.36500001	0.15065666E+18
0.69900002E-05	0.36500001	0.29519630E+18
0.70000001E-05	0.	0.34111472E+18
0.70500000E-05	0.	0.56871785E+18
0.71000004E-05	0.	0.82325686E+18
0.71499999E-05	0.	0.10842372E+19
0.71999998E-05	0.	0.13329382E+19
0.72500002E-05	0.	0.15543195E+19
0.73000001E-05	0.	0.17382395E+19
0.73499996E-05	0.	0.18798473E+19
0.74000000E-05	0.	0.19792246E+19
0.74499999E-05	0.	0.20404274E+19
0.74899999E-05	0.	0.20663852E+19
0.74999994E-05	0.	0.20702057E+19
0.80000000E-05	0.	0.19887109E+19
0.85000001E-05	0.	0.20006154E+19
0.90000003E-05	0.	0.19999505E+19
0.94999996E-05	0.	0.19998500E+19
0.99999997E-05	0.	0.19998793E+19
0.10500000E-04	0.	0.19998764E+19
0.11000000E-04	0.	0.19998765E+19
0.11500000E-04	0.	0.19998763E+19
0.12000001E-04	0.	0.19998765E+19
0.12500000E-04	0.	0.19998761E+19

Figure 5.3: Example of printed output (continued)

```
*************************************************************
.   A resonant tunneling device fabricated by Ray et. al.
.     (applied physics letters, 48(24), p. 1666, 1986) is
.     simulated.  Current is computed for several biases.
.     Results are not self-consistent.
*************************************************************

title          resonant tunneling device (appl. phys. lett., 48(24), p. 1666, 1986)


               -- input file "rtd" contains positions in angstroms

input          file=rtd format=zv
scale          cm=1.0e8

matter         nodes=21/32/43/54/76
 +             delec=0.0/0.365/0.0/0.365/0.0
 +             emass=0.067/0.08953/0.067/0.08953/0.067

doping         nodes=10/65/76  nd+=2.e18/0./2.e18


               -- compute current for biases:
               --   0 V, 0.05 V, 0.10 V, 0.15 V, 0.20 V, 0.25 V, 0.30 V

device         temp=300.0  area=9.e-6
 +             bias=0.00/0.05/0.10/0.15/0.20/0.25/0.30

solve          itmax=0  prec=3  inject=r-to-l
maxima         watch=38


print          tcoeff=*  format1=zen  format2=*  verbose=true
output         file=rtd00/rtd05/rtd10/rtd15/rtd20/rtd25/rtd30
 +             data=dt
```

Figure 5.4: Input deck for the application of several different biases

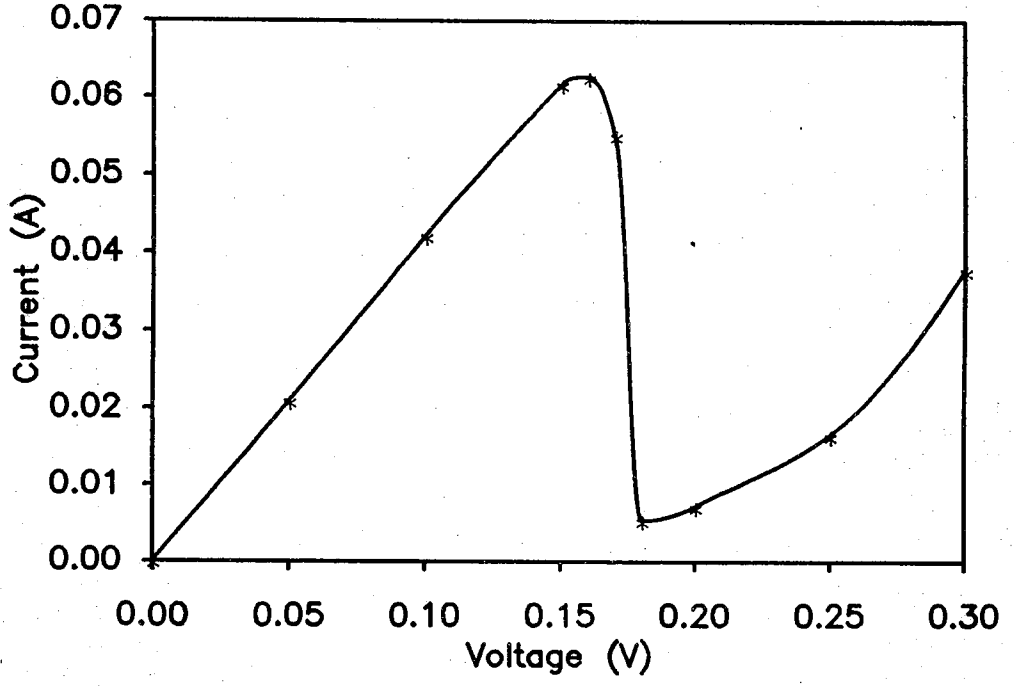


Figure 5.5: Current-voltage characteristic for the resonant tunneling device

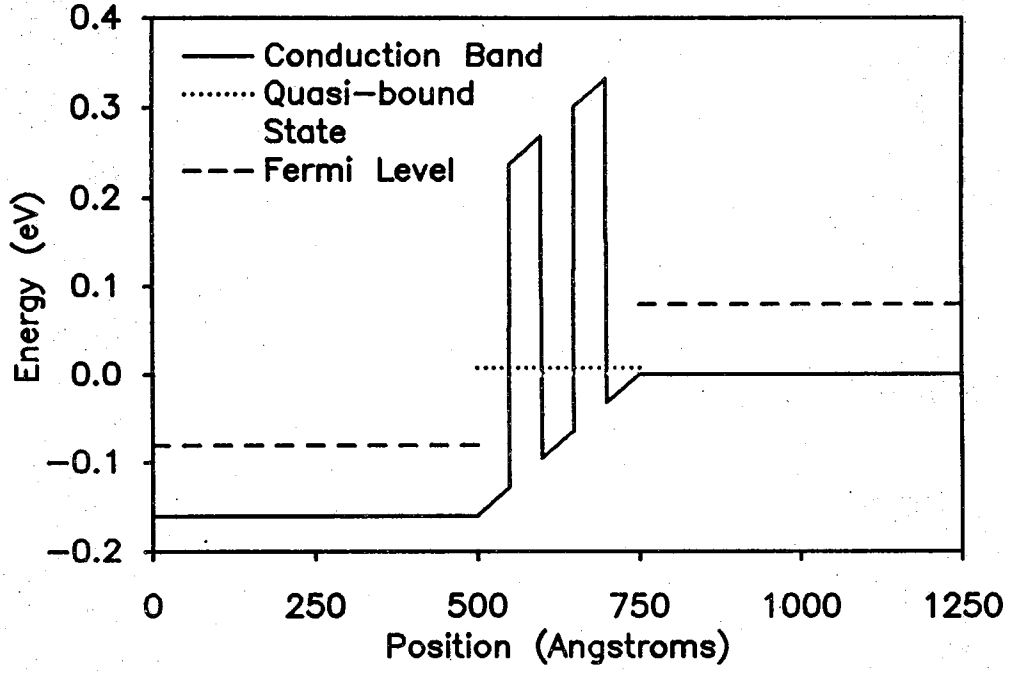


Figure 5.6: Conduction-band profile of the resonant tunneling device for the bias of peak current

