CLIMA 2016 - proceedings of the 12th REHVA World Congress

Heiselberg, Per Kvols

Publication date:
2016

Document Version
Final published version

Link to publication from Aalborg University

Citation for published version (APA):
Heiselberg, P. K. (Ed.) (2016). CLIMA 2016 - proceedings of the 12th REHVA World Congress: volume 9. Aalborg: Aalborg University, Department of Civil Engineering.

On time-alignment of weather data in Building Performance Simulation

Livio Mazzarella[#1], Martina Pasini[#2]

[#]Energy Department, Politecnico di Milano
Via Lambruschini, 4, 20156 Milano, Italy

[1]livio.mazzarella@polimi.it

[2]martina.pasini@polimi.it

Abstract

While simulating complex systems, information exchange among components is one of the most important aspects. A specific kind of information is that related to weather data. The format of the climatic data diffusely used in Building Performance Simulation tools (BPSts) contains information about weather variables which are different from each other as far as concern their nature and timing. They have a statistical origin and, in the majority of the cases, are provided on an hourly basis. Given this inhomogeneity and hourly time base, care had been taken to manage their timing and different approaches are today's in use by BPSts. Furthermore, when the building involves complex components and control strategies, sub-hourly simulation are needed to understand the efficiency of the enquired system. This necessity has led to the implementation of even more different interpolation routines. The capability of these interpolation routines to represent weather conditions that change much more frequently than shown on an hourly basis is here investigated. Besides, BPSts are today used also at operational time, as predictive tools for control strategies and/or Fault Detection and Diagnosis. In this scenario, the statistical validity of climatic data is not anymore sufficient, while their variability profile, recorded with high frequency, and their correct interpretation/synchronization (integral values vs instantaneous values), might became relevant. In this article will be presented a review of the choices implemented by two well-known software, such as TRNSYS 17 and EnergyPlus 8.4.0, to handle weather data and further considerations will be made upon possibilities offered or denied by this choices when different components are involved in the simulation..

Keywords - building energy performance; weather data; simulated and real performance

1. Introduction

In developing OpenBPS -an object-oriented, open source building simulation code- two of the first architectural choices made, regarded input timing and information exchange synchronization between different objects involved in the simulation.

The common requirement, when dealing with information exchange, for both internal and external objects involved in the simulation, is the knowledge of the nature and exact timing of this information.

Following these considerations, a review of the algorithms implemented in two well known Building Performance Simulation tools (BPSts), such as TRNSYS and

EnergyPlus, has be performed in order to better understand the compliance of their algorithms with the need to:

- synchronize information exchange between objects involved in the simulation, one of which is the climatic data manager (i.e. the object responsible to expose weather data information to the other objects);
- perform simulation with sub-hourly time steps;
- perform "realistic" vs "statistic" simulation.

Different tools have made different choices regarding time-synchronization architectural choices, therefore an analysis of the internal-consistency of these choices is here presented.

1. Information exchange: timing and consistency

The first/most important synchronization in a numerical simulation regards time alignment in information exchange. Among the data to be exchanged, weather-related data are the most complex to synchronize due to the different nature of its quantities, some of which are intensive/scalar and other extensive/vectorial.

Since different kinds of uncertainties are already ingrained in numerical simulation, while defining the strategy to handle time-variant information inside the simulation the main pursued goal was to avoid assumptions not strictly needed.

As far as concern weather data, a review performed on both the weather data file formats manuals [1] and the Guide to Meteorological Instruments and Methods of Observation [2] had shown that in the majority of the cases weather data are recorded as *instantaneous* values associated to a *time stamp*.

These *instantaneous* values, or *observations*, are an average or smoothed value derived by a number of samples, i.e. of single spot or instantaneous readings gained by a measuring system, but they are not an average value on the recording time base.

However, weather data, which are relative to solar radiation, are given in all the most used weather data file formats as integrated over a time step, i.e. as the total amount of solar radiation received during the period ending at the time stamp associated with the datum and starting at the timestamp associated with the previous one.

Another aspect to take into consideration is the nature of the time stamp associated with each datum that, in all the recent versions of weather file formats, is local time instead of solar time. This choice is coherent with the fact that, in the majority of the cases, others time-dependent input-provided data, as the schedules defined for describing user habits (such as working hours, etc.), are based on local time.

