EPSRC Vacation Scheme: Week 1 Review

Matthew Knowles

Department of Mathematics
University of York

mk1320@york.ac.uk

26^{th} July 2021

Goals of last week

1. Implement 'by eye' algorithm for finding convex hull of piecewise linear functions.

Goals of last week

1. Implement 'by eye' algorithm for finding convex hull of piecewise linear functions.
2. Look around in literature to see what I could find.

Task 1 Progress

- Wrote this in roughly 60 lines of code

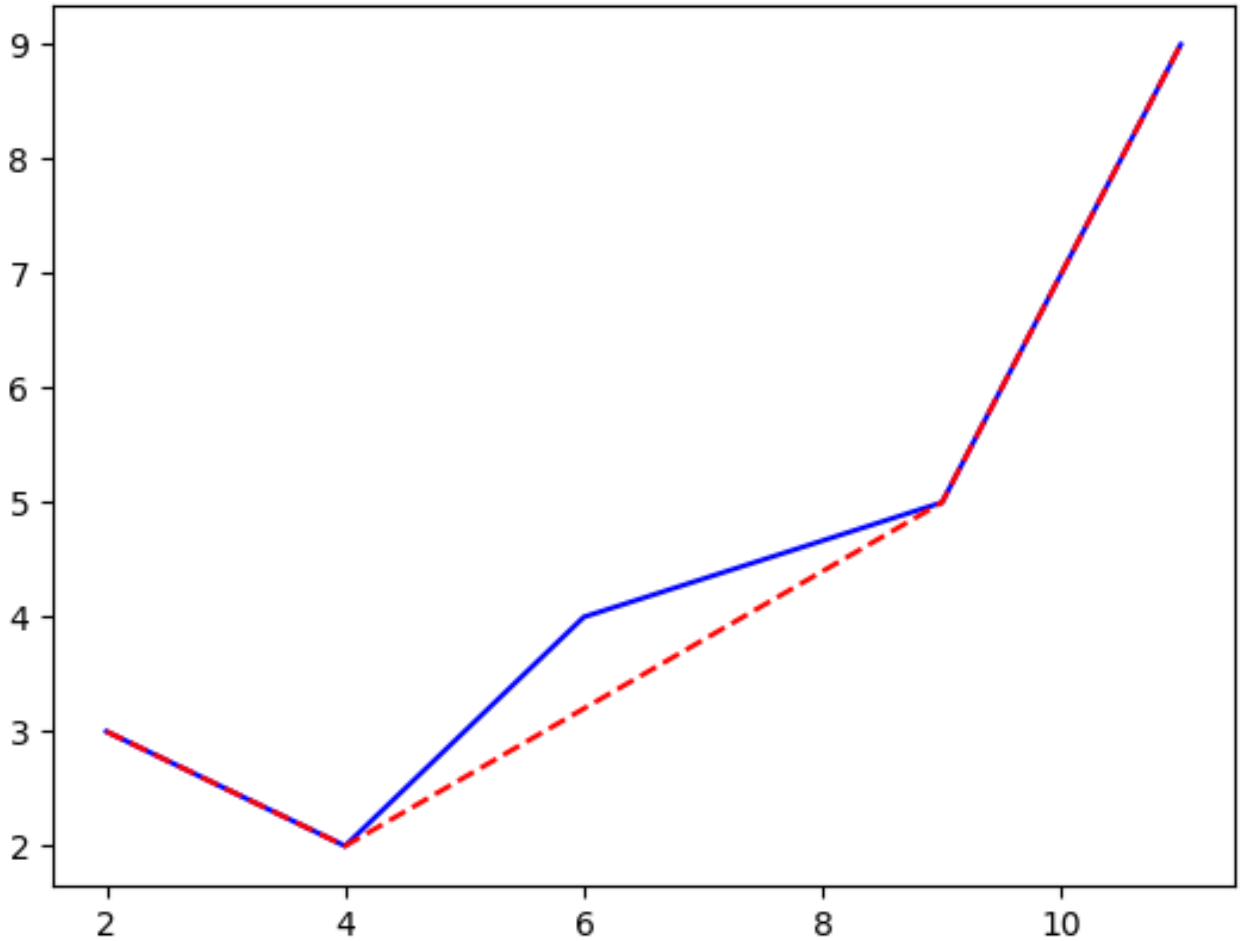
Task 1 Progress

- Wrote this in roughly 60 lines of code
- Tested on a few test datasets. Here are those results

