SAND2012-10367P

Using QCAD for the design of double quantum dots

Erik Nielsen, Suzey Gao, Ralph Young, Rick Muller, Andy Sallinger

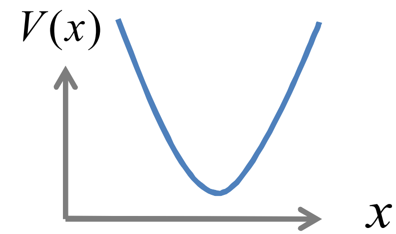
2nd Albany Developers Meeting
Oct 2, 2012

Sandia National Laboratories is a multi-program 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.

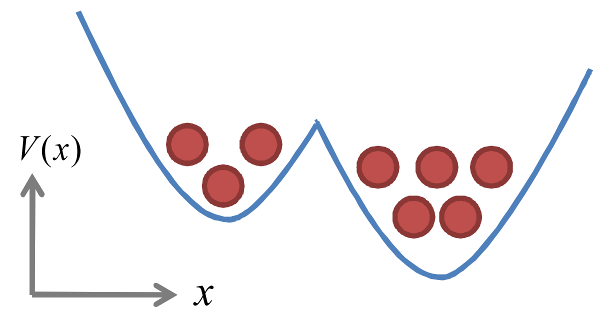
Sandia National Laboratories

Quantum dots

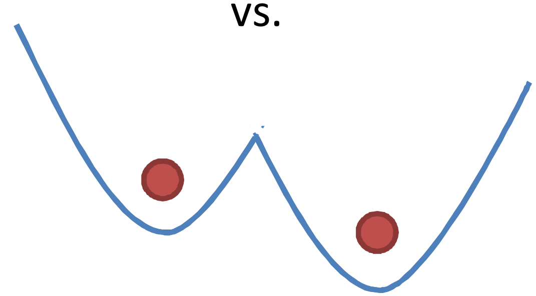
- **Quantum dot**: a potential energy "pit" that confines electrons to a point in space



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



vs.



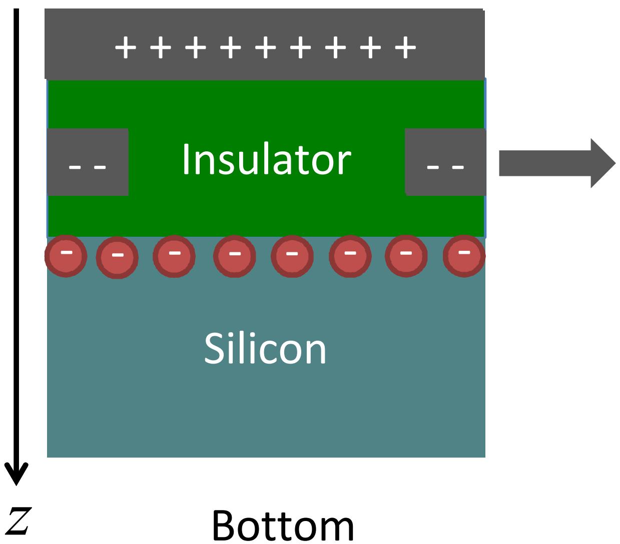
- **Useful** for quantum computing **if** we have just a **few electrons in each dot** and can move them back & forth.
  - # of e^{-} depends on "depth" and "size" of dots

Sandia National Laboratories

Double quantum dot devices

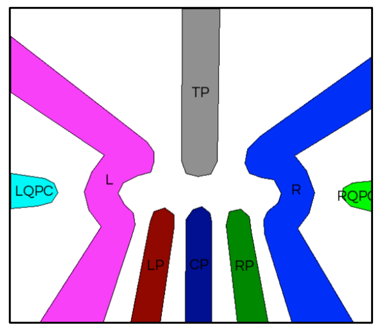
- One way to make double quantum dots:

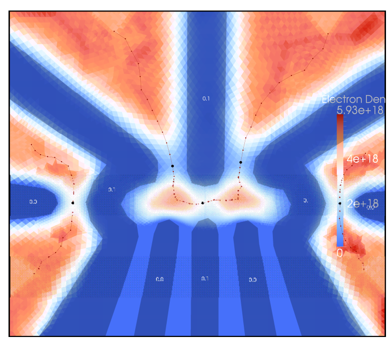
Top



Bottom

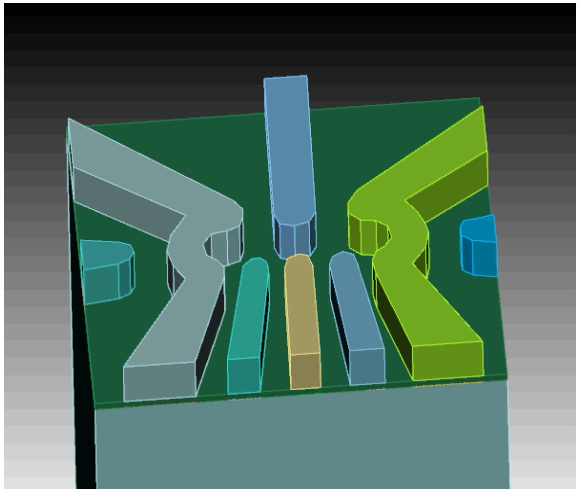
Horizontal cut (constant z)
"Depletion Gates"





