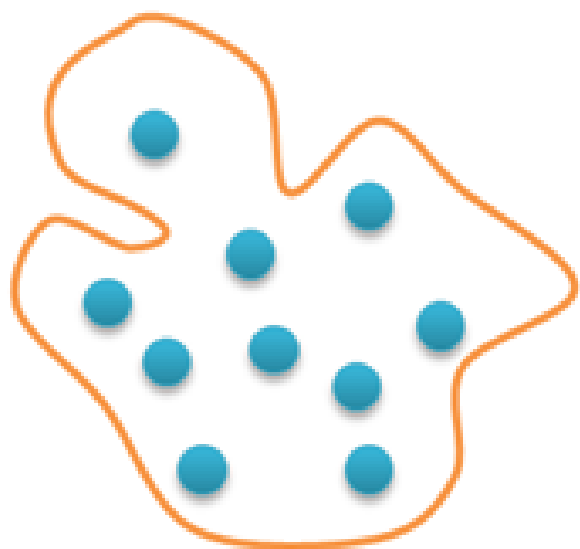


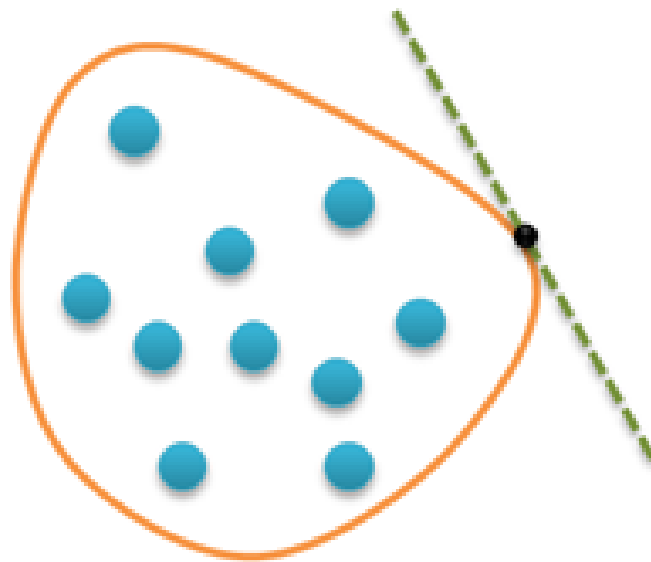
Реализация алгоритмов построения минимальной выпуклой оболочки

