How to Program Numerical Differentiation Method: Idea

Formulation of the problem: reminder. We have:

- an algorithm $f(x_1, \ldots, x_n)$;
- measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$, and
- accuracies $\Delta_1, \ldots, \Delta_n$.

What we want to compute. We want to compute the values $\underline{y} = \widetilde{y} - \Delta$ and $\overline{y} = \widetilde{y} + \Delta$, where $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ and

$$\Delta = \sum_{i=1}^{n} |f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) - \widetilde{y}|.$$

How can we represent all this in a computer. In Java, f is a method. We want our program to be as general as possible, to be applicable to any n. For each n, a natural way to represent measurement results is by an array, similarly accuracies.

How we can program it.

```
public static void numerical(double[] tildexi, double[] deltai){
  double tildey = f(tildexi);
  double delta = 0.0;
  for(int i = 0; i < tildexi.length; i++)
    {tildexi[i] += deltai[i];
     delta += Math.abs(f(tildexi) - tildey);
     tildexi[i] -= deltai[i];}
  System.out.println("The lower endpoint is ", tildey - delta);
  System.out.println("The lower endpoint is ", tildey + delta);}
```