INL RPT-23-74749
Revision 0

FORCE Integration with DRAFT and IDAES

September 2023

Gabriel J. Soto
Botros Hanna
Paul Talbot
Idaho National Laboratory

Integrated Energy Systems

INL RPT-23-74749
Revision 0

FORCE Integration with DRAFT and IDAES

Gabriel J. Soto
Botros Hanna
Paul Talbot

September 2023

Idaho National Laboratory
Integrated Energy Systems
Idaho Falls, Idaho 83415

http://www.ies.inl.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

EXECUTIVE SUMMARY

Integrated energy systems (IES) combine power from variable, renewable energy sources and nuclear power plants (NPP) in mutually beneficial ways. Under certain market conditions, this union produces multiple commodities and improves economic viability. Technical and economic analysis of IES requires simulation of complex processes with software models having enough fidelity to capture real-world dynamics using reasonable computing resources. The open-source Framework for Optimization of Resources and Economics (FORCE) tool suite, developed at Idaho National Laboratory (INL), has enabled comprehensive modeling and simulation of IES. The capabilities within FORCE include grid portfolio optimization through the Holistic Energy Resource Optimization Network (HERON) and the transient process model analysis library HYBRID, among others.

Recent developments for the FORCE toolset have centered on strengthening the flexibility and modularity to link with external models and other available software to enhance IES simulations. Adding versatility to the FORCE toolset improves capability, resulting in higher fidelity techno-economic simulation of nuclear and IES components, including their expected performance after deployment. It also helps leverage existing work in the IES field and improve efficiency in national code development. This report focuses on two such endeavors: integration of the Dynamic Reliability Analysis Framework Tool (DRAFT) and the Design Integration and Synthesis Platform to Advance Tightly Coupled Hybrid Energy Systems (DISPATCHES) software packages into FORCE.

DRAFT, development of which is led by Oak Ridge National Laboratory (ORNL), is used to estimate the reliability of the IES components using probabilistic models to compute the probability of a component's failure over the component lifetime and will be integrated with HERON and HYBRID. DRAFT calculates the reliability—as a function of component operation (considering magnitude and frequency of ramping events and how often these near the operation limits)—and outputs the mean time to failure (MTTF) of each component. Integrating DRAFT with HERON is useful because HERON requires information about the components' lifetime and maintenance. Integrating DRAFT and HERON is expected to lead to a more accurate estimate of the optimized size of each component and optimized component dispatch informed by variable operational maintenance costs. Two features have been implemented in HERON to help the user understand how the IES economic metrics change after accounting for the components' reliability. These two features are:

(A) The time-dependent breakdown of the cashflow types for each component. This feature helps the analysts determine the economic impact of each unit in an IES system and estimate the cashflow associated with each IES component.

(B) The breakdown of the component resources (such as electricity production or steam consumption). This capability helps the analysts estimate metrics such as the total mass of the produced hydrogen or carbon dioxide or the total production of electricity. All these metrics are essential to evaluate the capability of the IES of interest (besides the net present values (NPV) metric which is already calculated by HERON).

DISPATCHES, development of which is led by the National Energy Technology Laboratory, provides algebraic models for IES components for capacity and dispatch optimization under uncertain market conditions and is being implemented as a new workflow within HERON. DISPATCHES offers algebraic models written in the Institute for the Design of Advanced Energy Systems (IDAES) framework as depicted in Figure 1.

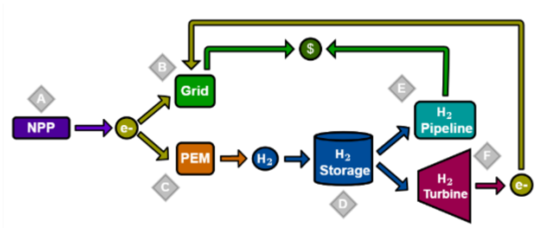

Figure 1: Example diagram of a DISPATCHES model for nuclear-hydrogen production.

There, a nuclear power plant is shown to interface with a proton-exchange membrane (PEM) for electrolysis, production of hydrogen, storage of hydrogen, and later, sales to a hydrogen pipeline or combustion to produce extra electricity as an arbitrage capability. Similar models as these are simulated within HERON, but DISPATCHES and IDAES provide capabilities to conduct the bi-level optimization of capacity and dispatch decisions, respectively, within a single level of optimization (for problems of limited size dependent on computational resources). DISPATCHES integration also leverages more accurate and higher fidelity component models for IES to offer a middle ground between the linear resource transfer models in HERON versus the transient models of HYBRID and Modelica.

Developments for FORCE integration within fiscal year (FY) 2023 have resulted in an improved workflow for HERON which allows a user to select between the current RAVEN-runs-RAVEN paradigm of bi-level optimization and the usage of bespoke IDAES models for a single level optimization solution.

Page intentionally left blank

CONTENTS

EXECUTIVE SUMMARY iii

ACRONYMS ix

1. Background 1

2. Dynamic Reliability Analysis Framework Tool (DRAFT) Integration 1

   2.1 Cashflows breakdown 3

   2.2 Component's activity 6

3. IDAES Integration 7

   3.1 DISPATCHES 8

      3.1.1 Workflows and Case Studies 8

      3.1.2 Stochastic Optimization within HERON versus DISPATCHES 11

      3.1.3 Previous FORCE-DISPATCHES Integration Efforts 12

   3.2 Implementations within HERON through HERD 12

      3.2.1 New Feature: Static Histories 14

      3.2.2 Improvement to Flowsheet Implementation 15

   3.3 Future Roadmap 16

4. Conclusion 18

5. References 18

FIGURES

Figure 1: Example diagram of a DISPATCHES model for nuclear-hydrogen production. iv

Figure 2. The DRAFT code output. The cumulative distribution function of the component probability of failure and the component mean time to failure (MTTF). 3

Figure 3. A diagram of an IES System Configuration. 4

Figure 4. The cashflows of each component in the IES and the net cashflow per year. 5

Figure 5. Cumulative cashflows (inflows and outflows) over the entire project lifetime. 5

Figure 6. The total activity (production) over the project lifetime from the NPP, the steam turbine, wind, and HTSE. 6

Figure 7. The total activity (production) over the project lifetime from the HTSE, the hydrogen storage unit. 7

Figure 8: Diagram of DISPATCHES workflows, usage of flowsheets and corresponding unit models. 9

Figure 9: Diagram of DISPATCHES nuclear case components and the exchange of resources within the model. 10

Figure 10: Diagram of the DISPATCHES renewable energy case and the exchange of resources within the model. 10

Figure 11: Outer-inner optimization problem as implemented within HERON via a RAVEN-runs-RAVEN algorithm. 12

Figure 12: Simple overview of HERD workflow including IDAES Multiperiod class construction and TEAL metric creation. 13

Figure 13: Schematic of IDAES Multiperiod constructor and subsequent stochastic build using intermediate flowsheet block method in HERD. 15

Figure 14: Overview of current and future HERD, DISPATCHES, IDAES and Pyomo workflow structure. 17

Figure 15: Possible future configurations of IES systems using DISPATCHES algebraic models. 17

TABLES

Table 1: An example of input data to the DRAFT ORNL code: Component operational data. 2

Table 2: Comparison results for nuclear case study simulations between DISPATCHES and HERD using multiple scenarios. 14

Page intentionally left blank

ACRONYMS

CAPEX	capital expenditures
DISPATCHES	Design Integration and Synthesis Platform to Advance Tightly Coupled Hybrid Energy Systems
DRAFT	Dynamic Reliability Analysis Framework Tool
FOM	fixed operation and maintenance
FORCE	Framework for Optimization of Resources and Economics
FY	fiscal year
HERON	Holistic Energy Resource Optimization Network
IDAES	Institute for the Design of Advanced Energy Systems
IES	integrated energy systems
INL	Idaho National Laboratory
LMP	locational marginal price
MTTF	Mean Time To Failure
NPP	nuclear power plant
NREL	National Renewable Energy Laboratory
ORNL	Oak Ridge National Laboratory
PEM	proton-exchange membrane
RAVEN	Risk Analysis Virtual Environment
ROM	reduced-order model
VOM	variable operation and maintenance
VRE	variable renewable energy
ARMA	autoregressive moving-average
TEAL	Tool for Economic Analysis
MOPED	Monolithic Optimizer for Probabilistic Economic Dispatch
HERD	HERON Runs DISPATCHES
CSV	comma-separated value
IPOPT	interior point optimizer

Page intentionally left blank

1. Background

Technical and economic analysis (TEA) of integrated energy systems (IES) requires modeling of complex behavior and dynamics of markets, power generators, and other commodity production under uncertain present and future conditions. The focus of IES is to cleanly and reliably meet power demands while improving the economic performance of both predictable nuclear power plants (NPPs) and unpredictable variable renewable energy (VRE) sources. Idaho National Laboratory (INL) develops its own open-source software for the techno-economic modeling and simulation of such systems: the Framework for Optimization of Resources and Economics (FORCE). Capabilities within FORCE include stochastic optimization and parametric sweeps of component capacities within a grid portfolio under unique realizations or scenarios of a market within the plugin Holistic Energy Resource Optimization Network or HERON. FORCE also provides transient process model analysis through the HYBRID library.

Among the many strengths of FORCE is its modularity and capability for coupling with other codes as external models. For such a complex, stochastic problem as IES techno-economic performance optimization, users should be presented with multiple tools for solving it. Code-coupling improves the versatility of FORCE, allowing the user to choose a level of fidelity for which to conduct TEA to best suit their needs. Code-coupling also improves the national development of IES modeling software to better reach clean-energy standards in the United States and beyond. Many pieces of software, developed by academia or industry, exist in isolation where they are better used in concert with other existing software.

