VOLUME 25, NO. 1 ★ JANUARY/FEBRUARY 2004 (PUB-439)

BUILDING ENERGY SIMULATION

For Users of EnergyPlus, SPARK, DOE-2, BLAST, Genopt, Building Design Advisor, ENERGY-10 and their Derivatives

USER NEWS

EnergyPlus Wins R&D 100 Award

R&D Magazine recently announced that EnergyPlus is one of the winners of the 41st annual R&D 100 Awards, which honor the 100 most technologically significant new products of the year.

Features

Software

Departments

DesignBuilder and EnergyPlus

DesignBuilder Software

Mission: "To provide state-of-the-art building modelling tools in a package accessible to all."

Dr Andy Tindale
DesignBuilder Software Ltd.
Palace Chambers, 41 London Road
Stroud, Glos, GL5 2AJ, UK
www.designbuilder.co.uk sales@designbuilder.co.uk

DesignBuilder is a software tool for creating building models and generating visual, thermal and lighting performance data. Its innovative productivity features allow even complex buildings to be modelled rapidly by non-expert users. Internally, DesignBuilder uses the state-of-the-art thermal simulation engine EnergyPlus to generate its thermal performance data.

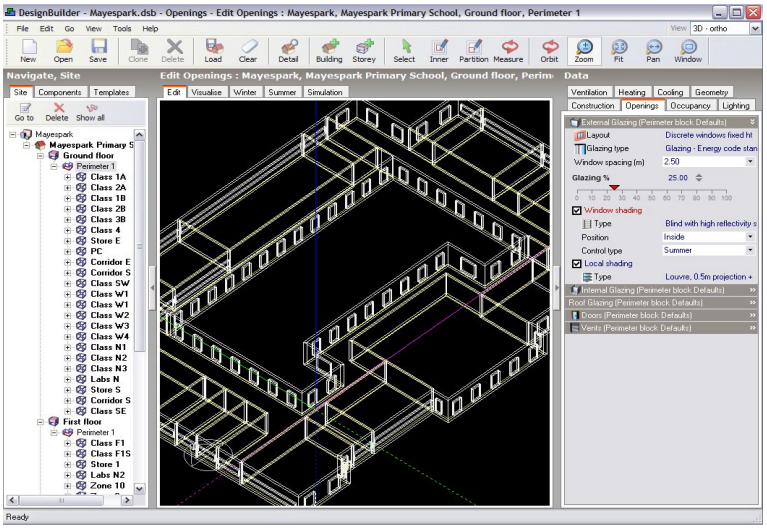
Visualisation and shading

Uniquely, DesignBuilder provides full control over the model detail. The whole building may be modelled or, for more precision, you may concentrate on a single zone. Or you can create a model with a very simple definition of construction but put considerable detail in each item of installed equipment.

These features allow DesignBuilder to be used effectively at any stage of the design process, from the concept stages where just a few parameters are needed to capture the building design, to much more detailed building models for established designs (or existing buildings).

User Interface and Modelling Features

The result of a considerable R&D programme over the last four years, DesignBuilder combines next-generation solid modelling technology with the latest simulation techniques to provide a powerful yet intuitive building-modelling tool. At the core of the program is the OpenGL geometric modeller, which allows building models to be assembled by positioning "blocks" in 3-D space. Blocks can be any shape and can be positioned adjacent to each other or in isolation. Once placed, blocks can be moved at will and divided up into zones simply by drawing the partition walls. Working with realistic 3-D solids (rather than abstract lines) provides visual feedback of actual element thickness and allows accurate calculation of floor areas and volumes.



OpenGL geometric modeller with explorer (L) and attribute editor (R)

Continued

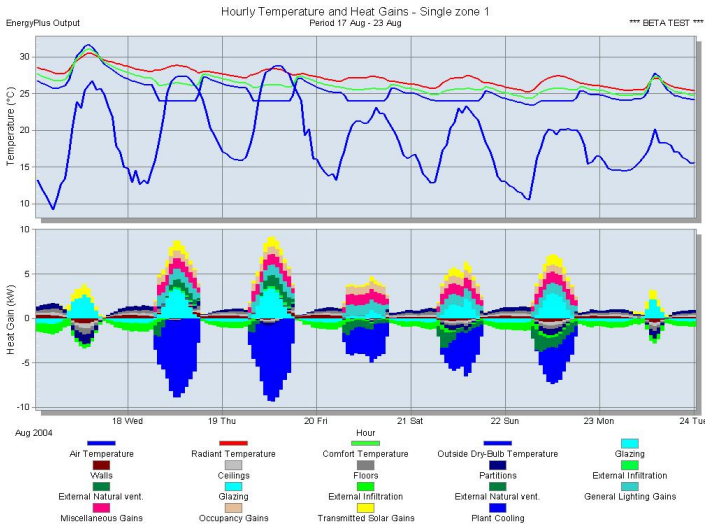
Data templates allow you to load common building constructions, activities, HVAC and lighting systems into your design by selecting from drop-down lists. You can also create your own templates if you often work on similar types of buildings. This, combined with data inheritance, allows global changes to be made at building, block or zone level.

Performance and design data

Display of environmental performance data is tightly integrated with the model edit screens and any calculations and/or simulations required to generate the data are started automatically. This means that you can concentrate on your modelling work without the distraction of running external modules and importing data. The following data can be shown in annual, monthly, daily, hourly or "sub-hourly" intervals:

- Energy consumption broken down by fuel and end-use.
- Internal temperatures, including temperature distribution charts.
- Weather data
- Heat transmission through building elements including walls, roofs, infiltration, ventilation etc.
- Heating and cooling loads.
- CO2 generation.

Heating and cooling plant sizes can be calculated using design weather data and a choice of calculation methods (EnergyPlus, 3TC, CIBSE admittance).

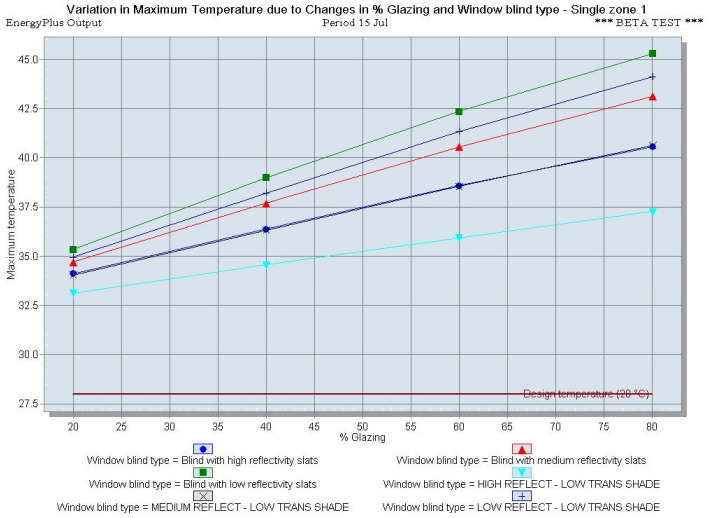


Hourly temperature and fuel consumption data for seven days

Performance comparisons

An important part of most modelling projects is to compare the performance of different building configurations. This is especially true in the early stages of the design process when the design is still fluid. DesignBuilder has built-in features to facilitate such comparisons:

- 'Bookmarks' mark multiple points in the design flow, allowing you to return to previous designs and also to compare performance data among all bookmarked designs.
- The parametric analysis facility allows you to investigate the effect of variations in design parameters on a range of performance criteria, simply by selecting the parameters to vary from drop-down lists (picture right).



Parametric analysis chart showing the effect of various facade designs on the maximum summertime temperature

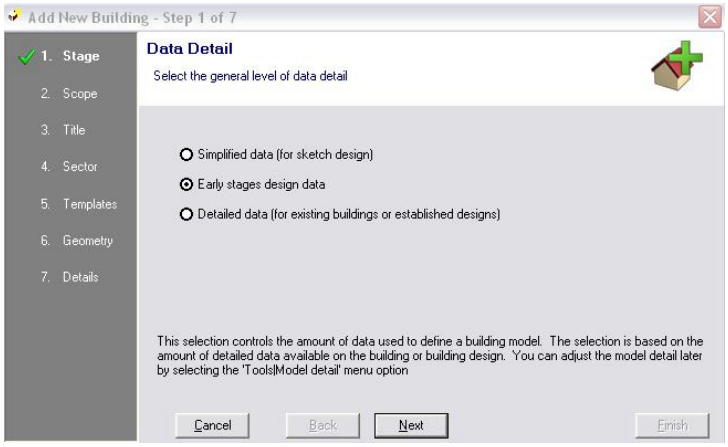
Continued

- A stock management capability, which allows buildings to be grouped together and modelled en masse. Energy-saving measure scenarios can be applied to the stock and the performance for each scenario reported. This feature will be of particular interest to holders of building stock portfolios to assist with policy development, stock aggregation and other research activities.

