







ARTICLE OPEN ACCESS

Open-Source Anaerobic Digestion Modeling Platform, Anaerobic Digestion Model No. 1 Fast (ADM1F)

Kuang Zhu¹  | Wenjuan Zhang² | Elchin Jafarov³  | Satish Karra³  | Kurt Solander³ | Meltem Urgun Demirtas⁴  | Lutgarde Raskin¹  | Steven Skerlos^{1,5} 

¹Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, Michigan, USA | ²Department of Mathematical and Statistical Sciences, University of Colorado Denver, Denver, Colorado, USA | ³Computational Earth Science Group, Los Alamos National Laboratory, New Mexico, USA | ⁴Applied Materials Division, Argonne National Laboratory, Lemont, Illinois, USA | ⁵Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA

Correspondence: Lutgarde Raskin (raskin@umich.edu) | Steven Skerlos (Skerlos@umich.edu)

Received: 18 July 2024 | **Revised:** 16 November 2024 | **Accepted:** 26 November 2024

Funding: This study was supported by US Department of Energy Office of Energy Efficiency and Renewable Energy Bioenergy Technologies Office (BETO) under contract numbers (DE-AC02-06CH11357) with Argonne National Laboratory and (DE-EE0009284) with the University of Michigan.

Keywords: anaerobic co-digestion | anaerobic digestion model | digester stability | numerical methods

ABSTRACT

An open-source modeling platform, called Anaerobic Digestion Model No. 1 Fast (ADM1F), is introduced to achieve fast and numerically stable simulations of anaerobic digestion processes. ADM1F is compatible with an iPython interface to facilitate model configuration, simulation, data analysis, and visualization. Faster simulations and more stable results are accomplished by implementing an advanced open-source library of numerical methods called Portable Extensive Toolkit for Scientific Computation (PETSc) to solve the ADM1 system of equations. Leveraging PETSc, ADM1F can consistently complete a steady-state simulation under 0.2 s, over 99% faster than a benchmark ADM1 model implemented with MATLAB while achieving agreement of model outputs within 1% of those obtained with the benchmark model. For dynamic simulations, however, ADM1F has a computational speed advantage only when the influent characteristics update more frequently than every 4 h. The ability of ADM1F to be useful as a tool to study anaerobic digestion systems is demonstrated through two example implementations of ADM1F: (1) a two-phase co-digestion scenario evaluating the impact of the organic loading rate and the substrate composition on reactor performance and stability, and (2) a conventional digester scenario assessing the effectiveness of recovery strategies after disruptions that led to instability. These examples demonstrate how the high simulation speed and the convenience of the iPython interface allow ADM1F to complete complex analyses within minutes, much faster than computational strategies currently reported in the literature.

1 | Introduction

During anaerobic digestion (AD), organic compounds are degraded by anaerobic microorganisms to produce methane, an energy carrier that can be used to generate heat and electricity. AD systems treat and recover energy from liquid and solid waste streams, such as food processing wastewater, wastewater sludge, and food waste (Karki et al. 2021; Parkin and

Owen 1986). However, the low growth rates of anaerobic microorganisms and the inherently complex, interdependent biochemical reactions make AD systems expensive to build and challenging to manage (Briones and Raskin 2003). For example, conventional AD systems consist of large reactors due to the long solids retention time (SRT) needed to prevent biomass washout (LEITAO et al. 2006; Parkin and Owen 1986). Moreover, a sudden change in operating conditions, an influx of

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial](https://creativecommons.org/licenses/by-nc/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2024 The Author(s). *Biotechnology and Bioengineering* published by Wiley Periodicals LLC.

inhibitory materials, or an increase in the organic loading rate (OLR) can lead to disruption in operations and may require reactor shutdowns. Long recovery periods are often needed after disruptions due to the slow growth rates of anaerobic microorganisms (Briones and Raskin 2003; Xing, Criddle, and Hickey 1997). Such disruptions can be difficult to predict or diagnose without mathematical models such as the Anaerobic Digestion Model No. 1 (ADM1), a model that has been developed to help improve the understanding of AD systems while facilitating the design and operation of a variety of AD process configurations (Batstone et al. 2002; Rosen et al. 2006).

ADM1 uses a system of nonlinear ordinary differential equations (ODEs) to describe the essential biochemical reactions and physicochemical processes in AD systems operated as continuous stirred-tank reactors (CSTRs; Batstone et al. 2002). The degree of nonlinearity in this system of ODEs is nontrivial, requiring numerical solvers (Rosen et al. 2006). Selecting the appropriate numerical method to solve the ADM1 system of ODEs is demanding due to its stiffness, a characteristic of ODE systems that greatly challenges the ability to find solutions efficiently (Spijker 1996). Efforts to simplify and speed up the solution of ADM1 have occurred over the years. For example, the differential equations describing H_2 production and charge balance, which are used to calculate the pH, have been converted into algebraic equations that can be solved using the Newton–Raphson method to reduce stiffness, turning the system of ODEs into a system of differential algebraic equations (DAEs). This modified version of ADM1, specifically the MATLAB ADM1 developed by (Rosen et al. 2006), can be solved by an explicit numerical method, such as MATLAB's ODE45, more efficiently than previous approaches (Gernaey et al. 2014).

Several previous studies have extended ADM1 to include new biochemical processes, employing additional strategies to address stiffness. For example, Flores-Alsina et al. (Flores-Alsina et al. 2016) added the speciation, complexation, and precipitation of sulfur, iron, and phosphorus to ADM1. They coupled the DAE approach with a multidimensional Newton–Raphson algorithm reinforced with the Simulated Annealing method to address the stiffness introduced by the additional ionic speciation and charge balance calculations (Flores-Alsina et al. 2015). Another example is Puyol et al. (2018), who utilized a speciation/complexation aqueous-phase chemistry model to resolve the stiffness caused by pH calculation when simulating the addition of zero-valent iron for enhancing biogas production within ADM1. While these examples were successful, implementing similar approaches to reduce stiffness for novel modeling targets requires mathematical solution strategies to be developed on a case-by-case basis. Implicit numerical methods that utilize root-finding algorithms can address ODE system stiffness by reducing the need to take excessively small time-steps (Kelley 2003). Implicit methods can be carried out with MATLAB's own stiff solvers (ODE15s and ODE23s) and have been utilized in other ADM1 implementations, like the ADM1jl by Allen et al. (2024) under the Julia environment. However, the selection of an implicit method can be challenging since their required iterative root finding algorithms are computationally intensive. Higher order implicit methods, such as ODE15s in MATLAB and the fourth order Rosenbrock method used in ADM1jl, can find solutions with less iterations, but are not

guaranteed to be stable. The intrinsically stable backward Euler method, on the other hand, usually requires more iterations and, hence, more computational resources and time. The development of individual solution strategies presents a barrier to more widespread use of ADM1, particularly when new bio-reactor designs are the targets of the modeling effort (Cobbledick et al. 2016; Ersahin et al. 2014; Fairley-Wax, Raskin, and Skerlos 2022; Fonoll et al. 2024; Smith et al. 2012).

In this study, we describe the development and evaluation of ADM1F (F, for fast) as a new modeling platform to overcome stiffness challenges by applying the unconditionally stable backward Euler implicit solver while maintaining a high computational efficiency. We also describe the ADM1F modeling environment that streamlines model configuration, system setup, model runs, and post-simulation data analyses and visualizations. The relative speed and accuracy of ADM1F were assessed via benchmark comparisons to a previously developed and validated ADM1 implementation in MATLAB (Rosen et al. 2006). Two practical demonstrations of the ADM1F solution strategy and modeling environment are provided: (1) a two-phase co-digestion system to enhance biogas production, and (2) a conventional digester to assess scenarios to recover from disruptions that led to instability. We further provide instructions for accessing the open-source ADM1F environment for future modeling scenarios.

2 | Methods

2.1 | Model Description

Figure 1 illustrates the overall structure of the ADM1F computational modeling platform, consisting of the ADM1F core model and an iPython interface. The Supporting Information and the tutorial website (https://elchin.github.io/ADM1F_docs/index.html#) provide details on how to modify and run ADM1F. Examples provided on the tutorial website demonstrate how to customize operating conditions and reactor configurations. The open-source code of the ADM1F core model is available in a GitHub repository (<https://github.com/lanl/ADM1F>).

