PyADM1: a Python implementation of Anaerobic Digestion Model No. 1

P. Sadrimajd^a, P. Mannion^b, E. Howley^b, P. N. L. Lens^a

^a*Microbiology Department, School of Natural Sciences, Ryan Institute, National
University of Ireland Galway, University Road, Galway, Ireland*

^b*School of Computer Science, National University of Ireland Galway, University Road,
Galway, Ireland*

Abstract

Anaerobic Digestion (AD) is a waste treatment technology widely used for wastewater and solid waste treatment, with the advantage of being a source of renewable energy in the form of biogas. Anaerobic digestion model number 1 (ADM1) is the most common mathematical model available for AD modelling. Commercial software implementations of ADM1 are available but have limited flexibility and availability due to the closed sources and licensing fees. Python is the fastest growing programming language and is open source freely available. Python implementation of ADM1 makes this AD model available to the mass user base of the Python ecosystem and its libraries. The open easy to use implementation in PyADM1 makes it more accessible and provides possibilities for flexible direct use of the model linked to other software, e.g. machine learning libraries or Linux operating system on embedded hardware.

Keywords: Anaerobic digestion, Biogas, ADM1, Python

Email addresses: p.sadrimajd1@nuigalway.ie (P. Sadrimajd),
patrick.mannion@nuigalway.ie (P. Mannion), ehowley@nuigalway.ie (E. Howley),
piet.lens@nuigalway.ie (P. N. L. Lens)

23 Required Metadata

24 Current code version

Nr.	Code metadata description	Please fill in this column
C1	Current code version	0.3
C2	Permanent link to code/repository used for this code version	https://github.com/CaptainFerMag/PyADM1
C3	Code Ocean compute capsule	https://codeocean.com/capsule/0554841/tree
C4	Legal Code License	MIT
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Compilation not required; Windows, Mac OSX, Linux, Python, SciPy, NumPy
C7	Compilation requirements, operating environments & dependencies	Copy, Scipy, Numpy, Python 3.8
C8	If available Link to developer documentation/manual	https://github.com/CaptainFerMag/PyADM1
C9	Support email for questions	p.sadrimajd1[at]nuigalway.ie

Table 1: Code metadata

25 1. Motivation and significance

26 Large amounts of organic waste (e.g. food waste, municipal wastewater,
 27 industrial wastewater, and agricultural waste) are being disposed every
 28 year. Compared to traditional waste treatment (i.e. landfill and composting)
 29 Anaerobic Digestion (AD) not only reduces emissions to the environment, but
 30 also produces renewable energy in the form of biogas [1]. There are numerous
 31 biogas plants around the world and with the growing concerns about climate
 32 change there will be more biogas plants in the future. According to the In-
 33 ternational Energy Agency (IEA), in 2019 there were 10000 biogas plants in
 34 Germany, 1000 in UK, and 400 in Switzerland [2].

35 Anaerobic Digestion Model number 1 (ADM1) from the International
 36 Water Association (IWA) [3] (the most commonly used model for AD) de-
 37 scribes AD as a five steps process (disintegration, hydrolysis, acidogenesis
 38 or fermentation, acetogenesis, and methanogenesis). ADM1 is a complex
 39 mathematical model that includes 35 state variables and their respective dif-
 40 ferential or algebraic equations describing the interrelated change between
 41 states [4]. This makes the use of computers and numerical solvers necessary
 42 for ADM1 application.

ADM1 has been implemented in different commercial softwares (e.g. GPS-X, SIMBA, WEST, MATLAB/SIMULINK, and AQUASIM) and programming languages (i.e. JAVA, FORTRAN, and C) [5, 6, 7, 8]. However, commercial packages involve license costs and are not flexible (configurable) enough for automation of modeling workflows and applications, and for model customization and updates. Benchmark Simulation Model number 2 (BSM2) [6] developed in the MATLAB/SIMULINK environment is one of the most widely used open source ADM1 implementations. BSM2 is available in the MATLAB/SIMULINK environment where the model code is available in C programming language. BSM2 was developed in older versions of MATLAB and requires modifications to run properly in MATLAB versions after 2017. In order to use BSM2 a purchase of MATLAB license is required and there is a learning curve involved in using MATLAB/SIMULINK environment.