Electron Density (**red** = higher)

3D view

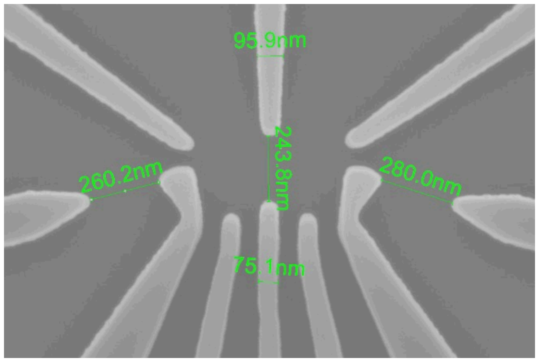


(this gets meshed)

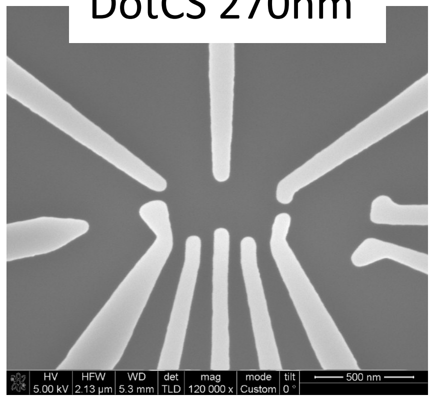
Sandia National Laboratories

Many patterns

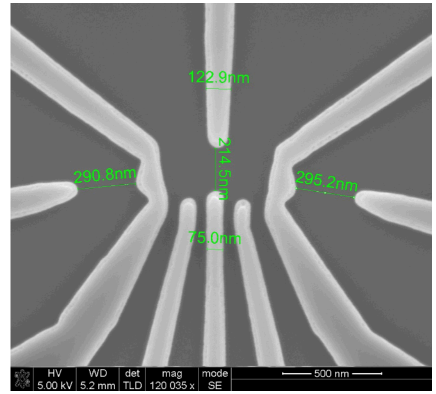
Ottawa Thin B 270nm



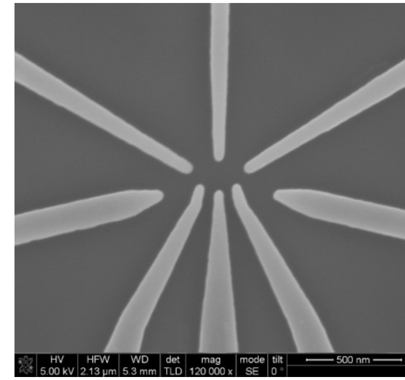
Ottawa Thin B DotCS 270nm

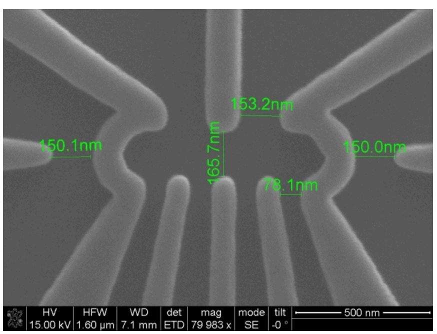


Ottawa Thin 270nm

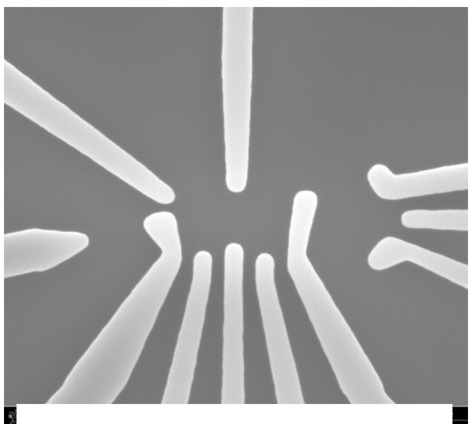


Radial 2CS

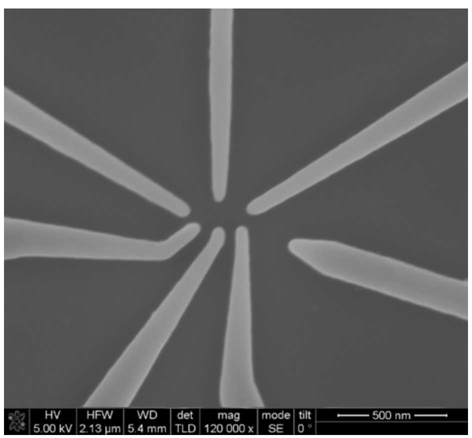




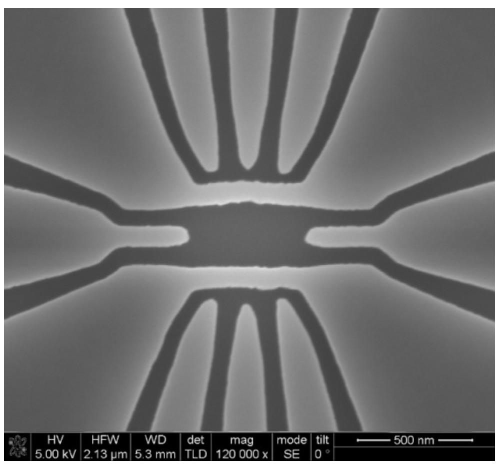
Ottawa Fanned Mod



Ottawa Thin B Open DotCS 270nm



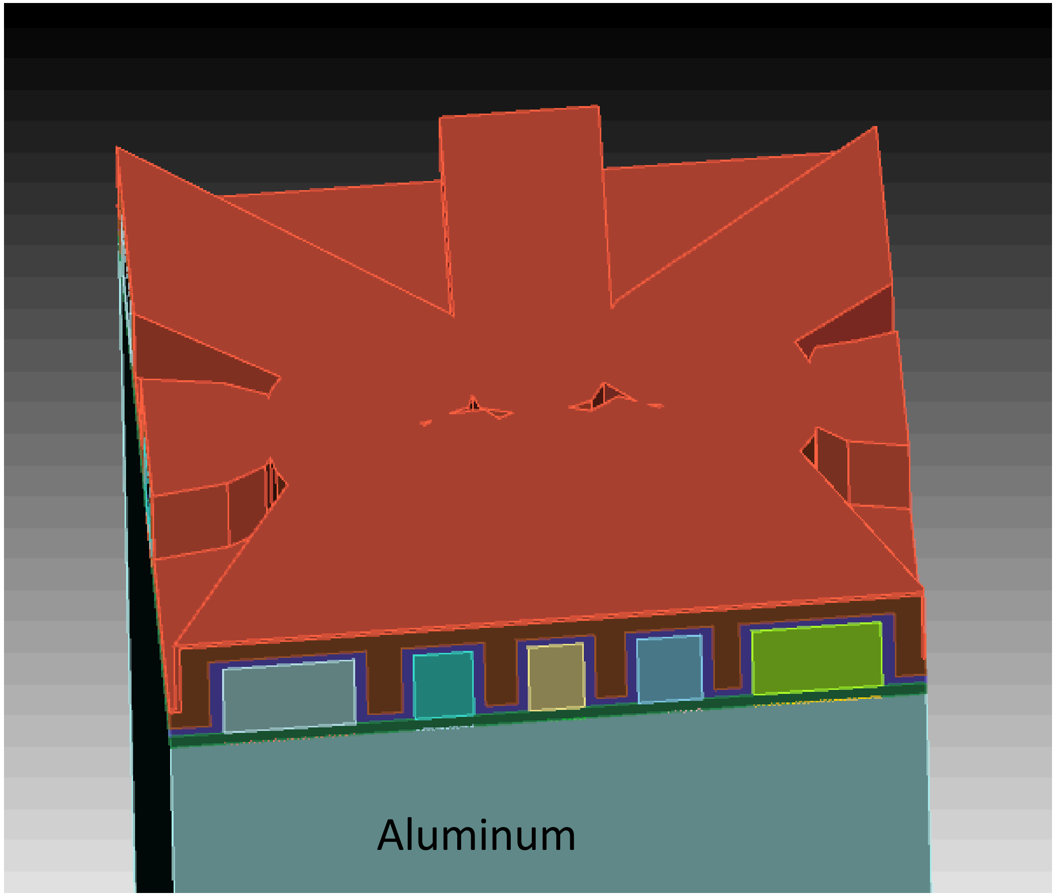
Radial 1CS



Gated Wire 60nm

Sandia National Laboratories

Layers of a Double quantum dot:



Sandia National Laboratories

Designing good Double Quantum Dots

Central Question: How do I make a "good" quantum dot?

(experiments take a long time, would like to predict good designs)

What makes a good double quantum dot?

1. Small number of **electrons in dot**
2. **Barriers** allowing electrons in and out of dot on border of being on/off
3. Channel which measures the electrons in the dot is operational (another **barrier**)

Responses

What knobs can we turn?

1. Voltages
2. Geometry, i.e. placement of gates

Parameters (mesh mover?)

Sandia National Laboratories

Equations to solve (reminder)

- **Poisson's equation** (like heat diffusion)