Выполнил студент группы ФН2-31Б Пиневич Владислав

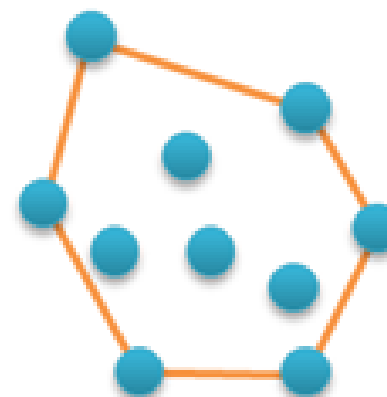
Выпуклая оболочка



Оболочка

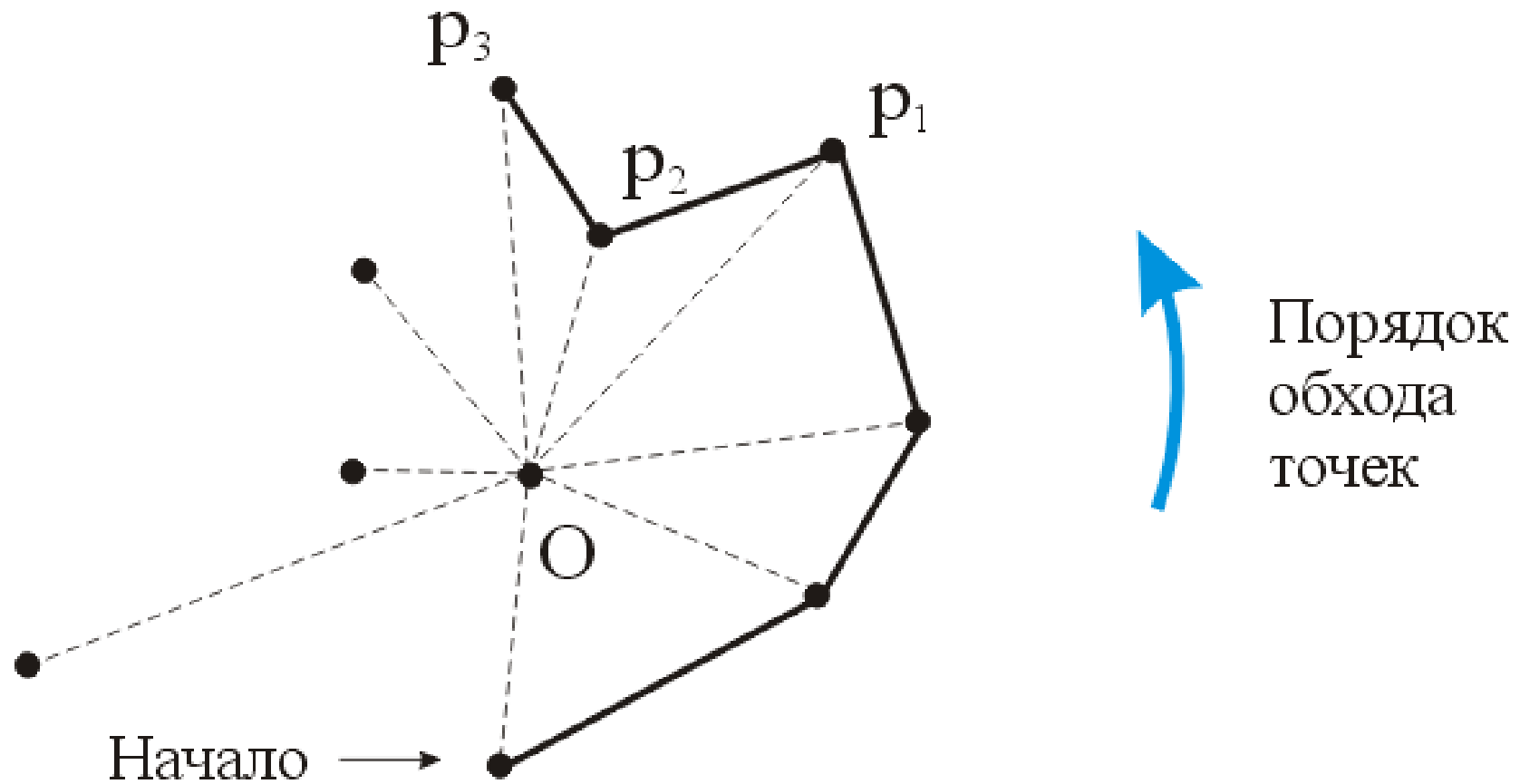


Выпуклая оболочка

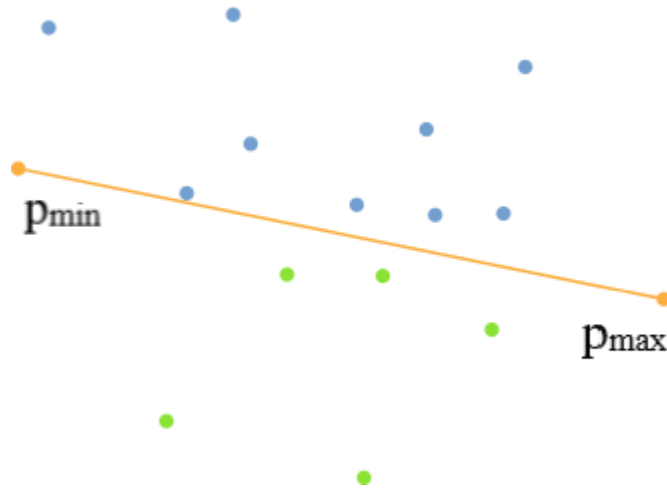


Минимальная
выпуклая оболочка