Python being a freely available open source programming language, is the fastest growing programming language and the first programming language that many young scientists learn [9]. Also, many modern scientific programming and data analysis libraries (e.g. SciPy [10]) are developed in Python. Implementing ADM1 in Python makes it more accessible and easier to use for a wider community of researchers and engineers, enabling them to expand ADM1 applications and anaerobic digestion research.

2. Software description

PyADM1 is a Python implementation of ADM1 based on BSM2 that can be used with basic Python programming knowledge in any platform running Python. PyADM1 uses `scipy.integrate.solve_ivp` to solve systems of differential equations and `numpy` for math operations.

2.1. Software Architecture

The general Ordinary Differential Equation (ODE) form of ADM1 is:

$$\frac{dS_i}{dt} = \frac{Q_{ad}}{V_{liq}}(S_{i,in} - S_i) + \sum_{j=1}^{19} \nu_{i,j} \rho_j + transportterms; where i = 1...12 \quad (1)$$

$$\frac{dX_i}{dt} = \frac{Q_{ad}}{V_{liq}}(X_{i,in} - X_i) + \sum_{j=1}^{19} \nu_{i,j} \rho_j + transportterms; where i = 13...24 \quad (2)$$

where Q_{ad} is influent flow rate, V_{liq} is the volume of the liquid fraction of reactor, S_i and X_i are solubles matter and particular matter states for variable i , respectively. The terms ν (Biochemical rate coefficients) and ρ (kinetic

equations for process j) are from the Peterson Matrix format of ADM1 [3]. Equations 25 and 26 are related to cations and anions, respectively. Equations 27-32 are for ion states and differ between the ODE and Differential Algebraic Equations (DAE) form of ADM1. Equations 33-35 are related to the gas phase.

The software architecture is presented in Figure 1. Initial states (S_i and X_i), influent states ($S_{i,in}$ and $X_{i,in}$), simulation time (**days**), number of time steps (**timeSteps**), and model parameters are defined as global variables which are modifiable by the user, depending on the simulation scenario or experimental data [3, 11, 12, 13] for modeling purposes. After integration and solving the system of model equations over the given time period, PyADM1 delivers the results of the simulation as system state outputs (S_i). The current version of PyADM1 is capable of steady-state simulations with DAE and ODE forms of ADM1. Dynamic simulation capabilities will be added to the future versions of PyADM1. A user can switch between ODE/DAE modes by setting the **DAE_switch** variable to 0 for ODE and 1 for DAE.

2.2. Software Functionalities

There are three main functions. Function **ADM1_ODE(t, state_zero)** receives simulation time t and initial states $state_zero$ as input and calculates the derivatives of states as an ODE. Function **simulate(t_step, solvmethod)** uses **scipy.integrate.solve_ivp()** to solve the initial value problem (ivp) integration at each time step and integrate the ODE equations. In other words, it receives the initial states at the beginning of each time step and solves the differential equations to deliver the state values at the end of the time step. Function **DAESolve()** calculates the derivatives for S_{H^+} , S_{H_2} , S_{va^-} , S_{bu^-} , S_{pro^-} , S_{ac^-} , $S_{HCO_3^-}$, and $S_{NH_3^-}$.

Variable **DAE_switch** is to choose between complete ODE form (**DAE_switch=0**) and DAE form (**DAE_switch=1**) of ADM1 for the simulation. If **DAE_switch=0** then **ADM1_ODE()** includes ion state gradients in the ODE system and **simulate()** integration is over the whole simulations time and the final states in the end of the simulation time are reported. Hence, if **DAE_switch=1** then a loop iterates over time steps and at each time step **ADM1_ODE()** excludes ion states from the ODE and **simulate()** integration is only over the current time step. Further, the simulated states in the current time step are delivered to **DAESolve()** to calculate the rest of the states at the end of the time step. Then the iteration over the current state is over and the loop moves to the next time step until the end of the simulation time.

