SAND2017-0531C

Cooking with QCAD: Salt, Pepper, and Plans

Erik Nielsen

Sandia National Laboratories

Albany Users Meeting, Jan 2017

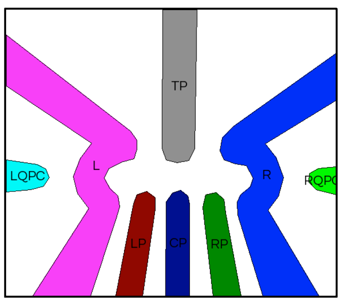
Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

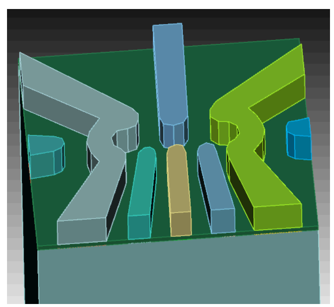
Overview of this talk

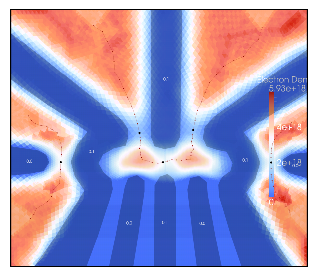
- QCAD is an “old” Albany application, not currently under much active development.
- But it continues to be well utilitzed:
  - Many (~30) Theorists and Experimentalists here at Sandia
  - Deployed on a server at Princeton University
  - Other users we don’t know about?
  - 3 dedicated machines running QCAD.
- My goals in this talk:
  - Review briefly describe what QCAD is and does.
  - Look in more detail at some of the aspects of QCAD that have made is successfully accessible to so many people (and prolong it’s use) – the “salt & pepper”.
  - Pontificate on future directions: things I’d like to add to QCAD and for which I’m not sure how difficult this would be.

What exactly is QCAD?

- QCAD = "Quantum Computer Aided Design"
- Primary Goals:
  - **Integration** (components work together seamlessly)
  - **Flexibility** (different problems; different devices & materials)
  - **High Throughput** (can explore parameter space quickly)
- A tool which solves **simple** physics on complex geometries:
  - **Poisson's equation** (like heat diffusion) $\nabla \cdot \varepsilon \nabla \phi = \rho(\phi)$
  - Single and many-electron **Schrodinger equations** $H\psi_i = E_i \psi_i$
- Solve discretized equations on a finite element mesh:







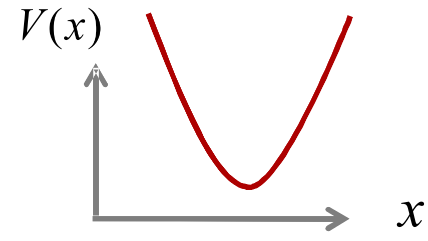
Electron Density (**red** = higher)

- Ideally, experimentalists measuring these devices would be able to run the code.

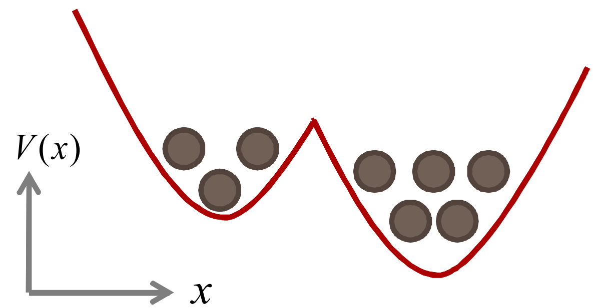
"QCAD simulation and optimization of semiconductor quantum dots," JAP 114, 164302 (2013)

Primary application:
Devices used for **quantum computing**

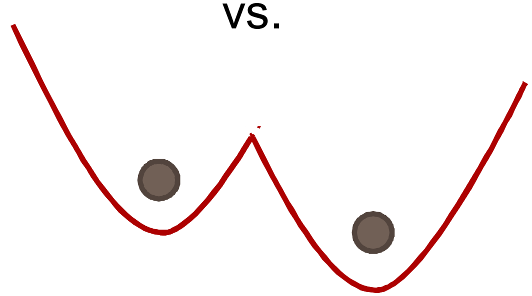
- **Quantum dot**: a potential energy “pit” that confines electrons to a point in space



- **Double quantum dot**: two of them



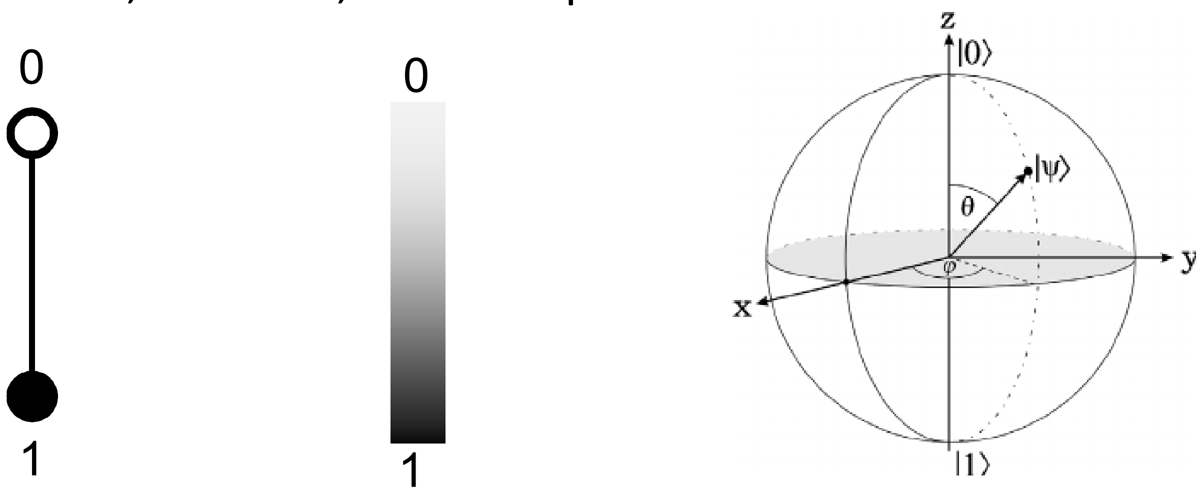
vs.



- **Useful** for quantum computing **if** we have just a **few electrons in each dot** and can move them back & forth.
  - # of e^- depends on “depth” and “size” of dots

