Proceedings Paper:

Valavanis, A, Cooper, JD, Grier, A et al. (1 more author) Quantum wells, wires and dots (QWWAD): development of an open-source simulation suite for semiconductor nanostructures. In: UNSPECIFIED UK Semiconductors 2014, 09-10 Jul 2014, Sheffield, UK. . (Unpublished)

Quantum Wells, Wires and Dots (QWWAD): Development of an open-source simulation suite for semiconductor nanostructures

A. Valavanis[1], J. D. Cooper[1], A. Grier[1] and P. Harrison[2]

[1]Institute of Microwaves and Photonics, School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, U.K.
[2]Materials and Engineering Research Institute, Sheffield Hallam University, S1 1WB, UK.

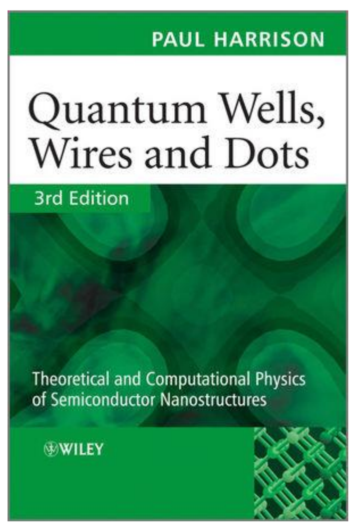
Sheffield Hallam University

UNIVERSITY OF LEEDS

1. Introduction

A number of robust software tools are available for simulating semiconductor nanostructures, including purpose-made products such as Nextnano [1] and generic finite-element solvers such as COMSOL Multiphysics [2]. Although highly regarded, the majority of available software is supplied under a proprietary license, meaning that its source code cannot be studied, modified or redistributed freely by its users. As such, there is currently a lack of free software for students wishing to learn the mathematical and computational techniques that underpin modern nanoscale semiconductor physics.

We present a non-commercial, free-and-open-source project, *Quantum Wells, Wires and Dots* (QWWAD) [3], which is released under the GNU General Public License 3.0 [4] and is currently open for beta testing.



This free software accompanies the 4th edition of "Quantum Wells, Wires and Dots" by P. Harrison & A. Valavanis, which will be published in 2015 by J. Wiley and Sons, Chichester.

2. Functionality

QWWAD currently provides code to solve a wide range of physical models, namely:

- **Analytical Schrödinger solvers** (e.g., quantum wells, superlattices)
- **Impurities, excitons and diffuse systems**
- **Numerical Schrödinger and Poisson solvers**
- **Carrier distributions**
- **Tunnelling systems** (single barriers, RTDs)
- **Pseudopotential calculations**
- **Scattering models** (carrier–phonon, carrier–carrier)
- **Semi-analytical models of quantum wires & dots**

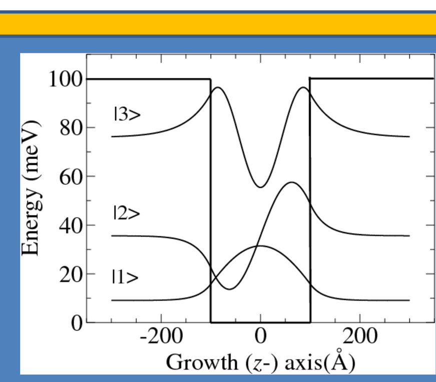
3. Software architecture

QWWAD aims to ensure high software quality by applying robust software engineering principles from the outset. **Automated unit testing and manual functional testing** procedures will ultimately permeate the code at every level, ensuring its reliability and accuracy.

We currently provide three types of software interface, which are each ideal for a particular class of user.

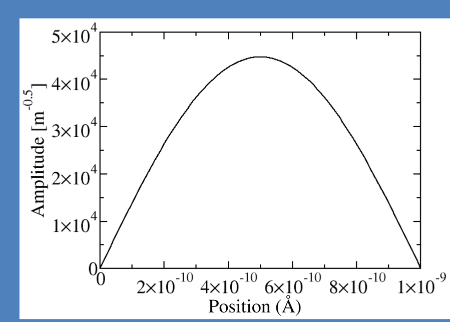
Example scripts: Simple "one-line" commands generate data for hundreds of example simulations. No knowledge of programming or scripting techniques is needed to gain useful results instantly.

```
$ finite-well-wavefunctions.sh
```



Core programs: A set of flexible C++ programs provide the "building blocks" for customised simulation scripts. This example computes the ground state of a 10-Å-wide GaAs infinitely deep quantum well:

```
$ efiw --width 10 --mass 0.067 --states 1
```



Application programmers interface (API): The underlying functionality of QWWAD is available for use in custom C++ programs. To solve the above infinite-well example:

```
SchroedingerSolverInfWell solver(mass, length, n_spatial_points);
solutions = solver.get_solutions();
```

Reliability: Automated builds and tests run on *Launchpad* after any change to code

