Oak Ridge National Laboratory

APRIL 13, 2026 | ICANS XXV

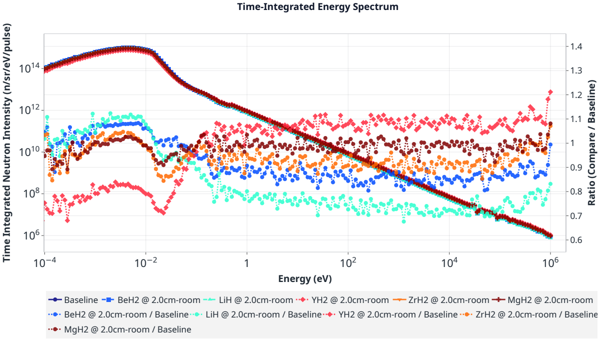
McDakDriver for Optimization and Uncertainty Quantification of Pulsed Neutron Systems

M. R. Altahhan, W. Lu, and F. X. Gallmeier

Spallation Neutron Source (SNS) Neutronics Group, Oak Ridge National Laboratory (ORNL)

Introduction

From material screening to BeH_2 optimization



Companion ICANS XXV study: five solid hydrides were screened against the baseline water pre-moderator.

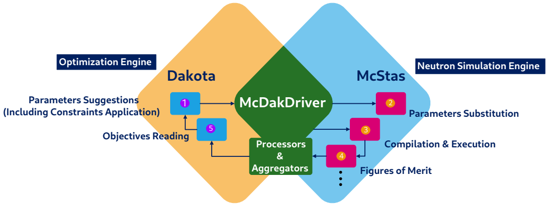
What the first presentation established

Candidates. BeH_2, LiH, YH_2, ZrH_2, and MgH_2 were compared in the same SNS geometry.

Result. BeH_2 performed well among those solid pre-moderators across different thicknesses and temperatures.

Consequence. It became the natural first material for dedicated thickness–temperature optimization.

McDakDriver: reusable optimization and UQ engine



McDakDriver sits between Dakota and the transport solver, making the workflow reusable across MCNP and McStas studies.

Configuration-driven core

Because `config.py` is plain Python, users can embed helpers, call external scripts, or compute derived quantities inline.

Key benefit: switching studies means editing only `config.py`—driver code stays unchanged.

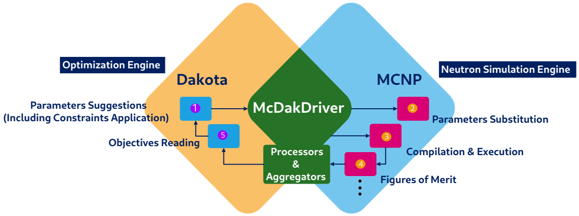
What the framework adds

Automation. Template fill, MPI launch, tally extraction, and response writing in one path.

Reuse. One driver supports both **optimization** and **UQ**.

Traceability. Per-evaluation CSV/JSON logging with failure capture.

McDakDriver for MCNP: template-driven orchestration



MCNP-focused view: Dakota suggestions pass through McDakDriver, while `config.py` processors and aggregators define the returned figure of merit.

Placeholder control

A configurable prefix (e.g. `$$`) inserts Dakota variables into MCNP decks. Optional `pstudy @@@` expressions create derived inputs.

Dependent parameters. Python helpers in `config.py` can generate derived quantities, including on-the-fly ACE data via `tslforge`.

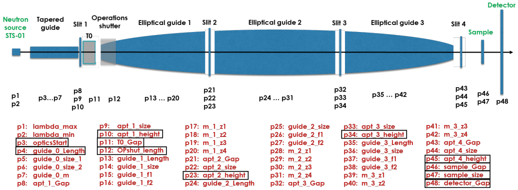
Processors, aggregators, objectives

Processor: callable extracting one scalar QoI from raw tally data.

Aggregator: combines processor outputs into one FOM returned to Dakota.

Objective: one `config.py` entry that binds tallies, processors, and aggregator.

McDakDriver in practice: science-driven beamline optimization



Example: 31-variable SANS beamline optimized with Dakota on 80 compute nodes (128 ranks per node).

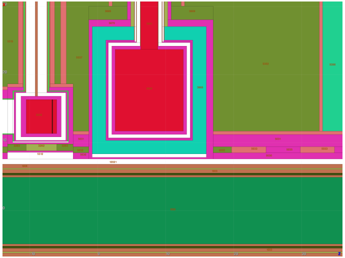
OBJECTIVES in config.py

```
OBJECTIVES = {
  "fom_on": {
    "monitors": [
      {"filename": "Sample_psd.dat",
       "quantity": "I", "weight": 1.0},
      {"filename": "QDetector_1.dat",
       "processor": process_a2_term},
      {"filename": "CorMap_pair_1.dat",
       "processor": cormap_penalty},
      # ... additional monitor / CorMap terms
    ],
    "aggregator": science_on_fom,
    # processors / aggregators are Python functions
  },
  "fom_off": {...}, "resolution": {...}
}
```

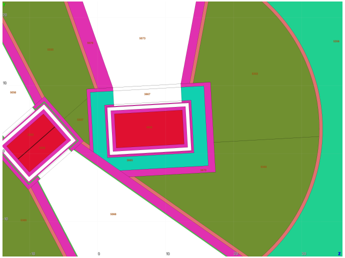
Outcome. Sample intensity **doubled** while maintaining $Q_{\text{min}} \approx 0.05\ \text{Å}^{-1}$. The study used 80-node (128 ranks per node) HPC resources, demonstrating scalable optimization with McDakDriver.

