THE UNIVERSITY of TENNESSEE UT KNOXVILLE

Development of an ORIGEN-based Reactor Analysis Module for Cyclus

Steven E. Skutnik
Assistant Professor
Department of Nuclear Engineering

Cyclus progress update meeting @ ANL
10/23/2014

Overview of topics

- ORIGEN-based Nuclear Fuel Inventory Module (NFIM)
  - Background, goals, and design approach
- ORIGEN design improvements for fuel cycle options analysis
  - Descriptive reactor data library formats
  - Generalized cross-section interpolation
  - ORIGEN "builder's API"
- Progress and future outlook

Why go with physics based-approaches?

- Many fuel cycle simulator tools (e.g., VISION, DANESS, et. al.) rely on pre-calculated fuel “recipes” for nuclide tracking
- **Problems with recipes:**
  - Difficult to perturb for fuel cycle operating conditions
  - Cannot reasonably capture effects from material recycle; <u>key to fuel cycle options analysis</u>

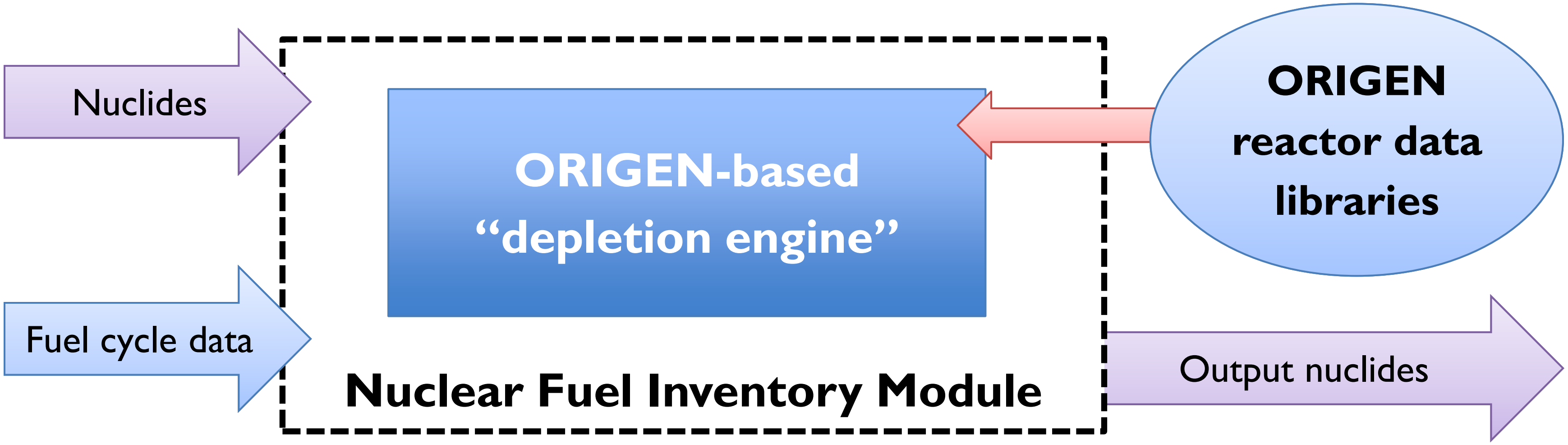
Why incorporate ORIGEN for reactor analysis?

- **Validated capabilities**
  - Extensive radiochemical assay benchmark data
- **Flexibility**
  - Most fuel cycle operations supported
  - Able to accommodate new fuel & reactor types based on TRITON transport calculations
- **Scope**
  - ORIGEN tracks over 2000 individual nuclides; essential for capturing a broad range of timescales

How do we incorporate ORIGEN into Cyclus?

- Newest ORIGEN API facilitates direct, in-memory calls to ORIGEN solver
- By developing a portable, embeddable "**depletion engine**," ORIGEN operations can be "wrapped" into a Cyclus-friendly format
  - "Depletion engine" builds from ORIGEN API
  - **Portability:** a buildable API toolkit which can exist outside of SCALE

How does the NFIM "fit in" to Cyclus?