Therefore we have *known* input data at specific local time stamps, which are relative to users-schedules and *instantaneous* weather climatic data, such as dry temperature, relative humidity, wind velocity, etc. On the other hand, we have integral information about solar radiation over specific time steps (i.e. irradiation). This last information cannot be simply translated into an *instantaneous* value (i.e. irradiance) at the midpoint of the corresponding time step, being the integral average value times the time step. The known values are then integral average values for both direct horizontal or direct normal irradiance and diffuse horizontal irradiance. The weather processor has instead to provide to each object the required total irradiance on certain specified oriented surfaces. Thus these values cannot be directly used, even though correctly aligned, because also sun position has to be correctly aligned to calculate the right amount of radiation for each oriented surface. That means that when a rule is set to assure time alignment of provided weather climatic data, a consistent rule should be set also for the time alignment of the sun position. A strictly coherent synchronization of these different kinds of information might not be so important when dealing with *statistically* derived data, but it might become relevant when studying realistic situations, such as during the empirical validation of BPSts against measured data (taken in local time) or at operational time, when using monitoring weather data to optimize the performances of the building system.

To avoid misunderstanding some definitions are now given regarding the time:

- **time stamp**: time which the weather data are provided to from the input (just as in the weather file);
- **global simulation time (gs-time)**: time which the air node energy balance equation object (EBEO) is synchronized with;
- **local simulation time (ls-time)**: time which a generic object is synchronized with;
- **time stamp time step (ts-time step)**: absolute value of the difference between two contiguous time stamps;
- **global time step (gs-time step)**: time step size of time partitioning for EBEO;
- **local time step (ls-time step)**: time step size of time partitioning for a generic object.

It is evident that the global simulation time is leading the simulation and then the other two have to be synchronized in a consistent way.

2. Objects information exchange requirements

Other aspects to consider, when defining the strategy for information synchronization during the simulation, are the time-specific requirements of involved objects. Different objects might need information to set up their Boundary Conditions (BCs) at different time than the actual global simulation time; for instance, weather data may be required for both the actual gs-time and a gs-time one step before or for times in between them. To explain the concept, we will focus on the most common objects in BPSts, walls, which implement numerical solutions of mathematical problems modelled with partial differential equation. In this case, all numerical schemes require *instantaneous* values of the independent variables and BCs, (temperatures or fluxes), for their *domain of resolution* (local domain) with a time resolution that depends on their own specific assumptions. Both finite volume and finite different numerical schemes (or conduction transfer functions), provide as results the *instantaneous* values of their dependent variables, temperatures and/or fluxes, at the current ls-time. To get this solution they need to know the *instantaneous* values of their initial state at the previous ls-time of calculation, relative to their local domain of resolution, and of their boundary conditions at the previous ls-time of calculation and/or at the current ls-time, depending on the chosen scheme. For example to solve the problem of conductive heat transfer in a wall, it is needed to know the fluxes entering its boundaries and the temperatures on their boundaries at precise ls-time. It is due of the simulation manager, which is working with the global simulation time, to synchronize the weather manager with it. Thus the weather manager, when inquired by an object, is providing to it weather data synchronized with the gs-time. After that, it is a responsibility of the object itself to manage this data and eventually to record past instantaneous values and to choose weighting strategies for its solution or for interpolation, when it ls-time step is less than the gs-time step. If we provide to the object an "*instantaneous*" value for one of its BCs that instead is the average value on the previous gs-time step, we will prevent the object to make its own consistent choices on how to manage its BCs. For example, let us assume an object implements a Crank-Nicholson numerical scheme for time discretization and that its ls-time step is the same as the gs-time step (i.e. 1 hour), therefore its result, at the current ls-time step, will be synchronized with the gs-time, e.g. 10:00 AM,. To calculate this result this scheme needs to know its BCs at the current ls-time (10:00 AM) and at previous ls-time (at 09:00 AM). If we assume that another object uses a full implicit scheme, it instead will use only BCs defined at the current ls-time (10:00 AM); and, if a third object exists which implements a full explicit scheme, it will use only BCs defined at the previous ls-time (09:00 AM). In any of these three cases the weather data manager has to provide data synchronized with the actual gs-time, i.e. 10:00 AM. Any object will choose what to use and eventually how to manage it. In the first case the object had to store the previous provided BCs, in the second not at all, in the third yes again. Furthermore, if the weather data manager at gs-time 10:00 AM provides instead values that are programmatically referred to 9:30 AM (being the interpolated value between 09:00 and 10:00 AM or the average value in such time interval) but labeled 10:00 AM, the CN scheme will use as its BCs "*adapted*" weather values at *8:30 AM* and at *9:30 AM*, instead of those at 09:00 AM and 10:00 AM as it should. The same happens for the other objects and the result is not only an "unrecognized" time shift of half an hour, but also the introduction of an error due to the fact that the numerical schemes do not require average values over the ls-time step, but instantaneous values just at ls-times.