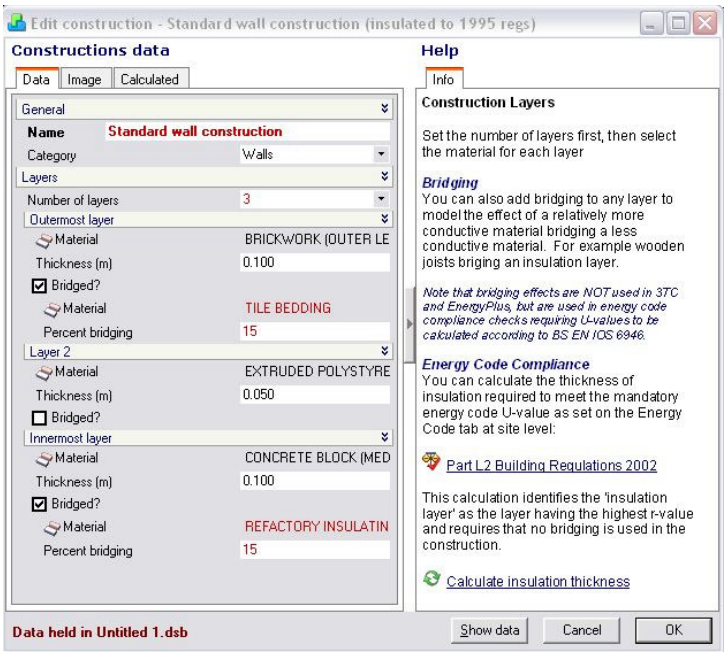
EnergyPlus Features Implemented

DesignBuilder implements all of the EnergyPlus fabric and glazing data input, providing databases of building materials, constructions, window panes, window gas, glazing units and blinds. Also implemented are:

- Shading by louvres, overhangs and sidefins as well as internal and mid pane blinds.
- COMIS natural ventilation with the option for windows to open based on a ventilation set point temperature. Crack sizes can either be calculated based on an airtightness setting or can be entered explicitly.
- Lighting control systems and calculating savings in electric lighting allow checking for optimal use of natural light.
- Heating and cooling load calculation using the "purchased air" system.
- Schedules using intervals of 60 to 6 minutes.
- Holidays - select the number of days per year and the local holiday schedule.



Add New Building wizard showing three basic model detail options



Edit Construction dialogue showing tool to calculate the insulation thickness for compliance with the local energy code

There are no limitations on surface shape – surfaces having more than four vertices are triangulated to ensure compatibility with the EnergyPlus simulator.

Nearly 1500 ASHRAE worldwide design weather data sets are included with the software and more than 580 EnergyPlus hourly weather files are available free on demand.

Continued

Uses and Users

DesignBuilder is suitable for use by architects, building services engineers, researchers, energy consultants, and students. Some typical applications are:

- Evaluating a range of façade options for the effect on overheating, energy use and visual appearance.
- Checking for optimal use of natural light, modelling lighting control systems and calculating savings in electric lighting.
- Visualisation of site layouts and solar shading.
- Thermal simulation of naturally ventilated buildings.
- Calculating heating and cooling equipment sizes.
- Communication aid at design meetings.
- Educational tool.
- Evaluating energy-saving measures on individual buildings or multiple buildings in a portfolio.

Availability and Cost

DesignBuilder v1.0 development is nearing its final stages and we expect to release a new beta version soon. If you would like to take part in the beta tests (and you haven't already registered by downloading the early beta test versions 0.3.x) please email sales@designbuilder.co.uk.

DesignBuilder Software

EnergyPlus University Course Teaching Material

The teaching of building energy simulation is well established at the university level at many institutions around the world. Many instructors have expressed an interest in using EnergyPlus in their courses if there was information available to help them switch. The U.S. Department of Energy is pleased to announce the availability of university course material for teaching building simulation using EnergyPlus. This course material is targeted specifically to the university environment for teaching students about building simulation while introducing them to EnergyPlus.

The course comprises 25 complete PowerPoint lectures (over 800 total slides) that cover topics such as strategies for using energy simulation, expectations, building input, primary and secondary systems, and advanced features. While many of the lectures deal explicitly with how to model various components and systems with EnergyPlus, the lectures also provide an appropriate overview of building systems so that the students can understand the context and purpose of the technology within the building. The presentations include text, graphics, photographs, color-coded input, and other features designed to make teaching and understanding the concepts of building energy analysis easier. The course material also includes notes to instructors about assumptions made when developing the course as well as a course outline/schedule to help the instructor organize the semester and homework assignments and projects.

The course material is available free of charge from the EnergyPlus web site. Those who download the course are simply requested to share feedback, example problems, experiences, etc. with DOE so that other instructors can benefit--and so that we can notify you of new material and updates in the future.

To download the materials, go to **http://www.eere.energy.gov/buildings/energyplus/cfm/training.cfm**

Ask An EnergyPlus Expert

AIR CROSS-MIXING

I am trying to add cross-mixing of air between zones into a building. The limitations of cross-mixing (not being able to link multiple zones) prevent me from using it where I have a hallway with two bedrooms along with an adjacent living space. Because of this, I have to use the standard mixing function, which does not affect energy balances of the source zones. Doesn't this generate "false energy" within a simulation? If another mixing function is added in reverse, in order to link the two zones, would the flow rates then have to be identical to keep the energy balances accurate?

Answer
You are correct in that improper use of MIXING objects can violate conservation of energy. The original intent of the MIXING objects was to model flow from one space to another to another, with the assumption that infiltration, or an HVAC system imbalance, was driving the flow to go through the zones and out the final zone as exfiltration. You should be able to use opposing MIXING objects to accomplish what you want.

CHILLER STATUS

Is it possible to turn a chiller off for less than an hour in EnergyPlus simulations? I ran into problems after defining the relevant schedules on a 24-hour basis.

Answer
You can create sub-hourly schedules using DAYSCHEDULE:INTERVAL and DAYSCHEDULE:LIST. Then use these schedules in your PLANT OPERATION SCHEMES object. Note that all schedules are resolved to the zone time step. For example, if you set TIMESTEP IN HOUR = 4, the zone time step will be 15 minutes. If you describe a sub-hourly schedule with 10-minute intervals, the schedule values will be interpolated to determine the value every 15 minutes to match the zone time step.

PEOPLE SENSIBLE VS. LATENT LOAD

I have two conditioned zones that are maintained at roughly the same comfort conditions. The latent percentage of the human-emitted heat in one of the zones seems fine, roughly varying between 20 to 30 percent. Yet my other zone reports the latent percentage at zero most of the time and at the maximum 1 percent. I looked at the engineering document to verify this but the equation does not show how such low latent percentages could appear. The heat given off in the first zone is 96W while the second zone is 64W. If someone could clear this up I would appreciate it.

Answer
The equation on p. 169 of the Engineering Reference (v1.1.1.012) determines the sensible heat gain from people as a function of metabolic rate and air temperature (dry bulb). The latent gain is the difference of total gain less the sensible gain.

Figure 72 on p.169 (pdf-p.189) shows plots of sensible gain vs. metabolic rate (total gain) at various air temperatures. Whenever the curves show a value of sensible gain ≥ the total gain, there will be no latent gain from the people. This occurs at low metabolic rates and low air temperatures. For example, at 21.1C (70F, the magenta line) shows sensible ≥ total at metabolic rates below approximately 80W. Those sensible gains that are greater than the total are artifacts of the curve-fitting process and are set in the algorithm to equal the total; this results in 100 percent sensible gains from people at those conditions.

Ask An EnergyPlus Expert

TIME STEP REDUCTION

I am interested in the comparison and relative assessment of heating concepts. One special focus is the "stochastic use" of individual rooms and the possibility of energy savings if very fast ("almost massless") heating devices are used. I can, of course, limit the periods of occupation to single hours that are distributed adequately over the various days of some example year, but I did not want to:

1. either define a starting time of the heating equipment relative to the onset of the room use with very short precursing time (like some minutes), or

2. tell EnergyPlus to report comfort and temperature readings in minute ranges after the heating sets in, which might be an alternative approach in assessing those interesting values.