Figure 1: PyADM1 code architecture and flow of calculations

110 3. Illustrative Examples

111 In order to validate the PyADM1 code functionality, a steady-state sim-
112 ulation was executed using the exact same input states, initial states, and
113 parameters as the BSM2 (default values for state and parameter variables in
114 the PyADM1 code). Simulation time was 200 days divided into 15 minutes
115 intervals (i.e. time steps) and substrate characterisation (defines influent
116 states) was based on wastewater treatment sludge.

117 PyADM1 DAE steady-state simulation inputs, results, and comparative
118 errors are presented in Tables 2, 3 and 4. Decimal places reported here
119 are based on the BSM2 report for the purpose of demonstrating numerical
120 calculations accuracy and the significance should be based on the unit of
121 measurement for each state (commented in front of each state in the source
122 code). Achieved absolute error levels for all simulated states were smaller
123 than 10^{-5} (i.e. it has no physical significance in relation to the units of
124 measurement for process states) and according to the BSM2 report [6] this
125 error level validates the PyADM1 model implementation.

state	Influent	Initial	BSM2	PyADM1	Error
S_{su}	0.01	0.012	0.01195482971696	0.01195482971230	5E-12
S_{aa}	0.001	0.0053	0.00531474017163	0.00531474016819	3E-12
S_{fa}	0.001	0.099	0.09862140093080	0.09862140187903	-9E-10
S_{va}	0.001	0.012	0.01162500646386	0.01162500650562	-4E-11
S_{bu}	0.001	0.013	0.01325072966627	0.01325072971277	-5E-11
S_{pro}	0.001	0.016	0.01578366628454	0.01578366597247	3E-10
S_{ac}	0.001	0.2	0.19762971693755	0.19763167669097	-2E-06
S_{H_2}	10E-8	2.30E-7	0.00000023594510	0.00000023594506	4E-14
S_{CH_4}	10E-5	0.055	0.05508877644596	0.05508904965331	-3E-07
S_{IC}	0.04	0.15	0.15267787062633	0.15267747569836	4E-07
S_{IN}	0.01	0.13	0.13022981580368	0.13022980517356	1E-08
S_I	0.02	0.33	0.32869766372153	0.32869772786660	-6E-08

Table 2: Steady state simulation results comparison between BSM2 DAE2 Matlab implementation and PyADM1 DAE (soluble matter)

state	Influent	Initial	BSM2	PyADM1	Error
X_{xc}	2	0.31	0.30869766372153	0.30869765844290	5E-09
X_{ch}	5	0.028	0.02794724043504	0.02794724038020	5E-11
X_{pr}	20	0.1	0.10257410610668	0.10257410605184	5E-11
X_{li}	5	0.029	0.02948304970729	0.02948304962503	8E-11
X_{su}	0.0	0.42	0.42016598245457	0.42016598174665	7E-10
X_{aa}	0.01	1.18	1.17917179892370	1.17917179900952	-9E-11
X_{fa}	0.01	0.24	0.24303534471944	0.24303534195974	3E-09
X_{C4}	0.01	0.43	0.43192110563598	0.43192110381834	2E-09
X_{pro}	0.01	0.14	0.13730590893395	0.13730591105219	-2E-09
X_{ac}	0.01	0.76	0.76056265831322	0.76056260560337	5E-08
X_{H_2}	0.01	0.32	0.31702295336127	0.31702295557648	-2E-09
X_I	25.0	25.6	25.61739532744300	25.61739454773450	8E-07

Table 3: steady state simulation results comparison between BSM2 DAE2 Matlab implementation and PyADM1 DAE (particulate matter)

