Using Radiance to Generate Lighting and Solar Surface Heat Gain Schedules for Energy Plus

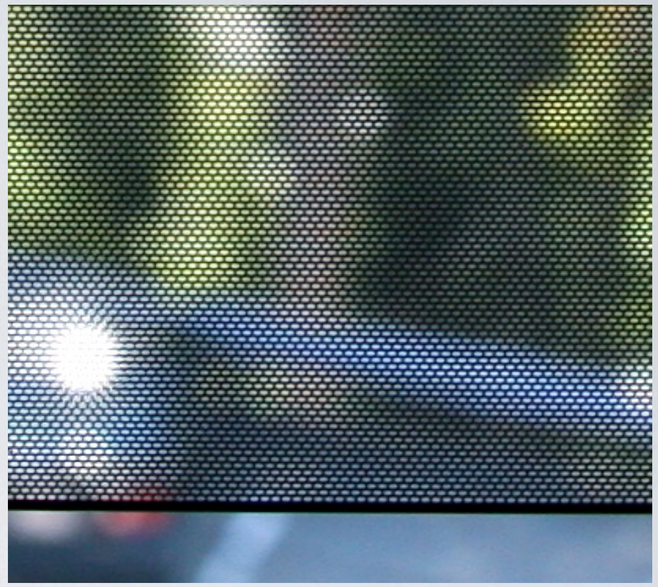
Andy McNeil, LBNL

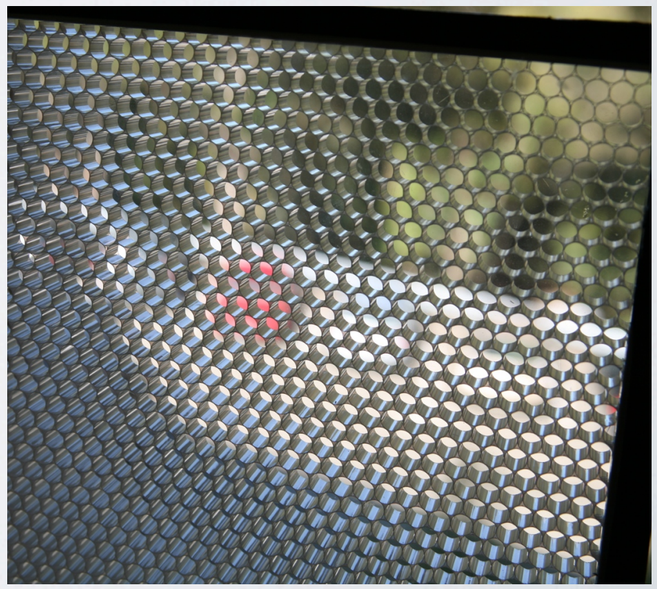
Contents

- Objecitves
- Process:
  - 1 - Make BSDFs (solar and visible wavelengths)
  - 2 - Generate lighting schedule (typical 3-phase)
  - 3 - Generate surface irradiance schedule (modified 3-phase)
  - 4 - Speed things up (dctimestep optimization)
  - 5 - Run energy plus

Objectives

- Whole building energy simulation
- Complex Fenestration Systems (CFS)
- Run parametric simulations to simulate performance for:
  - WWR
  - LPD
  - Climate
  - Building Energy Performance
  - Glazing Type





Objectives

- Why use Radiance to Replace E+ functionality?
    - Energy Plus doesn't support CFS (yet).
    - Split flux method isn't great

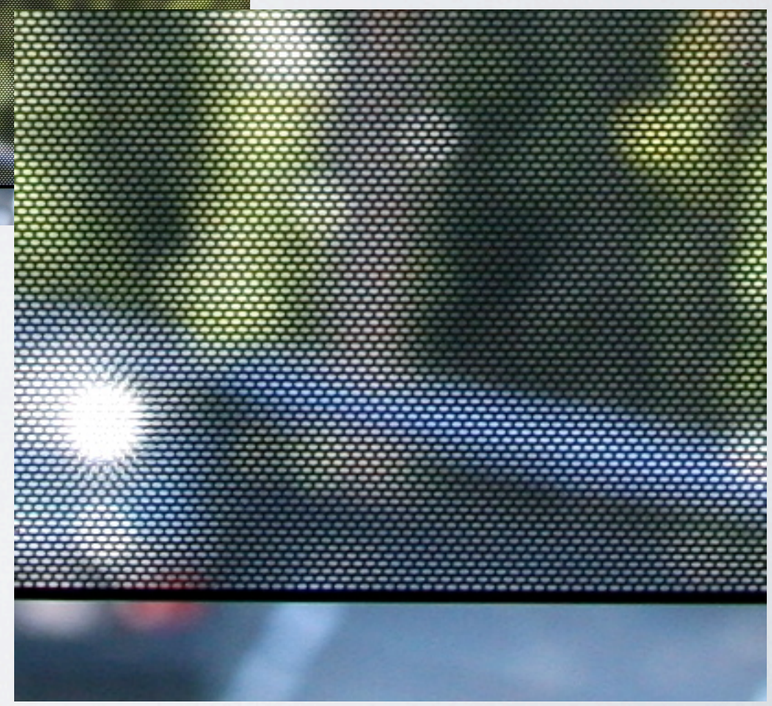
Make BSDFs for CFS

- Three systems:
    - Micro-perforated metal screen
    - Expanded metal mesh
    - Columnar structure

- Need BSDF for solar and visible spectrum
    - Run genBSDF using solar reflectance properties
    - change wavelength tag from “visible” to “NIR”
    - combine with visible in one BSDF

