Integration of Urban Microclimate Models using the QUIC EnvSim GPU Framework

Peter Willemsen, Matthew Overby, Kevin Briggs,
Brian N. Bailey, Daniel Alexander, Rob Stoll, Eric R. Pardyjak

Department of Computer Science
University of Minnesota Duluth, Duluth, MN, USA

Department of Mechanical Engineering
University of Utah, Salt Lake City, UT, USA

ICUC9 July 24, 2015

UMD
University of Minnesota Duluth
Driven to Discover

Department of MECHANICAL ENGINEERING
THE UNIVERSITY OF UTAH

This material is based upon work supported by the National Science Foundation under Grants No. 1133590, 0828206. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Talk Overview

Objective of this Talk

- Show use of our microclimate simulation (QES) for computing 3D environmental scalars in urban microclimates
- Demonstrate dynamic nature of QES framework for tightly coupling simulation specific models
- Encourage others to work with us and try out ideas with QES

Our Team

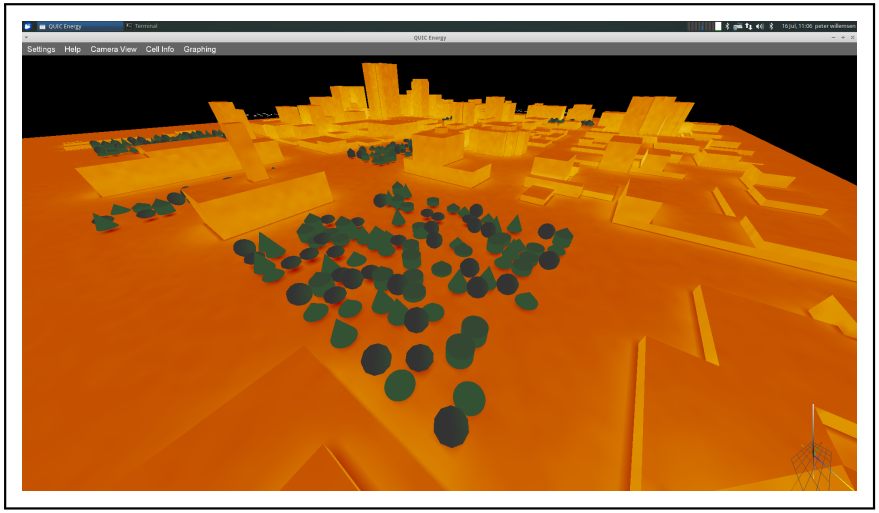
- Multidisciplinary - Meteorology, Computer Science, Engineering, and Urban Planning
- Focus is on microclimate simulation with coupling to mesoscale and optimization systems
- End goal is to provide researchers and scientists with effective, science-based tools

GEnUSiS Project - QUIC EnvSim - QES

This research is a part of the **Green Environmental Urban Simulations for Sustainability** (GEnUSiS)

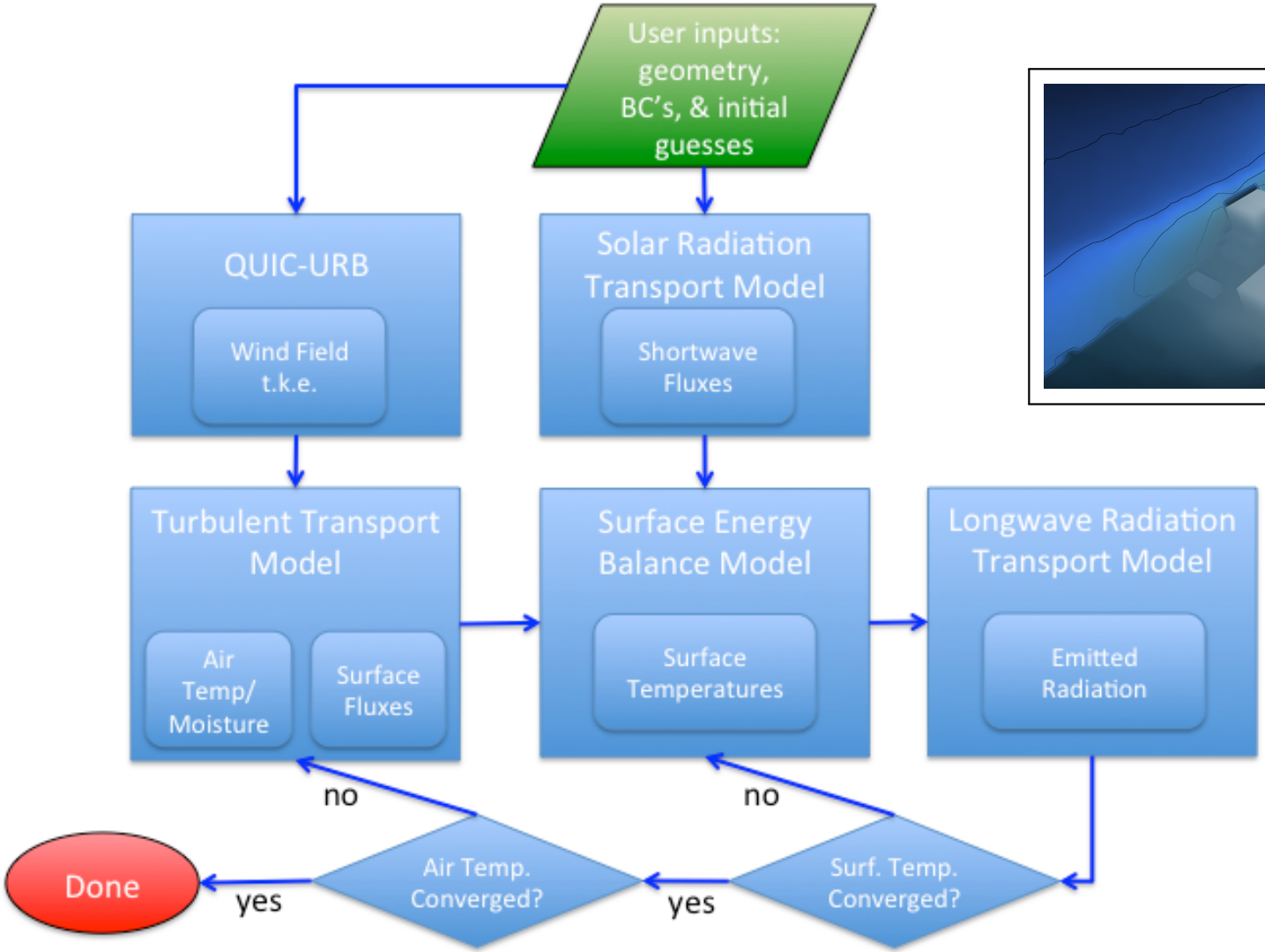
Use large-scale simulation science to investigate the impact of Green Infrastructure projects on urban energy use and microclimate.

