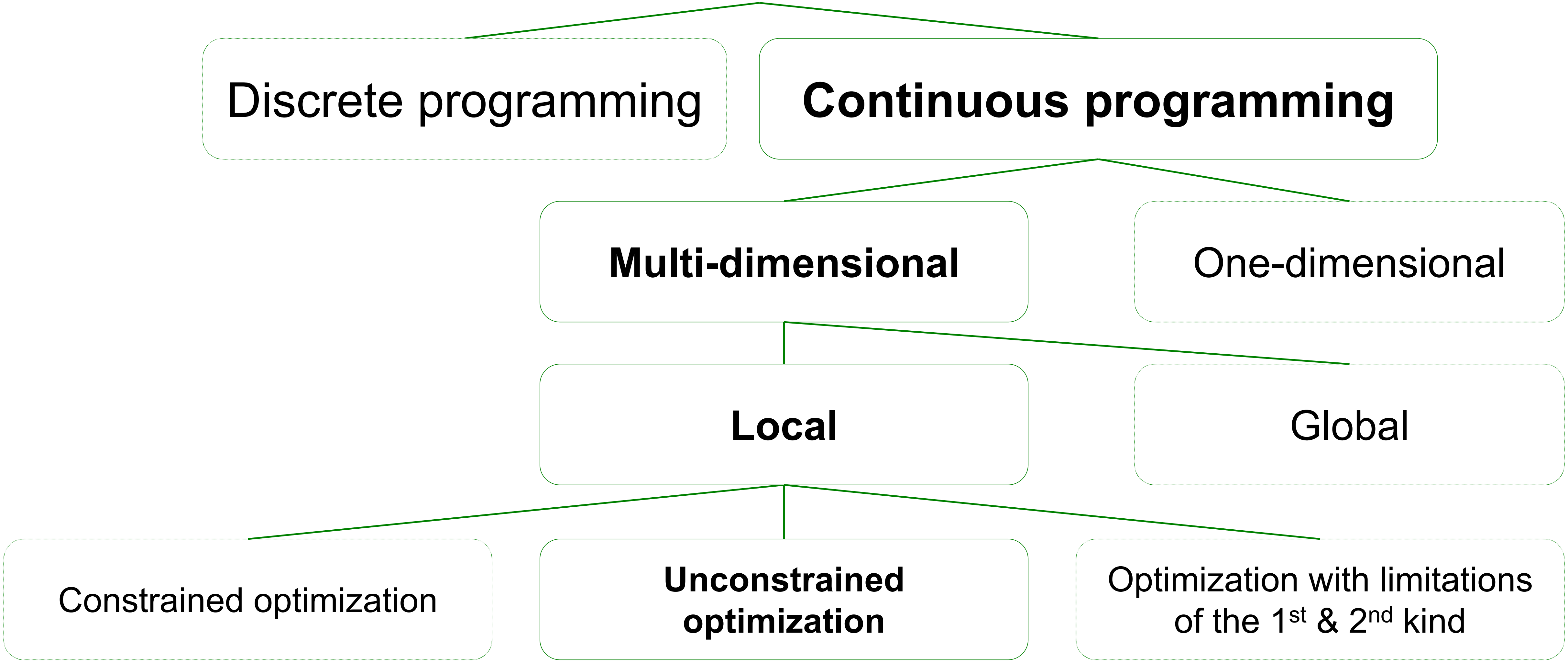
Gradient-based numerical optimization methods

