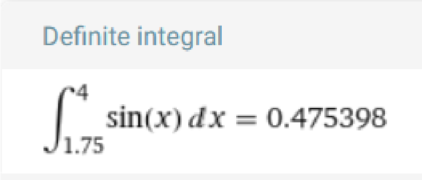
Grundsätzliche Funktion mit hoher Anzahl an Nodes

Anmerkung: Es wurden direkt eine Anzahl von Stützstellen und Nodes gewählt, sodass diese keine einfache Aufteilung (Division) und keine 2er-Potenzen sind. Deswegen wurde dies nicht mehr separat getestet. Zudem ist bereits auch Broadcast und Reduce enthalten. Die Quelle der Referenzbilder ist Wolfram Alpha/Mathematica.

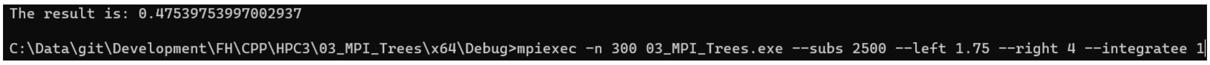


Definite integral

$$\int_{1.75}^{4} \sin(x)\,dx = 0.475398$$

Mit trapezoidal rule:

```
mpiexec -n 300 03_MPI_Trees.exe --subs 2500 --left 1.75 --right 4 --
integratee 1
```



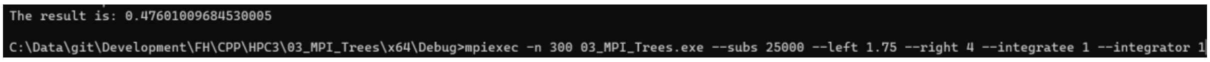
```
The result is: 0.47539753997002937

C:\Data\git\Development\FH\CPP\HPC3\03_MPI_Trees\x64\Debug>mpiexec -n 300 03_MPI_Trees.exe --subs 2500 --left 1.75 --right 4 --integratee 1
```

Ändern der Methode auf Simpson

Mit Simpson rule (braucht erheblich mehr Stützstellen und ist selbst dann ungenauer)

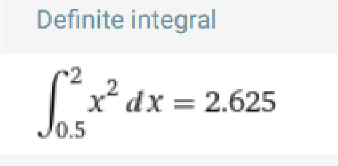
```
mpiexec -n 300 03_MPI_Trees.exe --subs 25000 --left 1.75 --right 4 --
integratee 1 --integrator 1
```



```
The result is: 0.47601009684530005

C:\Data\git\Development\FH\CPP\HPC3\03_MPI_Trees\x64\Debug>mpiexec -n 300 03_MPI_Trees.exe --subs 25000 --left 1.75 --right 4 --integratee 1 --integrator 1
```

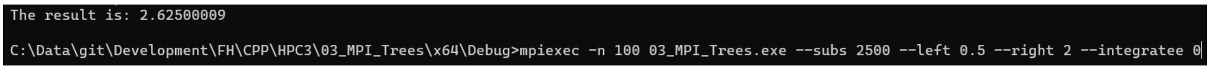
Ändern der zu integrierenden Funktion



Definite integral

$$\int_{0.5}^{2} x^2\,dx = 2.625$$

```
mpiexec -n 100 03_MPI_Trees.exe --subs 2500 --left 0.5 --right 2 --
integratee 0
```

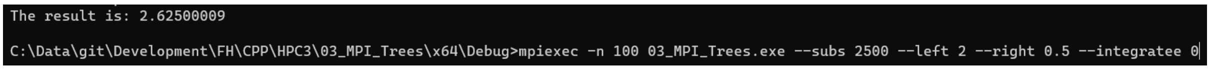


```
The result is: 2.62500009

C:\Data\git\Development\FH\CPP\HPC3\03_MPI_Trees\x64\Debug>mpiexec -n 100 03_MPI_Trees.exe --subs 2500 --left 0.5 --right 2 --integratee 0
```

Funktioniert auch für b < a:

```
mpiexec -n 100 03_MPI_Trees.exe --subs 2500 --left 2 --right 0.5 --
integratee 0
```

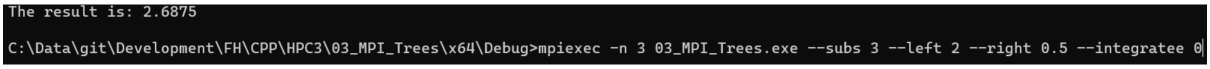


```
The result is: 2.62500009

C:\Data\git\Development\FH\CPP\HPC3\03_MPI_Trees\x64\Debug>mpiexec -n 100 03_MPI_Trees.exe --subs 2500 --left 2 --right 0.5 --integratee 0
```

Zu geringe Anzahl der Stützstellen – Ergebnis wird ungenauer:

```
mpiexec -n 3 03_MPI_Trees.exe --subs 3 --left 2 --right 0.5 --integratee 0
```