In my understanding the smallest time step is 1/6th of an hour, which would not be adequate for this kind of investigation.

Answer:
If all you need to do is change the number of time steps in an hour, you can do this (advanced feature) by changing the line in the EnergyPlus.idd:

```
  TIMESTEP IN HOUR,
          \unique-object
     N1 ; \field Time Step in Hour
          \required-field
          \note Number in hour: validity 1 to 6: 4 suggested
          \note Should be evenly divisible into 60
          \note Specifying 6 as maximum as higher values may cause instability.
          \default 4
          \type integer
          \minimum 1
--->      \maximum 6

to        \maximum 60
```

This would allow you to specify a time step in an hour of 60 (1 minute time steps). The value still needs to be evenly divisible into 60, so 30 (2 minute time steps) will also work.

ZONE GEOMETRY

I have a 5-story building with an exterior wall that is adjacent to 11 zones (two on each floor plus the atrium). I am modeling this wall as adibatic since I am only modeling a certain section of the building. My question is: can I enter it as one wall or must it be broken up into 11 smaller walls (one with each zone)? if it can be entered as one wall, then what should my input for "InsideFaceEnvironment" be (since EnergyPlus allows only one option)? OutsideFaceEnvironment is ExteriorEnvironment, which shouldn't matter as it is adibatic.

Answer
You should break up that wall into separate surfaces that each conform to the different zones. If you set the outside face to ExteriorEnvironment, then your wall is not going to be "adiabatic." The closest you can come to adiabatic (with boundary conditions) is to set the outside face to "otherzonesurface" and then use the name of the surface itself as the "Outside Face Environment Object." Note that this surface is still modeled with thermal mass and is not really "adiabatic" since it will store and release heat if the interior temperature changes as with setup/setback. You could also set an interior layer of the wall to an unrealistic, extremely low conductance if you really want to emulate "adiabatic."

Ask An EnergyPlus Expert

SHADING CONTROL

I would like the shading control "AlwaysOn" at night instead of during the day, and I need to have a control on the angle of the blind slat. It's an exterior blind and the angle I want is 105° respective to the horizontal. I need this control to maximize the solar gain in winter. Can you help me?

Answer:
Here's how to set the angle you require: Note that "AlwaysOn" means that the blind will be in place day and night.

1. In MATERIAL:WINDOWBLIND set
   Slat angle = 105
2. In WINDOWSHADINGCONTROL set
   Shading Control Type = AlwaysOn
   Type of Slat Angle Control for Blinds = FixedSlatAngle

BLINDS' EFFECT ON TEMPERATURE

I simulated interior window blinds in a naturally-ventilated space. The blinds were modeled following the EnergyPlus package example "PurchAirWindowBlind.idf": blinds controlled by incident solar on blinds. If the windows of the space face south, then indoor temperature (Tin) with blinds is noticeably higher than without blinds in the morning until 2:00 pm. Then in the afternoon, the space without interior blinds is warmer. Similarly, if the window faces west, with blinds is warmer 1-3F than without between 12:00 pm to 16:00 pm. I'm just wondering whether such result is valid and how to check?

Answer
Look at Report Variables "Window Blind Slat Angle" and "Fraction of Time Shading Device Is On" to determine whether the blinds are deployed during the morning hours. Depending on the control setpoint there may be enough incident diffuse solar to deploy the blinds in the morning. Blinds will affect both the solar and thermal performance of the window. Blinds reflect solar back outside, but they also convert transmitted solar into a rapid convective gain instead of allowing solar to be absorbed by building surfaces. With all of these interactions to consider, it is certainly possible that blinds might cause the space to be warmer in the morning.

ZONES WARNINGS

I am simulating a radiant slab cooling system (hydrolic low temperature radiant slab system) in a 40m*40m plan building. If I use DOE-2 to simulate a conventional HVAC system, this building could be considered as five different zones (one central zone and four perimeter zones) with no internal wall between zones. Now, if I use EnergyPlus to simulate the radiant slab system, do I need to zone the building to five zones? And if the building is taken as five zones, with no internal walls between zones, will the simulation result issue a warning for zones that do not have six surfaces. Could I omit this warning or should I add a internal wall between zones?

Answer
The presence of interior walls will affect both the radiant exchange and the thermal mass of the zones. If these walls do not exist in the building, then it is appropriate to model your zones without them. You might consider adding internal mass objects to each zone to account for the mass of furniture and cubicle partitions, if present. Warnings from EnergyPlus are intended to point out something unusual so the user does not omit something by mistake. Severe errors indicate that there may be a problem with the results. In this case, if you get warnings you can simply ignore them.

Ask An EnergyPlus Expert

ZONE TEMPERATURES

How does EnergyPlus determine the initial temperatures of zones ?

Answer
EnergyPlus starts off initial conditions with a fairly high temperature and then iterates until convergence is reached using the actual weather conditions. You will see "# warmup days" in the .eio file as an indicator or, if things did not converge, you will see "Loads initialization did not converge" in the .err file. You can affect the convergence tolerances by setting values in the "Building" object; this is also discussed a bit in FAQ Reference and should be in the Building Object itself.

INLET VENTS

I am attempting to model "self regulating inlet vents" as part of the natural ventilation strategy in a five story office building. Even though these inlet vents will be on different floors, they are designed to provide a relatively constant flow rate over a range of air pressure differences due to stack effect and external wind conditions. How can I model this using EnergyPlus? Also, is there a specific air-flow component that I could use or will I have to trick an opening/crack to achieve this?

Answer
If these inlet vents are expected to provide a constant flow rate, the simple Ventilation command may be appropriate. Using the polynomial coefficients you can define the flow relationship versus wind speed and temperature difference, or you can set the coefficients to provide a constant flow with no dependence on outdoor conditions.

or

If the self-regulation of the inlet vents is due totally to natural forces then you could model them as COMIS Openings. However, all COMIS Surfaces must be associated with a Heat Transfer Surface or Subsurface, so you would have to define a fictitious window of the appropriate dimensions and set its heat transfer and solar properties to zero.
Also, make sure its location, especially its height (Z) relative to the zone that it serves, are input correctly.

NIGHT PURGE

I am modeling a naturally ventilated building using the COMIS link. I want to investigate the effect a night purge has on daytime temperatures. The night purge would involve some sort of mechanical ventilation system that would run throughout the night or when the temperature inside reaches a certain point and the exterior temperature drops to a certain point. Does anyone know the best way to model this using Energy Plus.

Answer
We worked out some techniques for night purge that involve using schedules and multiple runs of EnergyPlus. We used results from an initial run to compute a schedule for when it makes sense to do night purge. Then we assigned the schedule to the minimum outdoor air in an outdoor air mixer (note this was broken in version 1.0.3). We then created a full, 8760-value schedule using external code and a decision making algorithm (e.g., cool OA, no warm-up heating, etc.). This isn't easy, but it is something useful, made possible by the versatile scheduling capabilities of EnergyPlus.

Alternatively, you could use a simplistic approach and use the VENTILATION object in each zone.

PUMP FATAL ERROR

I'm simulating a building with three thermal zones and a central "simple boiler"; I received this message:
`" Fatal ** The following pump is not the first component on its loop: HW CIRC PUMP"`
I have controlled the branch list and the inlet and outlet node and the pump is the first component of the supply side. Why the fatal error.

Answer
The first branch must be the supply side inlet branch and the first component on this branch must be the pump.