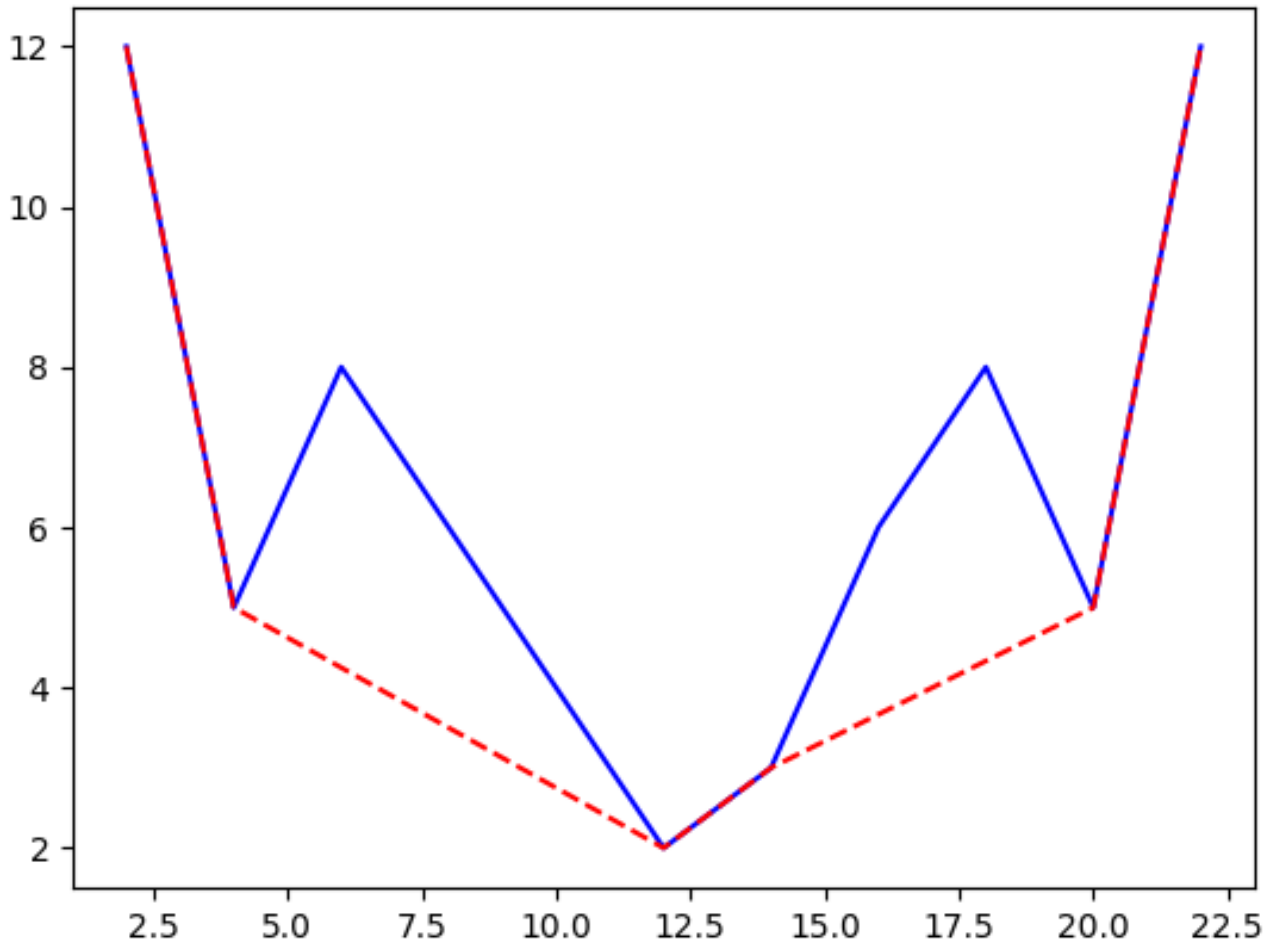
Task 1 Progress

- Wrote this in roughly 60 lines of code
- Tested on a few test datasets. Here are those results
- You will notice that they are not perfect