6

Special Considerations

6.1. Installation Notes

SEQUAL is written in standard Fortran 77, and has been implemented at Purdue University on Sun™ workstations and the Dual VAX 11/780 machines. To ensure reasonable accuracy, floating-point representations should use at least 48 bits to describe the mantissa (fractional part). Therefore, typical installations must use double-precision complex variables. Although this is not allowed in standard Fortran 77, the *f77* compiler for the Dual VAX machines will automatically double the precision of all floating-point variables, when the argument "-r8" is specified. Furthermore, many compilers now recognize "double complex" as a data type. To account for these variations in Fortran 77 compilers, three versions of SEQUAL exist. One version, for machines with a large word-length, uses ordinary, single-precision complex variables. Another, for machines which recognize the "double complex" data type, uses double-precision complex variables. Finally, a third version exists for machines with insufficient precision and a strict compiler; this final version simulates the double-precision complex type by using pairs of double-precision variables.

For the analysis of bound states, SEQUAL relies upon *eigrf* in the International Mathematical and Statistical Library (IMSL). If this subroutine is not available to the user, it can be removed from the program code, and the bound-state analysis can be disabled. To accomplish these tasks, the user must comment-out the subroutine call, which appears in the subroutine *bstate*:

```
c
c
c  zzz zzz zzz  MACHINE DEPENDENT CODE  zzz zzz zzz
c
c  If IMSL routines are unavailable, comment out the
c    following line, so that SEQUAL can be compiled:
c
c     call eigrf(matrix,nodes,ia,ijob,val,vec,ia,wk,ier)
c
```

In addition, the constant *imsl* in the main program should be properly set, to disable the bound-state analysis:

```
c
c  zzz zzz zzz MACHINE DEPENDENT CODE zzz zzz zzz
c
c    imsl ... if IMSL routines are unavailable for a host
c             system, the parameter 'imsl' below should
c             be assigned the value '.false.'
c
      integer zimax, mxque, maxiter, maxii, errmax
      integer mxcard, mxkeys, mxarry, mxoind, mxord, nexts
      real maxreal
      logical imsl
      parameter ( zimax=251,
     +           maxiter=30,
                    .
                    .
                    .
     +             imsl=.false.  )
c
```

Notice that both of these corrections are clearly marked in the program code by the comment:

```
c  zzz zzz zzz MACHINE DEPENDENT CODE zzz zzz zzz
```

In SEQUAL, all machine-dependent constants are marked in this manner. For a proper installation, the user should search the program code for all appearances of this comment, and follow the instructions immediately following it. Since the vast majority of machine-dependent constants are used to check for overflow or underflow, the corrections should be obvious.

6.2. Pitfalls to Avoid

Among the most frustrating problems that a user might encounter are those which arise from the physics of a particular analysis. Suppose a (weary) user has created a device description and an input deck, has run SEQUAL, and has obtained results which appear to be in error. Resisting his impulse to burn this manual, the user would receive his reward in this section; it is devoted to relieving both confusion and misery.

At the heart of the propagating-state analysis lies the assumption that contacts are in local, thermodynamic equilibrium. Unless the potential profile is sufficiently flat near each contact, this assumption will be violated. To guarantee a proper analysis,

it is prudent to include "buffer" regions between the ideal contacts (nodes 0 and N) and the actual region of interest in the device. For example, in Chapter 5, the analysis of a resonant tunneling device included 500 Å buffer layers of GaAs, separating nodes 0 and N from the interior of the device. These buffer layers are merely extensions of the contacts, with the same material properties and the same density of ionized donors. They should be long enough to allow equilibrium to be restored: Electron density should return to the density of ionized donors, and the electrostatic potential should flatten out (*i.e.*, electric fields should decay to zero), within these regions. Because of the recursive dependence of both the electron density and the electrostatic potential, the use of buffer layers is especially important in self-consistent calculations. Unless equilibrium is restored near the contacts, errors in either the electron density or the electrostatic potential will feed back into the solution, making convergence an impossible goal.

Another source of feedback can further aggravate problems in convergence: If the contacts (or buffer layers) are lightly-doped with respect to the internal device structure, a self-consistent solution may be difficult to obtain. In this case, most of the band-bending (and most of the applied bias) will appear in buffer layers. Restoring equilibrium conditions near the contacts will be difficult. To some degree, the length of buffer layers can be increased; the analysis of long devices, however, is itself a difficult task, which will be described below.

In the same manner, non-equilibrium solutions may be difficult to obtain for conduction-band profiles which do not obstruct the flow of electrons. For instance, if the user attempts to apply a bias to a uniform block of semiconductor (for which the equilibrium conduction-band profile is completely flat), the results obtained will be nonsense. For a reasonable bias, the electron density will be higher at one contact, and lower at the other. In a real device, the flow of electrons is somewhat impeded by interactions with scattering mechanisms; in the limit of ballistic transport, however, there is nothing to impede the flow of electrons, aside from interactions with the potential. A conduction-band profile without reflective features might be thought of as representing a device with infinite conductance; because it is unable to support a bias, the user should not consider applying a bias.

In the analysis of bound states, the use of buffer layers is also important. As boundary conditions to the finite-difference solution of the Schrödinger equation, it was assumed that the wavefunction is zero at each end of the device (nodes 0 and N). If the wavefunction decays sufficiently within the buffer regions, this assumption is valid; otherwise, the solution of eigensystem may be in error.

From the preceding discussion, one might be tempted to include buffer layers of classical dimensions (say, ~ 1 μm) in every calculation. Unfortunately, this solution would create an even larger problem. As the overall device length increases, the wavefunction magnitude becomes a rapidly varying function of k_z. To insure an accurate integration of the wavefunction, SEQUAL integrates the k_z intervals between peaks in the wavefunction magnitude (see section 2.1.3). Therefore, the number of points in k_z space (and hence, the number of wavefunction solutions) increases dramatically with increasing device length. For devices with classical dimensions (~ 1 μm), the execution time of SEQUAL may be prohibitive. Typically, buffer regions 500 Å to 1000 Å long are sufficient to guarantee proper results. As a rule, it is more efficient to start with buffer layers that are too short, rather than too long.

