A PARAMETRIC TOOL FOR COMMUNITY-SCALE MODELING

Peter Ellis
Big Ladder Software

ABSTRACT
Community-scale modeling is a rapidly growing area of interest for planners, policy makers, and researchers. There are, however, few software tools that enable large-scale parametric studies of communities based on detailed energy models of buildings. This paper presents a tool called Params-NZP for automatically generating and running full EnergyPlus models for the parametric analysis of entire communities.

Params-NZP has been developed as one component of a larger framework called the NZP Tool. Created by the US Army Corps of Engineers as a decision-making aid for Army planners and energy managers, the NZP framework provides a web-based graphical interface that allows users to model energy, water, and waste scenarios for hundreds of buildings at Army bases and installations. Such installations accommodate communities with typical populations between 10,000 to 50,000 residents.

The NZP framework interfaces with Params-NZP using XML input and output files. Params-NZP itself is composed of a command-line application manager, a generalized parametric template program called Params, a “Net Zero” (NZ) template library for EnergyPlus, and a collection of prototype building models. The prototypical building models include 14 Army building types, 16 commercial building types, and 4 residential building types.

INTRODUCTION
Community-scale modeling, also known as urban-scale modeling, is a rapidly growing area of interest for planners, policy makers, and researchers. Just as sustainability has been recognized as important for individual buildings, it has also been recognized for the greater impact that sustainability can have at the scale of communities—from campuses to cities. Planning for

the sustainability of communities also presents opportunities that are not found with individual buildings. There are system benefits and efficiency gains that can be made by grouping buildings together.

Energy modeling of individual buildings is an established part of the design process—even if still too infrequent. There are numerous free and commercial software tools that facilitate detailed whole-building modeling and analysis. Many of these tools also support capabilities for parametric analysis, i.e., the task of varying input values across multiple simulation cases.

While the industry for modeling individual buildings is approaching maturity, the modeling of communities is still at an early stage. There are few software tools that enable large-scale parametric studies at the community scale. There are even fewer tools (if any) that leverage detailed whole-building energy models for this purpose. This paper presents a tool called Params-NZP for automatically generating and running full EnergyPlus models for the parametric analysis of entire communities.

RELATED WORK
The literature describes a variety of community-scale modeling tools but very few that explicitly model building energy use. A recent review (Reinhart and Davila, 2016) surveys a number of software tools and approaches for community-scale modeling.

CitySim (Robinson et al., 2009) is perhaps the most well-published software tool that does model energy use in communities. CitySim uses the actual urban geometry of the building population and excels at modeling solar radiation and shading interactions between buildings. However, for the sake of computational efficiency CitySim uses a highly-simplified model for buildings based on a resistor-capacitor network. The HVAC systems and equipment
model is also simplified and only covers a limited selection of equipment technologies.

Another tool called umi (Reinhart et al., 2013) integrates building energy use, transportation, daylighting, and outdoor comfort. umi couples EnergyPlus and Radiance/Daysim as the core simulation engines. The EnergyPlus models are somewhat simplified.

A different approach for community-scale modeling has been applied by others (Marston et al., 2014). Instead of using an integrated software tool, such as CitySim or umi, to model the actual geometry of buildings, Marston et al. used generalized EnergyPlus models based on the US Department of Energy’s Commercial Reference Buildings (DOE, 2014) to simulate representative buildings. The results are then scaled by actual building floor area.

**NZP TOOL**

The NZP Tool—an installation sustainability and resiliency planner—is a web-based computational platform (Case et al., 2014) created as a decision-making aid for Army planners and energy managers. In development by the US Army Corps of Engineers since 2011, the NZP platform provides a web-based graphical interface that allows users to model energy, water, and waste scenarios for hundreds of buildings at Army bases and installations. Such installations accommodate communities with typical populations between 10,000 to 50,000 residents.

**Background**

The NZP platform was developed in order to help the Army meet Federal energy requirements. The Energy Independence and Security Act of 2007 (EISA, 2007) mandates that new and renovated federal buildings achieve a reduction in energy consumption of 55% by 2010, 80% by 2020, and net zero by 2030. The stringent requirements of EISA 2007 have driven the Army to adopt a bold policy for achieving net-zero installations (NZIs) for energy, water, and waste. The policy sets out milestones to achieve net-zero energy in five installations by 2021, twenty-five installations by 2031, and all installations by 2058. Prior to EISA the 2005 Energy Policy Act (EPAct, 2005) requires that federal facilities be built to achieve at least a 30% decrease in site energy over ASHRAE Standard 90.1-2004 (ASHRAE, 2004).

