IES
Integrated Energy Systems

LWRS
LIGHT WATER REACTOR SUSTAINABILITY

HERON Workshop Examples

A case-by-case training in running HERON

INL/MIS-22-65661

Idaho National Laboratory

Learning by Example

- HERON cases from basic to complex
- Exercises, follow-alongs
- Examples found in HERON/tests/workshop/
  - This case can be found in workshop/simple/

HERON: Workflow Writer

RAVEN Outer

Optimize Capacity

Portfolio

E[NPV]

RAVEN Inner

Sample

Postprocess

Ensemble:

SynthHist

Dispatch

TEAL

DispatchManager

Loop Structure

Optimize Dispatch

Technology Characteristics

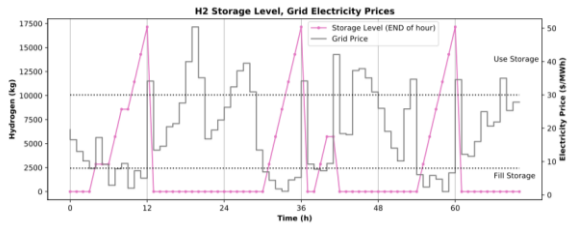
DispatchManager

For each year …

For each cluster …

… optimize dispatch

Optimization Window → Time History



HERON Calculation Flow

IES Integrated Energy Systems

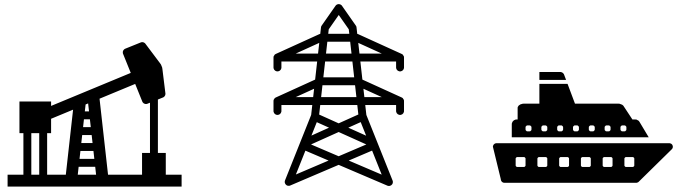
Example 1

As Simple As They Come

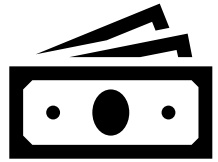
IES
Integrated Energy Systems

Starter Case

- Components
  - NGCC (natural gas electricity generator)
  - Import (imports electricity from external)
  - Grid (demand to be met)

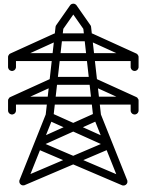


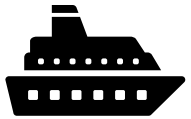
- Objective
  - How big should the NGCC be built to minimize costs?



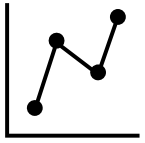
Starter Case

- Guiding Physics/Economics (Drivers)
  - Grid
    - fixed demand to be met
  - NGCC
    - Capital cost for sizing
    - Variable cost for dispatching
  - Import
    - Variable cost (expensive!) for providing electricity





Translating to HERON

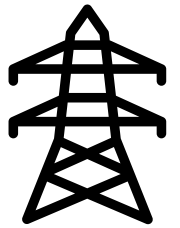


- See `HERON/tests/workshop/simple/heron_input.xml`
- Case
  - Global information about the problem
  - Time shape, discount rates, solvers, etc.
- Components
  - Technical and economic properties of each component
  - Produces and Demands
  - Dispatch: Independent and Fixed
- DataGenerators
  - Synthetic History Data Source

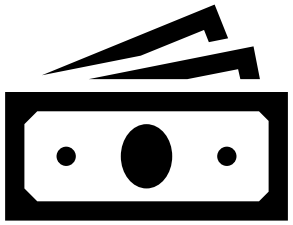
Component by Component

- Grid

Technical Specs	
Action	Demands electricity
Flexibility	Fixed (must meet demand)
Capacity	From synthetic time series (trained separately)



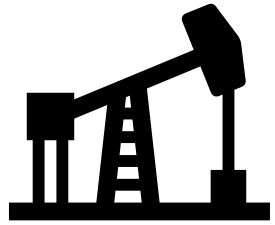
Economics	
None	System is regulated market; minimize cost to meet demand, so there is no sales profit at the grid



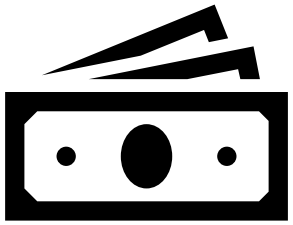
Component by Component

- NGCC

Technical Specs	
Action	Produces electricity
Flexibility	Independent (0 to [Capacity] each hour)
Capacity	Variable from 10 to 60 GW (note: units not included)