Given the density of ionized donors, SEQUAL will compute the Fermi-level E_F for each contact (nodes 0 and N). Because the electron density calculated for the propagating-state solution is extremely sensitive to the Fermi-level, the proper E_F must be calculated within SEQUAL. A problem can arise, however, when using the output of classical analysis programs, which allow for non-parabolicity of the energy dispersion relation. For a device with contacts of differing materials, the Fermi-levels calculated by SEQUAL may not correspond to those found in the (non-parabolic) classical analysis. In this case, the device will appear to have a different bias in SEQUAL (*i.e.*, a different separation in the contact Fermi-levels) than it did in the classical analysis. This bias would be noted in the printed-output section of final results. When listing the bias across device, SEQUAL provides both the actual bias (*e.g.*, separation in contact Fermi-levels) and the applied bias. To remedy this situation, the user could apply a bias across the device which will counteract the erroneous Fermi-level separation.

Although (hopefully) the discussion above might be helpful, it is not intended to cover every conceivable problem. One final word of advice, from an author who has invested countless hours in the examination of program output: If SEQUAL says so, it is probably correct. More often than not, I have found a problem to stem from my own misconceptions and mistakes, rather than from the regimented computations of the program.

Appendix

Release 2.1

The following is a reprint of an update summarizing changes in SEQUAL for release 2.1. A few example calculations are presented, illustrating some of the changes in program controls.

1

SEQUAL Release 2.1

At the request of various users, several enhancements have been added to SEQUAL. The purpose of this document is to explain the additions, and to illustrate the use of new input keys in example calculations. Important changes are as follows:

- A card *kzgrid* was added to the input deck, allowing the user to specify the mesh in k_z-space (see SEQUAL User's Manual, p. 8). Although the automatic (default) mesh is recommended for most purposes, it is sometimes desirable to "zoom-in" on a particular region of the transmission coefficient in energy space; an example of this use is presented near the end of this document.

- A key *itvar* was added to the *solve* card, allowing the user more control over self-consistent calculations. Previously, the current density was used to determine convergence: When the required number of significant figures had been attained (specified by *prec* on the *solve* card), iteration was terminated. For equilibrium calculations, an exact cancellation of oppositely flowing currents is difficult to achieve. In this case, the current density should not be used to control convergence. Furthermore, the current density is identically zero for bound state calculations, and therefore provides no information. The new key *itvar* allows the user to specify which program variable, current density or electrostatic potential (or both), should determine convergence.

- For SEQUAL 2.0, results of a self-consistent calculation were written only upon convergence of the final solution. Because the execution time for a self-consistent calculation can be quite long, the partial results from each iteration are now written (and over-written), according to the output requests of the user (see card *output* in the user's manual).

- Previously, it was assumed that starting position for any device (i.e., the first value specified for "z" in an input file) was zero; this restriction has been removed. For many practical devices, an "interesting" region of the

potential is surrounded by large regions over which the potential is constant. Because the length of a device can substantially affect the execution time of SEQUAL, it is prudent to trim the output from classical analysis programs. It is therefore convenient that the position-space grid be allowed to start from any value.

- A (small) bug in the solution of Poisson's equation (subroutine POISS) was corrected. This particular problem occurred **only** when the electron density was zero at some node, since a necessary logarithm became undefined. This problem prevented a self-consistent calculation of bound states, because the bound state wave function vanishes at the ends of the device. An example of a self-consistent calculation for the bound states of a AlGaAs/GaAs interface is presented near the end of this document.

The remainder of this update contains the following. Changes necessary to upgrade SEQUAL for use with IMSL 10.0 are presented below. A summary of the syntax for the *kzgrid* and *solve* cards is presented, followed by two example problems illustrating their use. Example output is presented for analyses of a double-barrier resonant tunneling device, and an AlGaAs/GaAs interface.

Conversion to IMSL 10.0

Solution of the bound states in SEQUAL 2.1 requires a solution of the eigensystem for a real general matrix. A subroutine in the IMSL library is used for this purpose.[†] SEQUAL 2.1 is released to be compatible with IMSL 9.2, although the most current version of IMSL is 10.0. To upgrade SEQUAL for use with this version, the following changes must be made:

[†] If IMSL is unavailable, any comparable eigensystem solver could be substituted in the subroutine BSTATE. If no such alternative is available, the bound state solution must be removed from the program (see the Sequal User's Manual, p. 59).

- Immediately before the first executable line in the main program, insert the following:

```
c
c zzz zzz zzz  MACHINE DEPENDENT CODE  zzz zzz zzz
c
c If IMSL version 10.0 is not available, comment out
c the following lines:
c
      common /worksp/  rwksp
      real rwksp(126022)
c
c ---------------------------------------------------------------
c Create workspace for IMSL eigenvalue solver...
c ---------------------------------------------------------------
c
      if (imsl) then
        call iwkin(126022)
      endif
```

- At the start of subroutine BSTATE, add:

```
      external evcrg, epirg
      real epirg
```

- In subroutine BSTATE, the call to IMSL subroutine "eigrf" must be changed to the following two IMSL calls:

```
c
c  zzz zzz zzz  MACHINE DEPENDENT CODE  zzz zzz zzz
c
c  If IMSL routines are unavailable, comment out the
c    following 2 lines, so that SEQUAL can be compiled:
c
      call evcrg(nodes,matrix,zimax,val,vec,zimax)
      bsperf = epirg(nodes,nodes,matrix,zimax,val,vec,zimax)
```

kzgrid specify k_z-grid for propagating states

Default State:

kzgrid auto=true from=0.0 to=0.0 steps=1

auto The key *auto* is a switch controlling the k_z-grid specification. If true, the automatic (default) grid is selected, and all other *kzgrid* keys are ignored. If false, the remaining keys specify the k_z-grid. Points in the user-specified grid are uniformly distributed in k_z-space, and form the basis for isolating wavefunction maxima (see Sequal User's Manual, p. 8). Because the user-specified grid may not always be sufficiently dense, the automatic grid is highly recommended for typical calculations.

from

to The energy range for the user-defined k_z-space grid is specified by the values assigned to *from* and *to*, in units of eV. For a given computation, only a single energy range is allowed; however, results from several energy ranges can be obtained by multiple assignments to these two keys.

steps The interval of k_z-space corresponding to the energy range defined by *from* and *to* is broken into an equal number of intervals, according to the value assigned to *steps*. As for the automatic grid, the step size is uniform with respect to k_z, **not** energy. Both the user-defined and automatic grids form a skeletal grid, to which k_z-nodes corresponding to wave function maxima are added. It is this composite grid which is used for output of the transmission coefficient. Hence, the number of points on this grid may be larger than the value assigned to *steps*, and the spacing (even in k_z-space) may not always be uniform.

solve — specify parameters controlling the analysis

Default State:

solve prec=3 itmax=9 itvar=jv states=prop inject=both

prec The number of significant figures desired for important quantities can be assigned to the *prec* key of the *solve* card. "Important" quantities include the electron density, the current density, and the Fermi-level in each contact. Since SEQUAL will struggle admirably (at the expense of CPU time) to achieve whatever precision is specified, the value should be kept within reasonable limits.