Considering the scale of the problem with hundreds—if not thousands—of buildings at Army installations, and hundreds of installations around the world, it is hard imagine that the Army will be able to address these challenging goals without a comprehensive modeling tool. A manual approach would be prohibitively expensive and time-consuming in terms of data collection, organization, modeling, and analysis.

The current status of the NZP platform is that it is still a work in progress but is being actively used on numerous production projects across a broad spectrum of Army installations.

**Capabilities**

The NZP platform offers several unique capabilities compared to other community-scale software tools. In addition to modeling energy across many buildings, the platform also models water and waste streams. The Army net-zero policy targets not just energy, but also water and waste.

The platform offers an integrated planning process that divides a study into five major steps:

1. Goal setting
2. Baseline data collection and calibration
3. Building-level optimization
4. Supply and distribution system optimization
5. Plan and project formulation.

The NZP platform allows large batches of buildings to be simulated using Params-NZP (see below). Buildings can be “clustered” to permit the evaluation of central plant systems for heating and/or cooling. The platform also has capabilities for optimizing supply and distribution systems.

**Architecture**

The NZP web application (see Figure 2) consists of a browser-based graphical user interface (GUI), an NZP server, a job server, and domain-specific modules.

- **Graphical User Interface** – a client-side browser-based interface that allows users to log in securely, enter data about their installation, run simulation planning studies, and view results.
- **NZP Server** – a server-side application with database that mediates communications between GUI clients, job server, and modules.
- **Job Server** – a server-side application that queues and distributes plugin jobs for processing across multiple processor resources.
- **Domain-Specific Modules** – programs that run on a distributed processor resource and perform a domain-specific job such as running an energy simulation or other calculation.
**Params-NZP** is the domain-specific module for performing building energy simulations using **EnergyPlus**. The web application interacts with **Params-NZP** by creating a job on the **NZP** Server and then passing it on to the Job Server which allocates a processor resource and calls the module. When the simulation is finished, results are returned to the **NZP** Server via the Job Server.

The **EnergyPlus** simulation engine (Crawley et al., 2004) was selected as the primary tool for building energy modeling. **EnergyPlus** offers state-of-the-art features, a highly-accurate simulation approach based on the heat balance method, and great flexibility for configuring HVAC systems and controls. The complexities of the program were mitigated by using the **Params** templates to provide a layer of abstraction between **EnergyPlus** and the **NZP** framework.

One of the disadvantages of **EnergyPlus** is the potential for long simulation runtimes. This is an especially critical issue for the **NZP Tool** considering the fact that an analysis of just one Army installation could require hundreds or thousands of **EnergyPlus** simulations. To combat this problem, several steps were taken to speed up runtimes for the **Params-NZP** models. The number of zones and surfaces was kept to a minimum. The convergence tolerances were reduced and the simulation timestep was set to one timestep per hour. Comparisons between results before and after the changes were made showed only a small difference of 1-2% in annual energy numbers, but with a significant speed gain of up to a factor of ten.

**PARAMS-NZP**

**Params-NZP** is composed of a command-line application manager, a generalized parametric template program called **Params**, a “Net Zero” (NZ) template library for **EnergyPlus**, and a set of prototype building models.

We began developing **Params** in 2010 as an in-house software tool to automate a parametric analysis for the US Army Corps of Engineers (USACE). The project (Liesen et al., 2012) required us to generate and run **EnergyPlus** models for two building types across 15 climate zones and 14 energy-efficiency measures. By 2011, USACE drafted us into the **NZP** project.

**Params-NZP** accepts an XML input file specifying prototype model type, weather file, and a set of high-level parameter values. It automatically generates an **EnergyPlus** model from the parameter values and then runs the simulation with the weather file. When complete, it post-processes the energy results into an XML output file that can be parsed by the **NZP** platform where the data are aggregated across the installation. The framework also uses an XML definition file that configures parameter inputs—according to climate zone—for energy code standards, predefined energy-efficiency measures, and predefined packages of measures.
**Params**

*Params* is a free and open-source, cross-platform framework for parametric modeling (Ellis, 2015). *Params* integrates two powerful concepts: templates and scripting. Templates are an old method for combining static content with dynamic inputs. Scripting is a relatively new method for automating the process of modeling.

We developed *Params* to automate the generation and management of *EnergyPlus* models but it is generic enough to be used with any simulation engine that accepts text-based input files.

By integrating templates and scripting, *Params* makes it possible to create a highly-efficient and flexible workflow for generating building energy models for parametric analysis.

Parametric analysis is an essential technique for building energy modeling. Almost every modeling project comes down to varying input parameters and studying the differences in results. Even the comparison of a baseline model to a proposed model is about changing multiple parameters to evaluate the effect.

