



Green's function calculation of Ballistic Electron Emission Microscopy currents (BEEM v2.1)

K. Reuter^{a,c}, P.L. de Andres^a, F.J. Garcia-Vidal^b, D. Sestovic^b, F. Flores^b, K. Heinz^c

^a Instituto de Ciencia de Materiales (CSIC), Cantoblanco, E-28049 Madrid, Spain

^b Department de Fisica Teorica de la Materia Condensada (UAM), Universidad Autonoma de Madrid, E-28049 Madrid, Spain

^c Lehrstuhl für Festkörperphysik, Universität Erlangen-Nürnberg, D-91058 Erlangen, Germany

Received 24 February 1999; accepted 25 October 1999

Abstract

We describe the FORTRAN-90 program BEEM v2.1 for the computation of real and reciprocal space currents in Ballistic Electron Emission Microscopy (BEEM), and Spectroscopy (BEES). Based on a LCAO Hamiltonian, the equilibrium one-particle Green's functions of a semi-infinite crystal are obtained using a decimation technique. By means of Keldysh's formalism, the STM tip is coupled to this crystal, and expressions for BEEM currents in real and reciprocal space are obtained. Finally, a Dyson-like equation and a self-consistent charge neutrality condition are applied to incorporate reconstructed or relaxed layers in the surface region. Running demos for Au and CoSi₂ are provided. © 2000 Published by Elsevier Science B.V. All rights reserved.

PROGRAM SUMMARY

Title of program: BEEM

Catalogue identifier: ADLR

Program Summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADLR>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Computer: Alpha DEC3000/400

Operating system: OSF1 DEC-Unix v4.0

Programming language: FORTRAN 90

No. of bytes in distributed program including test data, etc.:
834 471 bytes

High-speed storage required for the test run: 32 Mbytes

Disk storage required: a scratch area of at least 300 Mbytes is used for the test run

No. of bits in a word: 64

Keywords: Ballistic Electron Emission Microscopy, Green's functions, decimation, layer-doubling, surface electronic structure

Nature of problem

We describe the FORTRAN-90 program BEEM v2.1 for the computation of real and reciprocal space currents in Ballistic Electron Emission Microscopy (BEEM), and Spectroscopy (BEES) [1–4].

Method of solution

Based on a LCAO Hamiltonian, the equilibrium one-particle Green's Functions of a semi-infinite crystal are obtained using a decimation technique. By means of Keldysh's formalism, the STM

tip is coupled to this crystal, and expressions for BEEM currents in real and reciprocal space are obtained. Finally, a Dyson-like equation and a self-consistent charge neutrality condition are applied to incorporate reconstructed or relaxed layers in the surface region.

Typical running time

The test run took about 1 hr for Au and about 6 hrs for CoSi₂ (once the decimation is stored, the run takes around 0.5–0.6 hrs for both Au and CoSi₂).

References

- [1] F.J. García-Vidal, P.L. de Andres, F. Flores, Phys. Rev. Lett. 76 (1996) 807.
- [2] P.L. de Andres, F.J. García-Vidal, D. Sestovic, F. Flores, Phys. Scripta T 66 (1996) 277.
- [3] K. Reuter, P.L. de Andres, F.J. García-Vidal, D. Sestovic, F. Flores, K. Heinz, Phys. Rev. B 58 14036 (1998).
- [4] K. Reuter, F.J. García-Vidal, P.L. de Andres, F. Flores, K. Heinz, Phys. Rev. Lett. 81 (1998) 4963.

LONG WRITE-UP

1. Introduction

Ballistic Electron Emission Microscopy (BEEM) is a technique that extends the Scanning Tunneling Microscope (STM) to allow the study of buried objects like interfaces. The STM tip is used as a local electron gun, injecting electrons inside a narrow area into a metallic thin film, that usually has been deposited on a semiconductor substrate forming a metal-semiconductor (MS) interface [1–3]. In close analogy to the STM, the apparatus can be operated in two different but complementary modes: scanning a BEEM image, i.e. $J(r_{\parallel})$ while crossing over the metal-semiconductor interface for a fixed voltage in the junction, or measuring spectroscopic $J(E)$ curves for a fixed tip position. The combination of these two operating modes makes the technique highly valuable to study the MS interface. Moreover, the wealth of information that this experiment carries about the metallic thin film itself is also becoming more and more appreciated. In particular, its potential to study electronic transport in metallic thin films has recently been recognized [3–5], a problem that can compete in technological importance with the study of the Schottky barrier.

The first physical picture of the technique was based on the ballistic passage of electrons through the metal, i.e. electrons were not supposed to interact with the metal, and the model is summarized by free space propagation at an energy dictated by the bottom of the metallic band and the tip voltage. While this picture is simple and convenient, allowing to simulate the system by E -space Monte Carlo techniques, it cannot easily be reconciled with a number of experimental and theoretical facts. One of the first important discrepancies emerged about the size of the propagating electron beam. Some experimental evidence [6] indicated an effective size off the theoretical predictions by about a factor 10. Furthermore, many experiments had been performed on thin films of transition metals grown in fcc(111) orientations with widths of around 100 Å: it is well known that these metals exhibit a band gap for electronic propagation in (111) directions at energies in the vicinity of the Fermi-level, so the theoretical question of whether or not the electrons could propagate perpendicularly through such films without feeling the elastic interaction with the lattice was naturally raised in the literature [7]. Further debate had been started on the role of several inelastic and elastic interactions and their influence on the $J(E)$ curves [8]. Yet, the conclusions drawn seemed to depend most critically on the proposed model, resulting in numerical values for transport parameters differing by as much as one order of magnitude [9]. Finally, even a discussion on the energy functional dependence of the electron–electron inelastic mean free path that would be needed to bring agreement with the experimental data was initiated [4,5]. To cut short a long story, we shall only mention one more controversial point: whether k_{\parallel} is conserved at the MS interface or not [10]. The non-conserving transmission across the interface has been used to explain the similarities between results on (100) and (111) semiconductor surfaces, in spite of the different projected band structure associated to both directions. All these points simply illustrate the necessity to have a theory as accurate as possible to interpret the experiments, trying to avoid discussions originating merely from a weakness of the model. Therefore, we present a program that computes

surface and bulk Green's functions for a given metal characterized by a tight-binding Hamiltonian. While the bulk Green's functions are directly obtained from the Hamiltonian, the surface ones are computed via a decimation technique where layer doubling is practiced until the two surfaces of the slab are effectively decoupled. Applying a Keldysh's formalism that needs these Green's functions as main ingredients, we compute BEEM currents either as a function of the tip position, r_{\parallel} , or the energy, E . A side product of these calculations are the surface band structure and the surface density of states obtained within this tight-binding scheme. These quantities are interesting in many different contexts, and we facilitate their writing through the program, although the limitations of such calculations must always be borne in mind. Finally, we discuss how to include a reconstructed (or relaxed) surface, applying a Dyson-like equation, and taking into account the possible existence of a different periodicity in the direction parallel to the surface.