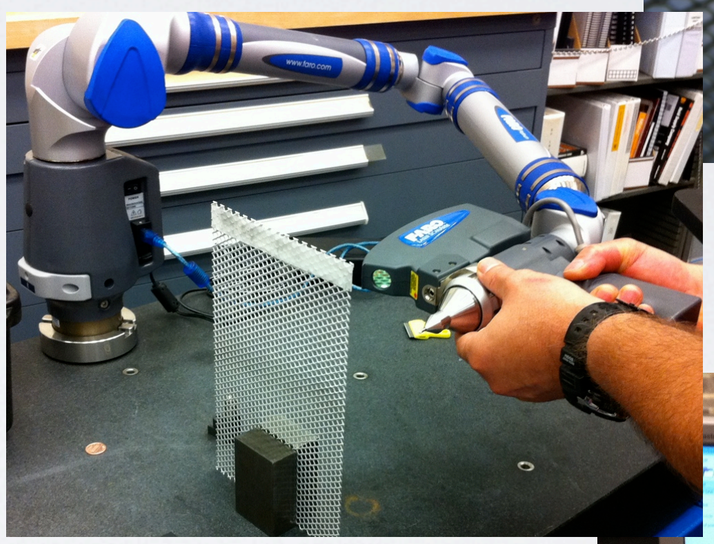
Micro Perf Mesh

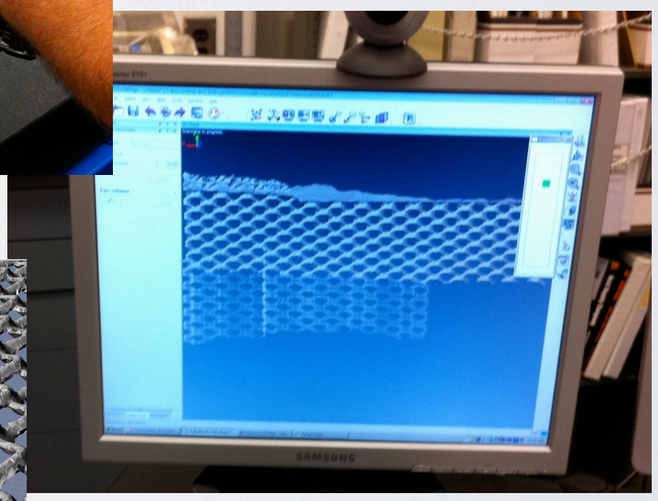
- Geometry from manufacturer
- Un-punched sample to get surface reflectance properties
- (David discussed yesterday)

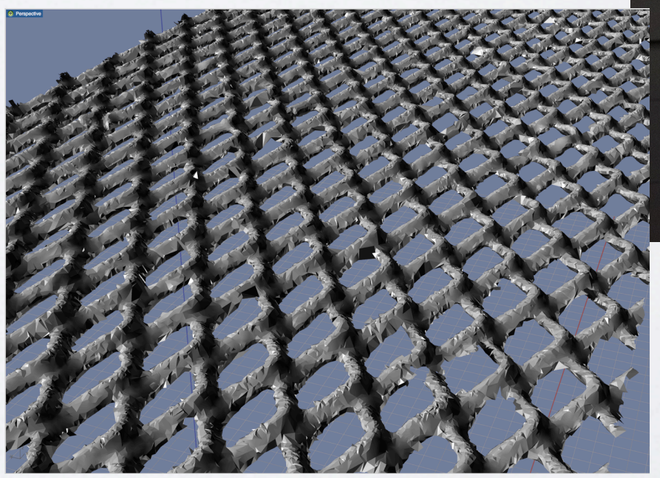


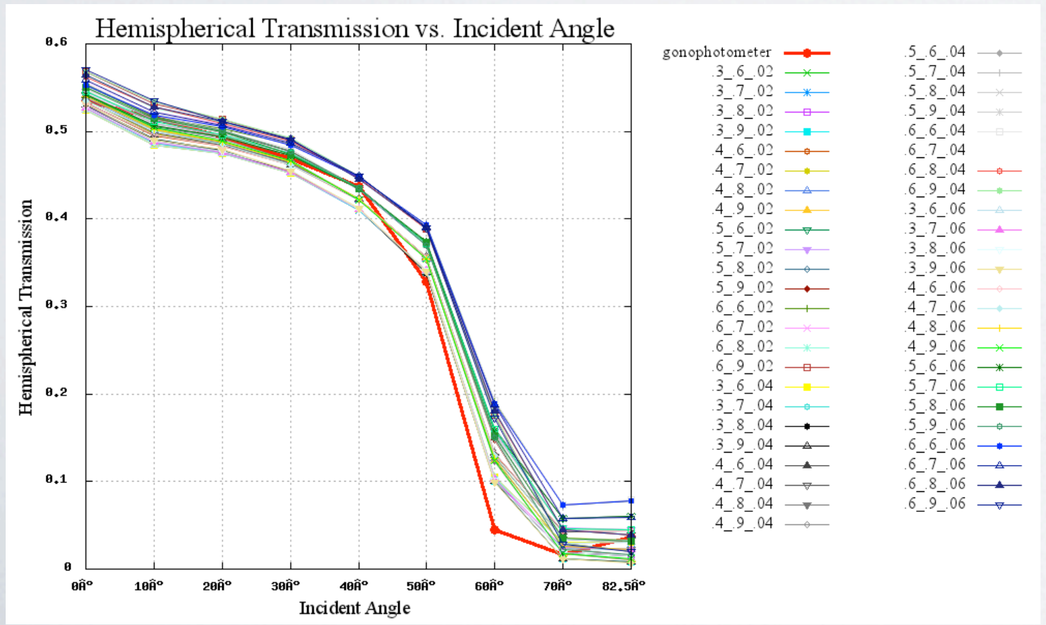
Expanded metal mesh

- Laser Scan to get geometry
- Parametric trial to fit gonio-photometer mesh measurement



