Mehrgitter und konjugierte Gradienten Verfahren

Juri Schröder, Alexander Schmidt

10.02.17

Inhaltsverzeichnis

Mehrgitter

```
function MG_Iteration(N, p, rhs)
    smooth(N, p, rhs)
    if N ≤ 8 then
        solve(p, rhs)
    else
        res ← compute_res(N, p, rhs)
        res_c ← restrict(N, res)
        e_c ← MG_Iteration(N/2, e_c, res_c)
        e ← interpolate(N, e_c)
        p ← add(p, e)
        smooth(N, p, rhs)
```

Mehrgitter Implementierung:

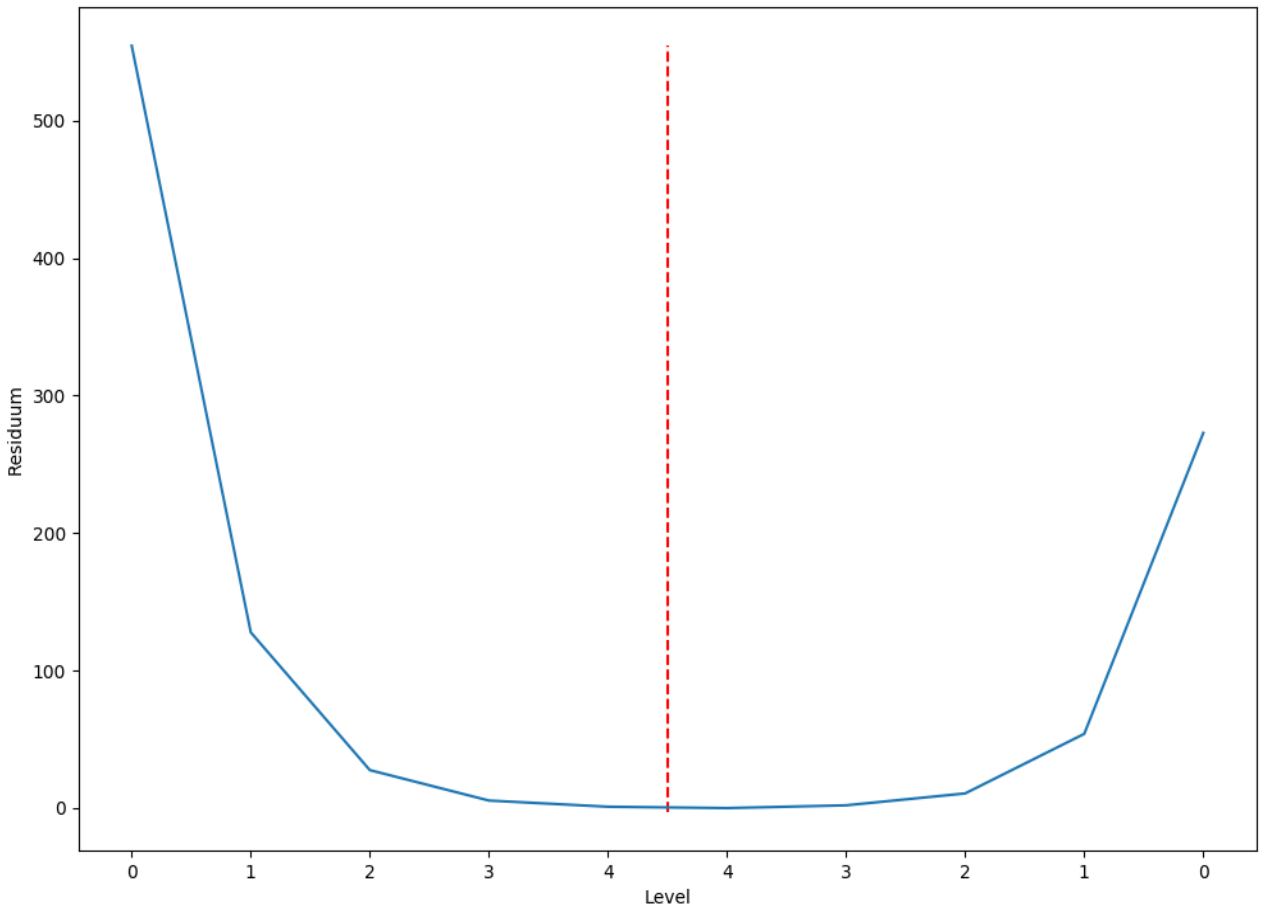
Generell:

- Glätten mit 2 Gauß-Seidel-Iterationen
- Lösen sobald $N \leq 8$ mit SOR

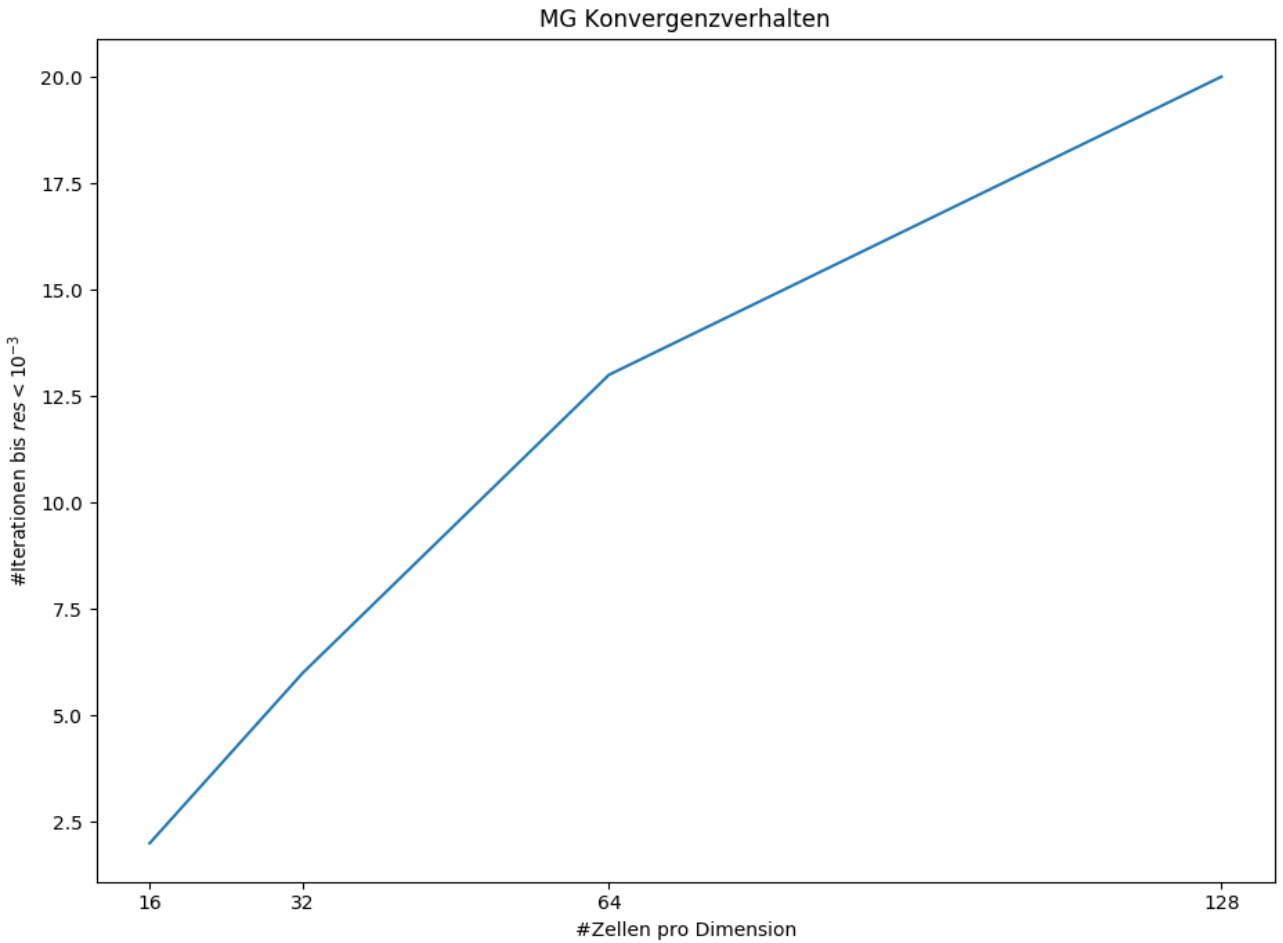
Parallelisierung:

- RedBlack-GS und RedBlackSOR
- Boundary-Austausch nach jedem Glätte- / Löser-Zyklus

