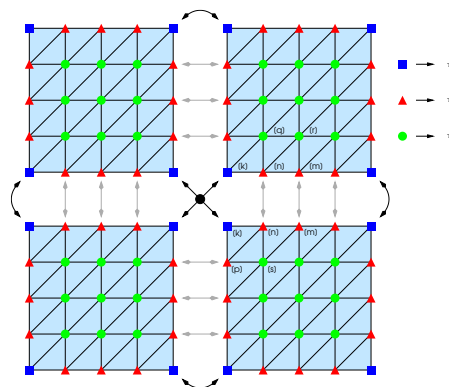




## FINAL PROJECTS

---

High Performance Computing II, Summer Term 2020



Prof. Dr. Stefan Funken  
Prof. Dr. Karsten Urban  
M.Sc. Constantin Greif

Institute for Numerical Mathematics

Das vorliegende Manuskript beinhaltet die Ausarbeitungen der Projekte im Rahmen der Vorlesung High Performance Computing 2, welches im Sommersemester 2020 an der Fakultät für Mathematik und Wirtschaftswissenschaften der Universität Ulm stattgefunden hat.

Bedanken möchten wir uns an dieser Stelle nochmals bei allen Teilnehmern für ihre tollen Beiträge und gelungenen Vorträge, welche alle zusammen wesentlich zum Gelingen der Veranstaltung beigetragen haben.

Ulm, 2020

Stefan Funken, Karsten Urban, Constantin Greif

# Projekt X

Eure Namen

4. August 2020

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
----------	-------------------	----------

# 1 Einleitung

Einordnung des Themas in HPC2, worum geht es.....

Hier noch ein Algorithmus:

---

**Algorithm 1** Variable metric hybrid inexact proximal point method
 

---

```

1: function VMHPPM( $f, s, z^0$ )
2:    $k \leftarrow 0$ 
3:   while  $k < k_{\max}$  do
4:      $(u^k, g^k, \epsilon_k) := \text{BUNDLE}(f, s, M_k, z^k, \delta_k)$      $\triangleright$  Approximate the proximal point
5:     if  $\frac{1}{2}\|g^k\|^2 \leq \rho \wedge \epsilon_k \leq \rho$  then
6:       return  $u^k$                                           $\triangleright$  Optimal or close to optimal
7:     end if
8:      $z^{k+1} := u^k$                                         $\triangleright$  Step to the proximal point
9:      $k \leftarrow k + 1$ 
10:  end while
11: end function

```

---

Referenz: In Algorithmus 1.

Hier noch ein Code:

```

1 #include <mpi.h>
2 #include <stdio.h>
3 #include <string.h>
4
5 int main(int argc, char** argv) {
6   int my_rank;           /* Rank of process */
7   int p;                 /* Number of process */
8   int source;            /* Rank of sender */
9   int dest;              /* Rank of receiver */
10  int tag = 50;          /* Tag for messages */
11  char message[100];     /* Storage for the message */
12  MPI_Status status;     /* Return status for receive */
13
14  MPI_Init(&argc, &argv);
15  MPI_Comm_rank(MPLCOMM_WORLD, &my_rank);
16  MPI_Comm_size(MPLCOMM_WORLD, &p);
17
18  if (my_rank != 0){

```

```
19     sprintf(message, "Hello world from processor %d!", my_rank);
20     dest = 0;
21     /* Use strlen(message)+1 to include '\0' */
22     MPI_Send(message, strlen(message)+1, MPI_CHAR, dest, tag,
               MPLCOMM_WORLD);
23 } else {
24     for (source=1; source<p; source++){
25         MPI_Recv(message, 100, MPI_CHAR, source, tag, MPLCOMM_WORLD,
                  &status);
26         printf("%s\n", message);
27     }
28 }
29 MPI_Finalize();
30 }
```

./ProjectX/Code/HelloWorld.c

## Literatur

- [1] A. BEN-TAL, M. KOCVARA, A. NEMIROVSKI, J. ZOWE: Free Material Design via Semidefinit Programming, SIAM Review, Vol. 42, No. 4 (2000), 695-715.