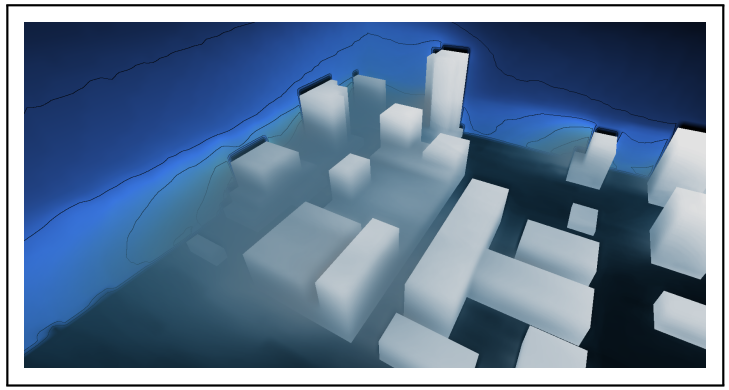
- Fast, efficient - Utilizes parallel processing
- Integrates 3D visualization and interaction systems
- Framework in C++



- Multiscale: tree and building resolving (2-5m) through city scale (1-2km)
- Turbulent Transport Model

QES - Coupled System for Computing Air and Surface Scalars



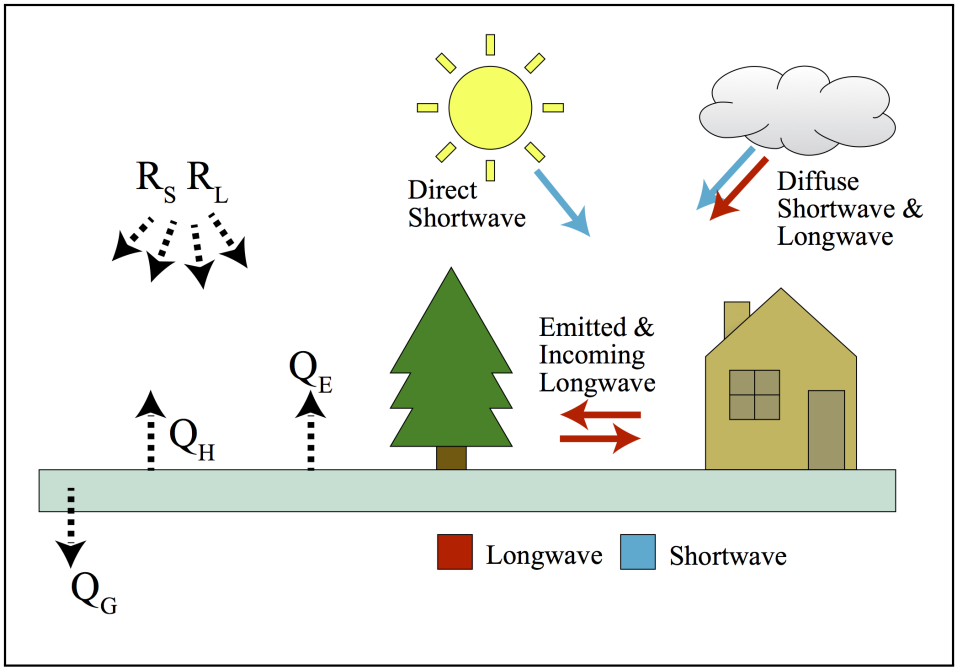


Computing Surface Energy Balance

- *Surface Energy Balance* computed across the domain
- Domain scale is arbitrary

$$Q^* + Q_h + Q_g + Q_e = 0$$

Q^*: Net Radiation
Q_h: Sensible Heat Flux
Q_g: Ground Heat Flux
Q_e: Latent Heat Flux

