Minimizing in 3D

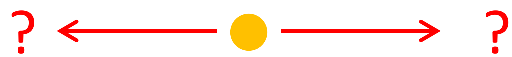
Method 3: Hooke-Jeeves

The Premise

The Hooke-Jeeves algorithm, also known as "pattern search", keeps track of the direction of travel as the process moves from point to point, instead of starting over from scratch at each new point.

The First Step

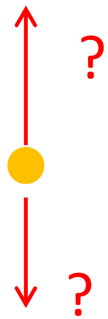
1. Test your initial point in one dimension only, for example by testing $(x_1 + .1, x_2)$ and $(x_1 - .1, x_2)$.



Save the best value of x_1.

The First Step

1. Test your initial point in one dimension only, for example by testing (x_1 + .1, x_2) and (x_1 - .1, x_2).

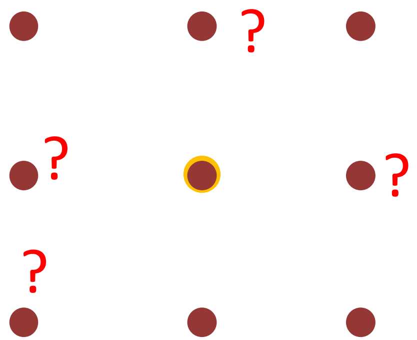


   Save the best value of x_1.

2. Repeat for $x_{2,}$ using the best value of x_1.

The First Step

1. Test your initial point in one dimension only, for example by testing $(x_1 + .1, x_2)$ and $(x_1 - .1, x_2)$.

   Save the best value of x_1.

2. Repeat for $x_{2,}$ using the best value of x_1.

   Now you have a new point made up of the new values of x_1 and x_2.

Practice Problem 1

Write a program that, given a function $f(x_1, x_2)$ and an initial point, tests the function in all four directions with a step of 0.1. Your program should print:

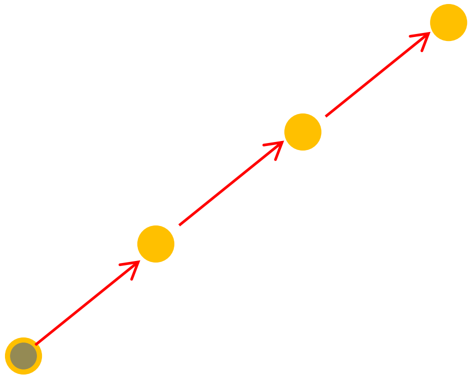
1) The original point
2) The improved point
3) The vector between them, found by subtracting (improved – original).

Test your code using $f(x_1, x_2) = (x_1 - 3)^2 + (x_2 + 1)^2$.

The Second Step

With the brute force method, we would just repeat the process from the new point.

Instead, with Hooke-Jeeves, we use the vector from the old point to the new point, and we search in that direction until we stop seeing improvement.



Practice Problem 2

Write a program that, given a function, a starting point, and a vector, moves the point in the direction of the vector until the function value stops decreasing.

To test your code,

Use $f(x_1, x_2) = (x_1 - 3)^2 + (x_2 + 1)^2$

1. Choose a starting point and run it through your code from problem 1 to find a vector.
2. Run the starting point and vector through your code for problem 2.
3. Repeat until the vector in step 1 returns (0, 0).

The Third Step

Once you stop seeing improvement in the direction of the original vector, you repeat the first and second steps: cast about to find a better point, then travel in that direction.

When the first step returns the original point as the best point (in other words, the vector is 0), then you move on to the next step.

Practice Problem 3

Using the function

$f(x_1, x_2) = (x_1 + x_2)^2 + (\sin(x_1 + 2))^2 + (x_2)^2 + 10$,

3a. Move between your first and second programs (from problems 1 and 2), with an interval of 0.1, until the vector returns 0. Transfer results between programs by hand at first.

3b. Then combine your two sets of code into one program that, given a function and starting point, will run the first two steps of the Hooke-Jeeves procedure and return the next starting point.

Repeat until the vector returns 0.

The Fourth Step

The fourth step is to decrease the incremental interval. If you started by adding and subtracting 0.1, now you might add and subtract 0.01.

Then you repeat everything, including reducing the interval, until you have reached the desired level of accuracy and there is no more improvement.

The Steps of Hooke-Jeeves

1. Cast about your original point to find the best point in its neighborhood.
2. Travel along that vector until you stop seeing improvement.
3. Repeat steps 1 and 2 until step 1 returns a vector of (0, 0).
4. Reduce the interval.
5. Repeat steps 1-4 until the interval is small enough to achieve the required tolerance.

The Steps of Hooke-Jeeves

Work in groups or alone to outline the basic loop structures* of a program that would run the entire Hooke-Jeeves method from start to finish (*plan* the program; do not *write* the program!):

*meaning: what kind of loop, where to start, when to stop, what gets done inside the loop

Practice Problem 4

Insert your code from Problem 3 into a loop that will run as long as the vector is nonzero.

Then, using $f(x_1, x_2) = (x_1 + x_2)^2 + (\sin(x_1 + 2))^2 + (x_2)^2 + 10$, run your new code with an interval of 1 to get a new point; from that starting point repeat with a new interval of .1; continue to repeat, reducing intervals, until you reach an interval of 0.0001.

Practice Problem 5

Insert your code from Problem 4 into a loop that will run Hooke-Jeeves with successive interval widths from 1 to 0.000001. This program, when completed, will execute the entire Hooke-Jeeves algorithm with one input line.

Test it on

$f(x_1, x_2) = (x_1 + x_2)^2 + (\sin(x_1 + 2))^2 + (x_2)^2 + 10.$

Document and save this program!