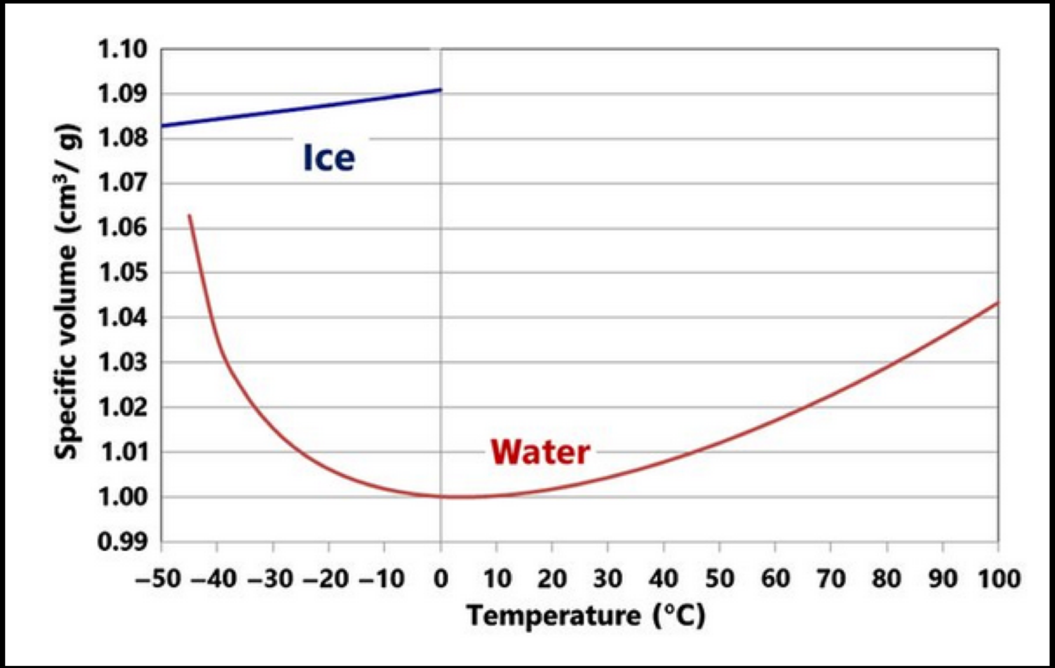
How to use sensitivity analyses to explore RuFaS performance

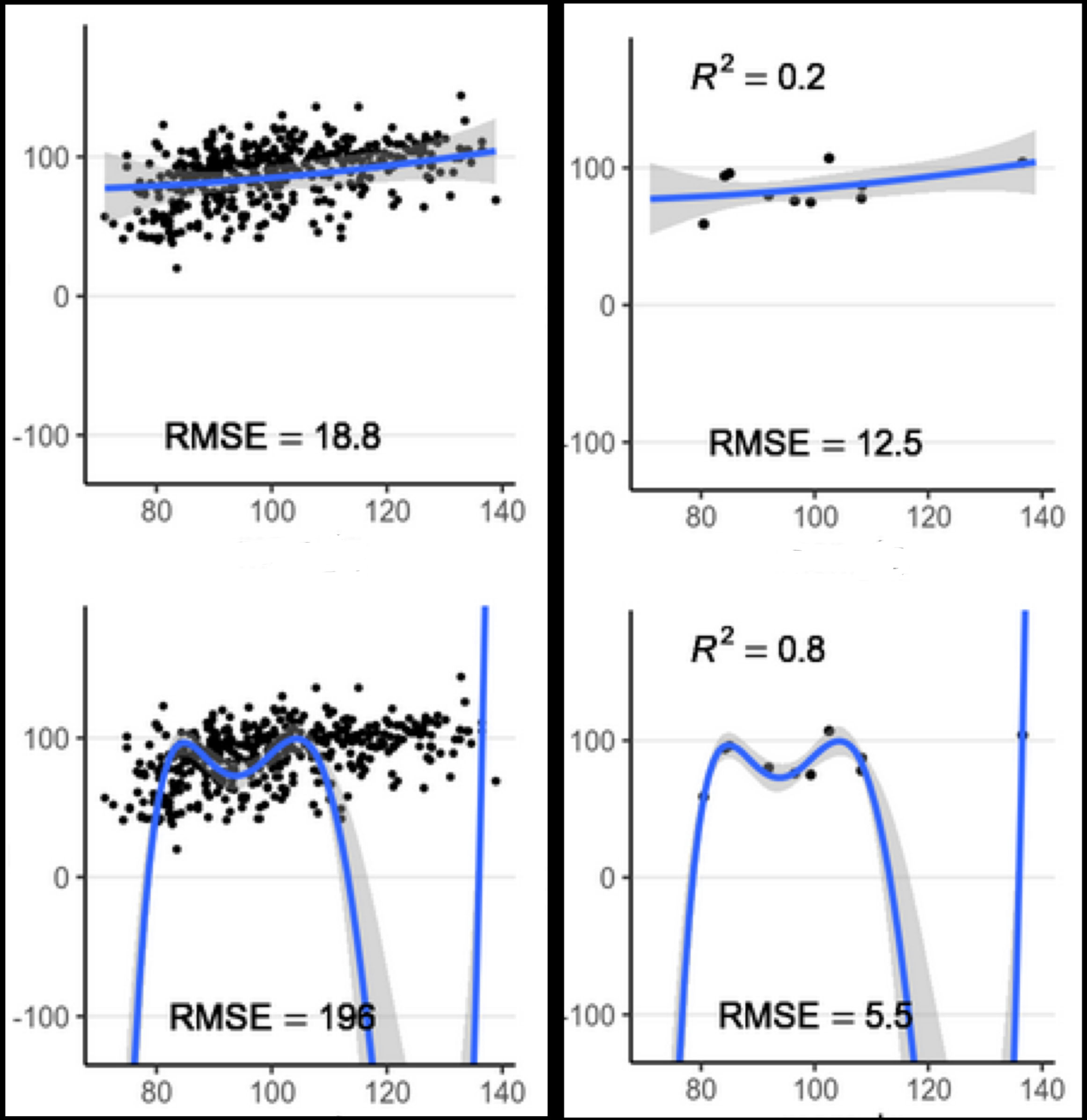
Or, not-so-spherical cows in parallel universes