2.1.1 | iPython Interface

The interface to ADM1F was created with iPython to make ADM1F more accessible. It allows the development of a variety of user-programmable scripts to carry out crucial functions needed for describing the entire AD operations such as (1) describing influent characteristics, which allows for blending of substrates at different ratios to simulate co-digestion; (2) adjusting stoichiometric and kinetic parameters; (3) specifying the number of bioreactors and their working volumes; (4) adjusting operating conditions, such as the temperature and flow rate; (5) selecting steady-state versus dynamic simulations; (6) extracting and processing intermediate outputs at each time step and the final simulation results; (7) connecting the modeling outputs with assessment tools developed by other researchers; and (8) analyzing and visualizing data. Users can

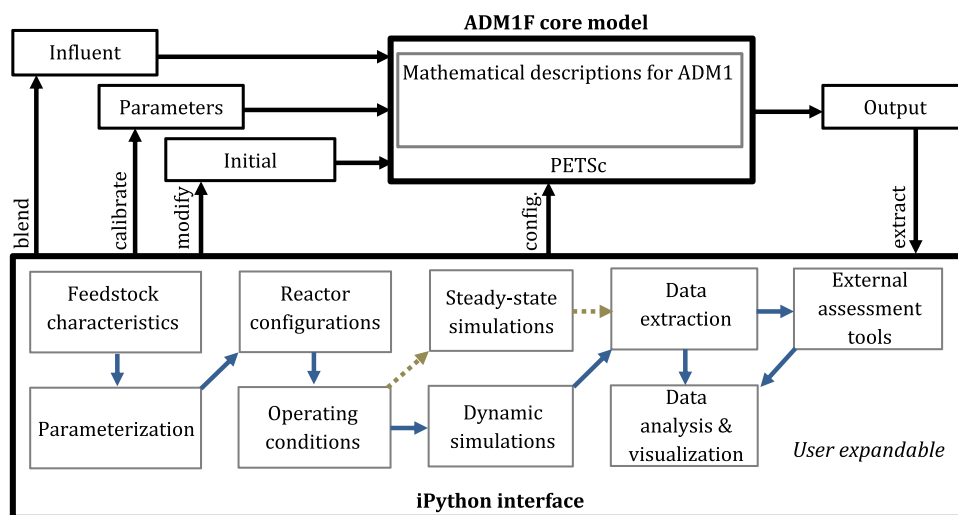


FIGURE 1 | The core-interface structure of the ADM1F modeling platform. The interface includes user-programmable scripts (boxed by the grey blocks) that can be combined to form a complete modeling workflow (connected by the solid blue arrows).

select, customize, and combine these modular scripts to form a complete modeling workflow, such as the one shown in Figure 1. To facilitate optimization and/or uncertainty analysis, users can set up loops of modeling workflows to perform different simulations sequentially while varying influent characteristics, reactor configurations, or operating conditions.

2.1.2 | ADM1F Core Model

The ADM1F core model includes the ADM1 system of equations and computational data structures for the numerical solution of these differential and algebraic equations. Because these equations are stiff and highly nonlinear, an implicit backward Euler with adaptive step size is applied to provide unconditionally stable conditions (Kelley 2003, Keyes, Reynolds, and Woodward 2006). To achieve fast and reliable solutions, the Portable Extensive Toolkit for Scientific Computation (PETSc; Balay et al. 2019; Mills et al. 2021), an open-source numerical method library, was utilized as the computational data structures for ADM1F. Specifically, the efficient matrix and vector data structures in PETSc were used to manage ADM1's model variables and parameters and formulate the mathematical abstraction needed to solve the system of ODEs in ADM1 and DAEs in modified ADM1 versions. Leveraging PETSc's numerical algorithms for implicit time-stepping, non-linear and linear solvers are then applied to solve the system of ODEs or DAEs using the Newton-Krylov method to ensure faster than linear convergence (Kelley 2003; Knoll and Keyes 2004).

The ADM1F core model was written in C++, a compiled language, to further increase computational efficiency (Stroustrup 2013). UNIX syntax is used by PETSc for model configuration and operation, allowing the iPython interface to provide programmable functions for model simulation and data analysis. This architecture offers flexibility to customize typical solver options, such as time-stepping strategies, linear and nonlinear approaches, simulation parameters (e.g., starting and ending time points), and solver tolerances, without the need to

modify and recompile the codes in the ADM1F core model. Recompilation is only needed when the ADM1 system of ODEs/DAEs is modified.

The implementation of ADM1F starts with three input files with a *.dat file extension, which can be edited with text or spreadsheet editors (Supporting Information S1: Table S1). The parameters file contains constants that describe the stoichiometry, kinetics, inhibition, proton dissociation, and mass transfer, as well as environmental factors during AD operation such as temperature and pressure. The influent file contains influent characteristics, such as the influent flow rate, the concentrations of monosaccharides, amino acids, inorganic carbon compounds (bicarbonate and carbonic acid), and inorganic nitrogen compounds (ammonia and ammonium), and biomass levels of different microbial populations (e.g., sugar degraders, amino acid degraders, etc.). The initial conditions file contains the concentrations of substrate components and biomass levels of different microbial populations. The complete list can be found in Supporting Information S1: Table S1. The output of the ADM1F model includes the predicted effluent concentrations of substrate components, inorganic compounds, and dissolved gasses, biomass levels of different microbial populations, and the production rates of different gas compounds (Supporting Information S1: Table S1). They are written to readable files with a *.out file extension at each time step, which can be utilized to carry out various interpretive data analyses during post-processing using the iPython interface.

2.2 | Benchmark Comparison

A previously verified MATLAB implementation of ADM1 developed by Rosen et al. (2006) was used as the reference for benchmark comparisons with ADM1F. This MATLAB implementation of ADM1 (called MATLAB ADM1) is a slightly modified version of the original ADM1 that includes the conversions of the differential equations for calculating the hydrogen concentration and pH into algebraic equations to reduce stiffness. ODE45 was used to solve MATLAB ADM1 as it

has the shortest reported simulation time than other MATLAB numerical solvers, including ODE15s, ODE23, and ODE23tb, when solving the DAE version of MATLAB ADM1 (Gernaey et al. 2014). We also discovered that while ODE15s sometimes requires shorter simulation times than ODE45, it does not consistently produce reliable simulation results. For example, negative values of gas concentrations and biomass concentrations were produced by ODE15s under conditions where ODE45 produced positive values. This frequently occurs when the influent variables are conducive to digester disruptions, such as high carbon-to-nitrogen ratios and low hydraulic retention times. Three representative influent conditions and corresponding output values from ODE45 and ODE15s are included in Supplementary Material 2. An identical version of this modified ADM1 was implemented in ADM1F (called ADM1F DAE) along with an implementation of the original ADM1 without the conversions (called ADM1F ODE). The latter was used to verify the stability of ADM1F under different stiffness conditions. Computation time and simulation results obtained with MATLAB ADM1, ADM1F DAE, and ADM1F ODE were compared under both steady-state and dynamic conditions. The following ODE solver options were used for both MATLAB ADM1 and ADM1F: relTol (relative tolerance) was 10^{-5} ; the timestep was variable; and no non-negativity setting was used to avoid unnecessary errors. All simulations were carried out with a MacBook Pro (2019 model, 2.8 GHz Intel Quad Core i7, 16GB 2133 MHz LPDDR3 RAM) using MATLAB R2021a for the MATLAB ADM1 model and Python 3.7 for ADM1F models. Python packages utilized for ADM1F operations and data analyses include *glob*, *os*, *subprocess*, *numpy*, *scipy*, *pandas*, *lmsdu*, *xlr*, *time*, and *matplotlib*.

2.2.1 | Steady-State Simulations

For the steady-state simulations, the Latin Hypercube method was used to generate 100 different influent files with an upper limit of $110\% \times \text{default value}$ and a lower limit of $90\% \times \text{default value}$ for each variable (McKay et al. 1979). The generated influents are included in the SI with the first row being the default value (Supporting Information S2). This method was selected for its ability to produce near-random samples for high-dimensional files – in this case the 28-variable ADM1 influent files. This resulted in 100 steady-state simulations with randomly varying feed characteristics, organic and hydraulic loading rates, temperatures, and buffer capacities.