3. Strategies implemented for information exchange

Following these considerations, the first decisions made regarding the time-synchronization of the information exchanged inside the tool under development, have been:

- to use local time as global simulation time and
- to choose as default type of information exchange, instantaneous values at each gs-time.

This choices will not prevent the exchange of other kind of information when available and required by some object (for example integral values of solar radiation are available, therefore the weather data manager should expose them too to whatever object might need them).

Concerning these aspects, TRNSYS 17 weather data manager provides instead averaged values on the simulation time step of those instantaneous variables recorded in the weather file. This choice has led to the need of making several assumptions, instead of using directly the information acquired by the weather data file. Furthermore, the use of linear interpolation has as a consequence the smoothing of local maxima and minima in the values assumed by the variable.

To make an example, we consider hourly-recorded weather data, marked with “o'clock” local time (i.e. at 01:00 AM, 02:00 AM and so on time stamp). In this case the *instantaneous* value of temperatures, wind direction, etc., are known for each recorded local time stamps. TRNSYS weather data manager will not provide directly this information if a simulation time step of 1 hour is used, but it will provide linearly interpolated values of those values at 00:30 AM, 01:30, etc., in the majority of the cases. There is a workaround to avoid this: to use a starting simulation time of half time step after the weather data first time stamp, such as 01:30 AM, and a simulation time step of one hour. To avoid this weather data manipulation the user can use more flexible components (Type), such as data readers, to import weather data information inside the simulation. However, if the user chose this way, the solar processor, will anyway give to the other component a sun position which correspond to the hour angle at the mid of the simulation time step.

EnergyPlus 8.4.0, instead, seems to calculate sun position at each global simulation time, and it seems to avoid the interpolation that we have described for TRNSYS when the gs-time is the same that the weather data recording time basis. However, in its documentation, EnergyPlus associates to the simulation time label of its data a time step interval and it declares its inputs and outputs as averaged value over the time steps. Therefore, its features about this aspect are not clear enough and a deeper analysis of its source code should be done to better evaluate its internal consistency. Furthermore, EnergyPlus does not allow choosing the exact time in a day for the beginning of the simulation; therefore some of the workaround mentioned for TRNSYS might not be applicable to it, if needed.

4. Implication of using hourly weather data with short time steps

Until know we have not yet considered the problem of having a global simulation time step which might be different from the weather data recording time basis, the time stamp time step (usually 1 hour). Even without considering this aspect, we have seen that the time alignment of the different variables contained in weather data file had raised some concerns and had needed interpolation. Another reason for weather data interpolation is expressed by the need of performing sub-hourly simulation, while hourly weather data are available. Numerous studies have shown that hourly simulations are not anymore sufficient, particularly when dealing with complex systems and control strategies.

Therefore, when questioning ourselves upon the algorithms to be implemented in the tool under development, two main enquiries arose:

- do the algorithms used for the calculation of sun position, have enough accuracy when the simulation time step is of the order of magnitude of minutes?
- which interpolation strategy is the most effective when dealing with data estimation between its recorded values?

In the following sub paragraph we will discuss the answers that we have found to these last two questions.

Sun Position Algorithm accuracy

First of all, we have done a review of the algorithms broadly diffused for the calculation of sun position [3]. Since we detected some ambiguity among different sources concerning the definition of the fractional year, we have recovered the original source for the definition of the most accurate equations for the calculation of sun declination and equation of time [4]. We have found a testimony [5] that reported one error in the original publication [4] in the first coefficient of the equation of time. We will report it here only for completeness:

$$E=0.0000075+0.001868\cos T-0.032077\sin T-0.014615\cos(2T)-0.040849\sin(2T) \quad (1)$$

In Spencer's equation of time we can find a fractional-year time "T" depending on the day-of-the-year number "d" which ranges from 0 on 1 January to 364 on 31 December, and which has the equation: $T= 2 \pi d/365$.

Since in these kind of simplified calculations, sun declination is kept constant over one day, and the author [4] suggested to use those equations in years in the middle of a 4-year leap cycle, we have wondered:

- what might be the rate of change of declination in one day and
- what would happen if we need to compute sun position when using monitored weather data near or for a leap year.