Nuclides

Fuel cycle data

ORIGEN-based "depletion engine"

Nuclear Fuel Inventory Module

ORIGEN reactor data libraries

Output nuclides

The depletion engine must be portable

- Depletion engine must encapsulate all required ORIGEN functionality without dependence on outside SCALE modules
- Implies **folding in** numerous SCALE operations modules / capabilities
  - Cross-section library interpolation (ARP)
  - Output processing (OPUS)

The depletion engine must be self-sufficient

- Resources used by the depletion engine (i.e., reactor data libraries) should be self-contained, having no outside dependencies
- Implies a **new approach** to data libraries
  - Self-describing: Reactor data libraries have sufficient information to describe all necessary interpolation operations
  - Self-interpolating: Reactor data libraries capable of generating **problem-specific** data

Proposed capabilities of the NFIM

- Generate a Cyclus-style vector of depleted nuclides, given:
  - Input Cyclus nuclide vector
  - Basic fuel cycle data
- Decay of a given Cyclus-style nuclide vector
- Continuous feed/removal of nuclides
- Stream batching & batch separations

Note this supports **more** than just reactor archetypes!

ORIGEN design improvements for fuel cycle options analysis

Infrastructure enhancements to ORIGEN designed to support the development of the Nuclear Fuel Inventory Module

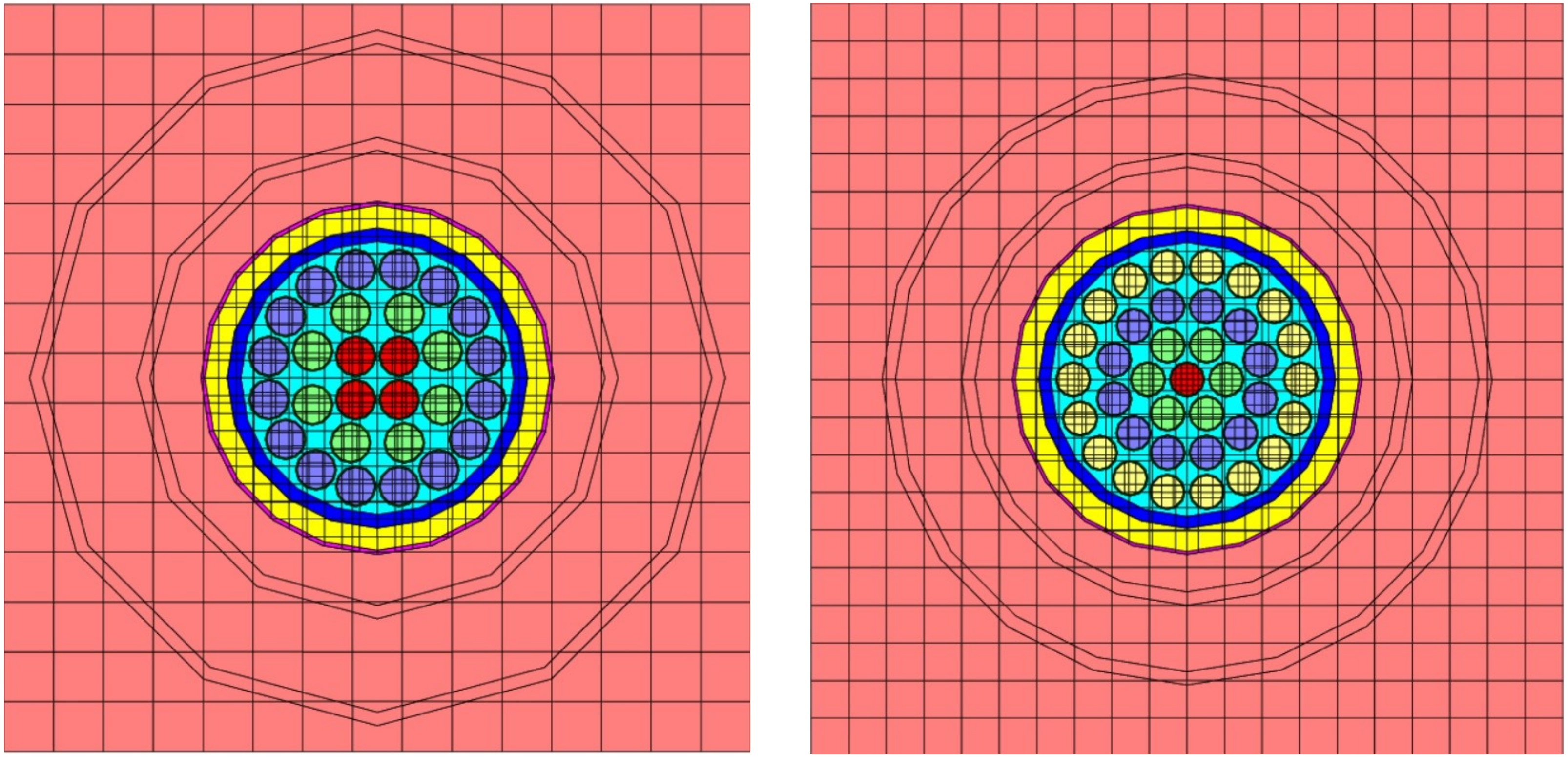
ORIGEN infrastructure development for NFIM