The *Params* framework is composed of a parametric templating system and a comprehensive library of *EnergyPlus* templates. The templates are plain-text files that consist of standard *EnergyPlus* input file (IDF) syntax “marked up” or parameterized with dynamic content in the Ruby scripting language. Because the templates are mostly made up of IDF objects, the templates can be readily modified and extended by modelers and other non-programmers.

Ruby is a full-featured, modern programming language that allows nearly unlimited flexibility for configuring and connecting templates. Yet it has a simple, easy-to-learn syntax that makes everyday template tasks as straightforward as programming an *Excel* spreadsheet macro. Templates can be parameterized with any number of inputs.

*Params* offers several advantages compared to other parametric tools that are currently available. *Params* does not require a knowledge of an extensive API or object-oriented programming structures. *Params* only requires a basic understanding of programming concepts such as math operations, if-then logic, and looping with arrays. *Params* is also more capable than other tools because it provides full access to all of the...
Features and power of EnergyPlus. Perhaps the greatest strength of Params is that it makes it easy to swap out entire HVAC systems parametrically in a model.

Template Library

The Params framework includes a template library that we have developed for EnergyPlus. The library includes ready-to-use templates for space loads, zone HVAC equipment, central HVAC systems, and more. They are designed to be modular and completely reusable from project to project. The template library has been developed and refined over numerous modeling projects. The templates encapsulate a significant body of knowledge with respect to modeling best practices in EnergyPlus.

Params templates are distinct from the HVACTemplate input objects (DOE, 2015) that are available in EnergyPlus. The rules that generate the HVACTemplate output are hard coded and compiled into an executable program. The actual source for HVACTemplate objects cannot be viewed or changed by the user. Params, on the other hand, is based on true text-based templates that can be viewed and modified by the user.

Prototype Models

We have developed 34 prototype building models in Params format for the NZP platform. The prototype models include 14 Army building types, 16 commercial building types, and 4 residential building types. The commercial models are derived from the US Department of Energy’s Commercial Reference...
Buildings (DOE, 2014).

All of the prototype models are fully parameterized for common modeling inputs. Available parameters include:

- Wall details (assembly type, cavity and continuous insulation R-values)
- Roof details (assembly type, cavity and continuous insulation R-values)
- Slab/basement details (horizontal/vertical insulation widths/depths and R-values)
- Window details (U-value, SHGC)
- Infiltration (air leakage rate, air leakage schedule, use of vestibules)
- Occupancy (people densities and schedules by space type)
- Lighting (lighting power densities and schedules by space type)
- Equipment (equipment power densities and schedules by space type)
- Domestic hot water (flow rates, temperatures, schedules)
- Zone HVAC (unit type – ATU, FCU, PTAC, PTHP, Win AC, WSHP, VRF, UH, radiant)
- Zone HVAC operation (thermostat setpoints, setback/setup schedule, outdoor air)
- Zone HVAC performance (fan/heat/cool efficiencies)
- Central HVAC (system type – VAV, CAV, DOAS, HW, CHW, VRF, dual duct)
- Central HVAC operation (schedule, outdoor air, supply setpoints, controls)
- Central HVAC performance (fan/pump efficiencies, heating/cooling COPs)
- And more...

The models also include “baked-in” energy-efficiency measures. In addition to the efficiency measures that can be applied by changing a single parameter (e.g., lighting power density, fan efficiency, or chiller COP), there are parameters that act as “switches” to make larger changes to the model in order to simulate measures such as fan/pump type (constant vs. variable speed), boiler type (non-condensing vs. condensing), condenser type (air-cooled vs. water-cooled), daylighting controls, and even different HVAC system types.

CONCLUSIONS

While detailed energy modeling certainly presents significant computational overhead versus simpler approaches, it does provide a pathway to merge community-scale modeling with the detailed modeling of individual buildings (say with actual geometry) where desirable. The underlying modeling approach using detailed models is the same. Specific buildings could be selected from the pool of prototype models and upgraded with custom models. In some scenarios, e.g., smaller communities such as a university campus, it could be possible to replace all prototype models with custom models—or use a more selective approach to create custom models only for new projects, or projects already using modeling as part of the design process.

A future approach to provide the best of both worlds could involve creating detailed models that are then represented in the aggregate simulation with surrogate models.

ACKNOWLEDGMENTS

The author would like to acknowledge the generous support and collaboration of the Construction Engineering Research Laboratory (Champaign, IL) of the US Army Corps of Engineers for funding much of the development of Params-NZP. In particular the author would like to thank Dr. Michael Case and Dr. Richard Liesen for their vision and guidance.

REFERENCES