itmax For a self-consistent analysis, SEQUAL solves iteratively for the electrostatic potential and the electron density. Convergence is achieved when the number of significant figures in some quantity (between iterations) settles down to the number requested. To avoid excessive use of CPU time in obtaining convergence, SEQUAL terminates iteration when the number of iterations exceeds the value assigned to *itmax*. Therefore, iteration can be suppressed by specifying zero as the maximum number of iterations. If results are written to output files (see description of the *output* card), iteration can be continued at any point, by using the output of a previous run as the input device description file (see Figure 4.2 in the Sequal User's Manual).

itvar Self-consistent calculations are terminated when the quantity specified by *itvar* achieves the requested precision. This quantity can be current density, electrostatic potential or both:

Value	Quantity which determines convergence
j	Current density
v	Electrostatic potential
jv, vj	Current density **and** electrostatic potential

For bound state and equilibrium calculations, current density is zero, and the electrostatic potential is the recommended quantity for determining convergence.

states Both propagating and bound electronic states can be considered in the solution of the Schrödinger equation, according to the value assigned to *states*:

Value	Implication
prop	Consider propagating electronic states
bound	Consider bound electronic states
all	Consider propagating and bound states

The default is to consider only the propagating states, since bound states do not contribute to current density. Furthermore, the population of bound states is correct only for devices in equilibrium. For small deviations from equilibrium, however, the bound-state result obtained is a reasonable approximation to the correct solution. The consideration of bound states is particularly important for a self-consistent solution. Because the electrostatic potential is determined from the electron density, the bound-state contribution (even if it is only approximate) can significantly alter the final result.

inject In the solution of Schrödinger's equation for propagating states, electrons can be injected into the device from two contacts. Each contact provides a separate contribution to both electron density and current density. The value assigned to *inject* determines which of the contributions will be calculated:

Value	Implication
l-to-r	Consider electrons propagating from left to right (inject from left contact)
r-to-l	Consider electrons propagating from right to left (inject from right contact)
both	Inject from both contacts

Because the vast majority of devices require an analysis with injection from both contacts, the default value is "both." In special cases,

however, the contribution from one contact may be insignificant. Consider, for example, the calculation of electron current for a p-n junction. Injection from the p-type contact is unnecessary, since the current component would be negligible.

2

Example 1

The following represents a typical example use of the *kzgrid* card. A double-barrier resonant tunneling structure is analyzed with the default k_z-space grid, and the resulting transmission coefficient is shown in Fig. 1 below. Squares mark the actual points output from SEQUAL. These points are a combination of the uniform (automatic) grid and extra k_z-points corresponding to wave function maxima. Near the first peak in the transmission coefficient, the k_z-grid appears to be a bit sparse. SEQUAL uses considerably higher resolution for integration of the wavefunction, although these additional points are not printed out for the transmission coefficient. Therefore, to increase the output resolution of the transmission coefficient, we are left with two alternatives. The first is to reduce the automatic step size by assigning a fraction to the *kscale* parameter (see detailed discussion of the *maxima* card in the Sequal User's Manual). Although this solves the problem, it is a tremendous waste of CPU time, since we are only interested in increasing the resolution in a narrow band of energy. Instead, we add a *kzgrid* card to original input deck, and obtain the output attached on the following pages. A plot of the transmission coefficient with enhanced resolution is shown in Fig 2.

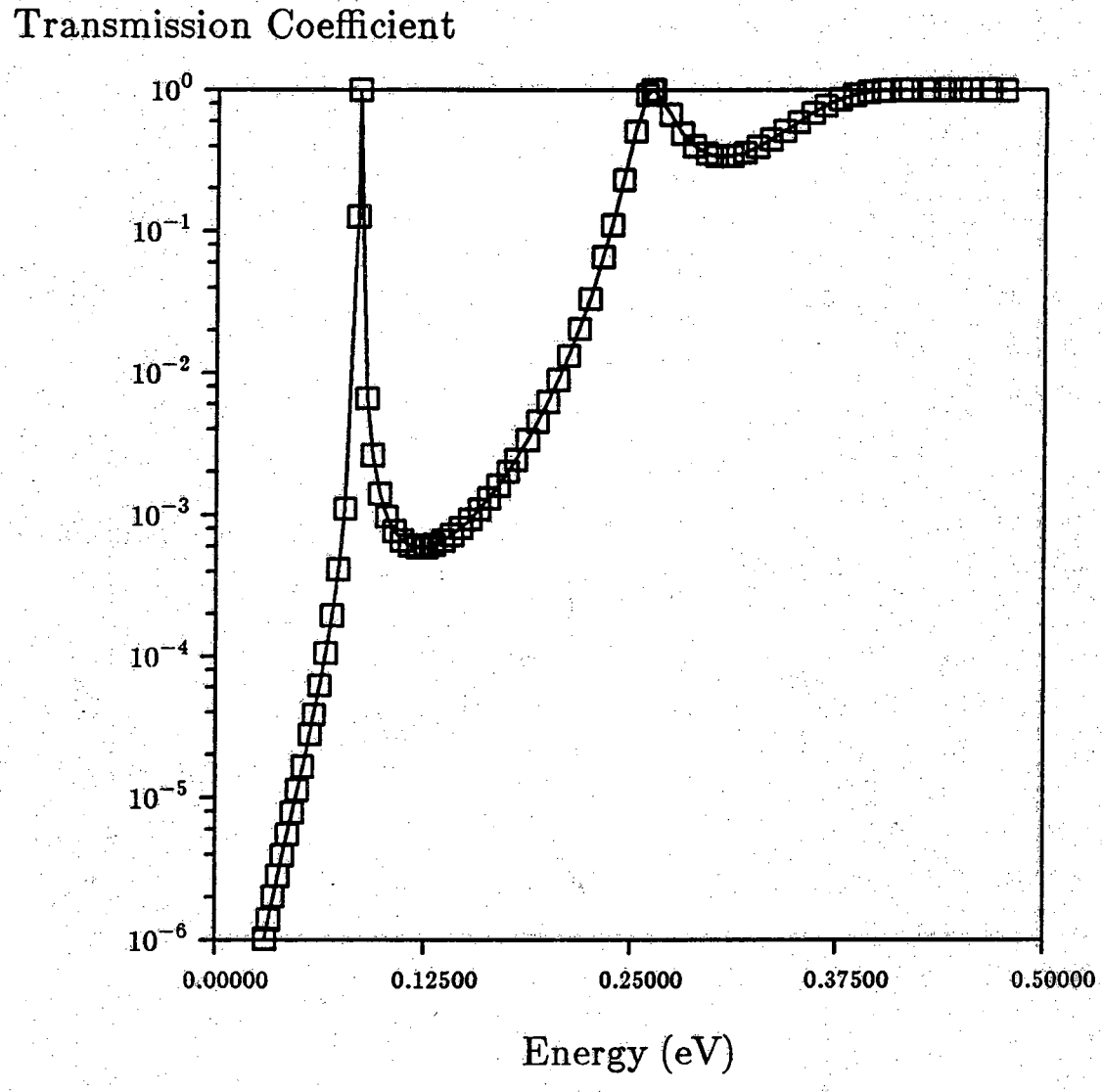


Fig. 1 A plot of transmission coefficient for a resonant tunneling structure. Squares (□) mark the grid points obtained from an analysis using the default (automatic) grid.