Oak Ridge National Laboratory

SNS coupled hydrogen moderator model and optimization problem



XZ view: cell 3650 is the coupled H_2 moderator; 3665 is the pre-moderator region.



YZ view: the BeH_2 slab in cell 3665 wraps the viewed cold moderator volume.

Study setup

SNS first target station conditions are **1.3 GeV, 2 MW, and 60 Hz**. Cell 3650 contains liquid para-H_2 at **20 K**. The design variables are `premod_thickness` $\in [1.00, 2.499]$ cm and `premod_temp` $\in [100, 450]$ K. Because BeH_2 has no pre-tabulated TSL, temperature-dependent ACE data are generated **on the fly**.

Workflow

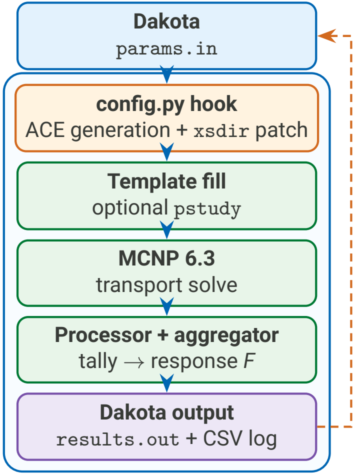
Driver workflow and response construction

What One Evaluation Actually Does

1. Dakota writes `params.in` for the current design point.
2. The `config.py` hook builds BeH_2 ACE data and patches the workdir `xsdir`.
3. McDakDriver fills the MCNP template, with optional `pstudy` preprocessing.
4. MCNP runs, then processors and aggregators reduce tallies to the scalar response F.
5. The driver writes `results.out` and appends the evaluation CSV log.

Reusable vs Study-Specific

Reusable McDakDriver logic. Read parameters, run transport, return responses.
Study logic in `config.py`. ACE generation and tally reduction.



On-the-fly temperature-dependent cross-section pipeline

T **from Dakota optimizer**

Phase 1: thermal scattering

NCrystal VDOS

↓

`ncmat2endf`

↓

NJOY THERMR / ACER

`h-beh.99t` `be-beh.99t`

→

Phase 2: continuous ACE

MODER → RECONR

BROADR → HEATR

GASPR → THERMR

PURR → ACER

`1001.99c` `4009.99c`

→

Phase 3: library integration

Write .99t and .99c entries to `xsdir`

Patch work-directory transport library map

`xsdir_mcnp6.3`

Key point — `tslforge` library

Phase 1 uses `tslforge`: a JSON-config-driven python pipeline that extracts VDOS from ncmat, generates TSL ENDF via `ncmat2endf`, and runs NJOY (THERMR → ACER). Phase 2 Doppler-broadens continuous ACE to the same *T*.

Objective function and design space

Weighted geometric mean of band ratios

$$F = R_{\text{cold}}^{0.80} \times R_{\text{thermal}}^{0.20} \qquad R_b = I_b \,/\, I_b^{\text{ref}}$$

Cold band $E \in [1, 10)$ meV

Thermal band $E \in [10, E_{\text{max}})$ meV with $E_{\text{max}} = 40$ (narrow) or 100 (wide)

Meaning $F = 1$ matches the water premoderator baseline; $F > 1$ improves the objective.

Band-integrated brightness. $I_b = \int_{E_{\text{min}}}^{E_{\text{max}}} \Phi(E)\, dE$ is the band-integrated brightness over band b for the fixed beamline viewport used here.

Baselines & design space

Reference values

$I_{\text{cold}}^{\text{ref}}$	7.9098×10^{12}
$I_{\text{therm,narrow}}^{\text{ref}}$	6.4123×10^{12}
$I_{\text{therm,wide}}^{\text{ref}}$	7.3824×10^{12}
w_{cold}	0.80

Search space

d [cm]	1.00–2.499
T [K]	100–450

Oak Ridge National Laboratory

Dakota EGO configuration and evaluation cost

Efficient Global Optimization setup

Search strategy

Method	`efficient_global`
Surrogate	Gaussian process
Infill metric	Expected improvement

Run settings

Initial samples	10
Max iterations	100
Convergence tol.	1×10^{-3}
Failure penalty	-1×10^{30}