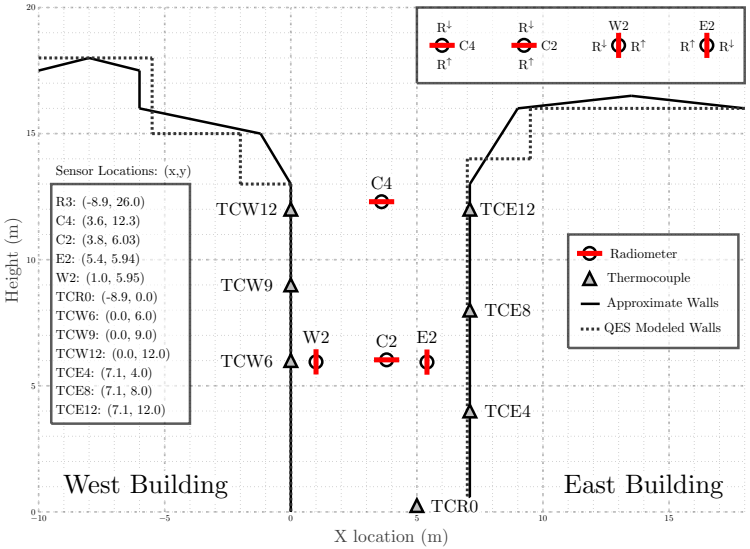


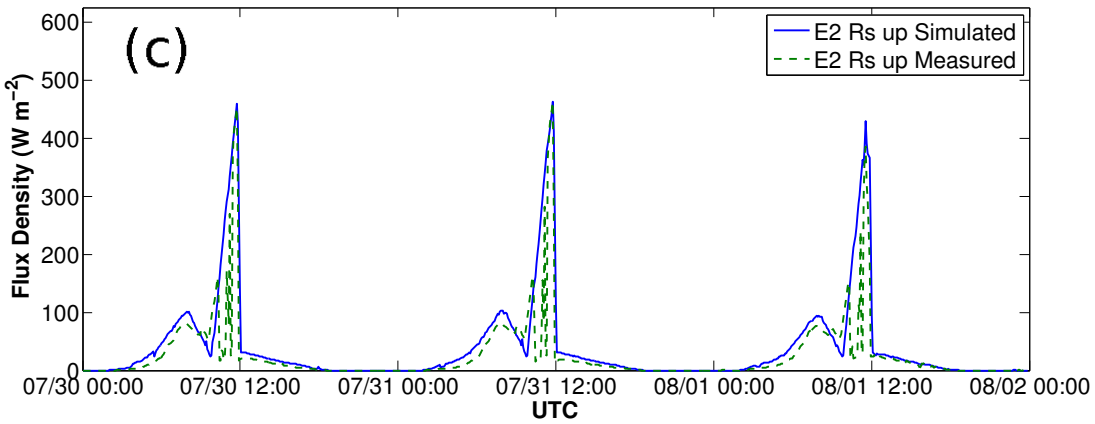
QES Radiation Model - Validation

Validation

M. Overby, P. Willemsen, D. Alexander, S. Halverson, and E.R. Pardyjak. Development of a Rapid Urban Energy Budget Modeling System using GPU Ray Tracing, *in submission to Urban Climate - April 2015*.

- Compares simulated SVF in Salt Lake City to Brown, Grimmond, & Ratti. *International Society of Environmental Hydraulics Conf.*, 2001.
- Compares radiative transport in urban canyon in Gothenburg, Sweden to Offerle et al., *Bound. Layer Meteor.*, 2007 and Eliasson et al., Atmos. Environ., 2006.





QES System Information

QESContext manages domain data:

- Provides an interface to the GPU
- Transfers data between CPU and GPU memory, often automatically
- Allows models to be developed to use either GPU or CPU resources
- Data organization promotes coupling of different models

Domain data stored in named buffers:

```
patch_fsky, patch_par, patch_nir, patch_longwave,
patch_temperature, patch_Qh, ...
```

