NEW FEATURES OF THE DOE-2.1C ENERGY ANALYSIS PROGRAM

W. F. Buhl, A. E. Erdem, J. H. Eto, J. J. Hirsch, F. C. Winkelmann

Applied Science Division
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

ABSTRACT

Five new features of the DOE-2.1C building energy simulation program are presented. The features are: (1) user input of new algorithms in LOADS; (2) a sunspace-atrium model; (3) simulation of the powered induction unit HVAC system; (4) cogeneration simulation in PLANT; and (5) a new ECONOMICS program able to handle most utility rate schedules. The capabilities of each new feature are outlined and some details of the implementation or simulation techniques of each feature area briefly discussed.

INTRODUCTION

DOE-2 is a large public domain computer program that simulates energy use in buildings. The program can handle a wide range of building types, HVAC systems and primary equipment at varying levels of detail. The program is extensively documented in various publications available from the National Technical Information Service [2-6]. Shorter descriptions of the program can be found in some publications of the Building Energy Simulation Group at Lawrence Berkeley Laboratory [7-10]. Periodically, new versions of the program are issued which correct problems discovered in older versions and add new features that are needed by the user community. The latest version, DOE-2.1C, contains the enhancements discussed in this paper. Detailed input descriptions of these new features can be found in the DOE-2 Supplement [10].

FUNCTIONS

The DOE-2 simulation program consists of 9 modules with about 73000 lines of FORTRAN source code. Although, with regular BDL input a user can describe many simulation tasks, there are cases when standard capabilities are not enough. Examples of these might be: (1) treatment of building components such as windows with special properties; (2) use of different algorithms for some types of calculation; (3) computation of coefficients or intermediate values that are used in the regular simulation algorithms; (4) input of values that are not allowed by the standard BDL rules; (5) output of values that are intermediate results or are a combination of calculated values; or (6) quickly testing certain algorithms without modifying and recompiling the BDL and simulation programs. In order to allow these broader simulation capabilities we have implemented the "functions" feature in the DOE-2.1C LOADS program.

Using BDL, one writes in a subset of the FORTRAN language a subroutine that is similar to the ones in the LOADS simulation program. This subroutine has the capability of accessing and modifying all the variables that are used in the regular simulation modules. This newly added subroutine is triggered by BDL keywords in each component of the LOADS input. For example, if in the WINDOW command we say FUNCTION=(*FUNC1*,*FUNC2*), in the simulation phase every time the window calculations are done for that window the function FUNC1 will be triggered before the standard DOE-2 window computation, and FUNC2 will be triggered after the standard DOE-2 window computation. FUNC1 may be used to preset some of the constants and variables that are used in the regular DOE-2 algorithms. FUNC2 may be used to modify or completely recalculate the results that are computed by the regular DOE-2 algorithms.

The specifications of FUNC1 and FUNC2 follow the usual LOADS input. They start by using the ASSIGN statement to associate the simulation variables (that the function will use or alter) with the local function variables. This stage looks redundant, but is needed because some variables that are used in the simulation are pseudo-FORTRAN variables that may be more than 6 characters in length, and may contain embedded dashes. The ASSIGN statement also causes BDL to produce the necessary information to access these variables. Any variable inside the DOE-2 calculations can be accessed by an offset with respect to the beginning location of a labelled common block, or by a constant offset plus a base pointer with respect to the blank common block. With the ASSIGN statement, the local variable that will be used inside the function gets this access information.

Inside the computation section of the function, the local variables mentioned in the ASSIGN statement can be used in algebraic or logical expressions, or in I/O statements, just as in FORTRAN. These expressions are converted into sequential instructions to be evaluated at simulation time. The instructions are operator-operand pairs that include arithmetic, logical, I/O, and goto operations that are used in a general purpose computer.

In the simulation phase, whenever the FUNCTION keyword is used the following events take place. The appropriate function description block is accessed. By looking at the variables table, and using the access information in it, the values to be used in the function module are fetched from the DOE-2 simulation data structure into the local variables mentioned in the function. The next step involves the step by step execution of the instructions that define the arithmetic, logical, and I/O statements of the user input. After this, and before returning to the regular DOE-2 simulation stream, the values that were changed by the function are returned back to the DOE-2 simulation data-structure, again using the access information belonging to this function's variables table.