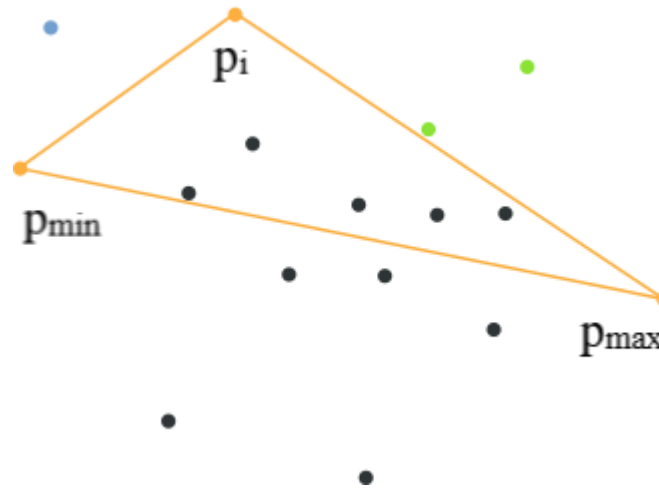
Метод перебора



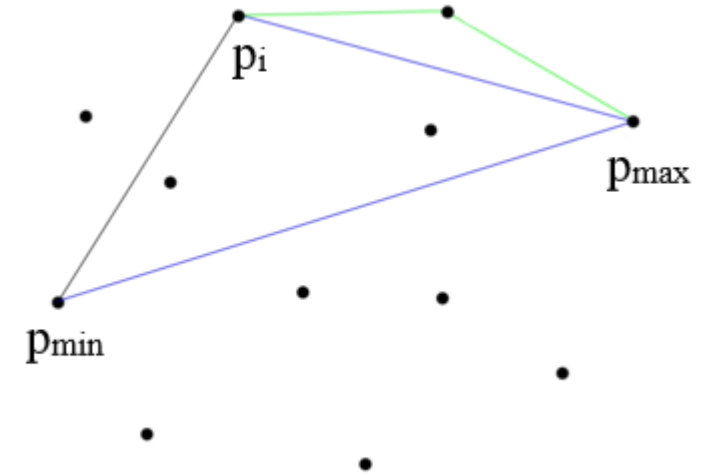
Алгоритм быстрой оболочки



1. Найдем самую левую точку p_{min} и самую правую точку p_{max} . Если таких несколько, то самую нижнюю и верхнюю соответственно