2.2.2 | Dynamic Simulations

For the dynamic simulations, three influent data sets were used during the benchmark comparison (the influent data, initial condition, and parameters are included in Supporting Information S2). The first set of influent data (Feed 1) was the 609-day dynamic anaerobic digester influent data of Benchmark Simulation Model No.2 (BSM2; Jeppsson et al. 2007). Primary and secondary sludge characteristics and production rates were calculated from the default BSM2 domestic wastewater input with diurnal and weekly patterns. Feed 1 was generated by converting the sludge mixture from the original activated sludge

model (ASM) format to the ADM1 format by the BSM2 asm2adm converter (Nopens et al. 2009). The hydraulic retention time (HRT) of Feed 1 ranges between 7 and 63 days, with an average of 19 days. Similar to the sensor noise implementation for the BSM2 influent, uniformly distributed stochastic variations with a standard deviation of 2.5% of the maximum value were applied to the concentration of each component in Feed 1 to represent measurement noise (Jeppsson et al. 2007). During the 609 days of simulation, the measurement noise was applied by updating the model influent every 15 min, resulting in a total of 58,464 sets of influent data. Two additional influent data sets, denoted as Feed 2 and Feed 3, representing sludge and domestic wastewater, were also generated following the same 609-day, 15-min influent update interval format. Feed 2 was generated based on Feed 1 but with higher stochastic variations to stress-test the numerical solvers used in ADM1F. It was formed by applying uniformly distributed random factors ranging from 10% to 190% to each 15-min update interval of Feed 1. Feed 3 was the default BSM2 domestic wastewater directly converted to the ADM1 format by the asm2adm converter, which has lower chemical oxygen demand (COD) concentrations and buffer capacities compared to Feed 1 and Feed 2. With an aim to test the applicability of the model to unconventional systems, such as anaerobic wastewater treatment systems and multi-phase AD systems, Feed 3 also has a high flow rate, resulting in an average HRT of 0.5 days, ranging from 0.3 to 1 day.

To implement the dynamic simulation with frequent updates of influent data in ADM1F, a script was created in the iPython interface to perform 58,464 15-min simulations with separate influent files. The model output from the previous 15-min time interval was extracted and used as the initial condition for the next 15-min interval, forming a stitched time-interval continuous dynamic simulation. To investigate the effect of influent update interval on the computation time during dynamic simulations, influent data sets with update intervals of 60, 240, and 1440 min were generated for all three substrates by averaging the original influent data (15 min update interval) within the new timeframe. The default initial condition and parameters in BSM2 were used for the MATLAB ADM1, ADM1F DAE, and ADM1F ODE models run as dynamic simulations. MATLAB ADM1 used the default explicit ODE45 solver while ADM1F DAE and ADM1F ODE used the implicit Euler method with adaptive time-stepping (Butcher 2016; Jeppsson et al. 2007).

2.2.3 | Quantification of Errors

The root-mean-square errors (RMSE) was calculated for the output of each simulation for both steady-state and dynamic simulations by Equation 1:

$$\text{RMSE} = \sqrt{\frac{\sum_i^N (y_{\text{Model1},i} - y_{\text{Model2},i})^2}{N}}, \quad (1)$$

where N can be either the number of simulations or output variables. $y_{\text{Model1},i}$ represents the output from the first model (e.g., MATLAB ADM1) while $y_{\text{Model2},i}$ represents the output from the second model (e.g., ADM1F DAE). RMSE was normalized by \bar{y}_{Model1} , which is the mean of model 1 (Eq. 2).

$$\text{Normalized RMSE} = \frac{\text{RMSE}}{\bar{y}_{\text{Model1}}} \quad (2)$$

2.3 | Modeling Scenarios Implemented by ADM1F

To demonstrate the automation of modeling workflows, two scenarios were investigated with ADM1F DAE. Throughout this work, ADM1F DAE has the identical system of equations as MATLAB ADM1 developed by Rosen et al. (2006), while being implemented and solved in the ADM1F environment.

2.3.1 | Scenario 1—Two-Phase Co-Digestion System

In the first scenario, a two-phase AD system with a first-phase HRT of 4 days and a second-phase HRT of 16 days co-digesting wastewater sludge and food waste was set up using the iPython interface for steady-state simulations. The effects of adding food waste in addition to sludge on reactor performance (e.g., methane production rate and overall AD stability) were evaluated using a stability score established by Cook et al. (Cook et al. 2017). The food waste contribution was represented by the percent increase of the OLR due to the addition of food waste, while food waste characteristics were represented by the fats, oils, and grease (FOG) content and carbon to nitrogen (C/N) ratio. FOG has high energy density but may also result in reactor disruptions caused by the accumulation of long chain fatty acids (LCFAs) and volatile fatty acids (VFAs; Palatsi et al. 2009, 2010, Silvestre et al. 2014). The C/N ratio influences the buffer capacity of the feed; a high C/N ratio causes biomass growth limitation due to the lack of nitrogen and a low C/N ratio can result in process inhibition by ammonia. Table 1 shows a set of stability indicators and their scoring strategies (Cook et al. 2017), which were used to calculate stability scores for 1000 combinations of food waste contributions and characteristics.

2.3.2 | Scenario 2—Recovery From Disruption in a Single-Stage CSTR AD

The second scenario considered the effectiveness of recovery strategies for a conventional digester after performance disruption caused by an increase in OLR (kg VS/m³ reactor/day). Model simulations were used to determine the recovery time resulting from adding buffer (NaOH) and decreasing the OLR by reducing the influent flow rate. The digester was modeled with a 20-day retention time (HRT = SRT) operating initially at steady state. The digester was then perturbed by increasing the OLR by 200% for 4 days, which reduced the retention time to 6.7 days. Combinations of the two recovery strategies were evaluated with steady-state simulations using the perturbed digester as the initial condition. Stability scores were calculated using model outputs extracted at each time step with an iPython script. Then, the time required for the reactor stability score to recover to at least 0.7 was reported for each combination. The stability score of 0.7 was chosen to represent a stable digester because it was observed during the simulations that once the stability score became higher than this threshold, it never decreased below it.

3 | Results and Discussion

3.1 | Benchmark Comparison

3.1.1 | Steady-State Simulations

In the 100 steady-state simulations conducted with 100 different influent files generated using the Latin Hypercube design, both ADM1F DAE and ADM1F ODE achieved an average computation time of 0.16 s per simulation (Figure 2). Despite the difference in stiffness between ADM1F ODE and ADM1F DAE as the result of converting the ODEs for hydrogen and pH to DAEs, no significant change in computation time was observed. The 0.16 s average computation time represents a reduction of over 99.9% compared to the 20.46 s average computation time observed using MATLAB ADM1 (Figure 2). This increased computational speed is important for model runs that conventionally require hours to days to complete using personal computers, such as the scenario discussed in Section 3.2.1 below. Moreover, ADM1F DAE and ADM1F ODE showed greater consistency in computation time compared to MATLAB ADM1 (Figure 2). This can be useful when it is necessary to accurately budget simulation time.

The steady-state simulation outputs from ADM1F DAE were similar to those from MATLAB ADM1, with normalized RMSEs averaged over all output variables ranging from 2×10^{-6} to 1.7×10^{-5} (Figure 3B). For each of the output variables, the average normalized RMSE over the 100 simulations was below 1.5×10^{-5} except for the anion concentration (S_{an-}) and cation concentration (S_{cat+}) (Figure 3A). The high normalized RMSE values were because S_{an-} and S_{cat+} are 0.04 and 0.02, respectively, in the initial condition but are 0 in the influent. MATLAB ADM1 converged near 10^{-11} at steady state for these two values while ADM1F converged at 10^{-14} . Since there was no difference in the model description of ADM1F DAE and MATLAB ADM1, these errors resulted from the differences in the numerical methods. There was practically no difference in the steady-state simulation results between ADM1F DAE and ADM1F ODE.

Modifications made to the ADM1 mathematical system, such as the conversion of the algebraic equations describing the H₂ and proton mass balance into differential equations, made little difference in the computation time and accuracy, likely because the implicit Newton-Krylov solver used in ADM1F is particularly efficient in solving large-scale, nonlinear, complex systems involving processes with disparate timescales (Keyes, Reynolds, and Woodward 2006; Rudi et al. 2015). This solver's benefits are likely to reduce the need for stiffness-reduction measures in future ADM1 extensions, particularly for steady-state calculations. This includes scenarios such as biomass retention in membrane bioreactors (Batstone et al. 2002) and mass transport in attached-growth systems (Boltz et al. 2009). Additionally, these benefits may simplify solving the system of equations for other modeling targets, such as the effects of zero-valent iron on biogas production and the interactions between sulfate, iron reduction, and phosphate. (Flores-Alsina et al. 2016, Puyol et al. 2018). This, combined with the high modeling throughput of ADM1F, has the potential to simplify the workflow for the implementation and validation of ADM1-based models, which can facilitate the development and optimization of novel AD technologies (Cobbledick et al. 2016; Ersahin et al. 2014;

TABLE 1 | Stability indicators and criteria used to calculate stability scores for anaerobic digesters, adapted from (Cook et al. 2017). Green scores are more stable and red scores are less stable.