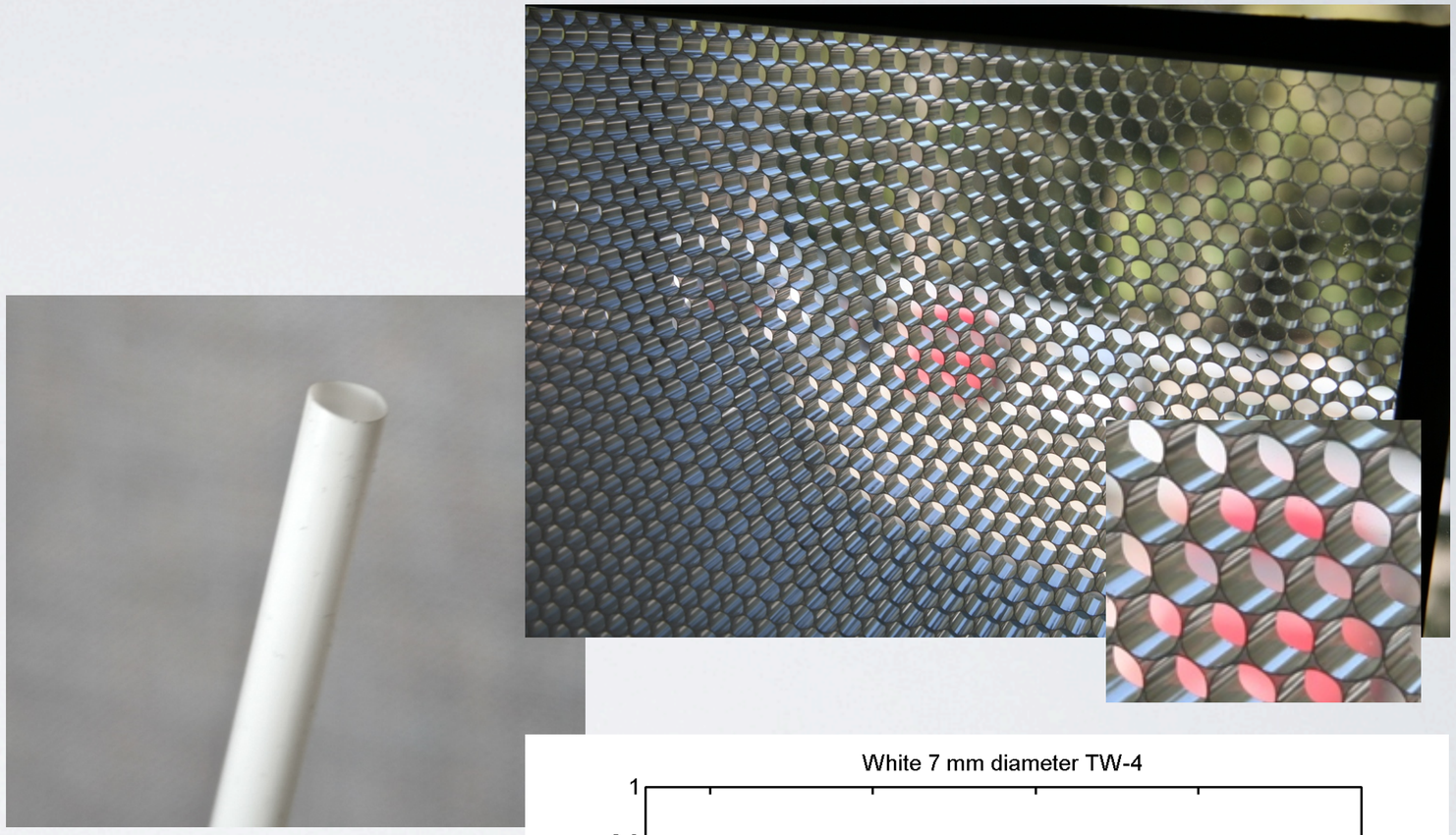


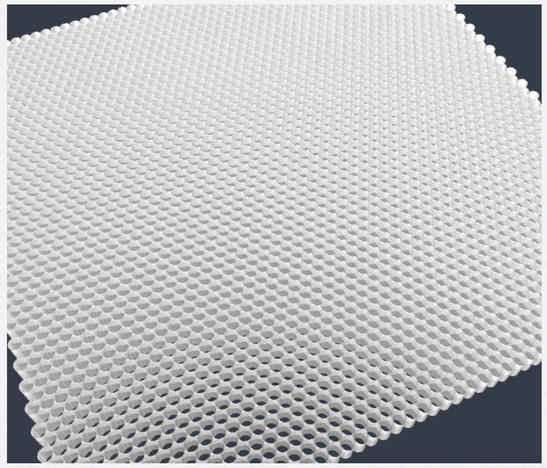


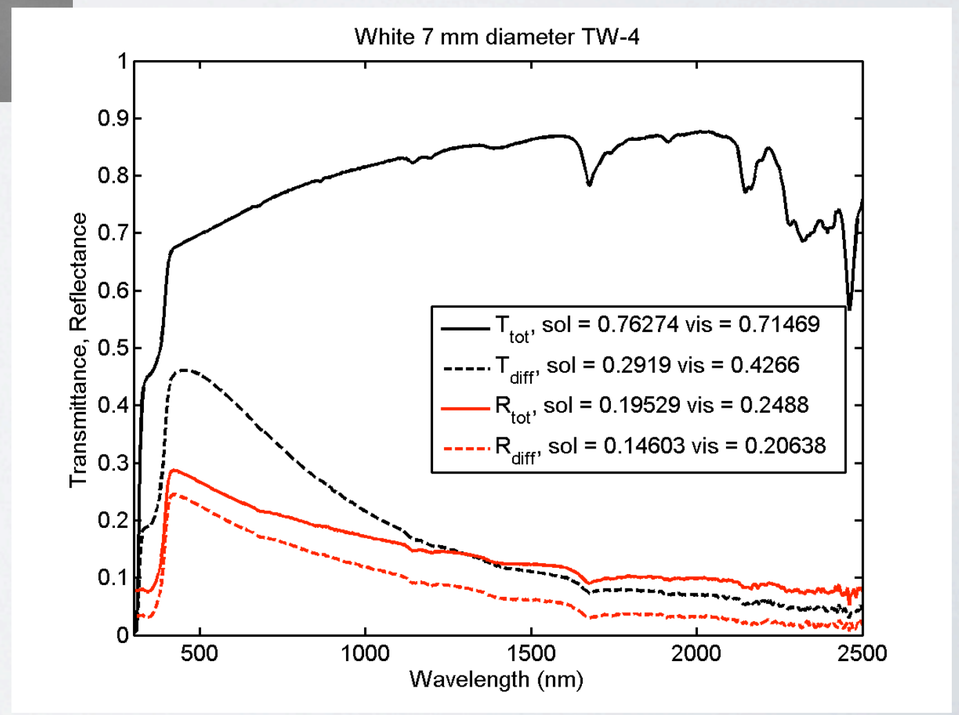


Columnar Structure

- Geometry from manufacturer
- Sample of plastic to measure transmission/ reflection properties
- genBSDF to generate BSDF of layer