Quantum bits & computing

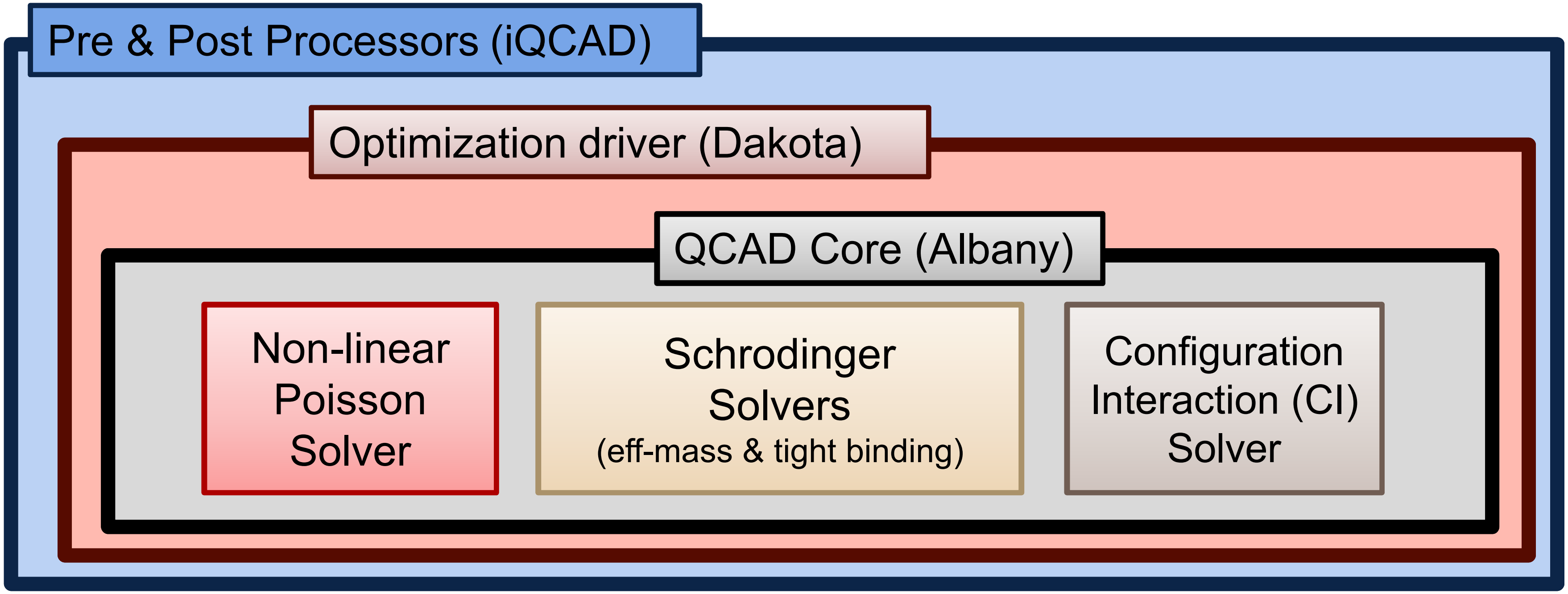
- Quantum computing promises algorithmic speedups.
- In one (popular) model of quantum computing, the quantum computer is composed of ”quantum bits”, or “qubits” which are manipulated by logic gates.
- By definition, a qubit is a 2-level quantum system that can be initialized, read out, and manipulated.



bit   probabilistic bit, “pbit”   qubit

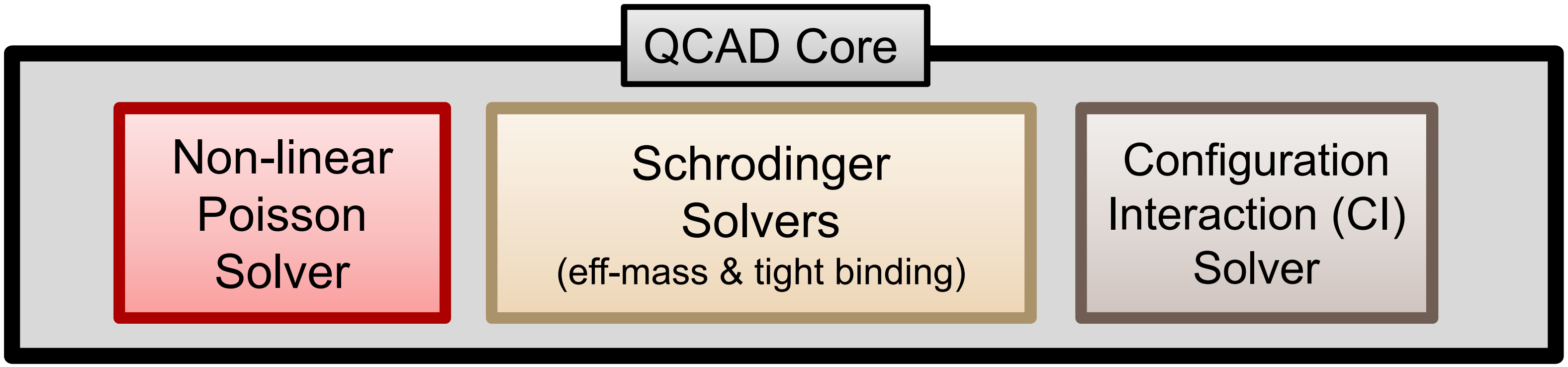
- In actuality, you never have just 2 quantum levels, so consider a low-energy or isolated 2-dimensional subspace.

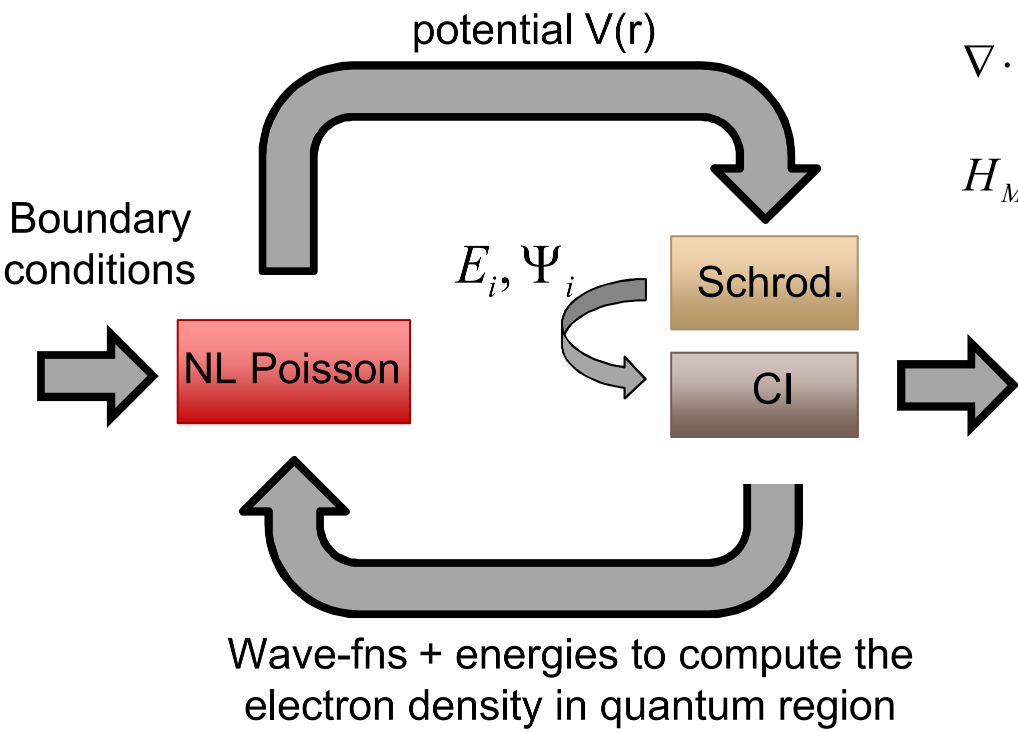
QCAD Implementation Structure



- Different types of problems (Poisson-only vs Schrodinger-Poisson vs Shrodinger-Poisson-CI) require more or less interaction among multiple Albany Application objects.
- For example:

Example: Coupled Nonlinear Poisson & Configuration Interaction





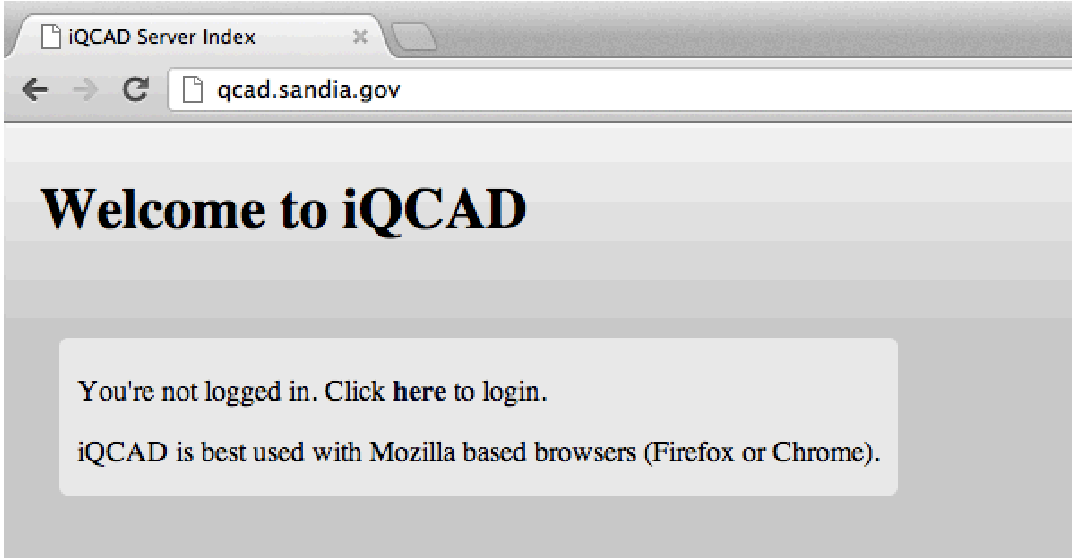
$$\nabla \cdot \varepsilon \nabla \phi = \rho(\phi, E_i, \Psi_i)$$

$$H_{MB}\Psi_i = \left[\sum_k \left(T_k + V_k(\phi) \right) + \sum_{k<l} V_{kl}(\phi) \right] \psi_i = E_i \Psi_i$$

- Potential
- electron density,
- many-electron wavefunctions (coherence measures)

A key to QCAD's success: **interfaces**

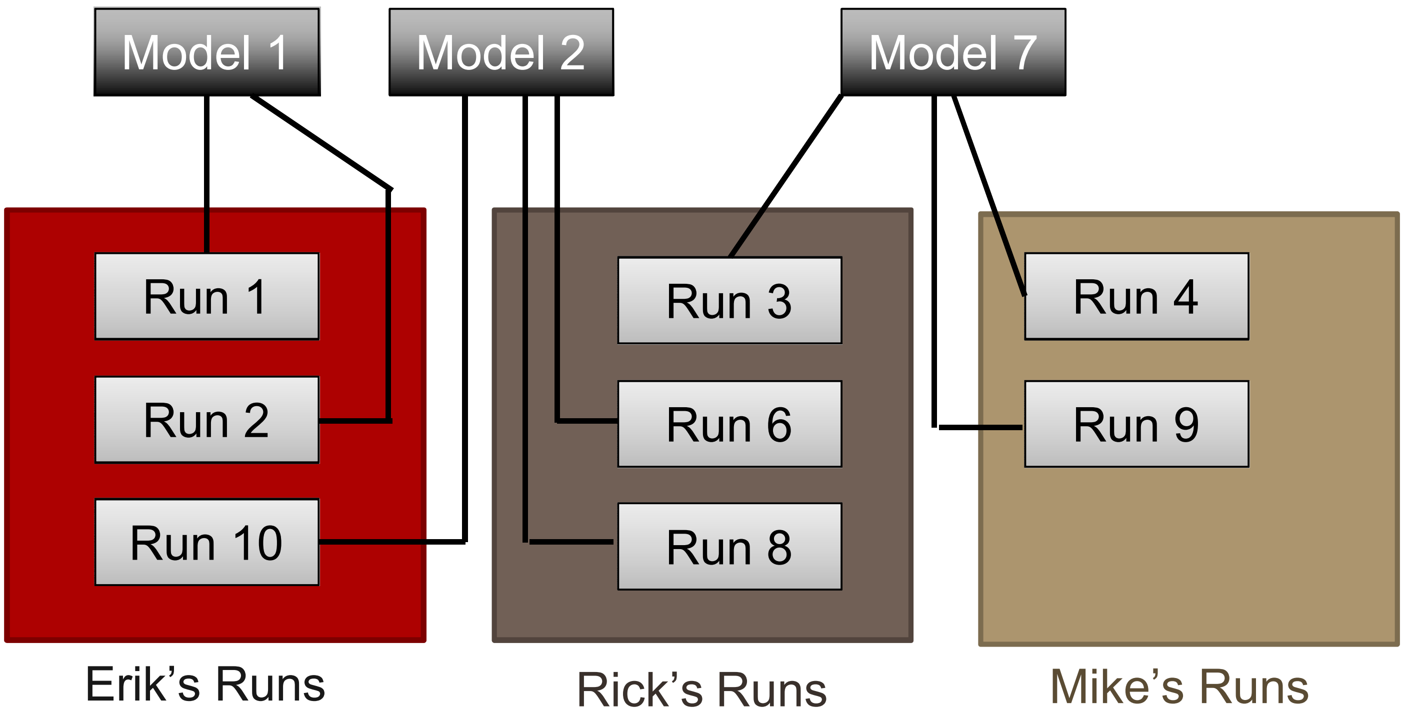
- We'd like to think of QCAD as a successful "product" because people use it.
- A lot of work has gone into creating easily usable interfaces for running QCAD – "make it more like COMSOL"
  - Single XML file & format control all types of QCAD solutions.
  - "iQCAD" web interface essentially adds a GUI to QCAD.



Web interface ties together:
- Device design (EaselJS)
- Meshing (Cubit)
- Execution (Albany)
- Imaging (Paraview)

iQCAD object basics

- Each object has a globally unique ID
- Each object has an owner.
- Objects reference one another. For example, each Run object is associated with a Model object.
- An exemplary diagram:



The iQCAD Home page

Get back to this page

Your user name

Demo's iQCAD II home page

My Home | Local Status

Logged in as **Demo (logout)**

Run Objects

Create a run

Run

Create new run

Heading indicates this section is for "run"-type objects

Name : [ ] Category : [ ] Filter | Set category | Run | Delete

Operations on checked runs

	Name	Category	Model	Date Created	Date Modified	Date Last Run
☐	**Test simple run**	tests	**Simple JSON model**	May 05 2015 02:35 PM	May 05 2015 02:35 PM	May 05 2015 02:35 PM
☐	**Test model with overlap layers**	tests	**Model with overlap layers**	May 05 2015 02:08 PM	May 05 2015 02:08 PM	May 05 2015 02:08 PM
☐	**Test vector sweep**	tests	**Simple JSON model**	May 05 2015 12:54 PM	May 05 2015 12:54 PM	May 05 2015 12:54 PM
☐	**Test device builder model**	tests	**Device from builder**	May 05 2		15 12:53 PM
☐	**Override model materials**		**Simple Python Model**	May 05 2		15 12:52 PM
☐	**Override model parameters**		**Simple Python Model**	May 05 2		15 12:51 PM
☐	**Parameter sweep with fixed geometry**		**Simple Python Model**	May 05 2		15 12:50 PM
☐	**Parameter sweep that varies geometry**		**Simple Python Model**	May 05 2		15 12:49 PM
☐	**Integrate a mesh region**		**Device with Region**	May 05 2015 12:47 PM	May 05 2015 12:47 PM	May 05 2015 12:47 PM

A list/table of all **your** runs. Click on one to go to that run's page.