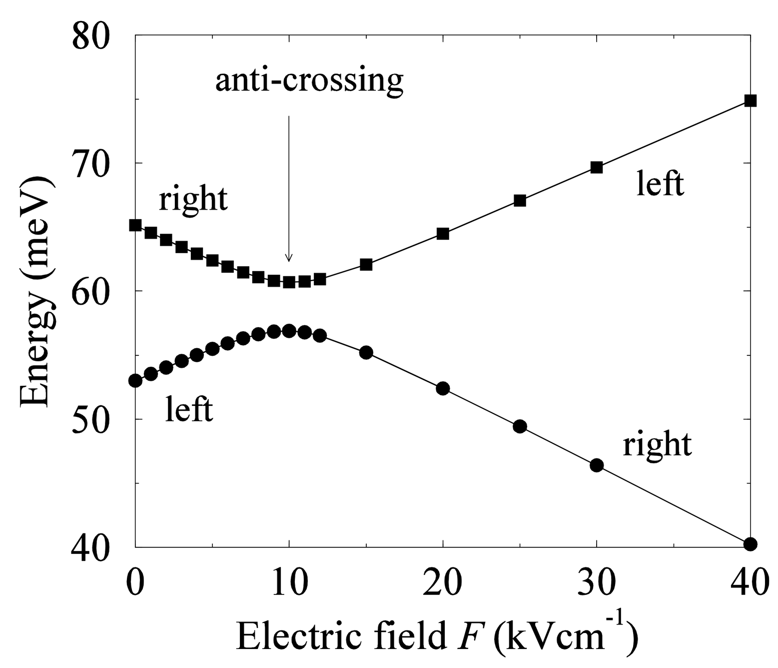
4. Example: Electric-field induced anticrossings

This example demonstrates excerpts of a QWWAD script for computing the anticrossing between conduction band states in a double quantum well as a function of the external electric field:

```
# Tabulate double well: width [angstrom], alloy, doping
echo 200 0.2 0.0 >  s.r
echo 60   0.0 0.0 >> s.r
echo 60   0.2 0.0 >> s.r
echo 50   0.0 0.0 >> s.r
echo 200 0.2 0.0 >> s.r

find_heterostructure # Generate sample mesh
efxv                 # Generate table of potential data

# Loop over field [0 - 40 kV/cm]
for F in `seq 0 40`; do
  find_poisson_potential --uncharged --field $F
  ...
  efss # Solve Schroedinger equation
done
```



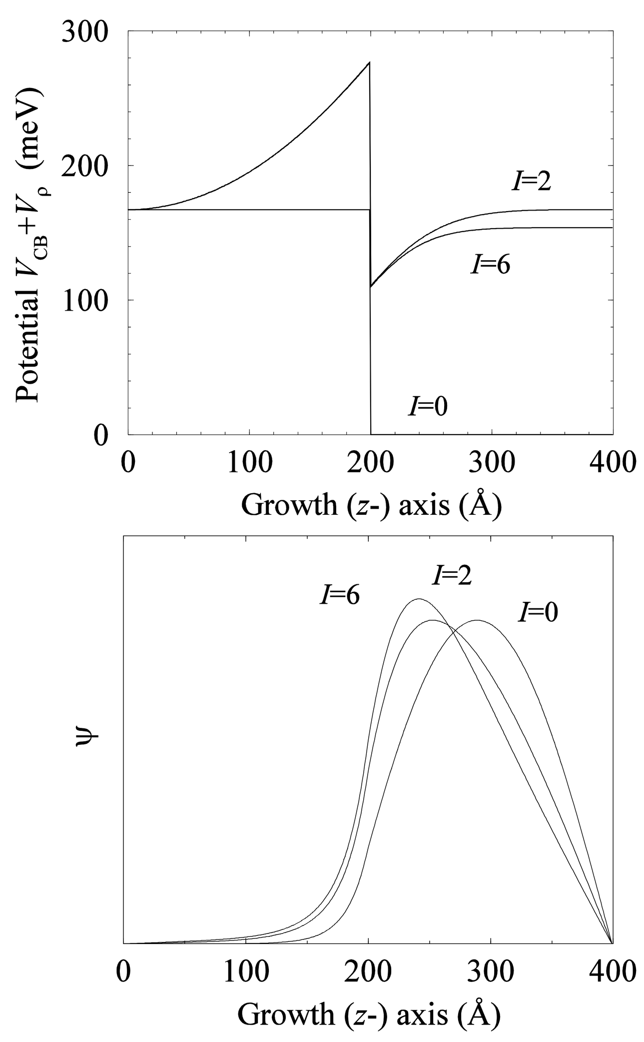
5. Example: Self-consistent solution for a HEMT

This example demonstrates excerpts of a QWWAD script for computing a self-consistent Poisson–Schrödinger solution for a high-electron mobility transistor (HEMT).

```
# Tabulate heterostructure: width [angstrom], alloy, doping
echo 200 0.2 2e17 > s.r
echo 200 0.0 0.0 >> s.r

find_heterostructure # Generate sample mesh
efxv                 # Generate table of potential data

# Perform 8 iterations of Poisson-Schrödinger solution
for I in `seq 0 8`; do
 efss # Solve Schroedinger equation
 ...
 find_poisson_potential
done
```



6. Conclusions

We have presented the motivation, architecture and user examples of the free-and-open-source QWWAD simulation suite. This software is freely available, and aims to serve as both a useful educational resource and a reliable set of research tools.

Acknowledgements

We thank our publishers, J. Wiley and Sons, Chichester, and acknowledge the generous time commitment of the QWWAD beta testers team.

References

[1] Nextnano, http://www.nextnano.de
[2] COMSOL Multiphysics, http://www.uk.comsol.com
[3] Quantum wells, wires & dots, http://launchpad.net/qwwad
[4] GNU General Public License 3.0, Free Software Foundation (2007) https://www.gnu.org/copyleft/gpl.html