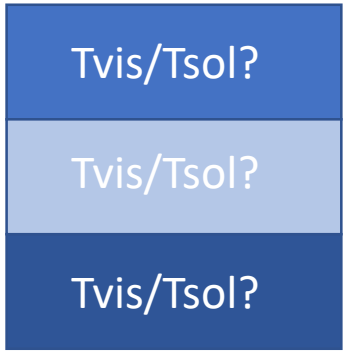
Workflow control and optimization for Radiance matrix algebraic methods

Taoning Wang, Eleanor Lee Lawrence Berkeley Lab
Greg Ward, Anyhere Software

Background

- A series of python scripts were developed to facilitate modeling the predictive control of dynamic façade systems
- As users of the LBNL tools, we had spent a lot of time finding our own errors
- Matrix basis/resolution sensitivity analysis are needed to provide insights
- Some simple operations can be (semi) - automated to reduce error

3-zone electrochromic window



+ Temp?

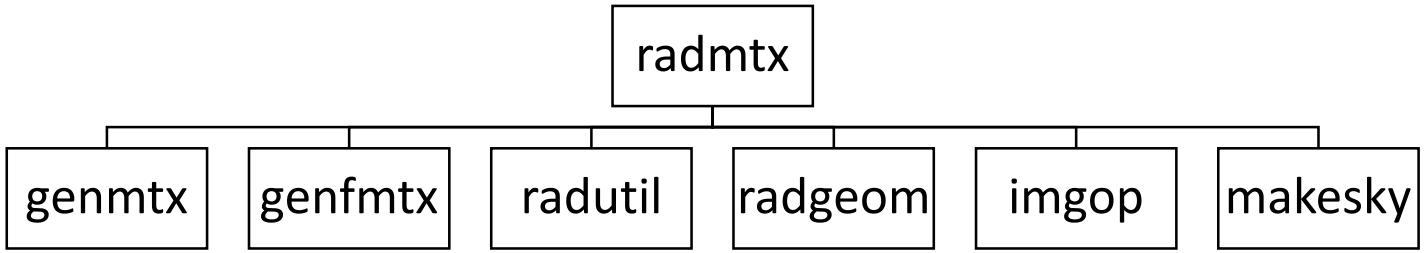
Minimize $
Subject to:
WP Illuminance > xxx
Glare(DGP) < xxx
Xxx < Temp < xxx

Objective: Develop tools to facilitate:

- annual simulation using matrix methods
- Model based façade control design

Key features:

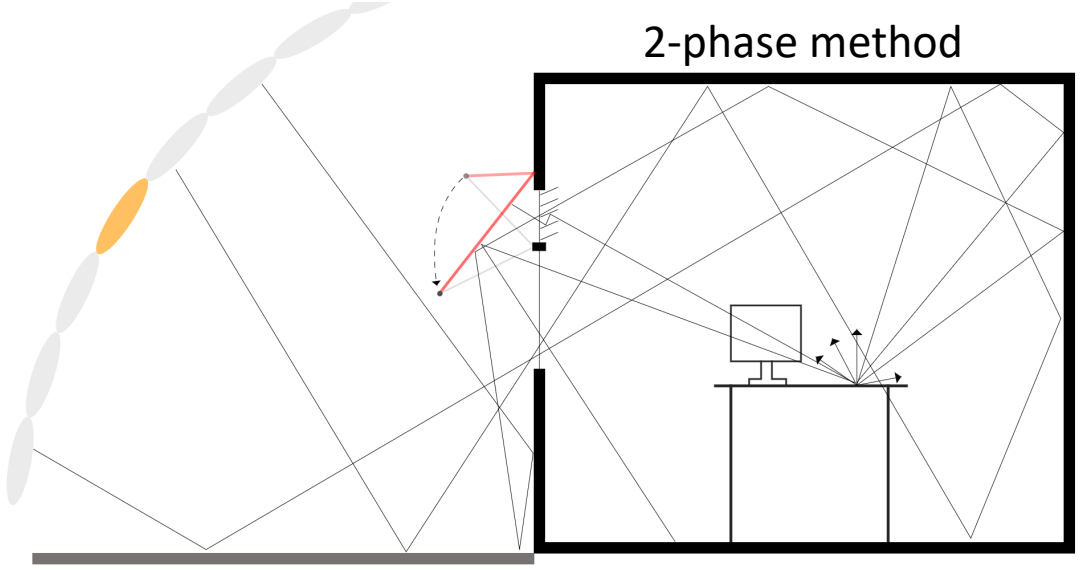
- Executive program **radmtx**, like Radiance **rad** program.
- Unix-toolbox model: command line friendly <, |, >
- Object oriented: **Genmtx** (sender, receiver, environmental, option)
- Run: **radmtx** setup.cfg