Model Objects

Create a model

Model

Create new model

Name : [ ] Category : [ ] Filter | Set category | Delete

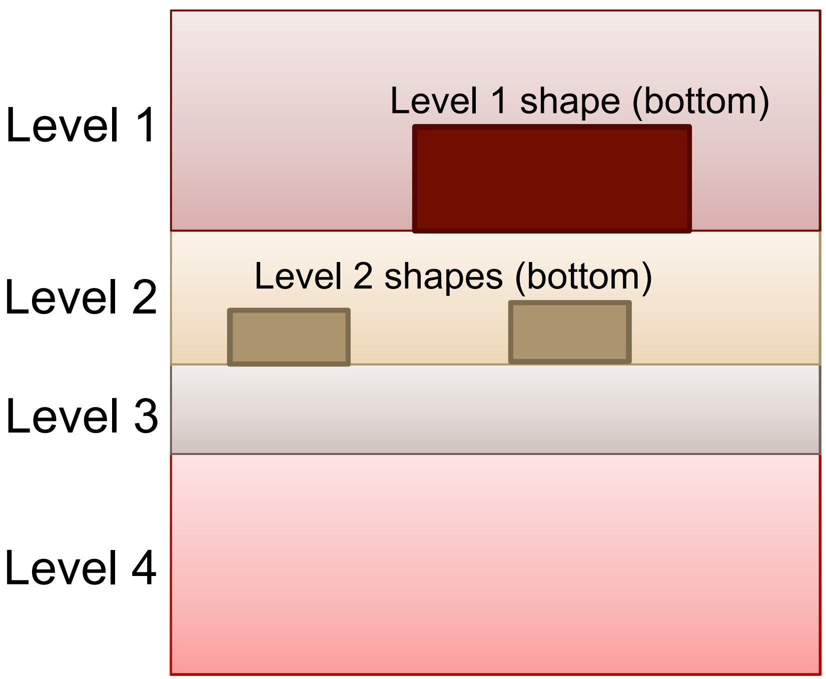
	Name	Category	Description	Date Created	Date Modified
☐	**Simple JSON model**		square MOS A-gate	May 05 2015 12:17 PM	May 05 2015 12:17 PM
☐	**Simple Python Model**		A-gate parameter		015 12:18 PM
☐	**Device from builder**				015 12:21 PM
☐	**Device using flood fill**				015 12:23 PM
☐	**Device with Region**				015 12:45 PM
☐	**Model with overlap layers**				015 02:06 PM

A list/table of all **your** models. Click on one to go to that model's page.

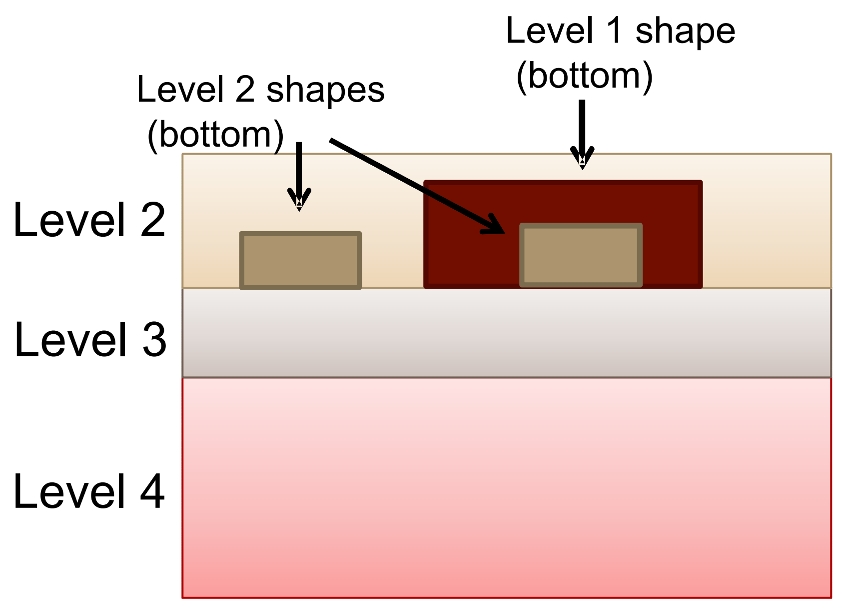
Geometry Building: Levels & Shapes

- Most geometries of interest have a layered structure, which lends itself to being designed in 2D.
- iQCAD contains a "layered 2D" javascript-powered geometry builder.
- An example geometry:

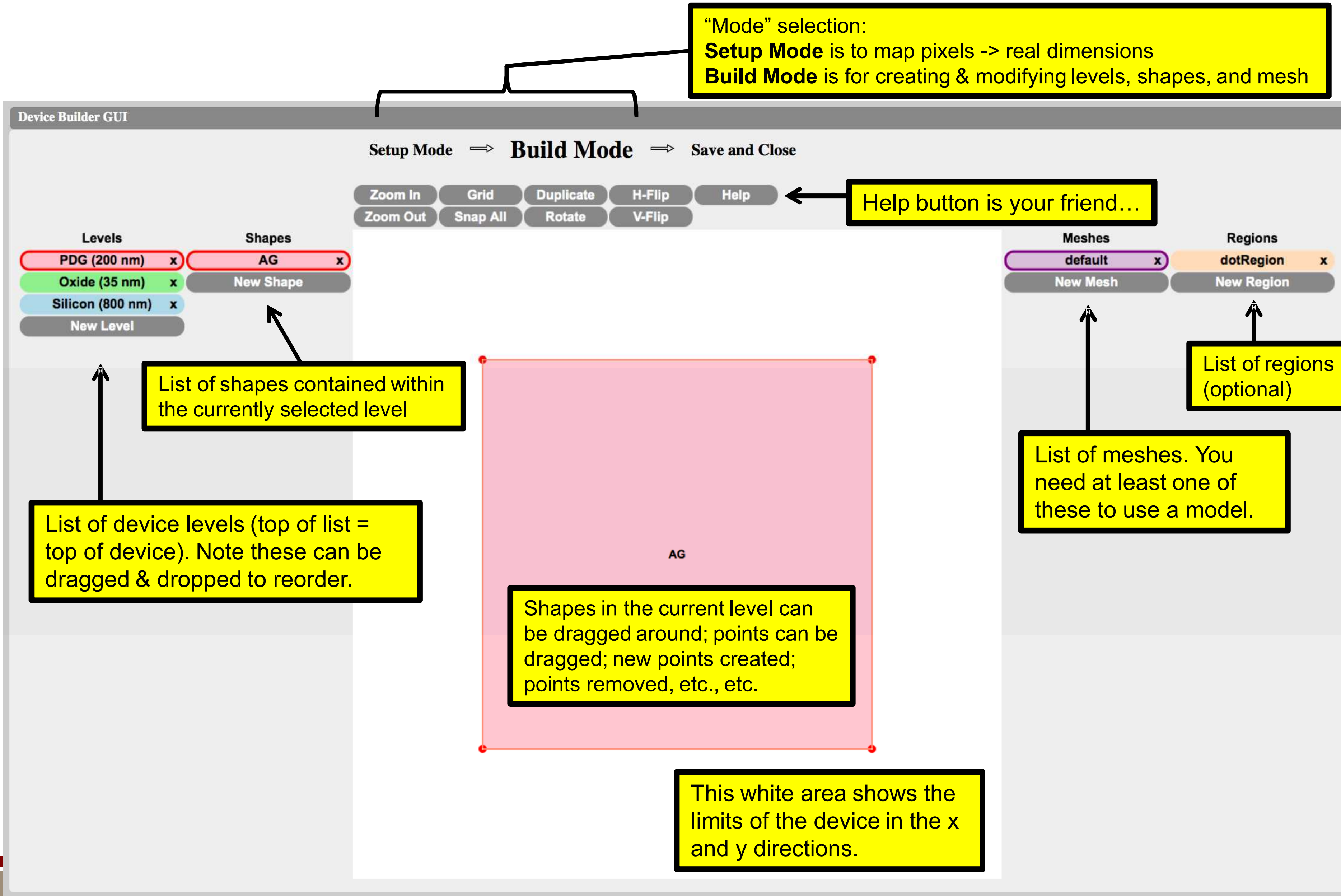
With no "overlay" layers



If Level 1 were an overlay layer:



Device Builder GUI – making models with ease…



Meshing output

Builder-generated geometries can be automatically meshed using Cubit.

Meshing Output

Created Meshes are in sync: **create a new run for this model**

What you might want to do next...

Exodus files can be viewed using Paraview (www.paraview.org)

Mesh Information

Number of dimensions = 3
Number of element blocks = 4
Number of sidesets = 0
Number of nodesets = 1
Number of bc sets = 1
Number of elements = 142182
Number of nodes = 26181

The "size" of the mesh – i.e. how many nodes (points) are used in the discretization. This is roughly proportional to the run time required.

Files

Output file	**mc.out**
Cubit output file	**mc.claro**
Cubit journal (default)	**default.jou**
Exodus mesh (default)	**default.exo**

ubit Journal Information

Element Block Names (green = found in all meshes)

eb_ag	pdg	oxide	silicon

Nodeset Names (green = found in all meshes)

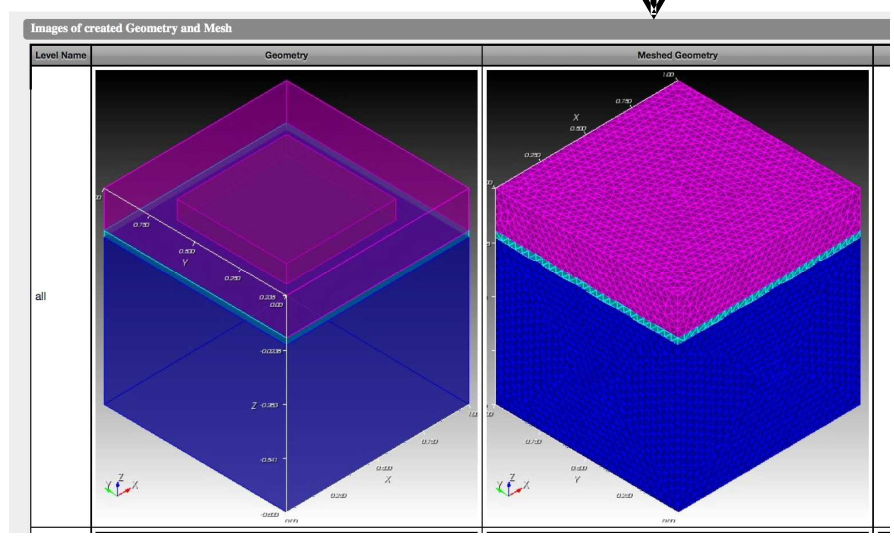
ag

Sideset Names (green = found in all meshes)

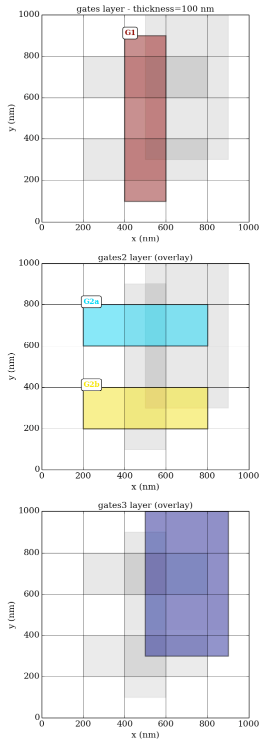
Images of created Geometry and Mesh

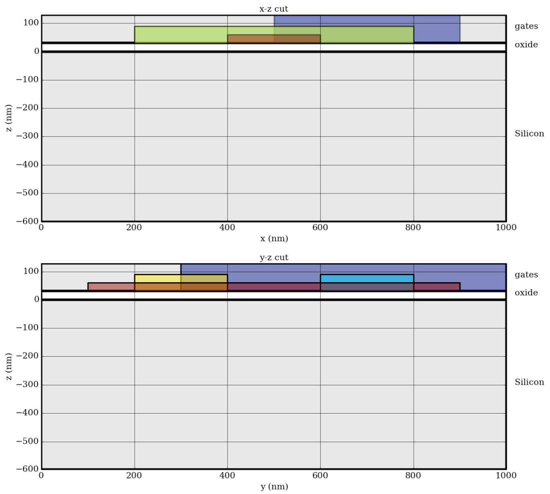
Click this dark gray bar to see images

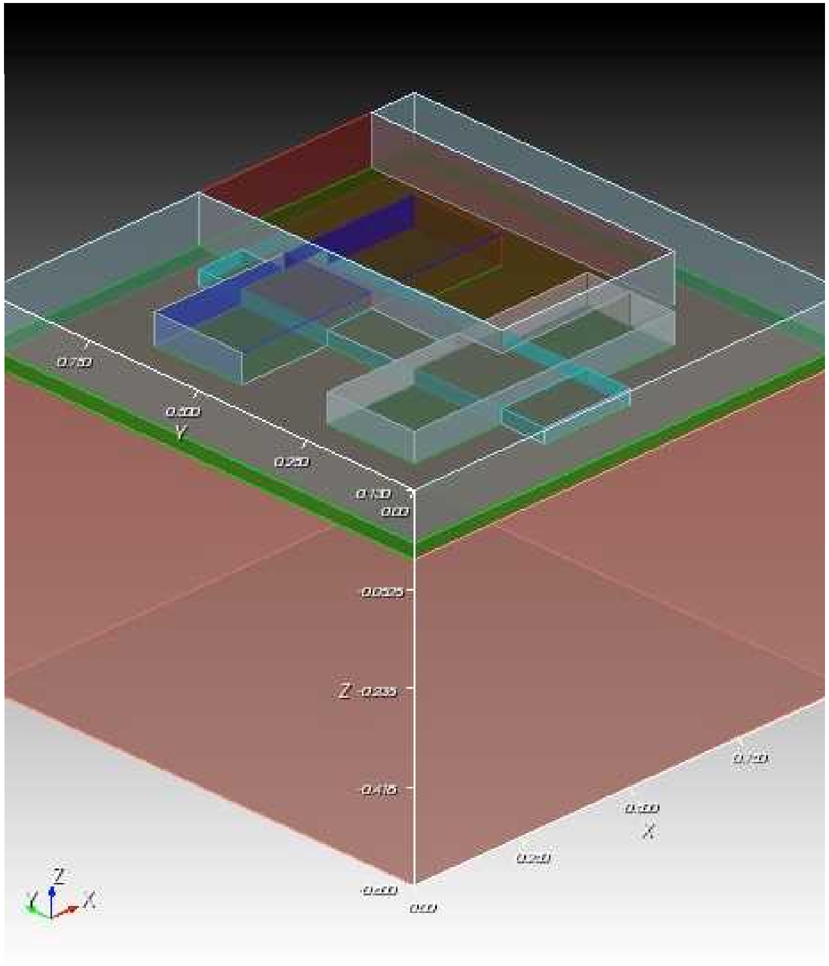
Next, click "**Save and create model**", which submits a mesh-creation job into the run queue of the machine running iQCAD. The model page lets you monitor the progress of this job, and after it finishes you should see summary information and images pertaining to the created mesh. If all looks good, you're ready to create a run!