2. Physics background

E -space Monte Carlo simulation fits to the experiment a number of key parameters determining the injected beam, various mean free paths in the metal (electron–electron, electron–phonon, and electron–defects interaction, etc. . . .). While the main advantage of this approach is its simplicity, a number of problems may appear when a rigorous physical interpretation of the various fitted parameters is needed. Adjusting several such parameters at the same time makes likely the occurrence of cross-correlations among them, leading to major difficulties in explaining physically values that might be associated to inaccuracies in other parts of the model. The aim of BEEM v2.1 is to provide a first-principles, parameter-free formalism for the elastic scenario. Inelastic effects can be incorporated to the theory afterwards through a phenomenological complex optical potential, η , which can be considered as the only free parameter of our approach. This situation is very similar to the one successfully followed since long ago in Low-Energy Electron Diffraction to theoretically interpret the experiments. Having only one free parameter obviously avoids the occurrence of cross-correlation related interpretative problems, and allows computing values for this parameter through other independent *ab-initio* theories (e.g., the Random Phase Approximation can be used for electron–electron (e–e) interaction in the energy range relevant for BEEM). This approach becomes especially appropriate in the low temperature regime, where the main mechanism responsible for attenuation is just e–e interaction. The reason is two-fold: first, e–e interaction takes out a significative amount of kinetic energy from carriers producing a straight attenuation of the BEEM signal; second, experiments performed at high temperature have to deal with the combined effect of at least two scattering channels: e–e and electron–phonon (e–p), resulting in an energy dependence of the imaginary part added to the energy difficult to predict. Therefore, in the regime of high temperature, the Monte Carlo approach may still be the most practical one.

In the original spirit of Slater and Koster [11], the tight-binding Hamiltonian introduced in the formalism is thought as a fast and convenient way to store and retrieve the bands computed previously by any standard *ab-initio* technique. For unreconstructed and unrelaxed surface, it is well known that the solution obtained applying the decimation technique, is accurate and representative of the true self-consistent solution. In fact, self-consistency can be introduced by merely adjusting the diagonal elements of the Hamiltonian. This point becomes more relevant for real surfaces exhibiting pronounced reconstructions or relaxations, where it is important to improve over the original parameter set by imposing self-consistency requirements.

3. Basic Hamiltonian

Our theoretical approach to the BEEM problem starts assuming that the Hamiltonians for the STM-tip (\hat{H}_T) and the metallic sample (\hat{H}_M) can be written and solved for separately. The problem reduces then to deal with the tip-sample system in the presence of some interaction described by a Hamiltonian \hat{H}_I . Furthermore, we assume these three Hamiltonians may be written in a Linear Combination of Atomic Orbitals (LCAO) basis (not explicitly denoted in the following, but implicitly represented by the $\hat{}$ on the operators):

$$\hat{H}_T = \sum \varepsilon_\alpha \hat{n}_\alpha + \sum \hat{T}_{\alpha\beta} \hat{c}_\alpha^\dagger \hat{c}_\beta, \quad (1)$$

$$\hat{H}_M = \sum \varepsilon_i \hat{n}_i + \sum \hat{T}_{ij} \hat{c}_i^\dagger \hat{c}_j, \quad (2)$$

where Greek indices (α, β) have been used to label sites on the tip, while Latin indices (i, j) are referred to sites in the metal. For the practical purpose of improving the computational speed, memory requirements and so on, a convenient approximation to the formal LCAO basis is given by a Slater–Koster parameterization [11]. Useful values for these parameters, covering many elements in the periodic table, can be found in the literature [12–14].

The interaction between the tip and the metal is described by a hopping matrix that couples the atomic orbitals in the site α in the tip with the atomic orbitals in the site i in the metal:

$$\hat{H}_I = \sum \hat{T}_{\alpha i} \hat{c}_\alpha^\dagger c_i. \quad (3)$$

Standard BEEM experiments are performed under conditions where the tip-sample distance is large. Under these conditions, it is known that mainly s-orbitals contribute to the coupling between the tip and the sample. Therefore, to keep the formalism as simple as possible, we shall assume in the following that only two s-orbitals enter into \hat{H}_I , although the generalization to include more orbitals is straightforward.

4. Decimation

To obtain the relevant surface properties needed to solve our problem we use the so-called decimation technique [15,16]. In this technique, the solid is best described as a stack of layers, and good quantum numbers are the energy, E , the momentum parallel to the surface, k_{\parallel} , and an index labeling layers in the perpendicular direction to the surface. The metal Hamiltonian matrix elements can then be written as:

$$\hat{H}_{l,m}(\vec{k}_{\parallel}) = \sum_{i,j=N\text{-neighbours}} \hat{T}_{i,j} e^{i\vec{k}_{\parallel}\vec{r}_N}, \quad (4)$$

where i and j run over different sites in layers l and m , respectively, \hat{T}_{ij} gives the tight-binding parameterized interactions between the different atoms i and j , and \vec{r}_N gives the positions of the different cells in one layer. For each pair of sites i and j , a suitable rotation [17] is made to bring the interaction matrix, \hat{T}_{ij} , along the direction defined by the bond. For the sake of simplicity, the program uses only first-neighbors in the above summation, but it is straightforward to include second neighbors or more if needed.

Given the bulk tight-binding parameters for some material, the Hamiltonian describing a single layer is denoted by $\hat{H}_{1,1}(\vec{k}_{\parallel})$ (the same for all layers of our system), and the interaction between two adjacent layers is denoted by $\hat{V}_{1,2}(\vec{k}_{\parallel})$, which is also the same all through the material. In the program, both are easily obtained as the diagonal and non-diagonal blocks of the Hamiltonian described by Eq. (4) (the size of the blocks depending only on the number of orbitals used in the basis and the number of atoms in the unit cell).

A solution for the semi-infinite material can be easily constructed in the following way: starting from two isolated layers we apply a Dyson-like equation to make a slab composed of two layers. This slab can be used, in the same way, to get a new one composed of four layers, etc. This *layer doubling* procedure is very effective in making a sufficiently thick layer, as its size is proportional to 2^n for the n th pass. When the two surfaces are effectively decoupled (which in practice depends on the value of the imaginary part added to the energy), we have found a solution for the ideal surface problem of the considered material (i.e. the unrelaxed and unreconstructed surface). Specifically, at iteration l , where we want to join together two l -layer slabs to produce a single one of width $2l$, Dyson's equation gives us the following expression:

$$\begin{aligned}
\begin{bmatrix} \hat{g}_{1,1} & \hat{g}_{1,l} & \hat{g}_{1,l+1} & \hat{g}_{1,2l} \\ \hat{g}_{l,1} & \hat{g}_{l,l} & \hat{g}_{l,l+1} & \hat{g}_{l,2l} \\ \hat{g}_{l+1,1} & \hat{g}_{l+1,l} & \hat{g}_{l+1,l+1} & \hat{g}_{l+1,2l} \\ \hat{g}_{2l,1} & \hat{g}_{2l,l} & \hat{g}_{2l,l+1} & \hat{g}_{2l,2l} \end{bmatrix} &= \begin{bmatrix} \hat{g}_{1,1}^0 & \hat{g}_{1,l}^0 & 0 & 0 \\ \hat{g}_{l,1}^0 & \hat{g}_{l,l}^0 & 0 & 0 \\ 0 & 0 & \hat{g}_{l+1,l+1}^0 & \hat{g}_{l+1,2l}^0 \\ 0 & 0 & \hat{g}_{2l,l+1}^0 & \hat{g}_{2l,2l}^0 \end{bmatrix} \\
+ \begin{bmatrix} \hat{g}_{1,1}^0 & \hat{g}_{1,l}^0 & 0 & 0 \\ \hat{g}_{l,1}^0 & \hat{g}_{l,l}^0 & 0 & 0 \\ 0 & 0 & \hat{g}_{l+1,l+1}^0 & \hat{g}_{l+1,2l}^0 \\ 0 & 0 & \hat{g}_{2l,l+1}^0 & \hat{g}_{2l,2l}^0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & V_{1,2} & 0 \\ 0 & V_{2,1} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{g}_{1,1} & \hat{g}_{1,l} & \hat{g}_{1,l+1} & \hat{g}_{1,2l} \\ \hat{g}_{l,1} & \hat{g}_{l,l} & \hat{g}_{l,l+1} & \hat{g}_{l,2l} \\ \hat{g}_{l+1,1} & \hat{g}_{l+1,l} & \hat{g}_{l+1,l+1} & \hat{g}_{l+1,2l} \\ \hat{g}_{2l,1} & \hat{g}_{2l,l} & \hat{g}_{2l,l+1} & \hat{g}_{2l,2l} \end{bmatrix}. & (5)
\end{aligned}$$