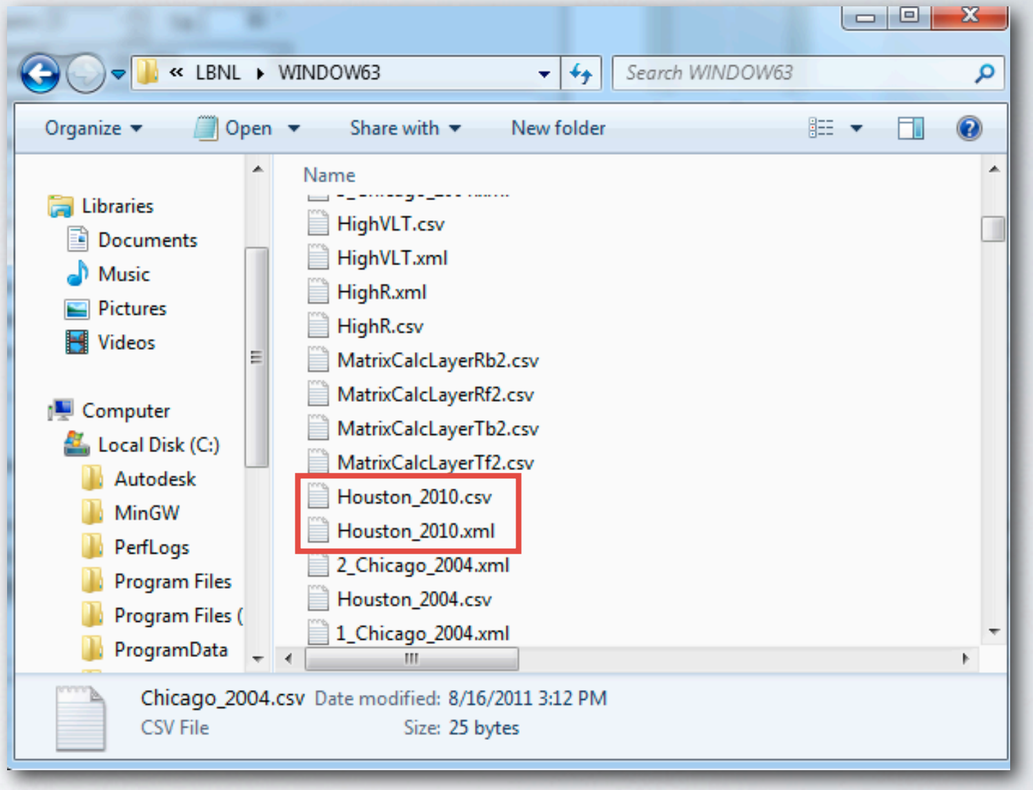




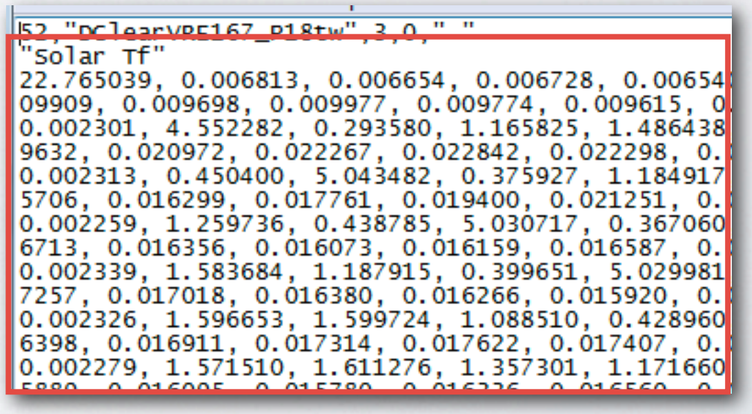


Combine BSDF with glazing

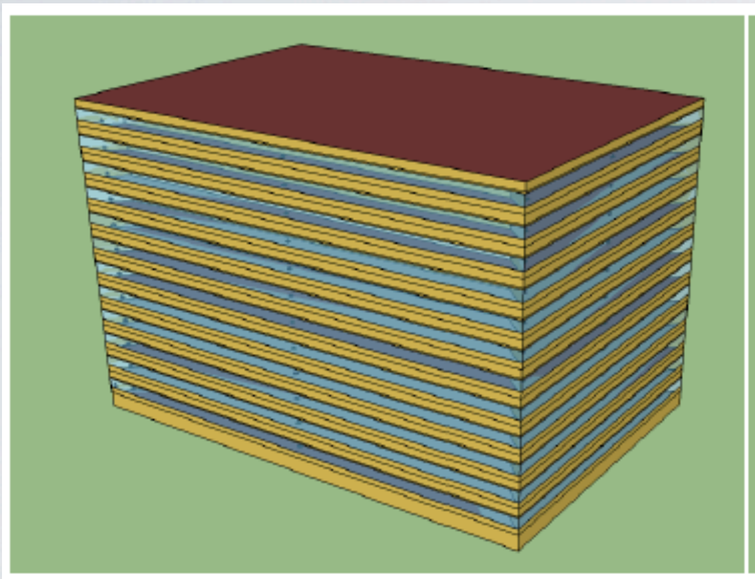
- Import into Window 6
- Add to IGU
- Calc
- Resulting XML only contains visible, use *.csv to get solar
- keep visible and solar BSDFs separate (use visible wavelength tag for both)