A example of the 3D geometry of device with "overlay" levels



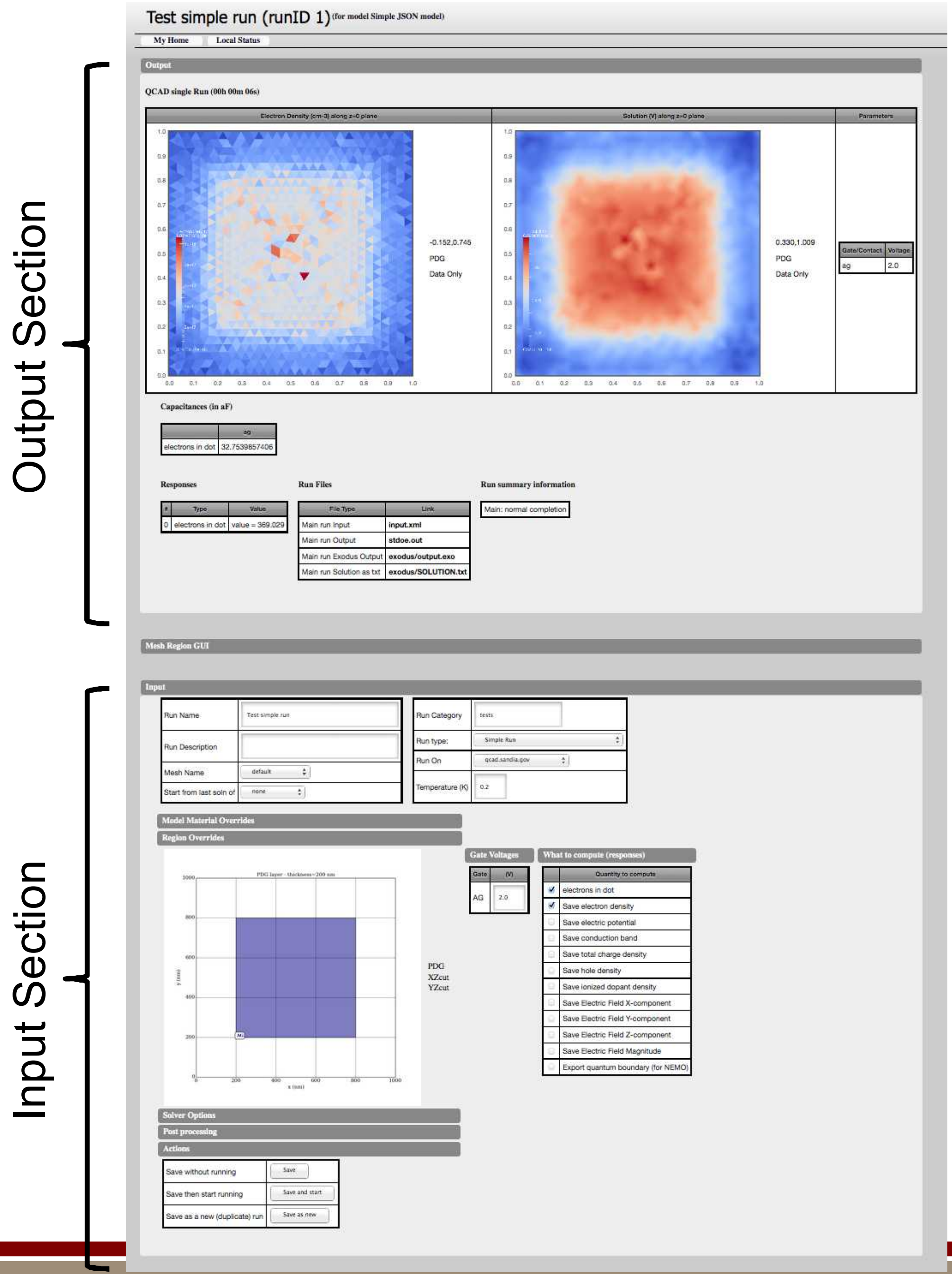




The Run Page

Each run page is divided into two main sections: **Output** and **Input**.

The **Output section** will not be present for Runs that have not been executed since they were last saved, i.e. for runs that don't have any up-to-date output to show you.



Run Page: Input Section

Sandia National Laboratories

Input

1

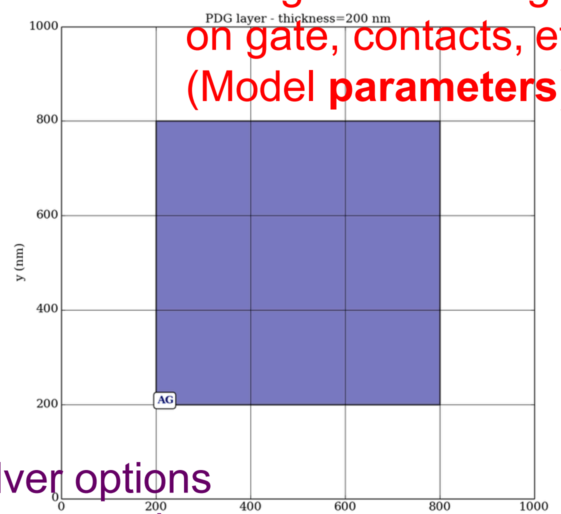
Run Name	Test simple run	Run Category	tests
Run Description		Run type:	Simple Run
Mesh Name	default	Run On	qcad.sandia.gov
Start from last soln of	none	Temperature (K)	0.2

General Information about the run

Model Material Overrides

Region Overrides

Voltages and charges set on gate, contacts, etc. (Model **parameters**)



PDG
XZcut
YZcut

Gate Voltages

Gate	(V)
AG	2.0

2

What to compute (responses)

3

	Quantity to compute
☑	electrons in dot
☑	Save electron density
☐	Save electric potential
☐	Save conduction band
☐	Save total charge density
☐	Save hole density
☐	Save ionized dopant density
☐	Save Electric Field X-component
☐	Save Electric Field Y-component
☐	Save Electric Field Z-component
☐	Save Electric Field Magnitude
☐	Export quantum boundary (for NEMO)