Joseph C. Waddell
Cornell University
2022 November 4

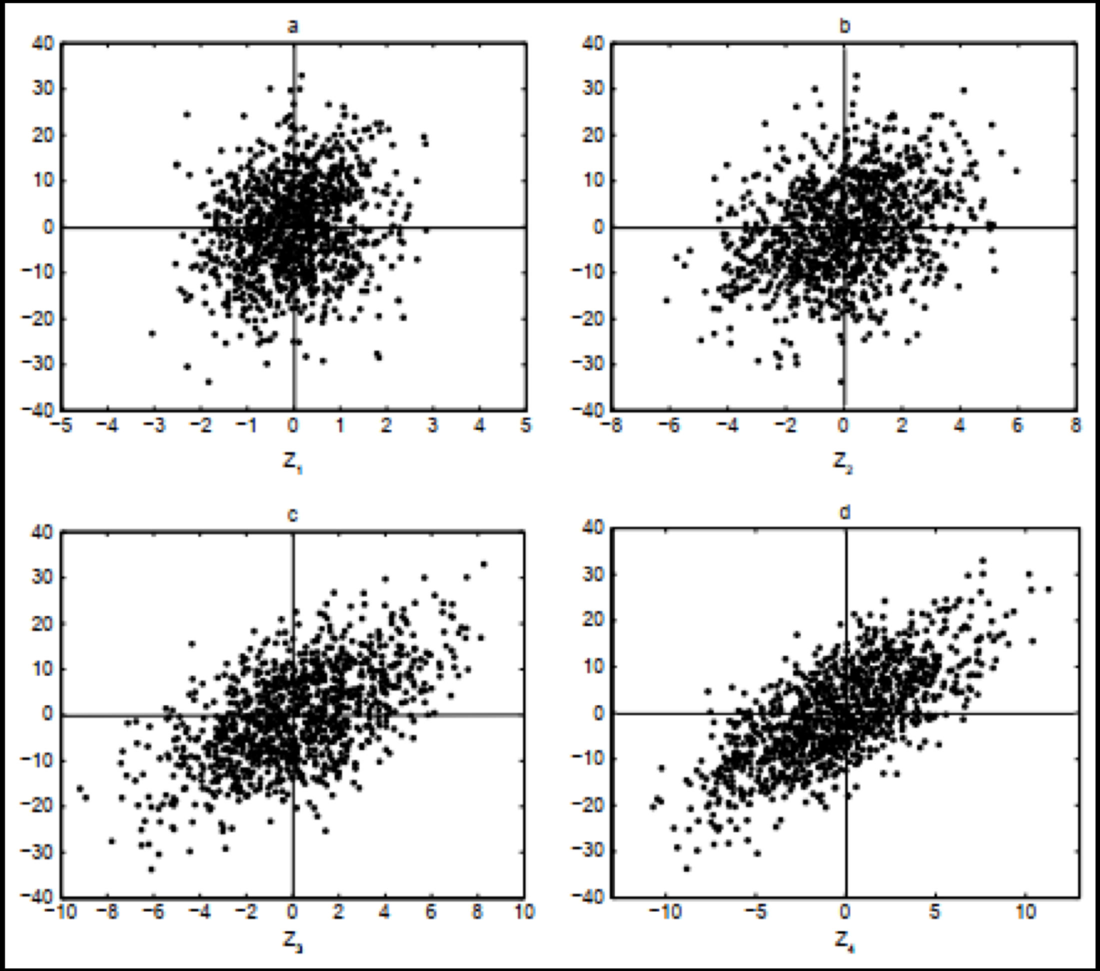
Models and Uncertainty

- Model 1: $R^2 = 0.2$
- Model 2: $R^2 = 0.8$



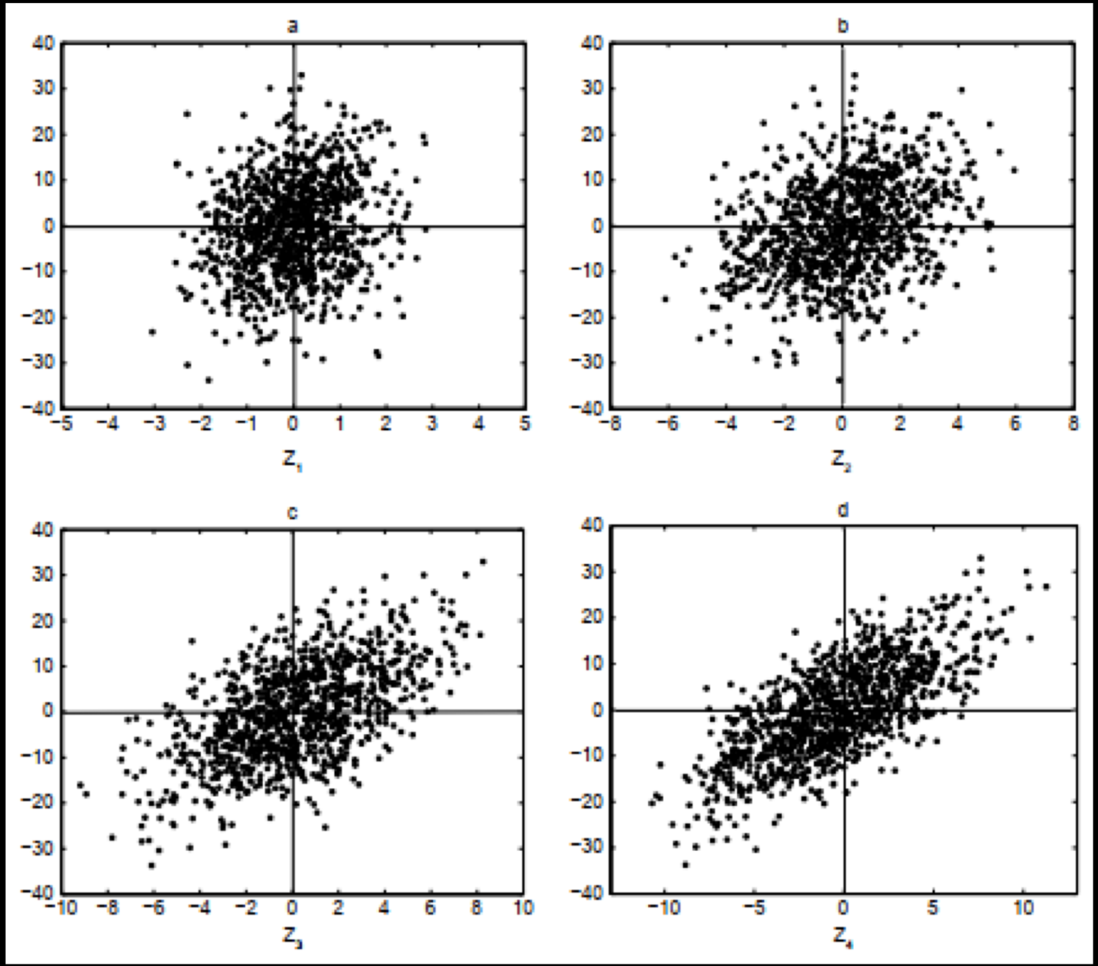


The **what** of sensitivity analyses



The **what** of sensitivity analyses

- A few key terms:
  - Diagnostic v. prognostic models
  - Law-driven v. data-driven models
  - Local v. Global analysis
  - Scatterplot v. derivative methodology



Question: which type(s) of sensitivity analysis approaches are important for RuFaS?

- Global analyses on a prognostic model that is both law- and data-driven

…also

- Local analysis of any given variable to inform future model improvements

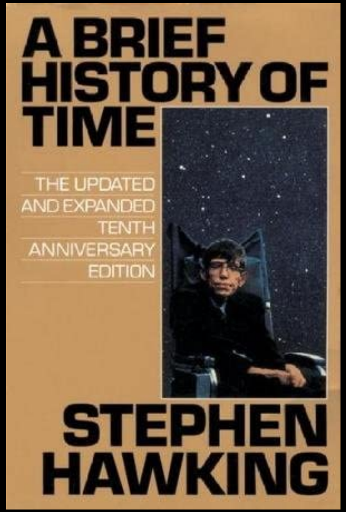
The **why** of sensitivity analyses for RuFaS

- <u>Model evaluation: computational insight</u>
  - Which parameters are most influential in the model?
  - Do certain parameter ranges cause unexpected behavior?
- <u>Model evaluation: the biological underpinnings</u>
  - Model performance and accuracy
  - Unexpected biological interactions?
- <u>Practical concerns in moving forward</u>
  - Which parameters should be measured in the field?

The **how** of sensitivity analyses

- Decide on a set of variables and their bounds
- Vary the inputs
- Run the model for each input set
- Record the outputs of interest
- Evaluate the results

….repeat?



Equation (1.36) can be rewritten as

$$\frac{V(E(Y \mid Z_i, \Omega_j))}{V(Y)} = S_{Z_i} + S_{\Omega_i} + S_{Z_i,\Omega_i},$$

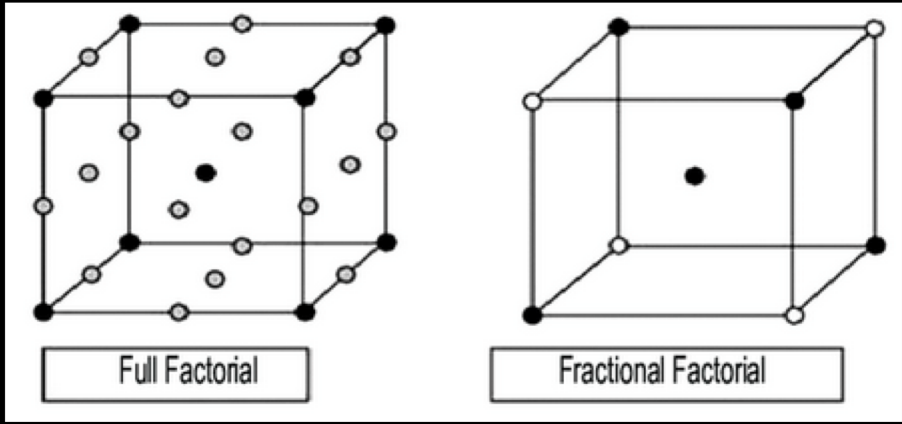
where $S_{\Omega_i} = 0$, $S_{Z_i,\Omega_i} = V_{Z_i,\Omega_i}/V(Y)$ and the term V_{Z_i,Ω_i} is the only type of nonzero second-order term in model (1.3).