Stability indicators	Units	Values assigned to stability scores									
		0	0.25	0.5	0.75	1					
Biogas composition	% methane (by volume)	52.25	55.00	57.75	60.50						
		0	0.25	0.5	0.75	1	0.75	0.5	0.25	0	
pH	(no unit)	5.80	6.10	6.40	6.70	7.47	7.88	8.30	8.72		
Alkalinity	g/L as CaCO ₃	1.90	2.00	2.10	2.20	18.0	19.0	20.0	21.0		
						1	0.75	0.5	0.25	0	
Free ammonia	mg NH ₃ -N/L					180	190	200	210		
Ammonium	mg NH ₄ ⁺ -N/L					2925	3088	3250	3413		
Total VFAs	mg COD/L					4500	4750	5000	5250		
LCFAs	mg COD/L					1260	1330	1400	1470		

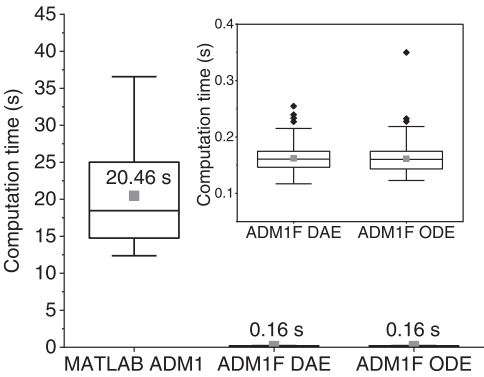


FIGURE 2 | Steady-state computation time comparison for 100 simulations with different influent files for MATLAB ADM1, ADM1F DAE, and ADM1F ODE. The upper-right box shows results for ADM1F DAE and ADM1F ODE using a different y-axis. The boxes indicate the 25th and 75th percentiles; the error bars represent the 1.5 interquartile ranges; the gray squares represent the means; the black diamonds represent outliers; the numbers represent the mean computation times.

Fairley-Wax, Raskin, and Skerlos 2022; Fonoll et al. 2024; Smith et al. 2012). Additionally, the high simulation speed and the access to extensive Python libraries of the ADM1F platform can accelerate sensitivity analyses to identify key variables impacting reactor performance and quantify the uncertainties caused by poorly characterized feedstocks (Arnell et al. 2016).

3.1.2 | Dynamic Simulations

Dynamic simulations represented diurnal and weekly patterns of influent change at a 15-min interval over 609 days of simulated co-digestion, as well as stochastic sensor noise in influent characteristics. As with the steady-state simulations, the differences in simulation results between MATLAB ADM1,

ADM1F DAE, and ADM1F ODE were small during the dynamic simulations for Feed 1, as shown in Figure 4. For instance, little difference was shown between ADM1F DAE and ADM1F ODE. The normalized RMSEs between the models did not exceed 0.0015 (Supporting Information S1: Figure S1). The comparison between the dynamic simulations of MATLAB ADM1 and ADM1F ODE followed the same pattern as the comparison between MATLAB ADM1 and ADM1F DAE (Supporting Information S1: Figure S4A,C).

The normalized RMSE deviations between ADM1F DAE and MATLAB ADM1 over 58,464 points compared for Feed 1 was on average higher than that between ADM1F DAE and ODE and between ADM1F ODE and MATLAB ADM1, which can be seen in Figure 4. This was, in part, driven by the high outliers at the beginning of the dynamic simulation (Supporting Information S1: Figure S4A). The difference in normalized RMSE reduced but did not disappear in the subsequent time steps (Supporting Information S1: Figure S4A).

While the model output of MATLAB ADM1, ADM1F DAE, and ADM1F ODE were similar during dynamic simulation, their computation times were highly variable. It took both ADM1F DAE and ADM1F ODE about 60 min to complete the dynamic simulation with an influent update interval of 15 min for all three types of feeds (Figure 5A). In contrast, the computation times for MATLAB ADM1 were between 2 and 4 min for Feed 1 and Feed 2 (sludges with significant stochastic variations at 19-day HRT), and about 150 min for Feed 3 (domestic wastewater at a 0.5-day HRT). It is notable that MATLAB ADM1 implementation was much faster than ADM1F for Feeds 1 and 2, and much slower for Feed 3.

The MATLAB implementation was sensitive to the feed while the ADM1F computation times were highly sensitive to the update interval, dropping significantly as update intervals were

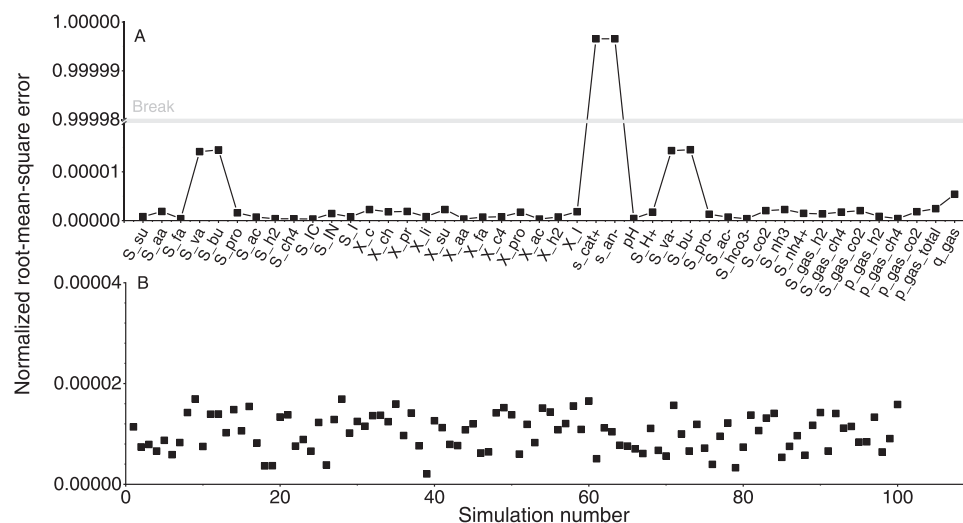


FIGURE 3 | The normalized root-mean-square errors (RMSEs) between the steady-state modeling results of MATLAB ADM1 and ADM1F DAE. (A) shows the normalized RMSEs for each output variable averaged for all 100 simulations; (B) shows normalized RMSEs for each simulation averaged for all output variables. The grey lines highlight breaks in the y-axis. There was no difference between ADM1F DAE and ADM1F ODE steady-state outputs (normalized RMSE = 0). Therefore, their normalized RMSEs are not shown.

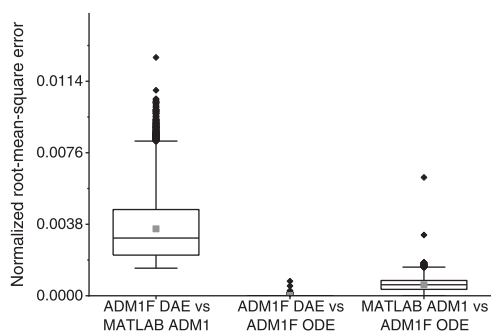


FIGURE 4 | The normalized root-mean-square errors (RMSE) between the dynamic modeling results of ADM1F DAE, ADM1F ODE, and MATLAB ADM1 for Feed 1. The boxes indicate the 25th and 75th percentile; the error bars represent the 1.5 interquartile range; the grey squares represent the means; the black diamonds represent outliers.

increased as seen in Figure 5. Specifically, increasing the influent update interval from every 15 min to every hour, every 4 h, and every 24 h reduced the overall computation time to about 15, 4, and under 1 min, respectively (Figure 5A). For MATLAB ADM1, increasing the influent updating interval from every 15 min to every 24 h only reduced the computation time by about 50% for Feed 1 and Feed 2 (Figures 5A and Supporting Information S1: S10A). When Feed 3 (domestic wastewater at a 0.5-day HRT) was used as the influent, MATLAB ADM1 computation time increased by around 5% with the update interval (Figure 5A).

