

Simulation and Scientific Computing 2 Seminar

David Uhl, Thomas Stadelmayer

Friedrich Alexander Universität Erlangen Nürnberg

June 3, 2015

- 1 Optimization
 - RBGS

Red-Black Gauss-Seidel

```
1 for (int iter = 0; iter < times; iter++){
2 // red points
3 #pragma omp parallel for
4 for (int j = 1; j < height-1; j++){
5     for (int i = 1; i < width-1; i++){
6         if(j == (height-1)*0.5 && i >= (width-1)*0.5) continue;
7         // i+j gerade
8         if( ((i + j) % 2) == 0){
9             u(i,j) = factor * (f(i,j) + h_2_inv * (
10                 u(i-1, j) + u(i+1, j) + u(i, j+1) + u(i, j-1)));
11 }}}
12 // black points
13 #pragma omp parallel for
14 for (int j = 1; j < height-1; j++){
15     for (int i = 1; i < width-1; i++){
16         if(j == (height-1)*0.5 && i >= (width-1)*0.5) continue;
17         // i+j ungerade
18         if( ((i + j) % 2) == 1){
19             u(i,j) = factor * (f(i,j) + h_2_inv * (
20                 u(i-1, j) + u(i+1, j) + u(i, j+1) + u(i, j-1)));
21 }}}}
```

```

1 void MGSolver::restrict_2d (...){
2     // restrict to coarser domain
3 #pragma omp parallel for schedule(static)
4     for(int j = 1; j < height-1; j++){
5         for(int i = 1; i < width-1; i++){
6             if(j == (height-1)*0.5 && i >= (width-1)*0.5) continue;
7             int mid_i = 2*i; int mid_j = 2*j;
8             u_2h(i, j) =
9                 rest.getw1() * u(mid_i - 1, mid_j + 1) +
10                 rest.getw1() * u(mid_i      , mid_j + 1) +
11                 rest.getw3() * u(mid_i + 1, mid_j + 1) +
12                 rest.getw4() * u(mid_i - 1, mid_j      ) +
13                 rest.getw5() * u(mid_i      , mid_j      ) +
14                 rest.getw6() * u(mid_i + 1, mid_j      ) +
15                 rest.getw7() * u(mid_i - 1, mid_j - 1) +
16                 rest.getw8() * u(mid_i      , mid_j - 1) +
17                 rest.getw9() * u(mid_i + 1, mid_j - 1);
18 }}}

```

Figure : Schrödelbert GmbH

time vs threads



