

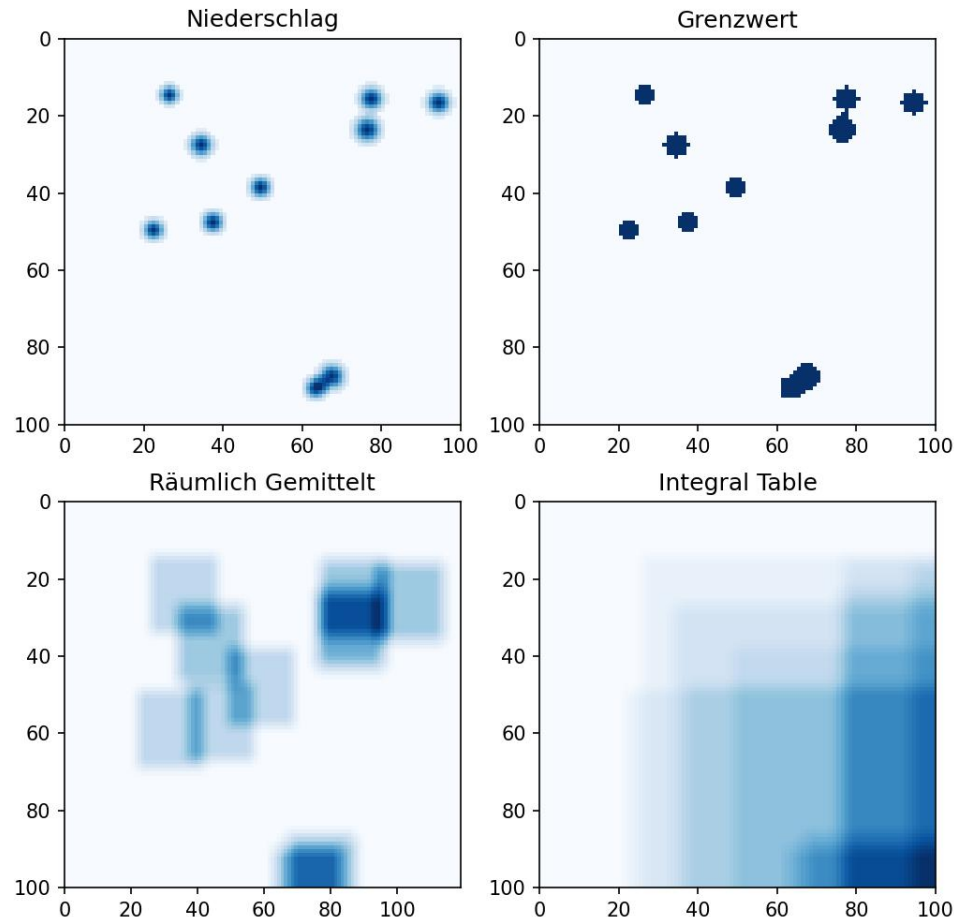
FSS in Fortran implementiert

Ludwig Wolfgruber

Numerische Modellierung in Fortran

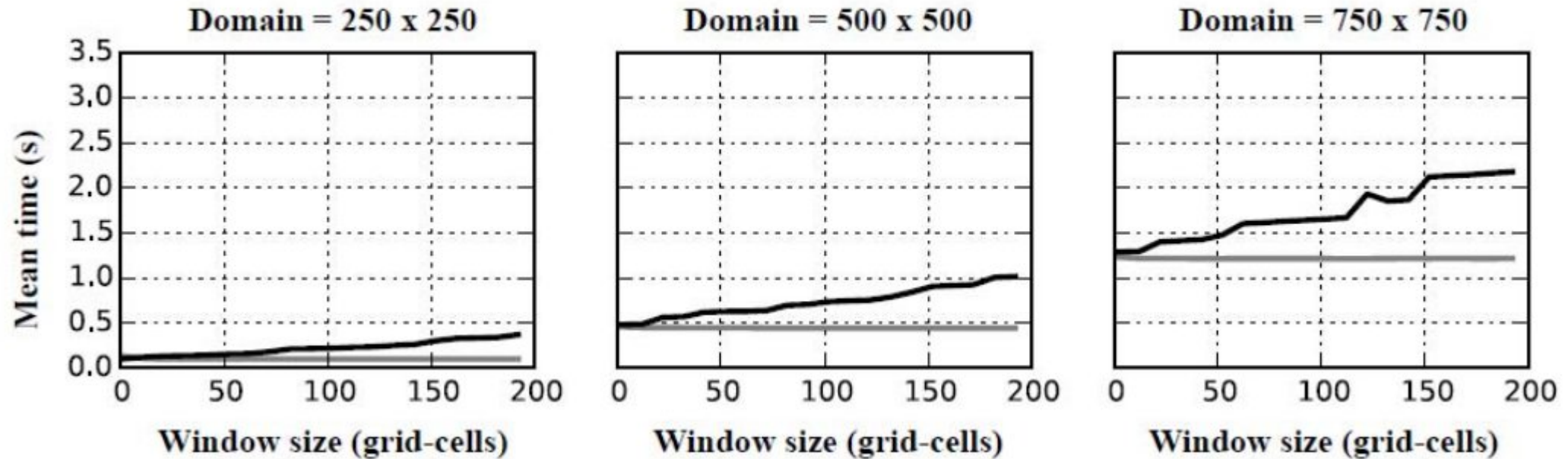
FSS

- Maß zur Niederschlagsverifikation
- Relativer quadrierter Fehler von räumlich gemittelten Feldern