What outputs you want computed (Model **Responses**)

More technical solver options
Post processing commands (what images to create)
Action buttons

4

Solver Options

Post processing

Actions

Save without running	Save
Save then start running	Save and start
Save as a new (duplicate) run	Save as new

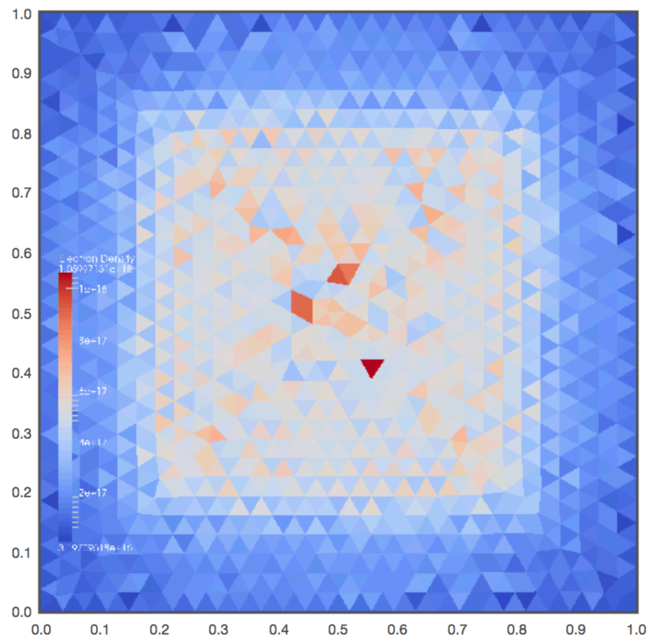
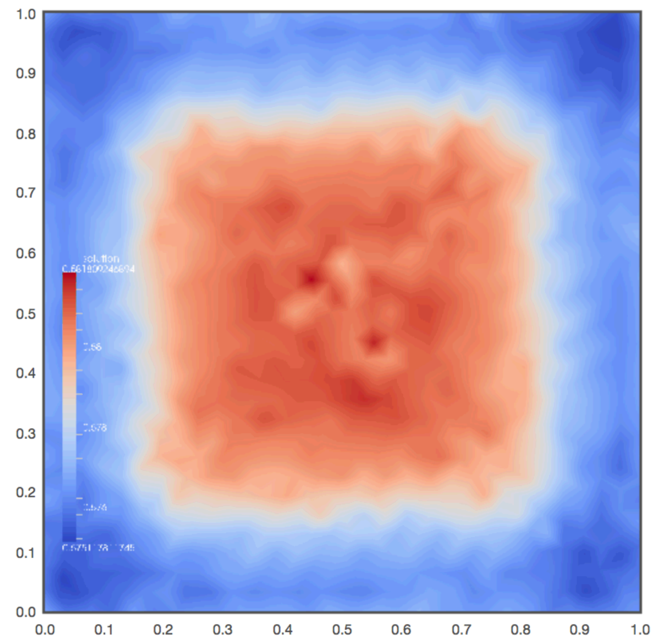
Fill out in order 1, 2, 3 then 4

Run Page: Output Section

Output

QCAD single Run (00h 00m 06s)

Table of images given by post-processing commands (see input section)

Electron Density (cm-3) along z=0 plane	Solution (V) along z=0 plane	Parameters
 0.047,0.954 PDG Data Only	 PDG Data Only	Gate/Contact: ag, Voltage: 2.0

Gate/Contact	Voltage
ag	2.0

Capacitances (in aF) – Note that "dot electrons" response must be checked to compute capacitances

	ag
electrons in dot	32.7539857406

Responses – correspond to which responses were selected

#	Type	Value
0	electrons in dot	value = 369.029

Run Files

File Type	Link
Main run Input	**input.xml**
Main run Output	**stdoe.out**
Main run Exodus Output	**exodus/output.exo**
Main run Solution as txt	**exodus/SOLUTION.txt**

Look here to see if run executed normally

Run summary information

Main: normal completion

– output.exo = "raw output"; visualize w/Paraview

Future work / capabilities

- Disorder Modeling
  - Gaussian “cloud” charge (recently implemented!)
  - Optimization of an unknown charge or potential profile (initial inquiry underway using Greg Von Winckel’s Rapid Optimization Library) – **requires distributed parameters & responses**.

- More (new) Physics:
  - Magnetic field in Schrodinger solutions - **requires complex potentials (perhaps possible w/TPetra now?)**
  - Valley physics: required to obtain more accurate solutions for semiconductor materials with multiple valleys (like Silicon). **I think Albany/Trilinos has everything needed for this – we just need time & money.**

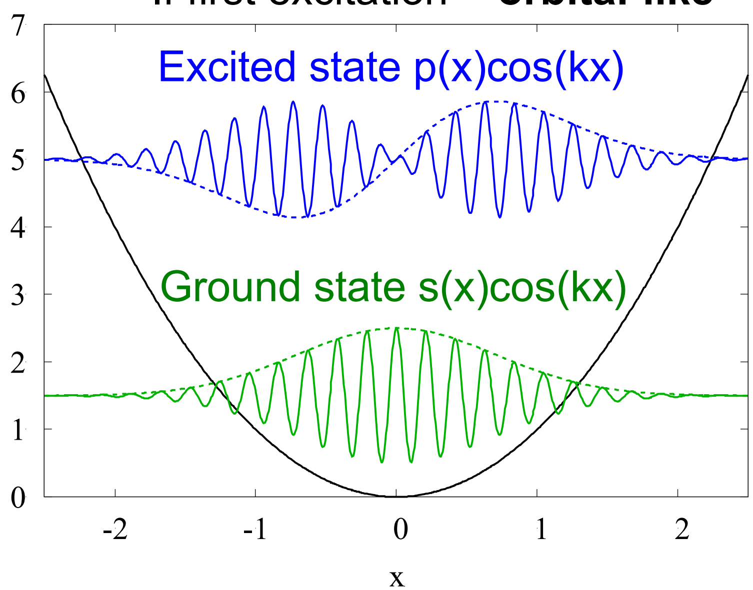
- Usability enhancements:
  - Output quad-point quantities on nodal mesh (for interpolation)
  - Cleaner interoperation between different Albany Application or model evaluator objects (like multiphysics, but same mesh)
  - Docs for folks new to Albany

Valley Physics in Albany

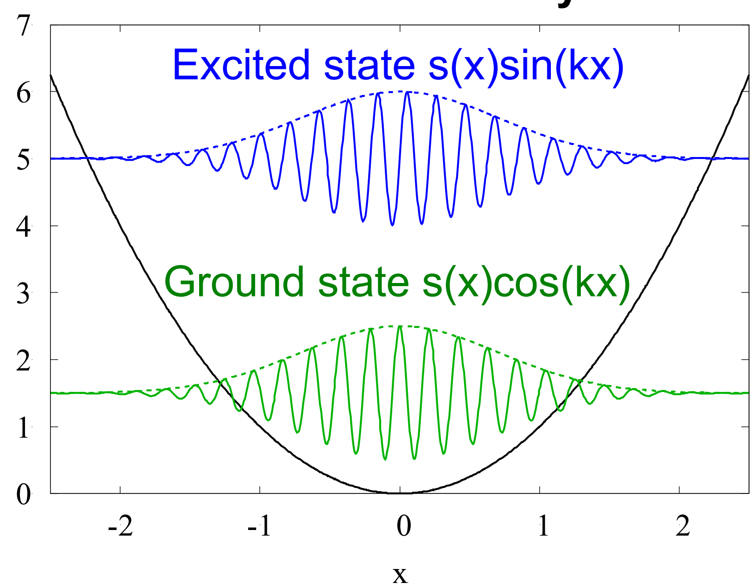
- Basic idea:
  - The wave function solution to the Schrodinger equation (just an eigenvalue equation) can be decomposed into a slowly-spatially-varying envelope multiplied by a fast-oscillatory term.
  - We want to solve for the envelope on a (coarse) finite element mesh, and handle the fast oscillatory pieces by coupling multiple equations of motion (one per valley).

