

Mehrgitter und konjugierte Gradienten Verfahren

Juri Schröder, Alexander Schmidt

9. Februar 2017

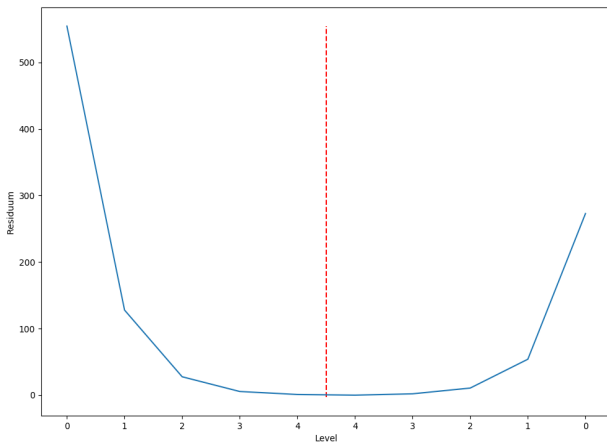
Inhaltsverzeichnis

- 1 Mehrgitter Verfahren
- 2 konjugierte Gradienten Verfahren

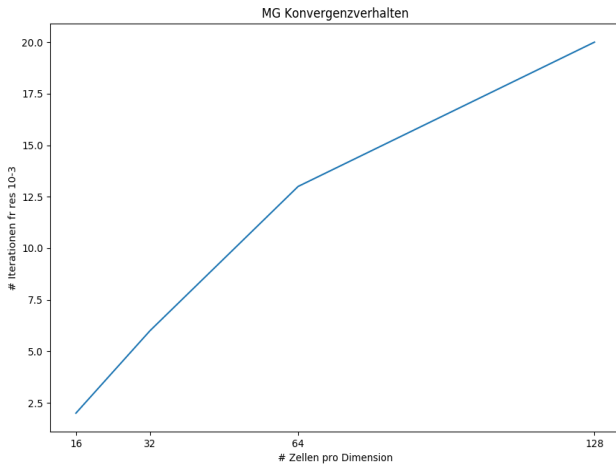
Mehrgitter

```
function MG_ITERATION( $N$ ,  $p$ ,  $rhs$ )  
  SMOOTH( $N$ ,  $p$ ,  $rhs$ )  
   $res \leftarrow$  COMPUTE_RES( $N$ ,  $p$ ,  $rhs$ )  
   $res\_c \leftarrow$  RESTRICT( $N$ ,  $res$ )  
  if  $N \leq 8$  then  
    SOLVE( $p$ ,  $rhs$ )  
  else  
     $e\_c \leftarrow$  MG_ITERATION( $N/2$ ,  $e\_c$ ,  $res\_c$ )  
     $e \leftarrow$  INTERPOLATE( $N$ ,  $e\_c$ )  
     $p \leftarrow$  ADD( $p$ ,  $e$ )  
    SMOOTH( $N$ ,  $p$ ,  $rhs$ )
```

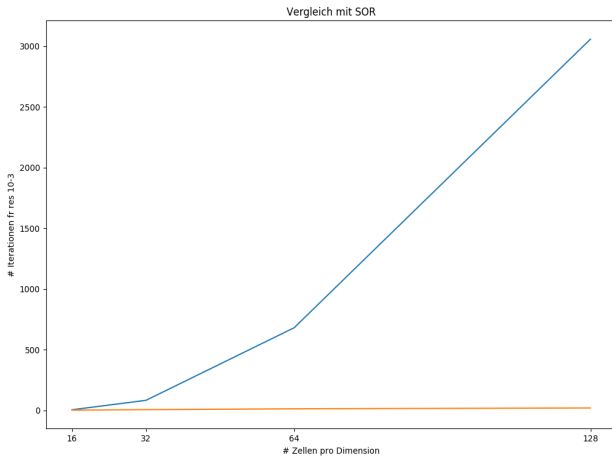
Mehrgitter Konvergenz



Mehrgitter Konvergenzanalyse



Vergleich mit SOR



Theorie

$Ax = b$, mit $A \in \mathbb{R}^{n \times n}$ positive-definit und symmetrisch, $b \in \mathbb{R}^n$

Wähle $x_0 \in \mathbb{R}^n$

$r = b - Ax$

$d = r$

while($\|r\| < tol$)