Comparison and implementation to real data-science and economic problems

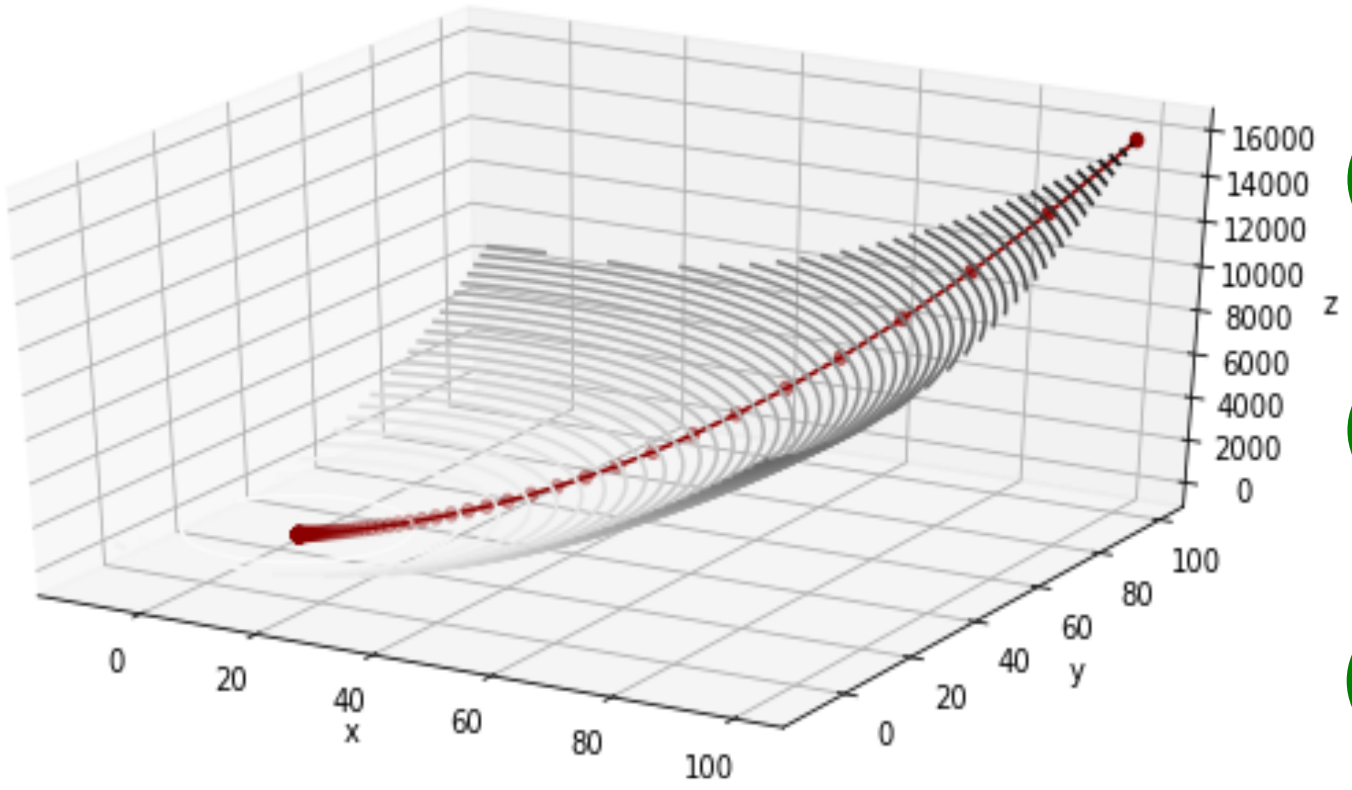
Aims of the research

- Classify optimization problems
- Find pros and cons of the chosen algorithm
- Realize 4 methods of the chosen class
- Apply methods to linear regression problem and analyze the speed of algorithms

Optimization



Gradient descent allocation



1. **Set a function**
   $f(x, y) = (x-5)^2 + (y-17)^2$

2. **Choose staring point** (x_0, y_0) **;**
   number of iterations (N);
   step length λ

3. **Find gradient vector**
   $\nabla f(x, y)$

4. **Update the starting point**
   $(x_1, y_1) = (x_0, y_0) - \lambda \nabla f(x_0, y_0)$

