IDAES® Institute for the Design of Advanced Energy Systems

IDAES Diagnostics Toolbox

Andrew Lee[a,b], Robby Parker[c], Alex Dowling[d], Sarah Poon[e], Dan Gunter[e], Michael Bynum[f], Bethany Nicholson[f]

[a] National Energy Technology Laboratory, [b] NETL Support Contractor, [b] Los Alamos National Laboratory, [c] University of Notre Dame, [d] Lawrence Berkely National Laboratory, [e] Sandia National Laboratories

Writing Good Models is Not Easy

Bad models are easy to write.

Poor quality models cost time and effort due to:

- Errors and debugging
- Fragile and non-reproduceable results
- Limited robustness
- Hard to reuse

How Can We Encourage Good Model Writing?

Provide tools to assist users with tools to help identify and resolve modelling issues.

Interviewed expert users to understand their modeling and debugging workflows and the tools they used.

We Cannot Fix It For You…

Fixing issues requires engineering knowledge.

… but We Can Tell You What is Wrong.

Tools to automatically identify many common issues.

Now Available

v2.2.0

Expert Assistance on Demand

Have you even wished you had the IDAES and Pyomo teams on call to help resolve your modeling issues? The new Diagnostics Toolbox puts these team's combined expertise at your fingertips through an easy-to-use interface.

The Diagnostics Toolbox provides:

- automated checking for a wide range of issues
- easy to read summaries of issues found
- automated recommendations for next steps to take

Some examples of issue we can help solve:

- **Potentially infeasible** (bounds violations, poor scaling, singularities)
- **Evaluation errors** (AMPL evaluation errors)
- **Poor convergence** (poor scaling, degeneracies)
- **Incorrect answers** (unit inconsistency, degeneracies)

Easy to Use

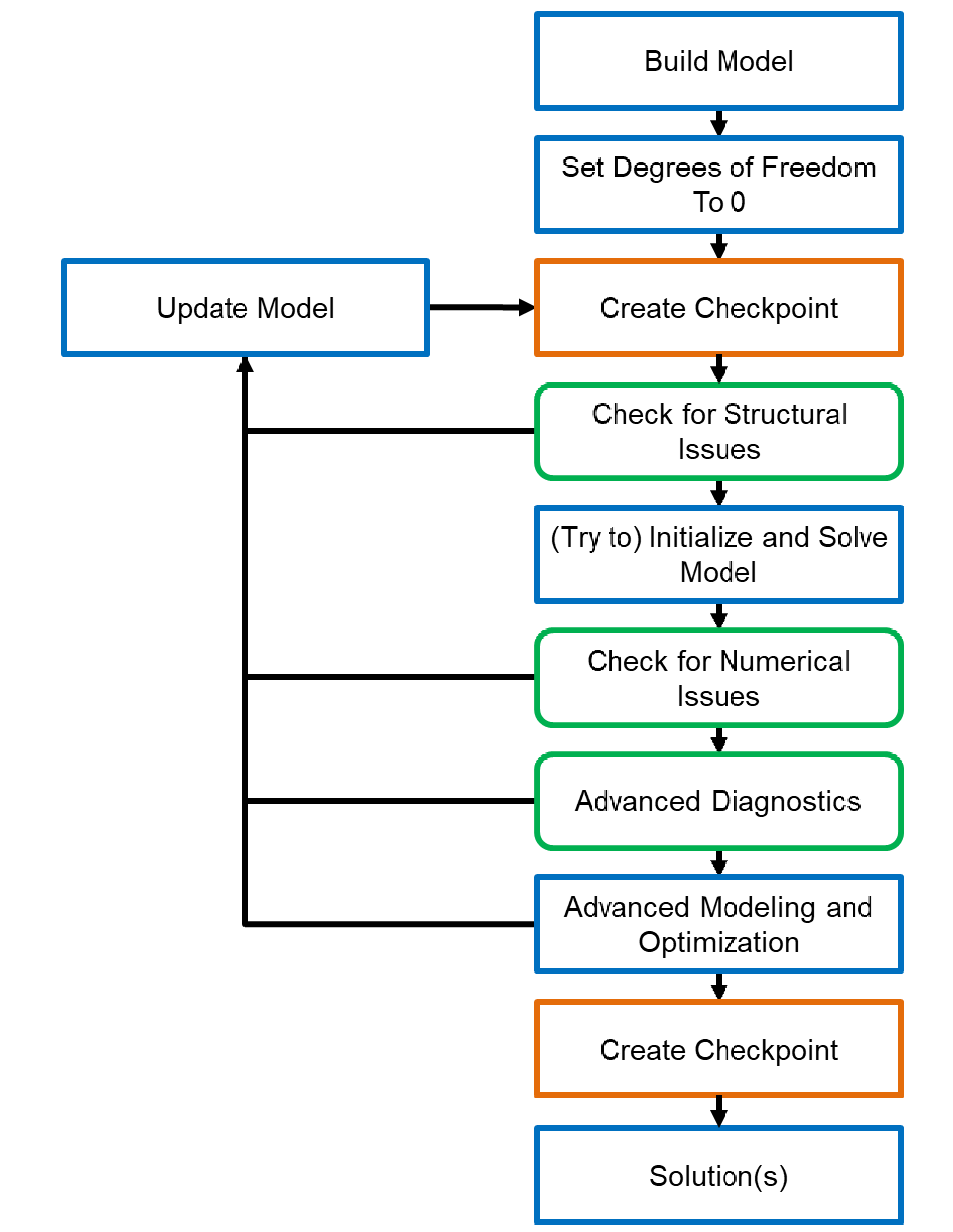
```
from idaes.core.util import DiagnosticsToolbox
dt = DiagnosticsToolbox(m)
dt.report_structural_issues()
```

