CamFort

CamFort is our free, open source refactoring and verification tool for Fortran providing:

- data gathering on the programming patterns common in scientific models;
- automatic checking & refactoring to improve the code quality of existing models;
- easy integration into your existing workflow.

Units-of-measure

Code that mixes up physical units can produce bogus output, leading to spectacular failures such as the Mars Climate Orbiter mission, or forced retractions of submitted scientific papers.

```
!= unit(N s) :: p1
!= unit(lbf s) :: p2
real :: p1, p2
! ...units-checker will detect an error here:
p1 = p2
```

Commands

`units-suggest` Find the minimum set of variables that must be manually annotated with units.

`units-infer` Analyse the code and determine the consistent units for all variables, including those without annotations.

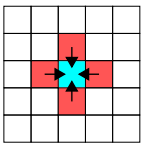
`units-synth` Output a copy of the original source code with the results of units inference inserted as annotations

`units-check` Verify all unit annotations in the files for consistency against the Fortran code in the files, even after any changes have been made.

For more information:
www.cl.cam.ac.uk/research/dtg/camfort
Source code:
www.github.com/camfort

UNIVERSITY OF CAMBRIDGE

Stencil specifications



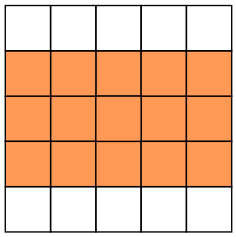
A common pattern in scientific code is the "stencil" where each cell in an array is updated with an equation using the adjacent cells. These stencils can involve writing very error-prone indexing code. Even just an off-by-1 mistake can produce radically different output, as shown:

Sample 2-D output of very similar stencil codes

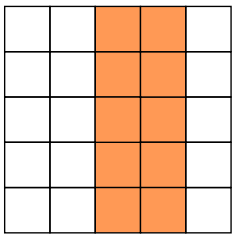
A stencil annotation like this might appear in code linking a variable with a region:

```
!= centered(depth=1, dim=1) :: a
a(i,j) = (a(i-1,j) + a(i,j) + a(i+1,j)) / 3
```

Regions can be combined to form more complex descriptions:

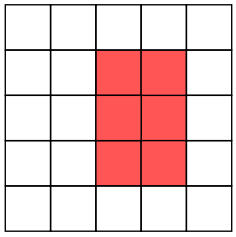


`centered(depth=1, dim=1)`



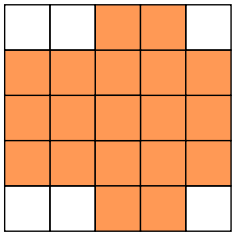
`forward(depth=1, dim=2)`

Conjunction



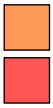
`centered(depth=1, dim=1)`
`*`
`forward(depth=1, dim=2)`

Disjunction



`centered(depth=1, dim=1)`
`+`
`forward(depth=1, dim=2)`

Cell key



may be read

must be read