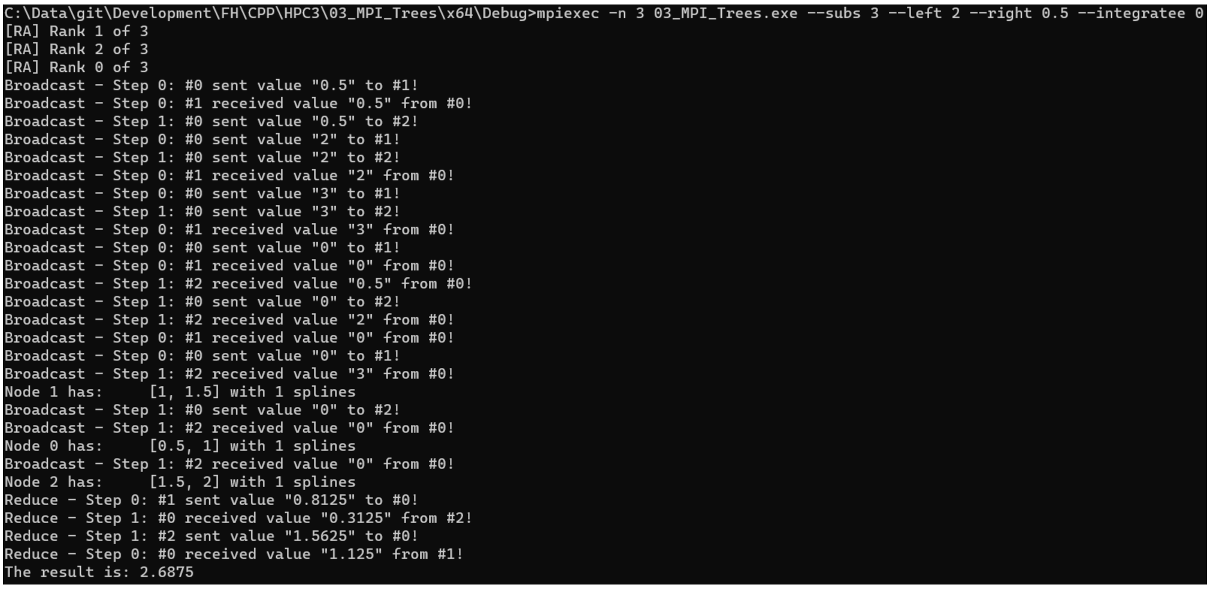


```
The result is: 2.6875

C:\Data\git\Development\FH\CPP\HPC3\03_MPI_Trees\x64\Debug>mpiexec -n 3 03_MPI_Trees.exe --subs 3 --left 2 --right 0.5 --integratee 0
```

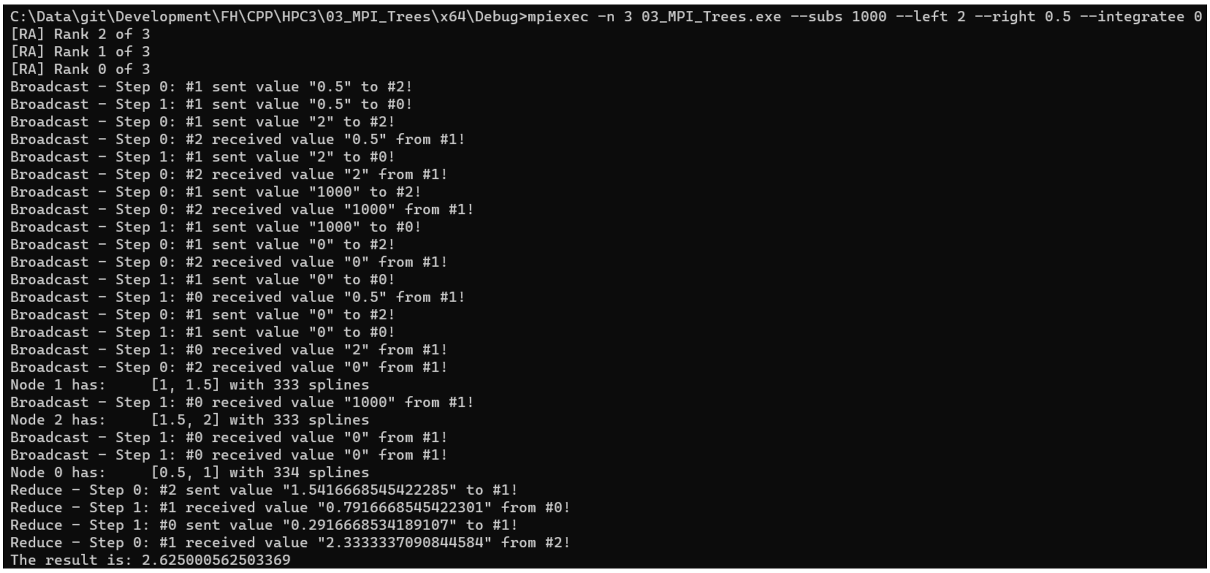
Wie gewollt, werden auch sämtliche Schritte, die MPI betreffen, gelogged:

a) Senden
b) Empfangen
c) Aufteilung der Workload (Intervalle & Splines)
d) Welcher Schritt und Broadcast/Reduce Unterscheidung



```
C:\Data\git\Development\FH\CPP\HPC3\03_MPI_Trees\x64\Debug>mpiexec -n 3 03_MPI_Trees.exe --subs 3 --left 2 --right 0.5 --integratee 0
[RA] Rank 1 of 3
[RA] Rank 2 of 3
[RA] Rank 0 of 3
Broadcast - Step 0: #0 sent value "0.5" to #1!
Broadcast - Step 0: #1 received value "0.5" from #0!
Broadcast - Step 1: #0 sent value "0.5" to #2!
Broadcast - Step 0: #0 sent value "2" to #1!
Broadcast - Step 1: #0 sent value "2" to #2!
Broadcast - Step 0: #1 received value "2" from #0!
Broadcast - Step 0: #0 sent value "3" to #1!
Broadcast - Step 1: #0 sent value "3" to #2!
Broadcast - Step 0: #1 received value "3" from #0!
Broadcast - Step 0: #0 sent value "0" to #1!
Broadcast - Step 0: #1 received value "0" from #0!
Broadcast - Step 1: #2 received value "0.5" from #0!
Broadcast - Step 1: #0 sent value "0" to #2!
Broadcast - Step 1: #2 received value "2" from #0!
Broadcast - Step 0: #1 received value "0" from #0!
Broadcast - Step 0: #0 sent value "0" to #1!
Broadcast - Step 1: #2 received value "3" from #0!
Node 1 has:     [1, 1.5] with 1 splines
Broadcast - Step 1: #0 sent value "0" to #2!
Broadcast - Step 1: #2 received value "0" from #0!
Node 0 has:     [0.5, 1] with 1 splines
Broadcast - Step 1: #2 received value "0" from #0!
Node 2 has:     [1.5, 2] with 1 splines
Reduce - Step 0: #1 sent value "0.8125" to #0!
Reduce - Step 1: #0 received value "0.3125" from #2!
Reduce - Step 1: #2 sent value "1.5625" to #0!
Reduce - Step 0: #0 received value "1.125" from #1!
The result is: 2.6875
```

Es ist zudem möglich (im Quellcode) den root zu ändern. Anders sollte es auch garnicht gehen (?), da der Root sonst auch verteilt werden müsste, dieser aber schon in den must_receive/must_send Berechnungen enthalten ist. Nachfolgend ein Screenshot mit Node 1 als Root:



```
C:\Data\git\Development\FH\CPP\HPC3\03_MPI_Trees\x64\Debug>mpiexec -n 3 03_MPI_Trees.exe --subs 1000 --left 2 --right 0.5 --integratee 0
[RA] Rank 2 of 3
[RA] Rank 1 of 3
[RA] Rank 0 of 3
Broadcast - Step 0: #1 sent value "0.5" to #2!
Broadcast - Step 1: #1 sent value "0.5" to #0!
Broadcast - Step 0: #1 sent value "2" to #2!
Broadcast - Step 0: #2 received value "0.5" from #1!
Broadcast - Step 1: #1 sent value "2" to #0!
Broadcast - Step 0: #2 received value "2" from #1!
Broadcast - Step 0: #1 sent value "1000" to #2!
Broadcast - Step 0: #2 received value "1000" from #1!
Broadcast - Step 1: #1 sent value "1000" to #0!
Broadcast - Step 0: #1 sent value "0" to #2!
Broadcast - Step 0: #2 received value "0" from #1!
Broadcast - Step 1: #1 sent value "0" to #0!
Broadcast - Step 1: #0 received value "0.5" from #1!
Broadcast - Step 0: #1 sent value "0" to #2!
Broadcast - Step 1: #1 sent value "0" to #0!
Broadcast - Step 1: #0 received value "2" from #1!
Broadcast - Step 0: #2 received value "0" from #1!
Node 1 has:     [1, 1.5] with 333 splines
Broadcast - Step 1: #0 received value "1000" from #1!
Node 2 has:     [1.5, 2] with 333 splines
Broadcast - Step 1: #0 received value "0" from #1!
Broadcast - Step 1: #0 received value "0" from #1!
Node 0 has:     [0.5, 1] with 334 splines
Reduce - Step 0: #2 sent value "1.5416668545422285" to #1!
Reduce - Step 1: #1 received value "0.7916668545422301" from #0!
Reduce - Step 1: #0 sent value "0.2916668534189107" to #1!
Reduce - Step 0: #1 received value "2.3333337090844584" from #2!
The result is: 2.625000562503369
```