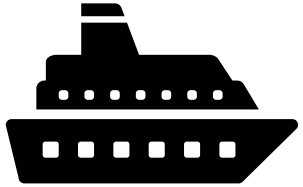
Economics	
Capex	Overnight cost of building component per GW installed
Variable O&M	Operation and fuel costs per GWh produced



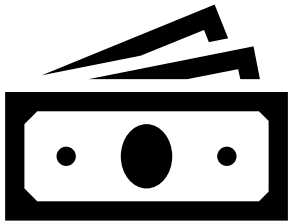
Component by Component

- Import

Technical Specs	
Action	Produces electricity
Flexibility	Independent (0 to [Capacity] each hour)
Capacity	Large number (infinite source)

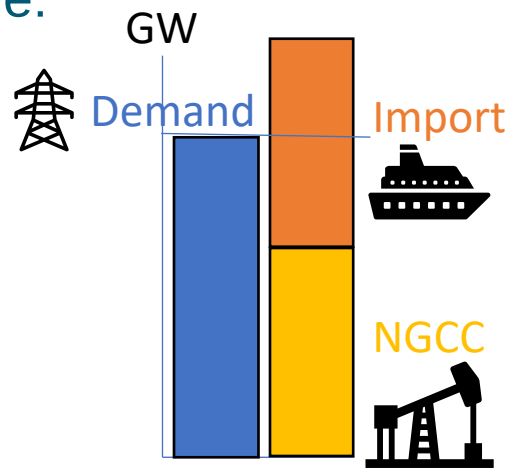


Economics	
Variable O&M	Operation costs per GWh imported



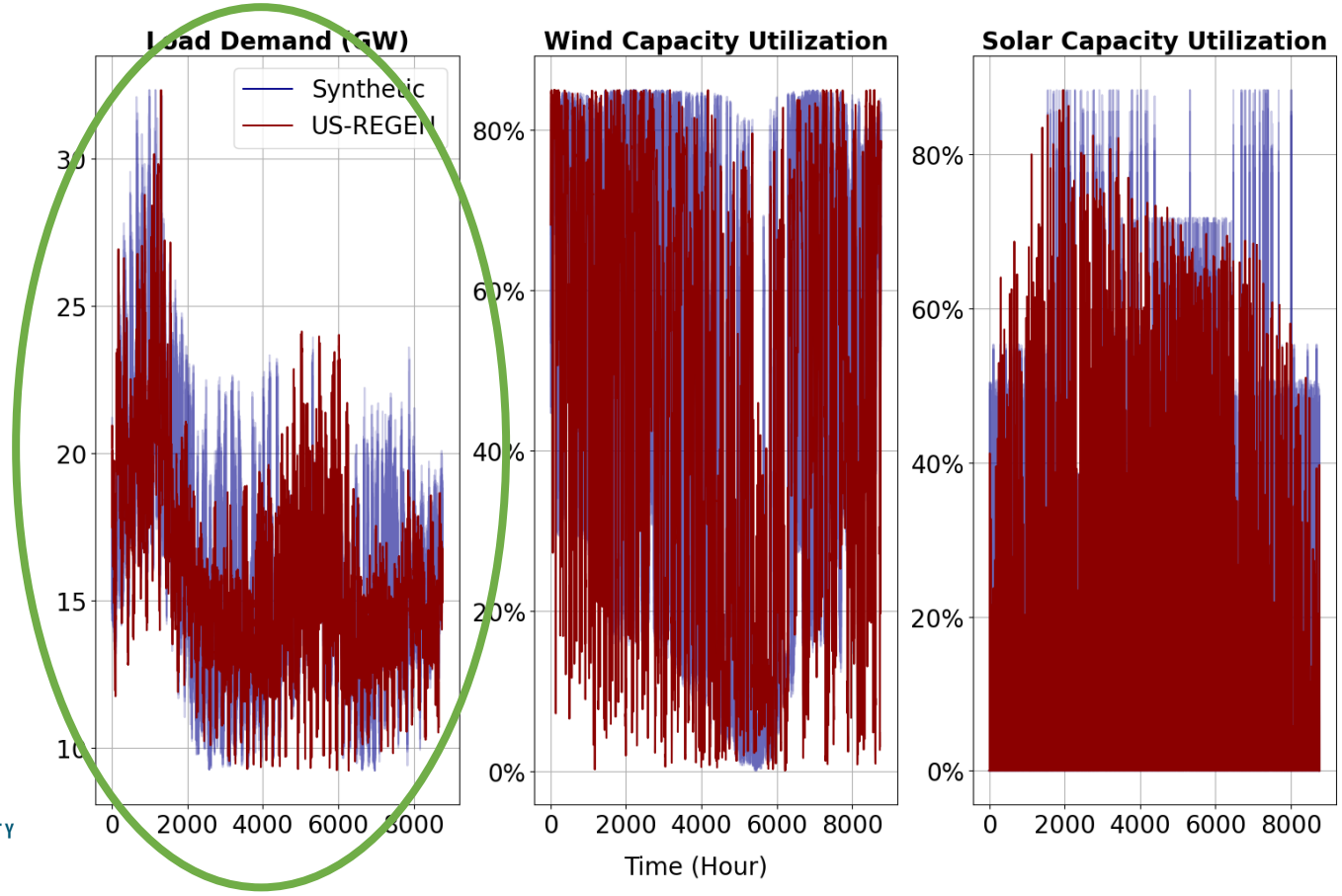
Consider the Case

- Why Import?
  - Why not just have NGCC?
  - If NGCC not enough to meet demand, solve “fails”
    - This doesn’t provide useful feedback to optimizer; it doesn’t know what’s wrong
    - Instead, provide large incentive to meet demand with NGCC
    - Example:



Demand

- Stochastically Trained
- Based on NYISO projections from EPRI



See INL/EXT-21-65473, "A Technical and Economic Assessment of LWR Flexible Operation for Generation/Demand Balancing to Optimize Plant Revenue", Dec 2021

LWRS LIGHT WATER REACTOR SUSTAINABILITY

IES Integrated Energy Systems

Running the Case

```
> /path/to/heron heron_input.xml
> /path/to/raven_framework outer.xml
```

- Sweeping over 6 build sizes
  - ~90 seconds, results may vary by OS, machine

- Things to look at (see following slides):
  - Screen Output
  - `1_simple_o/sweep.csv` (spreadsheet) progress