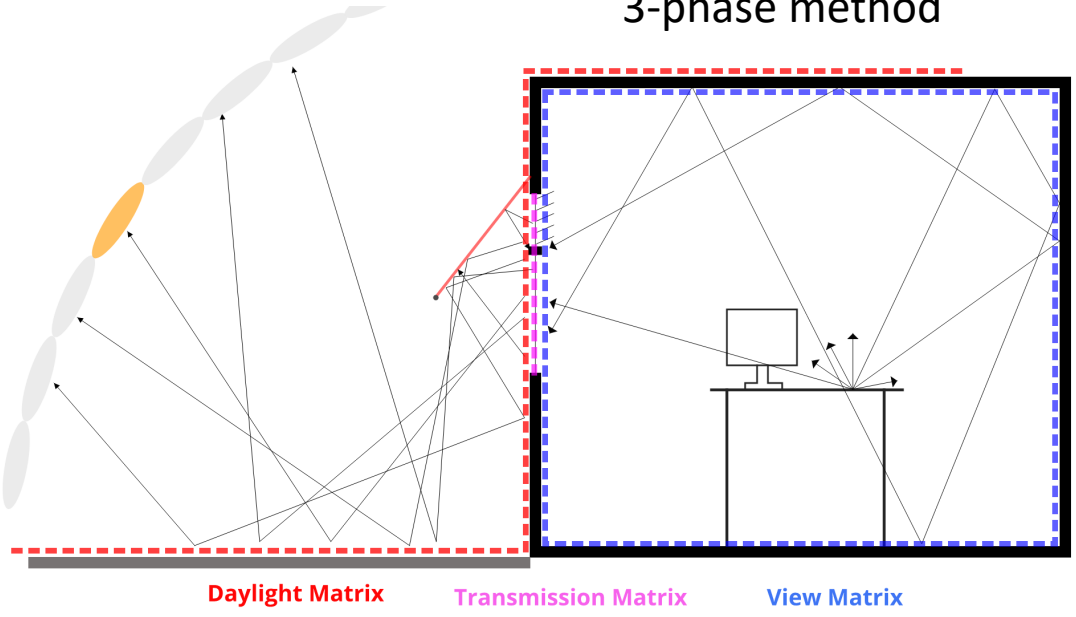
Matrix methods enable:

- efficient annual simulation
- building components parametric design
- façade operation modeling