$$FSS = 1 - \frac{\frac{1}{N} \sum_{i=1}^N (p_f - p_o)^2}{\frac{1}{N} \sum_{i=1}^N p_f^2 + \frac{1}{N} \sum_{i=1}^N p_o^2}$$

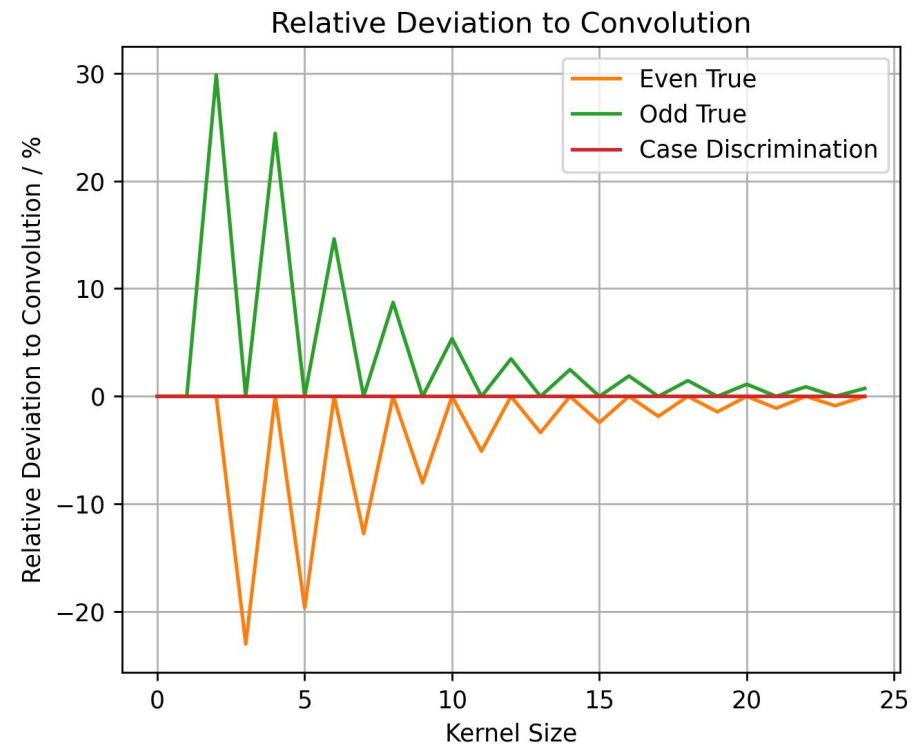
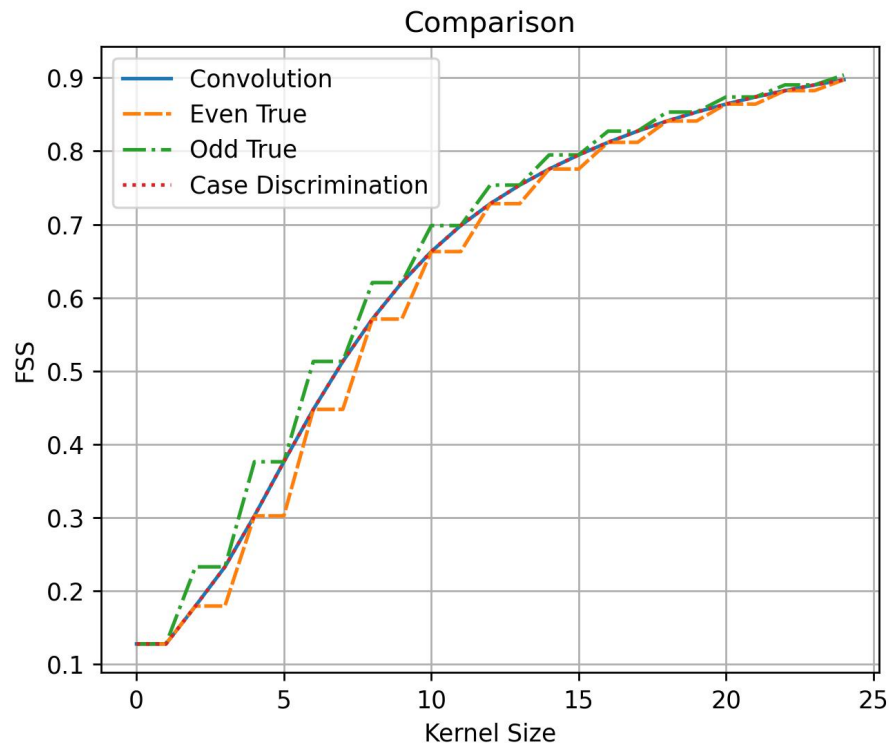
Faggians Algorithmus