Resources and evaluation cost

Campaign scale

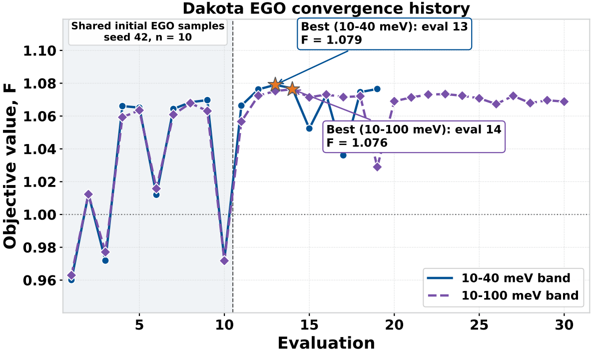
MCNP ranks / eval	480
EGO execution	serial
10–40 meV study	19 evals
10–100 meV study	30 evals

Observed timing (30-eval study)

10–100 meV total	5.29 h
Average / eval	$\sim$10.6 min
XS generation	$\sim$2 min

Results

Convergence to the best BeH_2 design



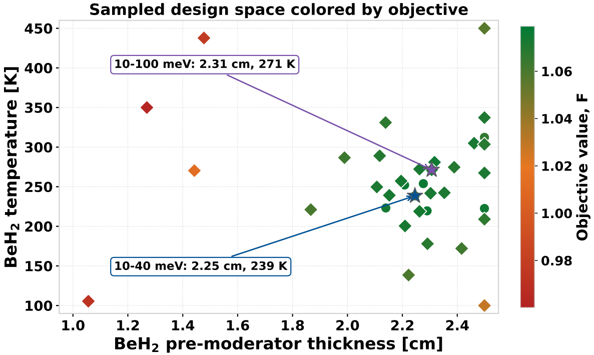
Best configurations

	10–40 meV	10–100 meV
d [cm]	2.246	2.307
T [K]	238.6	271.4
F	1.079	1.076
ΔI_{cold}	**+10.1%**	**+9.6%**
$\Delta I_{thermal}$	**−0.6%**	**+0.1%**

Why evals 1–10 match

Same Dakota seed, bounds, and 10 initial EGO points, so evals 1–10 coincide. The runs split at eval 11 when expected improvement targets the two thermal bands.

Sampled design space and trade-offs

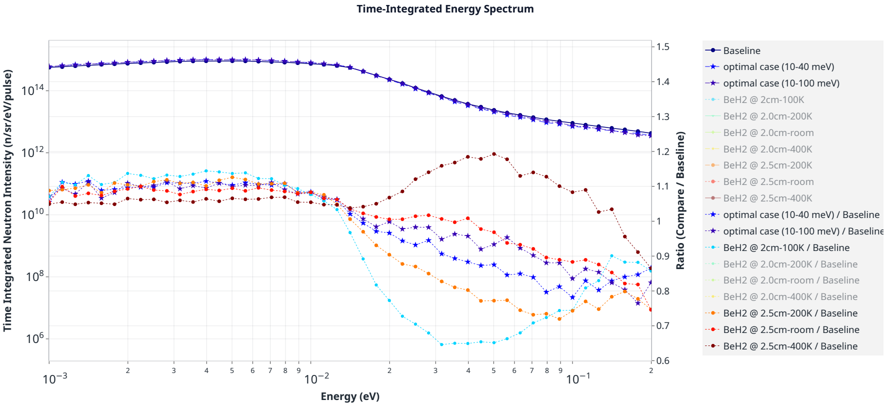


Thermal-band sensitivity

Markers	circles = 10–40 meV; diamonds = 10–100 meV
Shared optimum	both campaigns cluster at $d \approx 2.2$–2.3 cm, $T \approx 220$–280 K
Sensitivity	10–100 meV shifts the optimum in T while F changes $<0.3\%$

Cold-band gains of $\sim$10% are robust to the thermal-band upper bound.

Energy-spectrum comparison of fixed and optimized BeH_2 cases



Discussion. 2.0 cm at 100 K: the cold side improves, but the thermal ratio drops to about $0.6\times$ **baseline**. **Optimized cases:** both optima preserve the $\sim$**10%** cold-gain trend without that thermal collapse. **Takeaway:** the optima stay on the thicker side ($d \approx 2.2$–2.3 cm), while widening the thermal band shifts the optimal temperature, as expected, from $\sim$240 K up to $\sim$270 K.

Summary

Summary and next steps

What the results show

Balanced region. Best BeH_2 designs cluster near $d \approx 2.25$–2.31 **cm** and $T \approx 240$–270 **K**.

Best trade-off. The 10–100 meV optimum keeps thermal brightness nearly at the water baseline while preserving a $\sim$**10%** cold-brightness gain.

Too cold is not better. A 2.0 cm, 100 K case can push cold brightness higher, but thermal brightness falls to about **60% of baseline**.

Why McDakDriver matters

Reusable study logic. McDakDriver changes objectives and processors through `config.py` rather than new driver code.

One coupled workflow. XS generation, MCNP transport, and postprocessing stay in one reproducible optimization loop.

Scales to follow-on work. The same workflow compares baseline, fixed-point references, and optimized designs consistently, and extends naturally to UQ and engineering constraints.

Acknowledgments

This research was supported by the **Spallation Neutron Source**, a DOE Office of Science User Facility operated by **Oak Ridge National Laboratory**.