Proven Value

IDAES and Pyomo teams are already successfully using the toolbox

- New Mexico SBIR: fluidized bed reactor application
- PrOMMiS: scaling issues with trace component concentrations
- IDAES: improving robustness of problematic flowsheets

Diagnostics Workflow


Build Model
Set Degrees of Freedom To 0
Update Model
Create Checkpoint
Check for Structural Issues
(Try to) Initialize and Solve Model
Check for Numerical Issues
Advanced Diagnostics
Advanced Modeling and Optimization
Create Checkpoint
Solution(s)

Can You Find All 8 Issues?

8 Variables:

v_1 - [m]
v_2 - [m]
v_3 - $0 \leq v_3 \leq 5$
v_4
v_5 - $0 \leq v_5 \leq 1$
v_6
v_7 - [m], $0\ \mathrm{m} \leq v_7 \leq 1\mathrm{m}$
v_8

4 Constraints:

c1: $v_1 + v_2 = 10$
c2: $v_3 = v_4 + v_5$
c3: $2v_3 = 3v_4 + 4v_5 + v_6$
c4: $v_7 = 1 \times 10^{-8}\, v_1$

3 Fixed Variables:

$v_4 = 2, v_5 = 2, v_6 = 0$

```
import pyomo.environ as pyo

m = pyo.ConcreteModel()

m.v1 = pyo.Var(units=pyo.units.m)
m.v2 = pyo.Var(units=pyo.units.m)
m.v3 = pyo.Var(bounds=(0, 5))
m.v4 = pyo.Var()
m.v5 = pyo.Var(bounds=(0, 1))
m.v6 = pyo.Var()
m.v7 = pyo.Var(units=pyo.units.m, bounds=(0, 1))
m.v8 = pyo.Var()

m.c1 = pyo.Constraint(expr=m.v1 + m.v2 == 10)
m.c2 = pyo.Constraint(expr=m.v3 == m.v4 + m.v5)
m.c3 = pyo.Constraint(expr=2*m.v3 == 3*m.v4 + 4*m.v5 + m.v6)
m.c4 = pyo.Constraint(expr=m.v7 == 1e-8*m.v1)

m.v4.fix(2)
m.v5.fix(2)
m.v6.fix(0)
```

Try the Toolbox Yourself

Diagnostics Checks

Structural Issues

- Degrees of freedom
- Structural singularities
- Inconsistent units of measurement
- Potential evaluation errors (v2.3)
- Unused variables
- Variables fixed to 0

Numerical Issues

- Constraints with large residuals
- Variables at or beyond bounds
- Extreme Jacobian rows, columns and entries
- Variables near bounds
- Variables with extreme values
- Variables with no value

Advanced Checks

- Singular Value Decomposition (SVD) analysis (v2.3)
- Degeneracy Hunter (v2.3)

Contact: Andrew Lee, andrew.lee@netl.doe.gov

Disclaimer This presentation was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors. The Lawrence Berkeley National Laboratory (LBNL) is managed and operated by the University of California (UC) under U.S. Department of Energy Contract No. DE-AC02-05CH11231. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

NATIONAL ENERGY TECHNOLOGY LABORATORY · BERKELEY LAB · Sandia National Laboratories · Carnegie Mellon · West Virginia University · UNIVERSITY OF NOTRE DAME · Georgia Tech · U.S. DEPARTMENT OF ENERGY