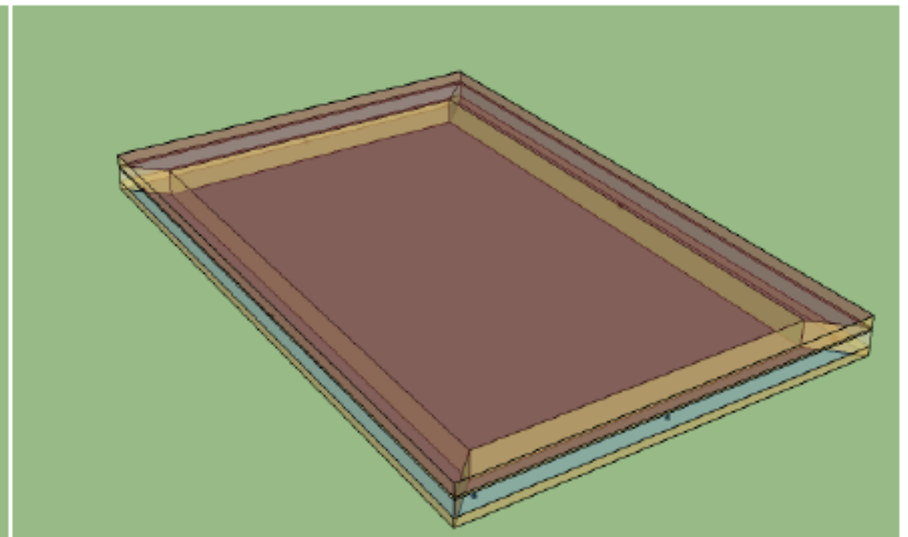


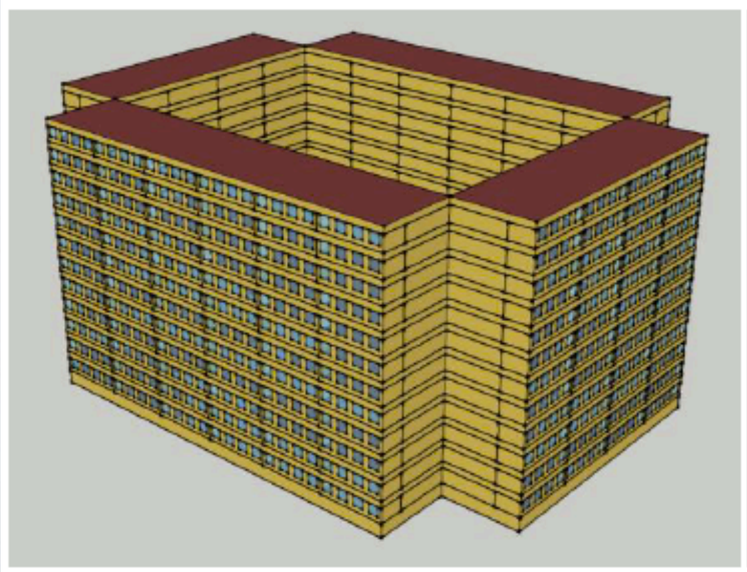
*.CSV

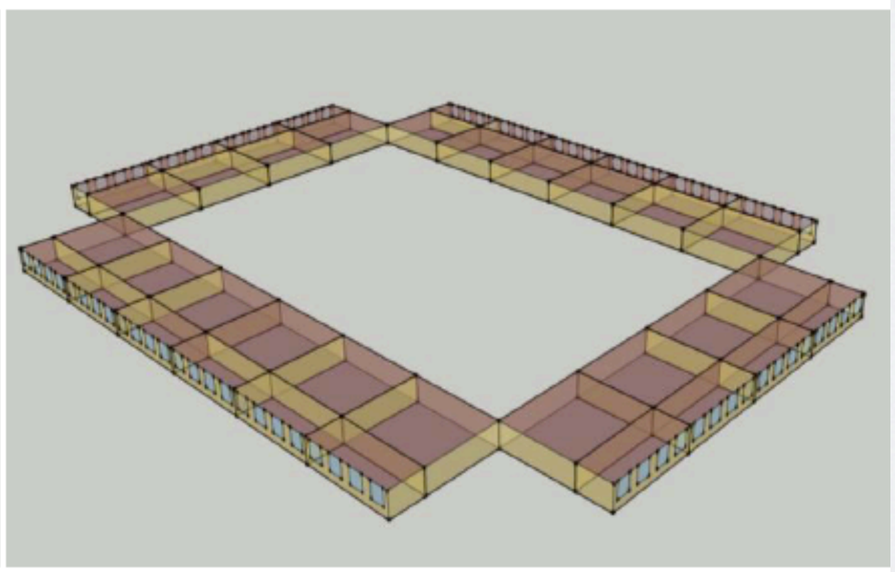


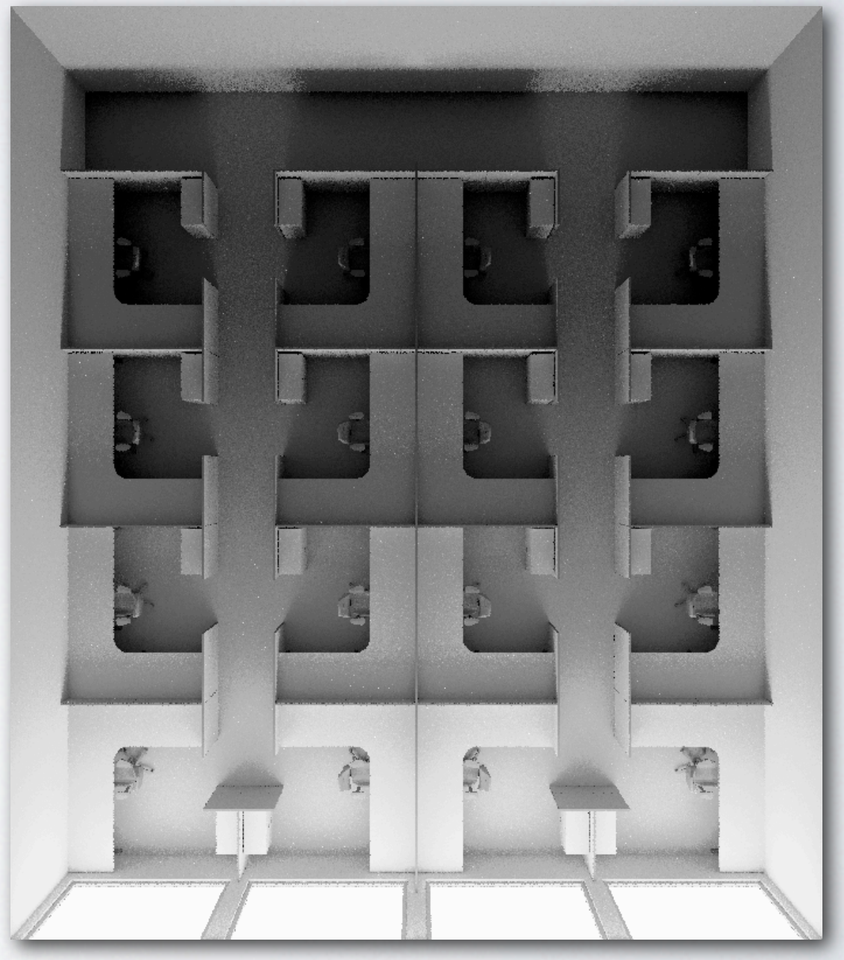
Model





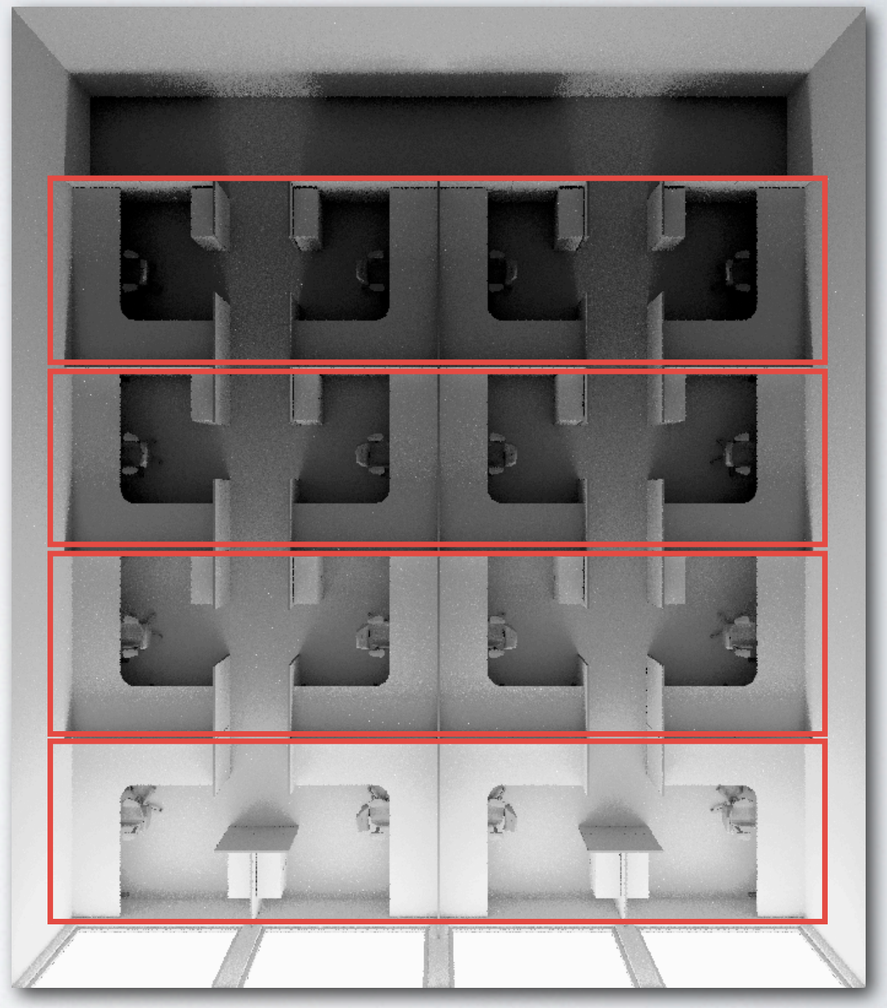