Ask An EnergyPlus Expert

COMPACT SCHEDULES

The staff at the EnergyPlus Users Office often receives questions about schedules. Here's a tip about the new (and much easier) schedule:compact feature. Usually you would start a schedule by creating one or more dayschedules, then a weekschedule, then a schedule. With compact schedules, you can do it all in one command. Here is a relatively simple schedule:

```
DAYSCHEDULE,
  OC-1,  !- Name
  Fraction,  !- ScheduleType
  0.0,  !- Hour 1
  0.0,  !- Hour 2
  0.0,  !- Hour 3
  0.0,  !- Hour 4
  0.0,  !- Hour 5
  0.0,  !- Hour 6
  0.0,  !- Hour 7
  0.0,  !- Hour 8
  1.0,  !- Hour 9
  1.0,  !- Hour 10
  1.0,  !- Hour 11
  0.8,  !- Hour 12
  0.4,  !- Hour 13
  0.8,  !- Hour 14
  1.0,  !- Hour 15
  1.0,  !- Hour 16
  1.0,  !- Hour 17
  1.0,  !- Hour 18
  0.5,  !- Hour 19
  0.1,  !- Hour 20
  0.1,  !- Hour 21
  0.0,  !- Hour 22
  0.0,  !- Hour 23
  0.0;  !- Hour 24

DAYSCHEDULE,
  OC-2,  !- Name
  Fraction,  !- ScheduleType
  0.0,  !- Hour 1
  0.0,  !- Hour 2
  0.0,  !- Hour 3
  0.0,  !- Hour 4
  0.0,  !- Hour 5
  0.0,  !- Hour 6
  0.0,  !- Hour 7
  0.0,  !- Hour 8
  0.0,  !- Hour 9
  0.0,  !- Hour 10
  0.0,  !- Hour 11
  0.0,  !- Hour 12
  0.0,  !- Hour 13
  0.0,  !- Hour 14
  0.0,  !- Hour 15
  0.0,  !- Hour 16
  0.0,  !- Hour 17
  0.0,  !- Hour 18
  0.0,  !- Hour 19
  0.0,  !- Hour 20
  0.0,  !- Hour 21
  0.0,  !- Hour 22
  0.0,  !- Hour 23
  0.0;  !- Hour 24

WEEKSCHEDULE,
  OC-WEEK,  !- Name
  OC-2,  !- Sunday DAYSCHEDULE Name
  OC-1,  !- Monday DAYSCHEDULE Name
  OC-1,  !- Tuesday DAYSCHEDULE Name
  OC-1,  !- Wednesday DAYSCHEDULE Name
  OC-1,  !- Thursday DAYSCHEDULE Name
  OC-1,  !- Friday DAYSCHEDULE Name
  OC-2,  !- Saturday DAYSCHEDULE Name
  OC-2,  !- Holiday DAYSCHEDULE Name
  OC-1,  !- SummerDesignDay DAYSCHEDULE Name
  OC-2,  !- WinterDesignDay DAYSCHEDULE Name
  OC-1,  !- CustomDay1 DAYSCHEDULE Name
  OC-1;  !- CustomDay2 DAYSCHEDULE Name

SCHEDULE,
  OCCUPY-1,  !- Name
  Fraction,  !- ScheduleType
  OC-WEEK,  !- Name of WEEKSCHEDULE 1
  1,    !- Start Month 1
  1,    !- Start Day 1
  12,  !- End Month 1
  31;  !- End Day 1
```

This whole process can be simplified by using a compact schedule (*example at right*):

A set of schedules in the new compact format for commercial and residential building types is included with EnergyPlus v1.1.1. Look for schedules.idf under datasets. Compact schedules are described in the Input/Output Reference beginning on page 49.

```
! Schedule Occupancy
SCHEDULE:COMPACT,
OCCUPY-1,
Fraction,
For: Weekdays SummerDesignDay,
Through: 12/31,
Until: 08:00, 0.0,
Until: 11:00, 1.0,
Until: 12:00, 0.8,
Until: 13:00, 0.4,
Until: 14:00, 0.8,
Until: 18:00, 1.0,
Until: 19:00, 0.5,
Until: 21:00, 0.1,
Until: 24:00, 0.0,
For: Sunday Saturday Holiday WinterDesignDay,
Through: 12/31,
Until: 24:00, 0.0;
```

EnergyPlus Version 1.1.1

To download a free copy of the program go to www.energyplus.gov

EnergyPlus Support Tools

Support software is listed on our website (http://SimulationResearch.lbl.gov/EP/ep_tools.html) and in Section 2 of this newsletter.

EnergyPlus Weather Data from www.energyplus.gov/

There are 275 locations in the United States, 16 California thermal zones, 55 Canadian locations, and 233 international locations in more than 80 countries.

Ask an EnergyPlus Expert

Questions from EnergyPlus users are answered promptly via email by program developers. To submit questions, join the EnergyPlus User Group at http://groups.yahoo.com/group/EnergyPlus_Support/.
A selection of questions/answers are compiled (yearly) into a downloadable PDF document:
Q and A for 2002, Q and A for 2003.

Are you an EnergyPlus Consultant ?

If you are engaged in EnergyPlus consulting, and would like to be listed in the *Building Energy Simulation User News* and on our website (http://SimulationResearch.lbl.gov), please send details to klellington@lbl.gov.

Join the EnergyPlus User Group

The developers of EnergyPlus have formed a support group to foster discussion and maintain an archive of information for program Users. We invite questions about program usage and suggestions for improvement to the code. Go to http://groups.yahoo.com/group/EnergyPlus_Support/

Translate EnergyPlus Web Pages

A new link on the main EnergyPlus web page (www.energyplus.gov/) allows you to view the pages in any of eight languages. Unfortunately, the translator doesn't work with PDF files. Look for the fish at the bottom of the web page. Pages may be translated into Chinese, French, German, Italian, Japanese, Korean, Portuguese and Spanish.

EnergyPlus is being developed by University of Illinois and Lawrence Berkeley National Laboratory, with the assistance of DHL Consulting, C. O. Pedersen Associates, Florida Solar Energy Center, GARD Analytics, the National Renewable Energy Laboratory, Oklahoma State University and others. Development of EnergyPlus is supported by the U. S. Department of Energy, Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies Program (Program Manager, Dru Crawley).

SPARK is an equation-based simulation environment that allows you to build customized models of complex physical processes by connecting calculation objects that represent system components like walls, fans, heat exchangers, chillers, ducts, mixing boxes, controls, etc. It is aimed at the simulation of innovative and/or complex building systems that are beyond the scope of whole-building programs like DOE-2 and EnergyPlus. VisualSPARK adds a graphical user interface to SPARK to simplify its use.

VISUALSPARK FREQUENTLY ASKED QUESTIONS – PART I

This is a list of frequently asked questions (FAQ) for VisualSPARK users. If you need help for something that is not covered by the SPARK Reference Manual, the SPARK Atomic Class API, the SPARK Problem Driver API, or this FAQ, please email us at sparksupport@SimulationResearch.lbl.gov. This FAQ is intended to supplement, not replace, the SPARK documentation. Before emailing us a question, you should first check to see if the topic is covered in the various manuals. Free documentation is downloadable from http://SimulationResearch.lbl.gov/.

Contents

SPARK Language

How to port atomic classes from SPARK 1.x to SPARK 2.x?

The syntax used to define the callback functions has changed in SPARK 2.x in order to support new features such as the extended callback mechanism and the multivalued inverses. In order to make the definition of the callback functions easier to use, C preprocessor macros that hide the implementation details of argument passing as well as the function prototype have been defined in the header file spark.h. Note that the syntax of the EQUATIONS and FUNCTIONS blocks in the parsed portion of the atomic class has not changed.

To demonstrate how to use the new macros we will show how to port the simple sum.cc atomic class from SPARK 1.x to SPARK 2.x. The next code snippet shows this atomic class as found in the globalclass directory of the SPARK 1.02 distribution.

```
#ifdef SPARK PARSER

PORT a    "Summand 1" ;
PORT b    "Summand 2" ;
PORT c    "Sum" ;

EQUATIONS {
  c = a + b ;
}

FUNCTIONS {
  a = sum subtract( c, b ) ;
  b = sum subtract( c, a ) ;
  c = sum add( a, b ) ;
}

#endif /* SPARK PARSER */
#include "spark.h"

double sum subtract ( ARGS )
{
    ARGDEF(0, c);
    ARGDEF(1, b);
    double a;

    a = c - b;
    return a ;
}

double sum add ( ARGS )
{
    ARGDEF(0, a);
    ARGDEF(1, b);
    double c;

    c = a + b;
    return c;
}
```

In order to port this atomic class to the syntax required by SPARK 2.x you need to:

- use the EVALUATE preprocessor macro to declare the callback function in place of the explicit function prototype, and
- use the RETURN preprocessor macro to return the new value for the target variables.