The difference in how the dynamic simulation computation time responds to the influent update interval between ADM1F and MATLAB ADM1 primarily resulted from their different dynamic simulation implementations and numerical solver time-stepping strategies. In MATLAB ADM1, the dynamic simulation was performed continuously with an input matrix containing influent characteristics and the timestamps at which influent characteristics should be updated. The timesteps that

the solver takes during dynamic simulations are not affected by the timestamps but are instead influenced by the feed characteristics (Supporting Information S1: Figures S10A and S11A). Compared to Feed 1 and 2, MATLAB ADM1 took over 98% smaller timestep sizes when solving Feed 3, resulting in over 37 times longer computation times, possibly because of the challenges with pH calculations due to the lack of buffering capacity (Supporting Information S1: Figure S11A). On the contrary, the implementation of dynamic simulation with ADM1F is designed to perform independent simulations sequentially for each time interval that corresponds to a specific influent. The output from the previous time interval is converted to the initial condition for the following time interval with an iPython script. This allows the timestep size to increase to large values when influent characteristics are not updated frequently (Supporting Information S1: Figure S9). However, each update of the influent characteristics starts a new simulation, resetting the timestep size to the small default initial value. As a result, although the ADM1F maintained a low computation time for each interval between 0.07 and 0.1 s regardless of its duration, frequent timestep resets caused by constant updates of influent characteristics can prevent the model from taking long time-steps and lead to a long overall computation time (Supporting Information S1: Figure S9). Given this result, future work can consider more aggressive time-stepping methods available in PETSc, such as pseudo-transient continuation (Kelley and Keyes 1998). However, such methods may not converge, and research is needed to understand convergence likelihoods and their overall potential to reduce dynamic simulation times in the ADM1F environment. For instance, one would expect a conventional CSTR AD with a long HRT of over 15 days treating typical wastewater sludge to have better convergence characteristics than reactors with shorter HRTs of 2–3 days, which are more sensitive to changes in influent characteristics and flow rates, for example, the acid phase of a two-phase AD system. Alternatively, increasing the influent update interval would decrease the frequency of timestep resets in the ADM1F environment, allowing the implicit time-stepping scheme to

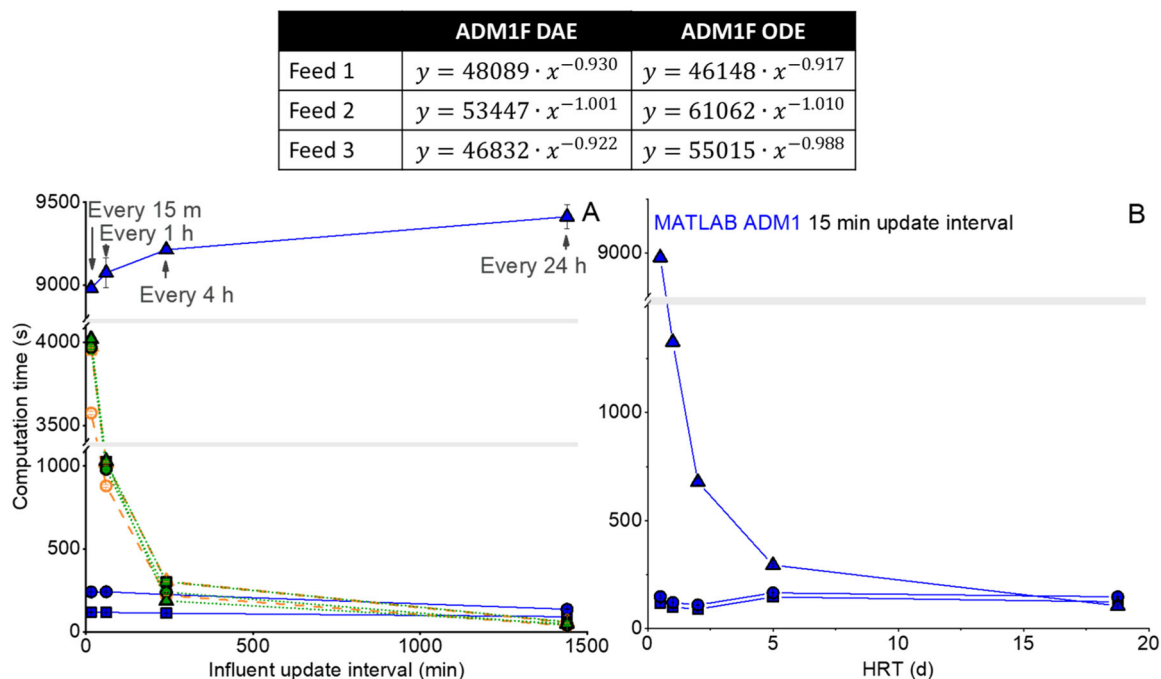


FIGURE 5 | Relationships between the computation times of MATLAB ADM1, ADM1F DAE, and ADM1F ODE and influent update frequencies (i.e., how often the model receives a new input) (A) and between MATLAB ADM1 computation time and hydraulic retention time (HRT) (B). Feed 1 is represented by squares (■); Feed 2 is represented by circles (●); and Feed 3 is represented by triangles (▲). The grey horizontal bars (—) indicate breaks on the y-axis. The table presents power functions predicting the relationship between the computation time(s), represented by y , and the influent update interval (min), represented by x , in ADM1F DAE and ODE models.

take larger timestep sizes, hence, fewer time steps and shorter simulation time (Supporting Information S1: Figures S10B and S10C).

For Feed 1 and Feed 2, ADM1F DAE and ODE showed computation time advantages over MATLAB ADM1 when the influent update interval was more than 4 h. This makes ADM1F more suitable to simulate reactors that are less sensitive to changes in influent characteristics, such as conventional anaerobic digesters, which are CSTRs with long retention times ($HRT = SRT$). For a CSTR with a 19-day HRT, the predicted performance by ADM1F DAE at a 4-h influent update interval successfully captured the diurnal variation and remained consistent with the predictions at 15-min and 1-h update intervals (Supporting Information S1: Figure S5). Four hours may be a practical limit for increasing the update interval, as diurnal changes, for instance, can be missed at larger update intervals as seen in Supporting Information S1: Figure S6.

MATLAB ADM1 demonstrated lower dynamic simulation computation times than the ADM1F models when the influent update interval was shorter than 4 h. This makes MATLAB ADM1 desirable for scenarios that require frequent influent updates, such as reactor startup where feeding strategies and reactor conditions change rapidly, and systems whose performances are sensitive to rapid changes of influent characteristics, such as high-rate systems with very short HRTs (Eftaxias et al. 2021; Rico et al. 2015). The low impact of influent update interval on computation time by MATLAB ADM1 also makes it more compatible with automated systems whose operations are based on sensor data, which may feed measurements to the model more frequently than every 15 min (Nguyen et al. 2015).

However, low simulation times at short influent update intervals are not guaranteed by MATLAB ADM1 as shown by the long computation times for Feed 3. Whereas simulation time was not dependent on HRT for Feed 1 and Feed 2 (Figure 5B), it was seen that increasing the HRT for Feed 3 to 19 days reduced MATLAB ADM1 simulation time to less than 2 min. Further study is required to understand why the MATLAB ADM1 simulation time was dependent on the feed to such a great extent. However, given this finding, test runs may be needed for MATLAB ADM1 to determine whether the conditions to be simulated would result in long computation times.

In contrast, low variation and high predictability were observed in the ADM1F computation times across feeds. A power function relationship was observed and estimated between influent update interval and simulation time for ADM1F ODE and DAE with R^2 values > 0.99 (Figure 5). Given that the computation time can be estimated a priori based on the update interval of the influent for ADM1F models, regardless of the influent characteristics, ADM1F can be useful in cases where the influent characteristics could greatly impact MATLAB simulation times. The ADM1F environment appears to be favorable for dynamic simulations with longer update intervals regardless of the feed. Longer update periods are suitable for common systems such as conventional CSTRs, the most widely adopted AD system.