Two main developments within FORCE are highlighted within this report for fiscal year (FY)-23. One is the integration of FORCE with the Dynamic Reliability Analysis Framework Tool (DRAFT). Integrating DRAFT and HERON is expected to lead to a more accurate estimate of the optimized size of each component and optimized component dispatch informed by variable operational maintenance costs. Two features have been implemented in HERON to help the user understand how the IES economic metrics change after accounting for the components' reliability. These two features are: the time-dependent breakdown of the cashflow types for each component and the breakdown of the component resources (such as electricity production or steam consumption).

The second development involves integration of FORCE with the Design Integration and Synthesis Platform to Advance Tightly Coupled Hybrid Energy Systems (DISPATCHES). DISPATCHES uses algebraic models which are often higher fidelity than the linear component models within HERON (though not as high fidelity as those in the HYBRID library) to conduct TEA. While HERON can model any system with simple transfer functions, DISPATCHES is dependent on a limited library of publicly available models within their repository; users can submit or create unique models as needed. For those models that are available, DISPATCHES also offers a different optimization strategy than the standard HERON workflow: a single level solve as opposed to HERON's bi-level solve of capacities and dispatch decisions. HERON has a separate, optional single level workflow but it still requires some additional development before a full release. For FY-23, new developments of HERON can now leverage the DISPATCHES optimization strategy using a new class called HERON Runs DISPATCHES (HERD) and call available algebraic models. HERD amplifies the solution tools available to users and strengthens the capabilities of FORCE.

2. Dynamic Reliability Analysis Framework Tool (DRAFT) Integration

The DRAFT is a code developed by Oak Ridge National Laboratory (ORNL) to estimate the reliability of the IES components using probabilistic models to compute the probability of a component's failure over the component lifetime. The reliability of each component is also a function of operation, such as the number of ramping events and how often each component operates near the operation limits. An accurate estimation of the reliability of each IES component enhances the accuracy of the components' maintenance cost estimate and the estimated net present value (NPV) of any IES configuration.

The objective of this work is demonstrating a nominal level of integration of the ORNL reliability module (DRAFT) into the FORCE toolset. A demonstration case is presented to formalize this activity. Related work has been completed at ORNL, and this report focuses on supporting their efforts. Integration work includes dynamically optimizing component rebuild/maintenance scheduling and dispatch optimization informed by component wear. The DRAFT code is still under development, and it is not released publicly yet but is expected that integrating DRAFT with the FORCE tools will enhance the IES techno-economic analysis as follows.

1. The inputs to the DRAFT code are the components' operational data (such as the data in Table 1). It is challenging to obtain these inputs (e.g., number of samples, failure/suspension cycle counts) from real-world data. Instead, we can run transient Modelica simulations using the INL FORCE tool, HYBRID [13].
2. The output of DRAFT is a cumulative distribution function of the component probability of failure such as the distribution function in Figure 2. The DRAFT code also calculates and returns the component mean life or the mean time to failure (MTTF). Integrating DRAFT with the FORCE tool HERON [14] is useful because HERON requires information about the components' lifetime and maintenance. Integrating DRAFT and HERON is expected to lead to a more accurate estimate of the optimized size of each component and optimized component dispatch informed by variable operational maintenance costs. Two features have been implemented in HERON to help the user understand how the IES economic metrics change after accounting for the components' reliability. These two features are:
    a. The time-dependent breakdown of the cashflow types for each component.
    b. The breakdown of the component resources (such as electricity production or steam consumption).

Table 1: An example of input data to the DRAFT ORNL code: Component operational data.

The component failure time (hours)	Was the test stopped or the failed component replaced/removed?
252	FALSE
280	FALSE
320	FALSE
328	TRUE
335	FALSE
354	FALSE
361	FALSE
362	FALSE
368	FALSE
375	TRUE
375	TRUE
375	TRUE

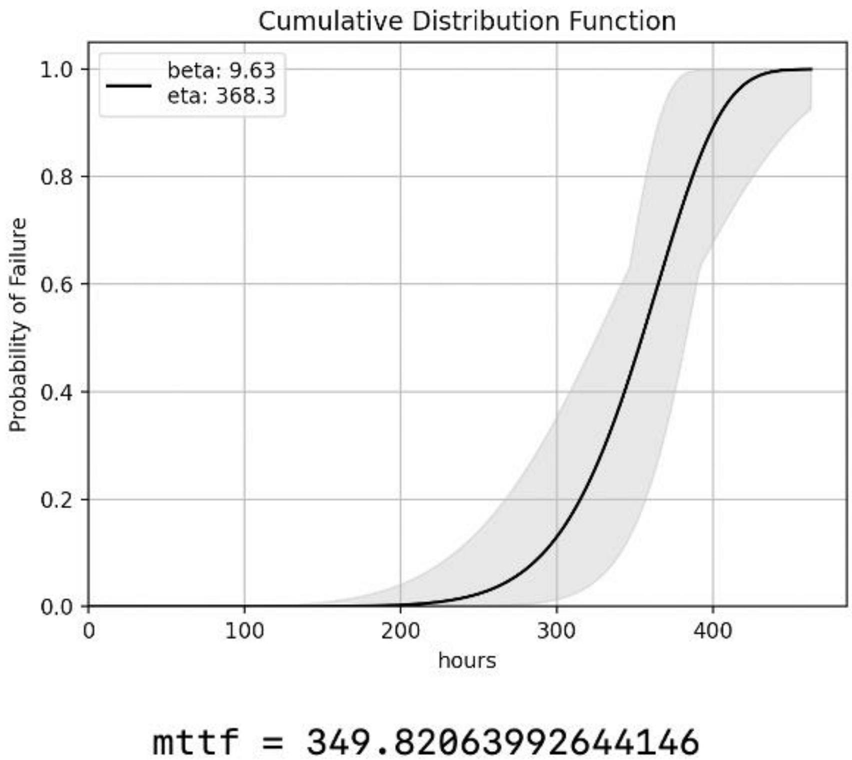

Figure 2. The DRAFT code output. The cumulative distribution function of the component probability of failure and the component mean time to failure (MTTF).

2.1 Cashflows breakdown

To analyze the economic impact of each unit in an IES system, it is necessary to estimate the cashflow associated with each IES component. In this report, the new HERON feature (cashflow breakdown plotting) that has been implemented in HERON, is demonstrated by plotting the breakdown of the cashflow of the IES system depicted in Figure 3.

The example, in Figure 3 shows a full IES system including an NPP, wind and steam turbines, a high-temperature steam electrolysis (HTSE) unit, a hydrogen storage unit, the markets for electricity and hydrogen.

The cashflow associated with each unit are plotted in Figure 4 and Figure 5. While the cashflows are negative in the beginning of the project, they eventually turn positive. In this example, the positive cashflow is mainly due to selling the electricity. The main cost driver is the capital expenditure (CAPEX) cashflow of the NPP (see Figure 5). This new HERON capability helps the analysts to determine which unit dispatch is likely to be profitable.

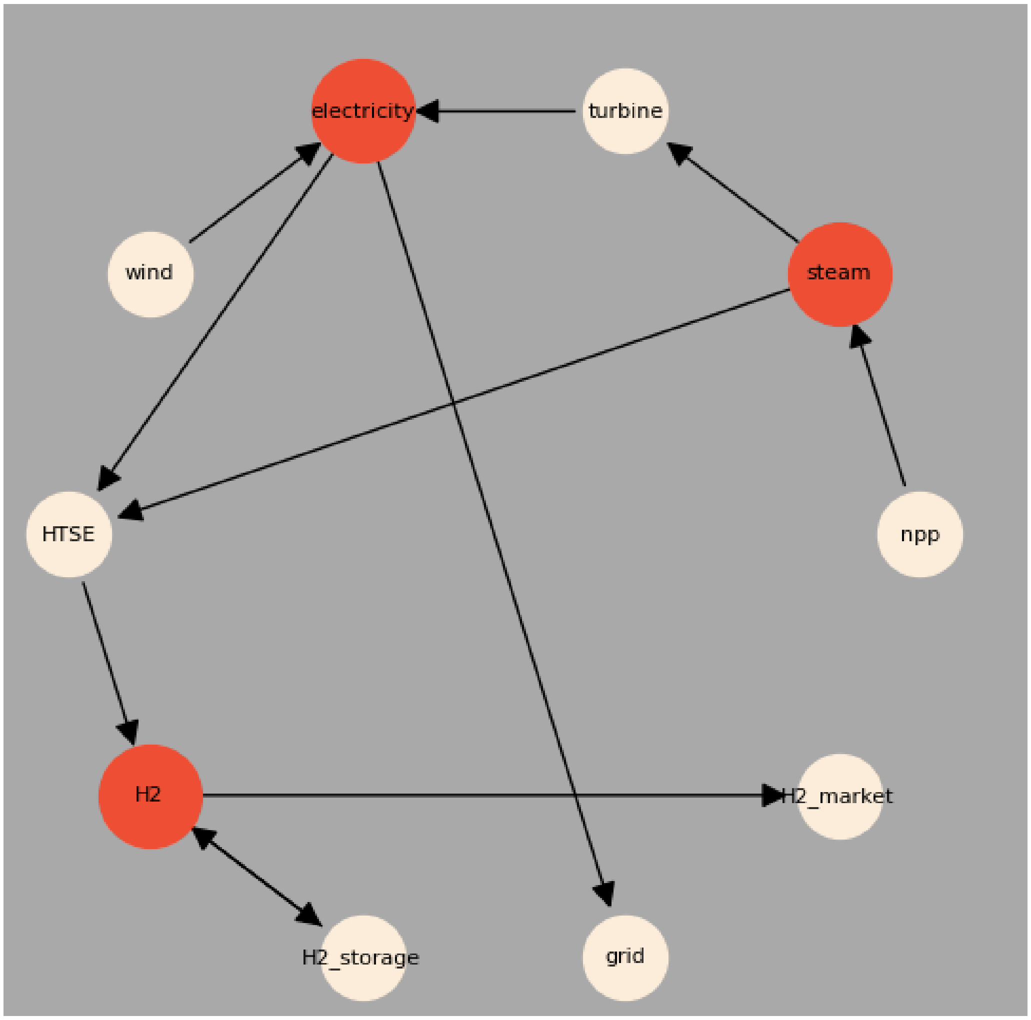

Figure 3. A diagram of an IES System Configuration.

