Simulations of the Brighter-Fatter Effect

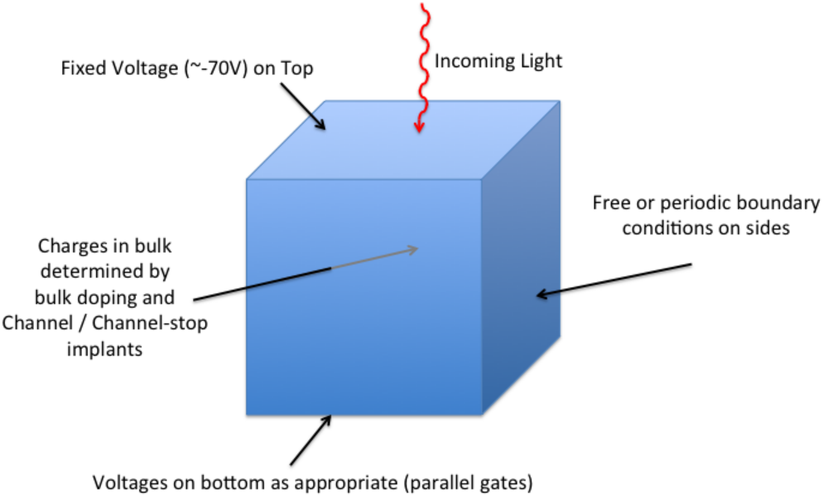
Craig Lage

September 10, 2015

Outline

- Introduction
- Code additions for Simulations of B-F effect.
- Results
- Summary Next Steps

Boundary Conditions - Typical Simulation $100\mu m$ Cube.



Solving Poisson's Equation on a Grid

$$\nabla^2\varphi = \rho$$

$$\frac{\partial^2\varphi_{\mathrm{i,j,k}}}{\partial \mathrm{x}^2} = \frac{(\varphi_{\mathrm{i+1,j,k}} - \varphi_{\mathrm{i,j,k}}) - (\varphi_{\mathrm{i,j,k}} - \varphi_{\mathrm{i-1,j,k}})}{\mathrm{h}^2}$$

$$(\varphi_{\mathrm{i+1,j,k}}+\varphi_{\mathrm{i-1,j,k}}+\varphi_{\mathrm{i,j+1,k}}+\varphi_{\mathrm{i,j-1,k}}+\varphi_{\mathrm{i,j,k+1}}+\varphi_{\mathrm{i,j,k-1}}-6*\varphi_{\mathrm{i,j,k}}) = \mathrm{h}^2*\rho_{\mathrm{i,j,k}}$$

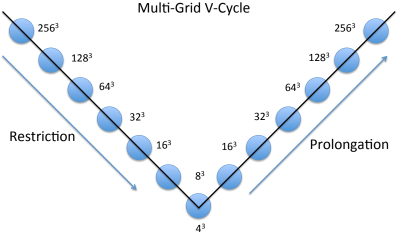
$$\varphi_{\mathrm{i,j,k}} = \frac{1}{6}*(\varphi_{\mathrm{i+1,j,k}}+\varphi_{\mathrm{i-1,j,k}}+\varphi_{\mathrm{i,j+1,k}}+\varphi_{\mathrm{i,j-1,k}}+\varphi_{\mathrm{i,j,k+1}}+\varphi_{\mathrm{i,j,k-1}}-\mathrm{h}^2*\rho_{\mathrm{i,j,k}})$$

$$\varphi^{(\mathrm{n+1})}_{\mathrm{i,j,k}} = \frac{1}{6}*(\varphi^{(\mathrm{n})}_{\mathrm{i+1,j,k}}+\varphi^{(\mathrm{n})}_{\mathrm{i-1,j,k}}+\varphi^{(\mathrm{n})}_{\mathrm{i,j+1,k}}+\varphi^{(\mathrm{n})}_{\mathrm{i,j-1,k}}+\varphi^{(\mathrm{n})}_{\mathrm{i,j,k+1}}+\varphi^{(\mathrm{n})}_{\mathrm{i,j,k-1}}-\mathrm{h}^2*\rho_{\mathrm{i,j,k}})$$

.

- Conceptually, we simply iterate until convergence.
- In practice, it converges very slowly - millions of iterations are required.