- ORIGEN reactor data library modernization
- Development of a new ORIGEN data library format
- Generalized ORIGEN data library interpolation capabilities
- Depletion engine "builder's API"

Modernized ORIGEN reactor data libraries

Assembly type	Lattice types
PWRs	
Babcock & Wilcox	**15x15**
Westinghouse	14x14, 15x15, 17x17, 17x17-OFA
Combustion Engineering	14x14, 15x15
Siemens	14x14
BWRs	
Atrium	9x9-9, 10x10-9
General Electric	7x7-0, **8x8-1**, **8x8-2**, **9x9-2**, 10x10-8
SVEA	64(8x8-1), 96(10x10-4), 100(10x10-0)
Others	
AGR	
CANDU	**28-pin, 37-pin**
RBMK	
VVER-400	Flat, radial enrichments (3.82, 4.25, 4.38)
VVER-1000	Flat enrichment

CANDU assembly models developed



Why do we need generalized interpolation?

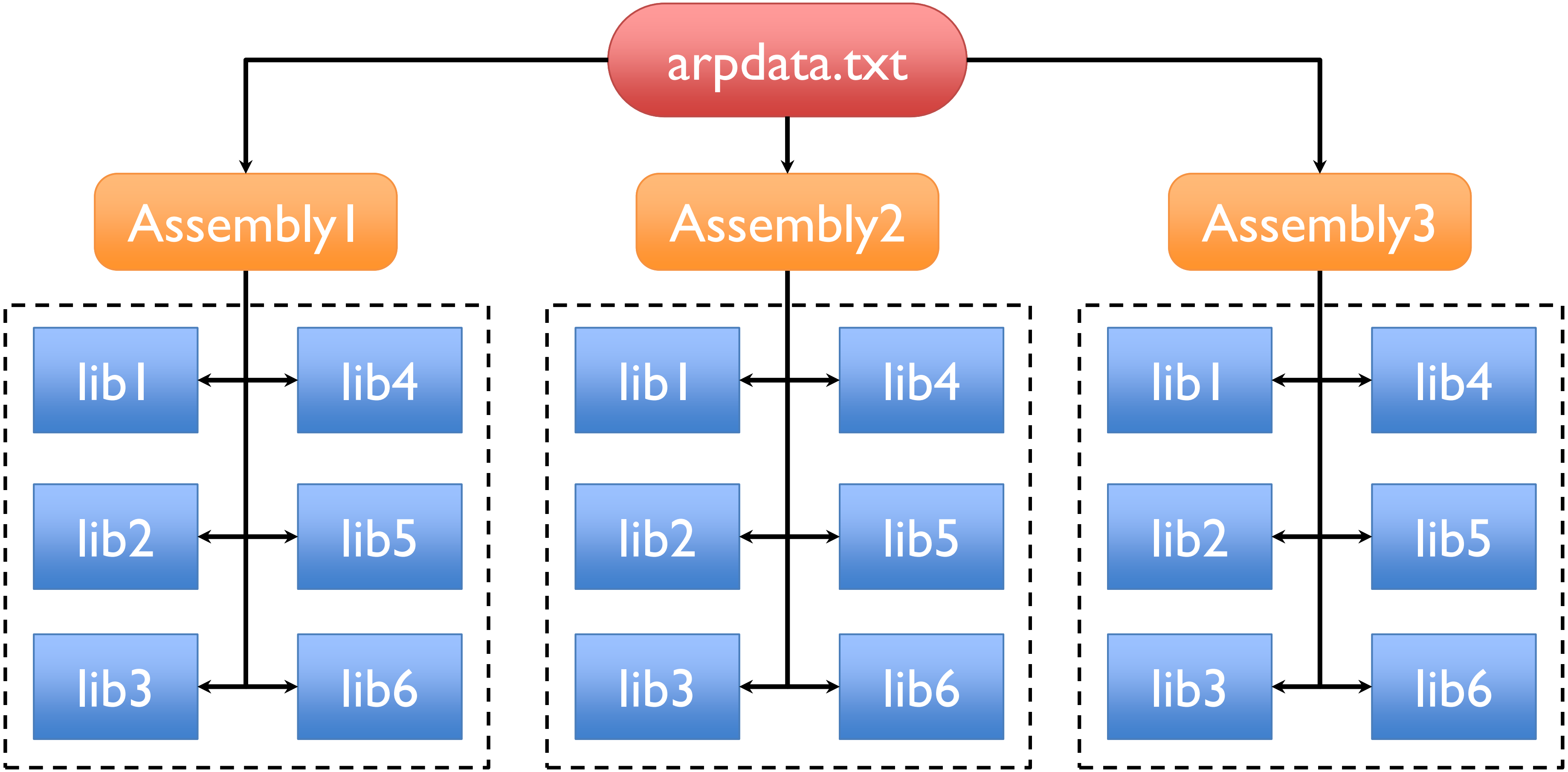
Portability

- Interpolation of problem-dependent reactor data libraries needs to happen “inline” with ORIGEN

Flexibility

- A more **general** approach to interpolation is warranted
- Allows for consideration of multiple reactor operating dimensions
- Offers a path forward for future fuel type interpolation

“Old” ORIGEN library format



arpdata.txt

Assembly1

lib1

lib2

lib3

lib4

lib5

lib6

Assembly2

lib1

lib2

lib3

lib4

lib5

lib6

Assembly3

lib1

lib2

lib3

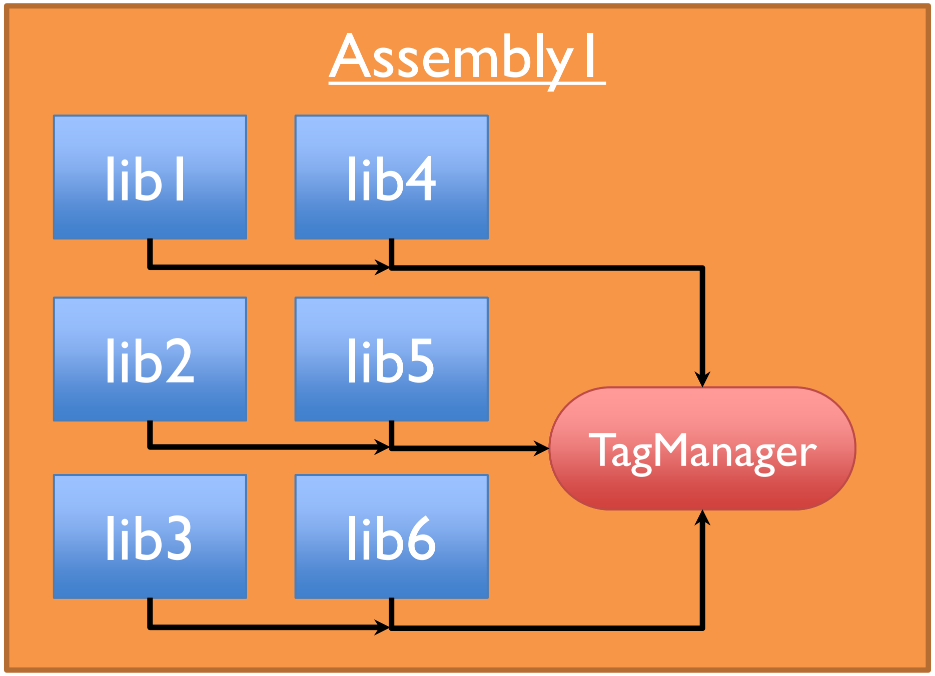
lib4

lib5

lib6

Problem: Serious portability issues!