Creating Lighting Schedule

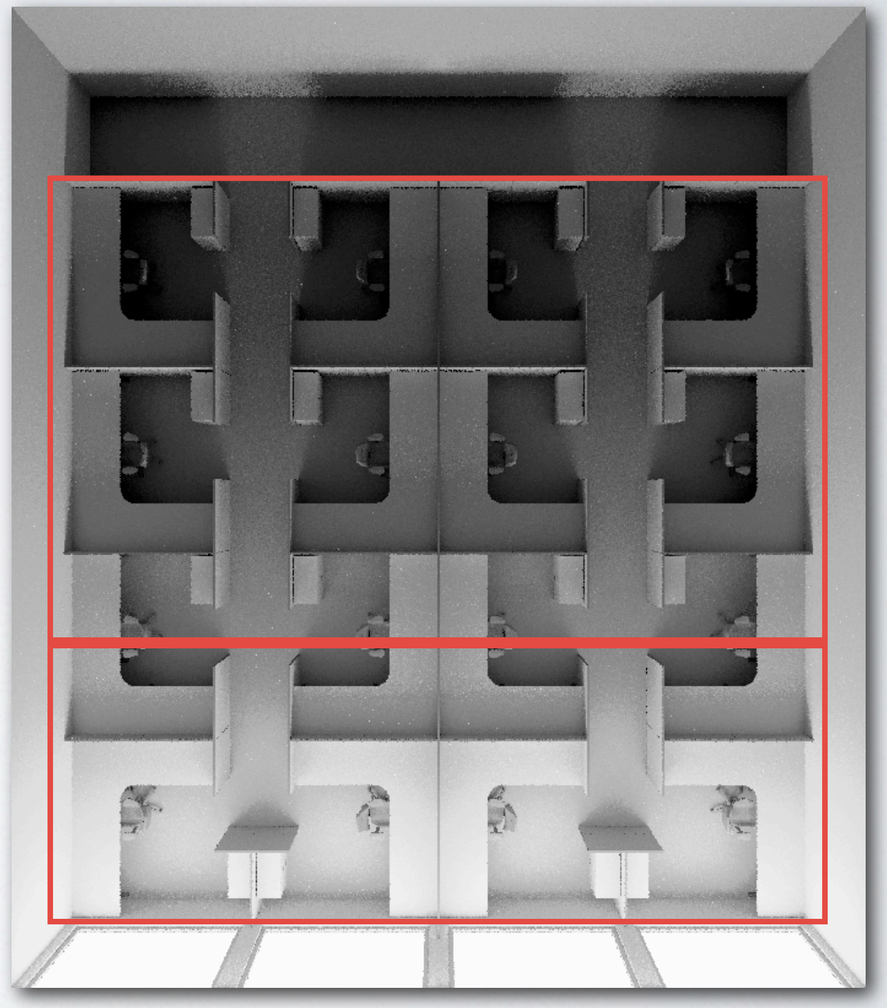
- Four 10’ zones
- Calculate daylight in each zone using 3-phase method
- Minimum illuminance in each zone is used to determine electric lighting power per zone.