To give an idea of how this system works we will explain the actions taken by the BDL part of the functions processor in the following example . This example is taken from the DOE-2 Supplement Manual, Version 2.1C.

This function re-defines the outside film heat transfer coefficient of an external wall so that wind direction is accounted for. It uses Kimura's algorithm to get local windspeed at the wall surface as a function of weather file wind speed and orientation of wall with respect to the global wind direction. (Kimura, Scientific Basis of Air Conditioning, 1977, p. 85)

```
In this example the annotations after the | sign are not part of input data.

EW-2= EXTERIOR-WALL                    | applicable exterior wall
       .
       .
       .
    FUNCTION=(*FNEW2*,*NONE*) ..   | we mention the FUNCTION keyword with two
       .                           | literal values FNEW2 and NONE.  This
       .                           | establishes a link between the function
       .                           | named FNEW2 to be described later and the
       .                           | current EXTERIOR-WALL.  Note that FNEW2
       .                           | is a 'before-function', and that the
       .                           |'after-function' is not used.
       .
    { rest of LDL input }
END ..
FUNCTION     NAME=FNEW2   LEVEL=EXTERIOR-WALL  .. | function definition
                                                  | starts here.
ASSIGN                    | This command establishes links between variables
                          | used in simulation algorithms and the ones used
                          | in calculation section of the function.

          WS = WNDSPD     | local variable WS is linked to weather data
                          | variable WNDSPD.  Functions processor now makes
                          | the connection and produces an access information.
                          | In this case the access information looks like:
                          | common block=/WEATH/ , offset=28
                          | which means that WS is attached to 28th location
                          | of the common block /WEATH/ .

          WD = WNDDRR     | Like before with common block=/WEATH/ , offset=31

          WA = XSAZM      | This variable contains surface azimuth of the
                          | exterior-wall from where the function is called.
                          | The access information looks like:
                          | common block=// , pointer=MX , offset=33
                          | that corresponds to the simulation variable
                          | <XSAZM> which is a so-called EDTT variable and
                          | same as AA(MX+33) in LOADS simulation source code.

          FU = FILMU   .. | This is the variable modified by the function
                          | FU is linked to FILMU which is the outside-
                          | air-film u-value of the exterior-wall.
                          | Its access information is:
                          | common block=/WALLV/ , offset=3

CALCULATE ..              | This command tells to functions processor that
                          | the following input contains the subset-FORTRAN
                          | computation section.  From now on the input
                          | looks very much like a FORTRAN subprogram.
                          | Comments start with character C in column 1,
                          | statement numbers are in columns 1-5, column 6
                          | is non-blank for continuation lines,
                          | and statements must be in columns 7-72.
                          | These statements are converted into instructions
                          | which will be executed in simulation time.
                          | The instructions are operator-operand pairs
                          | The generated instructions are listed in the
                          | following lines.

      RWD = WA + 3.1415 - WD                     |   01: LOAD          WA
                                                 |   02: ADD           3.1415
                                                 |   03: SUBTRACT      WD
                                                 |   04: STORE         RWD
      IF(RWD .GT. 3.1415) RWD = 6.283 - RWD      |   05: LOAD          RWD
                                                 |   06: SUBTRACT      3.1415
                                                 |   07: GO_IF_NOT_GT    11:
                                                 |   08: LOAD          6.283
                                                 |   09: SUBTRACT      RWD
                                                 |   10: STORE         RWD
C CONVERT GLOBAL WINDSPEED FROM KNOTS TO M/S
      VV = .553*WS                               |   11: LOAD          .553
                                                 |   12: MULTIPLY      WS
                                                 |   13: STORE         VV
C GET WINDSPEED AT WALL SURFACE
      IF( RWD .GE. 1.5708 )  GOTO  100           |   14: LOAD          RWD
                                                 |   15: SUBTRACT      1.5708
                                                 |   16: GO_IF_NOT_GE    18:
                                                 |   17: GOTO            30:
         IF( VV .GT. 2. )  VC = .25*VV           |   18: LOAD          VV
                                                 |   19: SUBTRACT      2.
                                                 |   20: GO_IF_NOT_GT    24:
                                                 |   21: LOAD          .25
                                                 |   22: MULTIPLY      VV
                                                 |   23: STORE         VC
         IF( VV .LE. 2. )  VC = .5               |   24: LOAD          VV
                                                 |   25: SUBTRACT      2.
                                                 |   26: GO_IF_NOT_LE    29:
                                                 |   27: LOAD          .5
                                                 |   28: STORE         VC
         GOTO  200                               |   29: GOTO            34:
100   VC = .3 + .05*VV                           |   30: LOAD          .05
                                                 |   31: MULTIPLY      VV
                                                 |   32: ADD           .3
                                                 |   33: STORE         VC

C  CONVERT BACK TO KNOTS
200   VC = 1.808*VC                              |   34: LOAD          1.808
                                                 |   35: MULTIPLY      VC
                                                 |   36: STORE         VC
C  COMBINED CONVECTIVE PLUS RADIATIVE CONDUCTANCE FOR
C  ROUGHNESS = 3 (SEE ENGINEERS MANUAL, P.III.59)
      FU = 1.90 + .38*VC                         |   37: LOAD          .38
                                                 |   38: MULTIPLY      VC
                                                 |   39: ADD           1.90
                                                 |   40: STORE         FU
      END                              | signals end of computation section of
                                       | function input.
  END-FUNCTION ..                      | signals end of functions input.
```