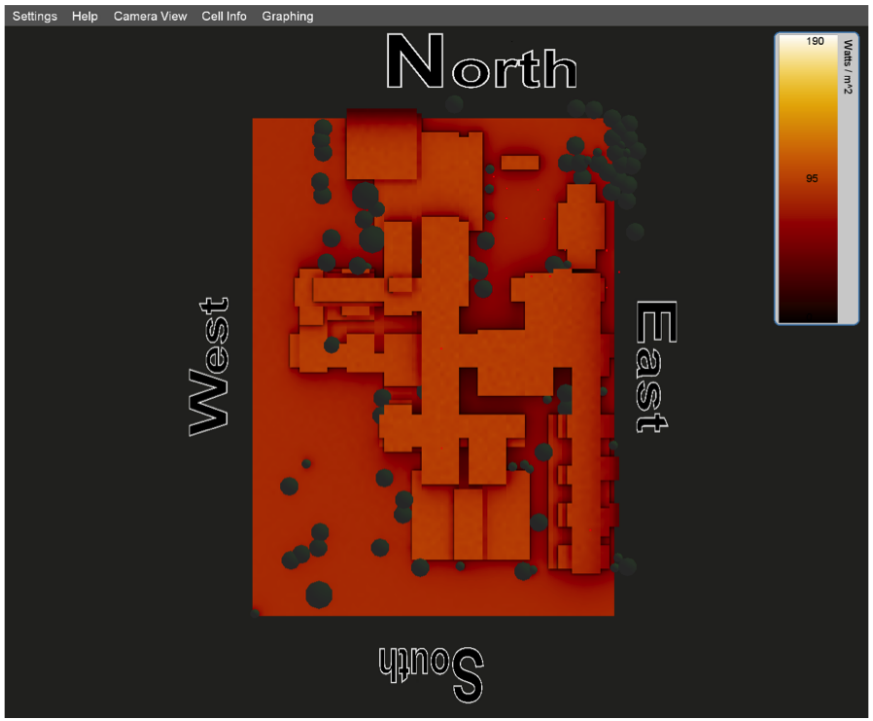
Modifying the code to be more accessible:

- EPFL Campus - Dan Nadeau, Laval University

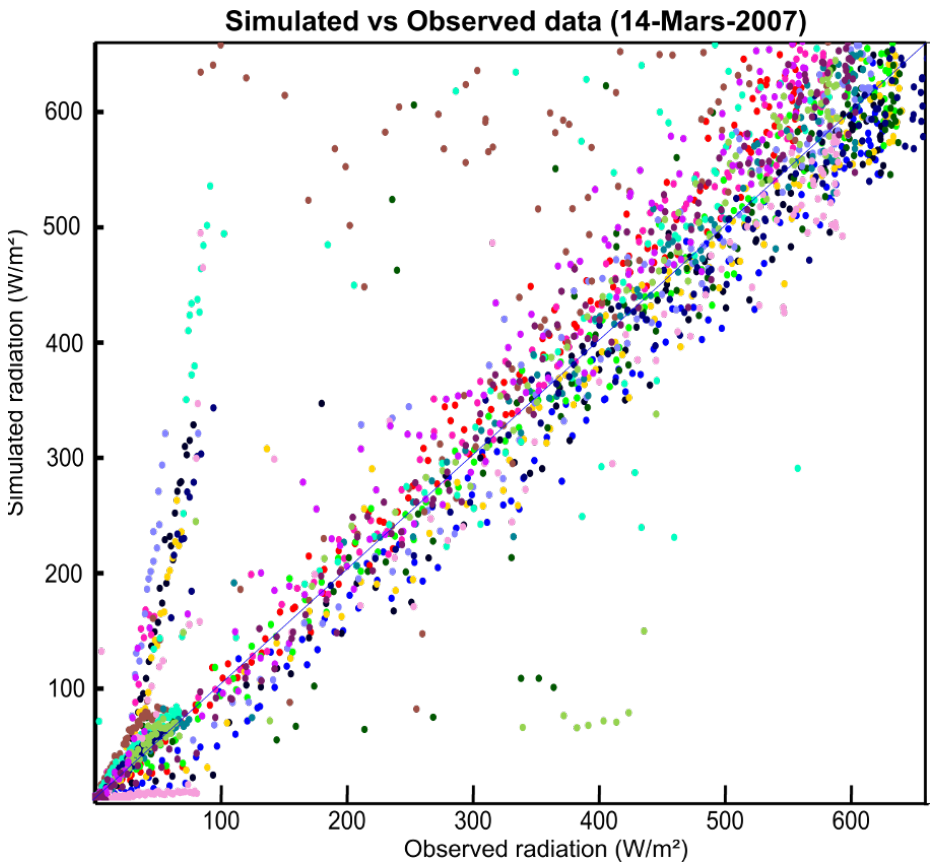
Recent Demonstration Outside of our Group