In the initial iteration, \hat{g}^0 is the Green's function for the isolated layer:

$$\hat{g}_{1,1}^0(E, \vec{k}_{\parallel}) = \frac{1}{E\hat{T} \pm i\eta - \hat{H}_{1,1}(\vec{k}_{\parallel})} \quad (6)$$

and the \pm signs correspond to the retarded and advanced Green's functions, respectively. In the next iteration, \hat{g}^0 becomes \hat{g} and a new solution for \hat{g} corresponding to the doubled layer is found.

As we show below, for the calculation of BEEM currents we need to compute Green's functions $\hat{g}_{1,m}$ connecting the surface layer, 1, and any other layer in the metal, m . This can be done efficiently by applying m -times to $\hat{g}_{1,1}$ the transfer matrix, \hat{M} , that propagates Green's functions from one layer to the next one:

$$\hat{g}_{1,m} = [\hat{M}]^{m-1} \hat{g}_{1,1} \quad (7)$$

with $\hat{M} = \hat{g}_{1,1} \hat{V}_{2,1}$ [18]. Therefore, the program stores $\hat{g}_{1,1}$ and \hat{M} in files gr11.dat and trans.dat, respectively, and reads them out when needed. The convergence is checked in the program by verifying that any propagation from one surface to the other, $g_{1,n}$, vanishes. As a side-product of the accomplished decimation the bulk, $\frac{1}{\pi} \Im \hat{g}_{l,l}$, and surface density of states, $\frac{1}{\pi} \Im \hat{g}_{1,1}$, are also available (for plane%space = 4 in the main input file these quantities are written to the files rho_e.dat and rho_kprll.dat, as a function of E and \vec{k}_{\parallel} , respectively).

5. Currents within Keldysh's formalism

Currents between the tip and the sample or between two nodes in the metal base (i and j) are computed applying the non-equilibrium Keldysh Green's function formalism [19]:

$$J_{ij} = \frac{e}{\pi \hbar} \int \text{Tr} \{ \hat{T}_{ij} [\hat{G}_{ij}^{+-}(E) - \hat{G}_{ji}^{+-}(E)] \} dE, \quad (8)$$

where the matrices \hat{G}_{ji}^{+-} can be calculated in terms of the retarded and advanced Green's functions of the coupled system, $\hat{G}^{R,A}$, and the Keldysh Green's functions of the non-interacting problem, \hat{g}_{ji}^{+-} , using the following expression:

$$\hat{G}^{+-} = (\hat{I} + \hat{G}^R \hat{T}) \hat{g}^{+-} (\hat{I} + \hat{T} \hat{G}^A). \quad (9)$$

\hat{T} is the hopping matrix coupling the atoms of the tip and the sample.

Moreover, the retarded and advanced Green's functions for the coupled system, $\hat{G}^{R,A}$, are obtained from a Dyson-like equation that uses the Green's functions of the uncoupled parts of the system, $\hat{g}^{R,A}$, and the same hopping matrix \hat{T} :

$$\hat{G}^{R,A} = \hat{g}^{R,A} + \hat{g}^{R,A} \hat{T} \hat{G}^{R,A}. \quad (10)$$

It is then clear that the currents we want to discuss can be written in terms of the advanced and retarded Green's functions of the uncoupled parts of our system, $g^{R,A}$, that are ultimately the important quantities in our problem.

6. STM and BEEM currents

It has been described elsewhere in more detail how the tunneling current can be expressed as [20]:

$$J_T = \frac{4\pi e}{\hbar} \int_{-\infty}^{\infty} \text{Tr}[\widehat{T}_{01}\widehat{\rho}_{11}(E)\widehat{T}_{10}\widehat{\rho}_{00}(E)][f_T(E) - f_S(E)] dE, \quad (11)$$

where 0 refers to the last atom in the tip and 1 to its first neighbor in the metal. The density of states on both atoms, ρ_{00} and ρ_{11} , are computed from the advanced and retarded Green's functions on those sites, and f_T and f_S are the Fermi distribution functions of tip and sample, respectively. This tunneling current is simplified in the program by using only s-orbitals (see routine *surf_tunnel*), but may easily be generalized if other orbital contributions are considered important.

Additionally, currents between any two nodes in the lattice can be computed to obtain the BEEM current. Real space current distributions for a given energy between atoms i and j are given by [21]:

$$J_{ij}(E) = \frac{4e}{\hbar} \Im \text{Tr} \sum_{\alpha\beta i'j'} [\widehat{T}_{ij}\widehat{g}_{j'i'}^R(E)\widehat{T}_{i'\alpha}\widehat{\rho}_{\alpha\beta}(E)\widehat{T}_{\beta j'}\widehat{g}_{j'i}^A(E)] \quad (12)$$

and are computed in the routine *current_real*. The summation runs over tunneling active atoms in the tip (α and β) and the sample (i' and j'). The corresponding Green's functions in real space appearing in the previous equation can be obtained by Fourier transforming the Green's functions in reciprocal space dealt with before:

$$g_{j'i'}^R(E, \vec{r}_{\parallel}) = \sum_{\vec{k}_{\parallel}} w_{\vec{k}_{\parallel}} g_{j'i'}^R(E, \vec{k}_{\parallel}) e^{i\vec{k}_{\parallel}\vec{r}_{\parallel}}, \quad (13)$$

where the summation is performed over a set of special points covering the two-dimensional Brillouin zone. The latter are read from a file supplied by an independent program [22]. The two-dimensional Fourier transform is performed in routine *direct_green*.

Reciprocal space current distributions between layers l and m are computed in routine *current_k*, and given by [21]:

$$J_{l,m}(E, \vec{k}_{\parallel}) = \frac{4e}{\hbar} \Im \text{Tr}[\widehat{T}_{l,m}(\vec{k}_{\parallel})\widehat{g}_{m,1}^R(E, \vec{k}_{\parallel})\widehat{T}_{1,0}(\vec{k}_{\parallel})\widehat{\rho}_{0,0}(E, \vec{k}_{\parallel})\widehat{T}_{0,1}(\vec{k}_{\parallel})\widehat{g}_{1,l}^A(E, \vec{k}_{\parallel})], \quad (14)$$

where 0 stands for the tip layer and 1 defines the outermost surface layer.

7. A reconstructed layer

A reconstructed surface layer can be introduced again making use of a Dyson-like equation. One must bear in mind that self-consistency is not automatically achieved and should be checked separately, e.g., by imposing a condition of local charge neutrality. The latter procedure requires a modification of the bulk diagonal tight-binding parameters, but because electric fields are screened efficiently in metals, only two or three layers need to be considered in most cases.