$$\nabla \cdot \varepsilon \nabla \phi = \rho(\phi)$$

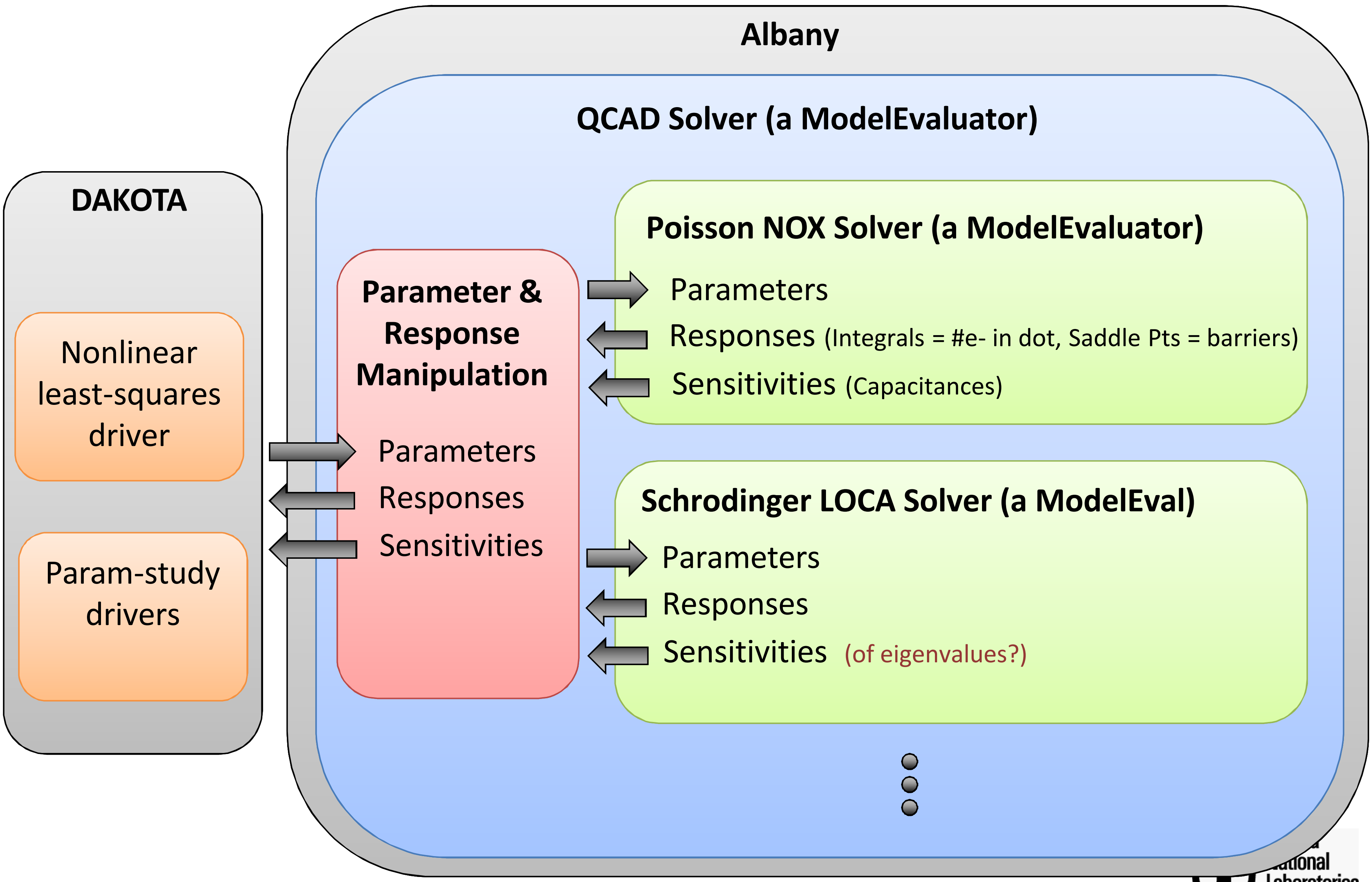
- Single and many-electron **Schrodinger equations** (matrix diagonalization)

$$H\psi_i = E_i \psi_i$$

But possibly multiple of each type in some types of solutions (e.g. quantum mechanical solutions) -- Albany very flexible