Multi-Grid Methods to the Rescue

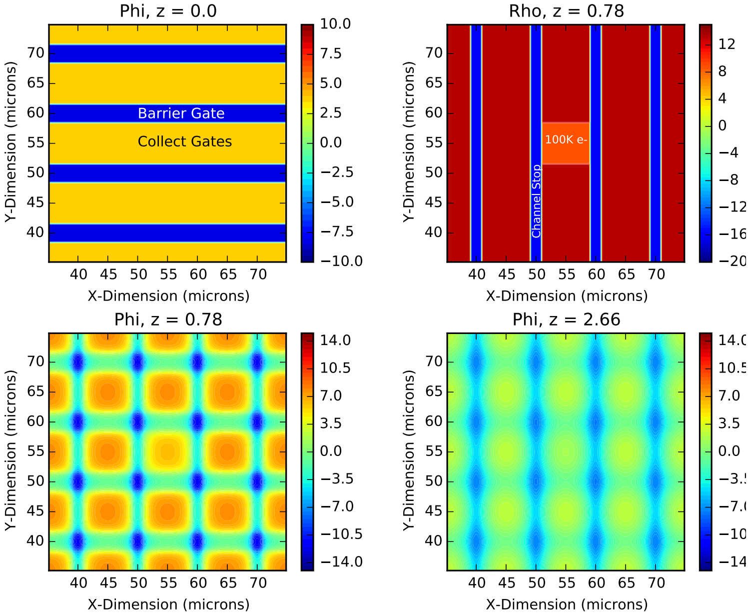


Finest Grid	Cells/Pixel	Grid Spacing	Time (laptop)
160^3	16	0.625 micron	5 sec.
320^3	32	0.3125 micron	48 sec.
640^3	64	0.15625 micron	5 min.

$\longleftarrow$

- Each successive step down is ≈ 8 times faster than the next larger grid.
- In practice, I iterate the coarsest grid to machine precision, then 2X fewer iterations at each finer grid, ending with 64 iterations at the finest grid.

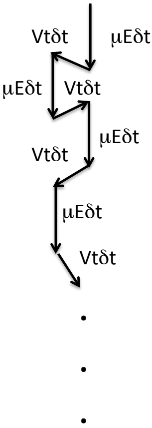
Pixel Array Summary Plot



Simulation Strategy for B-F effect.

- Solve Poisson's equation for postage stamp (using 9x9 pixels) with all pixels empty.
- Choose a random location within the central pixel as the center of a 2D Gaussian spot.
- Determine starting locations for N electrons in the 2D Gaussian spot.
- Propagate these electrons down to their collecting gates.
- Re-solve Poisson's equation with these wells now containing the appropriate charge.
- Repeat with N more electrons.
- I have been using 10,000 electrons per step, which places about 1000 electrons in the central pixel, so about 100 iterations are needed to fill the central pixel.
- In practice, repeat for more than one spot (typical 10), each with a different central location.

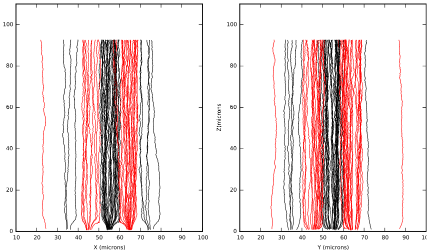
Electron Propagation



Each time step:
Drift velocity of μ*E
Thermal velocity of Vt in a random direction

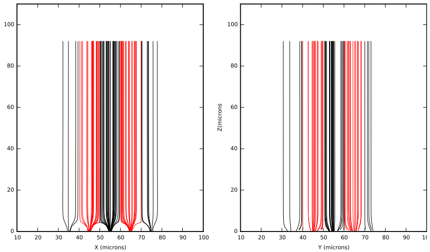
Impact of electron diffusion - electron paths