Things to Look At While Running

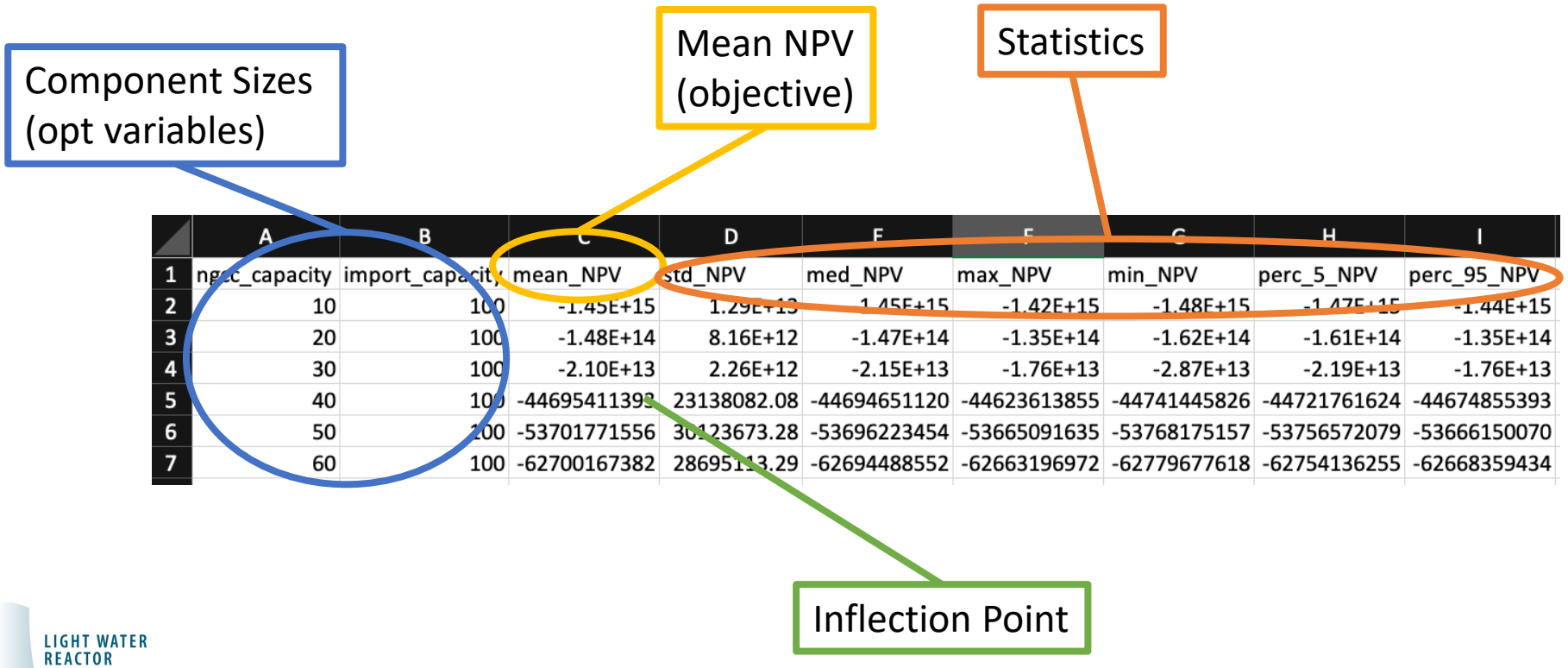
- Screen Output: How To Read the Matrix
  - Sweep Sample
    - Submission of each individual parametric sweep value

```
(           ) UTILS                    : Message         -> importing module /Users/talbpw/projects/HERON/tests/workshop/simple/write_inner.py
current working dir /Users/talbpw/projects/HERON/tests/workshop/simple/1_simple_o/sweep/1/1_simple_i
already exists, this might imply deletion of present files
(    0.04 sec) CODE MODEL              : Message         -> Execution command submitted: python /Users/talbpw/projects/raven/raven_framework.py inner.xml
(   14.30 sec) CODE MODEL              : Message         -> job "2" submitted!
(           ) UTILS                    : Message         -> importing module /Users/talbpw/projects/HERON/tests/workshop/simple/write_inner.py
current working dir /Users/talbpw/projects/HERON/tests/workshop/simple/1_simple_o/sweep/2/1_simple_i
already exists, this might imply deletion of present files
(   14.31 sec) CODE MODEL              : Message         -> Execution command submitted: python /Users/talbpw/projects/raven/raven_framework.py inner.xml
(   28.35 sec) CODE MODEL              : Message         -> job "3" submitted!
(           ) UTILS                    : Message         -> importing module /Users/talbpw/projects/HERON/tests/workshop/simple/write_inner.py