Sandia National Laboratories

QCAD in the context of Albany & DAKOTA



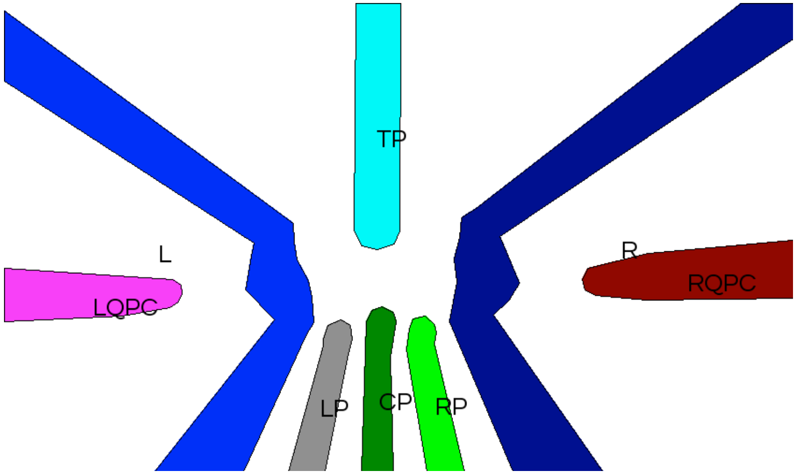
Optimization of OttawaFlat270

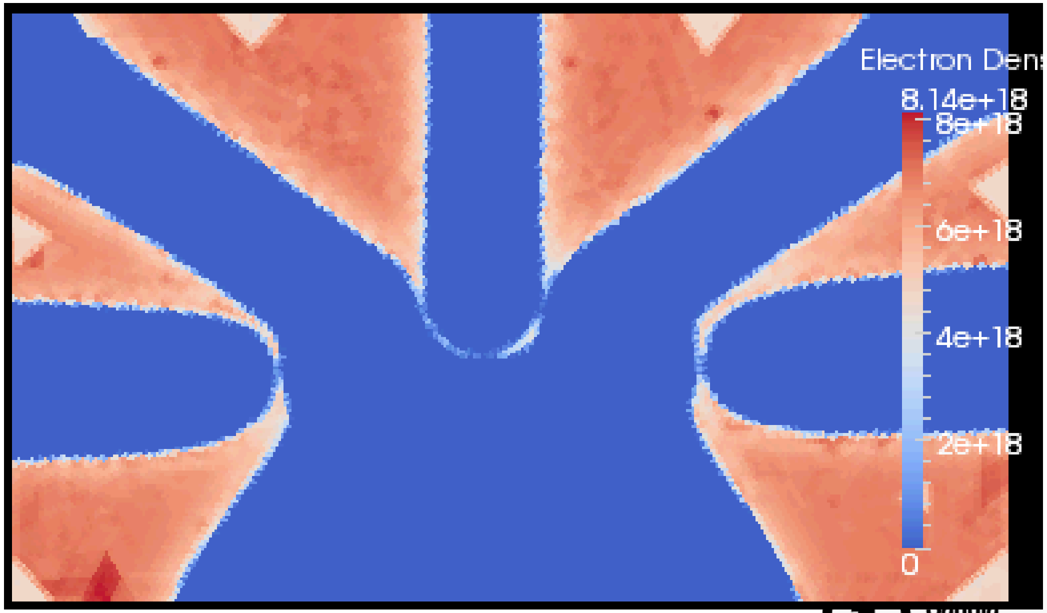
Targets (NL-least sq terms):

1. 1 e- in left dot
2. Left tunnel barrier just turning on
3. Dot barrier just turning on
4. Left QPC barrier just turning on

Adjustable parameters (gate voltages):

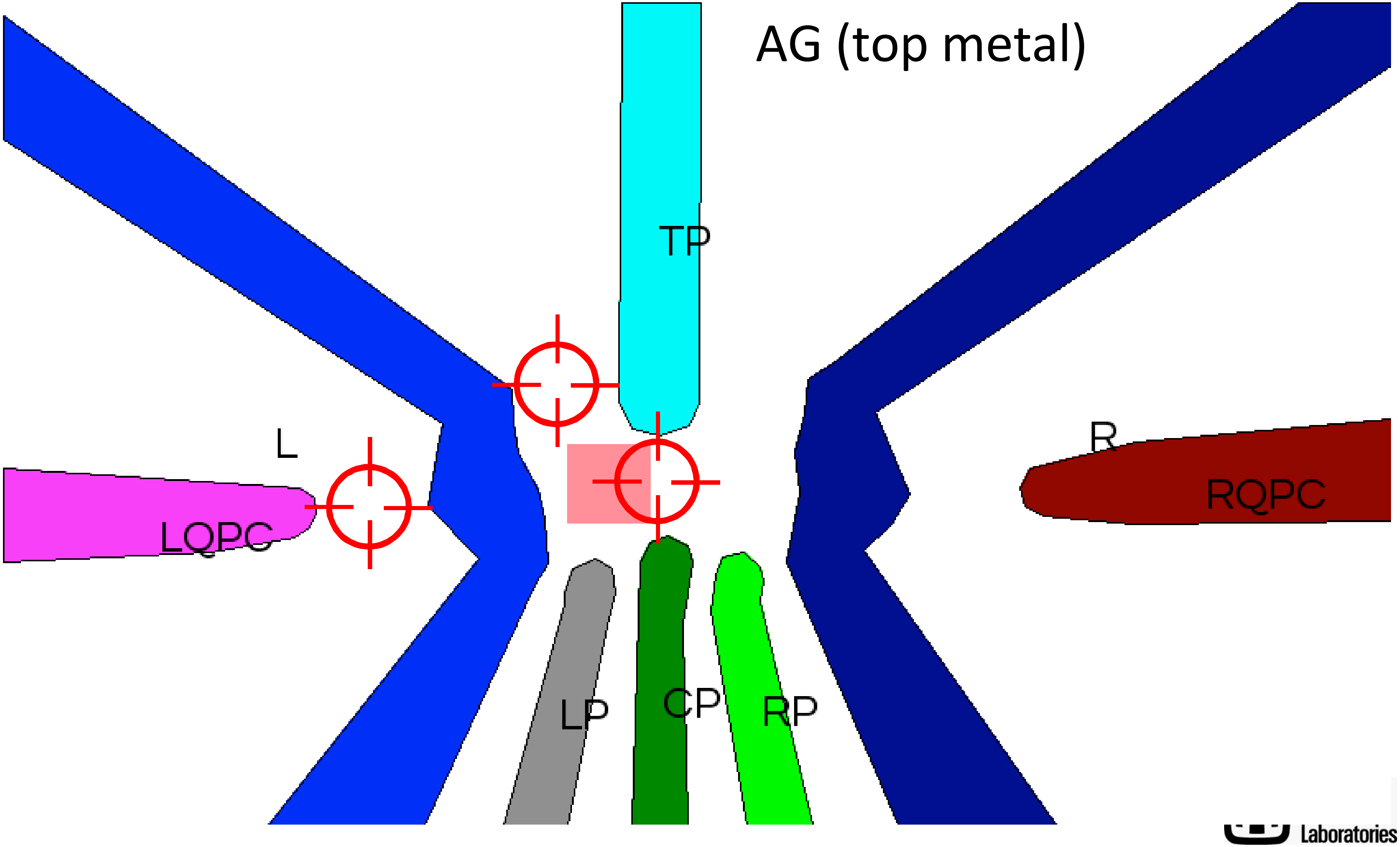
1. AG (top accumulation gate – not shown)
2. TP
3. CP
4. LP tied to RP
5. L tied to R
6. LQPC tied to RQPC