Toy Example: lowest two states in a parabolic external potential w/2 valleys

If first excitation = **orbital-like**



If first excitation = **valley-like**



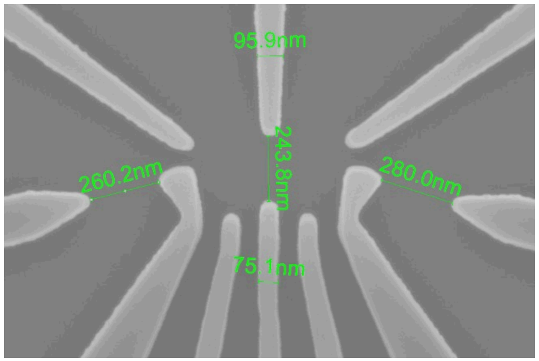
Summary

- **QCAD** is an example of a finite-element application which solves fairly simple equations related to understanding quantum devices.
- **We attribute much of QCAD's success at maintaining a broad user base to:**
  - Albany & Trilinos themselves being good at what they do (I didn't stress this enough!)
  - A relatively sophisticated (graphical) user interface that has been bolted on top of the core scientific software.
- Trilinos and Albany already contain some enabling features for potential future directions for QCAD; I'd love to see more!

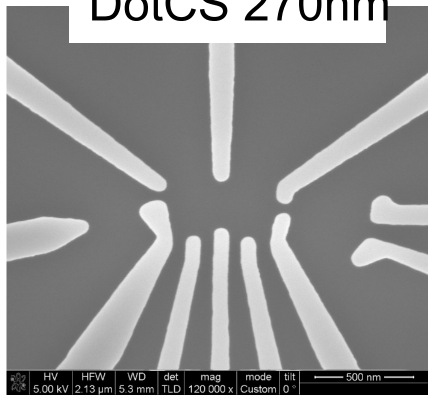
- **Thank you!!** (for everyone's work contributing to Albany)

Example geometries:

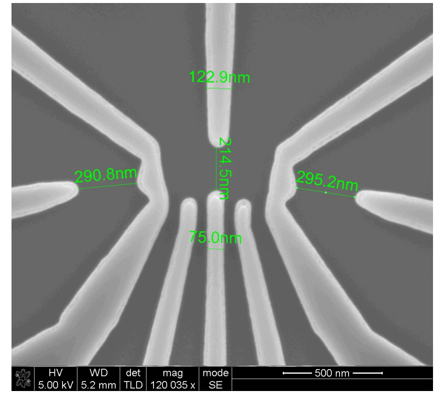
Ottawa Thin B 270nm



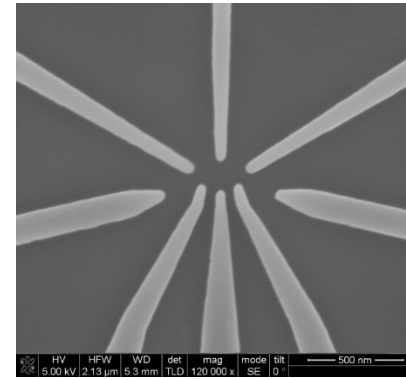
Ottawa Thin B DotCS 270nm

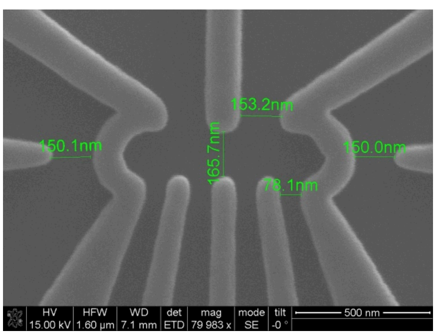


Ottawa Thin 270nm

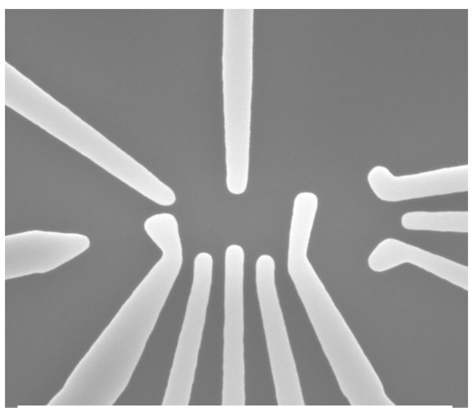


Radial 2CS

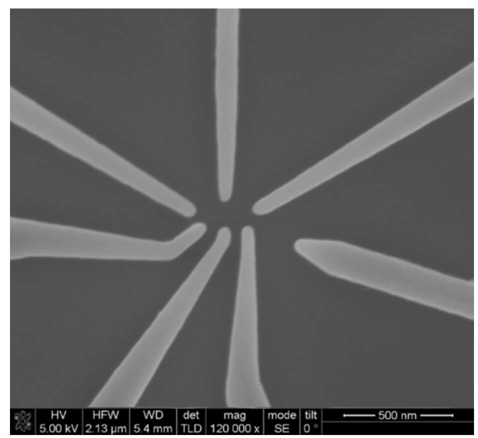




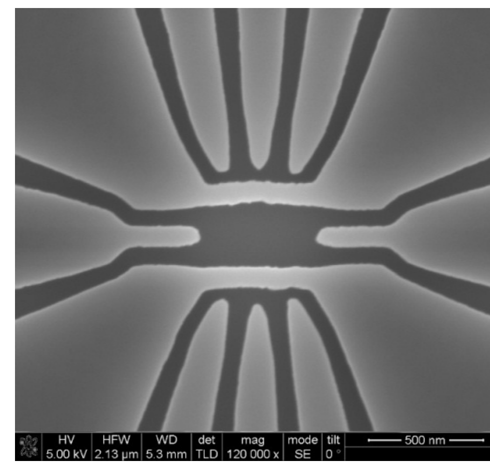
Ottawa Fanned Mod



Ottawa Thin B Open DotCS 270nm



Radial 1CS



Gated Wire 60nm

Summary of Device Modeling

- Physical device modeling involves lots of numerical studies
- Semi-classical (3D) models of devices go a long way to predicting behavior.
  - We believe discrepancies are caused more by disorder than quantum-ness not captured by the semi-classical equations.
- Capacitance modeling has proven very useful in many cases of initial device design & tune-up, and is much less computationally demanding than a nonlinear Poisson calculation.
- Other techniques not mentioned that we use frequently:
  - Valley-aware effective mass theory
  - Tight-binding (NEMO)