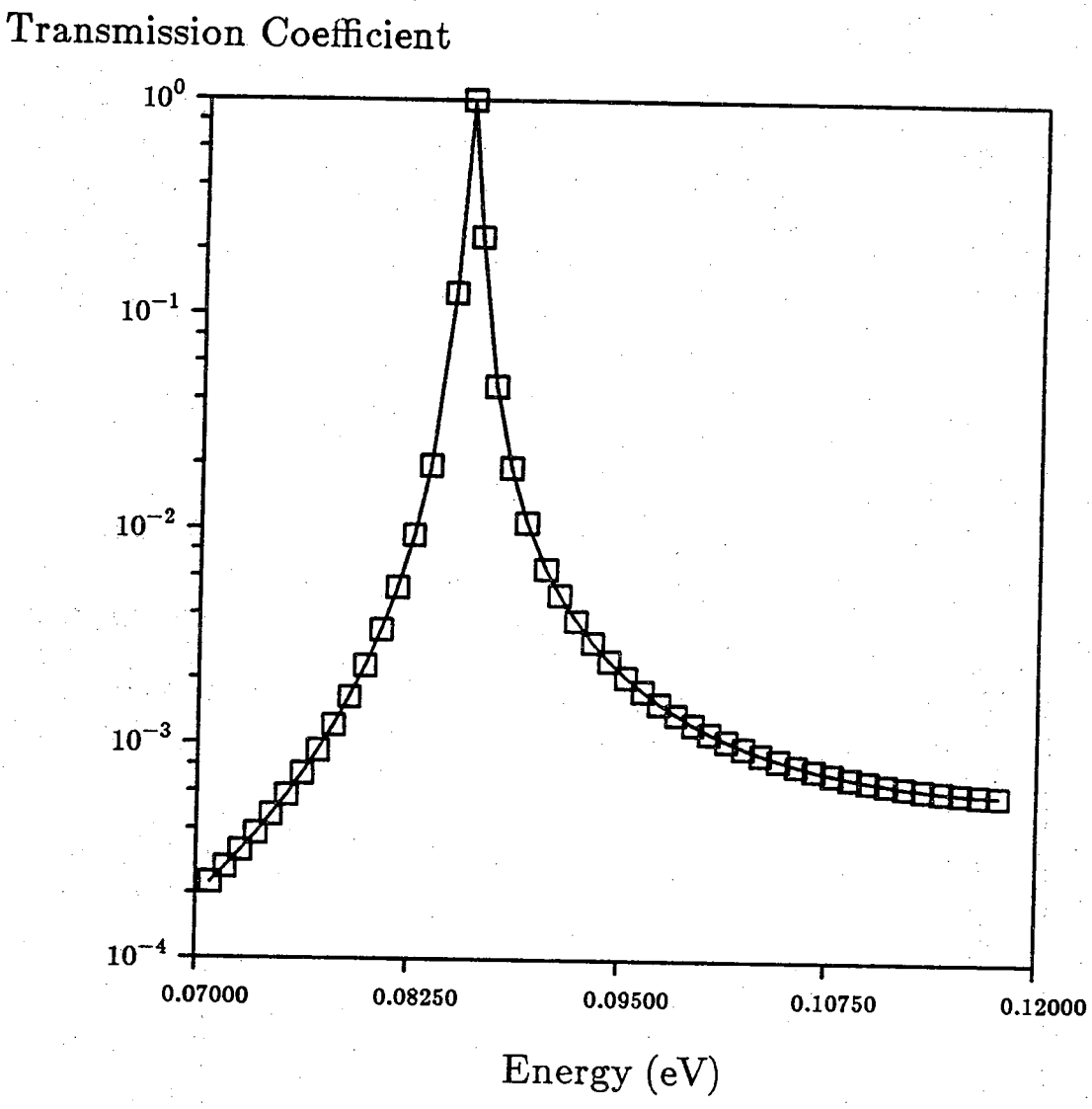
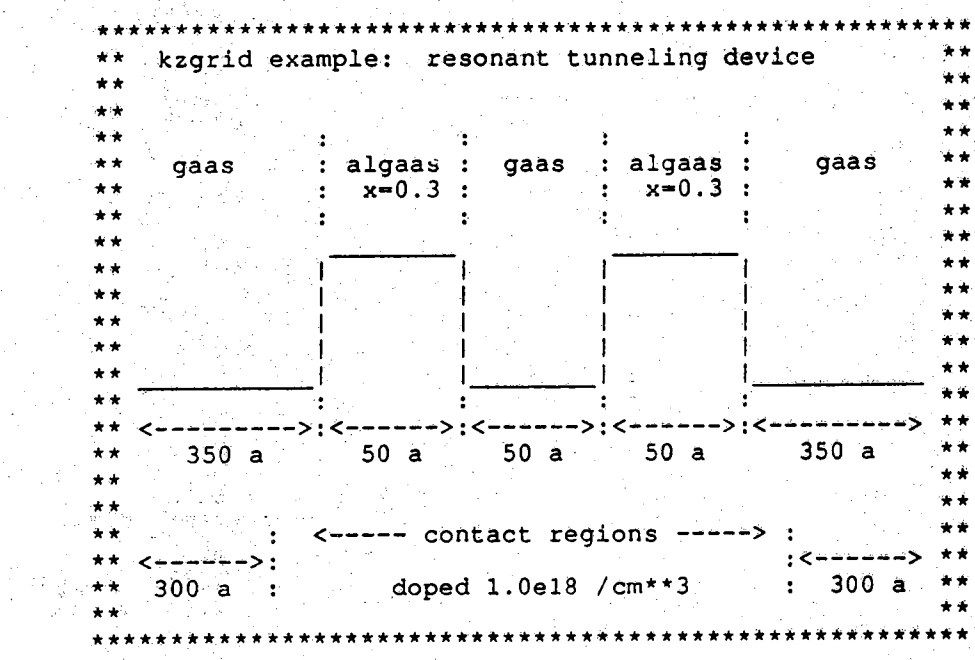