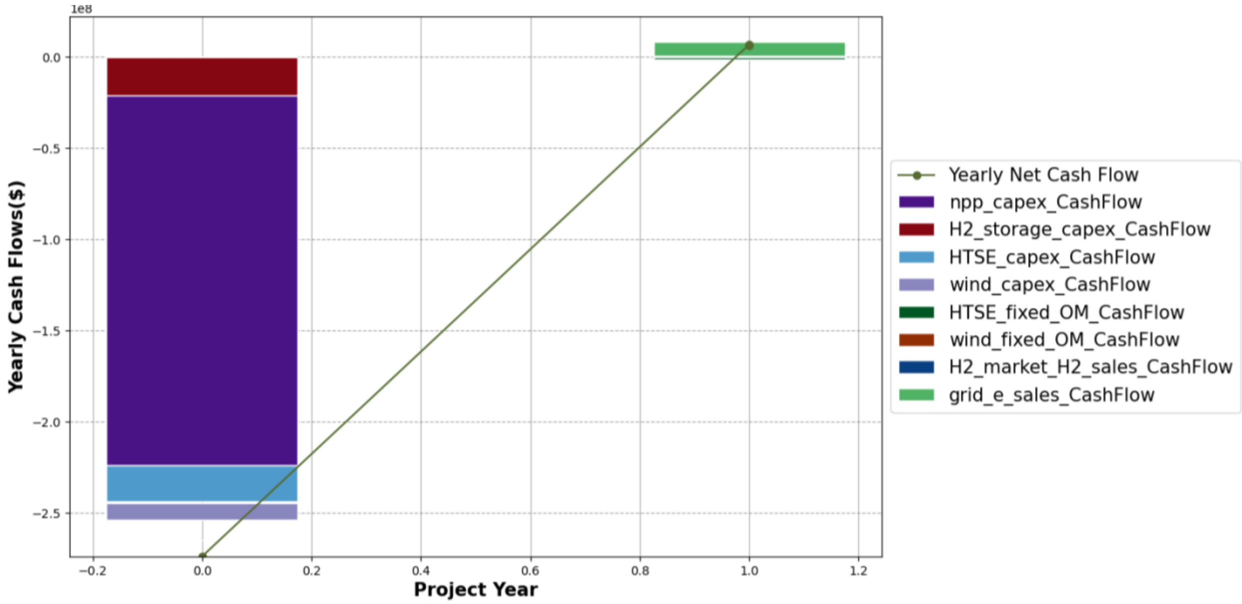

Figure 4. The cashflows of each component in the IES and the net cashflow per year.

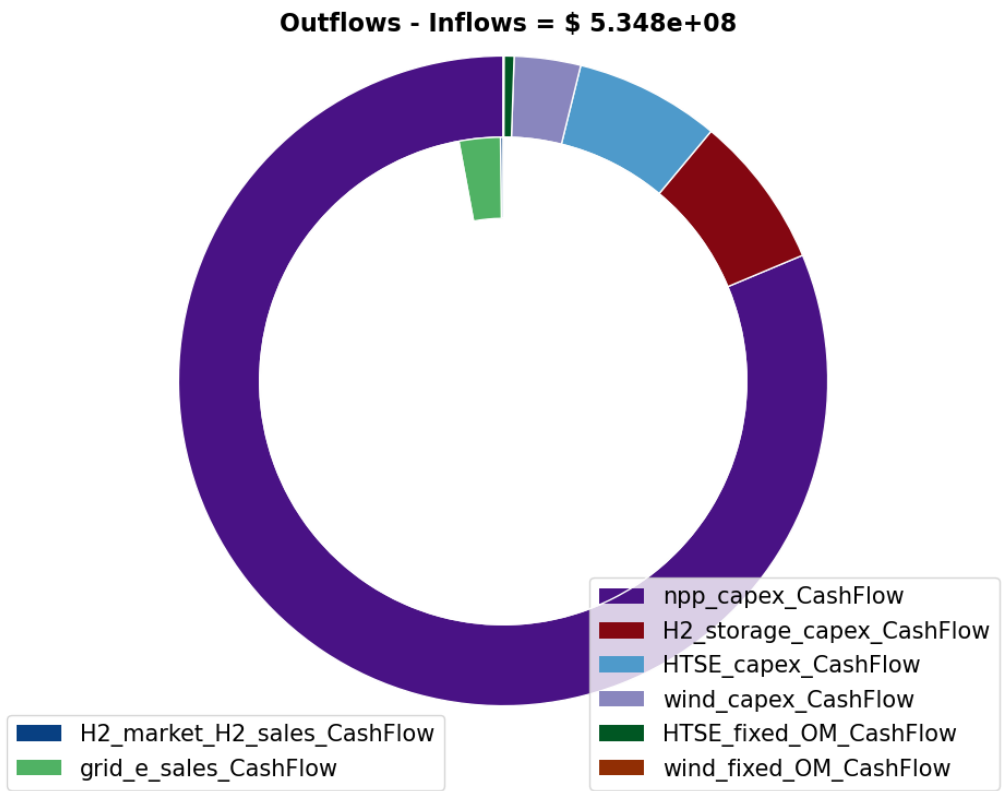

Figure 5. Cumulative cashflows (inflows and outflows) over the entire project lifetime.

2.2 Component's activity

A new feature is implemented to calculate the total activity of each component in the IES system as follows.

If the component activity is $A(c,t)$ where the activity is a component production or consumption (e.g., how many kilograms of hydrogen are produced per hour or how many Megawatts are produced from a nuclear reactor). $A(c,t)$ changes over time, *t* and computer for cluster, *c*. A cluster is a dataset for which a reduced-order model (ROM) is built. For example, if the signal (such as price, demand, weather), changed between summer and winter, the signal is them divided and a distinct ROM is trained on each cluster.

npp capacity	wind capacity	HTSE capacity	H2 storage capacity	med_TotalActivity__npp__production__steam	med_TotalActivity__turbine__production__steam	med_TotalActivity__turbine__production__electricity	med_TotalActivity__wind__production__electricity	med_TotalActivity__HTSE__production__H2	med_TotalActivity__HTSE__production__steam
25	5	10	0	627899	-626438	206725	61495	251850	-1461
25	10	10	0	599114	-597654	197226	123201	251850	-1461
50	5	10	0	1060548	-1059087	349499	61340	251850	-1461
50	10	10	0	898272	-896811	295948	122770	251850	-1461
25	5	15	0	628037	-626577	206770	61451	251850	-1461
25	10	15	0	600829	-599368	197791	122932	251850	-1461
50	5	15	0	1060271	-1058810	349407	61189	251850	-1461
50	10	15	0	902677	-901216	297401	122664	251850	-1461
25	5	10	25	627680	-626219	206652	61458	251850	-1461
25	10	10	25	598908	-597447	197158	122437	251850	-1461
50	5	10	25	1150716	-1149255	379254	31898	251850	-1461
50	10	10	25	1076735	-1075274	354840	64613	251850	-1461
25	5	15	25	628134	-626673	206802	61263	251850	-1461
25	10	15	25	602231	-600768	198253	122850	252240	-1463
50	5	15	25	1062492	-1061022	350137	61517	252380	-1464
50	10	15	25	903102	-901632	297539	122274	252372	-1464
25	5	10	50	627650	-626190	206643	61182	251850	-1461
25	10	10	50	600760	-599299	197769	122476	251850	-1461
50	5	10	50	1148548	-1147087	378539	33702	251850	-1461
50	10	10	50	1074379	-1072918	354063	64710	251850	-1461
25	5	15	50	628274	-626797	206843	61235	254588	-1477
25	10	15	50	602905	-601440	198475	122532	253417	-1470
50	5	15	50	1066879	-1065399	351582	61214	253658	-1471
50	10	15	50	901760	-900287	297095	122770	255125	-1480

Figure 6. The total activity (production) over the project lifetime from the NPP, the steam turbine, wind, and HTSE.

