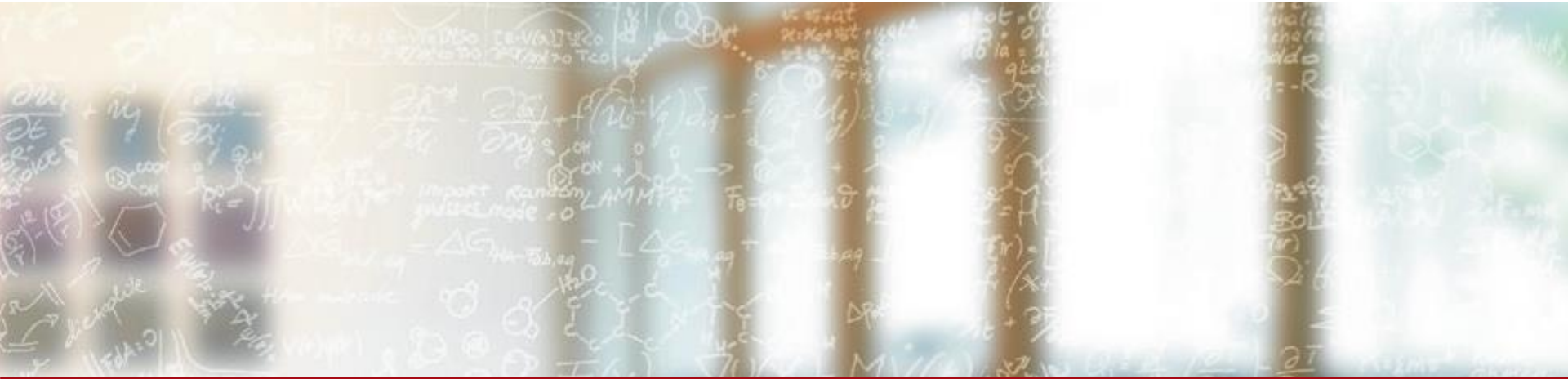




**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich



# Validating new development in ICON with real model reference data

Xiaolin Guo, ETH Zürich

Semester project

Supervised by Ulrike Lohmann, William B. Sawyer, Markus Wetzstein, Sylvaine Ferrachat, Xavier Lapillonne and Andrea Arteaga

# Introduction



- Port the saturation adjustment with OpenACC
- Validate the result → testing framework
- Assumption
  - Rewriting existing parts
  - Old code on CPU, new code on GPU (for ease of exposition)



**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich

# Testing GPU code

---

# GPU results differ from CPU ones

- Parallelizing algorithms may rearrange operations, yielding different numeric results.
- The CPU may be computing results in a precision higher than expected.
- Many common mathematical functions are not required by the IEEE 754 standard to be correctly rounded so should not be expected to yield identical results between implementations

Whitehead, Nathan, and Alex Fit-Florea. "Precision & performance: Floating point and IEEE 754 compliance for NVIDIA GPUs." *rn (A+ B)* 21 (2011): 1-1874919424.