We used a bold, blue font to indicate the modifications in the implementation of the sum.cc atomic class required for SPARK 2.x.

```
#ifdef SPARK PARSER

PORT a    "Summand 1" ;
PORT b    "Summand 2" ;
PORT c    "Sum" ;

EQUATIONS {
  c = a + b ;
}

FUNCTIONS {
  a = sum subtract( c, b ) ;
  b = sum subtract( c, a ) ;
  c = sum add( a, b ) ;
}
```

```
#endif /* SPARK PARSER */
#include "spark.h"

EVALUATE( sum subtract )
{
    ARGDEF(0, c);
    ARGDEF(1, b);
    double a;

    a = c - b;
    RETURN( a );
}

EVALUATE( sum add )
{
    ARGDEF(0, a);
    ARGDEF(1, b);
    double c;

    c = a + b;
    RETURN( c );
}
```

As in SPARK 1.x the macro ARGDEF is used to define an argument through its 0-based position in the argument list as specified in the FUNCTIONS block. The position of the first element in the list is thus referred to with the index 0, the position of the second element with the index 1, and so forth.

A synonym macro named ARGUMENT, and with the same syntax, was added in SPARK 2.x. For example, the following code snippet defines the argument in position 0 in the argument list of the callback sum_subtract as a const reference to a `SPARK::TArgument` object named `c`. It is exactly equivalent to using the ARGDEF macro as in the previous code snippet.

```
EVALUATE( sum subtract )
{
    ARGUMENT(0, c);
    ...
}
```

In SPARK 2.x the RETURN macro must be used to return the value of the target variable in place of the C++ language keyword `return` followed by the new `double` value.

- If you keep using the `return` keyword followed by a `double` value then your atomic class will not compile because the new callback prototypes are defined as `void` functions.
- If you do not use the RETURN macro to return the new value of the target variable, then the new value will not be propagated to the target variable, which will remain constant throughout the simulation.

Finally if the atomic class you want to port to SPARK 2.x defines a PRED function in the FUNCTIONS block, along with the main EVALUATE callback function, then you need to:

- replace the PRED language keyword in the FUNCTIONS block with the new PREDICT language keyword,
- use the PREDICT preprocessor macro to declare the callback function in place of the explicit function prototype, and
- use the RETURN preprocessor macro to "return" the new value for the target variables.

How to return the calculated values in a multi-valued inverse?

The RETURN preprocessor macro mimics the C++ language keyword `return` used in SPARK 1.x. However, it worked only with single-valued inverses since it assumed that only one value at a time could be returned from a callback function. Therefore, we added a new mechanism in SPARK 2.x to return the new value for each target variable of a callback function. The target variables are the variables listed on the left-hand side of the "=" sign of each inverse specified in the FUNCTIONS block.

Unlike in SPARK 1.x it is now possible in SPARK 2.x to gain *write* access to the target variables in a similar way that the ARGUMENT macro grants you *read* access to the argument variables. This is accomplished with the preprocessor macro TARGET that uses syntax similar to the one of the ARGUMENT macro. For each target variable you need to specify:

- the 0-based position of the target variable in the target list (i.e., starting at zero for the first variable), and
- the name of the reference to the `SPARK::TTarget` object representing the target variable.

Assigning the new calculated value(s) to each target variable ensures the proper data flow across the set of the unknown variables in the problem. Note that instead of using the RETURN macro, it is equivalent to declare the target variable in the body of the callback function with the TARGET macro and assign the new value to it. Also, note that the `SPARK::TTarget` class does not grant you *read* access to the current value of the target variable as this would break the topological dependency implied in the FUNCTIONS block used during the graph-theoretic processing in setupcpp. However, it is possible to access all other properties of a target variable, such as its name, its unit, its past values, its absolute tolerance, ..., etc.

We demonstrate the usage of the TARGET macro in the following code snippet that shows the mulivalued atomic class root2.cc. This atomic class computes the two real roots of a quadratic polynomial. Consult Section 4 of the SPARK Reference Manual for more details on this atomic class and its implementation.

```
// root2.cc
// Multi-valued object that returns the 2 roots (root plus, root minus)
// of a quadratic polynomial of the form a*x^2 + b*x + c = 0
////////////////////////////////////////////////////////////////////////

#ifdef SPARK_PARSER

PORT a;
PORT b;
PORT c;
PORT root_plus;
PORT root_minus;

EQUATIONS {
    a*x^2 + b*x + c = 0;
}

FUNCTIONS {
    root_plus, root_minus = root2  mroot2( a, b, c ) ;
}

#endif /* SPARK_PARSER */
#include "spark.h"

EVALUATE( root2  mroot2 )
{
    ARGUMENT( 0, a );
```

```
    ARGUMENT( 1, b );
    ARGUMENT( 2, c );
    TARGET( 0, root plus );
    TARGET( 1, root minus );

    double discriminantx = b*b - 4.0*a*c;
    if (discriminantx < 0.0) { // Atomic class error
        REQUEST  ABORT( "Cannot compute complex roots." );
    }

    double square discriminant = sqrt( discriminantx );

    root plus = (-b + square discriminant)/(2.0*a);
    root minus = (-b - square discriminant)/(2.0*a);
}
```

Note that in case the discriminant is a negative real value, we send an abort request to the solver to interrupt the simulation. This error checking is necessary because SPARK does not have native support for complex numbers.

When are the PRED_FROM_LINK variables updated?

The PRED_FROM_LINK variables are the variables in the problem definition that are specified with the keyword PRED_FROM_LINK, like the variable linkA in the following code snippet.

```
LINK linkA ... PRED FROM LINK=linkB;
```

- The PRED_FROM_LINK mechanism is invoked only if the target variable linkA is a break variable in the problem under study.
- The PRED_FROM_LINK variables are updated at the beginning of each simulation step after the prepare step callbacks are fired for all components. In our example the current value of the fromLink variable linkB is assigned to the current value of the target variable linkA.
- This prediction mechanism for the break variables happens before any components are evaluated and only once per step. At the end of the step the predicted values are likely to have been overridden with the solution values computed for each break variable for the current step.

When are the INPUT_FROM_LINK variables updated?

The INPUT_FROM_LINK variables are the variables in the problem definition that are specified with the keyword INPUT_FROM_LINK, like the variable linkA in the following code snippet.

```
LINK linkA ... INPUT FROM LINK=linkB;
```

- INPUT_FROM_LINK variables are input variables to the computational model, i.e., they are not computed by the SPARK solver. In that sense they serve a similar role as the INPUT and PARAMETER variables, except that they get their values using a different mechanism.
- The current value of an INPUT_FROM_LINK variable will always be the value of the fromLink variable from the last successfully computed step, i.e., the value of linkA in our example will always be the value of linkB from one step back. Using the SPARK mechanism to access the first past value, linkA = linkB[1].
- This requires that, for the calculation of the initial step, the fromLink variable linkB must be specified at the previous time step in order to ensure consistent initialization. Then the value will be propagated correctly to the variable linkA when solving the initial step. Not that specifying a value directly for linkA in an input file, for example, is useless as the SPARK solver will always propagate the value from linkB, even at the initial step.
- When reporting an INPUT_FROM_LINK variable at the end of a successful step the reported

value will show the current value of the fromLink variable although this is not the value that was used for the computation. The value used for the computation is always the value from the fromLink variable from one step back.

How to write comments in the SPARK-readable files?

Comments can be specified in input files, preference files (*.prf) and run-control files (*.run) in SPARK by using either one of the two supported formats:

- /* C-style comments */
- // C++-style comments

Note that C-style comments can spread over multiple lines. C++-style comments extend to the end of the current line only. However, there is one major limitation of the SPARK comment mechanism, whereby each starting comment delimiter like '/*' or '//' must be preceded either by a white space (to indicate a new token) or be specified at the beginning of a line. This is a major difference with the behavior of the C preprocessor. A comment delimiter appearing within a token will not be detected by SPARK.

```
// this a valid comment
some text// this is NOT a comment
some more text //another valid comment
some more text/* this is NOT a valid comment*/ more text after comment
some more text /* this is a valid comment*/ more text after comment
some more text /* this is still a valid comment*/more text after comment
```

Which problem variables are reported at runtime?

Only the variables specified with the REPORT keyword in the problem definition files (i.e., *.pr, *.cm, and *.cc files) or the variables with a write URL containing the REPORT tag will be reported. You can also use the map file mechanism to override the static write URLs for any problem variables.

How to specify the values in the native input files?

SPARK native input files must respect a predetermined format to ensure correct parsing at runtime.