Repeat until $\|\nabla(\boldsymbol{x_0}, \boldsymbol{y_0})\| < \varepsilon$

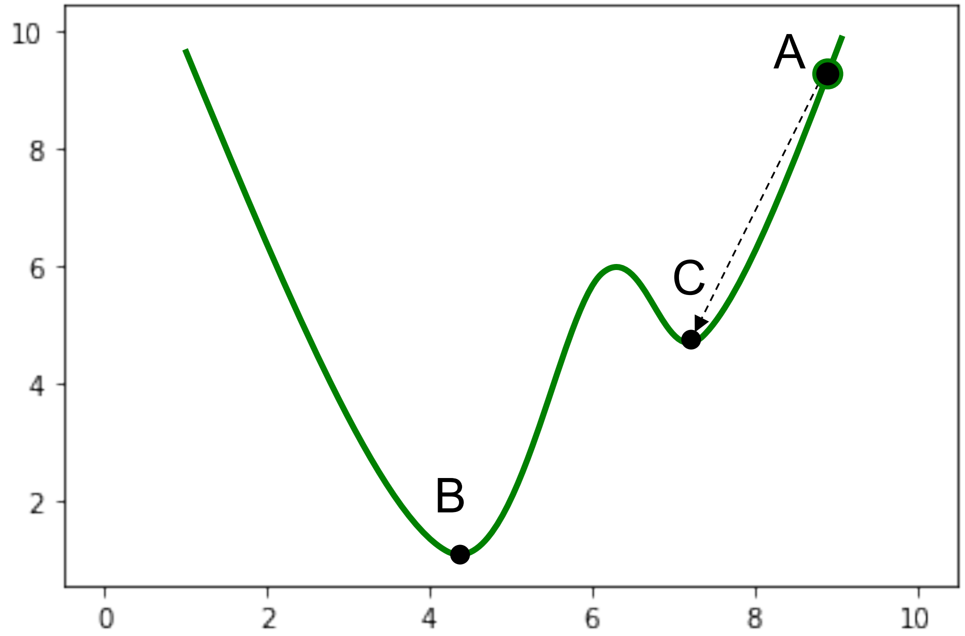
Problem 1

Correct choice of the step length

Correct choice of the number of iterations

Correct choice of the starting point

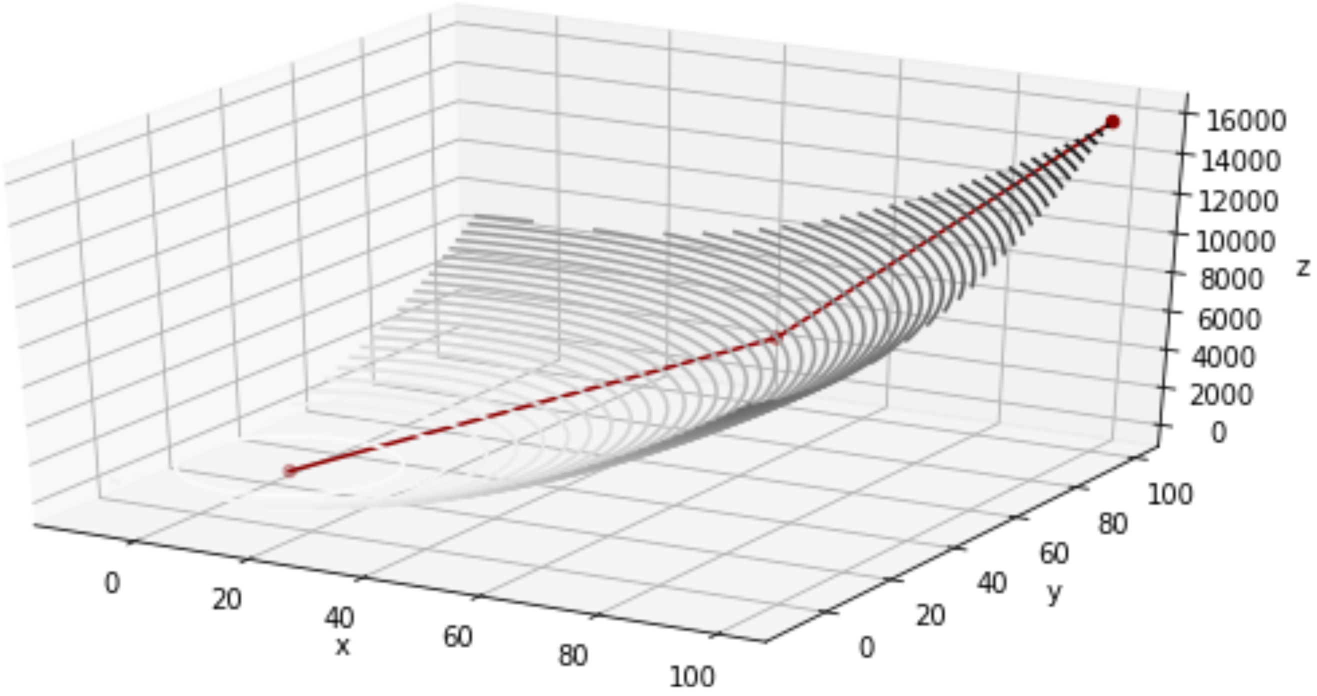
Problem 2



- Assume there is a function
- Starting point is A
- The nearest optimum to A is C
- Global optimum is B