current working dir /Users/talbpw/projects/HERON/tests/workshop/simple/1_simple_o/sweep/3/1_simple_i
already exists, this might imply deletion of present files
(   28.37 sec) CODE MODEL              : Message         -> Execution command submitted: python /Users/talbpw/projects/raven/raven_framework.py inner.xml
(   42.43 sec) CODE MODEL              : Message         -> job "4" submitted!
(           ) UTILS                    : Message         -> importing module /Users/talbpw/projects/HERON/tests/workshop/simple/write_inner.py
current working dir /Users/talbpw/projects/HERON/tests/workshop/simple/1_simple_o/sweep/4/1_simple_i
already exists, this might imply deletion of present files
```

Things to Look At While Running

- `1_simple_o/sweep.csv` – progress of the sampler

`1_simple_o` = case name; `o` = outer; `sweep.csv` = sampling record



	A	B	C	D	E	F	G	H	I
1	ngcc_capacity	import_capacity	mean_NPV	std_NPV	med_NPV	max_NPV	min_NPV	perc_5_NPV	perc_95_NPV
2	10	100	-1.45E+15	1.29E+13	-1.45E+15	-1.42E+15	-1.48E+15	-1.47E+15	-1.44E+15
3	20	100	-1.48E+14	8.16E+12	-1.47E+14	-1.35E+14	-1.62E+14	-1.61E+14	-1.35E+14
4	30	100	-2.10E+13	2.26E+12	-2.15E+13	-1.76E+13	-2.87E+13	-2.19E+13	-1.76E+13
5	40	100	-44695411393	23138082.08	-44694651120	-44623613855	-44741445826	-44721761624	-44674855393
6	50	100	-53701771556	30123673.28	-53696223454	-53665091635	-53768175157	-53756572079	-53666150070
7	60	100	-62700167382	28695113.29	-62694488552	-62663196972	-62779677618	-62754136255	-62668359434