- The first non-comment line in the file must specify the number of columns and then provide a string without space for each column.
- The following lines list the values for each column for the time stamp specified in the first column. Note that the time stamps must be specified in strictly increasing order.
- The values for each time stamp must be specified on the same row. It is possible to have comments in the row as long as they don't span over multiple lines.
- The '*' meta-character in the first column indicates that the same values as at the previous time stamp are to be kept until the end of the simulation.
- Similarly, the '*' meta-character specified in any column but the first one after the first row indicates that the previous value should be repeated for this column. E.g., in the following example of an input file, the variable y will also get the value 2.0 for the time stamp 1.0. It is illegal to use the '*' character in the first row since there are no previous value to propagate. This will produce a runtime error.

```
// Various comments pertaining to the time-varying values
3         x         y         z
0.0       1.0       2.0       3.0
1.0       1.1       *         2.4
// More comments possible
3.0       *         5.0       3.0    /* Comments */
*
```

- Finally, the '*' meta-character used as the first and only time stamp indicates that the values are constant for the entire duration of the simulation.

```
// List of constant values
2         A         B
*         1.0       2.0
```

Of course, when using the '*' meta-character as the first time stamp it is no longer possible to specify the '*' meta-character in place of a value as all values must be explicitly specified.

The next issue of the *Building Energy Simulation User* News will feature FAQs about SPARK Runtime Controls, SPARK Preference Settings and SPARK Requests

SPARK is an equation-based simulation environment that allows you to build customized models of complex physical processes by connecting calculation objects that represent system components like walls, fans, heat exchangers, chillers, ducts, mixing boxes, controls, etc. It is aimed at the simulation of innovative and/or complex building systems that are beyond the scope of whole-building programs like DOE-2 and EnergyPlus. VisualSPARK adds a graphical user interface to SPARK to simplify its use.

Download VisualSPARK free of charge from

http://SimulationResearch.lbl.gov > visualpark/

Please go to our website to download this new VisualSPARK documentation:

- **New Features, Bug Fixes, and Changes**
- **Frequently Asked Questions**
- **How To Port Atomic Classes To SPARK 2.*x***
- **Theoretical Speed-Up Using SPARK**

Don't forget the SPARK Reference Manual and VisualSPARK Users Guide

SPARK was developed by the Simulation Research Group at Lawrence Berkeley National Laboratory and by Ayres Sowell Associates, with Support from the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies Program of the U.S. Department of Energy, Program Manager Dru Crawley.

GenOpt 2.0
Generic Optimization Program

GenOpt *is an optimization program for the minimization of a cost function, such as annual energy use, that is evaluated by an external simulation program.*

GenOpt *can be used with any simulation program -- such as EnergyPlus, SPARK or DOE-2 -- that has text-based input and output. It also offers an interface for adding custom optimization algorithms to its library.*

New GenOpt Technical Reports

- *Generalized Pattern Search Algorithms with Adaptive Precision Function Evaluations* by Elijah Polak and Michael Wetter (click on the title to download the document)
- *Comparison of a Generalized Pattern Search and a Genetic Algorithm Optimization Method*, by Michael Wetter and Jonathan Wright (click on the title to download the document)
- *A Convergent Optimization Method Using Pattern Search Algorithms with Adaptive Precision Simulation*, by Michael Wetter and Elijah Polak (click on the title to download the document)

All reports and GenOpt documentation are available free of charge from http://SimulationResearch.lbl.gov/

New features of Version 2.0

Capability to Process Discrete Independent Variables

GenOpt can now process discrete independent variables, such as different window constructions, either for optimization problems with mixed discrete and continuous independent variables or for doing parametric studies.

New Optimization Algorithms

The following optimization algorithms are new in *GenOpt 2.0*:

- `GPSCoordinateSearch` and `GPSHookeJeeves`: These algorithms are members of the family of Generalized Pattern Search (GPS) algorithms. They can be used to solve optimization problems with continuous independent variables. Both algorithms can be run using multiple starting points to increase the chance of finding the global minimum if the cost function has several local minima.
- `DiscreteArmijoGradient`: An algorithm that approximates gradients by finite differences and uses the Armijo line search algorithm.
- `PSOCC`, `PSOCCMesh`, and `PSOIW`: These algorithms are members of the family of Particle Swarm Optimization algorithms (which are global heuristic optimization algorithms). They can be used to solve optimization problems with continuous and/or discrete independent variables.
- `GPSPSOCCHJ`: This is a hybrid global optimization algorithm that starts by performing a Particle Swarm Optimization for continuous and discrete independent variables and then switches to the Hooke-Jeeves Generalized Pattern Search algorithm to refine the continuous independent variables.

Pre- and Post-Processing

Some simulation programs, such as EnergyPlus, cannot pre-process the independent variables or post-process values that are computed during the simulation. For such situations, *input function objects* and *output function objects* can now be used without having to modify *GenOpt's* source code.

GenOpt 2.0 **(with documentation) may be downloaded free of charge from**

http://SimulationResearch.lbl.gov > GenOpt

Recent Reports

Download document *Proc., 8th Int'l IBPSA Conference, Eindhoven, Netherlands August 11-14, 2003*

Comparison of a Generalized Pattern Search and a Genetic Algorithm Optimization Method

Michael Wetter (mwetter@lbl.gov)
Simulation Research Group
Building Technologies Department
Environmental Energy Technologies Division Lawrence
Berkeley National Laboratory
Berkeley, CA 94720, USA

and

Jonathan Wright
Department of Civil and Building Engineering,
Loughborough University
Loughborough
Leicestershire, LE11 3TU, UK

ABSTRACT
Building and HVAC system design can significantly improve if numerical optimization is used. However, if a cost function that is smooth in the design parameter is evaluated by a building energy simulation program, it usually becomes replaced with a numerical approximation that is discontinuous in the design parameter. Moreover, many building simulation programs do not allow obtaining an error bound for the numerical approximations to the cost function. Thus, if a cost function is evaluated by such a program, optimization algorithms that depend on smoothness of the cost function can fail far from a minimum. For such problems it is unclear how the Hooke-Jeeves Generalized Pattern Search optimization algorithm and the simple Genetic Algorithm perform. The Hooke-Jeeves algorithm depends on smoothness of the cost function, whereas the simple Genetic Algorithm may not even converge if the cost function is smooth. Therefore, we are interested in how these algorithms perform if used in conjunction with a cost function evaluated by a building energy simulation program. In this paper we show what can be expected from the two algorithms and compare their performance in minimizing the annual primary energy consumption of an office building in three locations. The problem has 13 design parameters and the cost function has large discontinuities. The optimization algorithms reduce the energy consumption by 7% to 32%, depending on the building location. Given the short labor time to set up the optimization problems, such reductions can yield considerable economic gains. the cost function, such as annual energy use, is computationally expensive, defined on continuous design parameters, and evaluated by the whole-building energy simulation program EnergyPlus (Crawley et al. 2001). A similar discussion applies if other system simulation programs, such as TRNSYS or COMIS, are used to evaluate the cost function. Such programs replace the cost function with a numerical approximation that is discontinuous in the design parameters. Furthermore, their solvers are usually implemented in such a way that establishing error bounds is not possible. In these situations optimization can only be applied heuristically, and it is not clear how the HJ algorithm and the sGA perform if used in conjunction with such discontinuous cost functions. First, we define the optimization problem and discuss the properties of the cost function. Next, we describe what can be expected from the optimization algorithms and show why the two algorithms are likely to yield different results. Then, we present numerical experiments that compare the performance of the algorithms. We analyze the observed numerical problems and close by proposing the use of a hybrid optimization algorithm.