We have seen how the BDL part of the functions processor enters the access information and the generated instructions into the standard file.

When the exterior-wall simulation starts the EW-2 calculation, values specific to that wall are precalculated (FILMU is one of them). Then the functions interpreter is triggered for FNEW2 which is a "before-function". The functions interpreter first looks at the variable link table and moves the values from the LOADS data structure to the value area that will be referenced by the function. The link-table now looks like this:

WS	/WEATH/	+28	(value)
WD	/WEATH/	+31	(value)
WA	// +MX	+33	(value)
FU	/WALLV/	+ 3	(value)

Next, the interpreter executes the instructions block of the function FNEW2; it uses the operand values that are inside the link-table. Before returning to the regular DOE-2 exterior-wall simulation algorithm, the values in the link-table are put back into the LOADS data structure. Note that FILMU = FU now contains the value calculated by the function. The exterior-wall heat transfer computation continues with standard DOE-2 algorithms which use the new value of FILMU.

SUNSPACE/ATRIUM MODEL

In a collaboration with the RAMSES group at the University of Paris-South, new capabilities have been added to DOE-2.1C to allow simulation of residential sunspaces (such as attached greenhouses) and commercial atria. This work was stimulated by the fact that sunspaces and atria are currently very popular architectural elements, and, being quite complex from a thermal point of view, require an accurate energy analysis to produce a design which is energy efficient and comfortable year-round.

The new algorithms allow simulation of different forms of heat transfer between the sunspace (or atrium) and adjacent rooms which could not be calculated with DOE-2.1B. These include (see Fig. 1):

(1) solar gain through interior windows or openings,
(2) convection through vents or an open doorway,
(3) delayed conduction through heavy interior walls,
(4) conduction through interior windows.

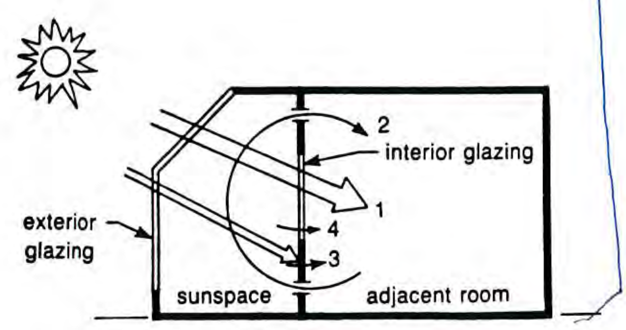


Figure 1. Cross section showing different forms of heat transfer between a sunspace and an adjacent room: (1) solar radiation passing from sunspace to room through interior glazing; (2) natural convection by thermocirculation through upper and lower vents; (3) delayed conduction through interior wall including effect of absorbed solar radiation; (4) conduction through interior glazing.

The calculation of beam solar radiation which passes through the sunspace into adjacent rooms is done by using the DOE-2 shadow routines to project the hourly-varying solar image of each exterior window onto the different interior surfaces in the sunspace. In this calculation external obstructions (such as overhangs and neighboring buildings) and internal obstructions are accounted for. Diffuse solar radiation incident on the interior surfaces of the sunspace is also calculated. This radiation, which is assumed to be uniformly distributed throughout the sunspace, originates by transmission through the exterior windows or by internal reflection of beam radiation.