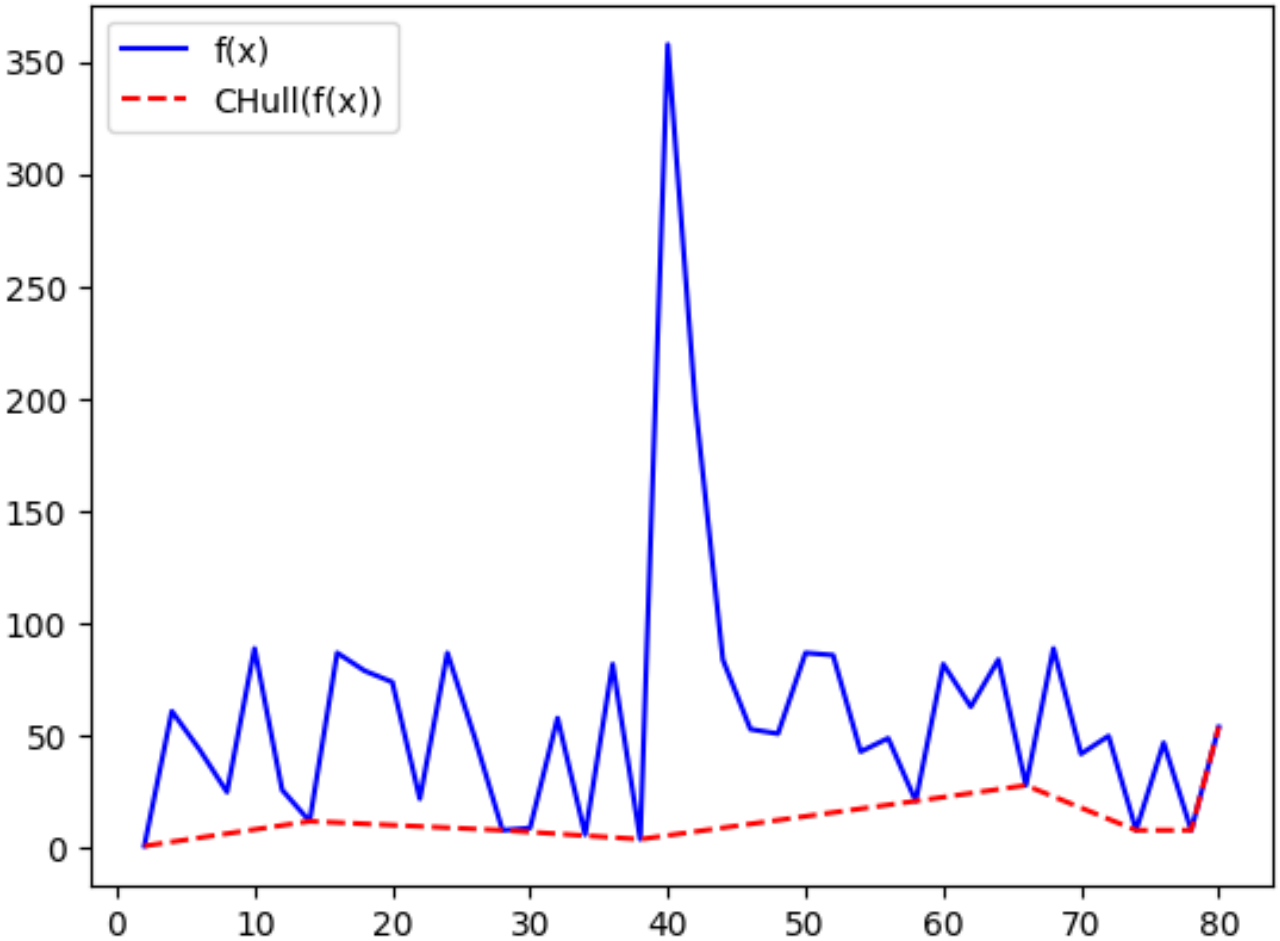
First test



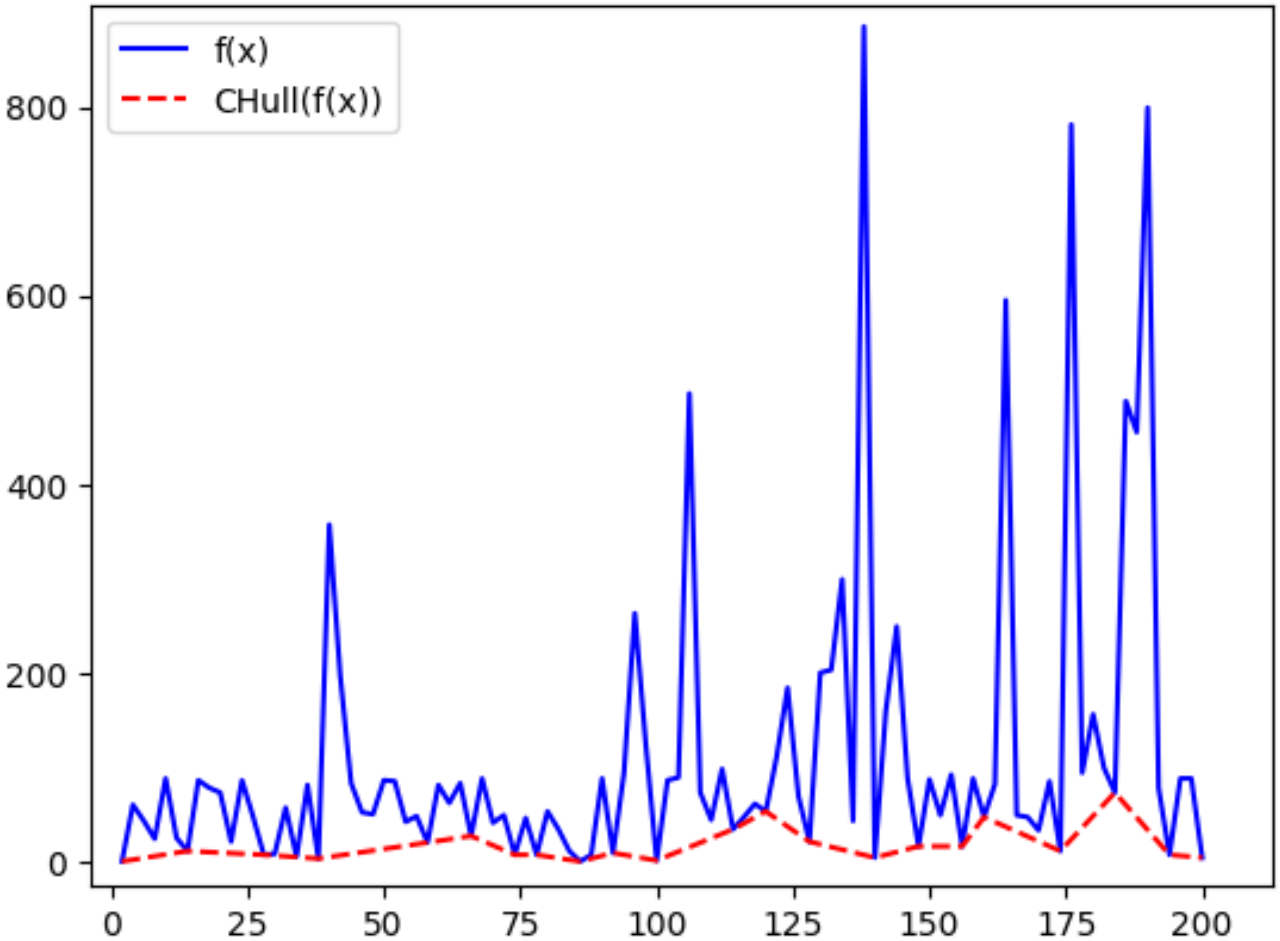
Slightly larger test



Test with 100 data points



Test with 200 data points



Problems

- Notice how the function it computes isn't entirely convex

Problems

- Notice how the function it computes isn't entirely convex
- Can remove this issue by recursively running through the hull

Problems

- Notice how the function it computes isn't entirely convex
- Can remove this issue by recursively running through the hull
- issue: this is going to get increasingly slower and slower

Problems

- Notice how the function it computes isn't entirely convex
- Can remove this issue by recursively running through the hull
- issue: this is going to get increasingly slower and slower
- Solution? Try another algo!

