

Development of ModCR Collisional Radiative Model

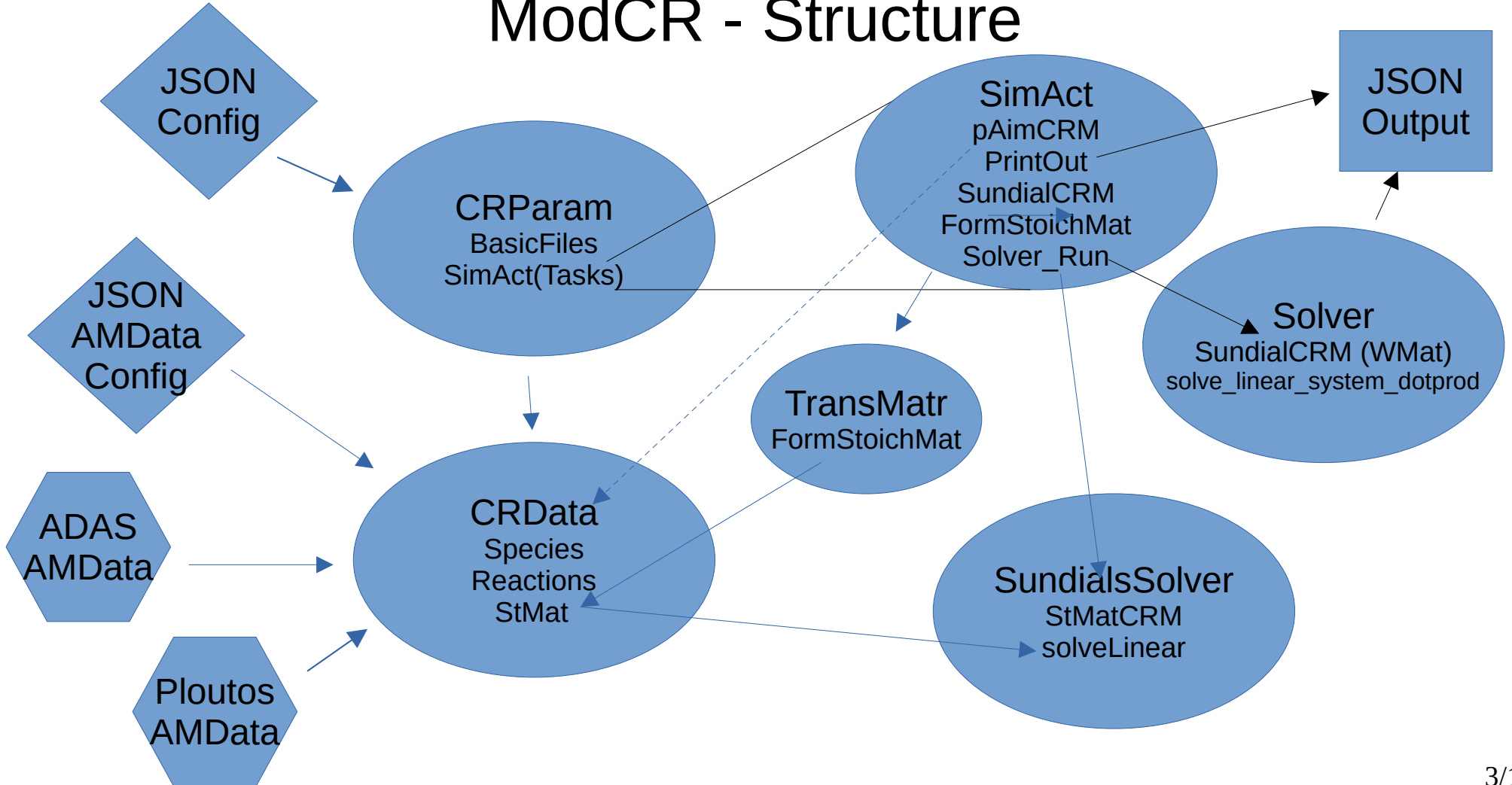
Huw Leggate¹, Dmitry Borodin²

- 1) Dublin City University/Advanced Computing Hub Garching
- 2) Forschungszentrum Jülich

ModCR - Outline

- Standalone Collisional Radiative Model
 - Can also be called as library functions
 - Use modern Fortran – OO principles
 - Include metastable, bundled and vibrational energy states
 - Use ADAS or PLOUTOS data
 - JSON input files
 - Use LLNL Sundials solver library (cvsolve)