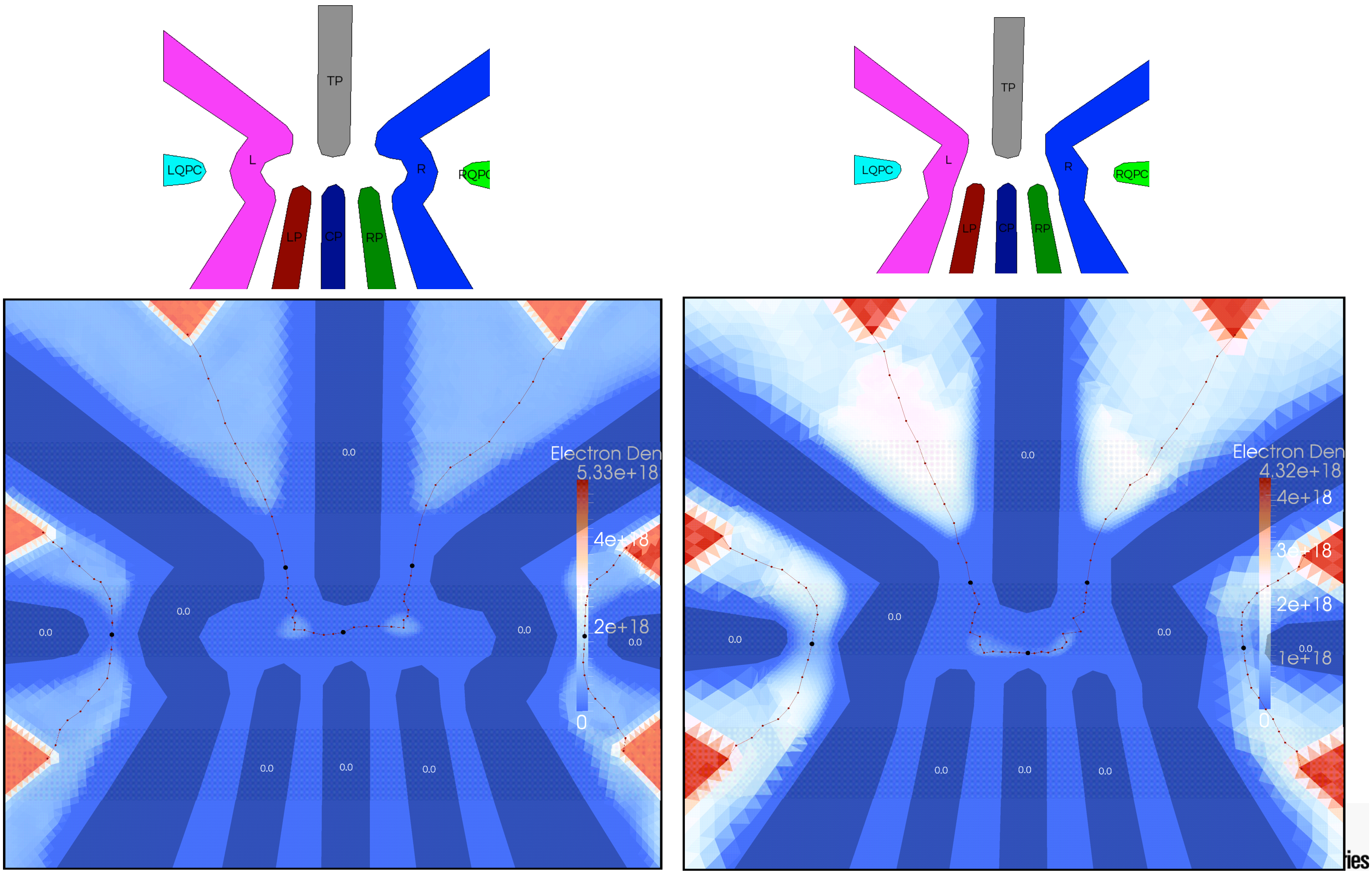


Sandia National Laboratories

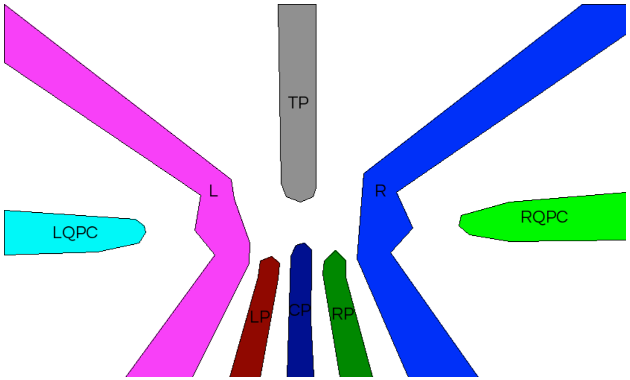
Optimization of OttawaFlat270

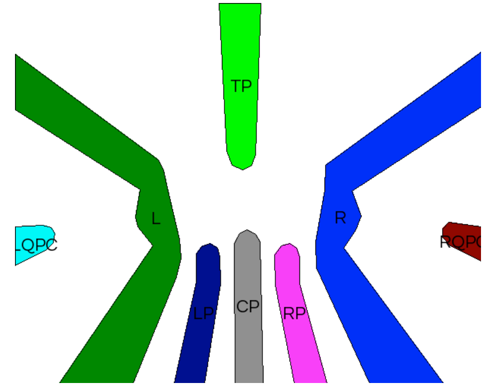


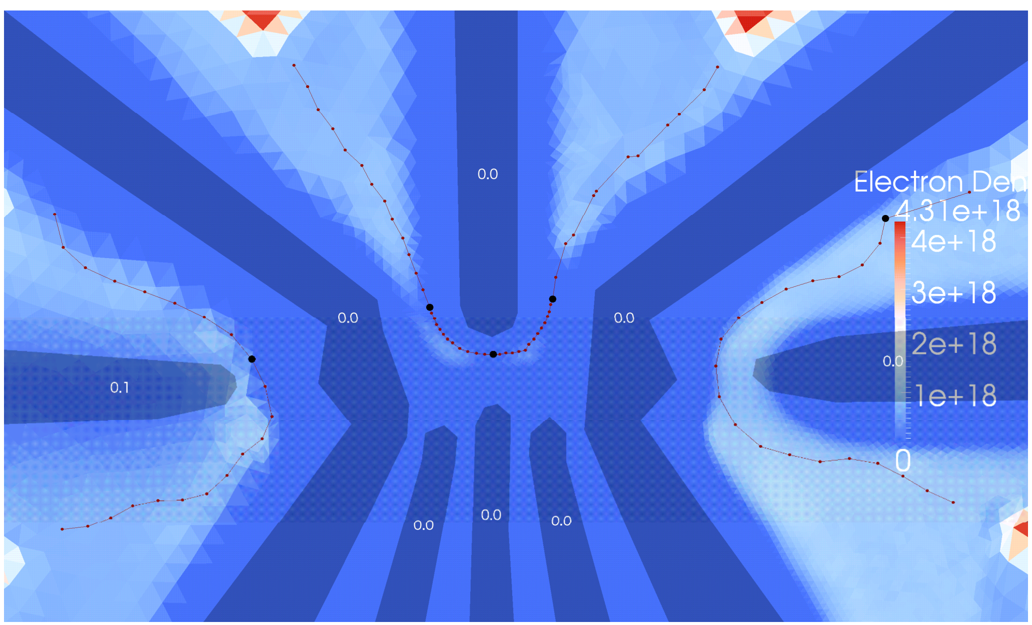
Design Comparison

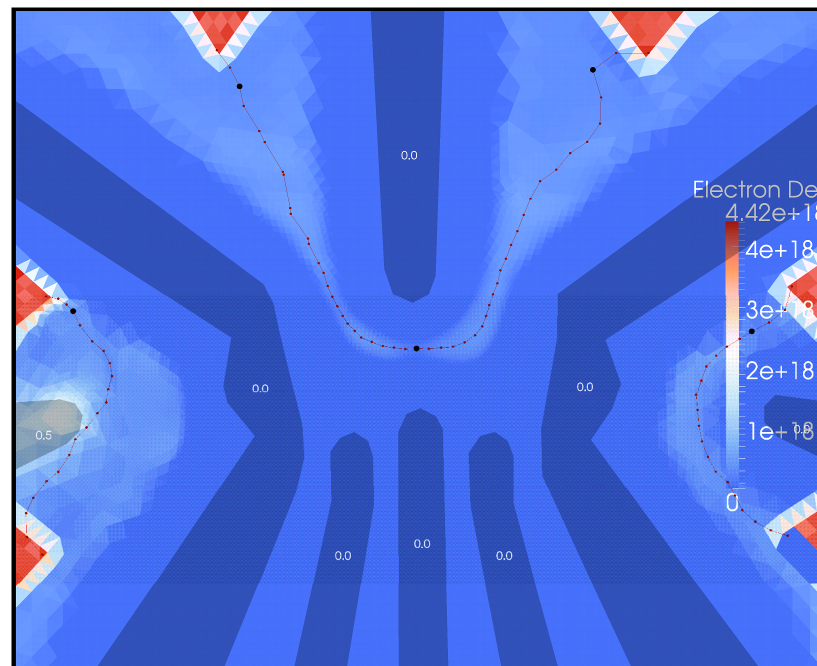


More designs









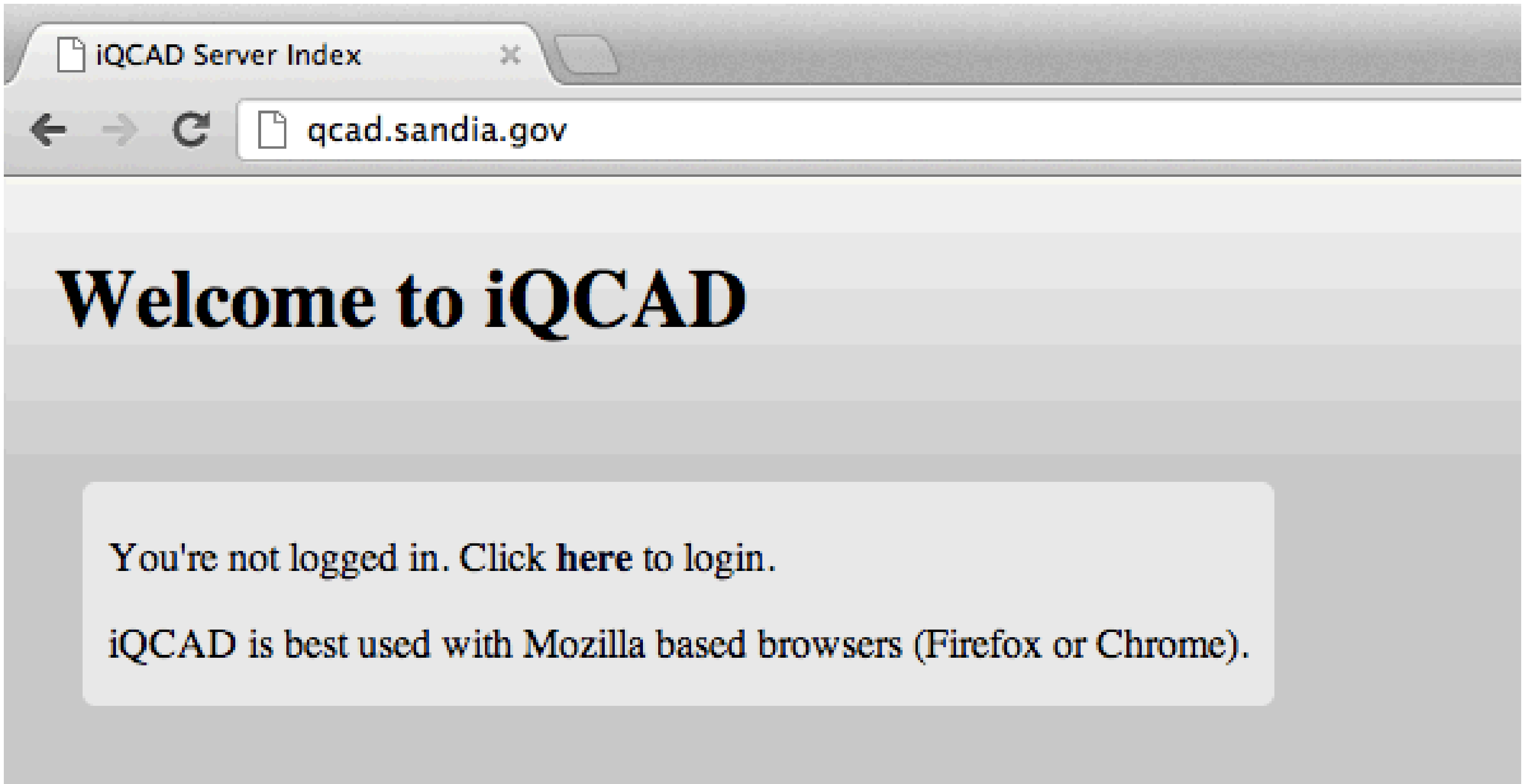
What's helped and what's lacking

- **Helpful features** of Albany architecture:
  - Ease of creating custom problems (implementing new evaluators). [Evaluation tree automatic]
  - Ease of creating custom responses (again, evaluators!)
  - Automatic Differentiation
    - no Jacobians to compute
    - speed up for NLS optimizations
    - C++ templating = only implement things once