The user can specify any of several different forms of convection between the sunspace and adjacent rooms. These include fan-forced or buoyancy-driven air circulation between upper and lower vents, free convection through a doorway opening, and for residential applications, use of a sunspace to preheat outside ventilation air. For commercial applications, the program allows return air from adjacent spaces to be passed into the sunspace, where, after mixing with the sunspace air, it is either exhausted or transferred back to the central air handling system.

The program makes available several mechanisms for controlling heat gain and loss from the sunspace and heat exchange with adjacent rooms:

(1) Air flow between sunspace and adjacent rooms can be turned on and off via a time-clock schedule or by a controller which acts on the the sunspace-to-room temperature difference.

(2) A venting algorithm allows outside air to be circulated through the sunspace to prevent overheating. The venting can either be fan-forced or natural. If forced, a fixed air-change rate is used. If natural, the program calculates the air-change rate from windspeed and inside-outside temperature difference using user-specified correlation coefficients.

(3) For sun control, fixed obstructions such as fins and overhangs can be modelled, or movable shading devices can be deployed on exterior or interior glazing according to a time-of-year/time-of-day schedule. Alternatively, shades can be deployed whenever transmitted solar radiation exceeds a user-specified threshold value (which may vary seasonally or by time of day). More complex dynamic controls are possible with functional input (see previous section).

(4) Conductive heat loss can be reduced by using movable insulation on sunspace exterior or interior windows. The insulation can be deployed according to a schedule or deployed whenever the outside temperature falls below a threshold value.

In order to track the large temperature swings that can occur in sunspaces, the venting and interzone convection calculation is done in small timesteps, ranging from 0.6 to 3 minutes depending on the sunspace heat capacity and load in a given hour. The conventional one-hour timestep used in DOE-2 was found to give unstable or unphysical results for the convection calculations, particularly when the temperature difference between sunspace and adjacent room was large at the beginning of an hour.

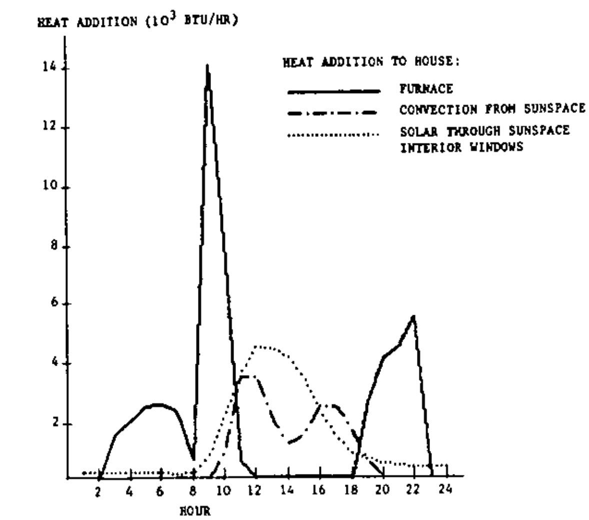
The interior walls in a sunspace are often fairly massive, so that the delay in the heat transfer by conduction through the wall can be important. The DOE-2.1B program was modified so that interior wall response factors are used in the SYSTEMS program to calculate delayed conduction taking into account time-varying space temperatures. This calculation also considers the interior sol-air effect, i.e., the effect of solar radiation absorbed on the sunspace side of an interior wall.

DOE-2 has a built-in routines which allow daylight illuminance levels in a sunspace to be calculated, although, due to the geometric complexity involved, the program cannot directly calculate illuminance levels in adjacent rooms due to daylight originating in the sunspace. However, functional input allows users to enter daylight factors, obtained from scale model measurements, which give room/sunspace illuminance ratios as a function of sky conditions. The program will then use these factors to determine hourly illuminance values and the corresponding reduction in electric lighting power for rooms with lighting control systems.

An important limitation of the sunspace model is that it does not account for air temperature stratification or in-room convective flows. We therefore caution against its use for multi-story atria unless the atrium air is well- mixed by the distribution system.

The French RAMSES group will validate the sunspace model against measurements of insolation, temperature, and inter-zone air flow for attached sunspaces on an apartment building near Paris. The program has been extensively tested in an energy analysis of the Pacific Museum of Flight in Seattle [1]. This building, which is currently under design, contains a 48000 sq ft exhibition gallery with a 100% glazed roof through which sunlight can pass to adjacent rooms.