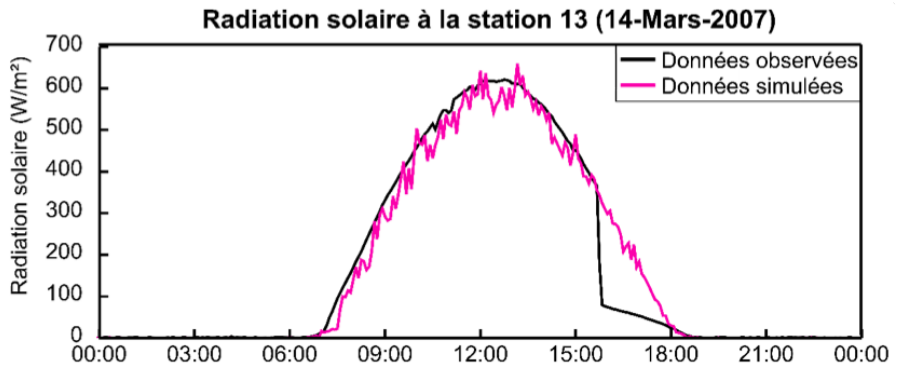
Girad P., Nadeau, D. et al. 2015

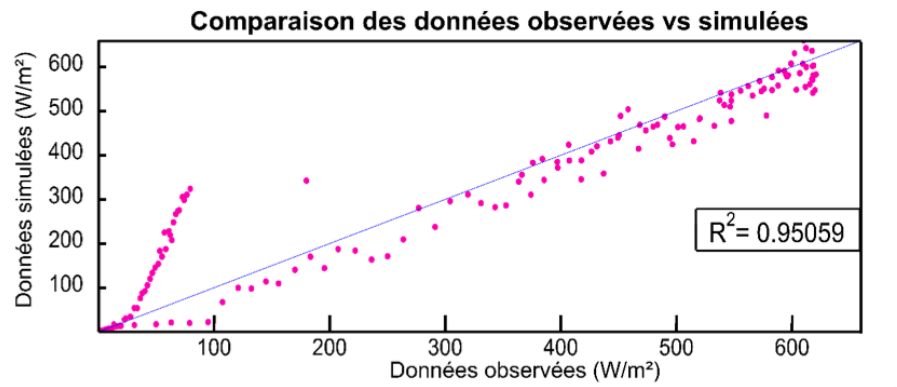
Girard P, et al. Performance des mesures de luttes aux lots de chaleur urbains l'chelle d'une quartier [Performance of urban heat island mitigation methods on a neighborhood scale]. Presented at: RHQ2015; 2015 June 9-10; Montral, Quebec. Canada

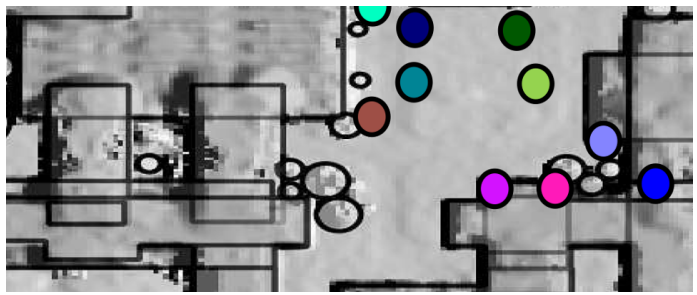


Girard, P., Nadeau, D. et al. 2015









Emphasizes the importance of detailed geometry!

QES Sky View Factor (SVF) Example

Computing SVF with QES involves the following code:

```
qes::QESContext context( 1 );

qes::ViewTracer vTracer;
context.joinModel( &vTracer );

loadScene( &context );

context.initialize();
context.runSimulation();

g_buffTracker->getBuffer<float>( "patch_fsky", &patch_fsky );
```

This *is* the code (minus error checking, scene loading, and data output).

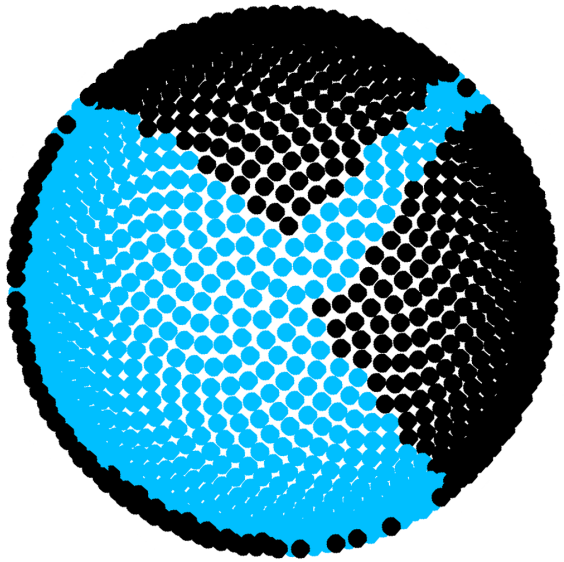
How is SVF computed?

The code:

```
context.runSimulation();
```

For all domain discretized patches:

- Send a hemisphere of rays from every patch and record if any object was intersected