Table 1.4 First- and second-order indices for model (1.3, 1.27, analytic), where $r = 4$, $c = 0.5$, $\sigma = (1, 2, 3, 4)$ for both Ω_i and Z_i

Factor	S_i, S_{ij}	*Factor*	S_{ij}	*Factor*	S_{ij}
Z_1	0.0006	Z_1, Ω_2	0	Z_3, Ω_3	0.183
Z_2	0.009	Z_1, Ω_3	0	Z_3, Ω_4	0
Z_3	0.046	Z_1, Ω_4	0	Z_4, Ω_1	0
Z_4	0.145	Z_2, Z_3	0	Z_4, Ω_2	0
Ω_1	0	Z_2, Z_4	0	Z_4, Ω_3	0
Ω_2	0	Z_2, Ω_1	0	Z_4, Ω_4	0.578
Ω_3	0	Z_2, Ω_2	0.036	Ω_1, Ω_2	0
Ω_4	0	Z_2, Ω_3	0	Ω_1, Ω_3	0
Z_1, Z_2	0	Z_2, Ω_4	0	Ω_1, Ω_4	0
Z_1, Z_3	0	Z_3, Z_4	0	Ω_2, Ω_3	0
Z_1, Z_4	0	Z_3, Ω_1	0	Ω_2, Ω_4	0
Z_1, Ω_1	0.002	Z_3, Ω_2	0	Ω_3, Ω_4	0

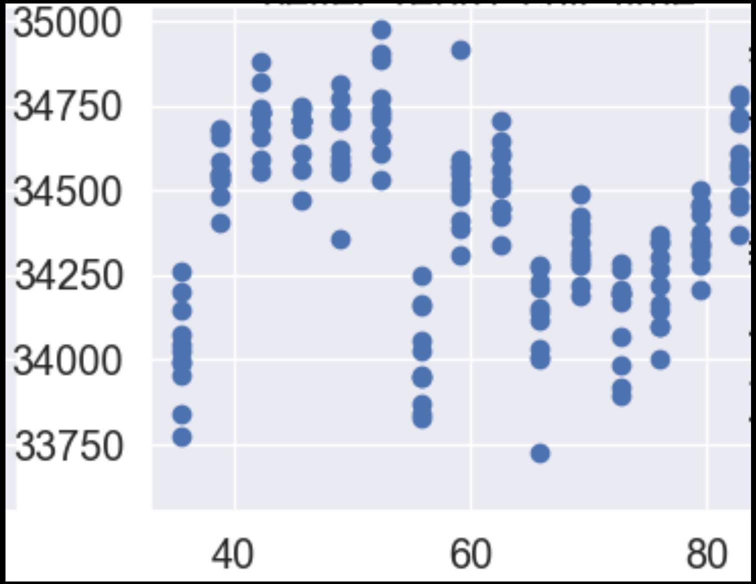
Sample approach: Fractional factorial design

Assay	Variable levels[a]					Response
	SCM[b]	CSL[c]	YEP[d]	K_2HPO_4	pH	Enzyme activity (U/ml)
1	–1	–1	–1	–1	1	2.27
2	1	–1	–1	–1	–1	1.11
3	–1	1	–1	–1	–1	4.11
4	1	1	–1	–1	1	12.44
5	–1	–1	1	–1	–1	3.59
6	1	–1	1	–1	1	14.68
7	–1	1	1	–1	1	5.56
8	1	1	1	–1	–1	0.20
9	–1	–1	–1	1	–1	2.40
10	1	–1	–1	1	1	5.89
11	–1	1	–1	1	1	5.77
12	1	1	–1	1	–1	0.20
13	–1	–1	1	1	1	5.12
14	1	–1	1	1	–1	0.68
15	–1	1	1	1	–1	4.82
16	1	1	1	1	1	11.92