Integral Table lookup statt Mittelwert:

$$(4 N N_k + 2N) / (N \log N k N_k) = (2 + 4 N_k) / (\log N k N_k)$$

Bug in Faggian



Im Code wird Kernelgröße halbiert und abgerundet, dadurch nur korrekt für gerade Kernels, durch Fallunterscheidung für beliebige Kernelgrößen verwendbar.

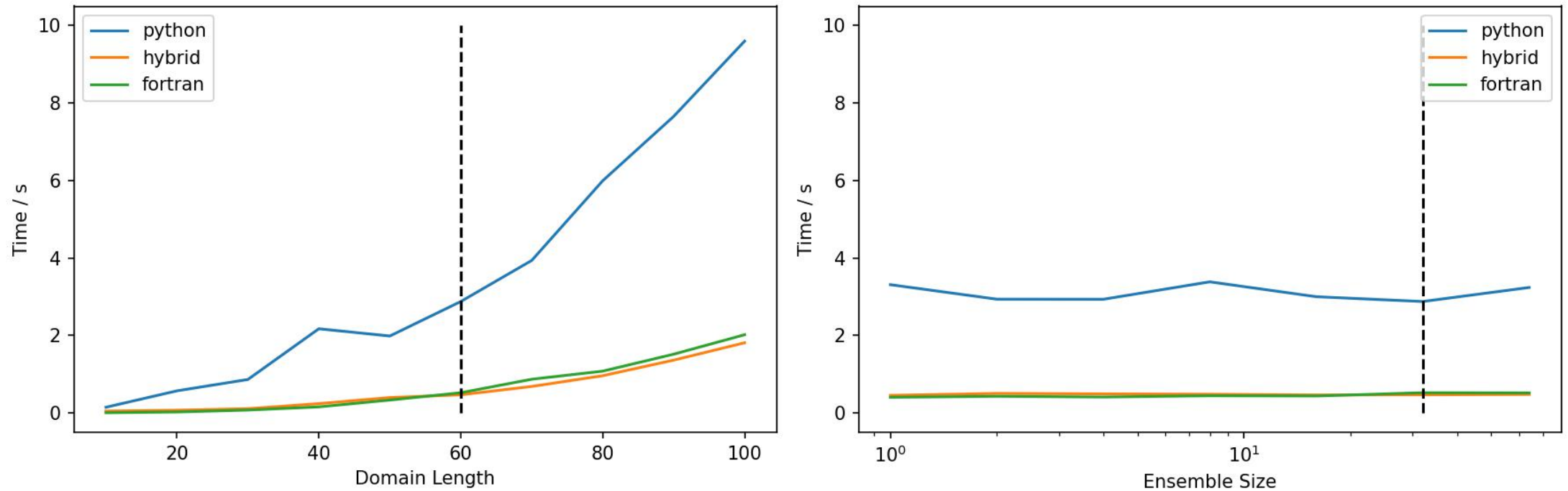
Fortran und Python – Vergleich

```
SUBROUTINE integral_table (n1, n2, &  
                           array, table)  
    INTEGER, INTENT(IN) :: n1, n2  
    INTEGER :: i, j  
    REAL, INTENT(IN) :: array(n1, n2)  
    REAL, INTENT(OUT) :: table(n1, n2)  
    table(1,:) = array(1,:)   
    DO i=2,n1  
        table(i,:) = array(i,:) + table(i-1,:)   
    END DO  
    DO j=2,n2  
        table(:,j) = table(:,j) &  
                     + table(:, j-1)  
    END DO  
END SUBROUTINE integral_table
```

```
import numpy as np  
  
def compute_integral_table(array):  
    return array.cumsum(1).cumsum(0)
```

Achtung Indizes!

Fortran und Python – Vergleich



Laufzeiten für verschiedene Domain- und Ensemblegrößen.

Fortran und Python kombinieren

- Kompilieren mit (numpy Bibliothek)

```
f2py -c -m package file.f90 --f90flags="-O2"
```

- In Python laden mit

```
from package import modname
```

- Achtung:

- Indizes werden auf C-Type (auch Python) umgestellt
- Argumente von Subroutinen und Funktionen können sich ändern