The Journal of Open Source Software

diyepw: A Python package for Do-It-Yourself EnergyPlus weather file generation

Amanda D. Smith[*1], **Benjamin Stürmer**[2], **Travis Thurber**[1], **and Chris R. Vernon**[1]

1 Pacific Northwest National Laboratory, Richland, WA, USA **2** Independent Researcher

DOI: 10.21105/joss.03313

Software

- Review
- Repository
- Archive

Editor: Tim Tröndle

Reviewers:

- @samuelduchesne
- @fneum

Submitted: 15 April 2021
Published: 29 August 2021

Summary

`diyepw` allows for quick and easy generation of a set of EnergyPlus weather (EPW) files for a given location over a given historical period. The user can obtain weather files using an open-source, automated workflow by simply specifying the location of interest using the World Meteorological Organization weather station ID number (*Integrated Surface Database Station History*, 2021), and specifying a year or set of years for which to generate EPW files. Building energy modelers can use these auto-generated weather files in building performance simulations to represent the actual observed weather conditions in the location(s) of interest, based on observed weather data obtained from the National Oceanic and Atmospheric Administration's Integrated Surface Database (*Integrated Surface Database (ISD)*, 2021; Smith et al., 2011). Because observed weather data are not available for every meteorological variable specified in the EPW format (*EnergyPlus Weather File (EPW) Data Dictionary*, 2015), `diyepw` starts with a widely-used set of typical meteorological year (TMY) EPW files (U.S. Department of Energy, n.d.-a), using them as the template to generate new EPW files by substituting in the observed values of selected meteorological variables that are known to affect building energy performance (see Using DIYEPW to generate AMY EPW files for details). Its output is a weather file or group of weather files that conform to the EPW format so they can be used with any building performance simulation software employing EnergyPlus (U.S. Department of Energy's (DOE) Building Technologies Office (BTO), 2020) as its simulation engine.

`diyepw` is available on Github (*Diyepw*, 2021), and it can be called directly as a package to incorporate EPW file generation into a custom script, and is designed to be customizable according to the modeler's needs. A step-by-step example tutorial is provided as a quick start option here: Tutorial.

Statement of need

Building energy modeling (BEM) practitioners and researchers have few options for obtaining EnergyPlus weather files that contain historical weather observations. Modelers often use EPW files that are based on typical meteorological year (TMY) data, which do not represent any given historical year and are usually only available for airport weather station locations. The Integrated Multisector Multiscale Modeling (IM3) project (*Integrated Multisector Multiscale Modeling*, 2021) needed a way to use observed weather data to drive simulations of model buildings using EnergyPlus for specific years in the past. Previous IM3 research (Burleyson et al., 2018) showed that for regional-scale BEM, where many buildings are aggregated, a model that is forced with weather files taken from stations throughout the region will have lower bias

[*] Corresponding author

in predicting the aggregate load than a model forced with only a few weather files that don't capture the heterogeneity in the region. Some commercial providers will offer weather files for given year(s) and location(s), but they may charge for each weather file and the source data and code used to process it will not be transparent to the user. Some modelers have created their own weather files by obtaining weather data and manipulating it to meet the EPW format, but it is a labor-intensive process and no open-source, automated software package existed to produce EPW files from publicly available weather observations until `diyepw`. This software will benefit the BEM community by allowing for easy use of reliable, quality-checked, publicly available weather data in their EnergyPlus simulations to represent actual historical years in specific location(s).

Relationship to other resources in this research area

`diyepw` was inspired by the Local Actual Meteorological Year File (LAF) application (Bianchi & Smith, 2019). LAF provided a free downloadable application that used web-based access to real meteorological data from the MesoWest database (University of Utah Department of Atmospheric Sciences, 2021) through an API (Synoptic Data PBC, n.d.) and processed it into an EPW weather file, allowing the user to combine results from multiple weather stations and from multiple years if desired. `diyepw` addresses some of its key limitations:

- LAF's workflow requires downloading and clicking and is not fully automated.
- LAF is no longer developed or maintained, and cannot be imported as a Python package.
- LAF relies on an API for downloading observed weather data that has limitations on the amount of data that can be downloaded without a paid account.
- LAF is not directly extensible to other sources of weather data, such as the NOAA ISD Lite format used here.