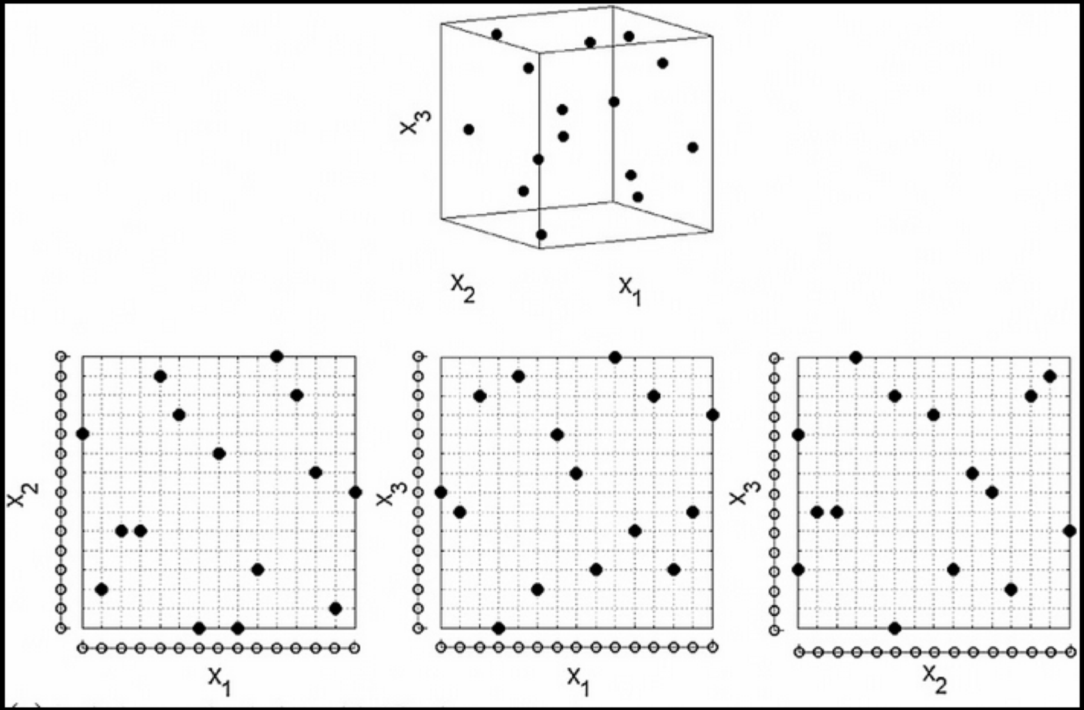
Raza et al 2013

Expectations of nonlinearity in RuFaS

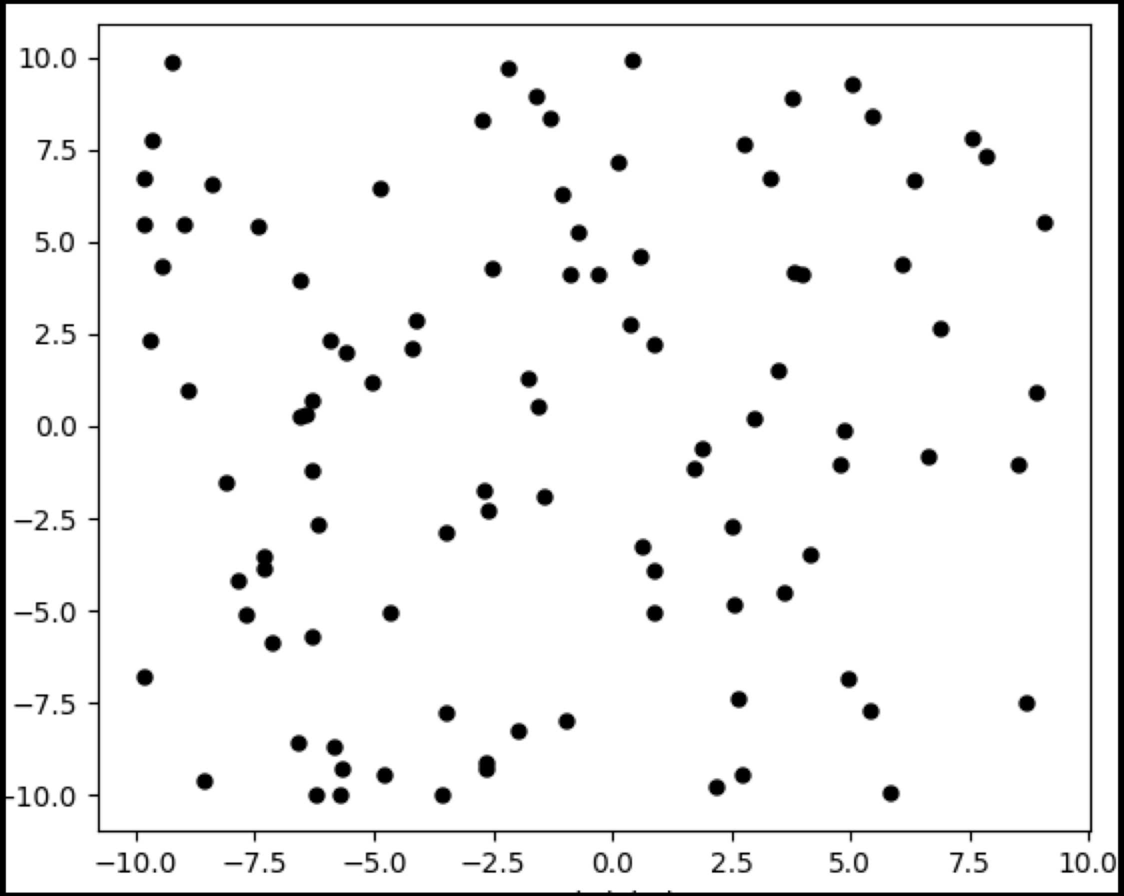


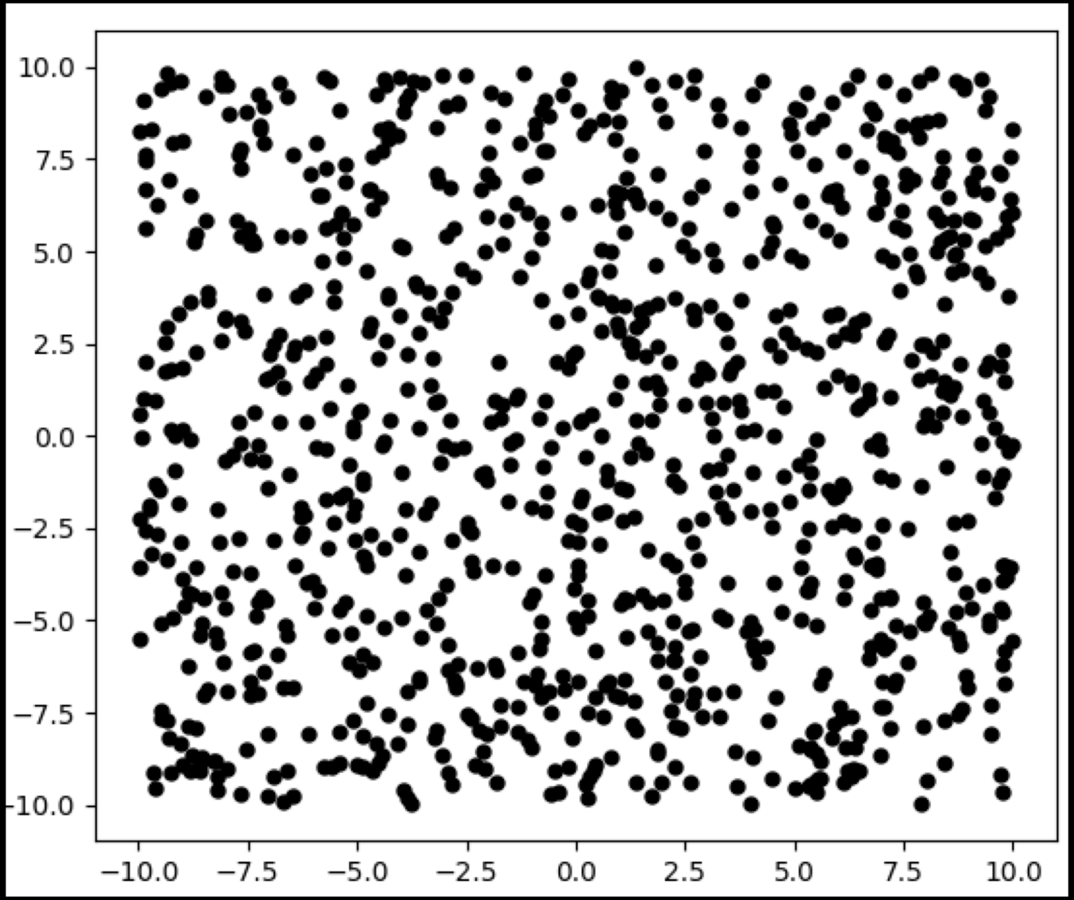
- Fractional factorial design *in conjunction with* other approaches

Assay	Variable levels[a]					Response
	SCM[b]	CSL[c]	YEP[d]	K_2HPO_4	pH	Enzyme activity (U/ml)
1	–1	–1	–1	–1	1	2.27
2	1	–1	–1	–1	–1	1.11
3	–1	1	–1	–1	–1	4.11
4	1	1	–1	–1	1	12.44
5	–1	–1	1	–1	–1	3.59
6	1	–1	1	–1	1	14.68
7	–1	1	1	–1	1	5.56
8	1	1	1	–1	–1	0.20
9	–1	–1	–1	1	–1	2.40
10	1	–1	–1	1	1	5.89
11	–1	1	–1	1	1	5.77
12	1	1	–1	1	–1	0.20
13	–1	–1	1	1	1	5.12
14	1	–1	1	1	–1	0.68
15	–1	1	1	1	–1	4.82
16	1	1	1	1	1	11.92