As far as concerns the first question, we have seen, by using the nautical almanac algorithms for the calculation of sun position, that the rate of change of its declination in one day is maximum 0,4°.

To answer the second question we have calculated the error of Spencer's simplified algorithm on the daily-average declination with respect to the more accurate algorithm of the nautical almanac. For a year before the leap one, this error is dependent from the longitude of the location and varies during the year, reaching maximum absolute values that are of the same order of the maximum daily declination variation, as can be seen in the following figure.

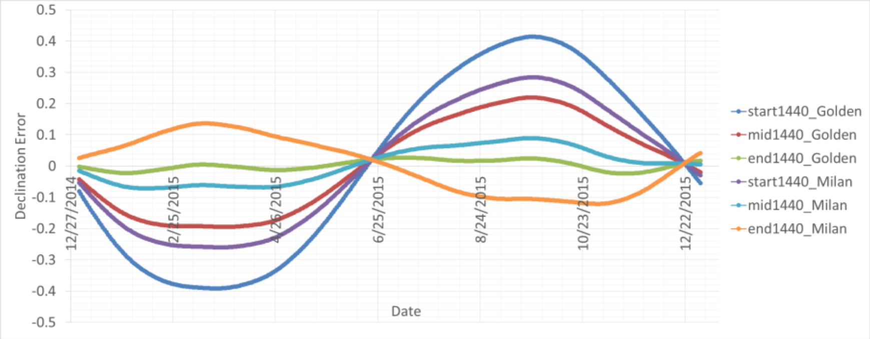

Fig. 1 Error on daily avarage values of sun declination

We have also observed that using the number of the day in the year in Spencer's equations, taking it at the beginning, mid or end of the day, might result in bigger or smaller error, depending on the longitude of the location, since they don't take into consideration universal time.

However, the accuracy reached with Spencer's simplified equations, is still acceptable when calculating the zenith of the sun even at very short time steps. We are confident that Spencer's simplified algorithm might be used with enough accuracy in a leap year (if needed to calculate sun position for monitored weather data), by changing the denominator in the fractional year definition.

Weather data interpolation algorithms

Certainly, when dealing with short time step simulations, the best solution is to work with weather data recorded with the same frequency.

However, we are still working with hourly weather data, even if Crawley, Hand and Lawrie had pointed out, already in 1999 [6], that this kind of data was no longer enough since interpolating between hourly observations does not accurately represent weather conditions that change much more frequently.

Given this *unresponsive* community behavior, we need to evaluate the possibilities that we have to overcome such lack of information.

While for the instantaneous variable, the only interpolation algorithm chosen coincides with linear interpolation, with solar radiation, different tools have chosen different strategies.

In particular TRNSYS and EnergyPlus have chosen different interpolation algorithms for solar radiation.

EnergyPlus has decided to assume the integral value of solar radiation (irradiation) associated with a time stamp in the weather data file as the average instantaneous value of solar radiation (irradiance) over the time stamp time step and to associate that value at the middle time between this time stamp and the previous one (backward middle time). After that the instantaneous value at each time stamp (irradiance) is calculated as linear interpolation between the average irradiance attributed to the backward middle time and that attributed to the forward middle time.

TRNSYS instead has chosen to interpolate the values obtained by the weather data file for horizontal solar radiation by using the curve for extraterrestrial radiation.

Even if this second strategy might be more physically-significant, it does not take into consideration those random phenomena that occur in the atmosphere, which are quite relevant when dealing with solar radiation. Furthermore, we have detected some strange behaviors of the solar radiation received on a surface when using this kind of interpolation. These strange behaviors are quite visible, especially for some façade orientation (west), as shown in the following figure.