Sample results of the sunspace simulation are shown in Fig. 2, which is based on SYSTEMS hourly reports from the residential sunspace example in the DOE-2.1C Sample Run Book [5].



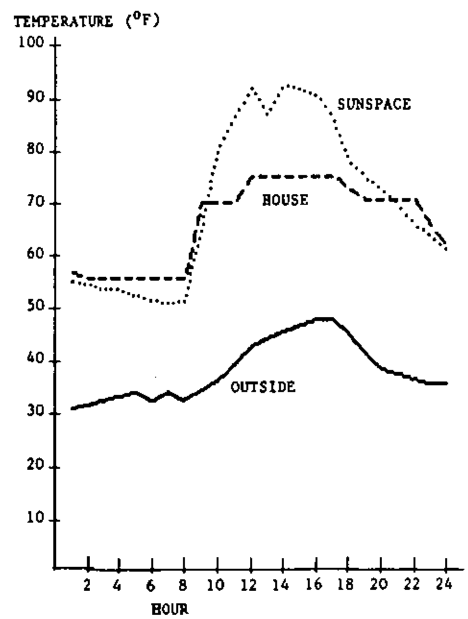


Figure 2. Sample hourly output for a clear January day in Chicago from a DOE-2.1C simulation of a house with a south-facing attached sunspace. The heating setpoint of the house is 70 °F during the day and 55 °F at night. Warm air from the sunspace is circulated to the house at 200 CFM whenever the sunspace-house temperature difference exceeds 4 °F, and the house temperature is below 74 °F. The sunspace is vented with outside air at 20 air changes per hour whenever the sunspace temperature exceeds 90 °F.

POWERED INDUCTION UNIT

The DOE-2 SYSTEMS program simulates a wide variety of secondary systems. Periodically new configurations appear on the market and are added to the DOE-2 simulation capabilities. The powered induction unit has gained popularity in recent years, as designers have attempted to mitigate some of the perceived deficiencies of VAV systems, and in addition to conserve energy by recovering excess heat from the building core. Consequently a model of the PIU system has been added to the DOE-2.1C SYSTEMS program.

The PIU system resembles the traditional variable air volume system with the addition, to the VAV terminal box, of a small fan or blower, which induces some amount of air from the plenum. The fan has two functions.

(1) Warm return air from the building core is sent into the plenum. The PIU fans draw this air from the plenum into the exterior zones requiring heat, thus conserving energy.

(2) The fan provides increased air movement when the VAV damper decreases primary air to the minimum, providing increased occupant comfort. In addition, primary air may be decreased below levels normally allowed in standard VAV systems, saving additional energy.

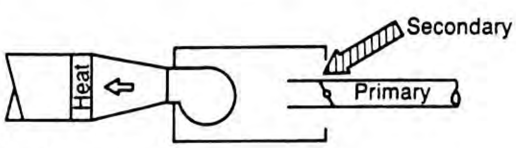


Figure 3.

Series PIU

DOE-2 models two types of PIU boxes - series and parallel. In the series unit (Fig. 3) the fan draws air from both the primary (central system) and the secondary (plenum) air streams. The proportion of primary to secondary air is controlled by the VAV damper. The amount of secondary plus primary air is constant, and the fan runs all the time (when the central fan is on) at constant speed. In the parallel unit (Fig. 4) the fan draws air from the secondary air stream only. In addition, the operation of the fan is intermittent -- a thermostat set point regulates turning the fan on and off. When cooling is required, the fan is generally off, and the unit operates as a normal VAV. When the primary air damper is at minimum, the fan is on, and the operation is constant volume ventilating/heating. Thus the parallel unit does not supply a fixed air quantity to the zone under all conditions.

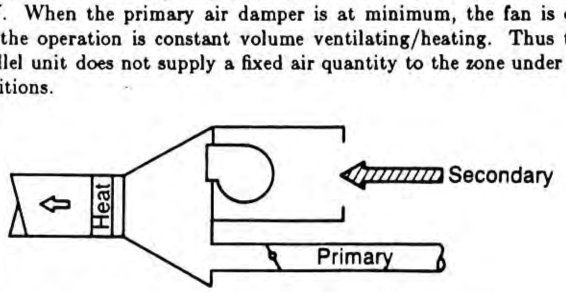


Figure 4.