state	Influent	Initial	BSM2	PyADM1	Error
S_{cation}	0.04	0.040	0.04	0.04	0E+00
S_{anion}	0.02	0.020	0.02	0.02	0E+00
S_{H^+}	1.0	0.00000003423	0.00000003423440	0.00000003423351	9E-13
S_{va^-}	0.0	0.011	0.01159624707260	0.01159624782654	-8E-10
S_{bu^-}	0.0	0.013	0.01322082624850	0.01322082703576	-8E-10
S_{pro^-}	0.0	0.016	0.01574278319160	0.01574278389282	-7E-10
S_{ac^-}	0.0	0.2	0.19724115543650	0.19724312096623	-2E-06
$S_{HCO_3^-}$	0.0	0.14	0.14277747939210	0.14277733996334	1E-07
S_{CO_2}	0.0	0.0	0.00990039123430	0.00990013573502	3E-07
S_{NH_3}	0.0	0.0041	0.00409092845840	0.00409102651559	-1E-07
$S_{NH_4^+}$	0.0	0.0	0.12613888734520	0.12613877865798	1E-07
S_{gas,H_2}	0.0	1.02E-5	0.00001024103560	0.00001024110692	-7E-11
S_{gas,CH_4}	0.0	1.63	1.62560720998142	1.62562665039951	-2E-05
S_{gas,CO_2}	0.0	0.014	0.01415053467840	0.01415239461914	-2E-06

Table 4: steady state simulation results comparison between BSM2 DAE2 Matlab implementation and PyADM1 DAE (ion and gas)

4. Impact

There are about 30000 peer-reviewed entries related to "Anaerobic Digestion" search term available in the Scopus database. Establishing the correctness of what is published (i.e. benchmarking approach) a Python

implementation of ADM1 will make it available for many more researchers and opens new possibilities for extension and configuration of ADM1. It will allow researchers to reproduce published work and compare the results of the experiments in different scenarios and for different digester regimes, e.g. batch or semi-batch, continuous, up flow, and circulation.

In contrast to commercial packages, the open source code of PyADM1 facilitates model modifications by the user in a modern programming language. More than 6000 search results show up on Google Scholar for the query "ADM1 modified anaerobic digestion" results. Based on the niche application, model modifications can add equations to simulate additional states (i.e. nutrients, nitrogen, phosphorus, sulphur, and metals), introduce the effect of temperature on the model parameters, or different microbial groups (as new types of particulate matter). Also, It is possible to extend PyADM1 linked to other models (similar to BSM2 approach) for plant wide simulation and development of digital twins.

Moreover, first principle mathematical models such as ADM1 are used in hybrid modeling for Industry 4.0 and Model Predictive Control (MPC) [14] as well as design, test, and simulation of Internet of Things (IoT) devices. Using Python in PyADM1 makes it easier to use in connection to common data science, machine learning, and web development libraries.

PyADM1 can serve as an educational tool [15, 16] to better demonstrate the anaerobic digestion process and different digester operational scenarios. Also, simulation of different scenarios is a low risk and low cost method to train biogas plant operators or evaluate process design by consultants.

Modeling and simulation of anaerobic digestion using PyADM1 can facilitate optimal design of experiments [17] for research about substrate utilisation and operational parameters. Plant designers and consultants use ADM1 to estimate plant performance [5], design parameters, and techno-economic studies. Models of wastewater treatment processes are important for process optimization and design [18], simulation of operational scenarios [15], and controller design [19].

5. Conclusions

PyADM1 was validated with the benchmark model BSM2 as a reliable Python implementation of ADM1 model for modeling and simulation of the anaerobic digestion process. The current version is capable of simulating steady-state operation of biogas reactors using ADM1 ODE/DAE forms. PyADM1 provides free transparent access to anaerobic digestion simulation for a broad Python user base as a low cost, reliable, and flexible solution for many applications, e.g. test and design of process control systems, biogas

169 plant commissioning, design of experiments, operator training, and educa-
170 tion.

171 Additional features such as dynamic simulation, data file read/write, ap-
172 plication programming interface (API), and online graphical user interface
173 will be added in the future versions. Linked to Python implementations
174 of Activated Sludge Models (ASM) [20], PyADM1 can be a basis for open
175 source wastewater treatment modeling and simulation software.

176 6. Conflict of Interest

177 We confirm that there are no known conflicts of interest associated with
178 this publication and there has been no significant financial support for this
179 work that could have influenced its outcome.

