

Sheet 03

PS Parallel Programming

Patrick Wintner

March 24, 2025

1 Mandelbrot

The execution time of the program mandelbrot is measured.

1.1 Source Code

```
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

// Include that allows to print result as an image
#define STB_IMAGE_WRITE_IMPLEMENTATION
#include "stb_image_write.h"

// Default size of image
#define X 1280
#define Y 720
#define MAX_ITER 10000

void calc_mandelbrot(uint8_t image[Y][X]) {
    for(size_t i=0; i<Y; ++i) {
        for(size_t j=0; j<X; ++j) {
            double x=0;
            double y=0;
            double cx=(double)j/(X-1)*3.5-2.5; // scale j to [-2.5, 1]
```

```

        double cy=(double)i/(Y-1)*2-1; // scale i to [-1. 1]
        size_t iteration=0;
        while(x*x+y*y<2*2 && iteration<MAX_ITER) {
            double x_tmp=x*x-y*y+cx;
            y=2*x*y+cy;
            x=x_tmp;
            iteration=iteration+1;
        }
        char norm_iteration=iteration*255/MAX_ITER; // scale iteration to [0. 255]
        image[i][j]=norm_iteration;
    }
}

int main() {
    uint8_t image[Y][X];

    calc_mandelbrot(image);

    const int channel_nr = 1, stride_bytes = 0;
    stbi_write_png("mandelbrot.png", X, Y, channel_nr, image, stride_bytes);
    return EXIT_SUCCESS;
}

```

1.2 Measurement Method

The measurements were taken using the script mandelbrot/main.sh:

```

#!/bin/bash
# Usage: ./main.sh
MEASUREMENT_RESULTS=mandelbrot_measurements.log
PROCESSED_RESULTS=mandelbrot_processed.log
make
rm $MEASUREMENT_RESULTS $PROCESSED_RESULTS
for ((i=0;i<=2;++i)); do
    /bin/time -f "%e" -a -o $MEASUREMENT_RESULTS ./mandelbrot
done
./process_results $MEASUREMENT_RESULTS $PROCESSED_RESULTS
make clean

```

The measurement results are stored in mandelbrot_measurements.log, which is read by process_results to compute the average execution time and standard deviation, which

are stored in mandelbrot_processed.log.

1.3 Measurement Results

time/s	mean/s	standard deviation/s
17.77	17.74	0.0265
17.72		
17.73		

1.4 Suggestions for performance improvement and parallelisation

The calculations of the colour of different pixels are independent, therefore those calculations could be done parallel.

2 False Sharing