Parallel PIU

The capabilities of the PIU model are fairly broad. As in all DOE-2 system descriptions, the user gives a list of zones belonging to the system. In addition, for the PIU, one zone (the "induced-air-zone", usually the core zone) is denoted as a supplier of secondary air to the rest of the zones in the system. For each zone, the user can select the terminal unit as being constant volume, VAV, series PIU, or parallel PIU. The system can be simulated with any combination of terminal boxes in each zone. Reheat coils are also optional in each zone. When PIU boxes are selected, the user generally sizes the blowers. For parallel terminal units, the user supplies a set point temperature for the blower (which can be scheduled, to allow for set-back). Normally the PIU fans operate in conjunction with the main system fan, but the user can decide to run the PIU fans independently at night and use the reheat coils to maintain a minimum temperature.

Two features of the PIU system have required changes in the DOE-2 simulation methods: the coupling of the "induced-air-zone" to the other zones, and the intermittent operation of the parallel PIU fan.

Normally DOE-2 models zones as being convectively independent, in order to avoid solving for all zone temperatures and extraction rates simultaneously. The exception has been plenums; air can be supplied from or returned through a plenum. In the case of a PIU system, however, air can be taken from one conditioned zone and supplied to other zones in the system. Since the zone temperatures and extraction rates are then strongly coupled, this situation would normally require iteration. DOE-2 avoids this by: (1) allowing only one zone to supply air; (2) simulating this zone first in each time step. Thus the temperature the air coming from the induced-air-zone is known, and the rest of the zones are simulated independently. For each zone the program must keep track of the amount of primary air and secondary air required by each zone, and the amount of secondary air available from the induced-air-zone. If more secondary air is required than is available from the induced-air-zone, the additional air is assumed to come from the actual zone being simulated.

For each zone DOE-2 obtains the zone temperature and extraction rate by simultaneously solving the temperature deviation equation (1),

$$\sum_{i=0}^{2} p_i \, (ER_{t-i\Delta} - \dot{Q}_{t-i\Delta}) = \sum_{i=0}^{3} g_i \, \Delta T_{t-i\Delta} \qquad (1)$$

and the thermostat equation (2).

$$ER_t = f(T_t) \qquad (2)$$

Here p and g are the coefficients of the room transfer functions, ER is the extraction rate, Q is the cooling load at constant temperature, ΔT is the difference between the constant temperature and the real room temperature, T_t. Normally $f(T_t)$ is assumed to be a piecewise linear function - the pieces being simply the heating action band, the dead band, and the cooling action band. The fan setpoint for parallel PIU's adds complexity by dividing either the cooling band or the dead band into two pieces (Fig. 5). If $f(T_t)$ were simply linear, ER_t and

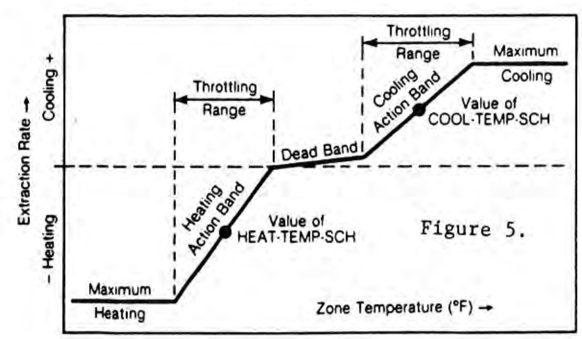


Figure 5.

T_t could be obtained trivially. Since $f(T_t)$ is piecewise linear, and since the limits of each band (the capacities) are temperature dependent, DOE-2 employs a bracketing technique. An extraction rate (at the top or bottom of one of the bands) is assumed, and the temperature response is calculated. Depending on the temperature response, the endpoints are adjusted, the correct band is chosen, and the actual extraction rate and zone temperature are solved for. For the parallel PIU, extraction rates are also assumed for the fan on and for the fan off. Depending on the temperature response, the fan is assigned to be on, off, or a combination of the two for the time step.

ADVANCED COGENERATION CAPABILITIES

Since its inception, the DOE-2 PLANT program has offered users the ability to study the performance of on-site electrical generation and co-generation as part of a building's central plant. Recent changes in Federal and State laws, however, have granted small power producers and cogenerators additional flexibility in the operation of dispersed electrical generation facilities (PURPA). These changes permit easier access to electric utility grids for both additional support on-site, as well as a regulated market for the sale of excess electrical energy. No longer must installations be sized to meet electrical peak demands, including the specification of redundant capacity, as in the days of the Total Energy Systems. Electricity may be both freely sold to and bought from the local grid.