180 Acknowledgements

181 This work was funded by Science Foundation of Ireland (SFI) Research
182 Professorship Programme on Innovative Energy Technologies for Bioenergy,
183 Biofuels and a Sustainable Irish Bioeconomy (*IETSBIO*³, award 15/RP/2763).

184 References

- 185 [1] S. D. Hafner, K. Koch, H. Carrere, S. Astals, S. Weinrich, C. Ren-
186 nuit, Software for biogas research: Tools for measurement and
187 prediction of methane production, *SoftwareX* 7 (2018) 205–210.
188 doi:10.1016/j.softx.2018.06.005.
189 URL <https://doi.org/10.1016/j.softx.2018.06.005>
- 190 [2] M. Gustafsson, J. Ammenberg, J. D. Murphy, IEA bioenergy task 37–
191 country reports summaries 2019 (2020).
- 192 [3] D. J. Batstone, J. Keller, I. Angelidaki, S. V. Kalyuzhnyi, S. G.
193 Pavlostathis, A. Rozzi, W. T. M. Sanders, H. Siegrist, V. A. Vavilin,
194 The IWA anaerobic digestion model no 1 (ADM1), *Water Science and*
195 *technology* 45 (10) (2002) 65–73. doi:10.2166/wst.2002.0292.
196 URL <https://doi.org/10.2166/wst.2002.0292>
- 197 [4] J. Jimenez, E. Latrille, J. Harmand, A. Robles, J. Ferrer, D. Gaida,
198 C. Wolf, F. Mairet, O. Bernard, V. Alcaraz-Gonzalez, H. Mendez-
199 Acosta, , D. Zitomer, D. Totzke, H. Spanjers, F. Jacobi, A. Guwy,
200 R. Dinsdale, G. Premier, S. Mazhegrane, G. Ruiz-Filippi, A. Seco,

- 201 T. Ribeiro, A. Pauss, J.-P. Steyer, Instrumentation and control of anaerobic digestion processes: a review and some research challenges, *Reviews in Environmental Science and Bio/Technology* 14 (4) (2015) 615–648. doi:10.1007/s11157-015-9382-6.
202
203
204
205 URL <https://doi.org/10.1007/s11157-015-9382-6>
- 206 [5] D. J. Batstone, J. Keller, Industrial applications of the iwa anaerobic digestion model no. 1 (adm1), *Water Science and Technology* 47 (12) (2003) 199–206. doi:10.2166/wst.2003.0647.
207
208
209 URL <https://doi.org/10.2166/wst.2003.0647>
- 210 [6] C. Rosén, U. Jeppsson, Aspects on adm1 implementation within the bsm2 framework, Department of Industrial Electrical Engineering and Automation, Lund University, Lund, Sweden (2006) 1–35.
211
212
- 213 [7] L. Pettigrew, A. Gutbrod, H. Domes, F. Groß, J. M. Méndez-Contreras, A. Delgado, Modified adm1 for high-rate anaerobic co-digestion of thermally pre-treated brewery surplus yeast wastewater, *Water Science and Technology* 76 (3) (2017) 542–554. doi:10.2166/wst.2017.227.
214
215
216
217 URL <https://doi.org/10.2166/wst.2017.227>
- 218 [8] J. Kauder, N. Boes, C. Pasel, J.-D. Herbell, Combining models adm1 and asm2d in a sequencing batch reactor simulation, *Chemical Engineering & Technology: Industrial Chemistry-Plant Equipment-Process Engineering-Biotechnology* 30 (8) (2007) 1100–1112. doi:10.1002/ceat.200700045.
219
220
221
222
223 URL <https://doi.org/10.1002/ceat.200700045>
- 224 [9] K. Srinath, Python—the fastest growing programming language, *International Research Journal of Engineering and Technology (IRJET)* 4 (12) (2017) 354–357.
225
226
- 227 [10] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., Scipy 1.0: fundamental algorithms for scientific computing in python, *Nature methods* 17 (3) (2020) 261–272. doi:10.1038/s41592-019-0686-2.
228
229
230
231
232 URL <https://doi.org/10.1038/s41592-019-0686-2>
- 233 [11] V. Razaviarani, I. D. Buchanan, Calibration of the anaerobic digestion model no. 1 (adm1) for steady-state anaerobic co-digestion of municipal wastewater sludge with restaurant grease trap waste, *Chemical Engineering Journal* 266 (2015) 91–99. doi:10.1016/j.cej.2014.12.080.
234
235
236
237 URL <https://doi.org/10.1016/j.cej.2014.12.080>