Random isn't (always) ideal



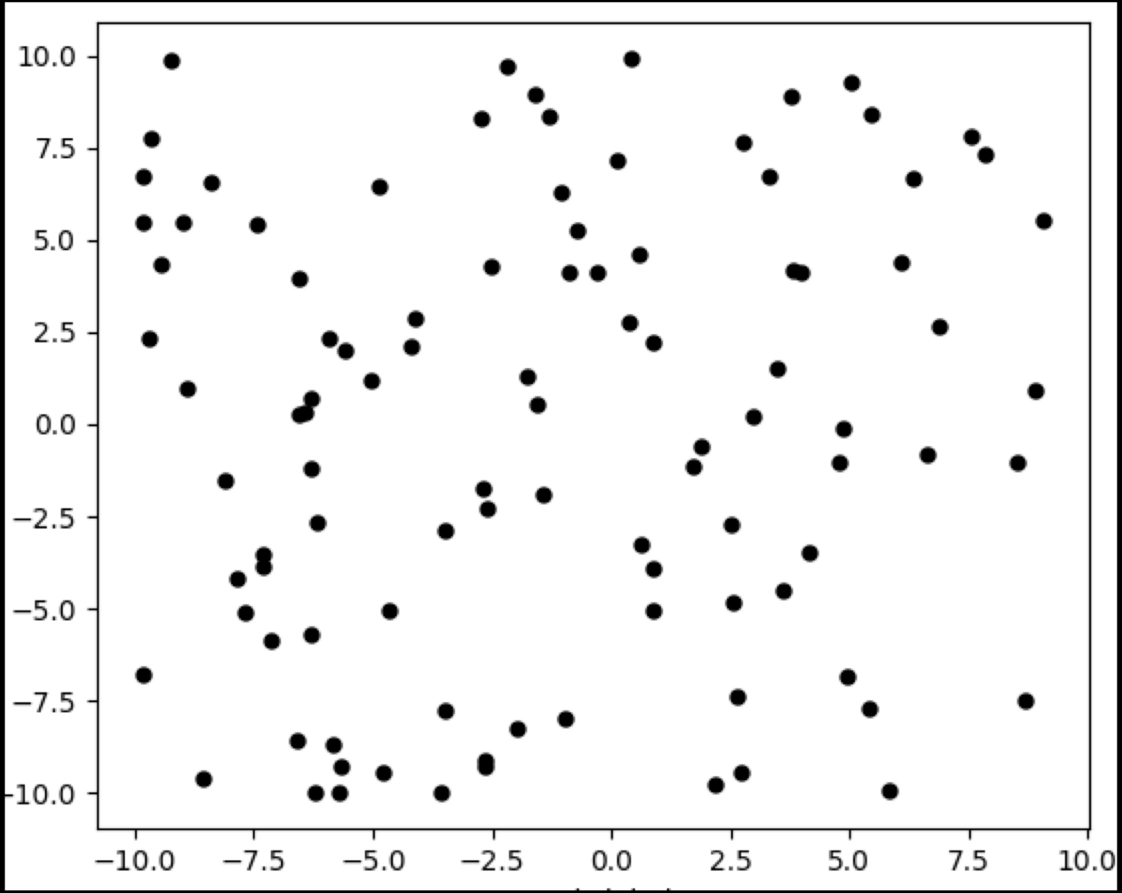


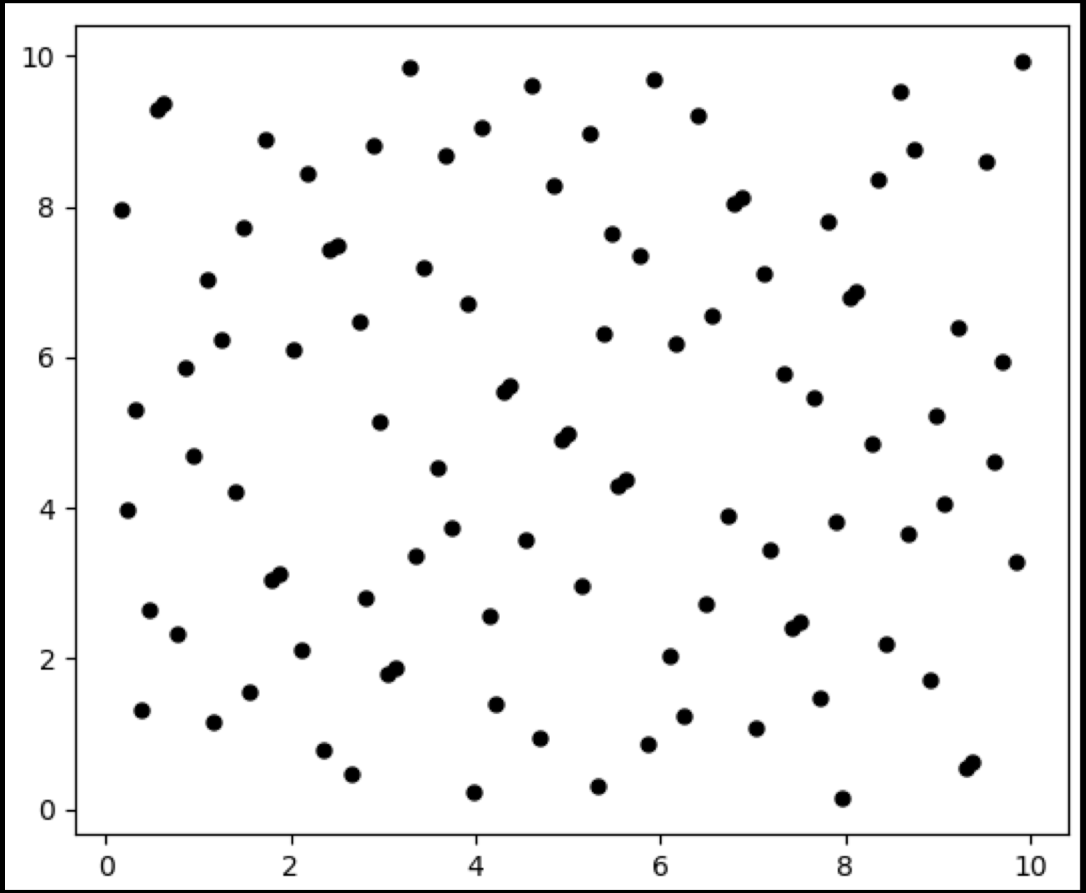
Pause for practicality: real-world implementation

- RuFaS simulation time: 5 minutes
- "Simple" set of *just* 1,000 simulations
- 5,000 minutes = 83.3 hours = ca. 3.5 *days*

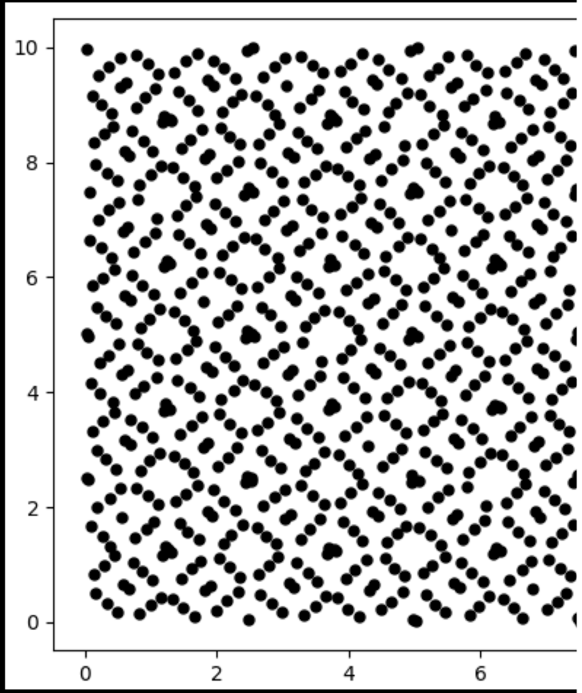
- *Spoiler alert:* we'll need closer to 40,000 simulations for a *single* analysis of a (relatively) narrow set of input variables
  - Almost five months of computing time (i.e., core hours)

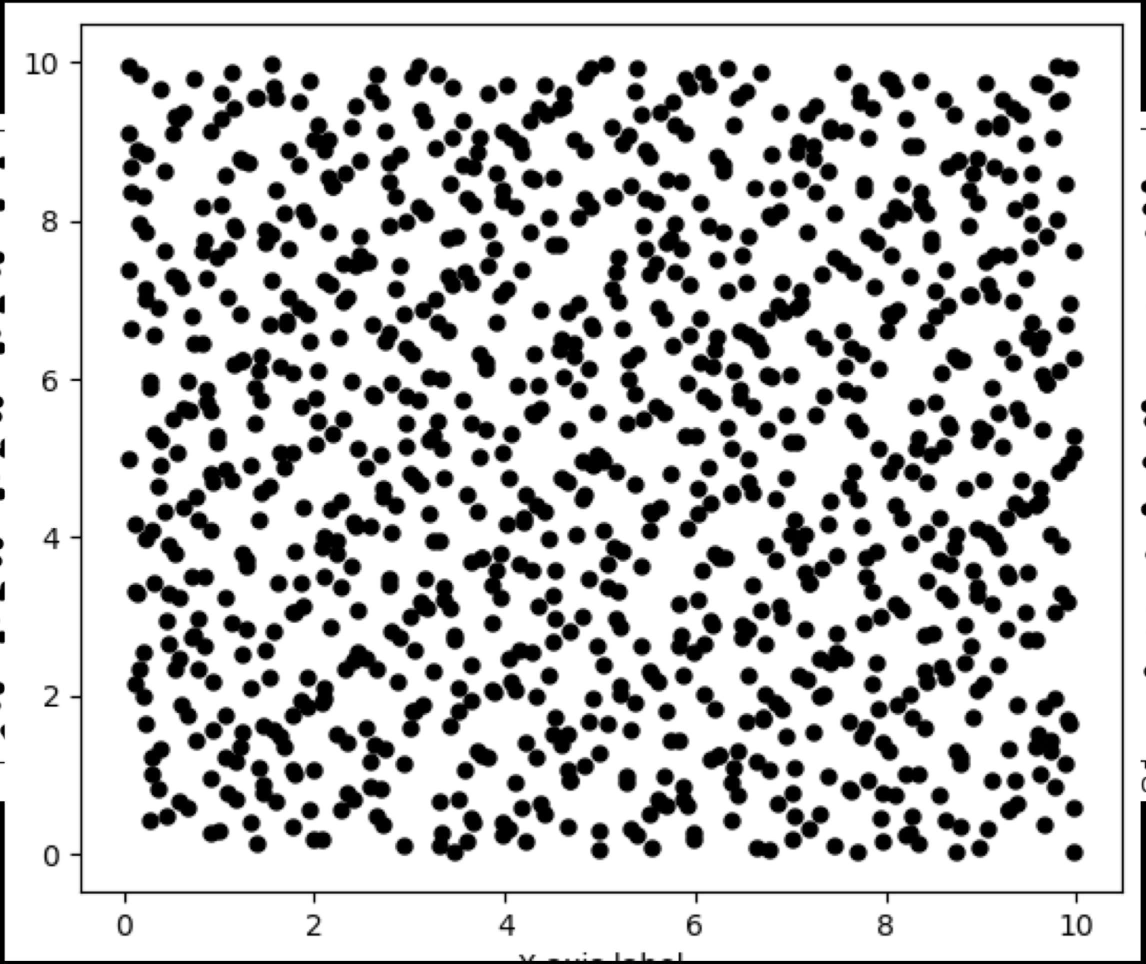
Random sampling v. quasi-random sampling via low-discrepancy sequences

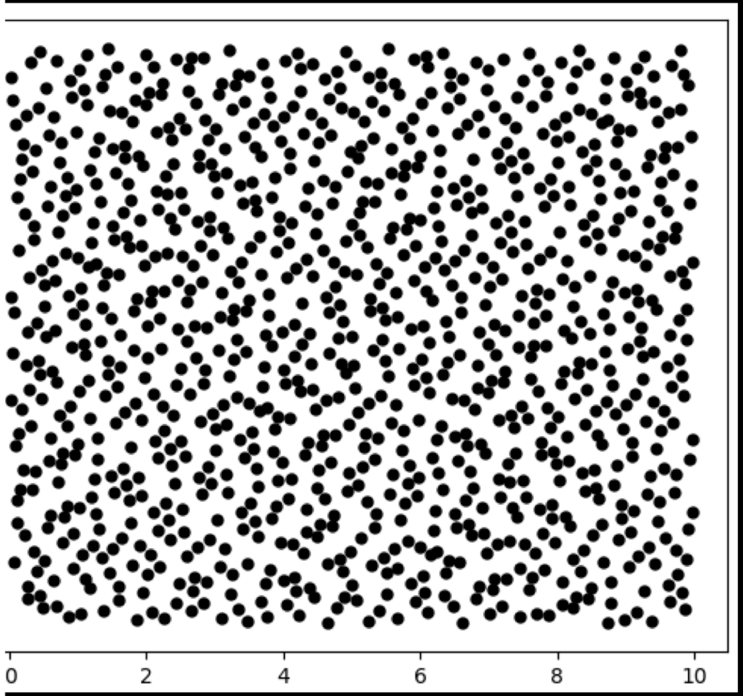




Quasi-random sampling with low-discrepancy sequences







Applying sensitivity analysis methods to RuFaS

- Python implementation: SALib
  - Usher et al 2016
- Decide on a set of variables and their bounds
  - Which variables do we want to evaluate? Over what ranges?
- Vary the inputs: populate necessary JSONs using appropriate sampling sequence
- Run RuFaS x times
- Record the outputs: collect reports
- Evaluate the results: calculate sensitivity indices for ea[ch] variable

```
variable_list_a = [
  ['breeding_start_day_h', 380, 38],
  ['heifer_repro_cull_time', 140, 14],
  ['estrus_detection_rate_h', 0.6, 0.12],
  ['cull_milk_production', 17.0, 8],
  ['cow_times_milked_per_day', 2, 1.5],
  ['wean_day', 60, 60*.25],
  ['avg_estrus_cycle_heifer', 24, 5],
  ['still_birth_rate', 0.065, 0.065*.25],
  ['horizontal_dist_to_milking_parlor', 1.6, .5],
  ['mature_body_weight_avg', 740, 50],
]
```