The cumulative activity per cluster is equals $\int A(c,t)\,dt$. If $A(c,t)$ is the power (in MW), the $\int A(c,t)\,dt$ can be the total energy produced per cluster (in MWh). The cumulative activity per cluster is multiplied by the cluster multiplicity, M_c, which is the number of clusters represented by this cluster within a year. Therefore, the total activity of the project over the project lifetime is:

$$\textit{The component's total activity} = \sum_{y=0}^{n} \left[\sum_{c=0}^{h} \left(\int A(c,t)\,dt \right) \times M_c \right]$$

where *h* is the number of clusters, *c*, and n is the number of years, *y* over the project lifetime.

npp capacity	wind capacity	HTSE capacity	H2 storage capacity	med_TotalActivity__HTSE__production__electricity	med_TotalActivity__H2_storage__level__H2	med_TotalActivity__H2_storage__charge__H2	med_TotalActivity__H2_storage__discharge__H2	med_TotalActivity__grid__production__electricity	med_TotalActivity__H2__market__production__H2
25	5	10	0	-8593	0	0	0	-259542	-251850
25	10	10	0	-8593	0	0	0	-311869	-251850
50	5	10	0	-8593	0	0	0	-402490	-251850
50	10	10	0	-8593	0	0	0	-410317	-251850
25	5	15	0	-8593	0	0	0	-259545	-251850
25	10	15	0	-8593	0	0	0	-311982	-251850
50	5	15	0	-8593	0	0	0	-401763	-251850
50	10	15	0	-8593	0	0	0	-411159	-251850
25	5	10	25	-8593	125925	0	0	-259352	-251850
25	10	10	25	-8593	125925	0	0	-310731	-251850
50	5	10	25	-8593	125925	0	0	-401660	-251850
50	10	10	25	-8593	125925	0	0	-410559	-251850
25	5	15	25	-8593	6388	-5468	5362	-259641	-251850
25	10	15	25	-8606	48623	-24469	23332	-312803	-251850
50	5	15	25	-8611	98147	-35832	34869	-403396	-251850
50	10	15	25	-8610	79764	-29074	29848	-410850	-251850
25	5	10	50	-8593	251850	0	0	-259302	-251850
25	10	10	50	-8593	251850	0	0	-311606	-251850
50	5	10	50	-8593	251850	0	0	-402191	-251850
50	10	10	50	-8593	251850	0	0	-411236	-251850
25	5	15	50	-8686	14360	-11372	8634	-259314	-251850
25	10	15	50	-8646	69383	-29582	28635	-312559	-251850
50	5	15	50	-8654	158861	-43094	39851	-403806	-251850
50	10	15	50	-8704	154979	-39513	36151	-411384	-251850

Figure 7. The total activity (production) over the project lifetime from the HTSE, the hydrogen storage unit.

In HERON, the component's total activity is calculated per each component for different values of components' capacities as demonstrated in and Figure 8. and Figure 8 show the median total activity of each component. Other statistics of the total activity are also calculated (not shown in the figures) such as the mean, the minimum, the maximin, the variance, and the 95% confidence interval.

This capability helps the analysts to estimate the total mass of the produced hydrogen or carbon dioxide or the total production of electricity. All these metrics are essential to evaluate the capability of the IES of interest besides the net present values (NPV) metric which is already calculated by HERON.

3. IDAES Integration

Integrated energy systems (IES) produce energy from various sources—VREs, NPPs, secondary storage—and use them to address demands for electricity and other commodities. IES leverage the baseload reliability of NPP-produced electricity and the peaking, daily, and seasonal intermittency of variable renewable energy to cleanly and reliably meet power demands from grid operators. Residual power after meeting power demands can be rerouted for storage, secondary commodity production, residential heating, or other industrial applications that can be facilitated with excess energy. Storage of either electricity, thermal energy, or other commodities from IES can also be used for arbitrage in their respective markets. Strategic charging and discharging of storage components can take advantage of fluctuations in energy prices for added revenue and profits.

Capacities of IES components and optimal dispatch strategies are dependent on many factors of uncertainty—market conditions can fluctuate based on the amount of participation and conditions of the grid, including the intermittent availability of VRE. INL has developed its open-source FORCE tool suite for the modeling and simulation of IES. The capabilities within FORCE include stochastic optimization of grid portfolios through the HERON and the transient process model analysis library HYBRID, among others. These are plugins for the more general a Risk Analysis Virtual Environment (RAVEN) software that can conduct stochastic gradient descent and other uncertainty quantification tools. Among HERON's many strengths is its modularity and ability to be coupled with external codes as well as many internally developed plugins for economic analysis; its creation and sampling of reduced-order model (ROMs) representations of physics-based transient models and economic or weather signals; and its post-processing and uncertainty analysis of stochastic processes. Enhancement of HERON's capabilities through additional coupling to other external software is a matter of continuing development at INL, and is the focus of this report, particularly regarding stochastic optimization methods. This report is a continuation of a FY-22 endeavor for a FORCE-DISPATCHES Integration Initial Demonstration [1].

3.1 DISPATCHES

DISPATCHES is an open-source Python software package developed to optimize IES for operation within power grid systems via energy market signals [2]. DISPATCHES was primarily developed by the National Energy Technology Laboratory, with contributions from INL, Lawrence Berkeley National Laboratory, the National Renewable Energy Laboratory (NREL), and Sandia National Laboratory.

3.1.1 Workflows and Case Studies

The DISPATCHES software package hosts a multitude of models and workflows for simulation, some of which are summarized in Figure 8. Two general assumptions can be made about the participation of an IES in a deregulated commodity market: it is either a price-taker or a price-maker. The former is the simplest to simulate, being that the commodity prices (e.g., electricity) are set independently from the actions of the system; the IES decides on dispatch strategies based on pre-determined market prices. The latter involves more careful simulation of the system external to the IES – actions of the IES influence the price of the commodity. DISPATCHES offers two strategies for each of these market types: a *multiperiod price-taker* workflow for the price-taker type and a *double market loop optimization* workflow for the price-maker market type. These workflows are generally conducted within Jupyter notebooks found in the DISPATCHES repository [3]. Here we focus on the price-taker model because it most closely resembles FORCE (specifically HERON) default capabilities.

Once a workflow strategy has been selected, one can simulate an IES configuration. These configurations are described by the user via flowsheets: Python scripts that house all components, connectors, and constraints necessary to describe the IES system at any point in time. These flowsheets combine algebraic physics models of the intended components. The most basic of these algebraic models are provided by the Institute for the Design of Advanced Energy Systems (IDAES) platform and include such things as resource ports, turbines, and compressors. DISPATCHES provides more complex models that combine basic IDAES algebraic models for such things as a hydrogen storage tank, concrete thermal energy storage, and heat exchanger tubes. A non-comprehensive list of DISPATCHES unit models is shown on the right of Figure 8. Not shown in that figure is another directory for property packages which include thermodynamic and fluid properties of various materials like Hitec salts or ideal gases [4,5].

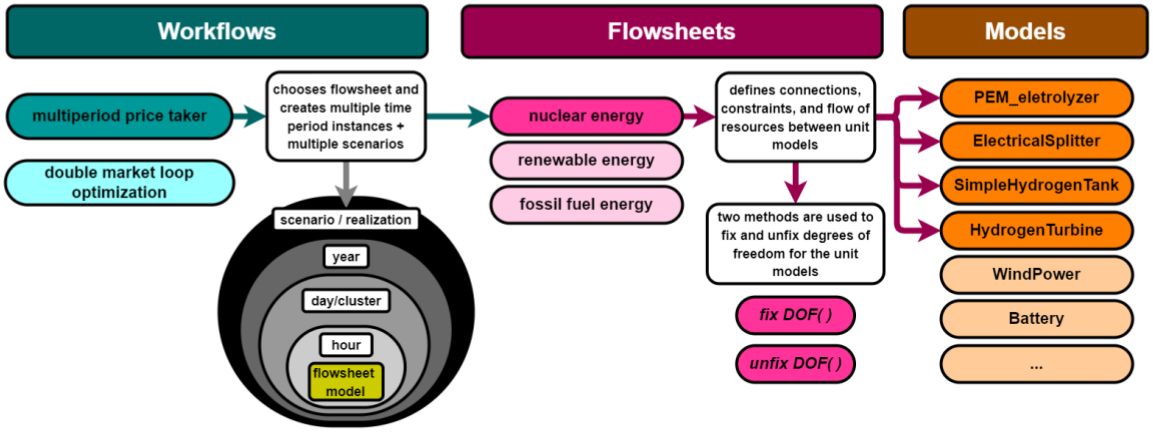

Figure 8: Diagram of DISPATCHES workflows, usage of flowsheets and corresponding unit models.

Using an established flowsheet, the multiperiod price-taker workflow creates a multiperiod model for all intended time period instances. DISPATCHES uses Pyomo sets to facilitate this process: a set of hours defines a day, a set of days defines a year, a set of years defines a specific scenario for the simulation as shown on the left of Figure 8 [6,7]. These multiperiod models are IDAES classes that can also create linkages between unit models from one period to the next using bespoke methods supplied by the user. This ensures that parameters such as tank storage levels match between time periods. The user can additionally fix or unfix certain degrees of freedom within the configuration. Typically, a user might fix most or all degrees of freedom to ensure that the model converges on a plant-balanced solution or to at least ensure that one is possible before running full simulations.

Three sample flowsheets are provided within the DISPATCHES repository based on distinct case studies: a nuclear, renewable energy, and fossil fuel case study flowsheet. This report focuses on implementation of the nuclear and renewable energy case studies. The IES nuclear case solved by the DISPATCHES software uses a fixed-output NPP and a split electricity flow from its outlet. The electricity is divided into two streams: one to the grid for direct sales, and the other to a proton-exchange membrane (PEM) electrolyzer that converts the electricity into hydrogen [8]. The hydrogen is stored in a hydrogen tank from which it can be sold for profit to a hydrogen market. Alternatively, hydrogen from the tank can be sent through a hydrogen turbine to be converted into electricity during opportune times, via combustion [9]. A flow diagram of this IES configuration is shown in Figure 9. Within the IDAES platform are Pyomo algebraic models for the NPP, PEM electrolyzer, hydrogen tank, and hydrogen turbine. DISPATCHES links together all the resource streams via Pyomo connectors, arcs, splits, and property tables.