New "self-describing" ORIGEN library format



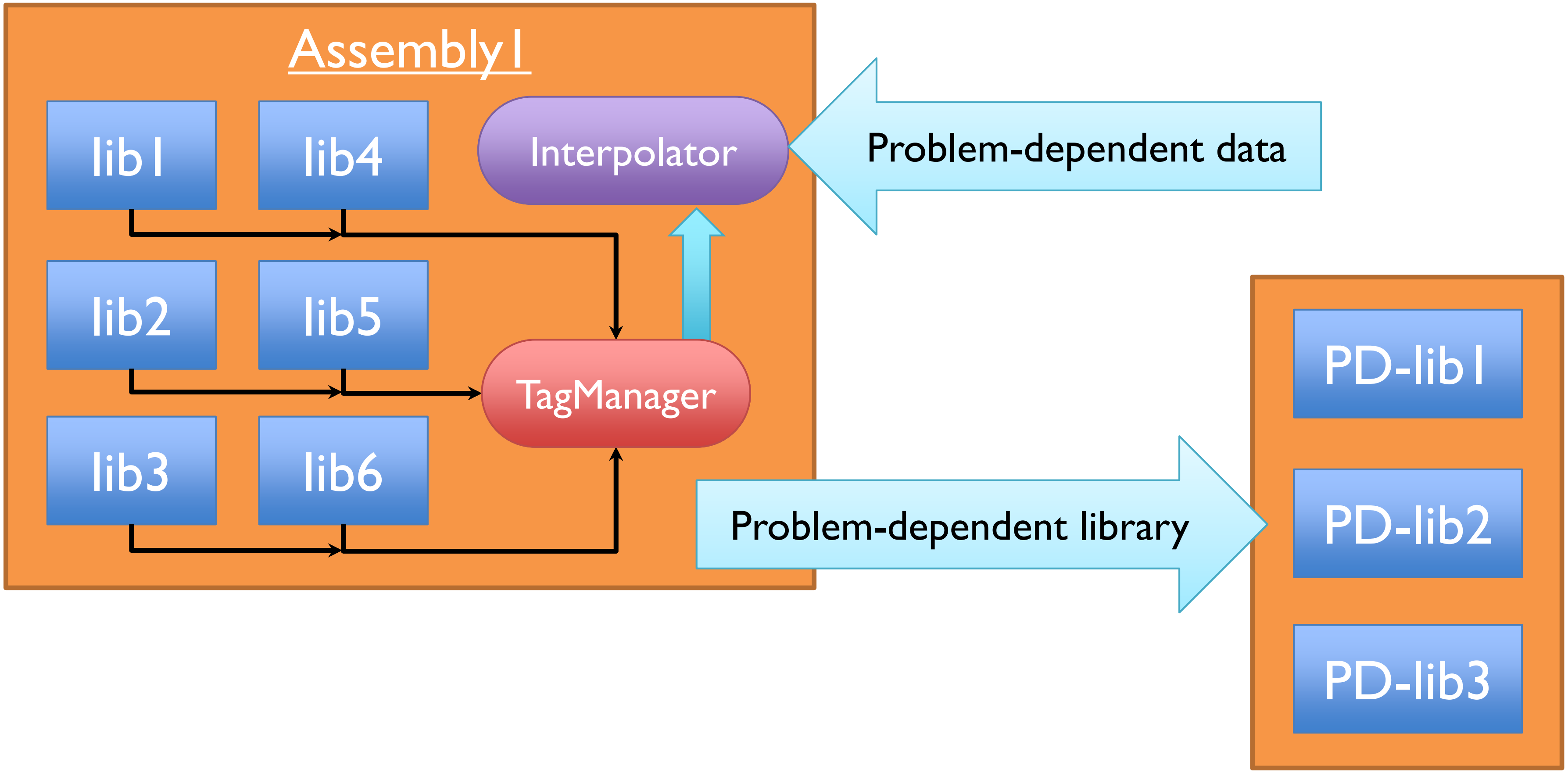
TagManager type identifies two types of information:

- **"Descriptive"** tags: Strings that describe discrete libraries (e.g., assembly type, etc.)
- **"Interpolable"** tags: Floating point data used for CX interpolation (e.g., enrichment, burnup, void fraction, etc.)