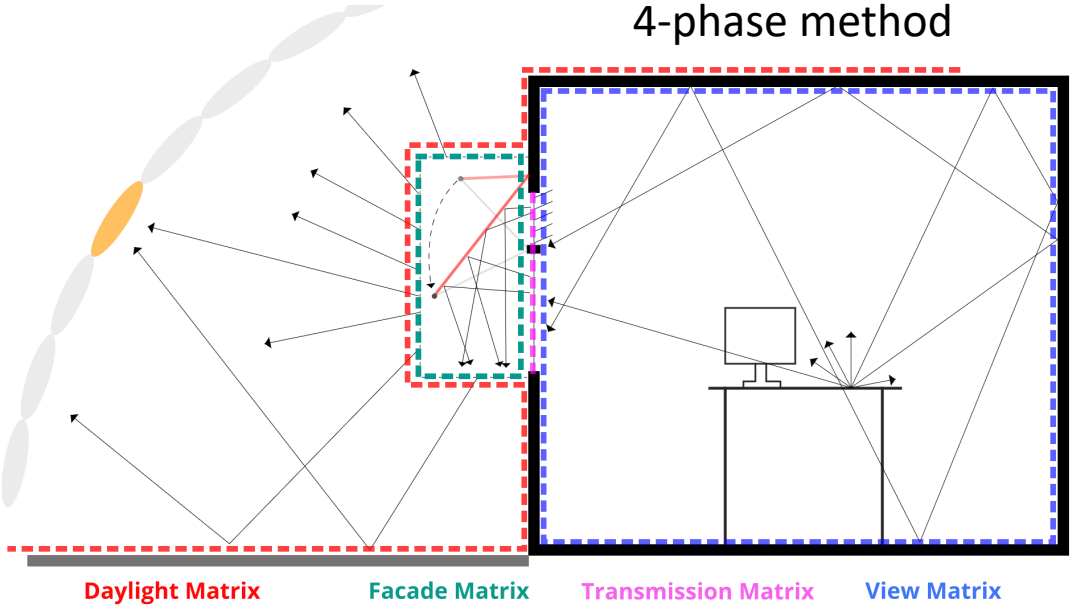
2-phase method



3-phase method



4-phase method



Matrix methods are, at the same time, **complicated** to implement.

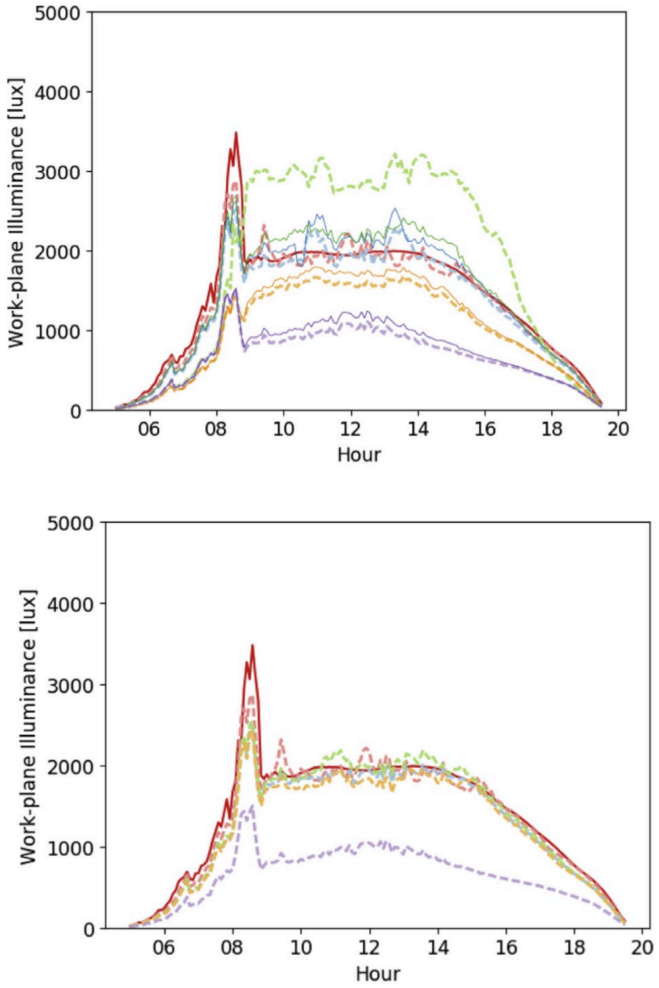
Question that needs to be answered:

1. Which method is best suited for my application?
2. Which matrix resolution/basis to use? Klems? Tregenza? X2? X4?
3. How to set my simulation parameters for my matrices?
4. The result looks plausible, but did I make a mistake somewhere?