LWRS LIGHT WATER REACTOR SUSTAINABILITY

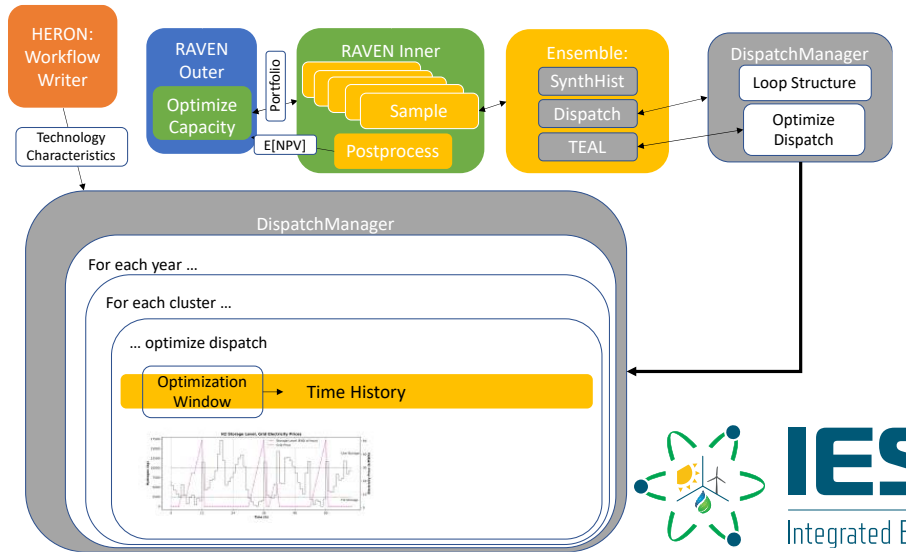
IES Integrated Energy Systems

Converting to Optimization

- Converting Input
  - Change <mode> to opt
  - Swap from <sweep_values> to <opt_bounds>
    - Use sweep results to establish bounds

	A	B	C	D
1	ngcc_capacity	import_capacity	mean_NPV	std_NPV
2	10	100	-1.45E+15	1.29E+13
3	20	100	-1.48E+14	8.16E+12
4	30	100	-2.10E+13	2.26E+12
5	40	100	-44695411393	23138082.08
6	50	100	-53701771556	30123673.28
7	60	100	-62700167382	28695113.29

- Optimization requires many more “outer” samples!
  - ~90 minutes for this case to converge
    - Time may vary by OS, machine
  - ~250 iterations



LWRS LIGHT WATER REACTOR SUSTAINABILITY

IES Integrated Energy Systems

Running the Case

```
> /path/to/heron heron_input.xml
> /path/to/raven_framework outer.xml
```

- Optimizing size (takes some time!)

- Things to look at (see following slides):
  - Screen Output
  - `1_simple_o/opt_soln_0.csv` (spreadsheet) optimizer progress

Things to Look At While Running

- Screen Output: How To Read the Matrix
  - Accepted Sample
    - This means a new optimal point has been found!

```
( 7704.86 sec) PointSet                : DEBUG          -> Wrote master cluster file to "opt_soln.csv"
( 7704.87 sec) PointSet                : DEBUG          -> Wrote sub-cluster file to "opt_soln_0.csv"
( 7704.88 sec) PointSet                : DEBUG          -> Printing metadata XML: "opt_soln.xml"
( 7704.88 sec) STEP MULTIRUN           : DEBUG          -> Just collected job   261    and sent to output "opt_soln"
( 7704.88 sec) GradientDescent         : DEBUG          -> ********************************************************************************
( 7704.88 sec) GradientDescent         : DEBUG          -> Trajectory 0 iteration 171 resolving new opt point ...
( 7704.88 sec) GradientDescent         : DEBUG          ->   ... change: -3.520e+05 new:  7.202887e+09 old:  7.203239e+09
( 7704.88 sec) GradientDescent         : DEBUG          ->   .. accepted!
( 7704.88 sec) GradientDescent         : DEBUG          -> Convergence Check for Trajectory 0:
( 7704.88 sec) GradientDescent         : DEBUG          ->   ...  gradient  : False, 3.33e-02 / 1.00e-04
( 7704.88 sec) GradientDescent         : DEBUG          ->   ... objective  : False, 2.17e-05 / 1.00e-08
( 7704.88 sec) GradientDescent         : DEBUG          ->   ... same point : False, 0.00e+00 / 1.00e+00
( 7704.88 sec) GradientDescent         : DEBUG          -> Resetting convergence for trajectory 0.
( 7704.88 sec) GradientDescent         : DEBUG          -> ********************************************************************************
( 7704.88 sec) STEP MULTIRUN           : DEBUG          -> Testing if the sampler is ready to generate a new input
( 7704.89 sec) GradientDescent         : DEBUG          ->   ... Sample point 262: {'ngee_capacity': 39.989015501936606, 'denoises': 20, 'import_co
 100.0}
( 7704.89 sec) CODE MODEL              : Message        -> job "262" submitted!
```