A technical point to be addressed when including layers displaying a different periodicity from the (1×1) case is the following: up to now, we have assumed a diagonal matrix representation for all relevant operators, allowing to consider \vec{k}_{\parallel} values only inside the 2D Brillouin zone. However, if the reconstructed layer displays a different periodicity (say a $(m \times n)$), the surface Brillouin zone becomes smaller, and if one insists on working in the original (1×1) 2D Brillouin zone, couplings between \vec{k}_{\parallel} values related by \vec{G} vectors corresponding to the new periodicity appear. In other words, the chosen basis is not diagonal any more, and we must consider matrix elements coupling \vec{k}_{\parallel} and $\vec{k}_{\parallel} + \vec{G}$. The simplest solution might seem to switch basis, and work only inside the new 2D Brillouin

zone, but this is inconvenient for the BEEM problem because ultimately we want to compute currents at the metal-semiconductor interface where the (1×1) periodicity prevails. Therefore, we choose to work in the original (1×1) basis, but hopping matrices from a (1×1) layer (say layer 1) to the reconstructed $(m \times n)$ layer (say layer 0) have to be normalized properly making the replacement:

$$\widehat{T}_{0,1}(\vec{k}_{\parallel}) \rightarrow (m \times n)^{-1/2} \widehat{T}_{0,1}(\vec{k}_{\parallel}, \vec{k}_{\parallel} + \vec{G}).$$

We then write explicitly the Dyson equation, $\widehat{G} = \hat{g} + \hat{g} \widehat{V} \widehat{G}$, where \hat{g} corresponds to the semi-infinite system, \widehat{G} is the Green function for the total system (slab plus reconstructed layer), and \widehat{V} is the interaction. In matrix form ($m, n \geq 2$):

$$\begin{bmatrix} \widehat{G}_{00} & \widehat{G}_{01} & \widehat{G}_{0n} \\ \widehat{G}_{10} & \widehat{G}_{11} & \widehat{G}_{1n} \\ \widehat{G}_{m0} & \widehat{G}_{m1} & \widehat{G}_{mn} \end{bmatrix} = \begin{bmatrix} \hat{g}_{00} & 0 & 0 \\ 0 & \hat{g}_{11} & \hat{g}_{1n} \\ 0 & \hat{g}_{m1} & \hat{g}_{mn} \end{bmatrix} \quad (15)$$

$$+ \begin{bmatrix} \hat{g}_{00} & 0 & 0 \\ 0 & \hat{g}_{11} & \hat{g}_{1n} \\ 0 & \hat{g}_{m1} & \hat{g}_{mn} \end{bmatrix} \cdot \begin{bmatrix} 0 & \widehat{V}_{01} & 0 \\ \widehat{V}_{10} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \widehat{G}_{00} & \widehat{G}_{01} & \widehat{G}_{0n} \\ \widehat{G}_{10} & \widehat{G}_{11} & \widehat{G}_{1n} \\ \widehat{G}_{m0} & \widehat{G}_{m1} & \widehat{G}_{mn} \end{bmatrix} \quad (16)$$

from which all the desired Green's functions can be obtained as shown in Appendix I.

8. Description of the program

BEEM v2.1 is a FORTRAN90 program for the computation of real- and reciprocal-space current distributions in BEEM/BEES. Real space currents can be used to understand the spatial resolution of the experiment and possible electronic surface band structure effects on BEEM images, while reciprocal space currents are needed to compute $I(V)$ -spectroscopic BEES curves. This version of the program has been built over an initial simplified version [23].

8.1. Features of BEEM v2.1

Matching the previous physical discussion, the main structure of BEEM v2.1 can be divided into four parts:

- (i) calculation of the LCAO Hamiltonian,
- (ii) calculation of the Green's functions of a semi-infinite crystal,
- (iii) calculation of surface density of states and surface band structure, and
- (iv) calculation of the BEEM currents.

The features and limitations of each of these parts will be treated subsequently:

(i) Calculation of the LCAO Hamiltonian: the metal film can consist of any non-simple crystallographic structure including an unlimited number of atoms of different chemistry per unit-cell. Interaction between all atoms can be accounted for by a set of tight-binding parameters. Optionally a reconstructed surface layer (containing an arbitrary number of atoms and of possibly different lateral periodicity) can be added on top of the film. At the present stage the program is limited to atoms with up to d-orbitals and to nearest-neighbor interaction. The extension of these limits should, however, not pose major difficulties.

(ii) Calculation of the Green's functions of a semi-infinite crystal: In a first step the one-particle Green's functions of one isolated bulk-like *superlayer* and of the optional reconstructed surface layer are obtained from the LCAO Hamiltonians. The Green's function of the bulk-truncated semi-infinite crystal is then derived by means of a 'layer doubling' technique (known in the group renormalization field as "decimation"), whereas the surface layer is finally added by direct application of Dyson's equation. If current distributions in real-space are to be obtained,

the resulting functions of \vec{k}_{\parallel} are 2D Fourier-transformed. The density in k -space used is in principle unlimited, allowing for the suppression of aliasing effects in the discrete Fourier transform to any desired extent. However, keeping the time and hardware requirements of the calculation to a reasonable limit, restricts the maximum number of k -points actually accessible to about 3000 per 2D Brillouin zone. With this, aliasing free distributions within an area of 40×40 atoms have been achieved down to the 30th layer. On the other hand, the only restriction for calculations in reciprocal space is the exponential damping of the current on its way through the crystal, eventually diminishing the numbers below the machine precision level. As a side product, the resulting Green's function of the semi-infinite crystal can be used for surface density of states calculations (iii).

(iv) Calculation of the BEEM currents: The only quantity required apart from the previously determined functions is the hopping matrix between the tip and the metal film. In the program, a simple s-wave tunneling model is supplied, but this routine has been designed to be easily augmentable to more sophisticated approaches. Yet, the present model is a reasonable approximation for the BEEM problem, where the tip-sample distance is usually kept high. Tunneling is allowed from just one atom in the tip to an arbitrary number of inequivalent atoms within the first superlayer of the crystal (may it be bulk-truncated or reconstructed). The BEEM current distributions can be computed in real or reciprocal space in any bulk-like layer in the crystal. Calculation of the BEEM current within the topmost surface layer is not necessary, since it would equal the tunneling current, and is hence not supported by the program. As a simple warning feature, current conservation in real space is always checked for (only achieved for elastic conditions, $\eta \rightarrow 0$, of course).

Finally, we address two memory saving features included into BEEM v2.1: when using a large number of special k -points, as required for high precision and real-space calculations, the amount of intermediary data produced during the program run becomes enormous, since all tight-binding matrices (Hamiltonian of bulk and surface, as well as the coupling between them) are functions of k_{\parallel} . In order to reduce the required memory size of the program to an absolute minimum, all these matrices are stored to internal binary scratch files. With this procedure, the required memory has become almost independent of the number of k -points used. On the other hand, around 300 Mbytes scratch hard disc space are needed for high precision calculations (including around 3000 special k -points per Brillouin zone). As a second feature, the final retarded Green's function matrix after the decimation is dynamically stored, its actual size depending on the given input. In principle, the decimation technique would provide this matrix between all atoms in the surface superlayer and all atoms in the superlayer containing the selected plane. Depending on the number of tunneling-active atoms on the surface and on the range of interaction between layers within the bulk (both numbers being provided by the main input file), only the submatrices corresponding to propagation between these really involved layers are actually kept in the matrix structure. This has made the resulting Green's function matrix a bit complex to read out, but with the achieved major memory reduction, the program now seldom requires more than 32 Mbytes of RAM and could be run on any modest personal computer.