Consolidation of libraries from common assembly types into binary "archives"

Key design goals: self-sufficiency and portability

“Self-describing” libraries become “self-interpolating”



Self-contained problem-dependent interpolation capability

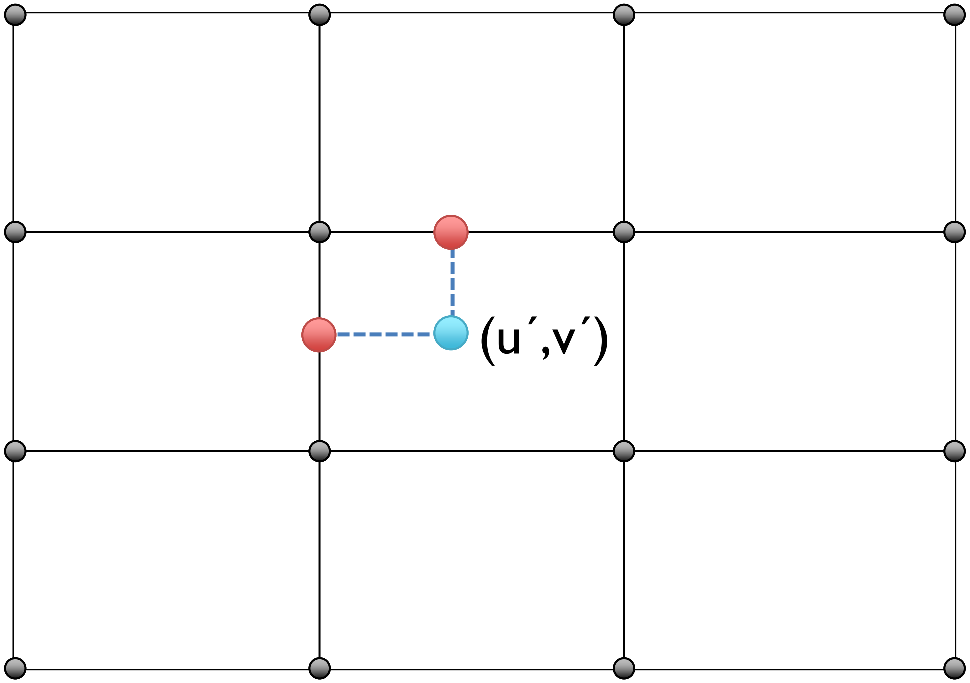
SCALE interpolation resources

- Developing a centralized SCALE interpolation resource API to be incorporated into the "depletion engine"
- Two approaches:
  - Trilinos-based interpolation
    - Analytical Numerical Algorithms (ANAs) developed by Sandia National Laboratory
    - **Advantages:** Highly optimized; future ability to propagate cross-section uncertainties through interpolation
  - "In-house" interpolation algorithms
    - **Advantages:** Allows for optional dependence on outside packages

N-dimensional interpolation approach

$$\overline{\overline{A}}(u',v') = \sum_i c_i(u') \sum_j c_j(v') \overline{\overline{A}}^{S}_{(i,j)}$$

1. Determine adjacent 1-D "knots" for each dimension
2. Determine independent weight factors for each dimension
3. Apply appropriate weights to cross-section data across each dimension



Depletion engine "builder's library" / API

- **Basis** of the NFIM; drives considerations of portability & self-containment
- A self-contained, portable resource other applications can build against directly to incorporate ORIGEN functionality
- Goal is a **distributable** ORIGEN-based depletion resource useful for supporting non-SCALE applications – like the NFIM