Gradient descent fails while searching for global optimum

Two-point step size method



1 $f(x,y) = (x-5)^2 + (y-17)^2$

2 (x_0, y_0); N; λ_k

3 $\lambda_k = s_{k-1}^T y_{k-1} / \|y_{k-1}\|_2^2$

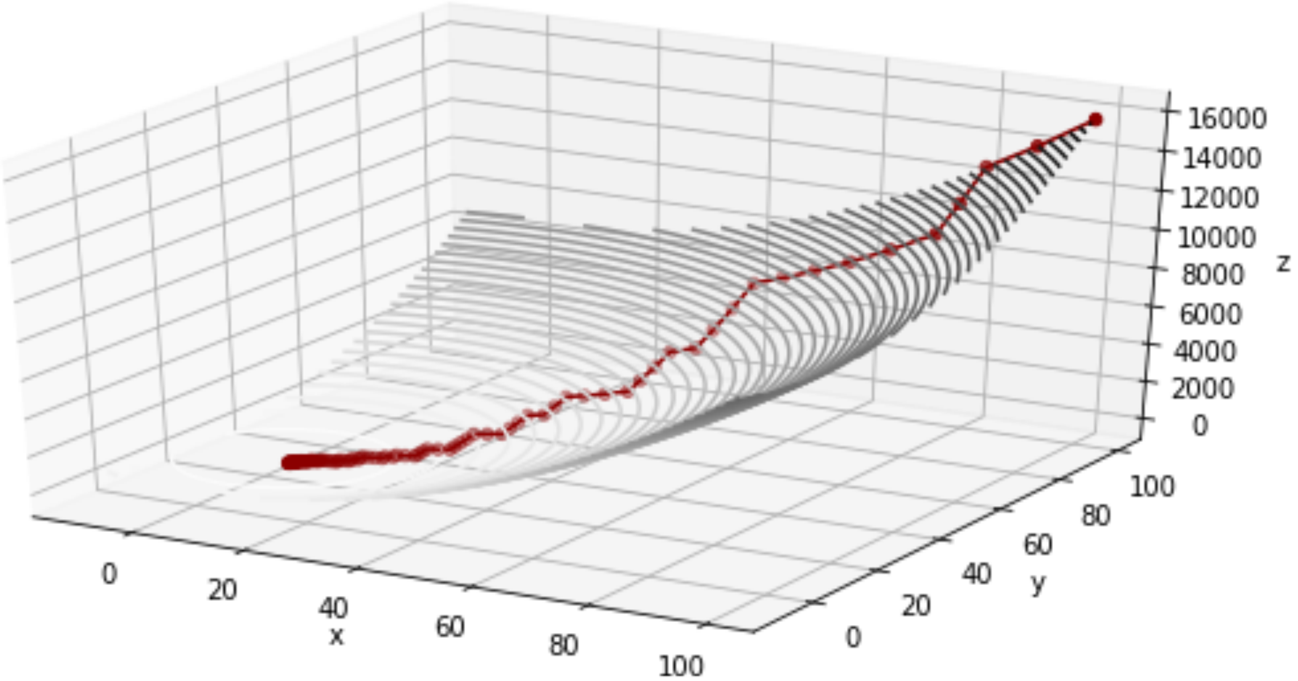
$s_{k-1} = x_k - x_{k-1},$

$y_{k-1} = x_k - \|\nabla f(x_k) - \nabla f(x_{k-1})\|$

4 $(x_1, y_1) = (x_0, y_0) - \lambda_k \nabla f(x_0, y_0)$