8.2. Program input

Physical units used throughout the code are Å for distances, Å⁻¹ for k wavevectors and eV for energies. The main input file is piped in from stdin and provides the names of further input files: in total BEEM v2.1 requires 3 or 4 input files, depending on whether a reconstructed surface layer is considered or not. The main input gives information on the BEEM calculation itself, one or two input files deal with the tight-binding parameters of bulk and surface and a final input file contains the list of all special k -points used in the calculation. A run-time protocol is punched to stdout verifying all given input. Consistency checks between the different input files are performed wherever possible to assure proper reading of the data.

All demo input files are commented and each subroutine header includes a full list of all variables plus detailed description, so that we shall concentrate here only on those of the input parameters that need further explanation.

Parameters in the main input file:

nat Number of atoms in the bulk unit cell. This unit cell defining a superlayer in the crystal needs to involve two identical cycles of atoms in the z -direction. The program later on determines the interaction between two identical bulk layers by the interaction between these two identical cycles within the input superlayer. E.g., the required number of atoms for a fcc(111) oriented crystal would be 6, i.e. 2 cycles of ABC stacked atoms.

nats (opt.) Number of atoms in surface unit cell. Here only one cycle is necessary.

ndim Dimension of bulk Hamiltonian matrix. This is determined by the number of atoms per unit cell **nat** times their corresponding orbitals. E.g., a Au(111) oriented crystal would have **ndim** = 54, i.e. 6 atoms each with 9 orbitals. This number needs to be given in the input for array dimensioning reasons. A consistency check on dimensions is performed within the program after reading the tight-binding data.

nvec Maximum number of neighbors. Again needed for array dimensioning. In a fcc crystal this number is 12, when only nearest neighbor interaction is considered. Providing a number higher than actually needed, will result in unused reserved memory.

plane% FORTRAN90 structure containing the atomic information on the layer, where the BEEM current is to be calculated. This plane extends laterally from **plane%n1a** to **plane%n1b** in x -direction, from **plane%n2a** to **plane%n2b** in y -direction and is the **plane%layer** layer within the crystal (counting the first bulk-like layer as 1). **plane%space** is a switch parameter determining, which currents are to be calculated: = 1 current in real-space, = 2 current in k -space, = 3 in both spaces, = 4 only DOS calculation. **plane%range** is the maximum distance to a layer, that the chosen layer has interaction with. All Green's functions to all of such layers have to be calculated, increasing the time and size requirement of the program considerably. In a next-neighbor fcc(111) crystal there is only interaction to the next layer, hence = 1. However, in a non-simple structure like CaF₂ next-neighbor interaction to all inequivalent atoms can result in interaction over more than one neighboring layer (e.g., 3 in CaF₂).

mi Imaginary part added to the energy in the one-particle Green's functions (complex optical potential or imaginary part of the self-energy). It is roughly related to an inelastic mean free path λ_{att} via the formula: $\eta = \frac{1}{\lambda_{\text{att}}} \sqrt{2E/m}$, where E is the current (real) energy.

maxiter Number of iterations in the decimation technique. With each iteration the number of layers in the 'semi-infinite' crystal is doubled. Convergence is reached, when the two surfaces of the slab are effectively decoupled. If maxiter has been chosen too small to reach this convergence, a warning message will be added to the runtime protocol. Choosing this value too large, will make the program unnecessarily slow.

switch Switch parameter to store or reread the results of the decimation technique, which is the most time-consuming part of the calculation. If current distributions in various layers of the crystal at the same energy are desired, these intermediary results can be reused. = 1 store results, = 2 read existing results.

surfrec Switch parameter indicating if a reconstructed surface layer shall be added to the semi-infinite bulk-truncated crystal. = 0 no, $\neq 0$ yes. In the latter case, a 4th input file with tight-binding information on that layer needs to be provided.

Parameters in the bulk tight-binding input file:

lattice% This FORTRAN90 structure contains all information on the crystallographic lattice. **lattice%a** is the lattice constant in Å. **lattice%a1/2** are the unit cell vectors parallel to the surface (so their z -component has to be zero). Both need to be given in units of the lattice constant.

atom% Another FORTRAN90 structure for all information on the atomic content in the unit cell. All atomic positions are in units of the lattice constant and the first atom should be placed in the origin. **atom%chem** indicates the chemistry of the particular atom and has to coincide with the number under which the tight-binding parameters for that element are listed later on in the input file.

tpa Array with all tight-binding interactions. In the input file, the first line of the interactions has to state the two numbers of the atomic species for which the interaction information is (e.g., 1 1 for interaction between atoms of type 1 with atoms of type 1). The number of lines following in the input-file then depends on the number of orbitals of both atomic species. There has to be one line for each orbital-orbital interaction: e.g., the interaction table for two p-type atoms has to include 4 lines: s–s, s–p, p–s and p–p. Depending on the orbital momentum, more interaction parameters have to be included in one line (e.g., s–p requires σ and π parameters). The information on this particular atomic combination pair has to end with a line $-1 -1$. Then the next atomic combination pair and so on until all combinations are included. Note, for simple one-atomic structures, there is only one such interaction paragraph.

rmax This provides the maximum radius within which the program searches for neighbors of the corresponding type. I.e. $rmax(1, 2)$ gives this maximum radius for atoms of type 1 and neighbors of type 2. Remember, that for each atom type, only nearest neighbors can be considered under the given program structure. So the user should choose this radius with care. The runtime protocol will provide information on how many neighbors have been found for the given radii.

Parameters in the surface tight-binding input file (opt.):

lattice% If a reconstructed surface layer is added, some further parameters will be added to this structure. **lattice%recx/y** give the lateral periodicity of this layer in Wood notation. **lattice%surfdist** is the vector connecting the origin of the surface unit cell to the origin of the bulk unit cell (this vector has to point into the surface!).

chem The number of different chemical species in the surface layer has to include all species in the bulk-layer that are interacting neighbors. So even if there is only one atomic type in the surface layer, but the bulk layers contain more species, the number given has to match the total number. Consequently, all tight-binding interactions have to be provided for all possible combinations following the above described pattern.

Parameters in the special k -point input file:

k% All information on the special k -points used in the calculation is included in this FORTRAN90 structure. The input file has to have a 4 column format with no comment lines. The first three columns give the $x/y/z$ coordinates of the k -point: note, that the z -component is always zero in this 2D Brillouin zone cut, but exists for historical reasons. The k -point coordinates have to be provided in units of 2π . The program internally rescales these values according to the given lattice constant. The last column gives the weight of the special k -point within the Brillouin zone: points on the borders belong to several zones and have hence less weight in the Fourier transform.

8.3. Routines of BEEM v2.1

In this subsection we give a brief description of the subroutines and functions in the code.

- MAIN beam – initializes the main arrays and subsequently calls all subroutines.
- SUBROUTINE calc_gamma – precalculates the matrix Γ containing all $(\vec{k}_{\parallel}, \vec{k}_{\parallel} + \vec{G})$ couplings in case of a reconstructed surface layer with different periodicity.