The DOE-2.1C version of PLANT acknowledges these regulatory changes with an entirely reworked concept of electricity generators, and heat and electricity driven chillers, when operated in a cogeneration mode. When applied to reciprocating internal combustion engines (diesel and spark-ignition) and gas turbines, cogeneration refers to the sequential generation of electricity and the recovery of heat for use in satisfying building thermal energy demands (heating and cooling). The new features have been used to study the economies of cogeneration in commercial buildings [11].

The essence of the new concept is the explicit recognition of the energy value of both outputs of the cogeneration process -- heat and electricity. Previously, the user was only permitted to schedule operation based on the electrical output of the prime movers. The associated heat recovered could only be used to the extent that a demand existed for thermal energy; any excess would be wasted. Now that the availability of a utility grid has loosened the requirement that operation be based solely on the on-site electrical needs, operation can be scheduled to eliminate wasted recovered thermal energy.

The user can now specify a cogeneration mode of operation that is based on either electrical or thermal loads. These features have been integrated with the existing command structure of PLANT to allow operation to be scheduled in either mode, during the course of a given day or on the basis of the magnitude of the load to be served. Thus the user can specify a cogeneration system that will track the thermal loads of a building, where these loads may consist of heating or cooling loads (if a heat-driven, absorption chiller is available to accept recovered heat from the prime movers). In a thermal tracking mode with interconnection to a utility, both outputs of the prime movers will be fully used resulting in the most efficient use of energy from the electricity conversion process. At other times, when utility rates for purchase are high, operation can be scheduled for the full output of the generators in order to maximize sales of electricity to the utility. When utility rates for sale are high, operation can be scheduled to follow electrical loads, to minimize purchases from the utility. In addition, when both heat and electricity driven chillers are available to meet cooling loads, the output of the prime mover can be specified to balance loads between the two chillers. For example, increased electrical output will be accompanied by increased recovered thermal energy output. Both outputs can be used to satisfy cooling demands, and the loads on the two chillers will be allocated accordingly.

The implementation of these new cogeneration capabilities has required modifications to two aspects of the PLANT program: (a) equipment simulations; and (b) load allocations.

The equipment simulations for the prime movers (reciprocating internal combustion engine and gas turbine) and the chillers (absorption, centrifugal, reciprocating, and double-bundle) have been re-written. The inputs and outputs of each type of equipment is now described with simple quadratic transfer functions. The coefficients to these equations remain user-definable, as before. The use of quadratic equations permits rapid determination of input values from output values, and vice versa, via the solution of the quadratic equation. This feature forms the cornerstone of the logic underlying the revised load allocation algorithms. Basically, these types of equipment are now no more than black boxes described by a handful of quadratic transfer functions.

The load allocation routines required several levels of modification. First, the original specification of prime mover equipment capacities had to be modified to include a rating for thermal output, as well as electrical output. Next, the user-input load allocations instructions for prime mover operation had to be similarly modified to permit load allocation on the basis of either thermal or electric loads. The allocation logic remains the same, only the nature of the loads being allocated has changed.

The major modification to the load allocation routines took place in the default allocation algorithms, which are used in conjunction with or in place of user-specified operation. The design of the PLANT program allows for varying levels of user-specified and default-optimized operation. For example, prime mover operation may be user specified on a daily/hourly basis, while chiller operation will determined by the program. At an even higher level of generality, only a cogeneration mode (e.g., track thermal loads) need be specified by the user. The user specified inputs for operation provide boundaries or constraints for the program's internal logic, which attempts to optimize the equipment operation. Generally the quadratic transfer functions are used to find the optimal operating point for each piece of equipment, and no iterations needed. For one situation, however - the determination of prime mover output when both heat and electricity driven chillers are present and a cooling load exists - an iterative solution was necessary.

EXPANDED TREATMENT OF ENERGY COSTS

The DOE-2.1C version of DOE-2 features greatly expanded capabilities for the treatment of energy costs. In the past decade, novel approaches for the billing of energy (electricity, in particular) have gained wide acceptance. Time-of-day prices, demand ratchets, and lifeline blocks are all increasingly familiar features of current and prospective rate schedules for energy. For dispersed producers of electricity, Federal law permits several valuation options for the sale of electricity to the electric grid. All of these developments have been incorporated into DOE-2.1C.