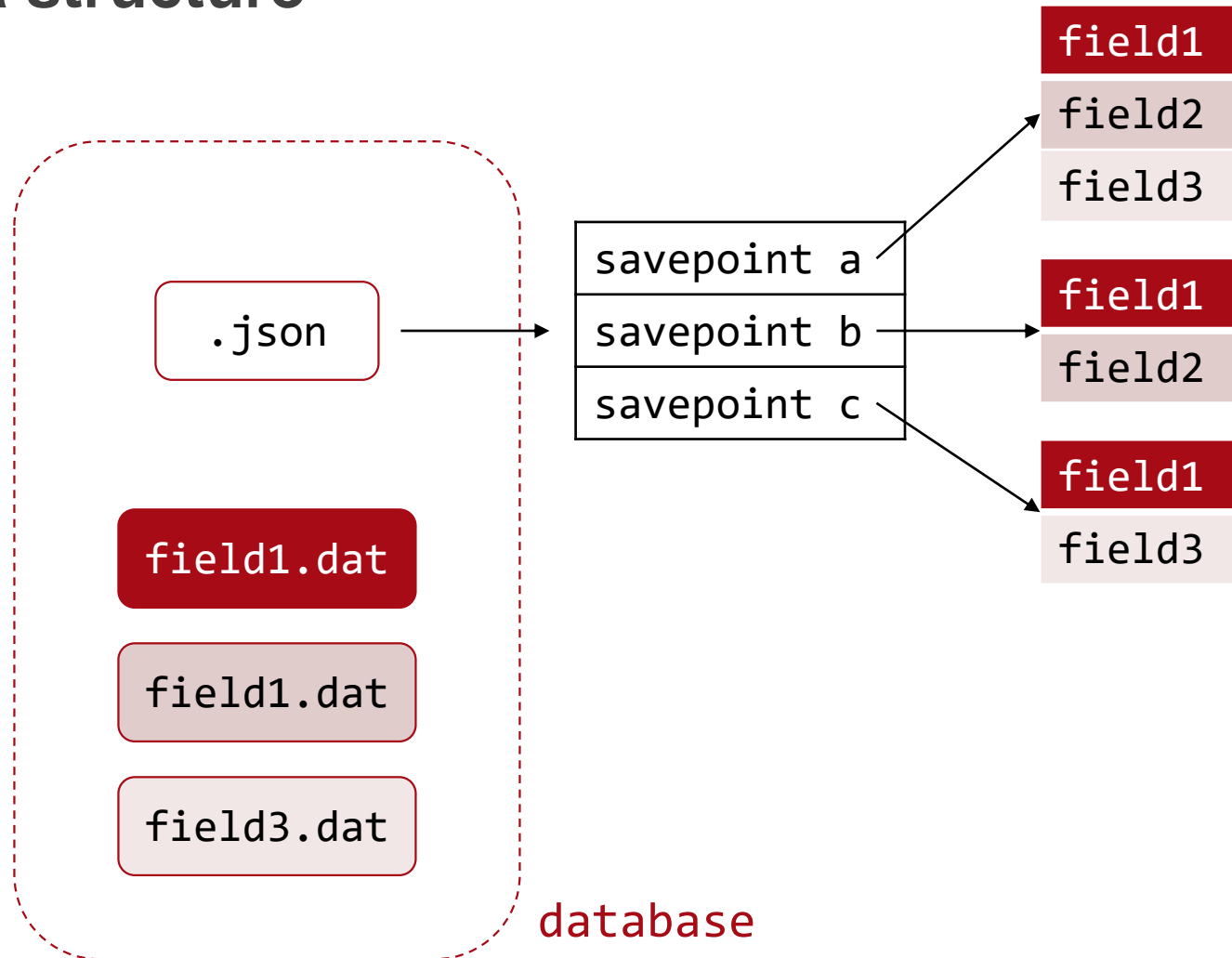
# Validation of new code

- Set a tolerance threshold
  - Error propagation
  - Use CPU input, compare within one step
- Serialization framework in STELLA