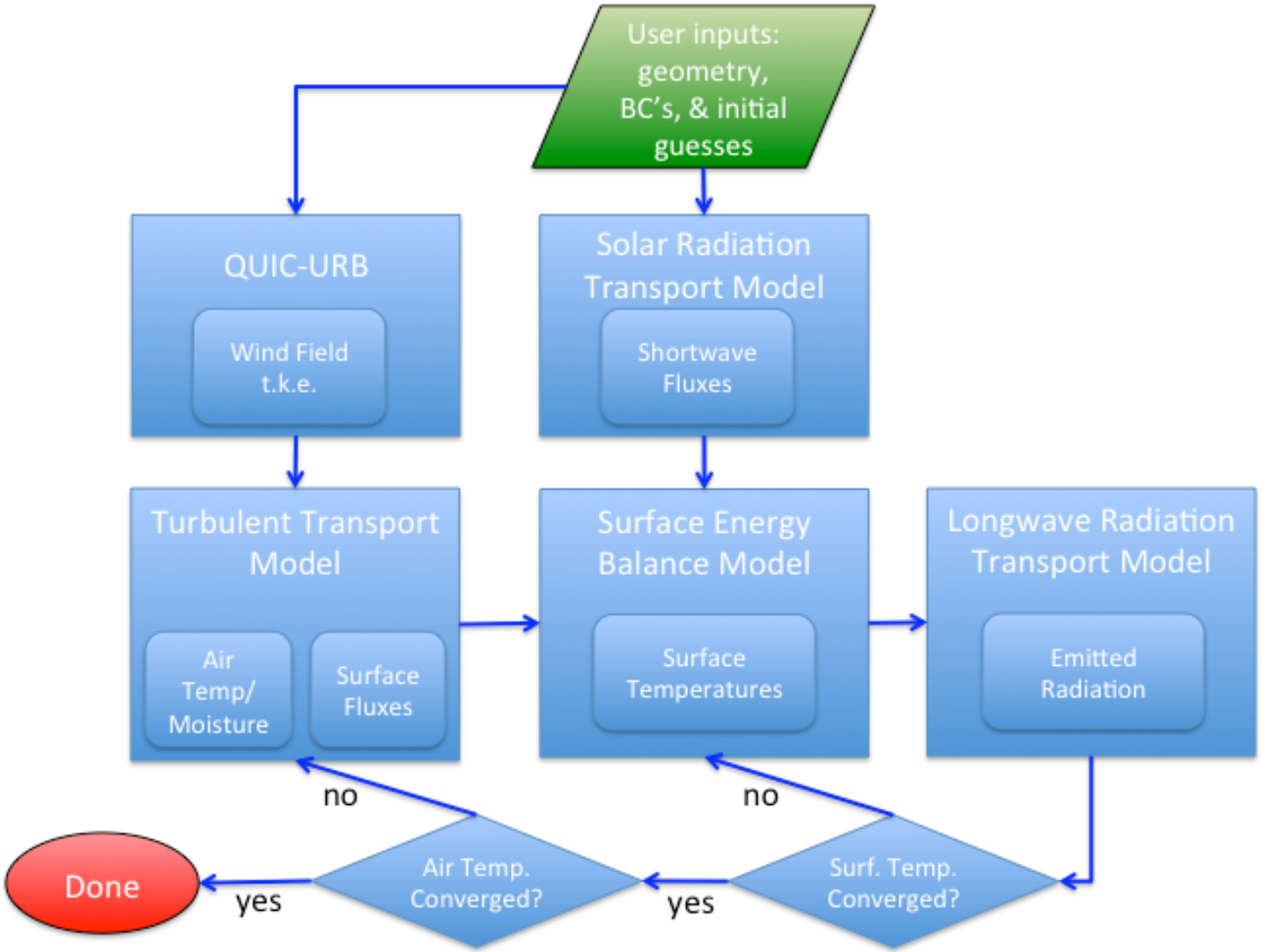


Computations are done in parallel!

- Launch of ray-object intersections happens concurrently on the GPU

QES Turbulent Transport Integration

QES - Coupled System for Computing Air and Surface Scalars



QES Turbulent Transport Integration

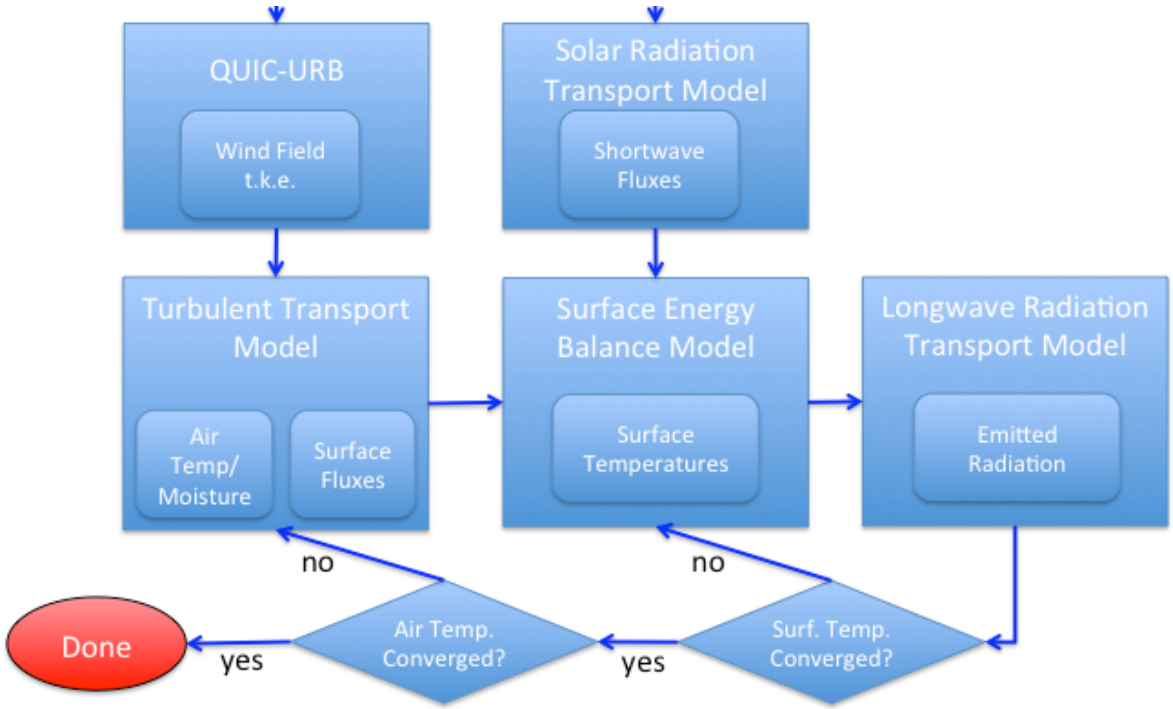
Integrating Turbulent Transport and Radiation Models with QES:

```
qes::RadiationTracer rad;

qes::QESSurface lsm;
lsm.setRadiationTracer( &rad );

qes::TurbTransModel ttm;
ttm.setLSM( &lsm );

context.joinModel( &rad );
context.joinModel( &lsm );
context.joinModel( &ttm );
context.initialize();

for( ts; ts < timesteps; ++ts ){
  context.sunTracker->updateTime( ts );
  context.runSimulation();
}
```