The changes are symbolized by the movement of the energy cost commands from the PLANT program to the ECONOMICS program. In previous versions of DOE-2, energy costs were calculated by PLANT and life cycle costs by ECONOMICS. The energy cost calculations in PLANT could handle some simple block rate structures, but could not deal with time-of-day charges, demand ratchets, or complex block structures. In DOE-2.1C an hourly file containing plant loads, fuel demands, and generation is passed to ECONOMICS, allowing ECONOMICS to do all of the energy cost calculations, as well as the life cycle costs.

Rate structures of a number of utilities were analyzed and the structures were reduced to 4 basic components. These components correspond to 4 DOE-2 commands, each with its associated keywords and codewords. This discussion focuses on the concepts involved rather than the DOE-2 inputs.

The first component (corresponding to the ENERGY-COST command in DOE-2) identifies the type of energy being valued (fuel type and billing units) and describes the components of a monthly bill (minimum charges, fixed charges, and rate limitations). The first component also contains features for simplified rate structures (all consumption valued at one rate) and provides links to two of the other three components for more sophisticated rate structures.

The second component (SCHEDULE, DAY-CHARGE-SCHEDULE) describes how energy used in the hours of the monthly billing period is allocated into different billing categories. This component is used primarily to specify time-of-day rate structures (e.g., on-, off-, and shoulder-peak periods) and rates that vary by season (e.g., Winter versus Summer) This component is optional; the default is that all consumption during the month falls into the same billing category.

The third component (CHARGE-ASSIGNMENT) describes how the energy in a given billing category is actually billed. This component primarily describes block rate structures; many rate structures provide for successive tiers of use, each of which has a specific rate (dollars per unit of consumption) associated with it. Included in this category are lifeline rates, as well as declining block rates. In addition, some rate structures define the tier boundaries endogenously, based on demand (e.g. kWh/kW).

The fourth component (COST-PARAMETERS) is used only to describe features of electricity tariffs. These features include the specification of demand ratchets, and the PURPA mandated options for the sale of electricity to utilities. Demand ratchets are essentially hurdle levels of demand, which the demand charge can not fall below. They are typically based on a previously recorded level of demand, but can also appear as an average of previous high demands. In addition, the duration of demand ratchet can vary from 12 months down to 1 month, or can be restricted by season. The PURPA-mandated options permit sellers of electricity to choose one of two accounting conventions for the valuation of electricity sales. The first, called Net Sale, offsets purchases of electricity on-site and sells the excess at the utility's avoided cost rates. The second, called Simultaneous Buy/Sell values all electricity generation at avoided cost, but bills on-site consumption at standard utility rates.

CONCLUSIONS

We have described five new features of the DOE-2.1C building energy simulation program. These features significantly enhance the simulation capabilities of the program. In addition, they indicate some of the directions in which the next generation of building simulation programs will go:

- increased power and flexibility of input (functions);
- integration of LOADS and PLANT (sunspace);
- strong convective coupling between zones (sunspace and PIU);
- standardization of equipment component models, allowing a wide range of operating modes (cogeneration);
- the ability to optimize buildings and plant operation to deal with a wide range of economic circumstances (economics).

REFERENCES

1. Bazjanac, V., "The Energy Performance of the Pacific Museum of Flight", Berkeley, April 1985.
2. LBL-8689 Rev. 2 DOE-2 Users Guide
3. LBL-8706 Rev. 2 DOE-2 Reference Manual
4. LBL-8688 Rev. 4, DOE-2 BDL Summary
5. LBL-8679 Rev.2, DOE-2 Sample Run Book
6. LBL-11353, DOE-2 Engineers Manual
7. LBL-18046, The DOE-2 building Energy Analysis Program
8. LBL-12300, The Theoretical Basis of the DOE-2 Building Energy-Use Analysis Program
9. LBL-14026, Simulation of HVAC Equipment in the DOE-2 Program
10. LBL-8607 Rev. 4 Suppl., DOE-2 Supplement
11. Eto, J., Commercial Building Cogeneration Opportunities, Proceedings of the American Council for an Energy-Efficient Economy 1984 Summer Study on Energy Efficiency in Buildings, Santa Cruz, CA, August 1984.