- 238 [12] D. Poggio, M. Walker, W. Nimmo, L. Ma, M. Pourkashanian, Mod-
239 elling the anaerobic digestion of solid organic waste—substrate charac-
240 terisation method for adm1 using a combined biochemical and kinetic
241 parameter estimation approach, *Waste management* 53 (2016) 40–54.
242 doi:10.1016/j.wasman.2016.04.024.
243 URL <https://doi.org/10.1016/j.wasman.2016.04.024>
- 244 [13] U. Zaher, P. Grau, L. Benedetti, E. Ayesa, P. A. Vanrol-
245 leghem, Transformers for interfacing anaerobic digestion models to
246 pre-and post-treatment processes in a plant-wide modelling con-
247 text, *Environmental Modelling & Software* 22 (1) (2007) 40–58.
248 doi:10.1016/j.envsoft.2005.11.002.
249 URL <https://doi.org/10.1016/j.envsoft.2005.11.002>
- 250 [14] S. Attar, F. A. Haugen, Simulation of a model-based predictive control
251 system to optimize the methane production of a biogas reactor, in: 2017
252 11th Asian Control Conference (ASCC), IEEE, 2017, pp. 1590–1595.
253 doi:10.1109/ASCC.2017.8287411.
254 URL <https://doi.org/10.1109/ASCC.2017.8287411>
- 255 [15] D. J. Batstone, J. Keller, J. Steyer, A review of adm1 extensions, appli-
256 cations, and analysis: 2002–2005, *Water science and technology* 54 (4)
257 (2006) 1–10. doi:10.2166/wst.2006.520.
258 URL <https://doi.org/10.2166/wst.2006.520>
- 259 [16] S. Caño de las Heras, B. Kensington-Miller, B. Young, V. Gonzalez,
260 U. Kruhne, S. S. Mansouri, S. Baroutian, Benefits and challenges of a
261 virtual laboratory in chemical and biochemical engineering: Students’
262 experiences in fermentation, *Journal of Chemical Education* (2021).
263 doi:10.1021/acs.jchemed.0c01227.
264 URL <https://doi.org/10.1021/acs.jchemed.0c01227>
- 265 [17] B. Peiris, P. Rathnasiri, J. Johansen, A. Kuhn, R. Bakke, Adm1 sim-
266 ulations of hydrogen production, *Water Science and Technology* 53 (8)
267 (2006) 129–137. doi:10.2166/wst.2006.243.
268 URL <https://doi.org/10.2166/wst.2006.243>
- 269 [18] D. Dochain, P. A. Vanrolleghem, Dynamical modelling & estimation in
270 wastewater treatment processes, IWA publishing, 2001.
- 271 [19] D. Vrecko, K. Gernaey, C. Rosen, U. Jeppsson, Benchmark simula-
272 tion model no 2 in matlab-simulink: towards plant-wide wwtp control
273 strategy evaluation, *Water Science and Technology* 54 (8) (2006) 65–72.

- 274 doi:10.2166/wst.2006.773.
 275 URL <https://doi.org/10.2166/wst.2006.773>
- 276 [20] M. Henze, W. Gujer, T. Mino, M. C. van Loosdrecht, Activated sludge
 277 models ASM1, ASM2, ASM2d and ASM3, IWA publishing, 2000.

278 **Current executable software version**

Nr.	Software metadata description	
S1	Current software version	v0.3
S2	Permanent link	<i>github.com/CaptainFerMag/PyADM1</i>
S3	Legal Software License	MIT
S4	Computing platforms	Python on any OS/Cloud
S5	Installation requirements & dependencies	SciPy, Numpy, Copy, Python 3.8
S7	Support email for questions	p.sadrimajd1[at]nuigalway.ie

Table 5: Software metadata