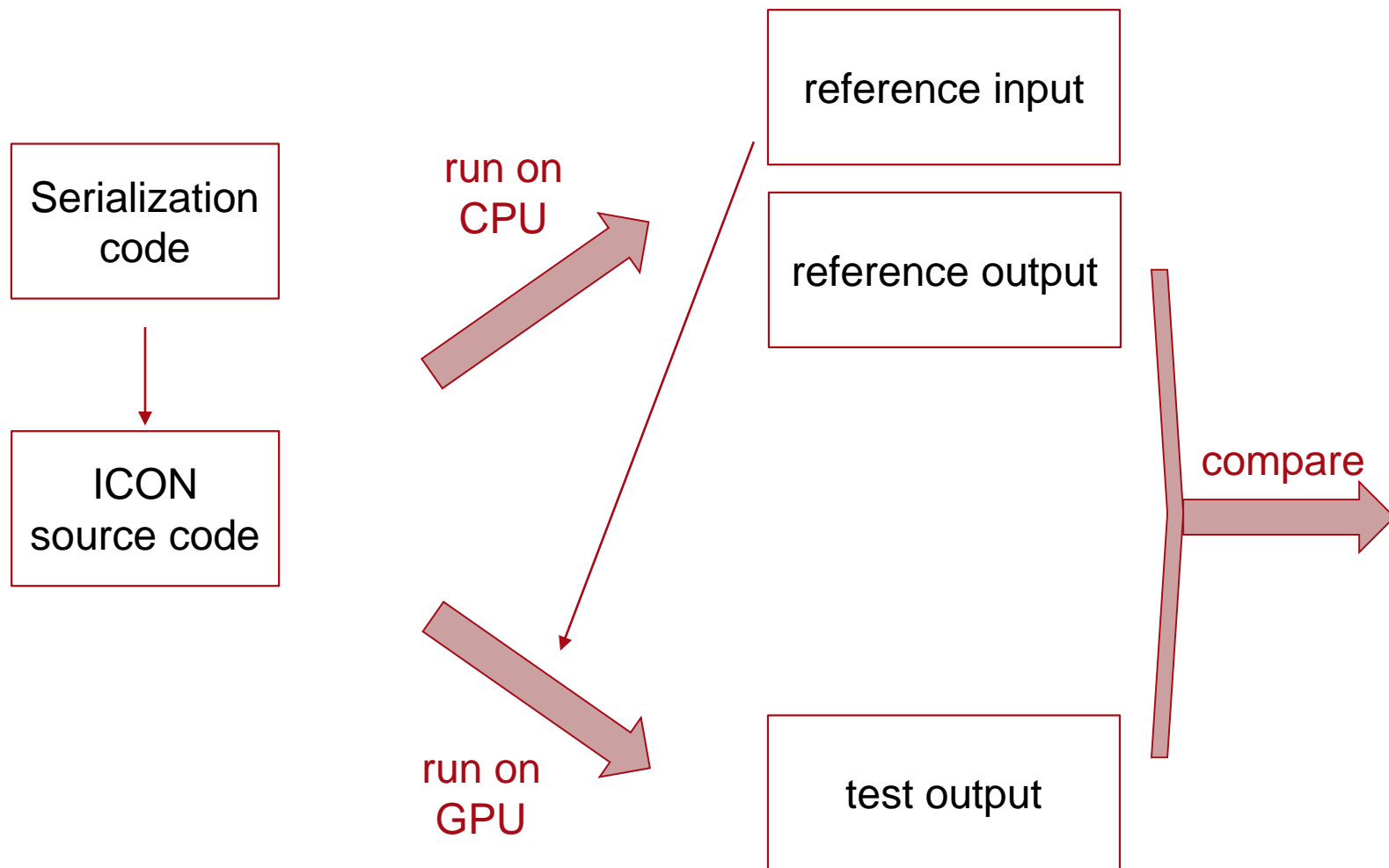
# Serialization framework

- Dump data & read data
- Fortran wrapper

# Data structure



# Workflow







**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich

# Source-to-source directives translation with `pp_ser.py`

---

# pp\_ser.py

- !\$ser directives

```
!$ser init directory='.'  
prefix='Field'
```



```
#ifdef SERIALIZE  
call  
ppser_initialize(directory='.',p  
refix='Field')
```

```
!$ser savepoint cuadjtq.DoStep-  
in iteration=test_counter
```



```
call  
fs_create_savepoint('cuadjtq.DoS  
tep-in', ppser_savepoint)  
call  
fs_add_savepoint_metainfo(ppser_  
savepoint, 'iteration',  
test_counter)
```

```
!$ser mode ser_test_mode
```



```
call  
ppser_set_mode(ser_test_mode)  
#endif
```

# A first look

```
SUBROUTINE cuadjtq()
```

```
!$ser init directory='.' prefix='Field'  
!$ser savepoint cuadjtq.DoStep-in iteration=test_counter  
!$ser mode ser_test_mode  
!$ser data pt=pt(:,kk) pq=pq(:,kk)
```

```
ACC_PREFIX PARALLEL, IF (i_am_accel_node)  
ACC_PREFIX END PARALLEL
```

```
!$ser savepoint cuadjtq.DoStep-out iteration=test_counter  
!$ser mode write  
!$ser data pt=pt(:,kk) pq=pq(:,kk)
```

```
END SUBROUTINE cuadjtq
```

# mode and data

- mode : 0 or 1

```
!$ser mode write = !$ser mode 0
```

```
!$ser mode read  = !$ser mode 1
```