Fig. 2 A plot of transmission coefficient in a small energy range near the first resonance. Squares (□) mark the grid points obtained from an analysis with a user-specified grid.

```
------- XXXX  XXXXX   XXXX   XX  XX   XXXX   XX ------------------------------
      XX     XX     XX  XX  XX  XX  XX  XX  XX                      SEQUAL 2.1
----- XXX   XXXX   XX  XX  XX  XX  XXXXXX  XX ------------- Purdue University
        XX  XX     XX  X   XX  XX  XX  XX  XX                     August 1988
-- XXXX   XXXXX   XXX X   XXXXX  XX  XX  XXXXXX ------------------------------

SEQUAL:  input deck
-------------------------------------------------------------------------------
```




```
title   example of manual kzgrid selection

    >>>>  use output of previous self-consistent analysis  <<<<
    >>>>   for input in present calculation.               <<<<

input   file=rtd.z...k  format=zev???dmk

device  temp=300.0  bias=0.0
solve   itmax=0  prec=3  states=prop

    >>>>  select manual kz-space grid to zoom in on first   <<<<
    >>>>   quasi-bound state.                               <<<<

kzgrid  auto=false  from=0.07  to=0.12  steps=50

print   format1=* format2=* verbose=true
output  file=rtd1  data=dt
-------------------------------------------------------------------------------

Executing a total of  1 calculation(s).
```

example of manual kzgrid selection

```
                                left contact ------------------------------
                                !  doping density:   0.10000006E+19  /cm**3 !
                                !  effective mass:   0.67000031E-01  m0     !
                                !         Ef - Ec:   0.41833043E-01  eV     !
  Nodes      ,+-------+ <--     ------------------------------------------
  watched:  +-------+'!
            !///////! + <-      0.0 Angstroms
       0 >  +-------+'!
            ! !!!! ! !                      temperature:          300.0000 K
            ! vvvv ! !                cross-sectional area:     1.0000000  cm**2
            !      ! !           bias applied to structure:    0.00000000E+00 V
            ! l->r ! !             propagating electron Et:     1.0000000  Kb T
            !      ! !             intrinsic carrier conc.:    0.17900000E+07 /cm**3
            !      ! !
            ! r->l ! !               electronic states are:  propagating
            !      ! !
            ! ^^^^ ! !
            ! !!!! ! + <-    850.0 Angstroms
     101 >  +-------+'!
            !///////! + <-- right contact ------------------------------
            +-------+'      !  doping density:   0.10000006E+19  /cm**3 !
                            !  effective mass:   0.67000031E-01  m0     !
                            !         Ef - Ec:   0.41833043E-01  eV     !
                             ------------------------------------------
```

```
PROPAGATING Kz-GRID:  user-defined energy window:

         From energy:   0.70000052E-01 eV
           To energy:   0.12000006     eV
     Number of steps:              50
```

```
INPUT (ASCII )-      XX      +------------+      XX      OUTPUT (ASCII )------
rtd.z...k         XXXXXXXX   !            ! XXXXXXXX   rtd1.z...k
                  XXXXXXXXXX ! SEQUAL 2.1 ! XXXXXXXXXX rtd1.tlr-ec
format:           XXXXXXXX   !            ! XXXXXXXX   rtd1.trl-ec
zev???dmk            XX      +------------+      XX
---------------                                        --------------------
```

SEQUAL 2.1 page 3
calculation 1 of 1 Statistics and Iteration Data

example of manual kzgrid selection

WAVEFUNCTION FOR PROPAGATING ELECTRONS: Use of kz-space nodes

Iteration:	# 0
kz-space nodes ...	
used in isolating maxima:	73
used in integration:	288
miscellaneous:	108
TOTAL:	469

WAVEFUNCTION FOR PROPAGATING ELECTRONS: Integration concerns

Iteration:	# 0
kz-space maxima found:	6
Average maxima separation / kz-step	
... left-to-right:	3.6942
... right-to-left:	3.6958
Number of kz-space intervals integrated:	12
Gauss-Legendre integration	
highest order:	10
lowest order:	6
average order:	7

SEQUAL 2.1
calculation 1 of 1 page 4
Comments on the Calculation

example of manual kzgrid selection

TYPE	REMARK	SUGGESTION / EXPLANATION
Warning	Using a manual kz-grid for injection of electrons in propagating states.	Electron/current density obtained may be a fraction of the proper value.

```
SEQUAL 2.1                                                        page   5
calculation  1 of  1                                        Final Results
-------------------------------------------------------------------------
example of manual kzgrid selection
=========================================================================
```

Current:		Voltage:	
LEFT-to-RIGHT:	-0.490938E+04 A	In Input File:	0.000000E+00 V
RIGHT-to-LEFT:	-0.490959E+04 A	Applied Bias:	0.000000E+00 V
Total Current:	0.210938E+00 A	Total Voltage:	0.000000E+00 V

3

Example 2

As a demonstration of the *itvar* key, we perform a self-consistent analysis of bound states at an AlGaAs/GaAs interface. Because we are considering only bound states, current density is zero. In this case, the electrostatic potential alone should determine convergence. To specify this, we include "*itvar*=v" on the *solve* card. Output for this example is presented on pages that follow. (For a schematic view of the device, see output page 1.)

The calculated sheet density for electrons confined near the interface is $n_s = 1.74 \times 10^{12}$ cm^{-2}. It is important to recognize that this density includes electrons in *both* AlGaAs and GaAs layers, as illustrated in Figs. 3 and 4. Fig. 3 presents the self-consistent conduction band profile, and Fig. 4, the corresponding electron density, for a region of the structure near the interface. Although typically the AlGaAs layer is assumed to be depleted, in this case a substantial amount of the sheet density n_s can be attributed to electrons in the AlGaAs layer.

Note that, according to SEQUAL, the device appears to be biased (see output pages 5 and 8 on the following pages). This is because the input structure was taken from FISH1D, a semiclassical heterojunction analysis program. In FISH1D, all valleys (including Γ and X) are taken into account to determine the contact Fermi potentials, whereas only the Γ-valley is considered in SEQUAL. Therefore, a "bias" appears across the structure in SEQUAL, although the same structure is in equilibrium for FISH1D. The "bias" can be removed in SEQUAL by applying additional bias to counteract it. However, in this example that is unnecessary. Bound states are populated according to the Fermi level of the **back** contact, so this false "bias" can be ignored.

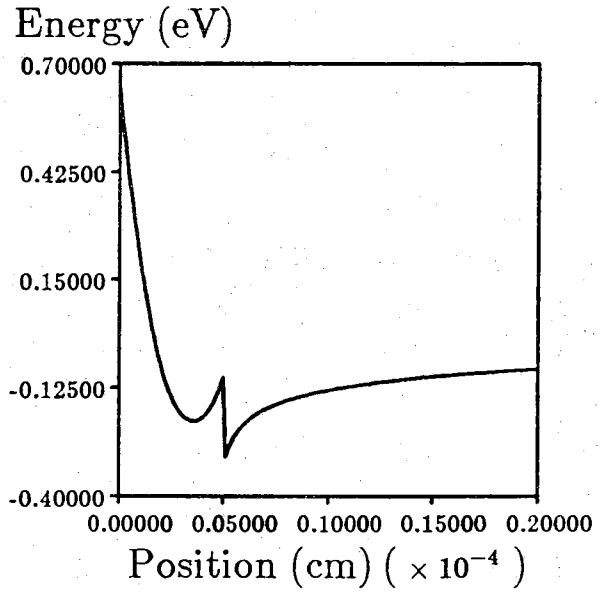


Fig. 3 The self-consistent conduction band diagram for an AlGaAs/GaAs interface, obtained by considering only bound states.