ModCR - Structure



Species & Reactions

- Species defined in AMData JSON

- Stored in single list of types

Abstract Species
name, charge, states

Atomic

Molecular

Both accessed through
type bound procedures
No public access routines

Components not private

- Reactions specified in AMData JSON – stored in reaction list

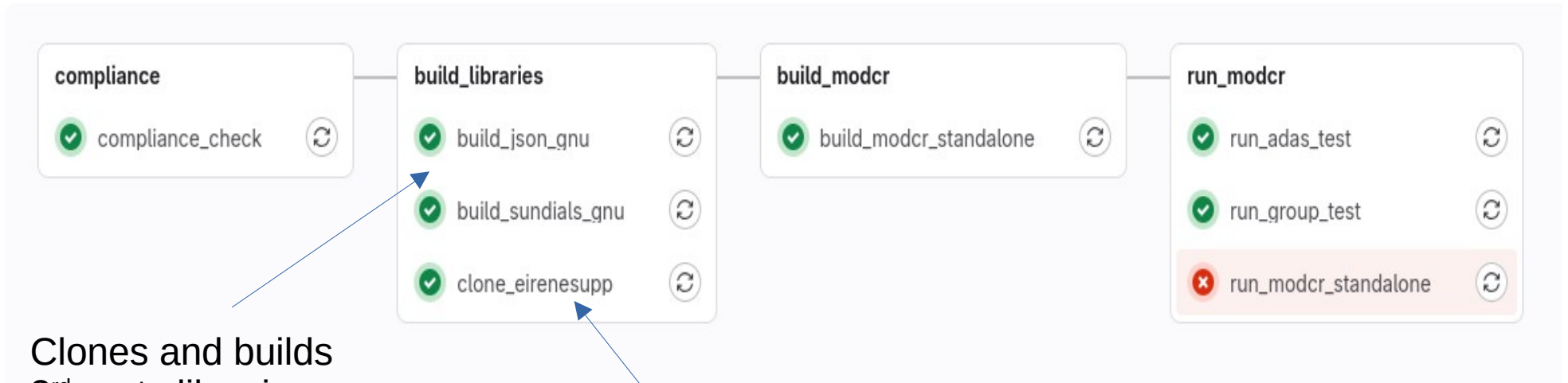
- Each reaction related to a charge/state pair
- 2D Temperature/Density
- Linear interpolation function returns scalar value

Testing

- MCRm_Tests.f90 - Contains test subroutines
- Program files in src/test – Some included in CI

Test Name	Test Purpose	In CI	Passes
testAdasRead	Reads ADAS reaction data and compares to selected known values	Yes	Yes
testPloutosRead	Reads Ploutos reaction data and compares to selected known values	No	No
testDualRead	Tests initialisation, reading of Plotous file, finalization then reading of ADAS file	No	No
testAdasGroupRead	Tests reading of grouped ADAS species and compares to selected known values	Yes	Yes
testLinIndSolver	Tests execution of Sundials solver – under development	No	Yes
generate_and_solve_linear_system	Tests solve_linear_system_dotprod routine	No	Yes

Continuous Integration



Clones and builds
3rd party libraries

Sparse checkout
reduces storage

Usage

- use :: ModCR
- MCRm_ModCR.f90 – wrapper routines to simplify use
 - ModCRInit, ModCRFinalise. ModCRRunAll
- Input files
 - Main JSON input file – passed to ModCRInit
 - Defines operation and location of Reaction data
 - AMData JSON files
 - Defines Species and Reactions to be read from ADAS or PLOUTOS

Usage

```
program ModCR_Main
```

```
  use :: ModCR
```

```
  implicit none
```

```
  ! Initialise, read input file and AM data  
  call ModCRInit()
```

```
  ! Run the sims  
  call ModCRRunAll()
```

```
  ! Clean Up  
  call ModCRFinalise()
```

```
end program ModCR_Main
```


Usage

```
program ModCR_Main
```

```
  use :: ModCR
```

```
  implicit none
```

```
  ! Initialise, read input file and AM data  
  call ModCRInit()
```

```
  ! Run the sims  
  call ModCRRunAll("my_input_file.json")
```

```
  ! Clean Up  
  call ModCRFinalise()
```

```
end program ModCR_Main
```

Timeline

April	Try to write project goals. Add CI
May	Start SRD. First 'bug' fixes.
June	SRD ongoing. Start reorganising code, control module. Fixed ADAS reading based on species charge..
July	Abandon SRD. Add Species and Reaction modules. Add Error and Report modules
August	Added tests to CI. Switch to different ADAS input. Add interpolation. Several changes in input format.
September	Update parse routines, agree on input format. Add ADAS parameters Link Reactions and Species. Introduced dbfile. Add ModCR wrapper.
October	Link Species and Reactions with Species/State pair. Added Stoichiometric matrix. Start on Sundials Solve based on existing code before starting again from scratch,
November	Get simple ODE solver running. Debug ChatGPT code...Multiple changes based on the assumption of working solver...

Thankyou