There are also a lot of tripping points, such as:

1. Surfaces and BSDF orientation
2. Window grouping and window subdivision

Workplane illuminance



Façade and view matrices generated using Klems vs Reinhart 2 basis

If parametric analysis of coplanar systems or modeling its operation

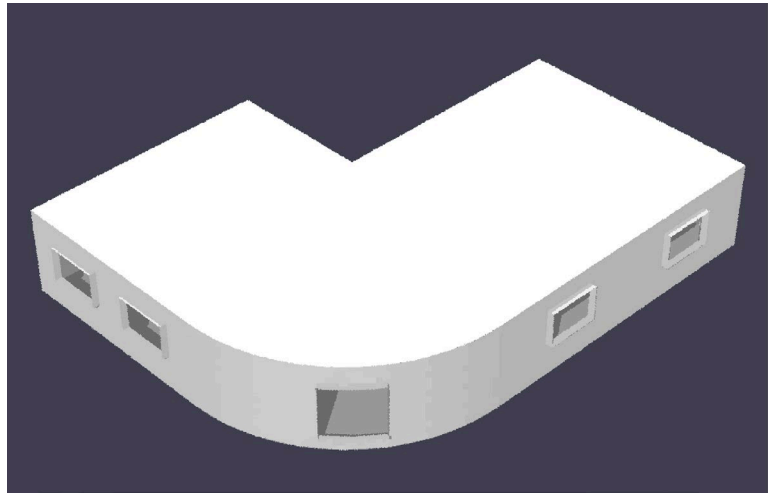
→ 3-phase method (5-phase if direct calculation needed)

else if parametric analysis of non-coplanar or modeling its operation

→ 4-phase method (6-phase if direct calculation needed)

else

→ 2-phase methods



Floor.rad
Ceiling.rad
Walls.rad
Window1.rad
Window2.rad

Example
Python:

Genmtx(sender=grid.pts, receiver=window.rad, out=vmx, env=[‘material.rad’,’floor.rad’,…])

Command-line:

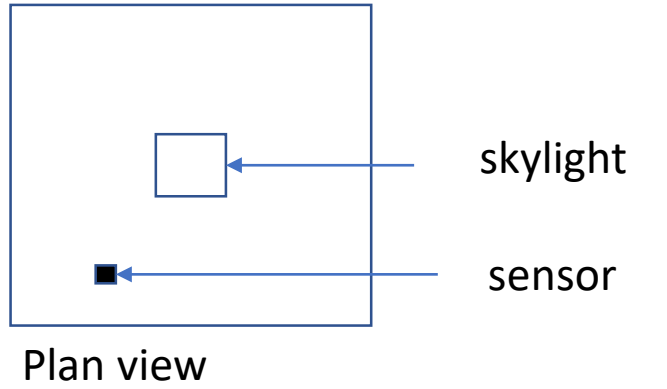
Genmtx –s grid.pts –r window.rad –rs kf –env material.rad floor.rad ceiling.rad walls.rad –o vmx

Matrix sensitivity

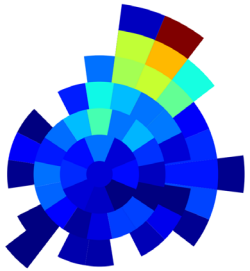
- Which basis to use?
- Which resolution is sufficient?
- What parameters to use?
- In progress