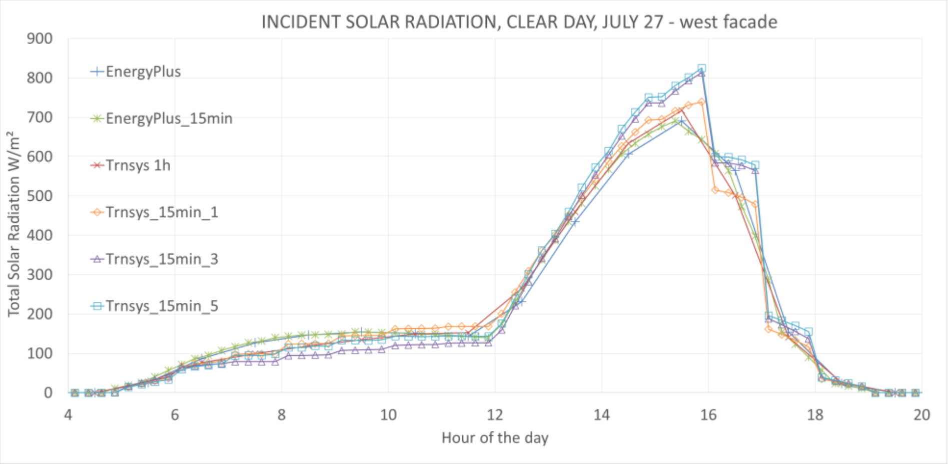

Fig. 2 Solar radiation interpolation on a time basis smaller than the time basis of recordings

We have tested three of the methods offered by TRNSYS to handle diffuse radiation, i.e. the:

- 1= Isotropic Sky Model
- 3 = Reindl Model
- 5 = Perez 1999 Model

The kind of interpolation implemented in TRNSYS is more heavy has far as concern computational effort and it seems to add little, if not less than the first one, to the estimate of the values assumed by the variable at in-between time stamps.

While trying to ensure the conservation of energy received on a horizontal surface, a valuable algorithm for the estimation of those unknown values might consider the statistical variability of that specific variable.

For example ground temperature has a very different time variability with respect to that of solar radiation.

Therefore, those variables that are associated with a higher “capacity” with respect to others should be interpolated with smooth filters. On the contrary, variables such as solar radiation or wind velocity might be better “mimicked” by applying a more realistic “pattern” to their interpolation. This statistical approach might have the goal to better evaluate the efficiency of a system and its control strategy under boundary condition fluctuations as much realistic as possible.

Statistics might be used to define a reduced number of patterns that might be applied to variables that show similar variability.

5. Conclusion

In developing OpenBPS -an object-oriented, open source building simulation code- weather data handling routines had to be implemented. To better understand the strength of the different possible implementation, a review of the better assessed and used data treatment algorithms has been performed.

This review has been focused on the routines that handle weather data, as implemented in two well-known software, such as TRNSYS 17 and EnergyPlus 8.4.0.

The aim of this review was to point out how weather data provided on a certain time basis are used in the simulation, when the simulation time step is smaller, equal or larger than the weather data recording time basis.

Even when the simulation time step is the same as the weather data recording time basis, we have seen that a clear architectural choice for the alignment of the different type of weather data is still missing. From the review, the more common strategy implemented seemed to be based on the exchange of averaged values of the variable over the time steps, even if their vales have to be assumed exactly as they are, such as with temperatures values. However, this strategy prevent the numerical scheme implemented in each simulation component to handle directly and correctly by itself the weighting process of the values assumed at different times by its boundary conditions.

A better architectural choice should preserve all the available information on the values assumed by each variable while other suppositions should be made only if strictly needed, as with solar radiation.

Furthermore, when the simulation time step is smaller than the weather data recording time basis, interpolation is needed. We have seen that both the simpler and the most complex version of the currently available interpolation routines might not be “significant enough” to test complex component and control strategies. An interpolation algorithm, which might be derived with a statistical analysis on weather data fluctuations, might be useful to overcome this lack of information. This algorithm will have the purpose of mimicking the variability of variables with similar capacity/pattern, to better evaluate the effectiveness of a system and of its control strategy when subjected to “realistic” boundary condition fluctuations.

References

[1] S. Wilcox and W. Marion. Users manual for TMY3 data sets. Golden, CO: National Renewable Energy Laboratory (2008)

[2] M. Jarraud, Guide to Meteorological Instruments and Methods of Observation (WMO-No. 8). World Meteorological Organisation: Geneva, Switzerland (2008).

[3] J.A. Duffie and W.A. Beckman. Solar engineering of thermal processes. Fouth Edition, John Wiley & Sons, Inc., New Jersey (2013)

[4] J. W. Spencer, Fourier Series Representation of the Position of the Sun. Search, 2 (5), 172 (1971).

[5] M. Oglesby personal correspindence, http://www.mail-archive.com/sundial@uni-koeln.de/msg01050.html, last visited the 30th of January 2016

[6] D. B. Crawley, J.W. Hand, and L.K. Lawrie. Improving the Weather Information Available to Simulation Programs, in N. Nakahara, H. Yoshida, M. Udagawa, and J. Hensen, editors, Proceedings of Building Simulation ’99, Volume II, Kyoto, Japan: IBPSA, pp. 529-536 (1999).