- SUBROUTINE couple_k – determines all possible couplings ($\vec{k}_{\parallel}, \vec{k}_{\parallel} + \vec{G}$) among the special k -points due to different surface and bulk periodicities.
- SUBROUTINE current_k – calculates the current distribution in reciprocal space.
- SUBROUTINE current_real – calculates the current distribution in real space.
- SUBROUTINE dec_driver – calling subroutine for the decimation technique.
- SUBROUTINE decimation – performs the decimation iterations in the calculation of the surface Green’s function and transfer matrix of the bulk-truncated semi-infinite crystal.
- SUBROUTINE det_matrixelements – computes the interatomic matrix-elements between all bulk atoms and their neighbors from their respective positions and the tight-binding parameters.
- SUBROUTINE det_surfmatrixelements – computes the interatomic matrix-elements between all surface layer atoms and their neighbors from their respective positions and the tight-binding parameters.
- SUBROUTINE det_neighbors – determines all interacting neighbors within the bulk superlayer.
- SUBROUTINE det_surfneighbours – determines all interacting neighbors within the surface layer.
- SUBROUTINE det_tightham – evaluates the bulk tight-binding Hamiltonian matrix.
- SUBROUTINE det_surftightham – evaluates the surface tight-binding Hamiltonian matrix and the coupling matrix between bulk and surface.
- SUBROUTINE direct_green – performs the 2D Fourier transform to obtain the final Green’s functions in real space.
- SUBROUTINE green_driver – main subroutine for the computation of the required Green’s functions. Calls all connected subroutines.
- SUBROUTINE ham_driver – main subroutine for the computation of the bulk tight-binding Hamiltonian. Calls all connected subroutines.
- SUBROUTINE layer_green – inverts the Hamiltonian matrix to obtain the one-particle Green’s function of one isolated superlayer.
- SUBROUTINE layer_ham – separates the Hamiltonian matrix of the two cycle superlayer into 2 matrices of a one cycle superlayer plus interaction matrix.
- SUBROUTINE organize_green – deduces the size of the retarded Green’s function matrix actually used in the calculation of the BEEM current.
- SUBROUTINE propagate_green – evaluates the Green’s function between the surface and the selected bulk superlayer from the surface Green’s function and the transfer matrix of the decimation technique.
- SUBROUTINE put_results – writes the output to the designated output files.
- SUBROUTINE read_beem – reads the main input file.
- SUBROUTINE read_in – reads the bulk tight-binding input file.
- SUBROUTINE read_k – reads the special k -point input file.
- SUBROUTINE read_surfin – reads the surface tight-binding input file.
- SUBROUTINE recsurface_green – computes the Green’s function of the reconstructed surface layer and couples it to the bulk by direct application of Dyson’s equation.
- SUBROUTINE rot_coord – library package for changing two separate spherical coordinate systems to one relative system. This is the only f77 routine used in this program. It belongs to an older library package of mathematical subroutines and has been left unchanged.
- SUBROUTINE select_subgreen – picks out the final Green’s functions corresponding to the selected layer and its interacting neighbor layers from the big Green’s function matrix of the complete superlayer.
- SUBROUTINE surf_density – determines the surface DOS.
- SUBROUTINE surf_tunel – determines the tip DOS and the hopping matrix between tip and surface.
- SUBROUTINE surfham_driver – main subroutine for the computation of the surface tight-binding Hamiltonian. Calls all connected subroutines.
- MODULE types – declares all user-defined FORTRAN90 structures.

In addition, BEEM v2.1 uses the following external mathematical routines, which are all collected in the subroutine `mat_tools.f90`:

- FUNCTION `brillzone` – determines whether a given k -point belongs to the first Brillouin zone or not.
- SUBROUTINE `calc_unit` – calculates a unit matrix of arbitrary dimension.
- FUNCTION `fermi_distr` – provides the Fermi-distribution.
- SUBROUTINE `mat_conjg` – conjugates a given square matrix.
- SUBROUTINE `mat_inv` – inverses a given square matrix using a further library subroutine `CMXDIV`.
- FUNCTION `trace` – gets the trace of a square matrix.

8.4. Program output

BEEM v2.1 produces two main output files: one for the BEEM current distribution in real space, the other in reciprocal space. Additionally, the tunneling current in reciprocal space is written to a file `Tname`, where ‘name’ is the name of the reciprocal space BEEM current file specified in the input. The latter file allows for easy normalization of the total BEEM current, if desired. Further, the results of the decimation are also stored (by keeping \hat{g}'_{11} and the transfer matrix \hat{M} , in files `gr11.dat` and `trans.dat` in binary format) so they can be reused after the first run is made. The side products of the decimation, the bulk and surface DOS, are finally written to `rho_e.dat` and `rho_kprll.dat` as commented above.

The format of the files containing the BEEM currents, either as a function of atomic position or as a function of \vec{k}_{\parallel} , is straightforward. The real space output file consists of seven columns, the first four of which contain the atomic position (first x/y in cell coordinates, then x/y in real coordinates (in Å), i.e. multiplied with the appropriate unit-vectors and lattice constants). Columns five and six display the total incoming and outgoing current on that atomic site. Taking into account the selected damping through the complex self-energy added to the Green’s functions, these two numbers should match each other, providing a simple way to check the proper working of the program via current conservation. The last column is the BEEM current in arbitrary units – normalization may be achieved using the above mentioned tunneling current distribution. The reciprocal space output file has only three columns: first k_x and k_y (in Å⁻¹), followed by the BEEM current again in arbitrary units.

In addition to these output files, a runtime protocol is punched to `stdout` providing detailed information on the input parameters, program flags and intermediary results.

8.5. Use of the program

In this section, we will give a short guide to the compilation and use of BEEM v2.1.

The program has been written and tested on DEC Alpha machines. A FORTRAN makefile is provided with the code. Due to the high number of matrix inversions and multiplications during the run, use of the default size `REAL*8` for floating point constants is highly recommended. For an easier understanding of the routines, no implicit variable declaration has been used with the exception of the subroutines `rot_coord.f90` and `mat_tools.f90` which both originate from library packages. When separately compiling all subroutines into object codes, remember to always include the module `mod_types.f90` with all user-defined FORTRAN90 structures. Hence, the following compilation has proved successful on the DEC Alpha machines:

```
f90 -c -r8 -u mod_types.f90 'subroutine' for all subroutines separately
f90 -o beam.exe 'list of all object codes' to join all object codes
```

Since BEEM v2.1 computes current distributions in an arbitrary layer in a metal film coupled to a STM tip, its primary purpose was to achieve detailed information on the current impinging on the semiconductor in the usual BEEM setup. The typical input for such calculations would comprise only one selected energy and special k -points within the complete 2D Brillouin zone. A second application of this program has been to obtain $I(V)$ curves for the spectroscopy mode (BEES). In this case, the special k -points would only include the states available

in the semiconductor at the given energy. The resulting current data can subsequently be summed up over these states (eventually considering quantum-mechanical transmission coefficients) giving the current contribution at the selected energy. After various program runs over all desired energies above the Schottky barrier, the summation of all such contributions leads to $I(V)$ curves that can be compared with experimental data.

Acknowledgements

We acknowledge financial support from the contracts PB97-1224 and PB97-28 (Spain); and from SFB292 (Germany). We wish to express our gratitude to all members of the Department of Condensed Matter Theory for all their encouragement and help while this project was being developed. Special thanks are due to Pilar Fernandez for checking thoroughly the program by writing an independent code and making an extensive comparison for Au(111).