Flux matrix visualization

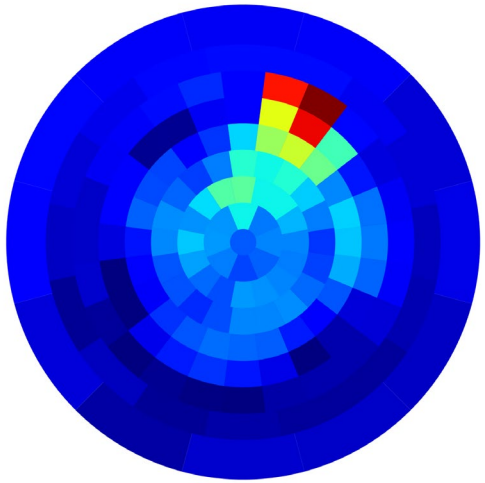
vecviz.py view.mtx -a kf



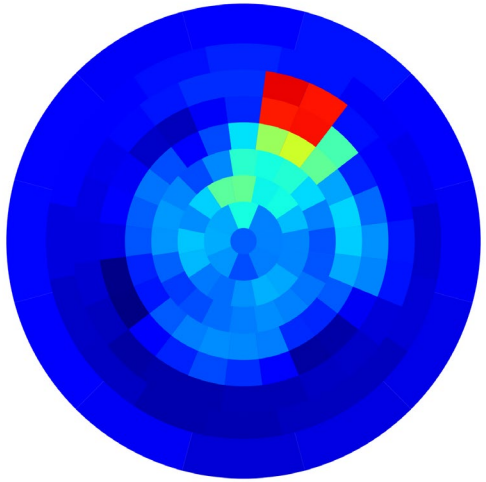
Plan view



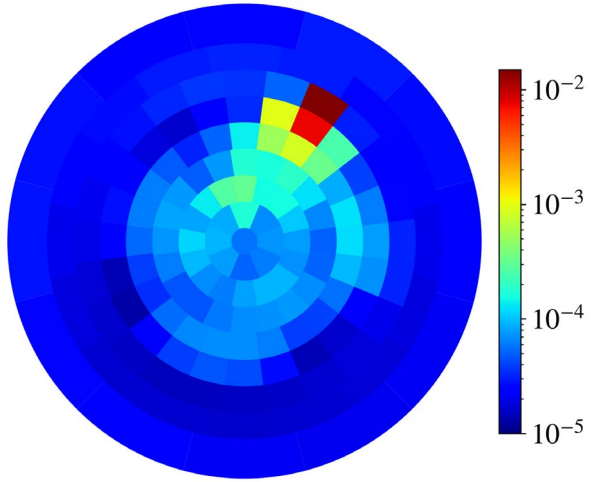
Ab 1
Ad 5000
Lw 1e-4
Time 0.05s



Ab 3
Ad 5000
Lw 1e-4
Time 1s



Ab 4
Ad 10000
Lw 1e-6
Time 3s



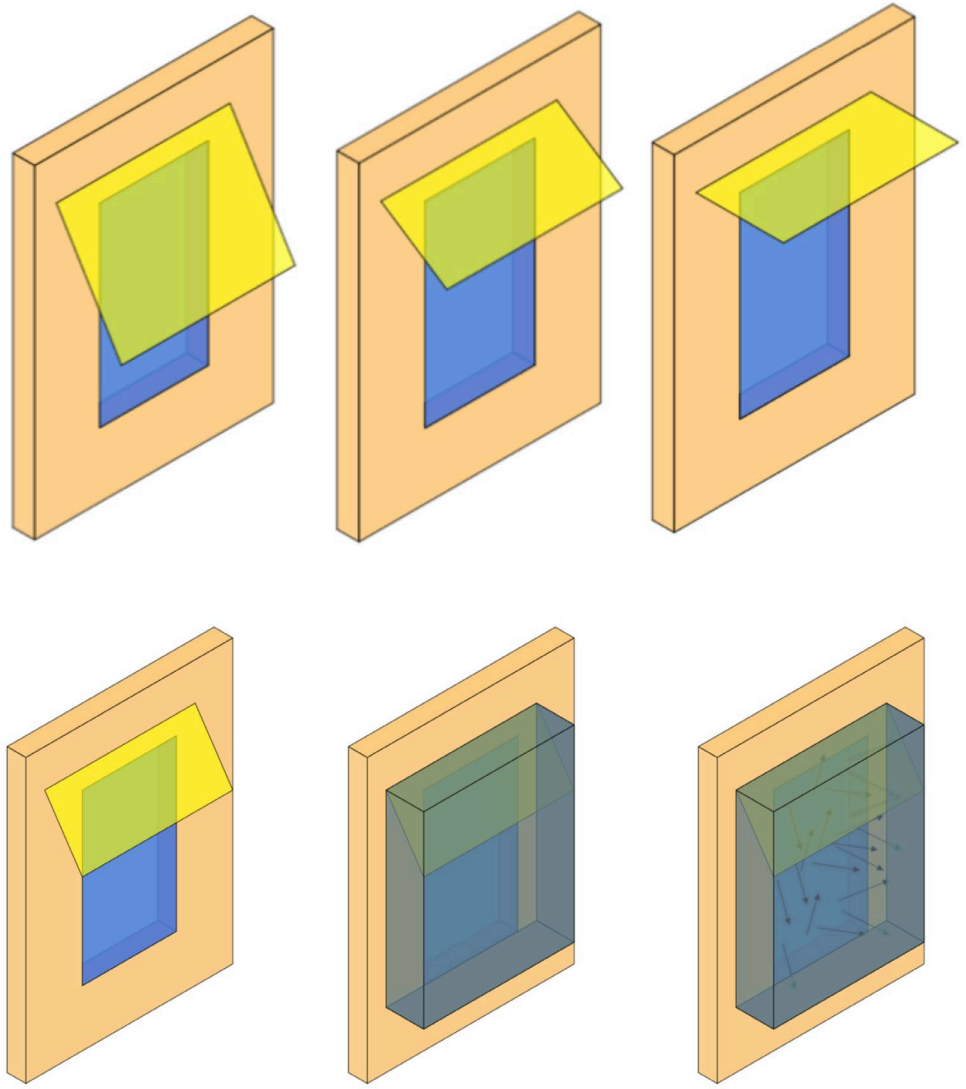
Ab 8
Ad 100000
Lw 1e-9
Time 75s

Low-settings -> high settings

Façade matrix generation

Automation steps:

1. Rotate window and non-coplanar geometry around Z axis
2. Rotate until the projected area onto XY plane is the smallest
3. Generate the bounding box
4. Rotate everything back to original state