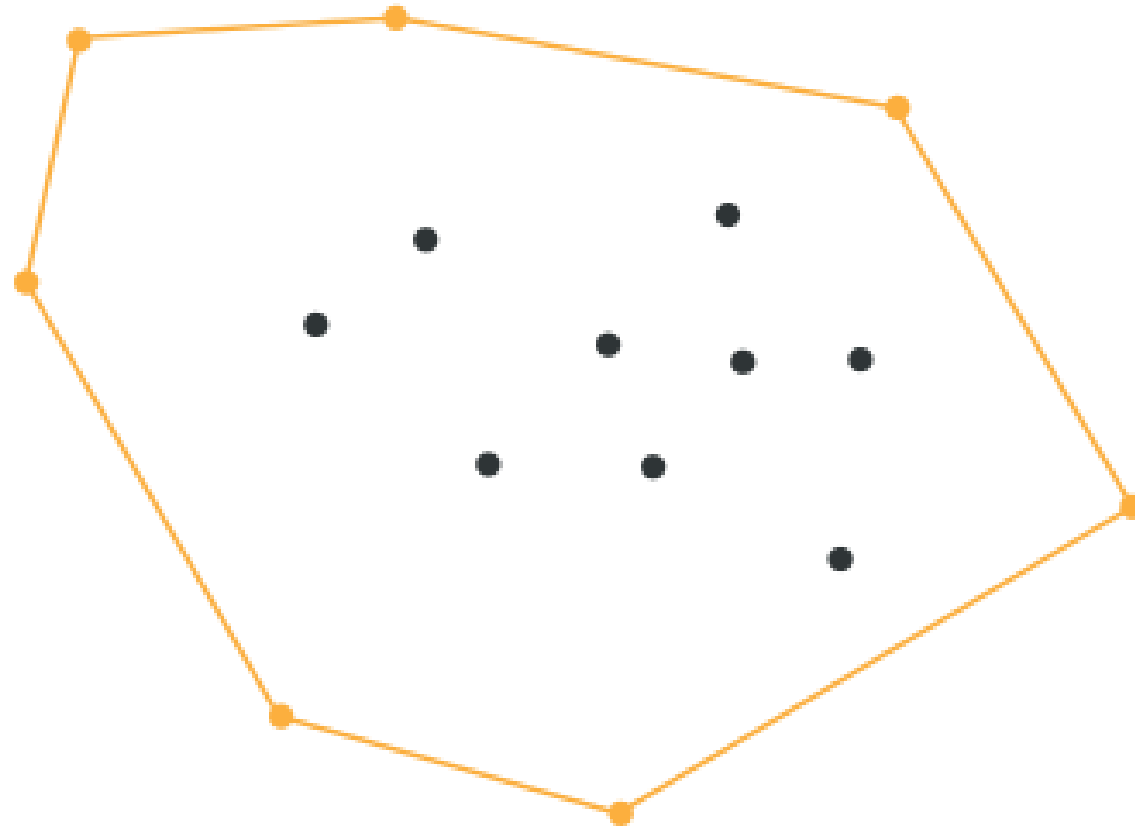


2. Найдем среди множества выше прямой $p_{min} p_{max}$ точку p_i , наиболее отдаленную данной от прямой