Repeat until $\|(\boldsymbol{x_0}, \boldsymbol{y_0})\| < \varepsilon$

Stochastic gradient descent



1. $f(x,y) = (x-5)^2 + (y-17)^2$

2. (x_0, y_0); N; λ

3. $\nabla f(x_0, y_0) = \nabla f(0, y_0)$

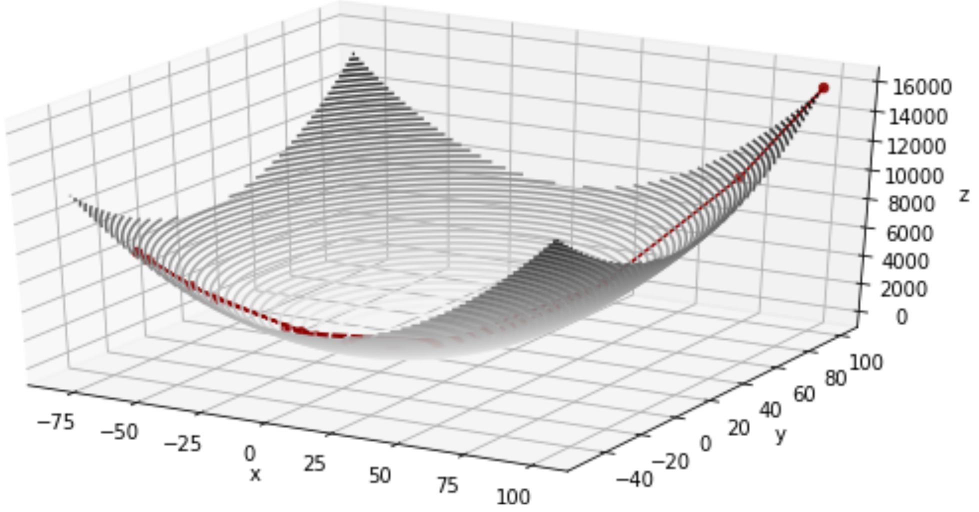
   or

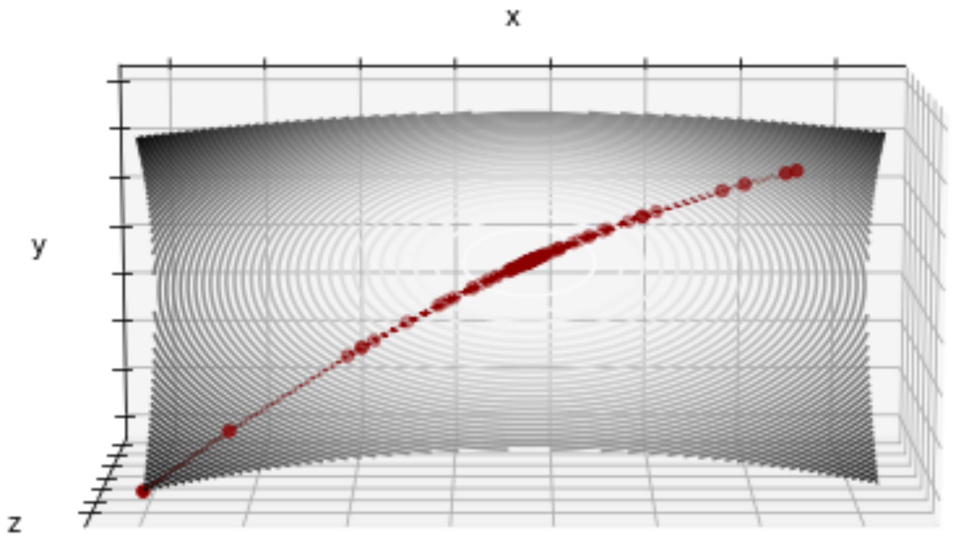
   $\nabla f(x_0, y_0) = \nabla f(x_0, 0)$