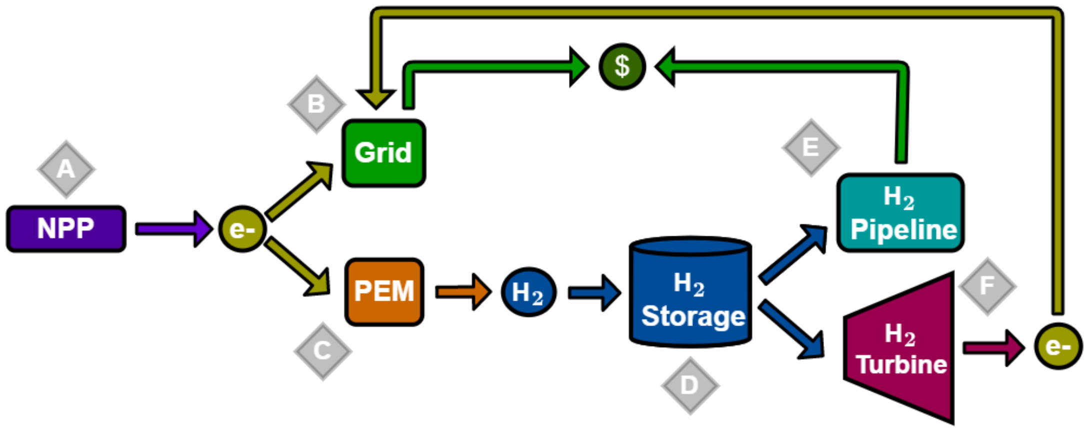

Figure 9: Diagram of DISPATCHES nuclear case components and the exchange of resources within the model.

The IES renewable energy case shown in Figure 10 replaces the NPP with a wind farm for electricity production. The wind farm reads wind speed profiles as a time series and uses the PySAM software package from NREL to calculate electricity generation and other properties of the wind component [10]. The electricity output from the wind component, similar to the nuclear case, can sell directly to the grid or divert a stream to a PEM which can generate hydrogen, store it and then either sell it directly to a hydrogen market or combust it for additional electricity. One more addition to the renewables case is an electric battery for storage of the electricity resource. This additional component can be used for arbitrage—store when prices are low, sell when prices are high.

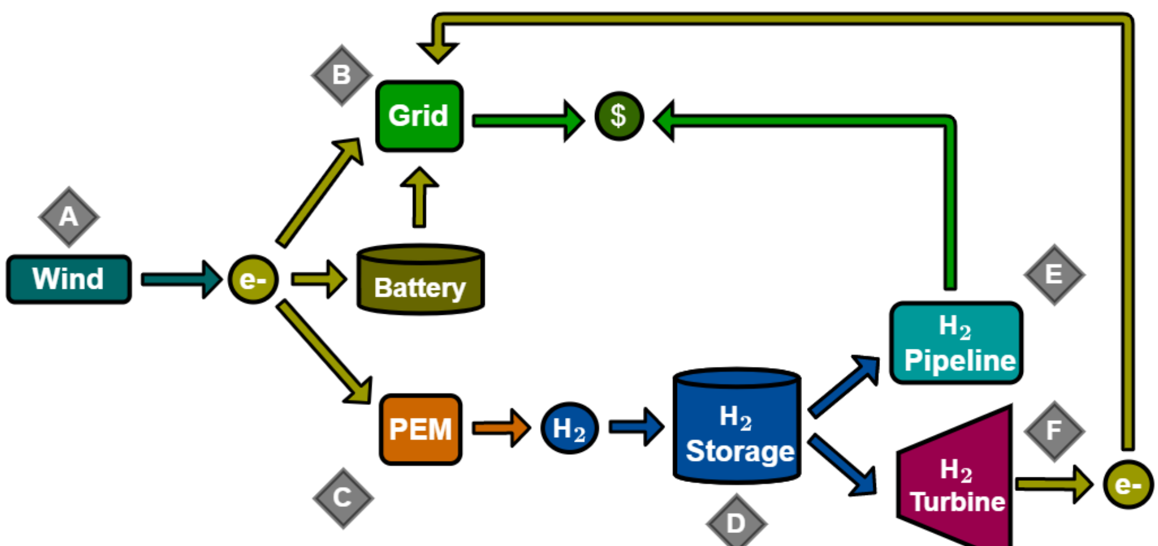

Figure 10: Diagram of the DISPATCHES renewable energy case and the exchange of resources within the model.

Regardless of flowsheet, the multiperiod price-taker simulations optimize the dispatch of resources between and out of components to maximize some metric. This metric is typically the net present value (NPV) of the system. Cost models are added for each component and include negative cash flows for capital expenditures (CAPEX), fixed operations and maintenance (FOM) and variable operation and maintenance costs (VOM) which are incurred once, yearly, and hourly, respectively. Positive cashflows are generated from electricity and commodity sales based on pricing data. Currently, the Jupyter notebooks can use a time series of locational marginal prices (LMP) which are settled historical prices from a deregulated market for a geographic location. LMP signals are given as hourly data points and are used as inputs to determine optimal hourly dispatch to maximize NPV.

3.1.2 Stochastic Optimization within HERON versus DISPATCHES

From a price-taker perspective, the complexities of the bidding of a multitude of independent agents, weather conditions on energy system performances (particularly on the intermittency of VREs), and contingency events in a geographic region introduce uncertainty to the market conditions and LMP signals. The FORCE toolset incorporates this uncertainty in the portfolio optimization problem using a tiered optimization strategy and usage of reduced-order models (ROMs) for time series. Specifically, RAVEN is used to train a synthetic history ROM such as an autoregressive moving average (ARMA) model on historic time series. RAVEN can then generate unique, synthetic histories from the trained ARMA model that are representative of the training data but retain its statistical properties. Through these synthetic histories, multiple simulations can be run on unique "realizations" of the intended market and weather conditions. These synthetic histories are typically time series such as LMP, grid demand or loading profiles, solar irradiance, and wind speed profiles. Optimization from these realizations can generate a distribution of optimal solutions from which additional post-processing steps can be applied.

Capacity optimization for a grid portfolio within FORCE is handled by HERON and previously has conducted its stochastic optimization via a two-stage algorithm commonly referred to as RAVEN-runs-RAVEN. HERON is now moving to provide a more modular approach to stochastic optimization for users, since different problems might be better solved with higher efficiency (with trade-offs in system precision and accuracy) by different solving approaches. The original RAVEN-runs-RAVEN algorithm consists of an outer optimization which explores the capacities of all proposed IES components while an inner optimization finds the optimal dispatch strategy for the given outer configuration. Each step in the outer optimization attempts to optimize the *expected value* of an economic metric (e.g., NPV) and so multiple inner optimizations are run for every outer step—these are unique realizations of the proposed IES portfolio performing under different synthetic histories sampled from a trained ARMA model. Figure 11 shows a diagram of this approach. The dispatch optimization within the inner steps is conducted for a set of hours, days, and years for each realization or sample. The set of days can be additionally simplified by implementing a clustering algorithm—RAVEN can train on all days of the year, then group together days that share characteristics. The dispatch optimization step can then conduct simulations just on the clusters and add back a multiplicity term for however many days each cluster is meant to represent. The economic metrics are calculated by cash flows generated in another RAVEN plugin: the Tool of Economic Analysis (TEAL). HERON is the tool used to write files for the outer and inner optimizations: each is actually a RAVEN input file and the outer RAVEN file calls the inner RAVEN files in a RAVEN-runs-RAVEN cadence. The inner RAVEN files call a dispatch manager within HERON as an external model for the actual dispatch optimization.

HERON now offers a second optimization approach through the Monolithic Optimizer for Probabilistic Economic Dispatch (MOPED). MOPED is a recent addition to HERON and offers an alternate workflow: it runs optimization via a single monolithic solve rather than the outer-inner multi-stage optimization implemented in the HERON-runs-RAVEN method [11]. This means that capacity variables and dispatch activity variables (for all time periods) are optimized within the same problem. MOPED conducts its optimization on automatically generated, linear Pyomo models. This new approach is best tailored to problems of limited size and project length dependent on available computational memory but can still provide users with flexibility on how to solve their IES optimization problems. HERON still retains the standard RAVEN-runs-RAVEN workflow for more complex cases.