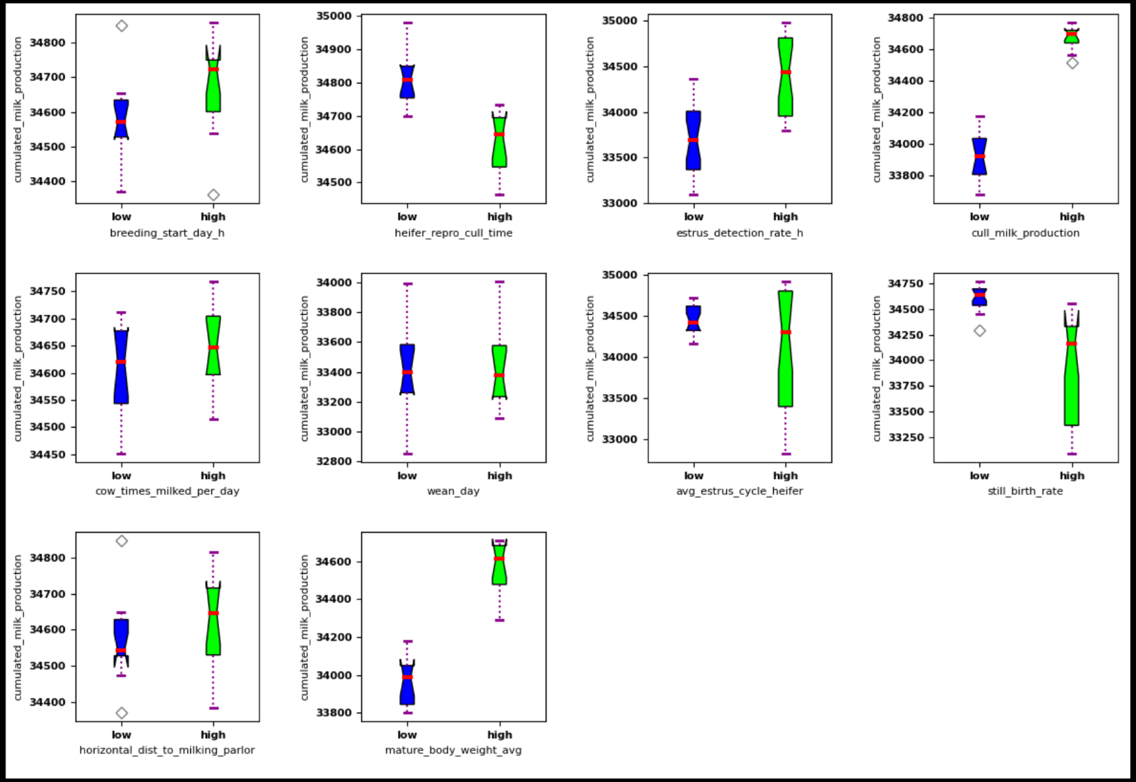
```
> herd_report_00000
> herd_report_00001
> herd_report_00002
> herd_report_00003
> herd_report_00004
> herd_report_00005
> herd_report_00006
> herd_report_00007
> herd_report_00008
> herd_report_00009
```

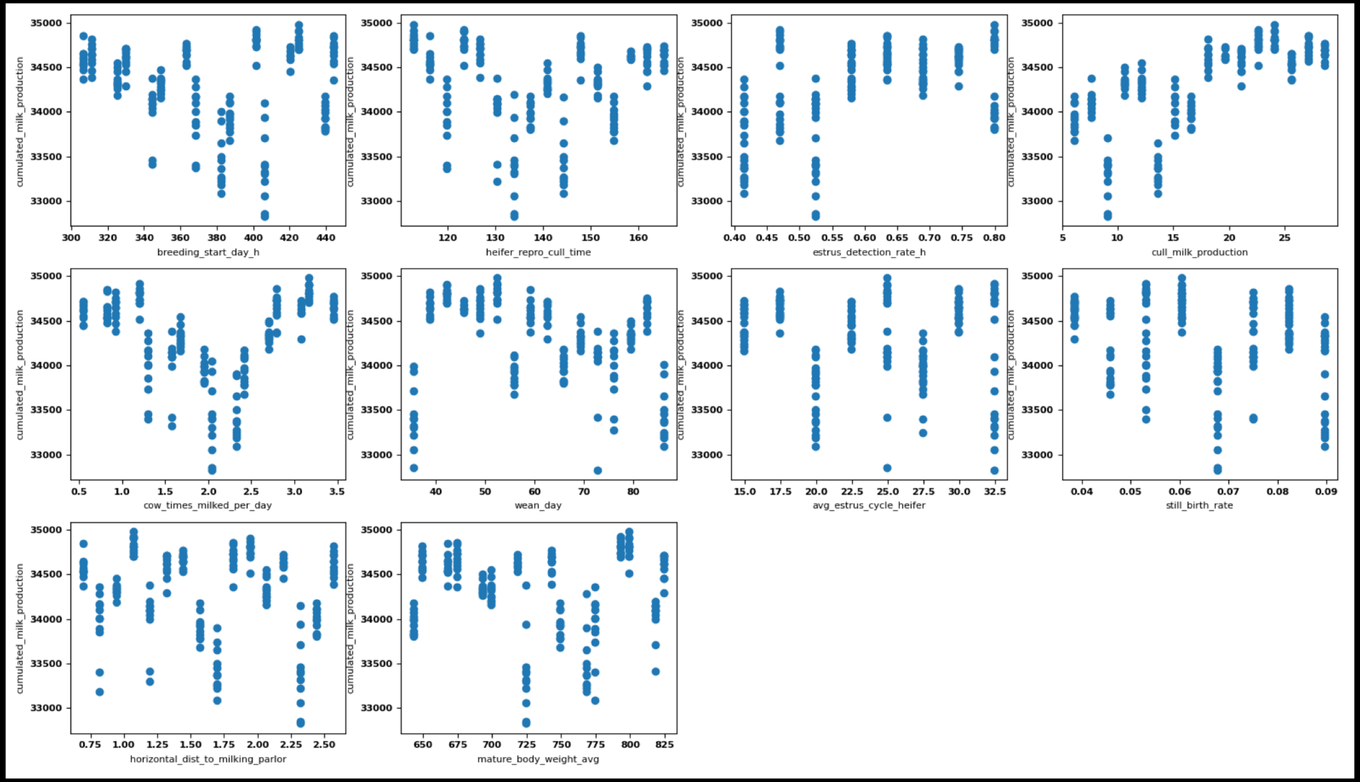
```
>>> param_values[1,:]
array([3.50711426e+02, 1.39176270e+02, 6.80595703e-01, 1.89277344e+01,
       9.94750977e-01, 6.12854004e+01, 2.60861816e+01, 7.59457397e-02,
       1.76296387e+00, 7.89475098e+02, 8.45092773e-01, 1.11206055e-01,
       5.25268555e-01, 3.94897461e-01, 6.62475586e-01, 2.77709961e-01])
>>> param_values[2,:]
array([3.42027832e+02, 1.42525879e+02, 6.80595703e-01, 1.89277344e+01,
       9.94750977e-01, 6.12854004e+01, 2.60861816e+01, 7.59457397e-02,
       1.76296387e+00, 7.89475098e+02, 8.45092773e-01, 1.11206055e-01,
       5.2
>>> param
array([3.4
       9.9
       1.7
       5.2
```

```
sis_cost_feed_heifer.txt
{'names': ['breeding_start_day_h',
           'ME': array([ 0.09
           'interaction_names
           'IE': [12.01642968
```