4. $(x_1, y_1) = (x_0, y_0) - \lambda \nabla f(x_0, y_0)$

Repeat until $\|(\boldsymbol{x_0}, \boldsymbol{y_0})\| < \varepsilon$

Heavy-ball method





1. $f(x, y) = (x-5)^2 + (y-17)^2$

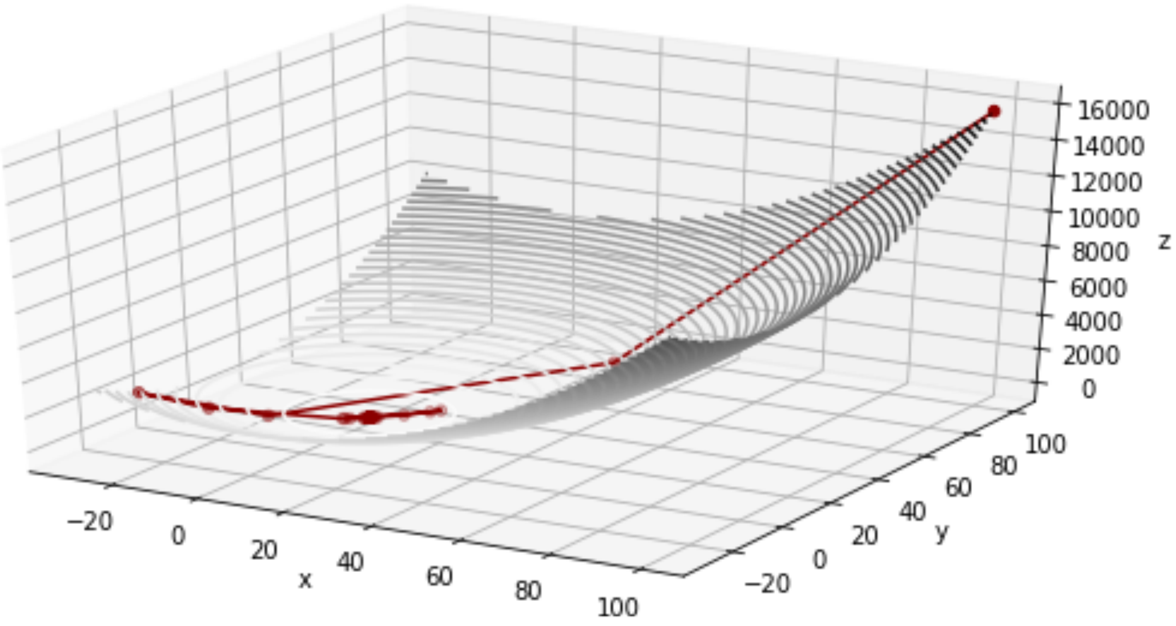
2. (x_0, y_0); N; λ

3. $\nabla f(x_0, y_0))$

4. $x_{k+1} = x_k - \lambda \nabla f(x_k) + \beta(x_k - x_{k-1})$

Repeat until $\|(x_0, y_0)\| < \varepsilon$

Nesterov accelerated gradient