Appendix I

In this appendix, expressions for the Green's functions \widehat{G}_{00} , \widehat{G}_{10} , and \widehat{G}_{m0} , are deduced taking into account the general case where a $(m \times n)$ periodicity exists in layer 0, different from the (1×1) assumed from layer 1 down to the interface. Let us consider first \widehat{G}_{00} , and \widehat{G}_{10} :

$$\widehat{G}_{00} = \widehat{g}_{00} + \widehat{g}_{00} \widehat{V}_{01} \widehat{G}_{10}, \quad (17)$$

$$\widehat{G}_{10} = \widehat{g}_{11} \widehat{V}_{10} \widehat{G}_{00}. \quad (18)$$

In the reciprocal \vec{k}_{\parallel} basis, we have for \widehat{G}_{00} :

$$\langle \vec{k}_{\parallel} | \widehat{G}_{00} | \vec{k}_{\parallel} \rangle = \langle \vec{k}_{\parallel} | \widehat{g}_{00} | \vec{k}_{\parallel} \rangle + \kappa \langle \vec{k}_{\parallel} | \widehat{g}_{00} | \vec{k}_{\parallel} \rangle \sum_{\vec{G}} \langle \vec{k}_{\parallel} | \widehat{V}_{01} | \vec{k}_{\parallel} + \vec{G} \rangle \langle \vec{k}_{\parallel} + \vec{G} | \widehat{G}_{10} | \vec{k}_{\parallel} \rangle, \quad (19)$$

where, $\kappa = \sum_{\vec{k}', \vec{k} + \vec{G}} \delta_{\vec{k}, \vec{k} + \vec{G}} = m \times n$. Because the Green's function corresponding to the isolated overlayer, \widehat{g}_{00} , is repeated on the Brillouin zone corresponding to the (1×1) periodicity, it holds the following expression for the non-diagonal terms:

$$\langle \vec{k}_{\parallel} | \widehat{g}_{00} | \vec{k}_{\parallel} + \vec{G} \rangle = \langle \vec{k}_{\parallel} | \widehat{g}_{00} | \vec{k}_{\parallel} \rangle \quad (20)$$

therefore, it is possible to show from Eq. (17):

$$\langle \vec{k}_{\parallel} | \widehat{G}_{00} | \vec{k}_{\parallel} + \vec{G} \rangle = \langle \vec{k}_{\parallel} | \widehat{G}_{00} | \vec{k}_{\parallel} \rangle. \quad (21)$$

Finally, we solve for \widehat{G}_{10} :

$$\begin{aligned} \langle \vec{k}_{\parallel} + \vec{G} | \widehat{G}_{10} | \vec{k}_{\parallel} \rangle &= \langle \vec{k}_{\parallel} + \vec{G} | \widehat{g}_{11} \widehat{V}_{10} \widehat{G}_{00} | \vec{k}_{\parallel} \rangle \\ &= \kappa \langle \vec{k}_{\parallel} + \vec{G} | \widehat{g}_{11} | \vec{k}_{\parallel} + \vec{G} \rangle \langle \vec{k}_{\parallel} + \vec{G} | \widehat{V}_{10} | \vec{k}_{\parallel} \rangle \langle \vec{k}_{\parallel} | \widehat{G}_{00} | \vec{k}_{\parallel} \rangle \end{aligned} \quad (22)$$

and \widehat{G}_{00} is obtained from Eqs. (19) and (21):

$$\langle \vec{k}_{\parallel} | \widehat{G}_{00} | \vec{k}_{\parallel} \rangle = \frac{\langle \vec{k}_{\parallel} | \widehat{g}_{00} | \vec{k}_{\parallel} \rangle}{1 - \kappa^2 \langle \vec{k}_{\parallel} | \widehat{g}_{00} | \vec{k}_{\parallel} \rangle \sum_{\vec{G}} \langle \vec{k}_{\parallel} + \vec{G} | \widehat{V}_{01} | \vec{k}_{\parallel} + \vec{G} \rangle \langle \vec{k}_{\parallel} + \vec{G} | \widehat{g}_{11} | \vec{k}_{\parallel} + \vec{G} \rangle \langle \vec{k}_{\parallel} + \vec{G} | \widehat{V}_{10} | \vec{k}_{\parallel} + \vec{G} \rangle}. \quad (23)$$

Note, that the matrix product in the sum in the denominator is what is precalculated in the routine *calc_gamma* in the program. Finally, a similar procedure allows us to get the Green's function \widehat{G}_{m0} , representing propagation from the bulk-like layer m to the overlayer, 0, inside the (1×1) Brillouin zone:

$$\langle \vec{k}_{\parallel} | \widehat{G}_{m0} | \vec{k}_{\parallel} \rangle = \langle \vec{k}_{\parallel} | \widehat{g}_{m1} | \vec{k}_{\parallel} \rangle \langle \vec{k}_{\parallel} | \widehat{V}_{10} | \vec{k}_{\parallel} \rangle \langle \vec{k}_{\parallel} | \widehat{G}_{00} | \vec{k}_{\parallel} \rangle. \quad (24)$$

References

- [1] L.D. Bell, W.J. Kaiser, Phys. Rev. Lett. 61 (1988) 2368;
W.J. Kaiser, L.D. Bell, Phys. Rev. Lett. 60 (1988) 1406.
- [2] M. Prietsch, Phys. Rep. 253 (1995) 164.
- [3] L.D. Bell, W.J. Kaiser, Annu. Rev. Mater. Sci. 26 (1996) 189.
- [4] L.D. Bell, Phys. Rev. Lett. 77 (1996) 3893.
- [5] K. Reuter, P.L. de Andres, F.J. Garcia-Vidal, F. Flores, U. Hohenester, P. Kocevar, Europhys. Lett. 45 (1999) 181.
- [6] A.M. Milliken, S.J. Manion, W.J. Kaiser, L.D. Bell, M.H. Hecht, Phys. Rev. B 46 (1992) 12826.
- [7] F.J. Garcia-Vidal, P.L. de Andres, F. Flores, Phys. Rev. Lett. 76 (1996) 807.
- [8] A. Bauer, M.T. Cuberes, M. Prietsch, G. Kaindl, Phys. Rev. Lett. 71 (1993) 149.
- [9] C. Manke, Y. Bodschwinn, M. Schulz, Appl. Surf. Sci. 117 (1997) 321.
- [10] R. Ludeke, Phys. Rev. Lett. 70 (1993) 214.
- [11] J.C. Slater, G.F. Koster, Phys. Rev. 94 (1954) 1498.
- [12] D.A. Papaconstantopoulos, Handbook of the Band Structure of Elemental Solids (Plenum, New York, 1986).
- [13] P. Vogl, P. Hjalmanson, J.D. Dow, J. Phys. Chem. Solids 44 (1983) 365.
- [14] S. Sanguinetti, C. Calegari, V.R. Velasco, G. Benedek, F. Tavazza, L. Miglio, Phys. Rev. B 54 (1996) 9196.
- [15] F. Guinea, C. Tejedor, F. Flores, E. Louis, Phys. Rev. B 28 (1983) 4397.
- [16] M. Lannoo, P. Friedel, Atomic and Electronic Structure of Surfaces (Springer, Berlin, 1991).
- [17] J.R. Alvarez Collado, J. Fernandez Rico, R. Lopez, M. Paniagua, G. Ramirez, Comp. Phys. Commun. 52 (1989) 323.
- [18] P.L. de Andres, F.J. Garcia-Vidal, D. Sestovic, F. Flores, Phys. Scripta T66 (1996) 277.
- [19] L.V. Keldysh, Zh. Eksp. Teor. Phys. 47 (1964) 1515; Sov. Phys. JETP 20 (1965) 1018.
- [20] A. Martin-Rodero, F. Flores, N.H. March, Phys. Rev. B 38 (1988) 10047;
F. Flores, P.L. de Andres, F.J. Garcia-Vidal, L. Jurczyszyn, N. Mingo, R. Perez, Prog. Surf. Sci. 48 (1995) 27.
- [21] K. Reuter, P.L. de Andres, F.J. Garcia-Vidal, D. Sestovic, F. Flores, K. Heinz, Phys. Rev. B 58 (1998) 14036.
- [22] R. Ramirez, M.C. Böhm, Int. J. Quantum Chem. 30 (1986) 391 (the special points program is available on request from Dr. R. Ramirez, Instituto de Ciencia de Materiales de Madrid (CSIC), Cantoblanco, E-28049 Madrid, Spain; Email: rafa@lot.icmm.csic.es).
- [23] D. Sestovic, P.L. de Andres, F.J. Garcia-Vidal, F. Flores, FORTRAN code BEEM V1.1, unpublished.