- data : read or write integer, double, array

```
!$ser data pt=pt(:,kk)
```

```
SELECT CASE ( ppser_get_mode() )
```

```
  CASE(0)
```

```
    ACC_PREFIX UPDATE HOST ( pt(:,kk) )
```

```
    call fs_write_field(ppser_serializer, ppser_savepoint, 'pt',  
pt(:,kk))
```

```
  CASE(1)
```

```
    call fs_read_field(ppser_serializer_ref, ppser_savepoint,  
'pt', pt(:,kk))
```

```
    ACC_PREFIX UPDATE DEVICE ( pt(:,kk) )
```

```
END SELECT
```

# Read or write input?

```
MODULE mo_cuadjust
```

```
!$ser verbatim USE mo_run_config, ONLY: ser_test_mode
```

```
SUBROUTINE cuadjtq()
```

```
!$ser savepoint cuadjtq.DoStep-in iteration=test_counter
```

```
!$ser mode ser_test_mode
```

```
!$ser data pt=pt(:,kk) pq=pq(:,kk)
```

```
.....
```

```
!$ser savepoint cuadjtq.DoStep-out iteration=test_counter
```

```
!$ser mode write
```

```
!$ser data pt=pt(:,kk) pq=pq(:,kk)
```

## Other features

- Two database: reference database & test database
- Test first 100 times: Turn off serializer after 100 call
- Remove INTENT(IN):

```
!$ser data pp=pp(:) removeintentin  
#ifdef SERIALIZE  
    REAL(wp) :: pp(kbdim)  
#else  
    REAL(wp), INTENT (IN) :: pp(kbdim)  
#endif
```

