Aufgabe 1

$numRows = Interlines * (n - 1) + n$

$numThreads = options \rightarrow number$

$$numRowsForThread(i) = \begin{cases} \lfloor \frac{numRows}{numThreads} \rfloor + 1, & \text{wenn } i < (numRows \mod numThreads) \\ \lfloor \frac{numRows}{numThreads} \rfloor, & \text{wenn } i \geq (numRows \mod numThreads) \\ 0, & \text{sonst} \end{cases}$$

$$startzeile(i) = 1 + i * \lfloor \frac{numRows}{numThreads} \rfloor + min(i, (numRows \mod numThreads))$$

$endzeile(i) = startzeile(i) + numRowsForThread(i) - 1$

i = 0
T = 5
numRows = 9 - 2(calculate() iteriert nicht über Randzeilen bzw. Spalten), numThreads = 5

t	rows_per_thread	start_row	end_row
0	⌊(0*8+7)/5⌋ + 1 = 2	1 + 0 * 1 + 0 = 1	1 + 2 - 1 = 2
1	1 + 1 = 2	1 + 1 * 1 + 1 = 3	3 + 2 - 1 = 4
2	1 = 1	1 + 2 * 1 + 2 = 5	5 + 1 - 1 = 5
3	1	1 + 3 * 1 + 2 = 6	6 + 1 - 1 = 6
4	1	1 + 4 * 1 + 2 = 7	7 + 1 - 1 = 7

Threads	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
unbeachtet	0	0	0	0	0	0	0	0	0

i = 1
T = 5
numRows = 15, numThreads = 5

t	rows_per_thread	start_row	end_row
0	⌊(1*8+7)/5⌋ + 0 = 3	1 + 0 * 3 + 0 = 1	1 + 3 - 1 = 3
1	3 + 0 = 3	1 + 1 * 3 + 0 = 4	4 + 3 - 1 = 6
2	3 + 0 = 3	1 + 2 * 3 + 0 = 7	7 + 3 - 1 = 9
3	3 + 0 = 3	1 + 3 * 3 + 0 = 10	10 + 3 - 1 = 12
4	3 + 0 = 3	1 + 4 * 3+ 0 = 13	13 + 3 - 1 = 15

Threads															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
unbeachtet	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Aufgabe 3

Pre-Set: ./partdiff-posix t 2 512 2 2 1000
paralleles Programm

threads	Messung 1	Messung 2	Messung 3	Mittelwert
1	416.81	416.67	416.81	416.76
2	208.59	208.76	208.70	208.68
3	140.21	140.17	140.21	140.20
4	105.01	105.07	105.20	105.10
5	84.44	83.94	84.00	84.13
6	70.19	70.20	70.22	70,20
7	60.17	60.20	60.36	60.24
8	52.91	52.78	52.80	52.83
9	47.26	48.01	47.16	47.48
10	42.60	42.70	42.66	42.65
11	40.08	38.87	39.30	39.42
12	36.28	37.14	37.92	37.11

Pre-Set: ./partdiff-posix 1 2 512 2 2 1000
sequentielles Programm

thread	Messung 1	Messung 2	Messung 3	Mittelwert
1	412.56	412.29	412.57	412.47

Anhand des Diagrammes sieht man, dass der Graf ein nahezu hyperbolisches Wachstum hat. Dies liegt daran, dass die Laufzeit sich stetig verringert, aber sich für steigende Thread-Anzahl einem Grenzwert nähert. Begründen kann man das an Amdahls Gesetz, welches besagt, dass der maximale Speedup an den sequentiellen Anteil f des Programmes gebunden ist

$$S_{\max} = \frac{1}{f}$$