Creating Surface Heat Gain Schedule

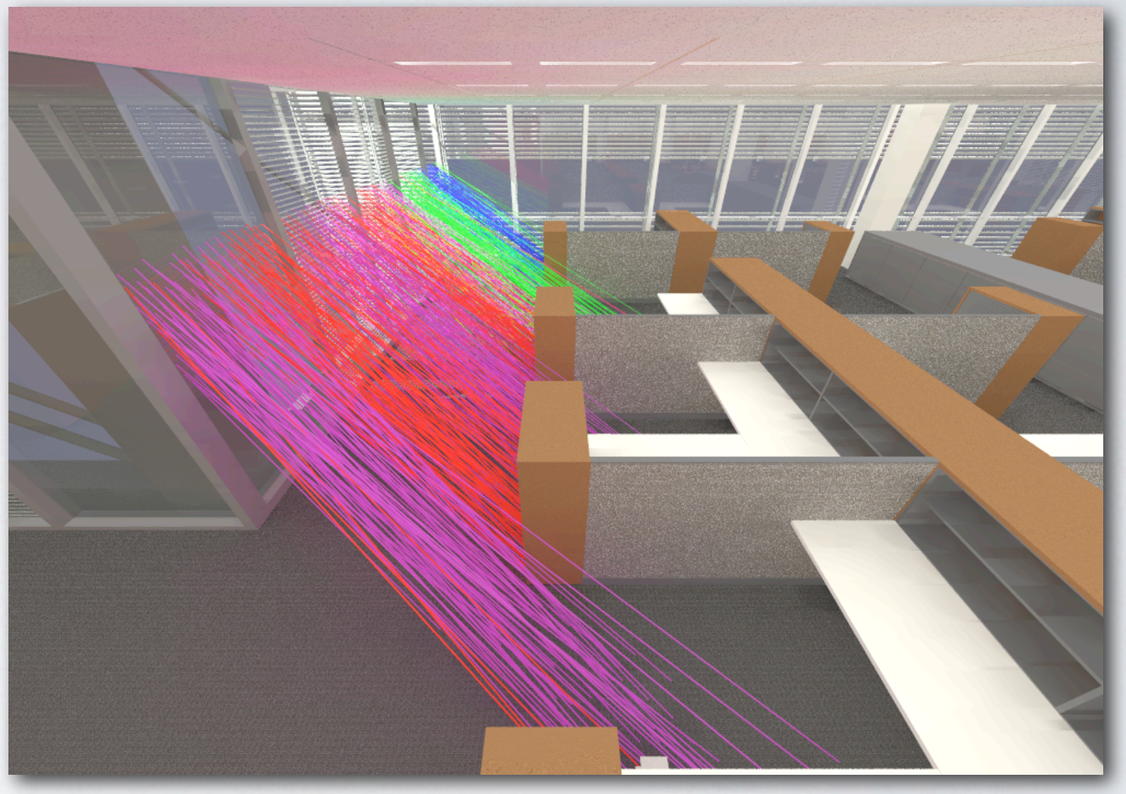
- 15’ perimeter, 25’ core zones

- Calculate heat gain on surfaces (ceiling, wall, floor furniture) in each zone



Modified View Matrix

- genklemsamp used to generate interior sample rays (with coordinate mods)
- trace rays (rtrace) returning intersection point.
- intersection points are parsed to determine zone and surface
- ray fractions accumulated, multiplied by cos(theta)/solid angle of the outgoing patch.



(Visualization from a different but similar project)
Ray color indicates zone & surface:

- Purple= floor, zone 2
- Red = furniture, zone 2
- Green = floor, zone 1
- Blue = floor, zone 1

Creating Surface Heat Gain Schedule

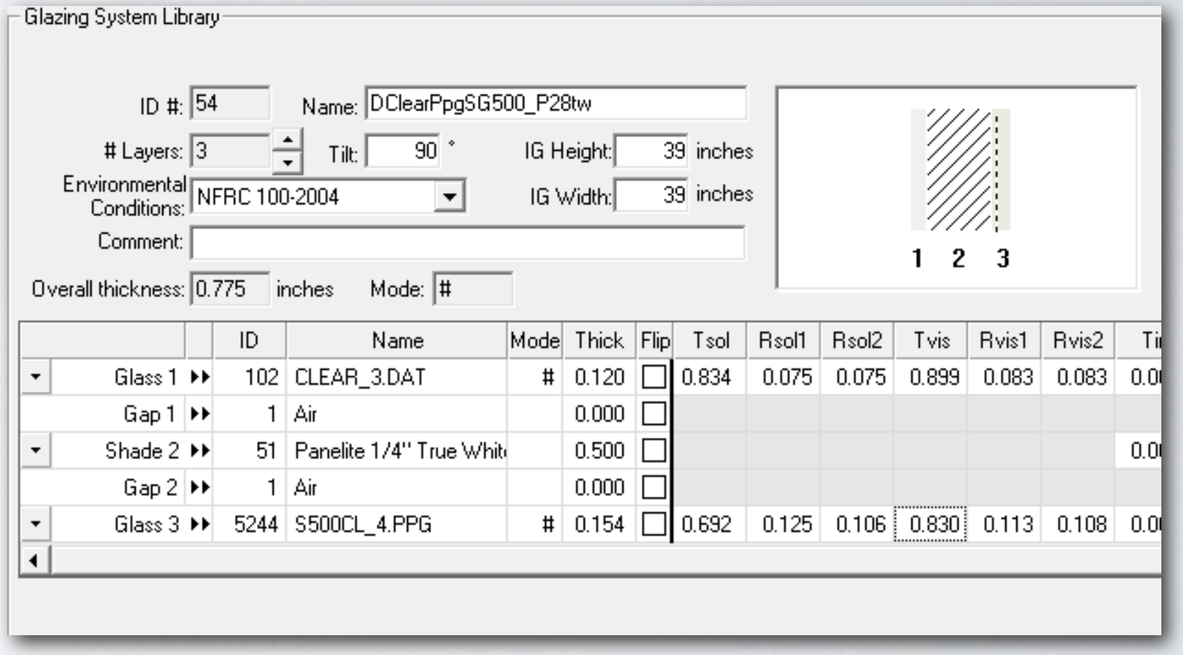
- gendaylit -O 1 (gensky if gendaylit fails)
- calculate vertical irradiance at facade
- genskyvec | dctimestep (using solar data)
- multiply by window area.