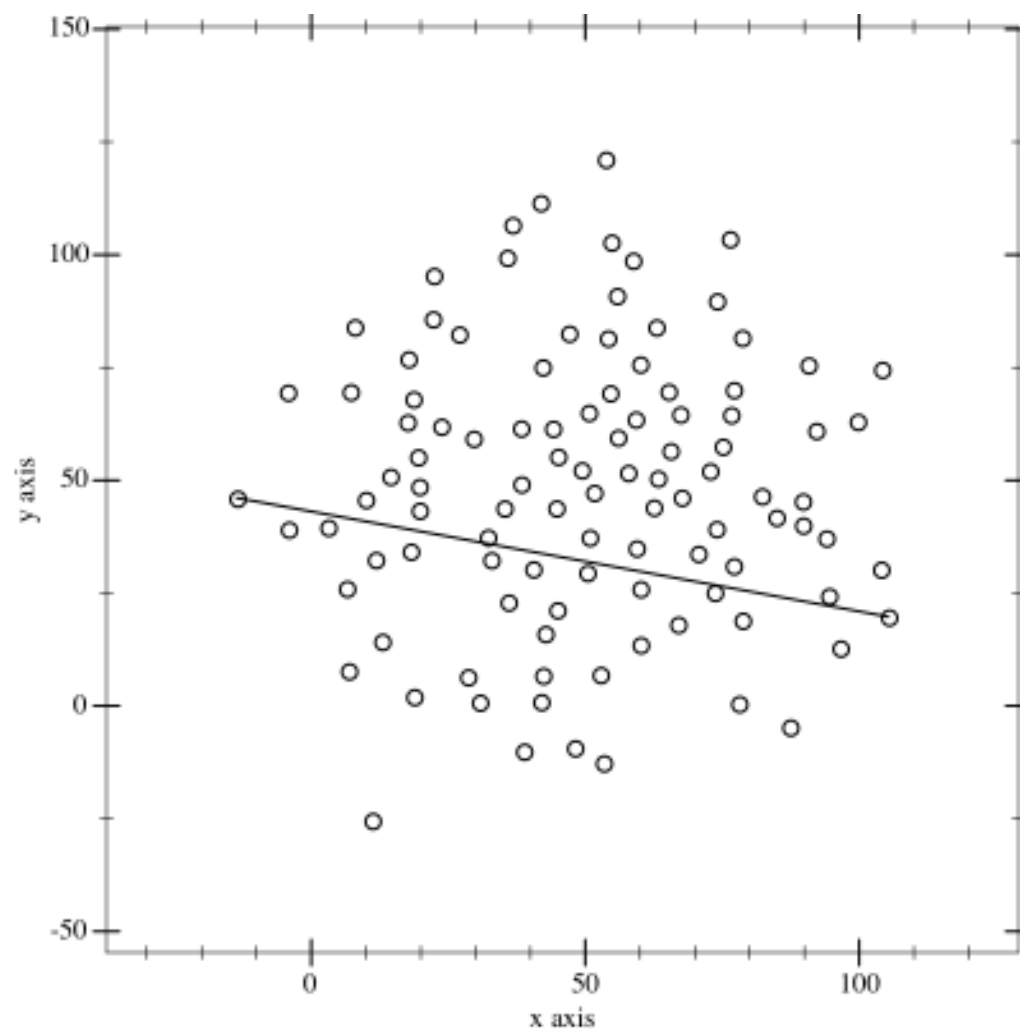


3. Рекурсивно повторить шаги 1-2 для прямых $p_{min} p_i$ и $p_i p_{max}$, пока есть точки выше каждой из прямых соответственно

Результат работы алгоритма быстрой оболочки



Работа алгоритма быстрой оболочки



Пример работы программы на C++

```
30
-498.749 63.5853
-306.696 308.741
85.0093 -20.127
-149.709 395.962
322.84 246.605
-325.892 358.943
210.501 13.535
-196.005 -485.015
-408.597 -135.548
-352.687 -334.101
488.525 -54.3077
-380.917 -495.331
-491.089 -122.12
31.663 71.1844
101.764 107.166
-333.766 163.045
-49.2111 -147.877
-442.961 107.685
283.319 302.606
19.8828 -198.05
375.973 226.676
455.901 425.718
39.3536 -357.662
-37.9192 -264.672
362.239 -290.399
279.656 343.654
496.796 499.695
111.499 -107.562
-233.787 -202.719
340.144 -476.257
```

Исходные данные

Результаты

1	8
2	496.7959999999999 499.6949999999999
3	-149.709 395.9619999999999
4	-325.892 358.9429999999998
5	-498.74900000000002 63.585299999999997
6	-491.089 -122.12
7	-380.91699999999997 -495.33100000000002
8	340.14400000000001 -476.25700000000001
9	488.52499999999998 -54.307699999999997

Перебор

1	8
2	-380.91699999999997 -495.33100000000002
3	-491.089 -122.12
4	-498.74900000000002 63.585299999999997
5	-325.892 358.94299999999998
6	-149.709 395.9619999999999
7	496.7959999999999 499.6949999999999
8	488.52499999999998 -54.307699999999997
9	340.14400000000001 -476.25700000000001