Things to Look At While Running

- Screen Output: How To Read the Matrix
  - Rejected Sample
    - This means the new proposed opt point is worse than the old opt point
    - Reasons: inaccurate gradient, too large step, no better points nearby
    - Happens frequently, especially near the end

```
( 7338.73 sec) STEP MULTIRUN          : DEBUG          -> Just collected job   249    and sent to output "opt_eval"
( 7338.74 sec) PointSet               : DEBUG          -> Wrote master cluster file to "opt_soln.csv"
( 7338.75 sec) PointSet               : DEBUG          -> Wrote sub-cluster file to "opt_soln_0.csv"
( 7338.75 sec) PointSet               : DEBUG          -> Printing metadata XML: "opt_soln.xml"
( 7338.75 sec) STEP MULTIRUN          : DEBUG          -> Just collected job   249    and sent to output "opt_soln"
( 7338.76 sec) GradientDescent        : DEBUG          -> ********************************************************************************
( 7338.76 sec) GradientDescent        : DEBUG          -> Trajectory 0 iteration 163 resolving new opt point ...
( 7338.76 sec) GradientDescent        : DEBUG          ->   ... change:  1.706e+06 new:  7.204945e+09 old:  7.203239e+09
( 7338.76 sec) GradientDescent        : DEBUG          ->   ... rejected!
( 7338.76 sec) GradientDescent        : DEBUG          -> ********************************************************************************
( 7338.76 sec) GradientDescent        : DEBUG          -> Canceling grad jobs for traj "0" iteration "163".
( 7338.76 sec) GradientDescent        : DEBUG          -> * Submitting new opt and grad points *
( 7338.76 sec) GradientDescent        : DEBUG          -> Adding run to queue: {'ngcc_capacity': 39.989015501936677, 'import_capacity': 100.0
```

Things to Look At While Running

- Screen Output: How To Read the Matrix
  - Rerun Sample
    - Return to best-so-far opt point and rerun the gradient, cut step size
    - Helps to resolve two reasons for rejecting proposed opt points

```
( 7370.07 sec) PointSet                : DEBUG          -> Wrote master cluster file to "opt_soln.csv"
( 7370.09 sec) PointSet                : DEBUG          -> Wrote sub-cluster file to "opt_soln_0.csv"
( 7370.09 sec) PointSet                : DEBUG          -> Printing metadata XML: "opt_soln.xml"
( 7370.09 sec) STEP MULTIRUN           : DEBUG          -> Just collected job   250    and sent to output "opt_soln"
( 7370.10 sec) GradientDescent         : DEBUG          -> ********************************************************************************
( 7370.10 sec) GradientDescent         : DEBUG          -> Trajectory 0 iteration 164 resolving new opt point ...
( 7370.10 sec) GradientDescent         : DEBUG          ->   ... change:  1.956e+06 new:  7.205195e+09 old:  7.203239e+09
( 7370.10 sec) GradientDescent         : DEBUG          ->   ... rerun!
( 7370.10 sec) GradientDescent         : DEBUG          -> ********************************************************************************
( 7370.10 sec) STEP MULTIRUN           : DEBUG          -> Testing if the sampler is ready to generate a new input
( 7370.10 sec) GradientDescent         : DEBUG          ->   ... Sample point 251: {'ngcc_capacity': 39.989015501936677, 'denoises': 20, 'import_
 100.0}
( 7370.10 sec) CODE MODEL              : Message        -> job "251" submitted!
```

Things to Look At While Running

- `1_simple_o/opt_soln_0.csv` – progress of the optimizer

`1_simple_o` = case name; `o` = outer; `opt_soln_0.csv` = optimization record

Optimizer Steps (sets of samples)

Component Sizes (opt variables)

Mean NPV (objective)

Status of each step
- Accepted: improvement!
- Rejected: went too far
- Rerun: re-check gradient

	A	B	C	D	E
1	iteration	accepted	ngcc_capacity	import_capa	mean_NPV
2	0	first	11.5	100	-1.423E+10
3	1	accepted	17.5	100	-8.773E+09
4	2	accepted	23.5	100	-7.436E+09
5	3	rejected	11.5	100	-1.423E+10
6	4	rerun	23.5	100	-7.432E+09
7	5	accepted	31.5	100	-7.229E+09
8	6	accepted	36.833333	100	-7.207E+09
9	7	rejected	26.166667	100	-7.292E+09
10	8	rerun	36.833333	100	-7.202E+09
11	9	rejected	29.722222	100	-7.234E+09
12	10	rerun	36.833333	100	-7.204E+09
13	11	rejected	33.092593	100	-7.224E+09

Note our losses are getting smaller!