1. $f(x, y) = (x-5)^2 + (y-17)^2$

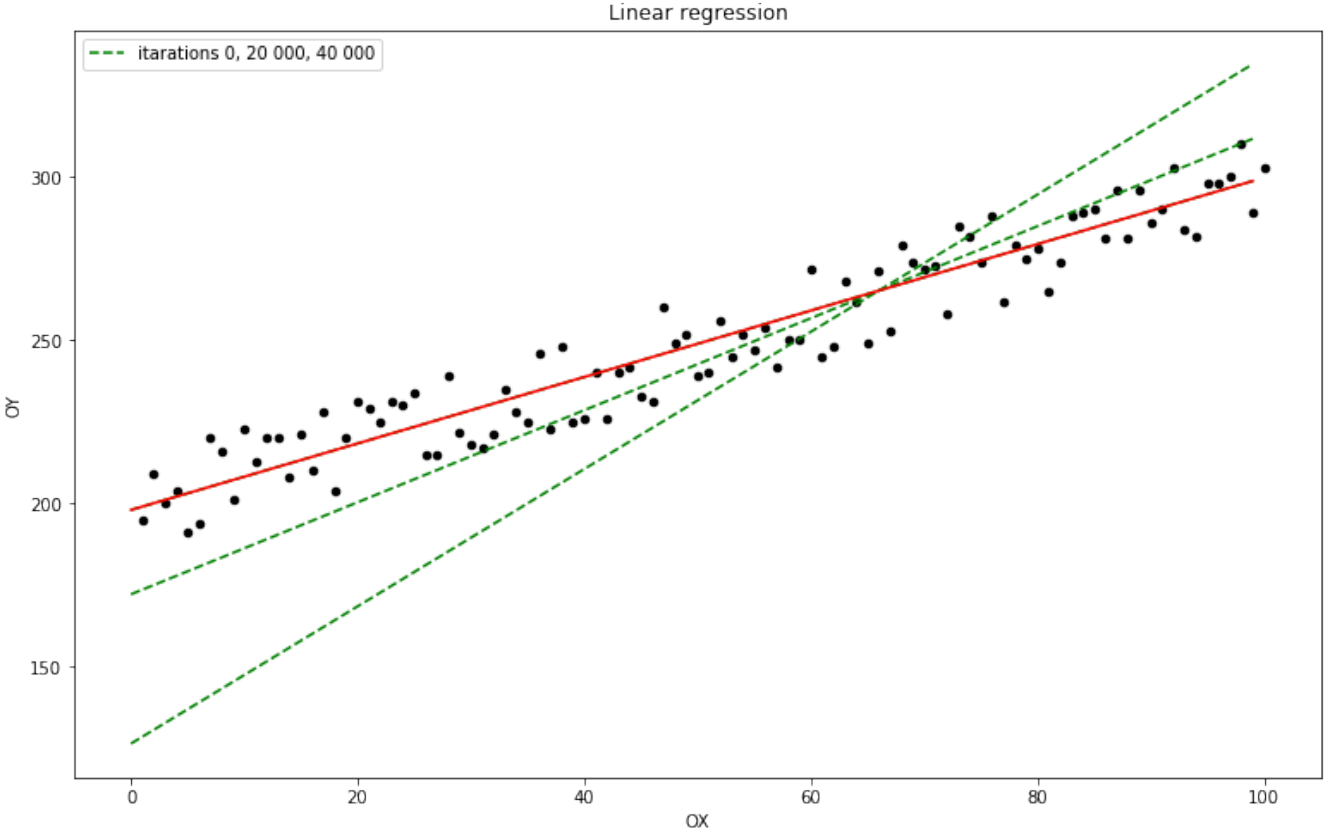
2. (x_0, y_0); N; λ

3. $\nabla f(x_0, y_0))$

4. $x_{k+1} = x_k - \lambda \nabla f\big(x_k + \beta(x_k - x_{k-1})\big) + \beta(x_k - x_{k-1})$

Repeat until $\|(\boldsymbol{x_0}, \boldsymbol{y_0})\| < \varepsilon$

Linear regression



1. $f\,(x_1, x_2, \dots\, x_n) = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$