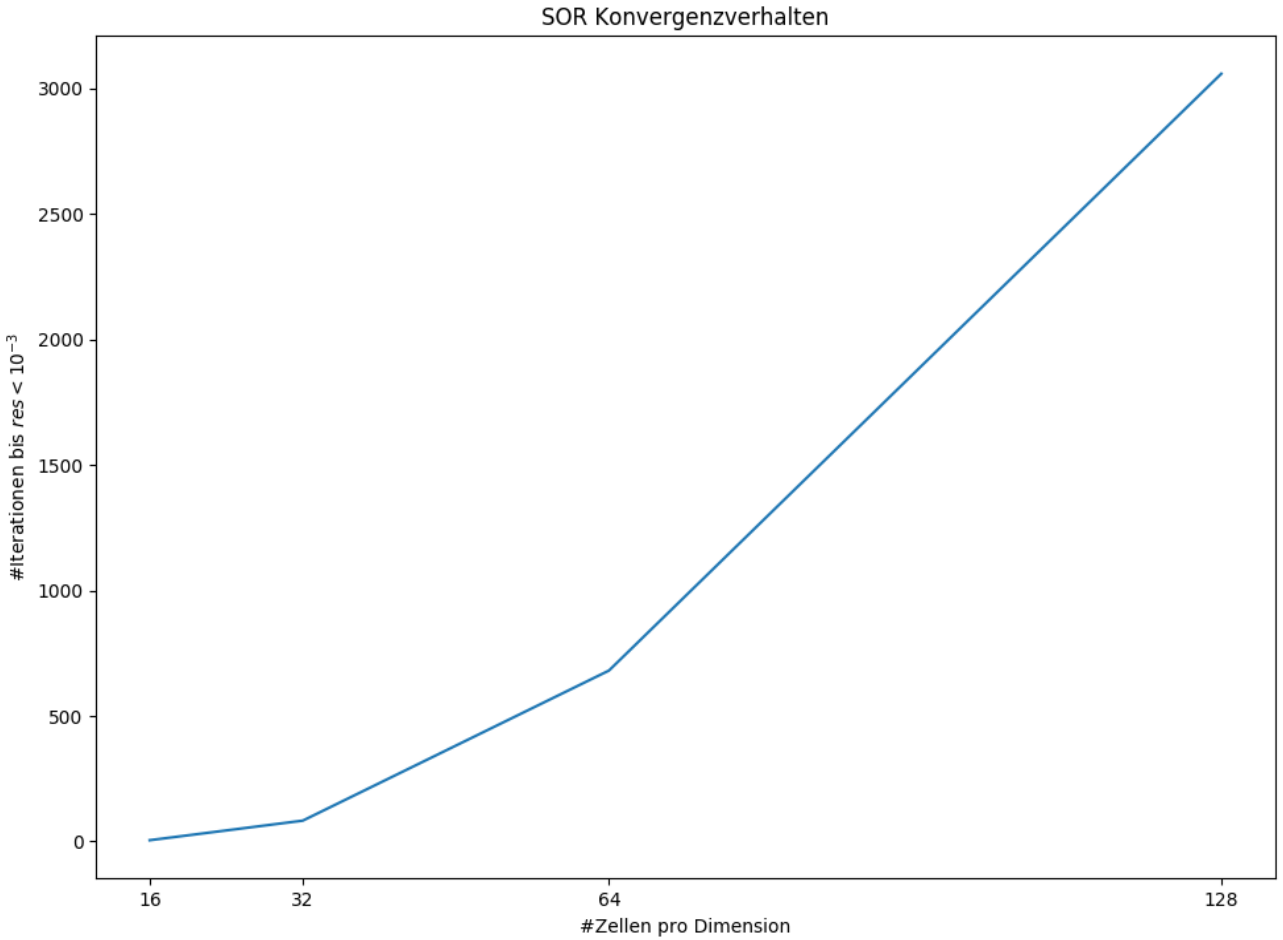
Mehrgitter Konvergenz



Mehrgitter Konvergenzanalyse



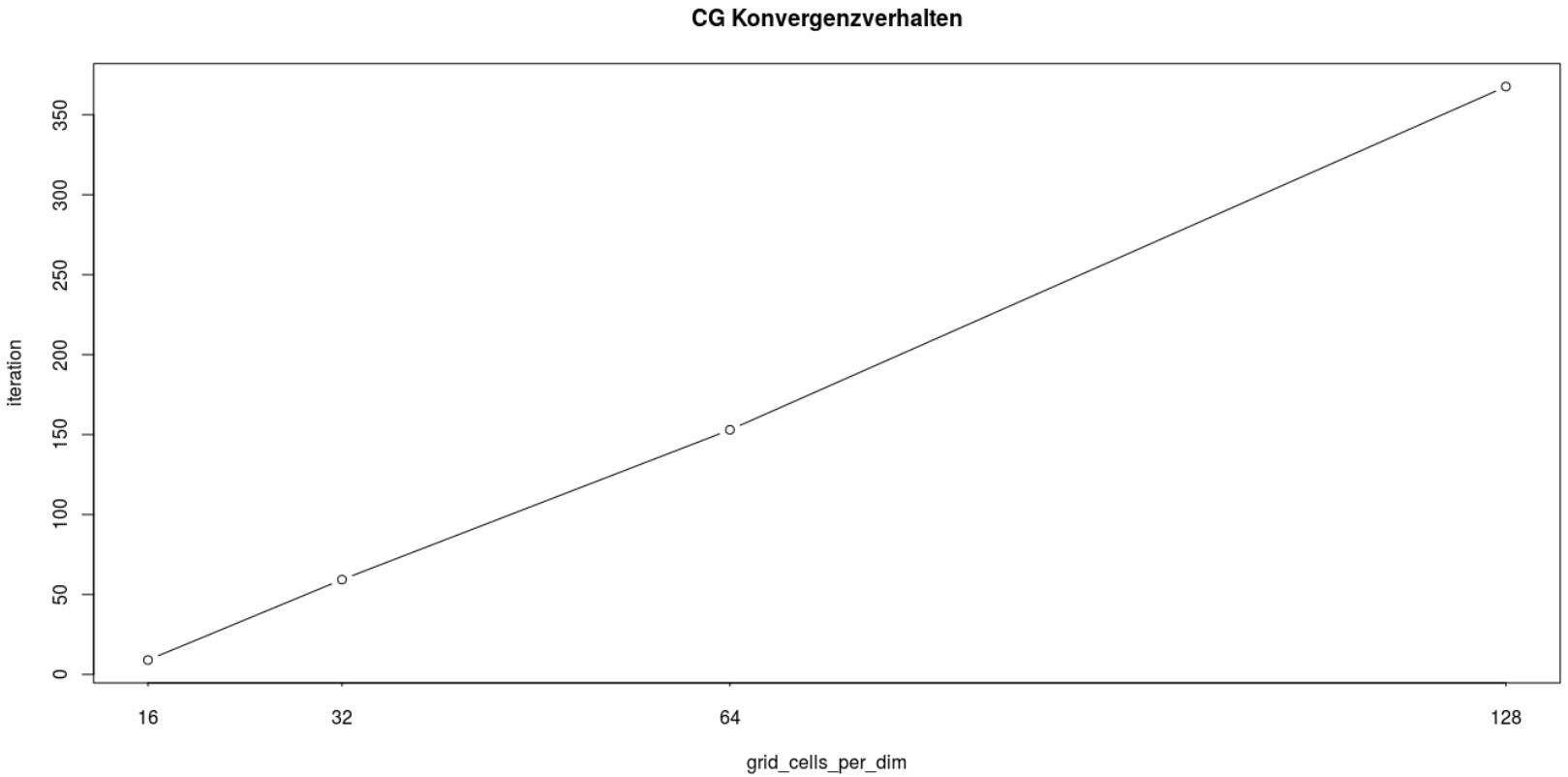
Vergleich mit SOR



konjugierte Gradienten Verfahren

```
function CG::Cycle( p, res, dir)
    for all InteriorIterator it do
        Ad.Cell(it) ← dir.dxx(it) + dir.dyy(it);
    res_old ← res.dotProduct(res)
    alpha ← res_old / dir.dotProduct(Ad)
    p ← p + alpha * dir;
    res ← res - alpha * Ad;
    res_new ← res.dotProduct( res );
    beta = res_old / res_new;
    dir ← dir + beta * res;
    return res_new;
```

Konvergenzverhalten



Konvergenzverhalten