The EnergyPlus website lists additional resources for obtaining EPW files for building energy modeling (U.S. Department of Energy, n.d.-b). Few data providers can produce weather files for specific locations over a given historical period, and when they do provide such EPW files, the raw data representing the weather observations may not be available to the user. Thus the processing of that data to produce the EPWs is not fully transparent and reproducible. The user may be required to pay for these files and would not have the option to adjust the standards for data quality–for determining which values are acceptable for a given meteorological variable, or for limiting the amount of data that is interpolated or otherwise imputed by the software generating the EPW files.

Dependencies

`diyepw` relies on functionality from the following Python packages: `NumPy` (Harris et al., 2020), `pandas` (McKinney, 2010; The pandas development team, 2021), and `xarray` (Stephan Hoyer et al., 2020; S. Hoyer & Hamman, 2017).

Acknowledgements

This work was supported by the U.S. Department of Energy, Office of Science, as part of research in the MultiSector Dynamics, Earth and Environmental System Modeling Program. Pacific Northwest National Laboratory is a multi-program national laboratory operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830. A portion of the research was performed using PNNL Institutional Computing at Pacific Northwest National Laboratory.

References

Bianchi, C., & Smith, A. D. (2019). Localized actual meteorological year file creator (LAF): A tool for using locally observed weather data in building energy simulations. *SoftwareX*, *10*, 100299. https://doi.org/10.1016/j.softx.2019.100299

Burleyson, C. D., Voisin, N., Taylor, Z. T., Xie, Y., & Kraucunas, I. (2018). Simulated building energy demand biases resulting from the use of representative weather stations. *Appl. Energy*, *209*, 516–528. https://doi.org/10.1016/j.apenergy.2017.08.244

Diyepw. (2021). Pacific Northwest National Laboratory; Github. https://github.com/IMMM-SFA/diyepw

EnergyPlus weather file (EPW) data dictionary. (2015). {Big Ladder Software LLC}. https://bigladdersoftware.com/epx/docs/8-3/auxiliary-programs/energyplus-weather-file-epw-data-dictiona html

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

Hoyer, Stephan, Hamman, J., Roos, M., Cherian, D., Fitzgerald, C., keewis, Fujii, K., Maussion, F., crusaderky, Kleeman, A., Clark, S., Kluyver, T., Hauser, M., Munroe, J., Nicholas, T., Hatfield-Dodds, Z., Abernathey, R., MaximilianR, Wolfram, P. J., … Bovy, B. (2020). *Pydata/xarray: v0.16.2* (Version v0.16.2) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4299126

Integrated multisector multiscale modeling. (2021). https://im3.pnnl.gov/

Integrated surface database (ISD). (2021). National Oceanic; Atmospheric Administration. https://www.ncdc.noaa.gov/isd

Integrated surface database station history. (2021). https://www1.ncdc.noaa.gov/pub/data/noaa/isd-history.txt

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Smith, A., Lott, N., & Vose, R. (2011). The integrated surface database: Recent developments and partnerships. *Bull. Am. Meteorol. Soc.*, *92*(6), 704–708. https://doi.org/10.1175/2011BAMS3015.1

Synoptic Data PBC. (n.d.). *Mesonet API*. https://developers.synopticdata.com/mesonet/

The pandas development team. (2021). *Pandas-dev/pandas: Pandas 1.3.2* (Version v1.3.2) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.5203279

U.S. Department of Energy. (n.d.-a). *Weather data*. https://energyplus.net/weather

U.S. Department of Energy. (n.d.-b). *Weather data for simulation*. https://energyplus.net/weather/simulation. https://energyplus.net/weather/simulation

U.S. Department of Energy's (DOE) Building Technologies Office (BTO). (2020). *EnergyPlus*. https://energyplus.net/

University of Utah Department of Atmospheric Sciences. (2021). *MesoWest Data*. University of Utah. http://mesowest.utah.edu/