3.2 | Modeling Scenarios Implemented by ADM1F

Two modeling scenarios are presented here to demonstrate how ADM1F can simplify modeling tasks. These two scenarios were

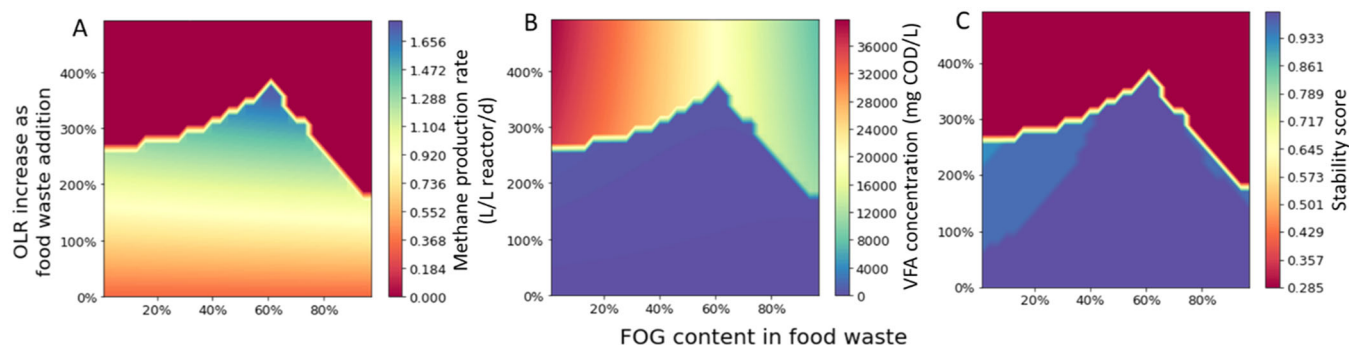


FIGURE 6 | The predicted impacts of food waste contributions, represented by OLR increase (y-axes), and characteristics, represented by FOG content as a percentage of the food waste COD (x-axes), on methane production rate (A), volatile fatty acid (VFA) concentration (B), and stability score (C) in a two-phase anaerobic digestion system that co-digests food waste with wastewater sludge. A higher stability score suggests that the system is more stable.

chosen due to their need to simulate and analyze up to 1000 different conditions, which can be time-consuming and labor-intensive when performed with MATLAB ADM1.

3.2.1 | Scenario 1—Two-Phase Co-Digestion AD System

Scenario 1 simulated the performance of a two-phase mesophilic anaerobic digester co-digesting wastewater sludge with food waste to illustrate the value of ADM1F's ability to perform rapid calculations. One thousand combinations of food waste loading rate and FOG content were evaluated for their predicted methane production rate and a calculated stability score. The food waste contribution was represented as an increase in the baseline OLR compared with the mono-digestion of sludge. The food waste characteristics were represented by the food waste FOG content, ranging from 0% to 100% of the food waste COD. The carbohydrate to protein ratio was kept constant, so increasing the FOG content decreased the protein content and increased the C/N ratio in the feed. For example, when the FOG content of the food waste was increased from 20% to 40%, the percentage of carbohydrates and proteins in the food waste each were decreased from 40% to 30%. The complete evaluation, including final data visualization (Supporting Information S1: Figure S12), took ADM1F DAE around 12 min whereas MATLAB ADM1 would have required over 11 h of computation time plus additional time for data extraction, analysis, and visualization.

The results shown in Figure 6 demonstrate a clear threshold of food waste contribution that separates stable reactor operation (blue-green) from unstable reactor operation (yellow-red). Below this threshold, the methane production rate increased with increasing food waste contribution without a significant reduction of the reactor stability score. Once the food waste contribution increased above the threshold, the predicted methane yield sharply decreased with little transition due to the accumulation of inhibitors like VFAs (Figure 6A,B). This also caused a decrease of the stability score (Figure 6C) as it was calculated from the biogas methane composition and common inhibitor concentrations (Table 1). The food waste tolerance, defined as the minimum predicted food waste contribution that caused reactor instability, changed with the food waste characteristics, represented by the FOG content. The highest

tolerance was observed when the food waste FOG content was 60%. For lower food waste FOG contents, the higher protein contents and corresponding higher ammonia concentrations inhibited methanogenesis, leading to a lower food waste tolerance. For higher FOG contents, the accumulation of hydrogen, the nitrogen limitation due to low protein content, and the limited buffer capacity inhibited both acetogenesis and methanogenesis, reducing food waste tolerance.

The relatively fast ADM1F simulations were made possible by both the simulation speed enabled by the PETSc numerical library and the convenience of the customizable workflows that unify the blending of co-substrates, simulation, data analysis, and visualization under one coherent environment. The iPython interface connects ADM1 model outputs with mathematical and computational tools that perform a variety of performance assessments, troubleshooting analyses, and solution-seeking techniques. This includes the stability tool employed in Figure 6C and the potential integration with existing parameter calibration and process optimization algorithms (Ahmed and Rodríguez 2017; Cook, Ragsdale, and Major 2000) and frameworks that facilitate life cycle assessments and techno-economic analyses to evaluate the environmental and economic sustainability of various system designs and application scenarios (Nyitrai et al. 2023).

3.2.2 | Scenario 2—Recovery from Disruption in a Single-Stage CSTR AD

Scenario 2 evaluated recovery strategies after a reactor disruption to showcase how ADM1F's ability to enable analysis automation that would otherwise be labor-intensive using MATLAB ADM1 by utilizing the iPython interface that is deeply integrated into the ADM1F core model. Scenario 2 considers how to select a mitigation strategy for recovering a conventional anaerobic digester disrupted by an increase of the OLR by 200% for 4 days. The OLR increase caused an accumulation of VFAs and inhibition of methanogenesis. This decreased digester pH, reduced biogas production, and decreased the predicted stability score to 0.374 by the end of the fourth day. Two recovery strategies were considered to improve the stability of the reactor: raising the pH and reducing the influent flow rate. Adding NaOH can raise the reactor pH to

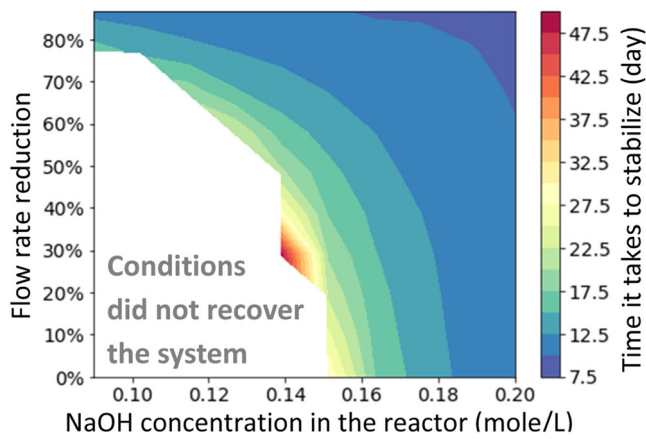


FIGURE 7 | The predicted impacts of two recovery strategies, flow rate reduction and NaOH concentration maintained in the reactor from NaOH dosing, on the recovery time of a single-stage CSTR anaerobic digestion that brings the reactor stability score above 0.7 after a disruption due to a 200% influent flow rate increase for 4 days.

the optimal range for methanogens (7–8), and the flow rate reduction can limit the further production of VFAs, allowing the methanogens to consume the accumulated VFAs.

An ADM1F workflow was built to estimate the recovery times required to increase the stability score above 0.7 considering 1000 combinations of NaOH dosages and flow rate reductions. ADM1F DAE was set up to allow an iPython script to calculate the stability score at each time step during steady-state simulation and report the time at which the stability score reached a value of 0.7. The same method cannot be applied using MATLAB ADM1 as the steady-state intermediate outputs at each time step are not made available while dynamic simulations are time-consuming to set up and perform. ADM1F took 6.5 min to produce all model-predicted recovery times shown in Figure 7 (Supporting Information S1: Figure S12). The results show that a reduction of the influent flow rate by 80% can recover digester performance in about 15 days without NaOH addition. This could be a useful strategy for facilities with sufficient storage or stabilization capacity. It was also observed that a continuous NaOH dosage to maintain a concentration higher than 0.16 mol/L in the reactor may recover the digester in about 22 days without influent flow rate reduction. Combining NaOH addition and flowrate reduction led to longer recovery times when NaOH was added to maintain a concentration of 0.14 mol/L in the reactor and the flow rate was lowered by 30%, as seen in Figure 7. This specific example illustrates the sensitivity of the simulated system with respect to the magnitudes of NaOH dosage and flow rate reduction. It also demonstrates the utility of employing ADM1 models to both comprehensively assess the amounts of these strategies applied and to understand how long such recovery strategies might require to be effective.