5. If façade not near-vertical, use the alternative methods of extruding enlarged window surface to encompass the non-coplanar shading geometry

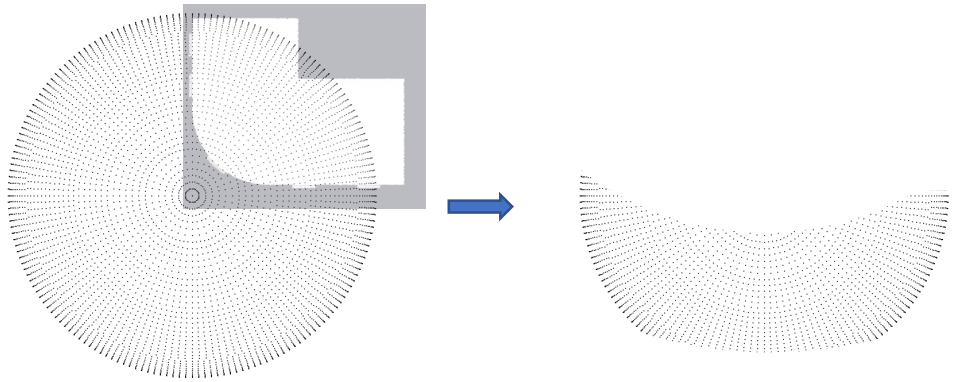


Minor speed up

- Sun culling: reduce the number of sun by window orientation and weather file

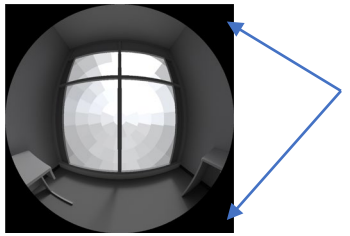
Climate: rmtxop –ff –c .3 .6 .1 –t oakland.smx | getinfo - | total –if5186 –t,
test value greater than zero
Windows: Test window normal to sun directions

Results:
Void light sol1 0 0 0
Sol1 source sun 0 0 4 0.2 0.3 0.4 .533
Void light sol2 1 1 1
Sol2 source sun 0 0 4 0.12 0.23 0.234 .533