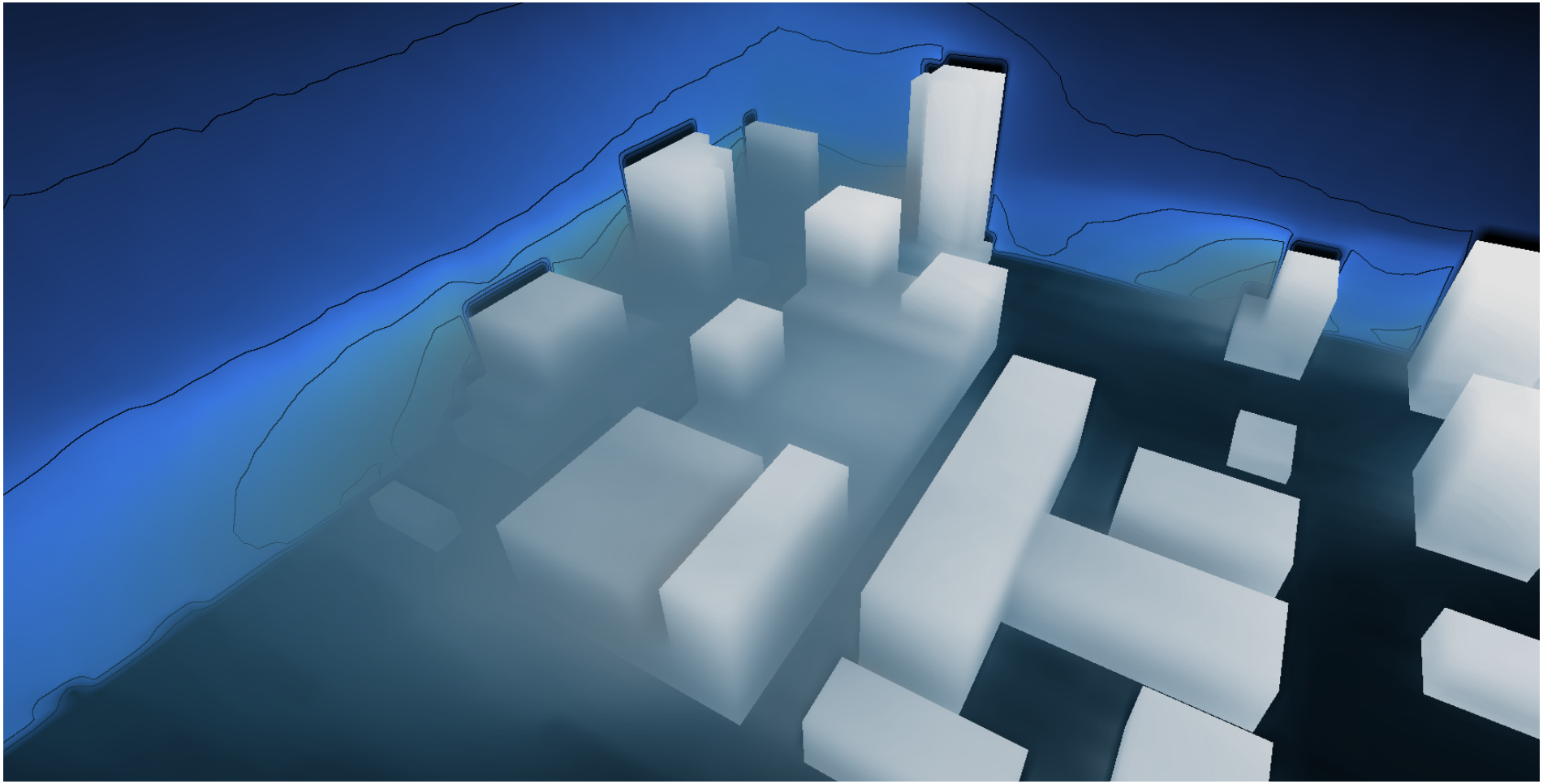
QES Turbulent Transport Integration

Setup of code hides many details. Tight coupling between models internally:

1. TTM requests updated surface energy components from LSM
2. LSM acquires shortwave energy, repeatedly re-acquiring longwave energy iteratively to converge on a surface temperature.
3. TTM may repeat 1 and 2 if not temperature not converged



QES Turbulent Transport Integration



- Validation is currently ongoing
- Models (and internal models - e.g. flux computations) are customizable
- Patches can run different models - apply specific knowledge of surfaces
- Can force models arbitrarily

Conclusions

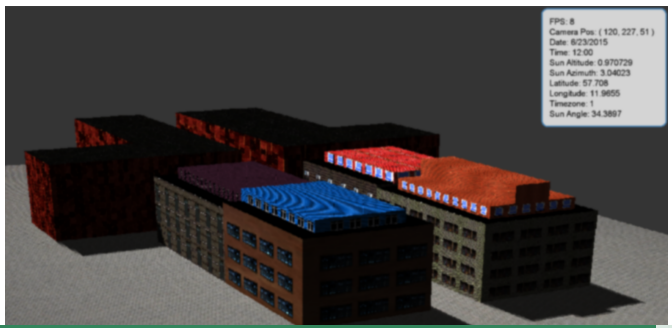
QES - QUIC EnvSim

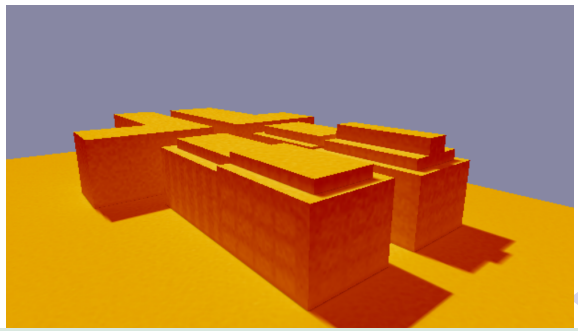
- Multiscale modeling system for urban microclimate
- Utilizes GPUs on desktop workstations
- Computes SVFs, Shortwave, Longwave, Turbulent Transport and Land Surface Model Interactions

QES - Customizable Coupling

- Affords multiple (and different) models per patch within the domain
- Not a static modeling system and dynamic - can select best model based on efficiency or accuracy
- Models are coupled and linked through buffers (abstracted GPU or CPU memory)

Looking forward WUDAPT L2, but can we also add L3 and L4?



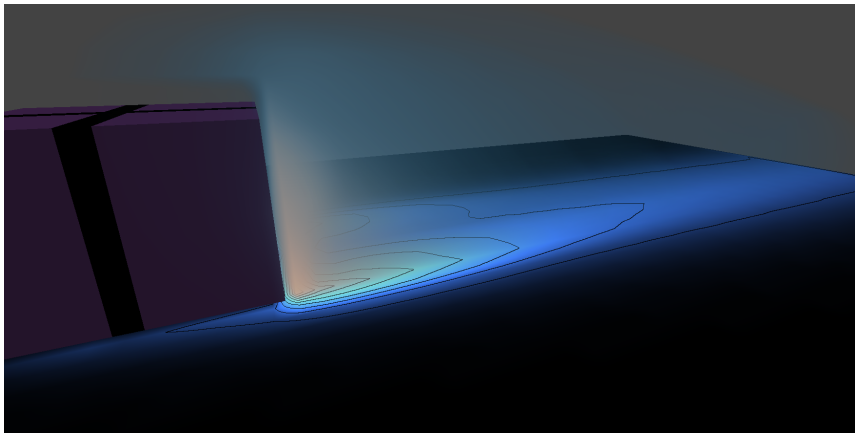


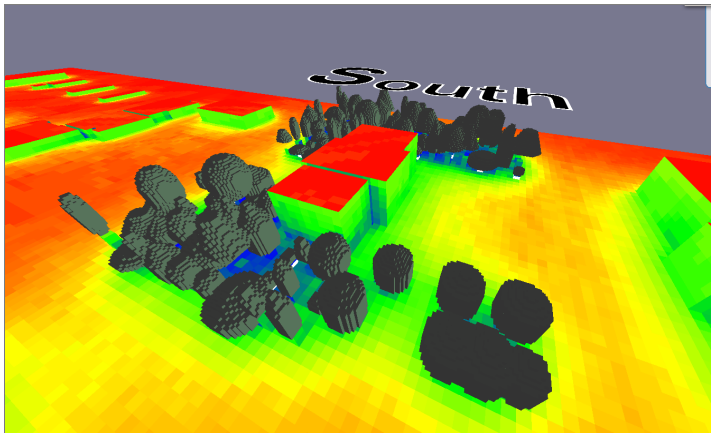
Questions?

Some final thoughts:

- Validations can be challenging - need detailed meta-data for comparing with experiments
- Consider orienting sensors normal to canyon walls to capture all information

Questions!





Acknowledgments: David Schroeder (volume visualization system)

This material is based upon work supported by the National Science Foundation under Grants No. 1133590, 0828206, CBET-PDM 113458, EPS 1208732, and AGS 1255662. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.