Window Layer Absorption

- Absorbed fractions per klems patch from Window 6

- Create view matrix for absorption

- gendaylit | dctimestep | rcalc like previous



Accelerating dctimestep

- This takes several hours:

```
for hour in 8760
   gendaylit | genskyvec | dctimestep
done
```

- When running parametric analysis we need it to be faster.

Accelerating dctimestep

- Problem 1: We are recalculating the same sky vectors for multiple runs in a climate

- Solution: store and zip sky vectors. Allows quick retrieval for subsequent runs using same climate

- Reduces time from >2 hours to 7 minutes.

Accelerating dctimestep

- Problem 2: We load matrices into memory and perform the same multiplication several times (V*T*D - only s changes).

- Wangda Zuo created an version of dctimestep optimized for annual runs (ibpsa 2011 conference).

  Usage:

  dctimestepcpu **V T D S**

  (sky matrix instead of vector)

Accelerating dctimestep

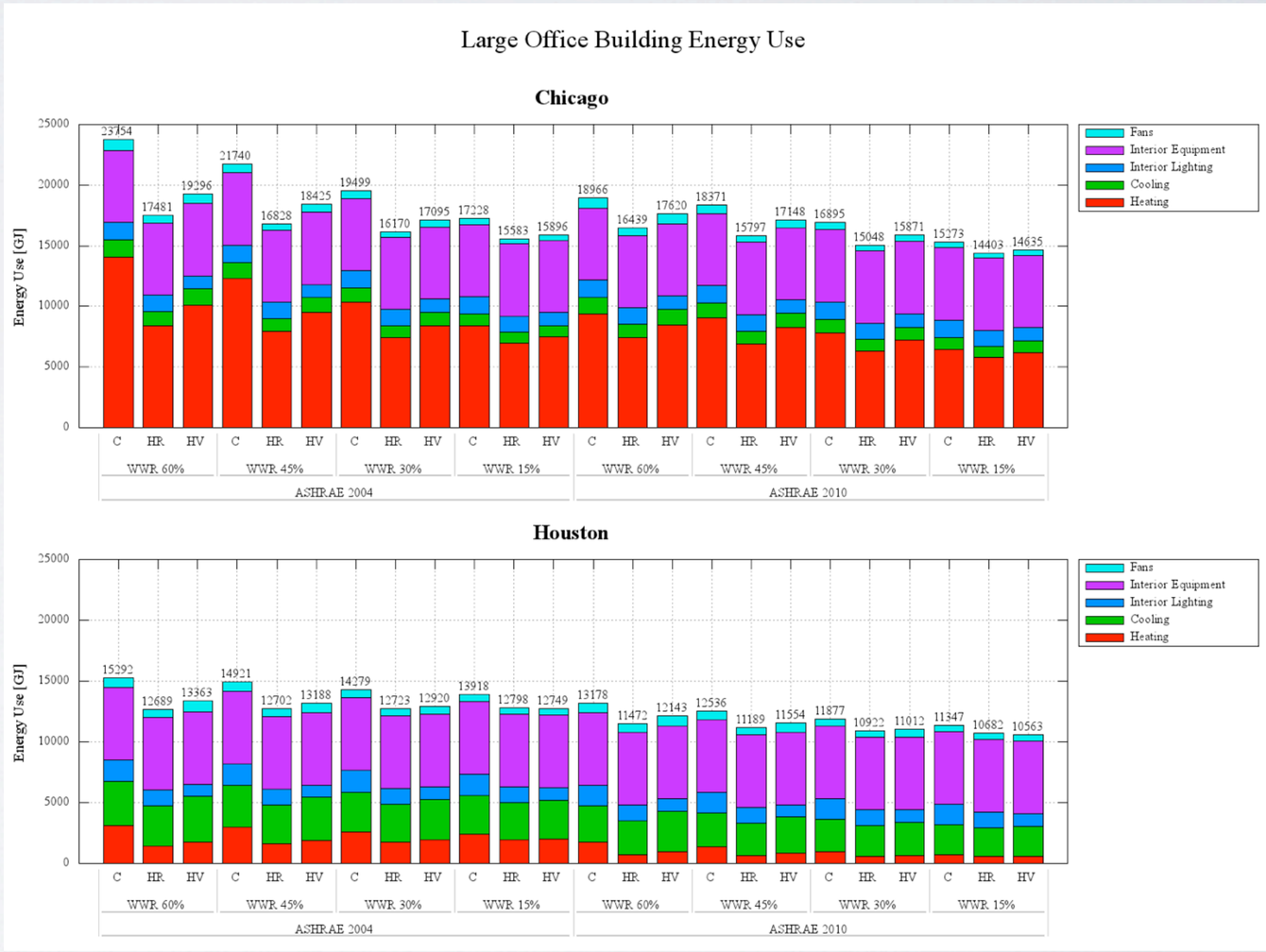
- One call for the entire year (matricies loaded once, multiplied once).
- RGB channels parallelized
- Runtime 30 seconds
- Energy Plus is now the slow part of the simulation!
  (if you ignore days of matrix pre-calculation :-)

Using Schedules in Energy Plus

- Lighting schedule capability exists.
- Energy Plus code modified to allow scheduling of surface heat gains and window layer absorption schedules
    - modification and validation (in progress) by Thierry Nouidui and Brian Coffey

runenergyplus !

- No real results yet, still running simulations.



Acknowledgements

- Jacob Jonsson & David Appelfeld - BSDF generation
- Christian Kohler - Window layer absorbed fraction (and other W6 stuff)
- Robert Hart - Window glazing makeup
- Wangda Zuo - dctimestep optimization
- Thierry Nouidui and Brian Coffey - Energy Plus modifications
- Xiufeng Pang - Energy Plus model
- Luis Fernandes - Comfen model
- Eleanor Lee - Principal Investigator
- CEC Pier & DOE - Funding
- Me!