LBNL-53139 (download document)

Moisture Effects on Conduction Loads

N. Mendes
Pontifical Catholic Univ., Paraná
Dept. Mechanical Engrg.
Thermal Systems Laboratory
Curitiba-PR, 80.215-901, Brazil

F. C. Winkelmann
Lawrence Berkeley National Laboratory
Environmental Energy Technologies Div.
Simulation Research Group
Berkeley – CA, 94720, USA

R. Lamberts and P. C. Philippi
Federal University of Santa Catarina
Florianópolis-SC
88.040-900
Brazil

ABSTRACT
The effects of moisture on sensible and latent conduction loads are shown by using a heat and mass transfer model with variable material properties, under varying boundary conditions. This model was then simplified to reduce calculation time and used to predict conduction peak load and yearly integrated wall conduction heat flux in three different cities: Singapore (hot/humid), Seattle (cold/humid) and Phoenix (hot/dry). The room air temperature and relative humidity were calculated with the building energy simulation program DOE-2.1E. The materials studied were aerated cellular concrete, brick, lime mortar and wood. It is shown that the effects of moisture can be very significant and that simplified mathematical models can reduce the calculation time with varying effects on accuracy. [DOE-2 related report]

Recent Reports

Download document *Proc., 8th Int'l IBPSA Conference, Eindhoven, Netherlands August 11-14, 2003*

A Convergent Optimization Method Using Pattern Search Algorithms With Adaptive Precision Simulation

Michael Wetter
Simulation Research Group
Building Technologies Department
Environmental Energy Technologies Division
Lawrence Berkeley National Laboratory,
Berkeley, CA 94720, USA

and

Elijah Polak
Department of Electrical Engineering
University of California at Berkeley
Berkeley, CA 94720, USA

ABSTRACT
Thermal building simulation programs, such as EnergyPlus, approximate solutions of a differential algebraic system of equations. While the theoretical solution is usually continuously differentiable in the building design parameters, the approximate solutions may not even be continuous, due to adaptive variations in solver iterations and the use of adaptive integration meshes. Hence, when a smooth cost function, defined on the design parameters, is evaluated using a thermal building simulation program, it becomes replaced with an approximation that fails to be even continuous. Consequently, when used in conjunction with an optimization algorithm that depends on smoothness of the cost function, the algorithm is quite likely to jam at a non-optimal point. Obviously, in such situations, the potential economic gains that optimization offers are not attained. As an illustration, we present an example, using the EnergyPlus whole building energy simulation program to evaluate our cost function, in which the Hooke-Jeeves algorithm terminates at a non-stationary point. To prevent such failures, we have developed an adaptive simulation precision control algorithm that can be used in conjunction with a family of derivative free optimization algorithms. The resulting composite algorithms are demonstrably convergent to an exact stationary point. We present the main ingredients of the composite algorithms and show by numerical experiments that using coarse approximations in the early iterations can significantly reduce the computation time.

Download document

Generalized Pattern Search Algorithms with Adaptive Precision Function Evaluations

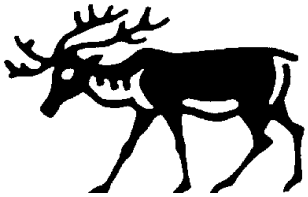
Elijah Polak
Department of Electrical Engineering
University of California at Berkeley
Berkeley, CA 94720, USA

and

Michael Wetter (mwetter@lbl.gov)
Simulation Research Group
Building Technologies Department
Environmental Energy Technologies Division
Lawrence Berkeley National Laboratory,
Berkeley, CA 94720, USA

ABSTRACT
In the literature on generalized pattern search algorithms, convergence to a stationary point of a once continuously differentiable cost function is established under the assumption that the cost function can be evaluated exactly. However, there is a large class of engineering problems where the numerical evaluation of the cost function involves the solution of systems of differential algebraic equations. Since the termination criteria of the numerical solvers often depend on the design parameters, computer code for solving these systems usually defines a numerical approximation to the cost function that is discontinuous with respect to the design parameters. Standard generalized pattern search algorithms have been applied heuristically to such problems, but no convergence properties have been stated. In this paper we extend a class of generalized pattern search algorithms to a form that uses adaptive precision approximations to the cost function. These numerical approximations need not define a continuous function. Our algorithms can be used for solving linearly constrained problems with cost functions that are at least locally Lipschitz continuous. Assuming that the cost function is smooth, we prove that our algorithms converge to a stationary point. Under the weaker assumption that the cost function is only locally Lipschitz continuous, we show that our algorithms converge to points at which the Clarke generalized directional derivatives are nonnegative in predefined directions. An important feature of our adaptive precision scheme is the use of coarse approximations in the early iterations, with the approximation precision controlled by a test. Such an approach leads to substantial time savings in minimizing computationally expensive functions.

DOE-2



DOE-2.1E (v. 121) 1,000-Zone version for Windows from ESTSC; other vendors of DOE-2 based programs are listed on our website: http://SimulationResearch.lbl.gov/.

Cost is as follows:

$ 300 U.S. Government, non-profit Educational
$ 575 U.S., Mexico, Canada
$ 1075 All Other Non-U.S. (except in Japan, where the price is $1268)

DOE-2 Documentation on a CD from ESTSC - Cost US$100

What is included on the CD?

- DOE-2 Reference Manual (Part 1)
- DOE-2 Reference Manual (Part 2)
- DOE-2 Supplement to the Reference Manual (2.1E)
- DOE-2 BDL Summary (2.1E)
- DOE-2 Engineers Manual (2.1A)

Order Software and ESTSC Documentation

Ed Kidd or Kim Buckner
NCI Information Systems, Inc.
Energy Science and Technology Software Center (ESTSC)
P.O. Box 1020
Oak Ridge, TN 37831

Phone: 865/576-1037
Fax: 865/576-6436
Email: estsc@adonis.osti.gov

Purchase DOE-2 Documentation

DOE-2 Sample Run Book (2.1E) -- The Sample Run book is the only remaining DOE-2 manual not available electronically. It must be purchased separately from NTIS; ordering information may be found at http://SimulationResearch.lbl.gov > DOE-2 > Documentation

Free DOE-2 Documentation (http://SimulationResearch.lbl.gov > DOE-2 > Documentation)

DOE-2 Basics Manual (2.1E)

Update Packages: Update Packages are **not** cumulative; each one contains different information. Download all four packages then print and insert the pages into your existing DOE-2 manuals.

- Update Package #1: DOE-2.1E Basics, the Supplement and BDL Summary
- Update Package #2: BDL Summary and Supplement.
- Update Package #3: Appendix A of the Supplement.
- Update Package #4: (1000-zone DOE-2.1E) BDL Summary.

DOE-2 Modeling Tips (pdf files) for 2003 for 2002
A compilation of all the "how to" and "DOE-2 Puzzler" articles from the *Building Energy Simulation User News.*

Changes and Bug Fixes to DOE-2.1E (txt file)
Description of all changes and bug fixes in a text document.

DOE-2 listings are continued on the next page

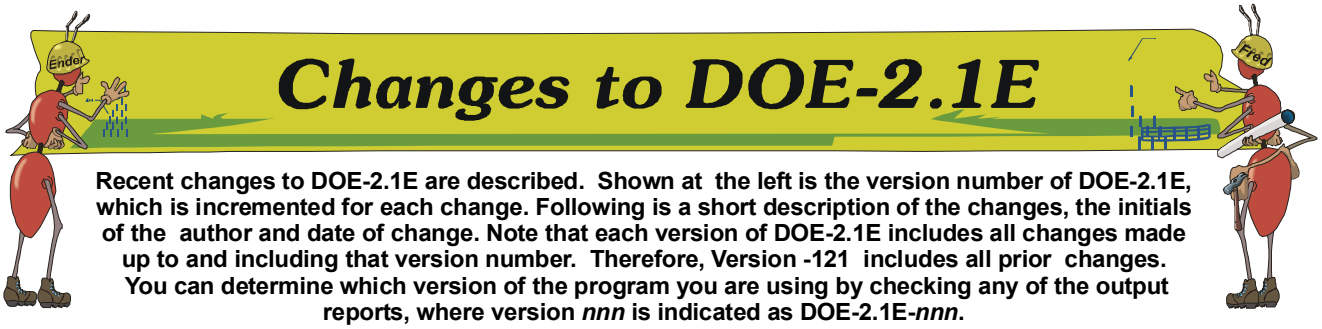
DOE-2 (continued)

DOE-2 Training

Private or group DOE-2 courses for beginning and advanced users.
Contact Marlin Addison at (602) 968-2040, marlin.addison@doe2.com

DOE-2 Help Desk

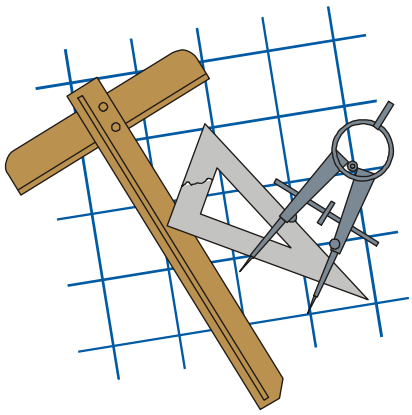
Email (klellington@lbl.gov) or fax the Simulation Research Group with your questions. Fax: (510) 486-4089



Recent changes to DOE-2.1E are described. Shown at the left is the version number of DOE-2.1E, which is incremented for each change. Following is a short description of the changes, the initials of the author and date of change. Note that each version of DOE-2.1E includes all changes made up to and including that version number. Therefore, Version -121 includes all prior changes. You can determine which version of the program you are using by checking any of the output reports, where version *nnn* is indicated as DOE-2.1E-*nnn*.