Test Run Input and Output

Input file hamAu111.dat

```

----- Input File for Tight-Binding Calculation -----
----- Data is for Au(111) Layer -----
  2.88379          ! Lattice parameter in A          F10.5
  1.00000  0.00000  0.00000  ! Unit_vector_1 (in units of a)      3F10.5
  0.50000  0.86602  0.00000  ! Unit_vector_2                      3F10.5
  6          ! No. of atoms in unit-cell        I3
  1          ! No. of diff. chemical species      I3
  2  2  2  2  2  2          ! lmax for each atom in unit-cell    XI3
      x/y/z in units of a  chem  es  ep  ed  3F10.5, I3, 3F10.5
  0.00000  0.00000  0.00000  1  0.32911  10.07119  -3.82119
  0.50000  0.28868  0.81650  1  0.32911  10.07119  -3.82119
  1.00000  0.57735  1.63299  1  0.32911  10.07119  -3.82119
  0.00000  0.00000  2.44950  1  0.32911  10.07119  -3.82119
  0.50000  0.28868  3.26600  1  0.32911  10.07119  -3.82119
  1.00000  0.57735  4.08250  1  0.32911  10.07119  -3.82119
Parametrized interactions between atoms in lattice (all in eV):
  1  1          ! Data between type i&j          2I3
  0  0  -0.90848  ! sssigma                        2I3, 3F10.5
  0  1  1.32262  ! spsigma
  0  2  -0.64219  ! sdsigma
  1  0  -1.32262  ! pssigma

```

```

1 1 2.42978 -0.22372 ! ppsigma,pppi
1 2 -0.87026 0.25786 ! pdsigma,pdpi
2 0 -0.64219 ! dssigma
2 1 0.87026 -0.25786 ! dpsigma,dppi
2 2 -0.67606 0.35686 -0.06215 ! ddsigma,ddpi,dddelta
-1 -1 ! End of data i&j 2I3
Maximum interaction radius between atoms i & j in lattice (in A):
1 1 2.88379 ! i,j,rmax(i,j) 2I3,F10.5
-----

```

Input file Au111.inp

```

----- Input File for BEEM-Current Calculation -----
----- Data is for Au(111) Layer -----
1. Data for calculating the H-matrix in ham_driver
hamAu111.dat ! Name of ham_driver input-file
ktotal111_1333.dat ! Name of special $k$-points input-file
outAu111real.dat ! Name of output-file (real-space current)
outAu111k.dat ! Name of output-file ($k$-space current)
6 ! No. of atoms in u-cell (even cycles,>6)I4
54 ! Dimension of H-matrix (atoms*orb) I4
1333 ! No. of special $k$-points (>convert) I4
12 ! No. of max. allowed neighbours (arrays)I4
2. Data on plane, where current is to be calculated
-20 ! n1a Plane goes in x-direction I4
20 ! n1b from n1a to n1b (index integer) I4
-20 ! n2a Plane goes in y-direction I4
20 ! n2b from n2a to n2b (index integer) I4
10 ! Layer-nr. of plane (surface=1) I4
1 ! Max. plane-distance of neighbours I4
3 ! Current in real(1), k(2) or both(3) I4
3. Data on tip and tunneling-process (currently T1-position)
9 ! No. of tip-orbitals (max=9) I4
0.000 ! Tip fermi-level in eV (sample=0.00) F7.3
1.000 ! STM bias in eV F7.3
3.000 ! Pot. barrier for tunneling in eV F7.3
0.000000 0.000000 -5.00000 ! Tip position in A (xyz) 3F7.3
5.100 ! Max. tunneling-dist. in A F7.3
1 ! No. of max. allowed t-atoms (arrays) I4
1 ! No. of atom-types/layers involved I4
0.000 ! Temperature in K F7.3
4. Data on decimation technique
0.1 ! Mean free path for G-functions F8.4
20 ! No. of iterations in decimation I4
1 ! Store(=1),reread(=2) decimation resultsI4
7.121 ! 2D-BZ area of Au in inv A^2(not used)F7.3
5. Data on energy-integral interval
1.000 ! Min. energy in eV F7.3
1.000 ! Max. energy in eV F7.3
0.500 ! Increment in loop in eV F7.3
6. Data on evtl. surface reconstruction
0 ! Consider (<>0) a possible surf. rec. I4
-----

```

Output file rho_e.dat

# Energy in eV	nats	x	DOS(surf),		
1.000	0.29887		0.27635	0.27411	0.00000
	0.00000		0.00000	0.85345	0.14944
	0.13818		0.13705	0.00000	0.00000
	0.00000		0.42672		

First 10 lines of output file Au111k.out

#	k-space coordinates		j_beem
	0.0000	0.0000	0.310266E-04
	0.0605	0.0349	0.263739E-04
	0.1210	0.0699	0.191912E-04
	0.1816	0.1048	0.174468E-04
	0.2421	0.1398	0.394784E-04
	0.3026	0.1747	0.307833E-03
	0.3631	0.2097	0.426448E-03
	0.4237	0.2446	0.488402E-03
	0.4842	0.2795	0.536007E-03
	0.5447	0.3145	0.581037E-03

etc

First 10 lines of output file Au111r.out

# Cell-index	Cell-coordinates		j_in	j_out	j_beem
-20 -20	-86.5137 -49.9484		0.54041E+08	-0.61728E+06	0.38578E+08
-20 -19	-85.0718 -47.4510		0.97571E+08	-0.46045E+07	0.79736E+08
-20 -18	-83.6299 -44.9536		0.11163E+09	-0.24740E+08	0.78638E+08
-20 -17	-82.1880 -42.4561		0.10766E+09	-0.81008E+07	0.93664E+08
-20 -16	-80.7461 -39.9587		0.76140E+08	-0.63771E+07	0.59561E+08
-20 -15	-79.3042 -37.4613		0.62116E+08	-0.28908E+08	0.29116E+08
-20 -14	-77.8623 -34.9639		0.18594E+08	-0.66009E+07	0.74139E+07
-20 -13	-76.4204 -32.4665		0.51066E+07	-0.76545E+06	0.18912E+07
-20 -12	-74.9785 -29.9690		0.11950E+08	-0.40356E+07	0.72009E+07
-20 -11	-73.5366 -27.4716		0.13827E+08	-0.75098E+07	0.54542E+07

etc

First 10 lines of output file TAU111k.out

#	k-space coordinates		j_tunnel
	0.0000	0.0000	0.242703E-02
	0.0605	0.0349	0.181881E-02
	0.1210	0.0699	0.889429E-03
	0.1816	0.1048	0.382560E-03
	0.2421	0.1398	0.211416E-03
	0.3026	0.1747	0.498390E-03
	0.3631	0.2097	0.667144E-03
	0.4237	0.2446	0.780801E-03
	0.4842	0.2795	0.884003E-03
	0.5447	0.3145	0.988953E-03

etc