LWRS LIGHT WATER REACTOR SUSTAINABILITY

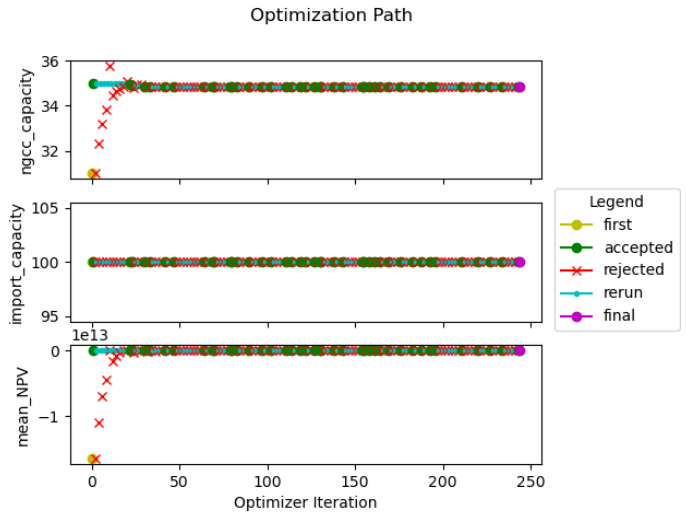
IES Integrated Energy Systems

Now that it's done …

- Things to look at (see next slides)
  - `opt_soln_0.csv`
    - filter opt CSV by "accepted" to see acceptance path
    - final solution is the last "accepted" point in the CSV
  - `opt_path.png`
    - visualization of opt_soln_0.csv
  - screen output

	A	B	C	D	E
171	169	rejected	39.989016	100	-7.21E+09
172	170	rerun	39.989016	100	-7.2E+09
173	171	accepted	39.989016	100	-7.2E+09
174	172	accepted	39.989016	100	-7.2E+09
175	173	rejected	39.989016	100	-7.2E+09
176	174	rerun	39.989016	100	-7.2E+09
177	175	rejected	39.989016	100	-7.21E+09
178	176	rerun	39.989016	100	-7.2E+09
179	177	rejected	39.989016	100	-7.21E+09
180	178	rerun	39.989016	100	-7.21E+09
181	179	rejected	39.989016	100	-7.2E+09
182	180	rerun	39.989016	100	-7.2E+09
183	181	rejected	39.989016	100	-7.2E+09
184	182	rerun	39.989016	100	-7.21E+09
185	183	rejected	39.989016	100	-7.21E+09
186	184	rerun	39.989016	100	-7.21E+09
187	185	rejected	39.989016	100	-7.21E+09
188	186	rerun	39.989016	100	-7.21E+09
189	187	rejected	39.989016	100	-7.2E+09
190	188	rerun	39.989016	100	-7.21E+09
191					

```
->  ****************************************
->  Optimizer Final Results:
->    - Final Optimal Point:
->              mean_NPV          -4.007e+10
->               trajID                  0
->          ngcc_capacity          3.485e+01
->  ****************************************
```



LWRS LIGHT WATER REACTOR SUSTAINABILITY

IES Integrated Energy Systems

Things to look at now that it's done

`Screen Output`

- Summary of optimizer status
  - Trajectory that found best point
  - Objective value
  - Opt Vars values

```
-> ****************************************
-> Optimizer Final Results:
->  - Final Optimal Point:
->           mean_NPV        -4.007e+10
->            trajID          0
->       ngcc_capacity        3.485e+01
-> ****************************************
```

objective

opt var

Things to look at now that it's done

`1_simple_o/opt_soln_0.csv`

- "final" point is solution
  - 243 iterations is low (good!)

1	iteration	accepted	ngcc_capacit	import_capac	mean_NPV	std_NPV	med_NPV
2	0	first	31	100	-1.64E+13	1.50E+12	-1.71E+13
3	1	accepted	35	100	-4.02E+10	20730339	-4.02E+10
238	236	rerun	34.845283	100	-4.006E+10	21055041	-4.006E+10
239	237	rejected	34.845283	100	-4.006E+10	16911537	-4.006E+10
240	238	rerun	34.845283	100	-4.005E+10	16726200	-4.005E+10
241	239	rejected	34.845283	100	-4.006E+10	22230139	-4.006E+10
242	240	rerun	34.845283	100	-4.006E+10	17962306	-4.007E+10
243	241	rejected	34.845283	100	-4.006E+10	28813156	-4.007E+10
244	242	rerun	34.845283	100	-4.006E+10	24491243	-4.005E+10
245	243	accepted	34.845283	100	-4.007E+10	18944063	-4.007E+10
246	243	final	34.845283	100	-4.007E+10	18944063	-4.007E+10