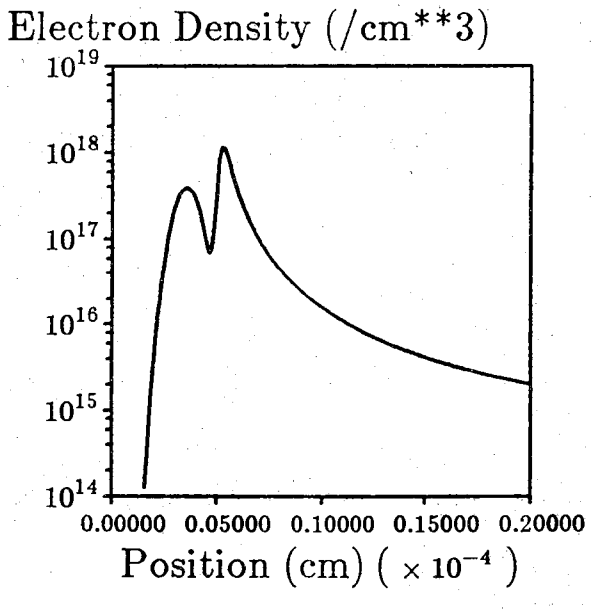


Fig. 4 The total electron density in all bound states, for the region of the conduction band shown above. Notice that the electron density in the AlGaAs layer (left hump) is substantial.

```
------- XXXX  XXXXX   XXXX   XX  XX   XXXX   XX ------------------------------
      XX     XX     XX  XX  XX  XX  XX  XX  XX                        SEQUAL 2.1
----- XXX   XXXX   XX  XX  XX  XX  XXXXXX  XX ------------- Purdue University
       XX  XX     XX  X   XX  XX  XX  XX  XX                        August 1988
-- XXXX   XXXXX   XXX X   XXXXX  XX  XX  XXXXXX ------------------------------
```

SEQUAL: input deck

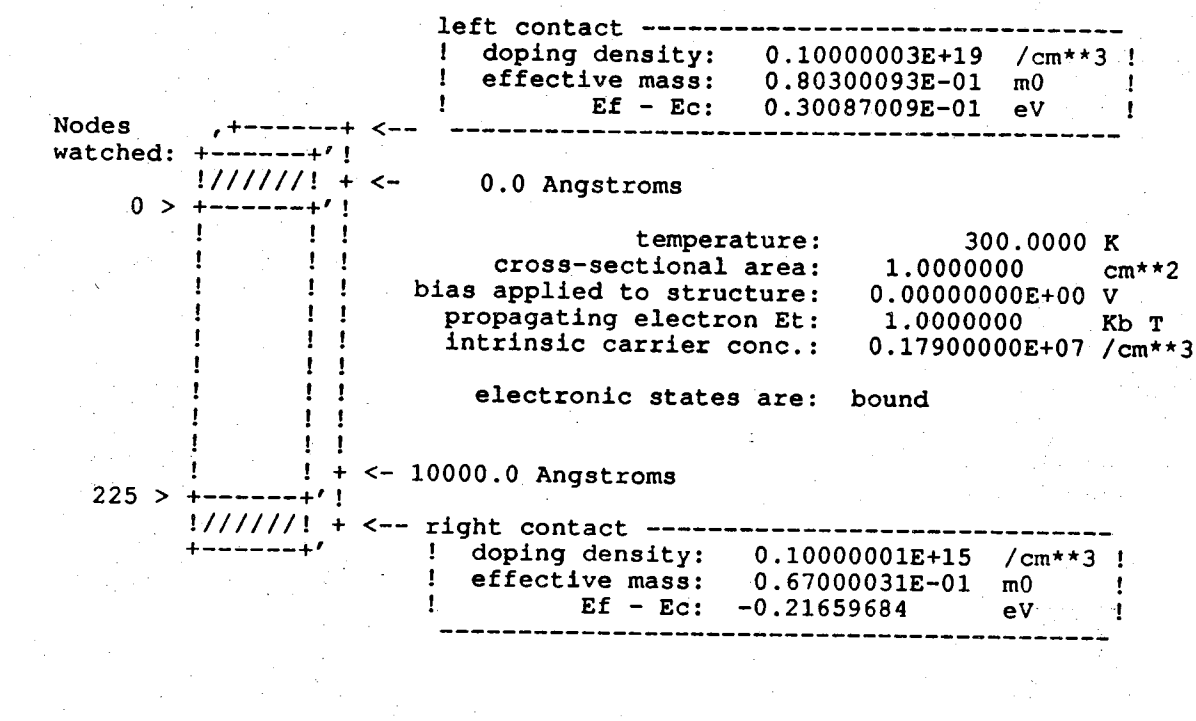
```
**********************************************************
**  bound state example:  algaas / gaas interface       **
**                                                      **
**   -------------------------                          **
**  |    schottky barrier     | <-- phi = 0.86         **
**  |-------------------------| - - -                  **
**  |     algaas (x=0.3)      |   ^                    **
**  |                         |   |  500 angst         **
**  |   nd = 1.0e18 /cm**3    |   v                    **
**  |-------------------------| - - -                  **
**  |          gaas           |   ^                    **
**  |                         |   |  9500 angst        **
**  |   nd = 1.e14 /cm**3     |   v                    **
**   -------------------------  - - -                  **
**                                                      **
**********************************************************
```

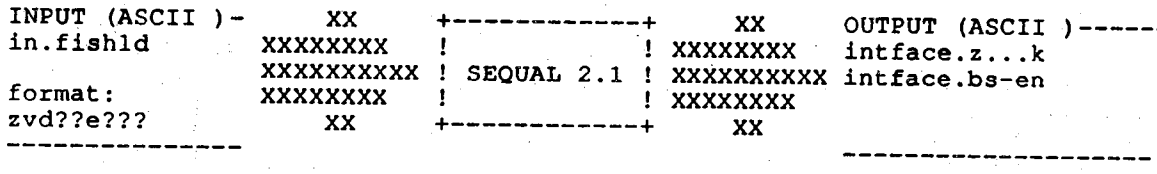
```
title   bound-state example:  algaas/gaas interface

    >>>>  output from fishld has position in microns.   <<<<
    >>>>  specify scale factor for proper length scale. <<<<

input   file=in.fishld  format=zvd??e???
scale   cm=1.e4

matter  nodes=50/225  emass=0.0803/0.067  krel=11.9/12.847
doping  nodes=50/225  nd+=1.e18/1.e14

device  temp=300.0  bias=0.0

    >>>>  for bound states, there is no current.  set itvar=v  <<<<
    >>>>   to use convergence of the electrostatic potential   <<<<
    >>>>   as the criterion for a self-consistent solution.    <<<<

solve   itmax=10  prec=3  states=bound  itvar=v

print   format1=* format2=* verbose=true
output  file=intface  data=db
```

```
Executing a total of  1 calculation(s).
```

```
SEQUAL 2.1                                                        page   2
preliminary information                          Summary of Input Information
-------------------------------------------------------------------------------
bound-state example:  algaas/gaas interface
===============================================================================

For all calculations, the following assumptions apply,
  regardless of data read from input file(s).

     ,+-------+ node   0
    +-------+'!                      effective mass =  0.803001E-01 m0
    !       ! !                    dielectric constant =   11.9000     e0
    !       ! + node  50
    +-------+'+ node  51
    +-------+'!                      effective mass =  0.670000E-01 m0
    !       ! !                    dielectric constant =   12.8470     e0
    !       ! + node 225
    +-------+'


     ,+-------+ node   0
    +-------+'!                  donor doping density =  0.100000E+19 /cm**3
    !       ! + node  50
    +-------+'+ node  51
    +-------+'!                  donor doping density =  0.100000E+15 /cm**3
    !       ! + node 225
    +-------+'
```