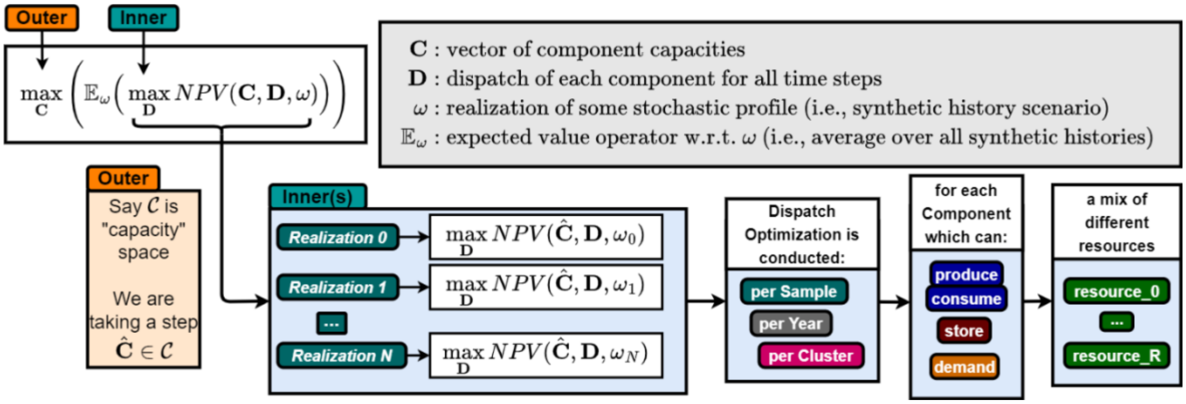

Figure 11: Outer-inner optimization problem as implemented within HERON via a RAVEN-runs-RAVEN algorithm.

Multiperiod optimization within the price-taker workflow in DISPATCHES can also conduct stochastic optimization by implementing multiple realizations of a given market and geographic region. The stochastic optimization, which has currently only been implemented for the nuclear case study, is similar to the MOPED approach in that it runs in a single stage as an "all-at-once" strategy. The optimization variables for this single stage are a mix of the component capacities as well as the time-indexed resource dispatch to the commodity markets and between components. In the base nuclear case Jupyter notebook, realizations of LMP signals are imported via a collection of synthetic histories pre-generated from an ARMA model. The LMPs are given hourly and in series for a 24-hour period. In the given signal data are two clusters of 24-hour time series per year and a total of 10 years of data per scenario, generated via RAVEN synthetic history training. A collection of five static sampled scenarios can be used within the nominal DISPATCHES workflow. The multiperiod notebook iteratively creates instances of the flowsheet for every 24-hour time series per cluster, cluster per year, year per scenario, and scenario per user specification. NPV is calculated manually within the notebook and used as the objective function of the overall optimization.

3.1.3 Previous FORCE-DISPATCHES Integration Efforts

In FY-22 and continuing into FY-23, there have been efforts in making FORCE tools available within the DISPATCHES package. RAVEN and TEAL are now installable as individual Python packages, available as optional dependencies within the DISPATCHES repository. A new nuclear case notebook has recently been added that can import RAVEN's external ROM model sampling methods to generate new synthetic histories from a trained ARMA model rather than cycling through the pre-generated synthetic LMP histories in the repository. A new interface script has also been implemented that can generate TEAL cashflows and metrics using Pyomo algebraic expressions from the DISPATCHES unit models. These efforts have been able to leverage the strengths of all toolsets in an external environment. A converse integration, however, is also of interest to users: to be able to import DISPATCHES algebraic models within the FORCE toolset as external models for dispatch optimization. This report details how HERON can use DISPATCHES as an IES evaluation approach and run flowsheets using an "all-at-once" optimization.

3.2 Implementations within HERON through HERD

Previous work from FY-22 resulted in a new class and workflow within HERON as an alternative to the outer-inner optimization approach [18]. HERD is a new class that interfaces between HERON inputs and creates an IDAES Multiperiod class based on a previously generated flowsheet as shown in Figure 12. HERD inherits the MOPED class architecture, which is an alternative workflow option to the standard

RAVEN-runs-RAVEN. MOPED provides the infrastructure to collect component information from the user-supplied HERON XML inputs. HERD now offers a third workflow option within HERON to also conduct a monolithic optimization strategy, but instead uses the DISPATCHES flowsheets and workflow with the IDAES Pyomo models, property tables, and other features within its platform. HERD borrows some MOPED methods to leverage existing architecture and adds new ones to facilitate gathering the necessary metadata for running the DISPATCHES workflow.

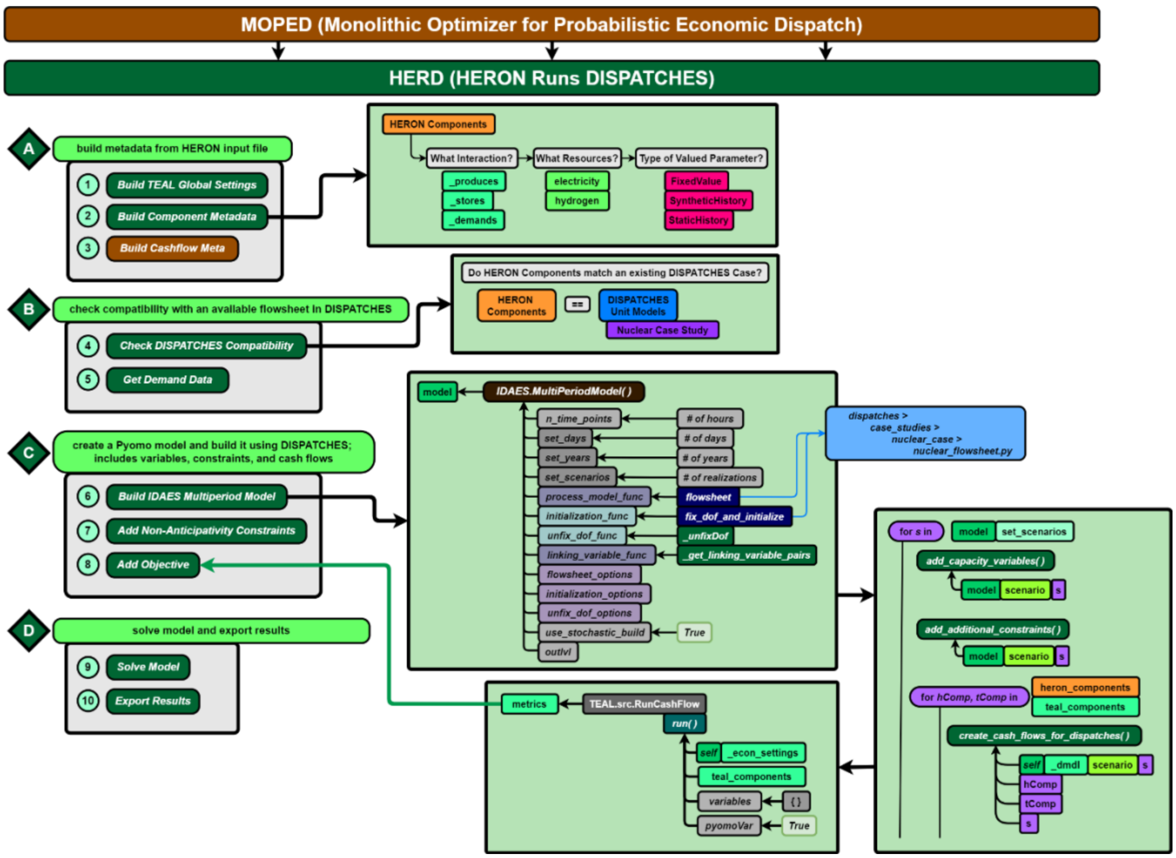

Figure 12: Simple overview of HERD workflow including IDAES Multiperiod class construction and TEAL metric creation.

Once a user selects the newest DISPATCHES workflow from a HERON input script, HERON reads in all case, component, and data generator information but, instead of writing an outer and inner XML file for RAVEN, it calls the HERD *run* method. Figure 12 shows a general outline of the HERD *run* method. The first portion (A) of the method parses through the HERON input metadata that has been previously collected to build the TEAL Global Settings object, a separate dictionary of HERON components (e.g., nuclear power plant, PEM, etc.) with their relevant actions and any potential data generators (e.g., ARMA model or static histories), and another dictionary of all component cashflow data. Here, potential ARMA models can be sampled to generate the required number of synthetic histories. The second portion (B) checks compatibility of the purported IES configuration with available DISPATCHES models. This method will be expanded in the future, as it only checks to see if a similar configuration to the nuclear case study in DISPATCHES has been selected. The third portion (C) builds the IDAES multiperiod object by linking relevant methods: a method to build a flowsheet, to initialize the model, to unfix degrees of freedom, to link variable pairs between time periods, and to add optional input parameters for any of these methods. Afterwards, capacity variables are added to the model using Pyomo classes and algebraic TEAL cashflows are created and used to calculate the final desired metric (e.g., NPV) for usage in the objective function. Finally, in portion (D) the model is solved using the interior point optimizer (IPOPT) solver and results are reported back to the user [12]. Some additional features and improvements have been implemented into HERD during FY-23.

3.2.1 New Feature: Static Histories

The nominal implementation of the nuclear case study within the DISPATCHES repository imports data signals from a JSON script containing five pre-generated samples from a trained ARMA model. Within HERON (as well as the newest RAVEN- and TEAL-integrated nuclear case DISPATCHES notebook), direct sampling from an ARMA model generates more interesting and comprehensive results by exploring new unique scenarios and can create less biased distributions of results. Moreover, users may want to try other ARMA models. However, it is still of use to import a static history or time series for testing purposes or just to generate quicker, preliminary results.

In FY-22, HERD was capable of sampling new synthetic histories from a given ARMA model. To test against previous results from the nominal DISPATCHES notebook, HERD imported directly from the JSON script in a method that was not open to the user. One new feature for HERD which facilitates the testing of the workflow is the ability to import and use static histories within the workflow. Currently, LMP signals are being imported via CSV file in the *DataGenerators* node. This new feature has streamlined testing for multiple realizations and has helped to fix some lingering bugs in the code. An enhanced comparison against the DISPATCHES notebook results has been conducted and shown in Table 2.