It is worth noting that the ADM1 equations used in ADM1F to model this scenario were identical to those in MATLAB ADM1 (Rosen et al. 2006), and did not include more recent descriptions for process inhibitions caused by LCFAs and VFAs (Palatsi et al. 2010; Xiao et al. 2013). Since stiffness had no impact on the speed and accuracy of ADM1F, future modification to the ADM1 equations under ADM1F should not require additional stiffness-reduction strategies.

4 | Conclusion

An open-source computational modeling platform, named ADM1F (F for fast), was developed to expedite the development, design, and management of AD systems. ADM1F comprises a numerically efficient and stable ADM1F core model, complemented by a versatile iPython interface. This platform enhances simulation speed for steady-state simulations while maintaining a high accuracy relative to a baseline MATLAB ADM1 model regardless of the stiffness in the mathematical system and across various influent characteristics. In dynamic simulations, ADM1F was shown to be more suitable for systems less sensitive to influent variations, such as conventional anaerobic digesters that do not require high update frequencies of influent characteristics. MATLAB ADM1 was shown to be faster when higher update frequencies were necessary, though this result was dependent on the influent characteristics.

The iPython interface for ADM1F provides an environment to set up workflows that integrate modeling tasks, such as configuring the AD system, blending substrates, running simulations, and performing post-simulation data analyses and visualization, allowing users to automate these tasks in large batches. The access to well-established mathematical and statistical libraries provided by the iPython interface and its deep integration with the ADM1F core model allows users to combine existing methods with their own modeling tools and strategies. To illustrate this point, two scenario-based experiments were presented that show how ADM1F can help researchers and operators address challenging design and operational conditions. Future developments of ADM1F are needed to further understand dynamic simulation strategies, establish workflows to streamline the expansion and validation of the ADM1 mathematical system, and interface the model environment with automatic data collection and control systems.

Author Contributions

Kuang Zhu: developed the ADM1F core model, developed the iPython interface, designed and performed simulations, write the manuscript. Wenjuan Zhang: developed the iPython interface, created the GitHub Repository and tutorials, reviewed the manuscript. Elchin Jafarov: developed the ADM1F core model, developed the iPython interface, created the GitHub Repository and tutorials, reviewed the manuscript. Satish Karra: developed the ADM1F core model, reviewed the manuscript. Kurt Solander: advised on the model development, reviewed the manuscript. Meltem Urgan Demirtas: Secured the funding, advised on the model simulations, reviewed the manuscript. Lutgarde Raskin: Secured the funding, advised on the model development and simulations, reviewed the manuscript. Steven Skerlos: Secured the funding, advised on the model development and simulations, reviewed the manuscript.

Acknowledgments

The authors would like to acknowledge Sonja Gagen for early contributions to this study and Tim Fairley-Wax, Renata Starostka, Pedro Puente, and Renisha Karki for suggestions during the writing process. Funding was provided by the US Department of Energy Office of Energy Efficiency and Renewable Energy Bioenergy Technologies Office (BETO) under contract numbers DE-AC02-06CH11357 with Argonne

National Laboratory and DE-EE0009284 with the University of Michigan.

Data Availability Statement

The codes of ADM1F model described this study, the model initial condition file, the parameter file, and the influent files used in steady state simulation and the scenario tests are openly available in a GitHub repository at <https://github.com/lanl/ADM1F>. Specifically, the model codes are in the subfolder: <https://github.com/lanl/ADM1F/tree/main/build>; the initial condition and parameter files are in the subfolder: <https://github.com/lanl/ADM1F/tree/main/simulations>; and the influent files are in the subfolder: <https://github.com/lanl/ADM1F/tree/main/simulations/data>. The tutorial for the model is available at https://elchin.github.io/ADM1F_docs/index.html#. The influent data used for dynamic in are made available in an Excel spreadsheet in Supplementary Materials 2. The MATLAB ADM1 model is available at <https://github.com/wwtmodels/Anaerobic-Digestion-Models>, a GitHub repository created and managed by the Process and System Engineering Center at Technical University of Denmark and the Division of Industrial Electrical Engineering and Automation at Lund University.

References

- Ahmed, W., and J. Rodríguez. 2017. "Generalized Parameter Estimation and Calibration for Biokinetic Models Using Correlation and Single Variable Optimisations: Application to Sulfate Reduction Modelling in Anaerobic Digestion." *Water Research* 122: 407–418. <https://doi.org/10.1016/j.watres.2017.05.067>.
- Allen, C., A. Mazanko, N. Abdehagh, and H. Eberl. 2024. "ADM1jl: A Julia Implementation of the Anaerobic Digestion Model 1." *SoftwareX* 26: 101682. <https://doi.org/10.1016/j.softx.2024.101682>.
- Arnell, M., S. Astals, L. Åmand, D. J. Batstone, P. D. Jensen, and U. Jeppsson. 2016. "Modelling Anaerobic Co-Digestion in Benchmark Simulation Model No. 2: Parameter Estimation, Substrate Characterisation and Plant-Wide Integration." *Water Research* 98: 138–146. <https://doi.org/10.1016/j.watres.2016.03.070>.
- Balay, S., S. Abhyankar, M. Adams, et al. 2019. "PETSc Users Manual Rev 3.12." Lemont, IL, USA: Argonne Scientific Publications.
- Batstone, D. J., J. Keller, I. Angelidaki, et al. 2002. "The IWA Anaerobic Digestion Model No 1 (ADM1)." *Water Science and Technology* 45: 65–73. <https://doi.org/10.2166/wst.2008.678>.
- Boltz, J. P., B. R. Johnson, G. T. Daigger, and J. Sandino. 2009. "Modeling Integrated Fixed-Film Activated Sludge and Moving-Bed Biofilm Reactor Systems I: Mathematical Treatment and Model Development." *Water Environment Research* 81: 555–575. <https://doi.org/10.2175/106143008x357066>.
- Briones, A., and L. Raskin. 2003. "Diversity and Dynamics of Microbial Communities in Engineered Environments and Their Implications for Process Stability." *Current Opinion in Biotechnology* 14: 270–276. [https://doi.org/10.1016/S0958-1669\(03\)00065-X](https://doi.org/10.1016/S0958-1669(03)00065-X).
- Butcher, J. C. 2016. *Numerical Methods for Ordinary Differential Equations*. Hoboken, NJ, USA: Wiley.
- Cobbledick, J., N. Aubry, V. Zhang, S. Rollings-Scattergood, and D. R. Latulippe. 2016. "Lab-Scale Demonstration of Recuperative Thickening Technology for Enhanced Biogas Production and Dewaterability in Anaerobic Digestion Processes." *Water Research* 95: 39–47. <https://doi.org/10.1016/j.watres.2016.02.051>.
- Cook, D. F., C. T. Ragsdale, and R. L. Major. 2000. "Combining a Neural Network With a Genetic Algorithm for Process Parameter Optimization." *Engineering Applications of Artificial Intelligence* 13: 391–396. [https://doi.org/10.1016/S0952-1976\(00\)00021-X](https://doi.org/10.1016/S0952-1976(00)00021-X).
- Cook, S. M., S. J. Skerlos, L. Raskin, and N. G. Love. 2017. "A Stability Assessment Tool for Anaerobic Codigestion." *Water Research* 112: 19–28. <https://doi.org/10.1016/j.watres.2017.01.027>.
- Eftaxias, A., D. Georgiou, V. Diamantis, and A. Aivasidis. 2021. "Performance of an Anaerobic Plug-Flow Reactor Treating Agro-Industrial Wastes Supplemented With Lipids at High Organic Loading Rate." *Waste Management & Research: Journal for a Sustainable Circular Economy* 39: 508–515. <https://doi.org/10.1177/0734242X21991898>.
- Ersahin, M. E., H. Ozgun, Y. Tao, and J. B. van Lier. 2014. "Applicability of Dynamic Membrane Technology in Anaerobic Membrane Bioreactors." *Water Research* 48: 420–429. <https://doi.org/10.1016/j.watres.2013.09.054>.
- Fairley-Wax, T., L. Raskin, and S. J. Skerlos. 2022. "Recirculating Anaerobic Dynamic Membrane Bioreactor Treatment of Municipal Wastewater." *ACS ES&T Engineering* 2(5): 842–852. <https://doi.org/10.1021/ACSESTENG.1C00394>.
- Flores-Alsina, X., C. Kazadi Mbamba, K. Solon, et al. 2015. "A Plant-Wide Aqueous Phase Chemistry Module Describing pH Variations and Ion Speciation/Pairing in Wastewater Treatment Process Models." *Water Research* 85: 255–265. <https://doi.org/10.1016/j.watres.2015.07.014>.
- Flores-Alsina, X., K. Solon, C. Kazadi Mbamba, et al. 2016. "Modelling Phosphorus (P), Sulfur (S) and Iron (Fe) Interactions for Dynamic Simulations of Anaerobic Digestion Processes." *Water Research* 95: 370–382. <https://doi.org/10.1016/j.watres.2016.03.012>.
- Fonoll, X., K. Zhu, L. Aley, S. Shrestha, and L. Raskin. 2024. "Simulating Rumen Conditions Using an Anaerobic Dynamic Membrane Bioreactor to Enhance Hydrolysis of Lignocellulosic Biomass." *Environmental Science & Technology* 58: 1741–1751. <https://doi.org/10.1021/acs.est.3c06478>.
- Gernaey, K. V., U. Jeppsson, P. A. Vanrolleghem, and J. B. Copp. 2014. *Benchmarking of Control Strategies for Wastewater Treatment Plants*. London, UK: IWA Publishing.
- Jeppsson, U., M. N. Pons, I. Nopens, et al. 2007. "Benchmark Simulation Model No 2: General Protocol and Exploratory Case Studies." *Water Science and Technology* 56: 67–78. <https://doi.org/10.2166/wst.2007.604>.
- Karki, R., W. Chuenchart, K. C. Surendra, et al. 2021. "Anaerobic Co-Digestion: Current Status and Perspectives." *Bioresource Technology* 330: 125001. <https://doi.org/10.1016/j.biortech.2021.125001>.
- Kelley, C. T. 2003. *Solving Nonlinear Equations With Newton's Method, Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics.
- Kelley, C. T., and D. E. Keyes. 1998. "Convergence Analysis of Pseudo-Transient Continuation." *SIAM Journal on Numerical Analysis* 35: 508–523.
- Keyes, D. E., D. R. Reynolds, and C. S. Woodward. 2006. "Implicit Solvers for Large-Scale Nonlinear Problems." *Journal of Physics: Conference Series* 46: 433–442. <https://doi.org/10.1088/1742-6596/46/1/060>.
- Knoll, D. A., and D. E. Keyes. 2004. "Jacobian-Free Newton-Krylov Methods: A Survey of Approaches and Applications." *Journal of Computational Physics* 193: 357–397. <https://doi.org/10.1016/j.jcp.2003.08.010>.
- Leitao, R., A. Van Haandel, G. Zeeman, and G. Lettinga. 2006. "The Effects of Operational and Environmental Variations on Anaerobic Wastewater Treatment Systems: A Review." *Bioresource Technology* 97: 1105–1118. <https://doi.org/10.1016/j.biortech.2004.12.007>.
- McKay, M. D., R. J. Beckman, and W. J. Conover. 2000. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code." *Technometrics* 42: 55–61. <https://doi.org/10.1080/00401706.2000.10485979>.
- Mills, R. T., M. F. Adams, S. Balay, et al. 2021. "Toward Performance-Portable PETSc for GPU-Based Exascale Systems." *Parallel Computing* 108: 102831. <https://doi.org/10.1016/j.parco.2021.102831>.