-121 : dkey drlc sys The units of TWR-DESIGN-APPROACH were fixed in Systems and Plant-Assignment, but not in Plant. The units should be R (for ΔT) not F. This makes the fix in Plant. In addition, the same unit error occurs for TWR-THROTTLE in Systems and Plant-Assignment; and for TWR-DESIGN-RANGE in Plant. A search revealed several other ΔT-type keywords with the wrong units. The units of RANGE and APPROACH in the hourly output of the cooling tower were incorrectly set to F rather than R. This makes the fix. [FB 20030922]

To access the complete bug/change list for DOE-2.1E, please visit our web site at http://SimulationResearch.lbl.gov and click on "Technical Reports" in the left menu. Under the DOE-2 heading, click on "bug fixes" or click here for the .txt file.



Building Energy Tools Directory

The web-based Building Energy Tools Directory contains information on more than 270 building-related software tools from around the world.

For each tool in the directory, a short description is provided, along with information about technical expertise required, users, audience, input, output, validation, computer platforms, programming language, strengths, weaknesses, technical contact, availability and cost. A link is also provided for directly translating the web pages into more than eight languages.

Know of a tool (yours?) that isn't in the directory? Visit http://www.eere.energy.gov/buildings/tools_directory/your_software_here.html or contact Dru Crawley at Drury.Crawley@ee.doe.gov.

ENERGY-10, VERSION 1.6

ENERGY-10 is a design tool for smaller residential or commercial buildings that are less than 10,000 ft^2 or buildings that can be treated as 1- or 2-zone increments. It performs whole-building energy analysis for 8760 hours/year, including dynamic thermal and daylighting calculations. ENERGY-10 was specifically designed to facilitate the evaluation of energy-efficient building features in the very early stages of the design process.

Version 1.6 Upgrades

Synchronize Libraries
Libraries may now be associated with more than one building.

Free Run Mode
Automated process of monitoring how a building operates without any HVAC system.

Clear All Internal Gains
The name is self-explanatory.

New Buttons on Provisional Data Dialog Box
Users may specify whether they want autobuild HVAC sizing to be computed with or without daylighting.

Performance Summary Reports
Three performance summary reports have been added. One is a simple performance summary, which breaks down the standard summary into more readable chunks and adds a column that reflects the percentage change of going from Building 1 to Building 2. The other two are daylighting reports that show the standard daylighting factor calculated for each lighting zone.

NewDefaults Library
A new set of libraries contains all the standard libraries such as floorlib, rooflib, etc. with updated values.

Registry Path for ENERGY-10 Data
New registry path allows users to maintain separate copies of the three most recent versions of *ENERGY-10*. In addition, the installation script allows installation for either "all users" or the "current user only."

Additional Tutorials on Installation CD
Three new tutorials are included in the slide show section of the installation CD, including Economics, Daylighting, and Using *ENERGY-10* in the Design Process.

Douglas K. Schroeder
1331 H Street N.W., #1000
Washington, DC 20004

Tel: 202.628.7400 ext 210
Fax: 202.383.5043
www.sbicouncil.org

Sustainable Buildings Industry Council (SBIC)

ENERGY-10 User Group www.sbicouncil.org/forum

SBIC Bookstore www.sbicouncil.org/store/resources.php - pubs

Riches adorn a house while virtue adorns a person.

BLAST*news*

www.bso.uiuc.edu

Building Systems Laboratory
University of Illinois, 30 Mechanical Engineering Building,
1206 West Green Street, Urbana, IL 61801
Tel: (217) 333-3977 - Fax: (217) 244-6534
support@blast.bso.uiuc.edu

The **Building Loads Analysis and System Thermodynamics (BLAST** program predicts energy consumption, energy system performance and cost for new or existing (pre-retrofit) buildings.

BLAST contains three major sub-programs:

- **Space Load Prediction** computes hourly space loads in a building based on weather data and user inputs detailing the building construction and operation.
- **Air Distribution System Simulation** uses the computed space loads, weather data, and user inputs.
- **Central Plant Simulation** computes monthly and annual fuel and electrical power consumption.

Heat Balance Loads Calculator (HBLC)
The BLAST graphical interface (HBLC) is a Windows-based interactive program for producing BLAST input files. You can download a demo version of HBLC (for MS Windows) from the BLAST web site (User manual included).

HBLC/BLAST Training Courses
Experience with the HBLC and the BLAST family of programs has shown that new users can benefit from a session of structured training with the software. The Building Systems Laboratory offers such training courses on an as needed basis typically at our offices in Urbana, Illinois.

WINLCCID 98
LCCID (Life Cycle Cost in Design) was developed to perform Life Cycle Cost Analyses (LCCA) for the Department of Defense and their contractors.

To order BLAST-related products, contact the Building Systems Laboratory at the address above.

Program Name	Order Number	Price
PC BLAST Includes: BLAST, HBLC, BTEXT, WIFE, CHILLER, Report Writer, Report Writer File Generator, Comfort Report program, Weather File Reporting Program, Control Profile Macros for Lotus or Symphony, and the Design Week Program. The package is on a single CD-ROM and includes soft copies of the BLAST Manual, 65 technical articles and theses related to BLAST, nearly 400 processed weather files with a browsing engine, and complete source code for BLAST, HBLC, etc. Requires an IBM PC 486/Pentium II or compatible running MS Windows 95/98/NT.	3B486E3-0898	$1500
PC BLAST Package Upgrade from level 295+	4B486E3-0898	$450
WINLCCID 98: executable version for 386/486/Pentium	3LCC3-0898	$295
WINLCCID 98: update from WINLCCID 97	4LCC3-0898	$195

The last four digits of the catalog number indicate the month and year the item was released or published. This will enable you to see if you have the most recent version. All software will be shipped on 3.5" high density floppy disks unless noted otherwise.

Building Energy Software

from the Environmental Energy Technologies Division of Lawrence Berkeley Laboratory

Free Downloads	
BDA 3.0 (Building Design Advisor) (building decision-making from design through completion)	**gaia.lbl.gov/BDA**
COMIS (multi-zone air flow and contaminant transport model)	**www-epb.lbl.gov/comis**
EnergyPlus 1.1.1 (new-generation whole-building energy analysis program, based on BLAST and DOE-2)	**http://www.energyplus.gov/ --or-- SimulationResearch.lbl.gov > EnergyPlus**
GenOpt® 2.0 (generic optimization program)	**SimulationResearch.lbl.gov > GenOpt**
Optics 5.1.02 (for analyzing optical properties of glazing systems)	**windows.lbl.gov/materials/optics5/**
RADIANCE 3.5 (analysis and visualization of lighting in design)	**radsite.lbl.gov/radiance/**
Desktop Radiance 2.0β (integrates the Radiance Synthetic Imaging System with AutoCAD Release 14)	**radsite.lbl.gov/deskrad/**
Radiance Control Panel (automates some Radiance tasks once the model has been created)	**www.squ1.com/site.html**
RESEM (Retrofit Energy Savings Estimation Model) (calculates long-term energy savings directly from actual utility data)	**eetd.lbl.gov/btp/resem.htm**
SUPERLITE (calculates illuminance distribution for room geometries)	**eetd.lbl.gov/btp/superlite2.html**
THERM 5.2 (models two-dimensional heat-transfer effects in building components where thermal bridges are of concern)	**windows.lbl.gov/software/therm/therm.html**
VisualSPARK 2.0 (Simulation Problem Analysis and Research Kernel) (connect component models to simulate innovative building envelope and HVAC systems)	**SimulationResearch.lbl.gov > VisualSPARK**
WINDOW 5.2 (thermal analysis of window products)	**windows.lbl.gov/software/window/window.html**
Free Software / Request by Fax from 510.486.4089	
RESFEN 3.1 (choose energy-efficient, cost-effective windows for a given residential application)	**windows.lbl.gov/software/resfen/resfen.html**
Web Based (free)	
Home Energy Saver (quickly computes home energy use) and ***Home Improvement Tool*** (simplified Home Energy Saver)	**hes.lbl.gov** and **hit.lbl.gov**