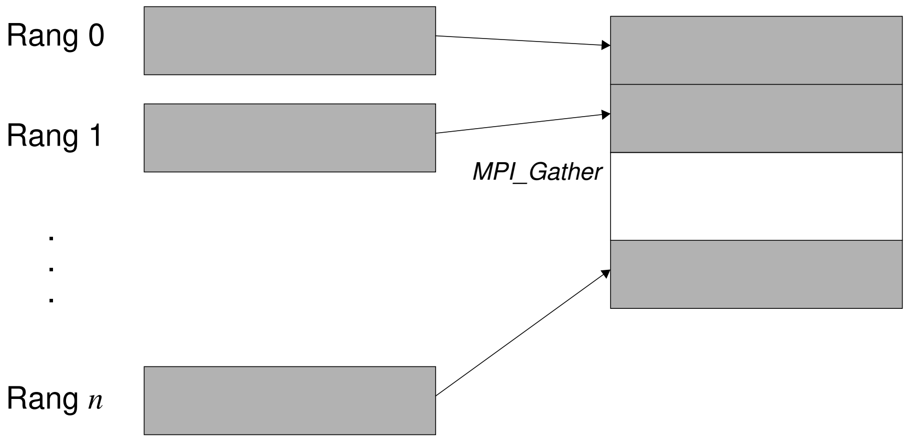
1 Aufgabe 2

- Implementieren Sie Aufgabe 1 mit der MPI Bibliothek.
- Unterteilen Sie dabei das Bild in *horizontale* Streifen und sammeln Sie die Ergebnisse mit `MPI_Gather()` ein.



2 Aufgabe 3

- Ändern Sie Aufgabe 2 so ab, daß *vertikale* Streifen verwendet werden!
- Sie müssen dazu einen eigenen Datentyp definieren, der einen einzelnen Streifen beschreibt.
- `MPI_Gather()` ist in diesem Fall leider nur durch eine nichttriviale Änderung des Datentyps verwendbar (über `MPI_Struct()` und `MPI_UB`), benutzen Sie daher den einfachen `MPI_Send()`/`MPI_Recv()` Mechanismus.

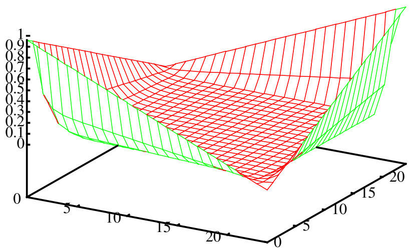
3 Aufgabe 4

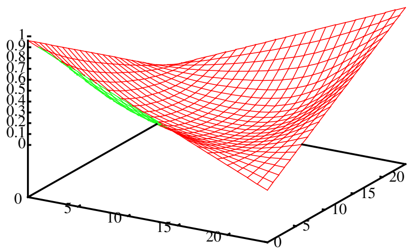
- Über die Mittelwertberechnung

$$grid[x, y] = \frac{(grid[x-1, y] + grid[x+1, y] + grid[x, y-1] + grid[x, y+1])}{4}$$

soll ein Gitter geglättet werden. Die Glättungsoperation ist dabei nicht auf die Randwerte anzuwenden.

Startgitter *geglättetes Gitter*





4 cont.

- Das serielle Programm ist auf der `gsm` unter `~i471000/src/grid` gespeichert.

- Das Programm erzeugt das Startgitter, speichert es unter dem Namen “`gitter.0`” ab, führt dann die Glättung durch und speichert das Ergebnis unter “`gitter.1`”. Über das “`plot`” Shellscript läßt sich die Glättung graphisch visualisieren.

- Parallelisieren Sie das Program mir Hilfe von MPI. Die Initialisierung soll noch von einem einzigen Prozeß durchgeführt werden. Dann soll das Gitter in *horizontale* Streifen unterteilt und mit `MPI_Scatter` an die Prozesse verteilt werden.

5 cont.

- Zur Vereinfachung der Aufgabe können Sie davon ausgehen, daß die Gitterdimension sich durch die Anzahl der Prozesse teilen läßt, so daß jeder Streifen gleich groß ist.
- Da in die Berechnung eines Gitterpunktes seine vier Nachbarn eingehen, müssen vor jeder Iteration die beiden Ränder eines jeden Streifen an die Nachbarprozesse verschickt werden. Dies kann z.B. mit zwei `MPI_Sendrecv` Anweisungen erfolgen.
- Nachdem die Anzahl der Iterationen abgearbeitet wurde, soll das Ergebnis mit Hilfe von `MPI_Gather` vom Prozeß 0 eingesammelt werden und dann abgespeichert werden.

6 cont.

- Diagramm:

Prozeß 0	*Prozeß 1*	*Prozeß n*
– Initialisierung des Gitters		
– Schreiben des Startgitters		
– MPI_Scatter()	– MPI_Scatter()	– MPI_Scatter()
– while (iterations--)	– while (iterations--)	– while (iterations--)
– Senden/Empfang oberer Rand – Senden/Empfang unterer Rand – Glättung des Streifens	– Senden/Empfang oberer Rand – Senden/Empfang unterer Rand – Glättung des Streifens	– Senden/Empfang oberer Rand – Senden/Empfang unterer Rand – Glättung des Streifens
– MPI_Gather()	– MPI_Gather()	– MPI_Gather()
– Schreiben des Ergebnisses		