Table 2: Comparison results for nuclear case study simulations between DISPATCHES and HERD using multiple scenarios.

Number of Scenarios	Source	Expected NPV ($B)	PEM Size (MW)	H2 Tank Size (kg)	H2 Turbine Size (MW)
1	DISPATCHES Notebook	1.596761	196.2616	2.0757E-05	9.0767E-05
	HERD	1.597810	196.2616	2.0888E-05	9.0767E-05
	% Error	-0.0657%	0.0000%	-0.6322%	0.0000%
2	DISPATCHES Notebook	1.631052	196.2616	1.4179E-05	9.0767E-05
	HERD	1.632101	196.2616	1.4248E-05	9.0764E-05
	% Error	-0.0643%	0.0000%	-0.4874%	0.0035%
3	DISPATCHES Notebook	1.660832	196.2616	1.4813E-05	9.0764E-05
	HERD	1.661881	196.2616	1.4885E-05	9.0764E-05
	% Error	-0.0632%	0.0000%	-0.4856%	0.0000%
4	DISPATCHES Notebook	1.654412	196.2616	1.4725E-05	9.0764E-05
	HERD	1.655462	196.2616	1.4797E-05	9.0764E-05
	% Error	-0.0634%	0.0000%	-0.4926%	0.0000%

3.2.2 Improvement to Flowsheet Implementation

In FY-23, HERD was updated to use the newest version of the IDAES codebase. One of the newest additions included directly instantiating the multiperiod class rather than calling an intermediate method from DISPATCHES. An intermediary method was previously needed to handle all period time sets, iteratively construct models, and collect them into a single model. The IDAES multiperiod class constructor was recently updated to handle hierarchical time sets and stochastic builds (i.e., multiple scenarios or realizations).

HERD can now call the IDAES multiperiod model constructor during runtime instead of using a single Pyomo ConcreteModel object. This occurs in the *_build_dispatches_model* method called within the *run* method. A schematic of what happens within is shown in Figure 13. The multiperiod model constructor is invoked with multiple input arguments, some of which are highlighted in the figure. The *process_model_func* input parameter in Figure 13 is shown to simply be an object instance of the nuclear flowsheet build method imported from the DISPATCHES repository; this is a simplified representation. In reality, a Python *partial* function is used so that the IDAES multiperiod class instead calls an intermediate function named *_flowsheet_block* within HERD. This could be considered a staging area that is called iteratively for every intended time period (e.g., every hour for every day for every year). The *partial* function takes as arguments a method and any additional arguments required downstream, creating a new function with new and larger sets of arguments. For example, we create a new *partial* function with the *_flowsheet_block* method and *staging_params* as an extra argument and use this as the *process_model_func* input to the IDAES multiperiod class constructor. Whenever the *process_model_func* is called within the multiperiod class, it points to the *partial* function rather than *_flowsheet_block* which holds an additional, fixed argument (*staging_params*) that is not in the original *process_model_func* call but is a required argument of the *_flowsheet_block* method.

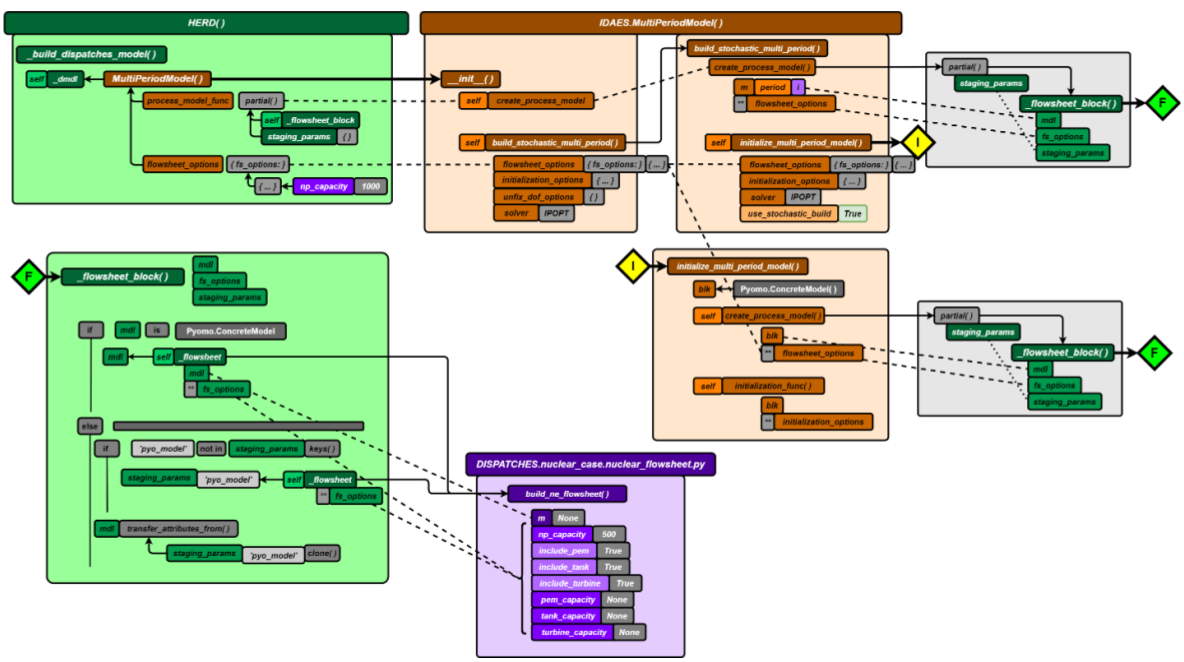

Figure 13: Schematic of IDAES Multiperiod constructor and subsequent stochastic build using intermediate flowsheet block method in HERD.

The purpose of the *_flowsheet_block* method is to speed up the iterative construction process of the multiperiod model. This is similar to what has been implemented within the DISPATCHES renewable energy case study: a brand new model is constructed for the first time period and saved to the empty *staging_params* dictionary. On subsequent iterations, instead of building a brand new model every time, the method clones the saved model and transfers all attributes to the new clone. This saves computation time for the nuclear case study. An important note is that for more complex models, some process initiation steps, including state propagation, will need to be conducted after the cloning process to ensure the cloned models are properly initialized with the correct inputs and parameters. The *_flowsheet_block* is being implemented currently on the DISPATCHES renewable energy case study. Nominally, the case study takes an extra wind speed time series for usage in the wind component, specifically when calling PySAM, that is loaded within DISPATCHES. Implementations in HERD are attempting to replace this call for static time series with synthetic histories from an ARMA model or static histories from a CSV file. This involves careful indexing within the *_flowsheet_block*.

3.3 Future Roadmap

The current implementation of HERD can reliably handle simulating the same IES configuration as the DISPATCHES nuclear case study using TEAL for cashflow generation and computing economic metrics. These are used in the objective function as well as adding stochasticity to the problem via synthetic history sampling from ARMA models. In essence, this is the converse of the newest multiperiod price-taker nuclear case Jupyter notebook in DISPATCHES which can externally load ROMs (in this case, ARMA models) for synthetic history generation and can call TEAL for cashflows and economic metrics. HERD will improve on this existing paradigm by providing a centralized class for running a multitude of flowsheet cases. Currently, this list of possible flowsheets is being expanded to include the renewable energy case study in DISPATCHES. Component parameters will be tunable from a HERON input script and provide the user with flexibility (constrained, of course, to the availability of case studies and algebraic unit models in the DISPATCHES and IDAES repositories).

An improvement to HERD which would make flowsheets more customizable is to add a catalog of wrapper classes for available components somewhere in HERON. While flowsheets contain all the important algebraic models for physical processes of components, some additional definitions have to be made after initializing the flowsheet. Some of these are shown in Figure 12, namely in sections C6 (the extraneous methods called when building the multiperiod model through IDAES), C7, and C8. In C6, after the flowsheet is called there are additional methods for fixing and unfixing certain degrees of freedom. Typically, these methods are unique to the configuration of selected components but if HERD is to be truly customizable, we can instead store these operations within a component wrapper class and loop through available components (a list of which we would have through checking for compatibility in B4). The wrapper classes could also contain non-anticipativity constraints, and constraints on linking variable pairs. An abstract class for these component wrappers will instruct future development when new components become available.

Figure 14 shows a current overview of all codebases. Stochastic optimization via simulation of multiple market and weather scenarios, as is done in HERON but with the “all-at-once” approach, has only been implemented for the nuclear case. Implementation of stochastic optimization in HERD for the renewable energy case, for which work is currently being conducted, will greatly expand the simulation capabilities of both codebases and allow the user to generate more useful optimization results of IES systems containing wind farms. Similar implementations can also be conducted for the fossil fuel case, which contains a complex model of an ultra-supercritical power plant with thermal storage.