Input variable	breeding start day h	heifer repro cull time	estrus detection rate h	cull milk productio n	cow times milked per day	wean day	avg estrus cycle heifer	still birth rate	dist to milking parlor	body weight avg
First-order indices	0.0081218	0.0614779	0.0809962	0.0280595	0.0071462	0.0113529	0.0241205	0.0045237	0.0203795	0.029872
Total	0.1855211	0.2133267	0.3581576	0.0611922	0.0935946	0.1113764	0.2242932	0.1351403	0.1873338	0.2006136
Second-order	0.1576839									
indices	0.0712432	0.3932851								
	0.3097194	0.3262099	-0.482161							
	0.1819857	0.3838208	-0.123091	0.2664781						
	0.1072775	0.1774542	-0.215749	0.245391	0.1320809					
	0.0507475	0.1909522	-0.143223	0.0866544	0.1177079	0.037056				
	0.1916498	0.3733005	-0.081864	0.2529797	0.2480618	0.0103796	0.3319189			
	0.0730517	0.276491	-0.042966	0.1005917	0.1012928	0.0364094	0.2830492	0.1344224		
	0.0610377	0.180422	-0.111087	0.0871367	0.0829026	0.04169	0.2175328	0.0581949	0.1692001	

Example outputs from RuFaS





Time needed to evaluate RuFaS

10 variables, with second-order effects (i.e., interactions)

- Fractional factorial design
  - 32 simulations, 2.6 hours to run

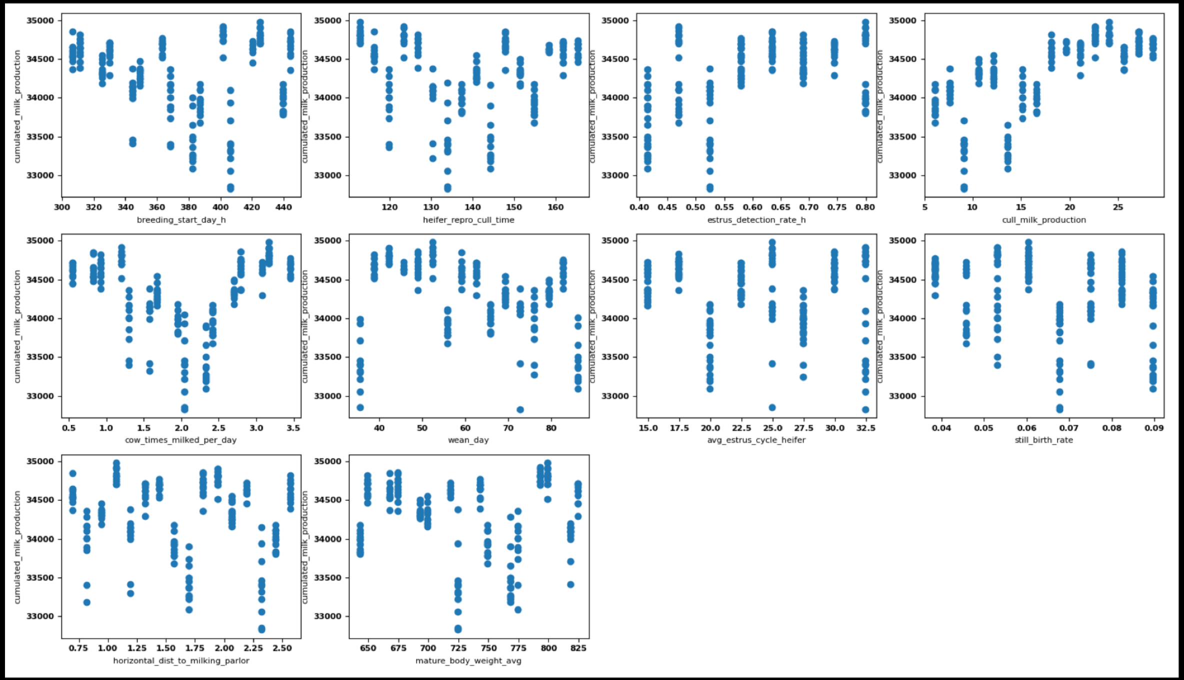
- Saltelli method
  - Approx. 35,000 simulations
  - Approx. 3,000 hours (**120+ days!**)

How can we improve model evaluation times?

Iterated fractional factorial approach

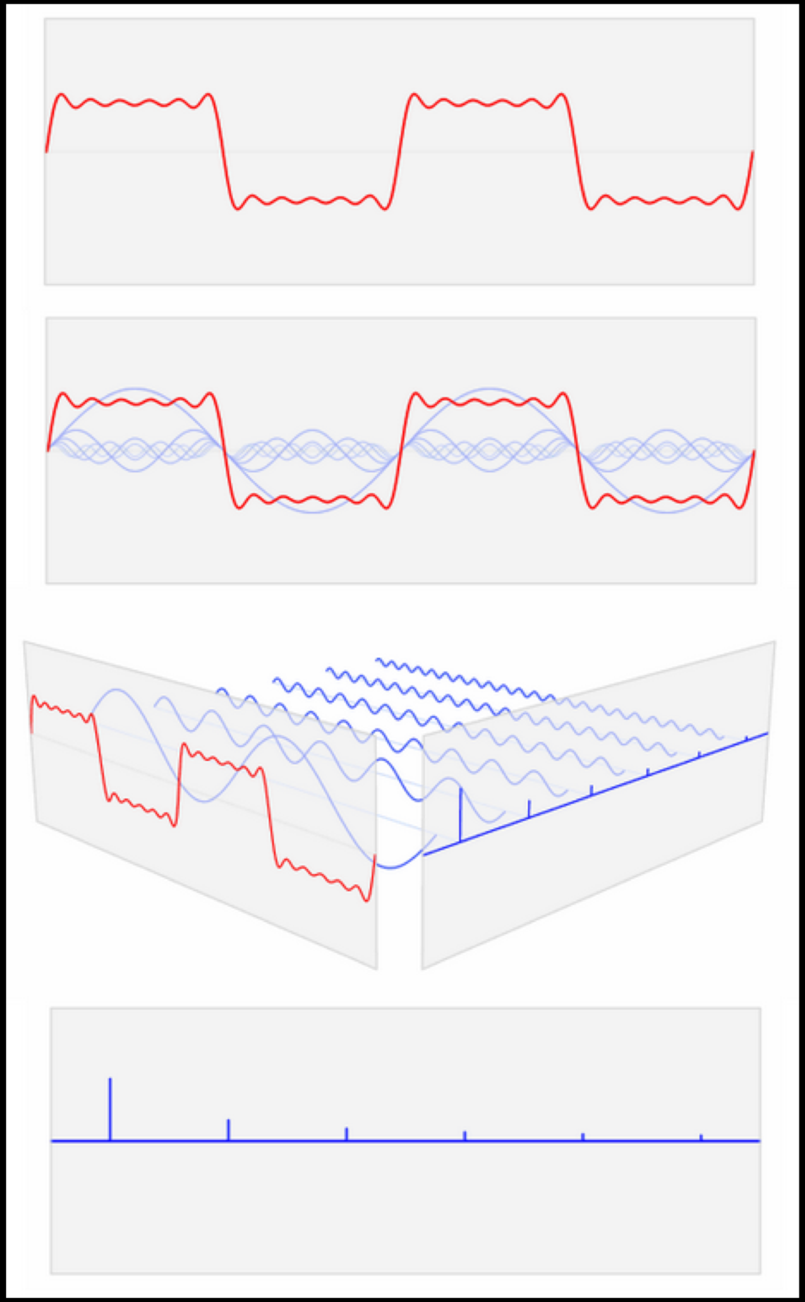
- Define & evaluate major endpoints of interest
- Evaluate ‘slices’ within these ranges
- …repeat

- Andres and Hajas, 1993



Fourier-transform-based approach(es)

- FAST method: Fourier amplitude sensitivity test
  - Saltelli et al 2012, Zhang et al 2015, Wang and Solomantine 2019
- SALib implementation: eFAST (extended Fourier amplitude sensitivity test)



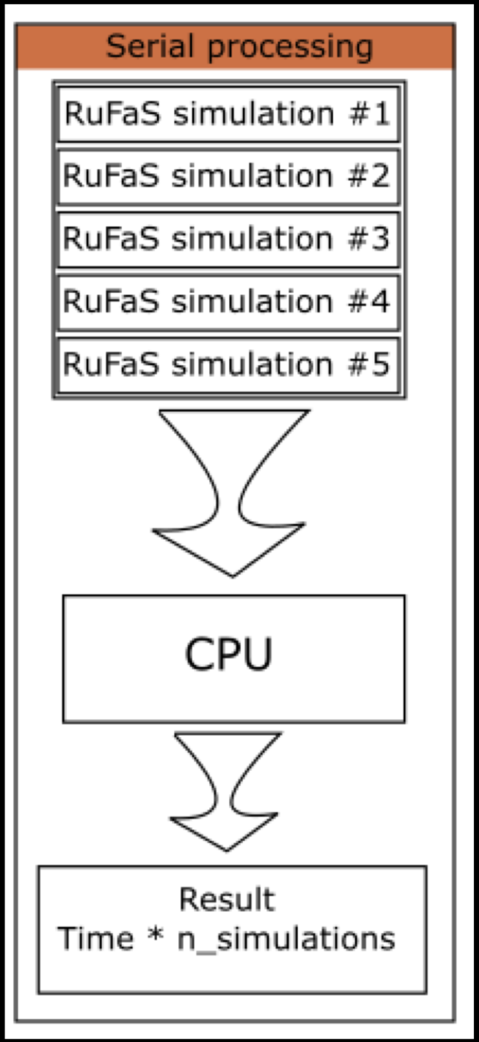
Simulation times for comparable approach(es) to evaluate RuFaS