$z = Ad$

$\alpha = \frac{r^T r}{d^T z}$

$x = x + \alpha d$

$c = r^T r$

$r = r - \alpha z$

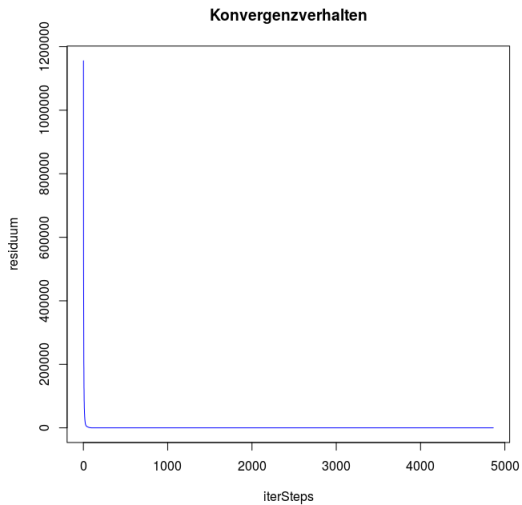
$\beta = \frac{r^T r}{c}$

$d = r + \beta d$

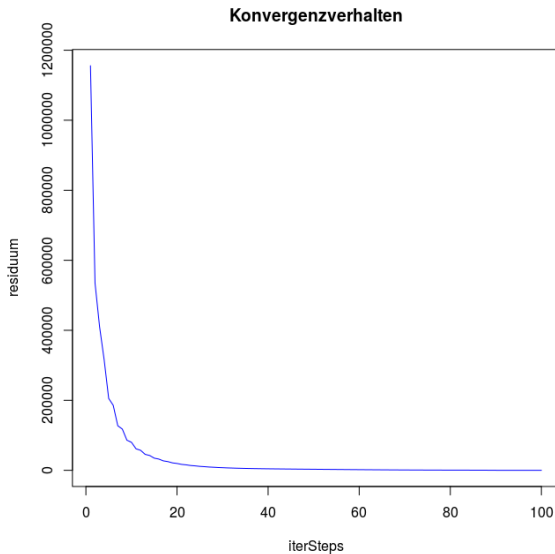
end while

```
for( it.First(); it.Valid(); it.Next() ){  
    Ad->Cell( it ) = _direction->dxx( it ) + _direction->d
```

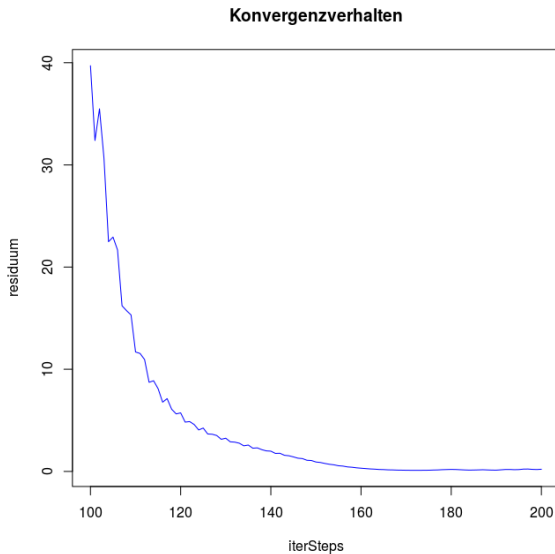

Konvergenzverhalten



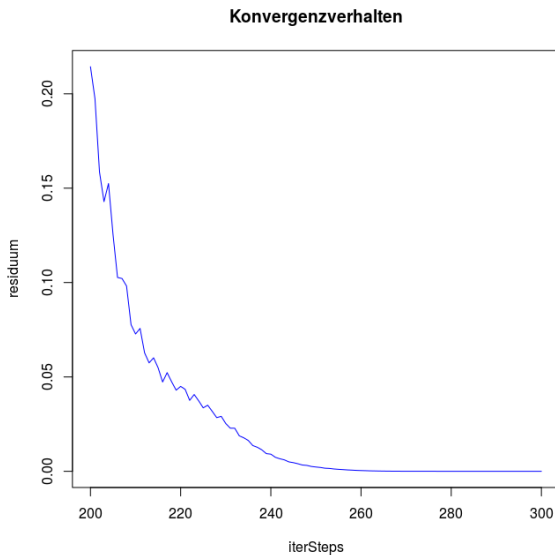
Konvergenzverhalten



Konvergenzverhalten



Konvergenzverhalten



Konvergenzverhalten