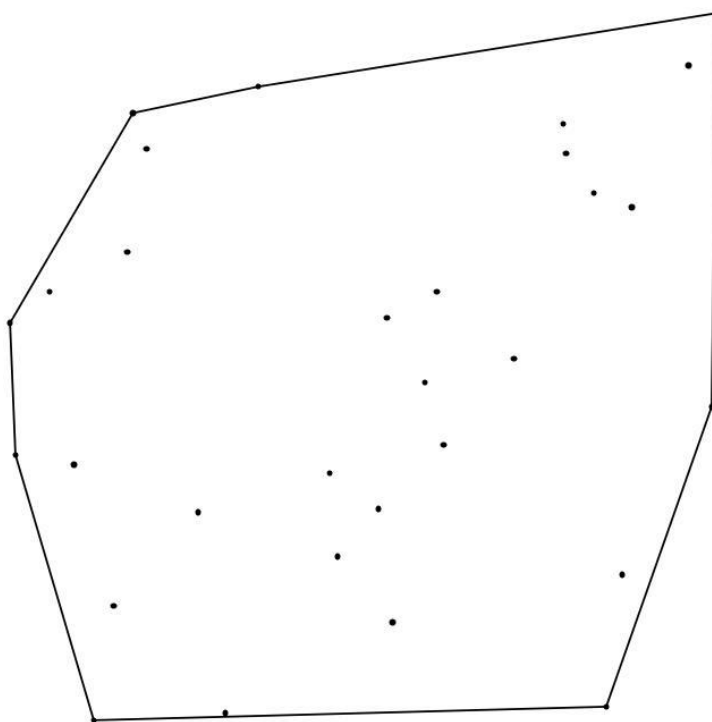
Quickhull

Пример работы программы в Wolfram Mathematica

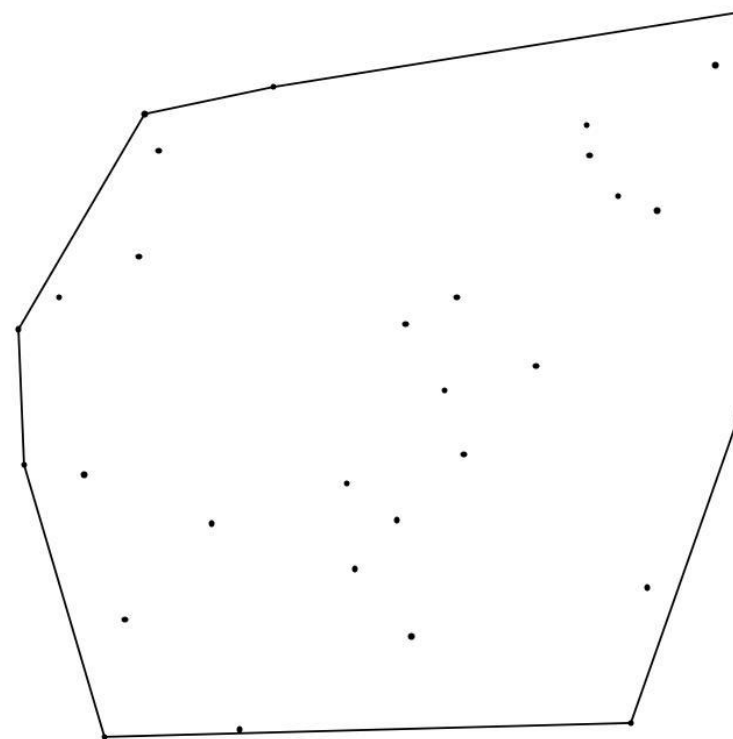
```
30  
-498.749 63.5853  
-306.696 308.741  
85.0093 -20.127  
-149.709 395.962  
322.84 246.605  
-325.892 358.943  
210.501 13.535  
-196.005 -485.015  
-408.597 -135.548  
-352.687 -334.101  
488.525 -54.3077  
-380.917 -495.331  
-491.089 -122.12  
31.663 71.1844  
101.764 107.166  
-333.766 163.045  
-49.2111 -147.877  
-442.961 107.685  
283.319 302.606  
19.8828 -198.05  
375.973 226.676  
455.901 425.718  
39.3536 -357.662  
-37.9192 -264.672  
362.239 -290.399  
279.656 343.654  
496.796 499.695  
111.499 -107.562  
-233.787 -202.719  
340.144 -476.257
```

Исходные данные

Результаты



Перебор



Quickhull

Сравнение времени исполнения в C++ и Wolfram Mathematica

```
1000 точек
1st test
Quick time: 0.002510
NonEffective time: 0.002567
W_Quick time: 0.59375
W_NonEffective time: 2.03125
```

```
2nd test
Release:
Quick time: 0.002545
NonEffective time: 0.002522
```

```
Debug:
Quick time: 0.009130
NonEffective time: 0.026308
W_Quick time: 0.609375
W_NonEffective time: 2.0625
```

```
10.000 точек
1st test
Quick time: 0.013101
NonEffective time: 0.015306
W_Quick time: 8.73438
W_NonEffective time: 31.75
```

```
2nd test
Release:
Quick time: 0.013179
NonEffective time: 0.014543
```

```
Debug:
Quick time: 0.079462
NonEffective time: 0.354602
W_Quick time: 8.79688
W_NonEffective time: 32.0781
```

```
100.000 точек
Release:
Quick time: 0.115154
NonEffective time: 0.115891
```

```
Debug:
Quick time: 0.768925
NonEffective time: 1.503285
```

```
1.000.000 точек
Release:
1st test
Quick time: 0.982837
NonEffective time: 1.771377
```

```
2nd test
Quick time: 0.970619
NonEffective time: 1.765405
```