bound-state example: algaas/gaas interface





SEQUAL 2.1 page 4
calculation 1 of 1 Statistics and Iteration Data

bound-state example: algaas/gaas interface

Iteration Number	Current (A)	Significant Figs in Current	Max Change in V(z)	Significant Figs in V(z)
1	0.0000000E+00	3	0.946E-02	1
2	0.0000000E+00	3	0.197E-02	2
3	0.0000000E+00	3	0.509E-03	3

SEQUAL 2.1
calculation 1 of 1 page 5
Comments on the Calculation

bound-state example: algaas/gaas interface

TYPE	REMARK	SUGGESTION / EXPLANATION
Warning	Obtaining convergence in a self-consistent calculation may be difficult.	Contacts should be more heavily doped than the internal device structure.
Caution	Population of bound-states may be incorrect, since device is under bias.	Device must be in (or near) equilibrium for Fermi-Dirac statistics to apply.

```
SEQUAL 2.1                                                        page   6
calculation  1 of  1                          Bound-State Calculation
-------------------------------------------------------------------------
bound-state example:  algaas/gaas interface
=========================================================================
```

NOTE: Confidence in the bound-state solution is high

Bound-State Energy (eV)	Electron Dens. (/cm**2)	Occupation (% of total)
-0.23346710	0.78350470E+12	45.018539
-0.19914645	0.35672346E+12	20.496582
-0.18572903	0.19334090E+12	11.108964
-0.16791409	0.12185246E+12	7.0013876
-0.15970147	0.77001327E+11	4.4243374
-0.14297843	0.41610449E+11	2.3908491
-0.13474482	0.34421989E+11	1.9778156
-0.12861872	0.24305697E+11	1.3965540
-0.11890638	0.16508072E+11	0.94851923
-0.11065400	0.12072239E+11	0.69364566
-0.10395455	0.95209677E+10	0.54705477
-0.99294603E-01	0.83339674E+10	0.47885251
-0.95357955E-01	0.69041439E+10	0.39669758
-0.90595961E-01	0.55891395E+10	0.32114017
-0.85954428E-01	0.46443397E+10	0.26685411
-0.81701398E-01	0.39352571E+10	0.22611165
-0.77840924E-01	0.33894725E+10	0.19475198
-0.74338734E-01	0.29628815E+10	0.17024100
-0.71162641E-01	0.26251244E+10	0.15083414
-0.68283856E-01	0.23555379E+10	0.13534427
-0.65848231E-01	0.21486211E+10	0.12345529
-0.63814104E-01	0.20005722E+10	0.11494869
-0.61324317E-01	0.18459904E+10	0.10606676
-0.59660703E-01	0.16749463E+10	0.96238911E-01
-0.58285769E-01	0.16415347E+10	0.94319165E-01
-0.55165745E-01	0.14038211E+10	0.80660641E-01
-0.54835804E-01	0.14185697E+10	0.81508040E-01
-0.51128648E-01	0.12003448E+10	0.68969309E-01
-0.50750017E-01	0.12002565E+10	0.68964243E-01
-0.47476452E-01	0.10425728E+10	0.59904065E-01
-0.46183787E-01	0.10021391E+10	0.57580810E-01
-0.44137236E-01	0.91646848E+09	0.52658349E-01
-0.41250527E-01	0.82743885E+09	0.47542900E-01
-0.41059058E-01	0.81377587E+09	0.46757862E-01
-0.38195815E-01	0.72830413E+09	0.41846838E-01
-0.36037434E-01	0.67760666E+09	0.38933869E-01
-0.35514742E-01	0.65661005E+09	0.37727434E-01
-0.32983769E-01	0.59524966E+09	0.34201797E-01
-0.30652590E-01	0.55390106E+09	0.31825993E-01
-0.30578151E-01	0.54215910E+09	0.31151343E-01
-0.28274480E-01	0.49596058E+09	0.28496861E-01
-0.26054014E-01	0.45527603E+09	0.26159197E-01
-0.25317345E-01	0.45730099E+09	0.26275564E-01
-0.23898955E-01	0.41872691E+09	0.24059173E-01
-0.21793514E-01	0.38631347E+09	0.22196755E-01

bound-state example: algaas/gaas interface

Bound-State Energy (eV)	Electron Dens. (/cm**2)	Occupation (% of total)
-0.20363860E-01	0.38428851E+09	0.22080414E-01
-0.19726913E-01	0.35665690E+09	0.20492751E-01
-0.17682794E-01	0.32975872E+09	0.18947251E-01
-0.15651178E-01	0.30423834E+09	0.17480899E-01
-0.15441049E-01	0.31453030E+09	0.18072255E-01
-0.13621282E-01	0.28147712E+09	0.16173095E-01
-0.11583120E-01	0.26009414E+09	0.14944464E-01
-0.97666979E-02	0.24839613E+09	0.14272321E-01
-0.95288977E-02	0.24009014E+09	0.13795078E-01
-0.74497312E-02	0.22146555E+09	0.12724947E-01
-0.53382628E-02	0.20421982E+09	0.11734042E-01
-0.34537381E-02	0.19271066E+09	0.11072751E-01
-0.31791739E-02	0.18766365E+09	0.10782760E-01
-0.81478781E-03	0.17179670E+09	0.98710805E-02
TOTAL:	0.17404044E+13 /cm**2 in 59 bound-states	

```
SEQUAL 2.1                                                    page   8
calculation  1 of  1                                    Final Results
-------------------------------------------------------------------------
bound-state example:  algaas/gaas interface
=========================================================================
```

Current:		Voltage:	
LEFT-to-RIGHT:	0.000000E+00 A	In Input File:	-0.888461E+00 V
RIGHT-to-LEFT:	0.000000E+00 A	Applied Bias:	0.000000E+00 V
Total Current:	0.000000E+00 A	Total Voltage:	-0.888461E+00 V