- Multiple MPI node: Add prefix in database name

```
Rank0_ref.json
```

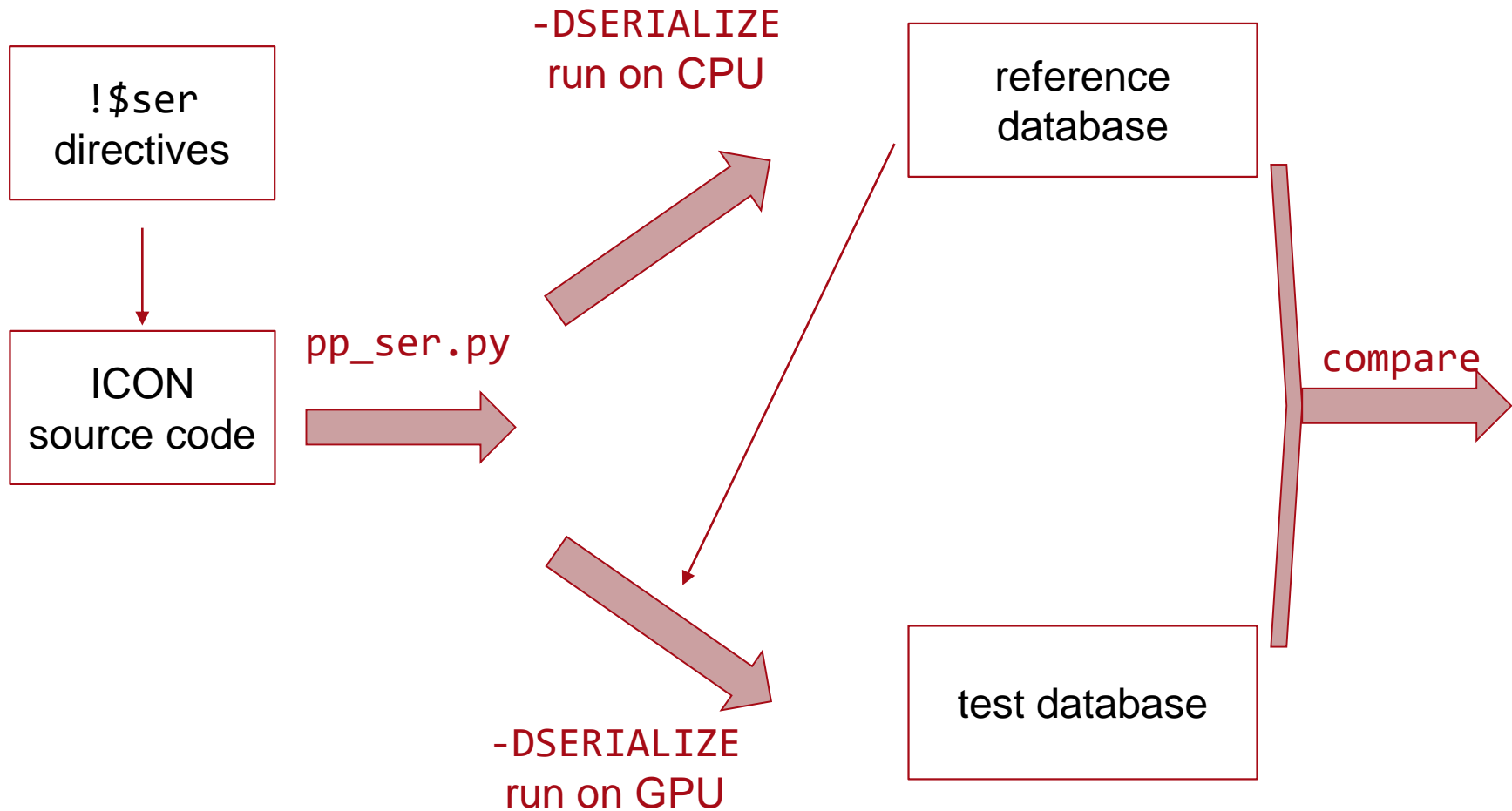
```
Rank1_test.json
```

# compare

- Set threshold
- Select fields
- Select field ranges

```
test_rank0_field
ref_rank0_field
-----
cuadjtq.DoStep-out
[ iteration=0 ]
      pq
      pt
-----
cuadjtq.DoStep-out
[ iteration=1 ]
      pq
      pt
...
-----
cuadjtq.DoStep-out
[ iteration=54 ]
      pq
      Number of values: 2048
      Number of errors: 3
      Percentage of errors: 0.15 %
      Maximum absolute error: 6.2238296015e-05
      Maximum relative error: 3.0619022273e-02
      pt
      Number of values: 2048
      Number of errors: 3
      Percentage of errors: 0.15 %
      Maximum absolute error: 2.1275478659e-01
      Maximum relative error: 8.0221016053e-04
...
-----
```

# Workflow







**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich

# Conclusion

---

# Conclusion

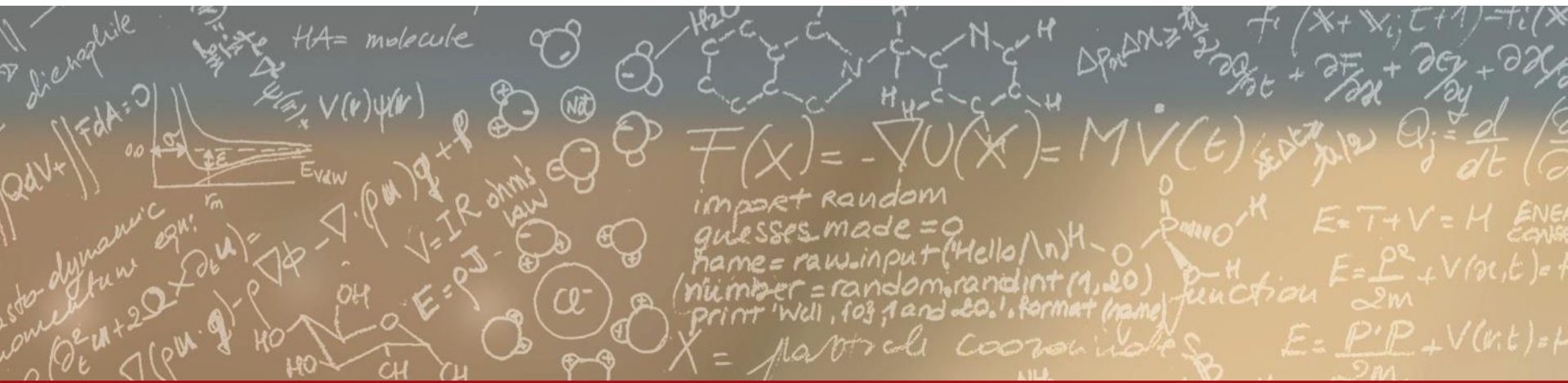
- Not a formal unit test
- !\$ser directives, easy to write, easy to ignore
- Test subroutine or code fragment
- Can be used for debugging



**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich



**Thank you for your attention.**