Monotone Chain

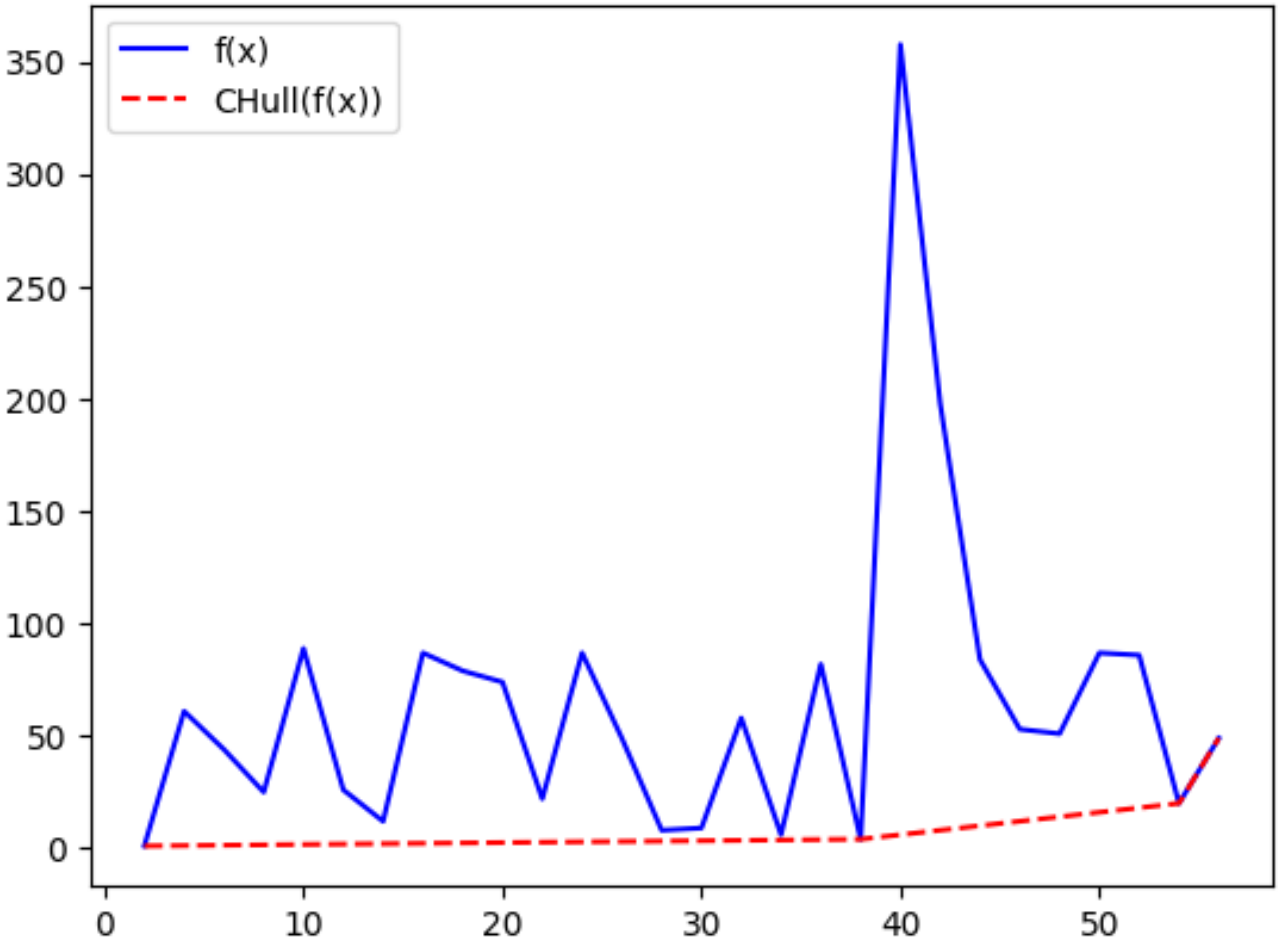
- The Monotone Chain Algorithm runs in $\mathcal{O}(n \log n)$ time

Monotone Chain

- The Monotone Chain Algorithm runs in $\mathcal{O}(n \log n)$ time
- Usually runs a sub-routine on upper and lower part of the set, we are only interested in the lower part however

Monotone Chain

- The Monotone Chain Algorithm runs in $\mathcal{O}(n \log n)$ time
- Usually runs a sub-routine on upper and lower part of the set, we are only interested in the lower part however
- Only looking at one hull reduces to $\mathcal{O}(n)$ complexity


f(x)
CHull(f(x))
350
300
250
200
150
100
50
0
0
10
20
30
40
50

Any Questions?