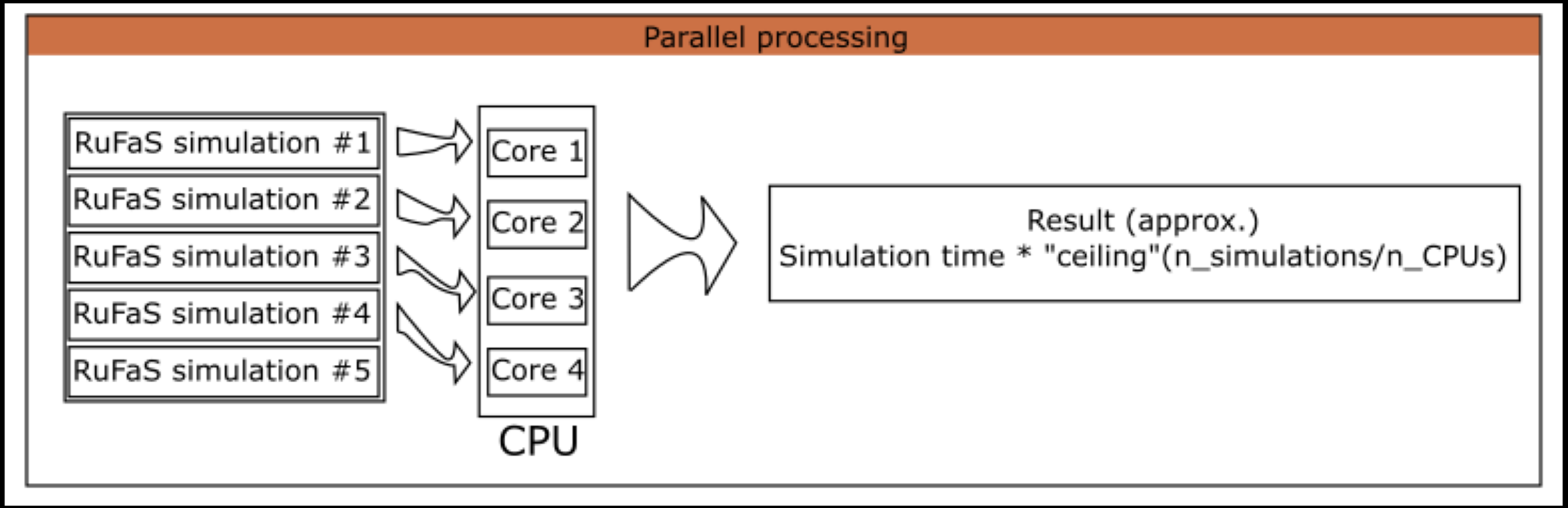
10 variables, with second-order effects (i.e., interactions)

- Saltelli method
  - >35,000 simulations
  - >3,000 hours = >4 months
- eFAST method
  - <17,000 simulations
  - <1,500 hours ("only" approx. 55 days!)

How can we *further* improve model evaluation times?

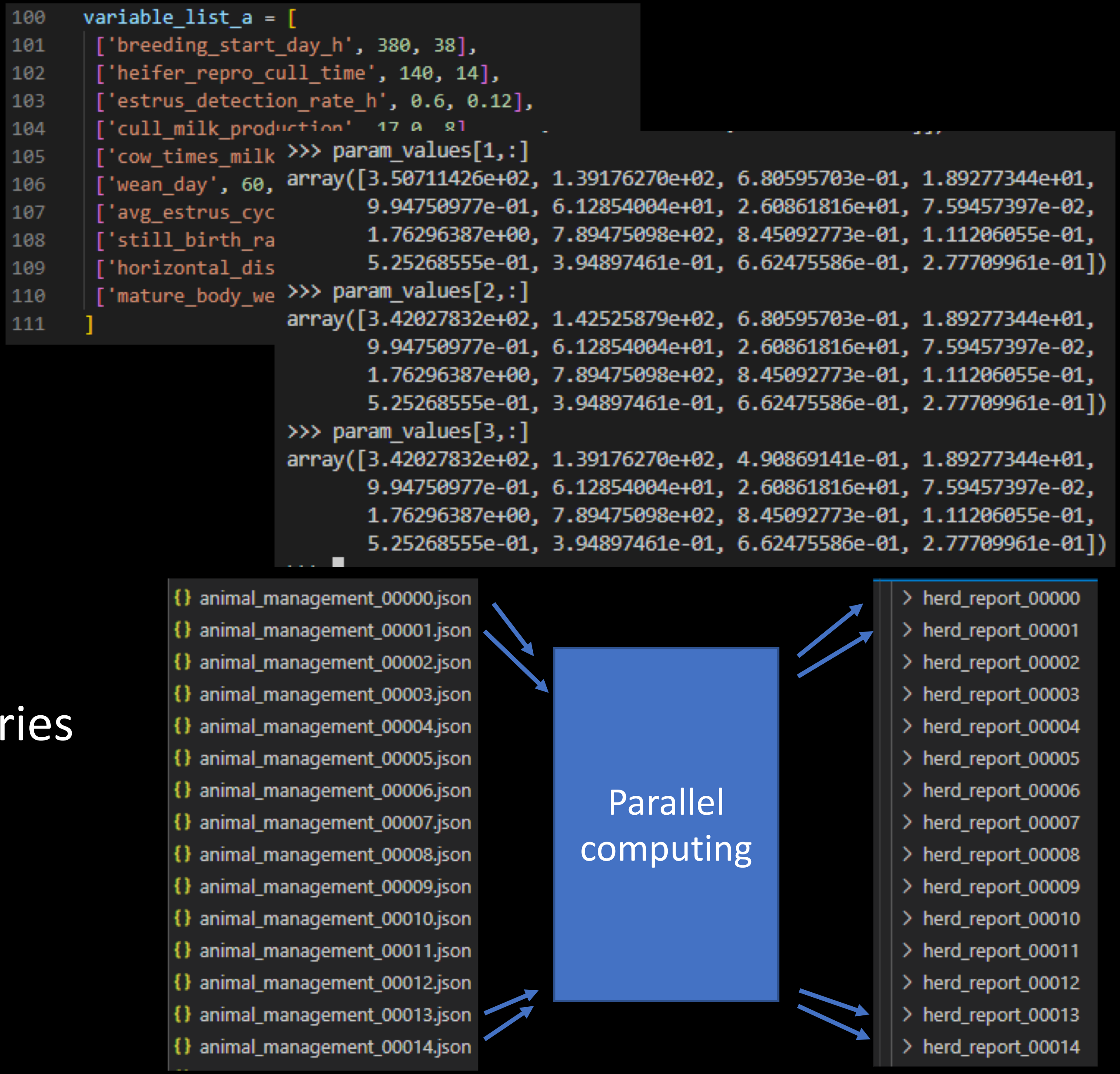
Sequential v. parallel computing





RuFaS parallelized

- Running simultaneous simulations from a pre-generated list of inputs
- General workflow:
  - Decide on input variables of interest
  - Set upper and lower boundaries
  - Generate input sequences/combinations



Simulation times for comparable approach(es) to evaluate RuFaS

10 variables, with second-order effects (i.e., interactions)

- Saltelli method
  > 35.k simulations
  3k hours (4 months) / X cores = approx. 2 weeks on an 8-core system

- eFAST method
  <17k simulations
  < 55 days / X cores = approx. 1 week on an 8-core system
  or, less than a day on a 64-core system

Resources: parallel & cloud computing option

Cornell University
Center for Advanced Computing

- “By hand” across more than one system
- Cornell’s CAC resources
  - Cloud-based Virtual Machines (V

Instances

Instance ID =

Displaying 1 item

	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status
☐	Initialtest_jw2574	-	128.84.8.67	c1.m8	RUFASjw2574	Shelved Offloaded

Ongoing objectives

- Yijing: detailed analysis of the Life Cycle portions of the Animal Module
- My immediate focus: *the rest* of the Animal Module
- Applicable to the entirety of RuFaS