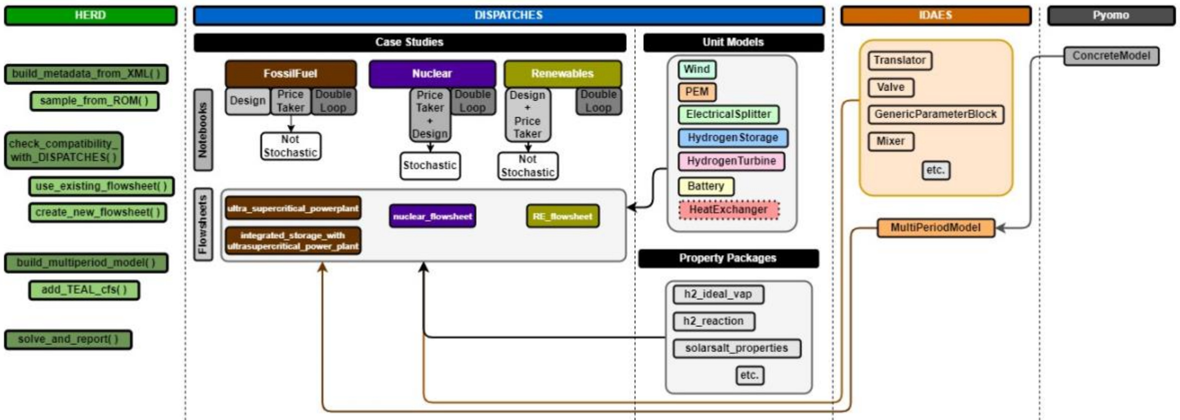

Figure 14: Overview of current and future HERD, DISPATCHES, IDAES and Pyomo workflow structure.

Currently, HERD can import already established flowsheets from the DISPATCHES directory. Future endeavors will aim to generate a toggle-able flowsheet for users to implement in their simulations. Figure 15 shows a diagram of what the full IES configuration could look like using existing DISPATCHES algebraic models. Since a Concrete thermal energy storage (TES) model is included, it might be of use to create a "power plant" model for nuclear that outputs thermal energy (or heat as shown in the diagram) rather than electricity. This resource can then be either stored or sent through a turbine model (available through IDAES) for generation of electricity. This subsystem can replace the NPP component in Figure 9.

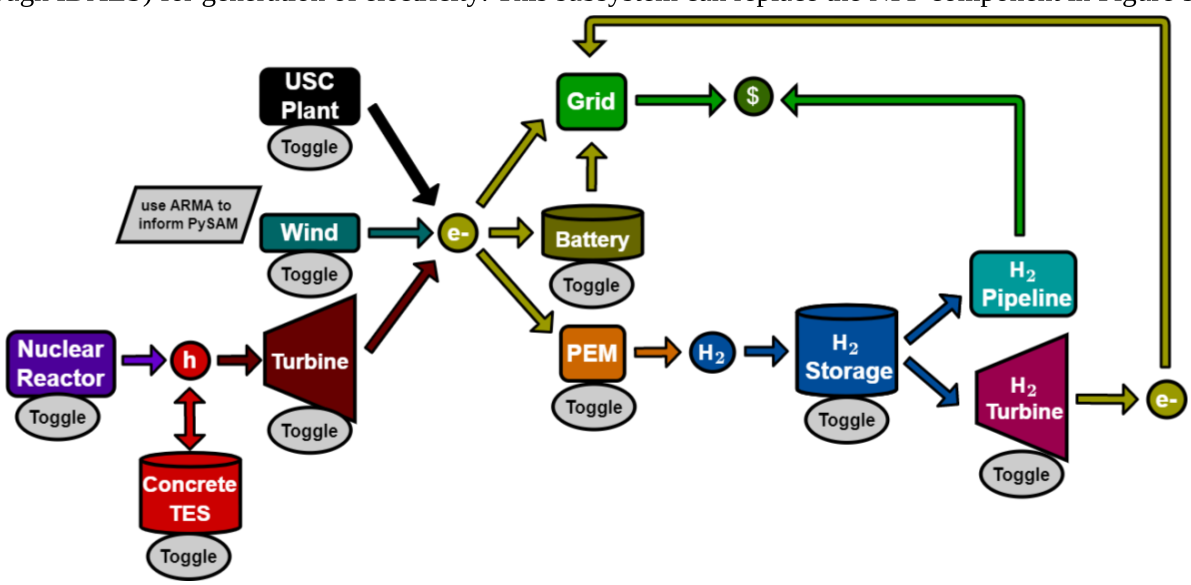

Figure 15: Possible future configurations of IES systems using DISPATCHES algebraic models.

Any subcomponent of this nuclear system can be toggled within the purported generalized flowsheet which would live within the HERON repository and used by HERD. Similarly, the user could choose to utilize a wind component or the fossil fuel plant as electricity generators. A battery component would store this electricity for later sales to the grid if selected by the user. Similarly, a PEM can be attached to generate a secondary commodity (hydrogen) for storage, sale, or combustion into electricity. Having this flexible IES configuration system although limited by availability of components) would give the user the option to choose from several optimization schemes in HERON—the outer-inner approach, the MOPED approach, or the DISPATCHES algebraic model approach—and determine which is the best tool for the job at hand.

4. Conclusion

Code-coupling developments for FY-23 have resulted in new interfaces for the external codes DRAFT and DISPATCHES within the FORCE toolset. These new integration strategies provide users with more ways to use the FORCE toolset and wider applicability to IES problem-solving.

The integration of DRAFT and FORCE has allowed users the capability of improving the accuracy of operating cost estimates by estimating the component's reliability and mean time to failure based on higher fidelity component failure models. Integrating DRAFT and HERON is expected to lead to a more accurate estimate of the optimized size of each component and optimized component dispatch informed by variable operational maintenance costs. Two features have been implemented in HERON to help the user understand how the IES economic metrics change after accounting for the components' reliability. These two features are: the time-dependent breakdown of the cashflow types for each component and the breakdown of the component resources (such as electricity production or steam consumption).

Implementation and improvement of HERD has provided a more robust architecture for interfacing with not just the nuclear-hydrogen case study models but the renewable energy and potentially any other publicly available DISPATCHES models. Future development and testing will be required to ensure the proper instantiation of DISPATCHES flowsheets within HERD, but the framework of those code-interfaces is currently available to users. The option of choosing between the standard RAVEN-runs-RAVEN optimization workflow versus the DISPATCHES workflow through HERD provides a lot of value to users who may want to model their IES components at different levels of fidelity, based on available computational resources and time.

5. References

1. Soto Gonzalez, G. J., and Talbot, P. W. 2022. "FORCE-DISPATCHES Integration - Initial Demonstration." Idaho National Laboratory. INL/RPT-22-69033-Rev000. https://doi.org/10.2172/1891636.

2. Gunter, D., et al. 2021. "Design, Integration and Synthesis Platform to Advance Tightly Coupled Hybrid Energy Systems (DISPATCHES) v0.1.0. Computer Software." U.S. Department of Energy. https://github.com/gmlc-dispatches/dispatches. https://doi.org/10.11578/dc.20211028.12.

3. Kluyver, T., et al. 2016. "Jupyter Notebooks -- a publishing format for reproducible computational workflows." *Positioning and Power in Academic Publishing: Players, Agents and Agendas.* IOS Press. 87-90. http://dx.doi.org/10.3233/978-1-61499-649-1-87.

4. Sohal, M. S., et al. 2010. "Engineering Database of Liquid Salt Thermophysical and Thermochemical Properties." Idaho National Laboratory. INL/EXT-10-18297. https://doi.org/10.2172/980801.

5. Chang, Z. S. et al. 2015. "The Design and Numerical Study of a 2MWh Molten Salt Thermocline Tank" Energy Procedia. **69**: 779-789. https://doi.org/10.1016/j.egypro.2015.03.094.

6. Bynum, M. L. et al. 2021. "Pyomo- Optimization Modeling in Python." Springer. Third Edition, **67**. https://doi.org/10.1007/978-3-030-68928-5.

7. Hart, W. E., Watson, J-P., and Woodruff, D. L. 2011. "Pyomo: modeling and solving mathematical programs in Python." *Mathematical Programming Computation.* **3**: 219-260. https://doi.org/10.1007/s12532-011-0026-8.

8. Yodwong, B., et al. 2020. “Proton Exchange Membrane Electrolyzer Modeling for Power Electronics Control: A Short Review.” C: Journal of Carbon Research. **6**: 29. https://doi.org/10.3390/c6020029.

9. Taamallah, S., et al. 2015. “Fuel flexibility, stability and emissions in premixed hydrogen-rich gas turbine combustion: Technology, fundamentals, and numerical simulations.” Applied Energy. **154**: 1020-1047. https://doi.org/10.1016/j.apenergy.2015.04.044.

10. NREL. n.d. “PySAM Version 2.2.0.” National Renewable Energy Laboratory. Accessed December 27, 2020. https://github.com/nrel/pysam.

11. Griffith, A. 2022. “Moped Storage and Consumption #200.” Github. Accessed September 2023. https://github.com/idaholab/HERON/pull/200.

12. Wächter, A., and Biegler, L.T., 2006. “On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming.” *Mathematical Programming.* **106**: 25-57. https://doi.org/10.1007/s10107-004-0559-y.

13. Frick, K. L., et al. 2022. “Hybrid user manual.” Idaho National Laboratory. INL/MIS-20-60624-Rev001. https://inldigitallibrary.inl.gov/sites/sti/sti/Sort_57481.pdf.

14. Talbot, P., et al. 2022. “HERON User Guide.” Idaho National Laboratory. INL/EXT-20-58976-Rev001, GDE-939. https://github.com/idaholab/HERON/tree/devel/doc

15. Saeed, R. M. et al. 2022. “Multilevel Analysis, Design, and Modeling of Coupling Advanced Nuclear Reactors and Thermal Energy Storage in an Integrated Energy System.” Idaho National Laboratory. INL/RPT-22-69214-Rev000. https://doi.org/10.2172/1890160.

16. Soto Gonzalez, G. J. et al. 2023. “FORCE Integration with DRAFT and IDEAS.” Idaho National Laboratory. INL/RPT-23-71960-Rev000.