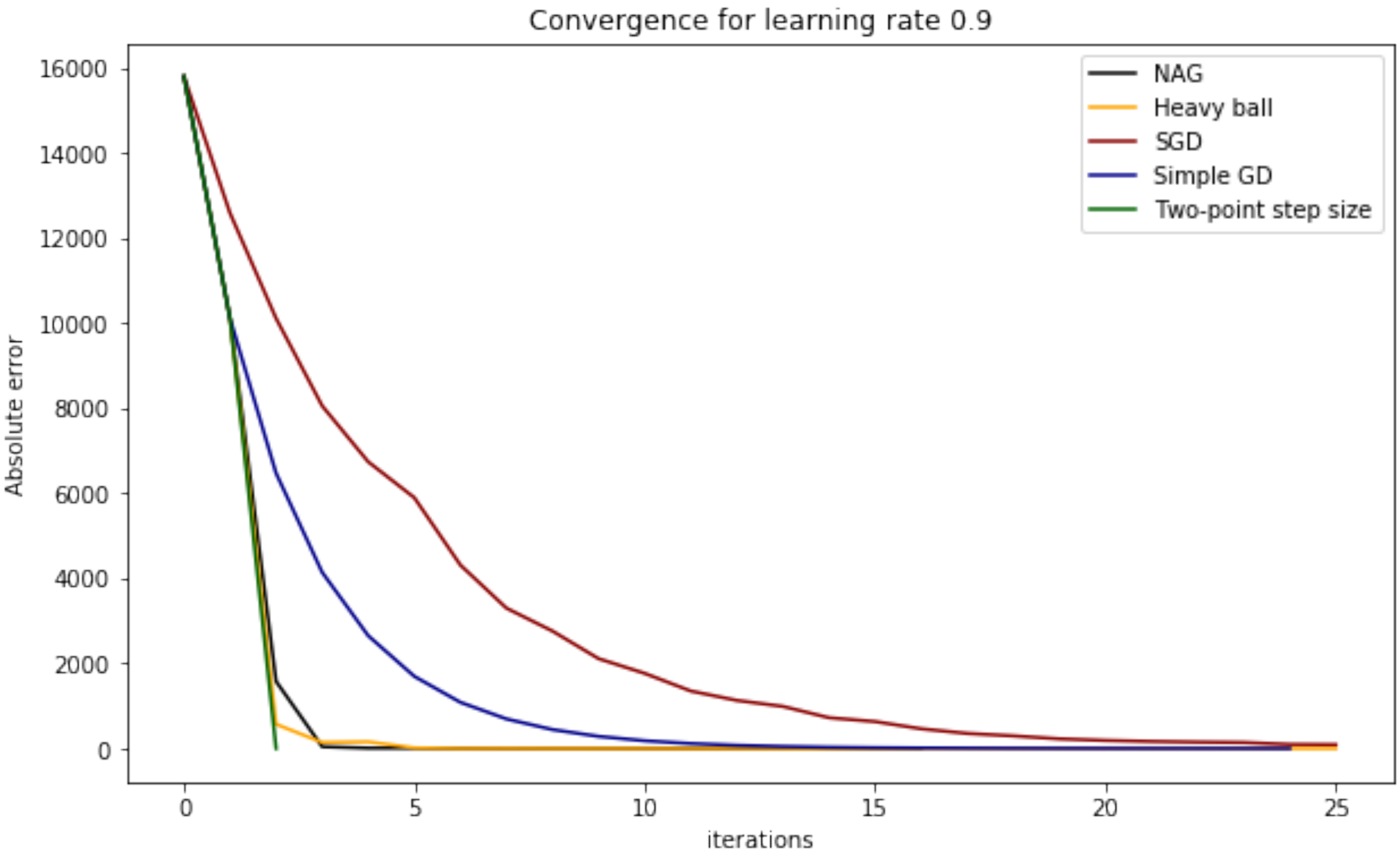
2. $MSE = \sum_{i=0}^{n}(y_i - \langle W * x_i \rangle)^2$

3. $\min_{W} \sum_{i=0}^{n}(y_i - \langle W * x_i \rangle)^2$

4. $W_k = W_{k-1} - \lambda \nabla MSE$

Repeat until $MSE \leq \varepsilon$**:**

Which algorithm is the fastest?



Conclusion

- Gradient descent is a simple, fast, strong algorithm
- GD cannot be used for global optimization
- There are various methods based on GD
- The fastest gradient-based algorithm is two-point step size method

Thank you for your attention!