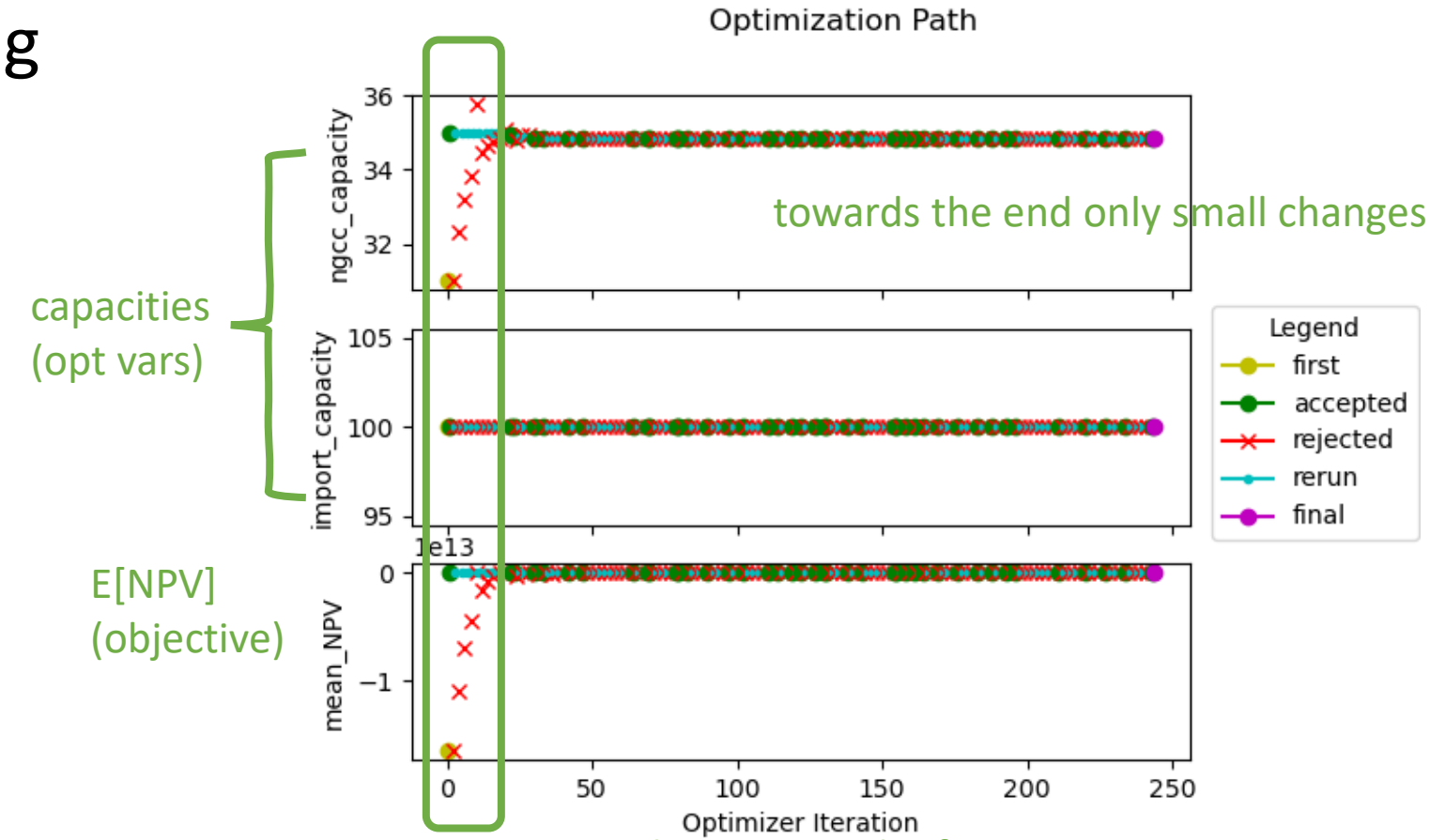
opt var

objective

Things to look at now that it's done

`1_simple_o/opt_path.png`

- Messy plot
- Many plots are bad
  - but some are useful
- Not intended for report quality
  - Indicative of process



Wrap up Example 1

- That's it!
- Takeaways:
  - 3-unit problem, single resource
  - Setting up feasible problems
  - Running and observing HERON runs
  - Viewing results