- Fisheye circle cropping for glare evaluation

Before:
vwrays –ff –x 1000 –y 1000 –vf view.vf –c 9 –pj .7 | rfluxmtx



Flushed sample rays

After:
vwrays –ff –x 1000 –y 1000 –vf view.vf –c 9 –pj .7 | rcalc –if 6 –of –e “DIM:1000;CNT:9” –e “pn=(recno-1)/CNT+.5 –e “frac(x):x-floor(x)” –e “xpos=frac(pn/DIM);ypos=pn/(DIM*DIM)” –e “incir=if(.25-(xpos-.5)*(ypos-.5)*(ypos-.5),1,0)” –e “$1=$1;$2=$2;$3=$3;$4=$4*incir;$5=$5*incir;$6=$6*incir” | rfluxmtx

Simulation configuration: setup.cfg

```
[simulation_control]
sim_quality = high
resolution_opt = on
dcmx_option = -ab 6 -ad 500000 -lw 1e-12
vmx_option =
fmx_option =
dmx_option =
nproc = 1
direct = off
matrix_only = off
group_window_by_orient = off

[file_structure]
matrices = Matrices
objects = Objects
octrees = Octrees
results = Results
raysenders = Raysenders
bsdf = Resouces/BSDF

[model] #these live inside 'objects' path
material = material.rad
opaque_surface = ceiling.rad floor.rad walls.rad

[window]
window1 = windows1.rad
window2 = windows2.rad

[climate]
smx =
wea =
epw =
lat = 37.7
lon = 122.2

[raysender]
grid =
view = south.vf

[sensor_surfaces]
surface = floor.rad
dist = 0.3
spacing = .3

[btdf]
vis = shade.xml shade2.xml shade3.xml
sol =
shgc =
mtx =
```

Simulation configuration: setup.cfg

[simulation_control]	comments
sim_quality	inputs to matrix sensitivity analyzer
resolution_opt	use matrix sensitivity analyzer to determine basis, resolution, and parameters
Dcmx_option	overwrite 2-phase methods matrix parameters
Vmx_option	overwrite view matrix parameters
Fmx_option	overwrite façade matrix parameters
Dmx_option	overwrite daylighting matrix parameters
View_ray_cnt	overwrite view ray count
Pixel_jitter	overwrite view ray count pixel jitter
nproc	number of processor to use
direct	separate direct and diffuse component
Matrix_only	generate matrices only
Group_window_by_orient	Group windows by their orientation

Simulation configuration: setup.cfg

[file_structure]	
matrices	Where matrices will be stored
objects	Objects files location include material and surfaces
octrees	Octree folder name
results	Results folder name
raysenders	Raysenders folder name
bsdf	BSDF file location

[model]	
material	Material description rad file
Opaque_surfaces	Rad file describing opaque surfaces in models
Non_coplanar_shade	Rad file(s) describing non-coplanar shades. If more than one, 4-phase method triggered.

[window]	
window	Rad file describing window polygons
window	Rad file describing window polygons

Simulation configuration: setup.cfg

[Climate]	
smx	Annual sky matrix file path. If provided below entries ignored
wea	Wea weather file path. If provided, below entries ignored.
epw	Epw file path. If provided, below entries ignored.
latitude	site latitude, used to download an epw file
longitude	Site longitude, used to download an epw file

[Raysenders]	
Grid_file	Sensor grid file path
View_file	View file (.vf) path

[sensor_surfaces]	
Surface_file	Sensor grid file path, if no grid file provided, this will be used to generate sensor grids
distance	Sensors will be places distance away from surface in its normal direction
spacing	Distance between the sensors.

Future work

- Matrix sensitivity analysis
- Adding window functionality with WINCALC https://github.com/LBNL-ETA/WinCalc
- Database integration for glazing and shading systems (BSDFs), IGSDB

Thank you and questions?

Acknowledgments

Dustin Davis, California Energy Commission
Amir Roth, U.S. Department of Energy
Marc Lafrance, U.S. Department of Energy

Jacob Johnson, Christoph Gehbauer, Anothai Thanachareonkit, Daniel Fuller, LBNL

https://facades.lbl.gov