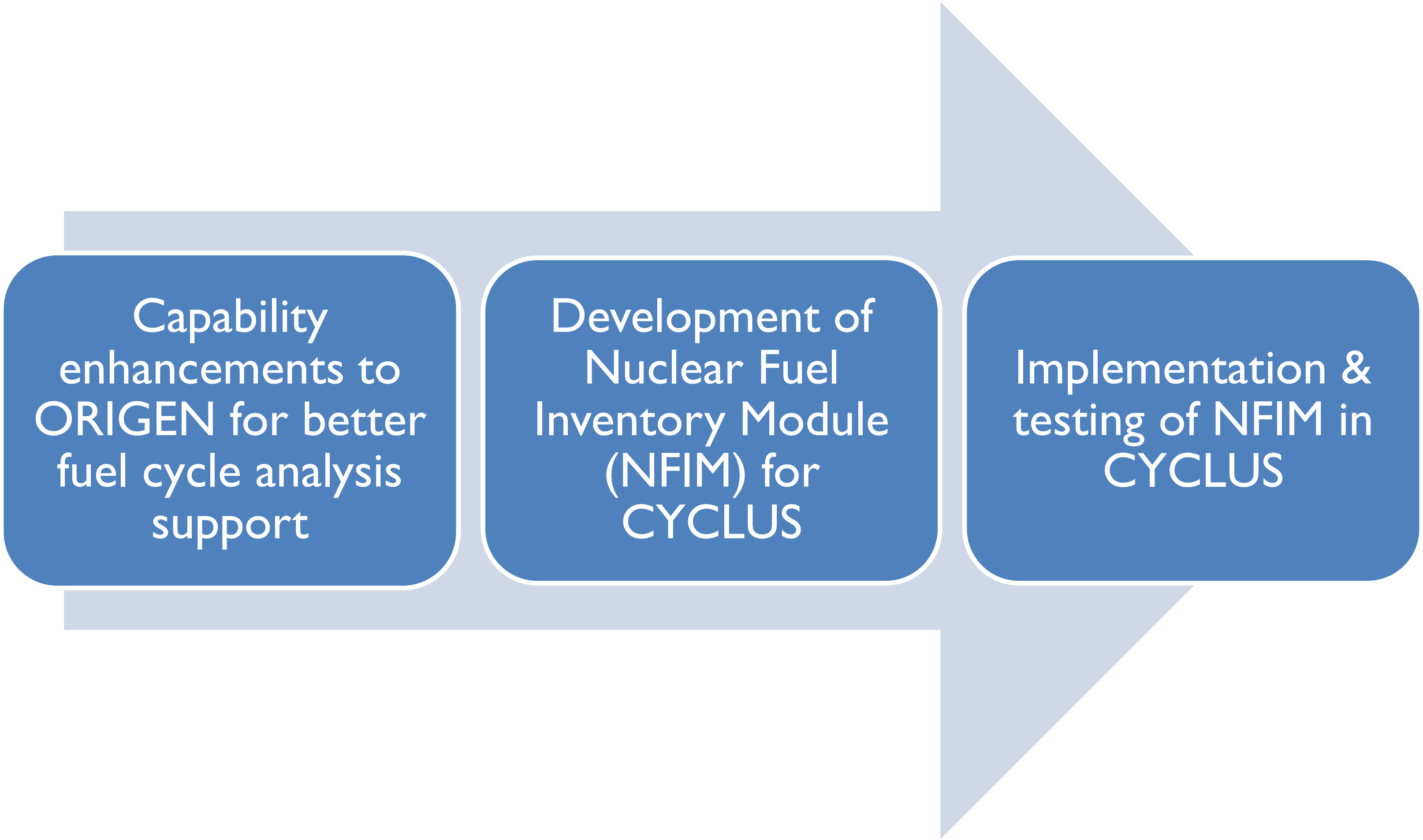
Design requirements of "builder's library"

1. Downloadable directly from RSICC website
2. Includes appropriate API documentation while not disclosing source code
3. Installable resource – self-extracting archive
4. Patchable via downloadable updates
5. Includes relevant ORIGEN data
    - ORIGEN reactor data libraries
    - Other cross-section & nuclear data

Development path for the NFIM

Upcoming tasks to support the development of a suite of ORIGEN-based Cyclus facility modules

NFIM development roadmap



Capability enhancements to ORIGEN for better fuel cycle analysis support

Development of Nuclear Fuel Inventory Module (NFIM) for CYCLUS

Implementation & testing of NFIM in CYCLUS

Schedule of NFIM development tasks

- **FY2015**
  - **Q1:** Complete & implement generalized interpolation
  - **Q2:** Finalize "depletion engine" builder's library / API
  - **Q4:** Develop Cyclus NFIM archetype
- **FY2016**
  - NFIM testing & benchmarking
  - Advanced reactor data library development

Task status: Generalized interpolation

- Currently in final testing
- **Remaining tasks**
  - Implement inline interpolation calls within ORIGEN reactor data library format
  - Development of user input / control over interpolation in ORIGEN input

Task status: Depletion engine “library” / API

- Final requirements document being drafted
- Coordinating with SCALE development team to develop appropriate build
- Contingent upon finalization of generalized interpolation

Task status: NFIM Cyclus archetype

- Initial archetype design has begun
- ORIGEN-based functionality contingent upon "depletion engine"
- Other functions (i.e., Cyclus-facing interfaces) can begin immediately

Acknowledgements

UT/ORNL team

- **UT**
  - Jennifer Littell (NFIM development)
  - Nicholas Sly (Interpolation resources)
  - Scott Richards (ORIGEN development)
  - Nathan Shoman (Reactor library development)
- **ORNL**
  - Will Wieselquist (ORIGEN technical lead)
  - Andy Worrall (Fuel cycle analysis)

Questions?

sskutnik@utk.edu