- **What we'd like to see**:
  - No complex numbers
  - Method of passing field data between sub-solvers could be more integrated into the parameter/response framework (currently we use field manager hooks)

Sandia National Laboratories

This slide is intentionally left blank

iQCAD: an QCAD-Albany GUI



Erik Nielsen, Suzey Gao, Ralph Young,
Rick Muller, Andy Sallinger

Sandia National Laboratories

Typical process to simulate a device

1. **CUBIT** to create mesh
2. **Spread** mesh (not necessary now thanks to Glen!)
3. Create Albany/Dakota **input files**
4. Run **Albany**
5. Check output file to **see if it succeeded**
6. **Join** output mesh
7. **Image** joined mesh using Paraview

Sandia National Laboratories

A web interface to Albany: “iQCAD”

A **web-based wrapper** (currently written in python) which performs the following pre/post processing functions:

- Input file generation (input.xml, materials.xml, dak.in, …)
- Submission to run queue (interfaces w/SLURM job queuing)
- Imaging & post-processing (scripted calls to Paraview)
- Saving input and output data in a database (MySQL backend)
- User control (keeps different users’ data separate)
- Mesh preparation and checking (check .exo vs cubit .jou; spreading meshes)

DEMO

(http://qcad.sandia.gov)

Why does it work well for QCAD?

- Lots of different classes of models/meshes
- Many similar models of each class
- Predictable location for imaging
- Potentially lots of use by non-experts