- Nguyen, D., V. Gadhamshetty, S. Nitayavardhana, and S. K. Khanal. 2015. "Automatic Process Control in Anaerobic Digestion Technology: A Critical Review." *Bioresource Technology* 193: 513–522. <https://doi.org/10.1016/j.biortech.2015.06.080>.
- Nopens, I., D. J. Batstone, J. B. Copp, et al. 2009. "An Asm/Adm Model Interface for Dynamic Plant-Wide Simulation." *Water Research* 43: 1913–1923. <https://doi.org/10.1016/j.watres.2009.01.012>.
- Nyitrai, J., X. F. Almansa, K. Zhu, et al. 2023. "Environmental Life Cycle Assessment of Treatment and Management Strategies for Food Waste and Sewage Sludge." *Water Research* 240: 120078. <https://doi.org/10.1016/j.watres.2023.120078>.
- Palatsi, J., J. Illa, F. X. Prenafeta-Boldú, et al. 2010. "Long-Chain Fatty Acids Inhibition and Adaptation Process in Anaerobic Thermophilic Digestion: Batch Tests, Microbial Community Structure and Mathematical Modelling." *Bioresource Technology* 101: 2243–2251. <https://doi.org/10.1016/j.biortech.2009.11.069>.
- Palatsi, J., M. Laurenzi, M. V. Andrés, X. Flotats, H. B. Nielsen, and I. Angelidaki. 2009. "Strategies for Recovering Inhibition Caused by Long Chain Fatty Acids on Anaerobic Thermophilic Biogas Reactors." *Bioresource Technology* 100, no. 20: 4588–4596. <https://doi.org/10.1016/j.biortech.2009.04.046>.
- Parkin, G. F., and W. F. Owen. 1986. "Fundamentals of Anaerobic Digestion of Wastewater Sludges." *Journal of Environmental Engineering* 112: 867–920. [https://doi.org/10.1061/\(asce\)0733-9372\(1986\)112:5\(867\)](https://doi.org/10.1061/(asce)0733-9372(1986)112:5(867)).
- Puyol, D., X. Flores-Alsina, Y. Segura, et al. 2018. "Exploring the Effects of ZVI Addition on Resource Recovery in the Anaerobic Digestion Process." *Chemical Engineering Journal* 335: 703–711. <https://doi.org/10.1016/j.cej.2017.11.029>.
- Rico, C., N. Muñoz, J. Fernández, and J. L. Rico. 2015. "High-Load Anaerobic Co-Digestion of Cheese Whey and Liquid Fraction of Dairy Manure in a One-Stage Uasb Process: Limits in Co-Substrates Ratio and Organic Loading Rate." *Chemical Engineering Journal* 262: 794–802. <https://doi.org/10.1016/j.cej.2014.10.050>.
- Rosen, C., D. Vrecko, K. V. Gernaey, M. N. Pons, and U. Jeppsson. 2006. "Implementing ADM1 for Plant-Wide Benchmark Simulations in Matlab/Simulink." *Water Science Technology* 54, no. 4: 11–19. <https://doi.org/10.2166/wst.2006.521>.
- Rudi, J., A. C. I. Malossi, and T. Isaac, et al. 2015. "An Extreme-Scale Implicit Solver for Complex PDEs: Highly Heterogeneous Flow in Earth's Mantle." In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '15)*. New York, NY, USA: Association for Computing Machinery, 1–12. <https://doi.org/10.1145/2807591.2807675>.
- Silvestre, G., J. Illa, B. Fernández, and A. Bonmatí. 2014. "Thermophilic Anaerobic Co-Digestion of Sewage Sludge With Grease Waste: Effect of Long Chain Fatty Acids in the Methane Yield and its Dewatering Properties." *Applied Energy* 117: 87–94. <https://doi.org/10.1016/j.apenergy.2013.11.075>.
- Smith, A. L., L. B. Stadler, N. G. Love, S. J. Skerlos, and L. Raskin. 2012. "Perspectives on Anaerobic Membrane Bioreactor Treatment of Domestic Wastewater: A Critical Review." *Bioresource Technology* 122: 149–159. <https://doi.org/10.1016/j.biortech.2012.04.055>.
- Spijker, M. N. 1996. "Stiffness in Numerical Initial-Value Problems." *Journal of Computational and Applied Mathematics* 72: 393–406. [https://doi.org/10.1016/0377-0427\(96\)00009-X](https://doi.org/10.1016/0377-0427(96)00009-X).
- Stroustrup, B. 2013. *The C++ Programming Language*, 4th Edition. Addison-Wesley Professional.
- Xiao, K. K., C. H. Guo, Y. Zhou, Y. Maspolim, J. Y. Wang, and W. J. Ng. 2013. "Acetic Acid Inhibition on Methanogens in a Two-Phase Anaerobic Process." *Biochemical Engineering Journal* 75: 1–7. <https://doi.org/10.1016/j.bej.2013.03.011>.

Xing, J., C. Criddle, and R. Hickey. 1997. "Effects of a Long-Term Periodic Substrate Perturbation on an Anaerobic Community." *Water Research* 31: 2195–2204.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.