Electron Path Plot - Vertical Zoom = 1



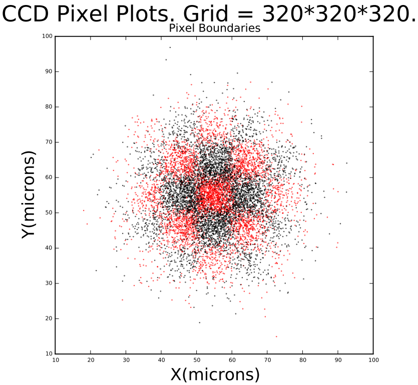
Theoretical Diffusion

Electron Path Plot - Vertical Zoom = 1

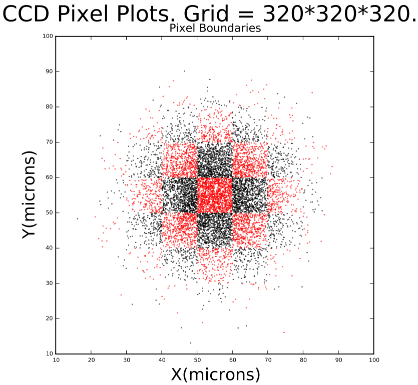


Diffusion turned off

Impact of electron diffusion - Pixel Locations



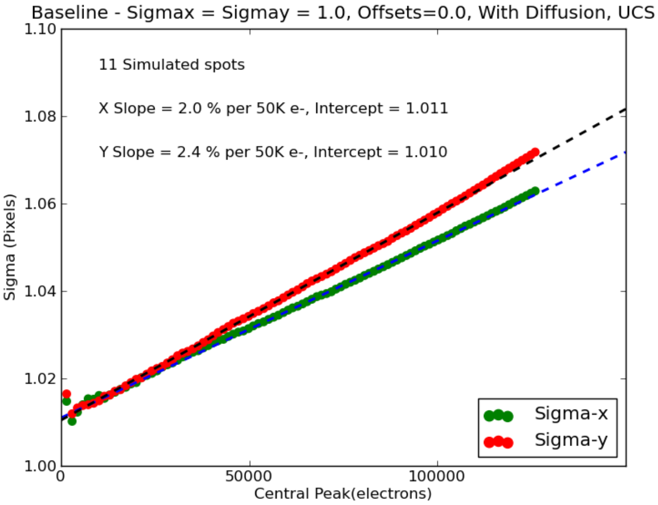
Theoretical Diffusion



Diffusion turned off

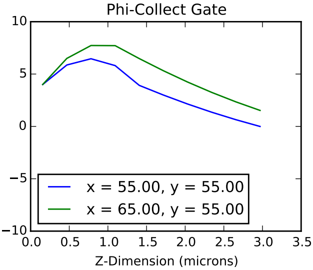
Colors coded by ending pixel location.

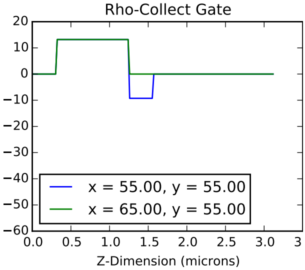
Initial Simulation of B-F Effect

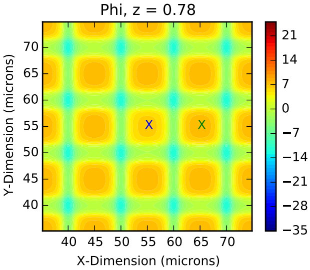


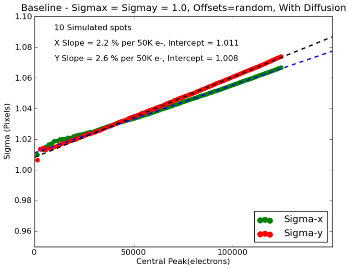
One spot (100 steps, 1,000,000 electrons) takes a little less than 1 hour on my laptop. Multiple spots can be run in parallel.

Vertical location of collected charge impacts BF slopes



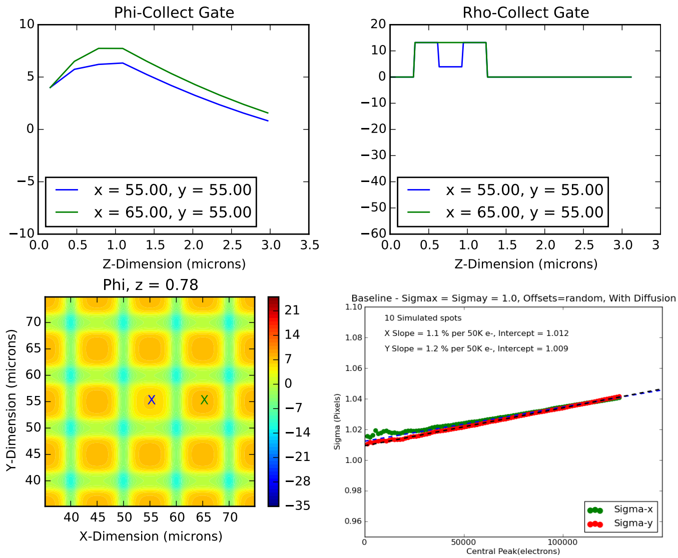






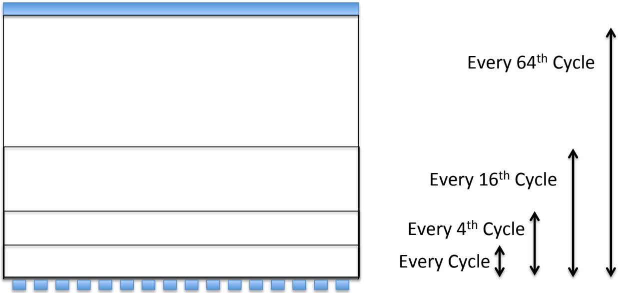
Collected Charge at $z = 1.4$ um

Vertical location of collected charge impacts BF slopes



Collected Charge at z = 0.8 um

Code Speed-up by concentrating iterations where the spatial frequency is highest



If T is the time to cycle through the entire array:

Old method (64 cycles) = 64T

New method (64 cycles) = 64(T/8)+16(T/4)+4(T/2)+2 = 16T

=> 4X Speedup

Summary and Next Steps

- Summary
  - Code is successfully simulating the B-F effect.
  - Slopes agree fairly well with measurements and can be tweaked by varying vertical charge placement (and hence dipole moment).
  - Y-slope > X-slope as expected due to stronger confinement due to channel stops.
  - Capability exists to vary applied voltages, etc.
- Next Steps
  - Improve B-F measurements and compare to simulations
  - Post the latest Poisson solver code on github. Earlier code is posted (https://github.com/craiglagegit/Poisson_CCD), but does not include the enhancements discussed today. This will be done soon.
  - Get the code running on NERSC so we can run more trials and more spots.