public double partialDeriv(double[] tildeX, int i, double h) {
    double tildeY = f(tildeX);
    double[] x_mod = new double[tildeX.length];
    int n = tildeX.length;
    for (int j = 0; j < n; j++) {
        x_mod[j] = tildeX[j];
        x_mod[j] += h;
        double y_mod = f(x_mod);
        return (y_mod - tildeY) / h;
    }
}

public double f(double[] x) {
    double sum = 0.0;
    for (int i = 0; i < x.length; i++) {
        sum += x[i];
    }
    return sum;
}

Mon Sept 29th.
Additional part:
R. Bafer, Kearfott.
10:30 - 11:20 pm
TBCR

Fri 12 pm - post-conf workshop
Templeton suite
Sat - Oct 5 until noon
NSF

Thurs Sept 25th - preview of the test
Tue Sept 30th - test.
Thurs Oct 2nd - work on your project
tue Oct 6th - Report.
Problem:
we have: \( f(x_1, x_2 \ldots x_n) \)
\( \hat{x}_1, \hat{x}_2 \ldots \hat{x}_n \)
\( \hat{u}_1, \hat{u}_2, \ldots \hat{u}_n \).
We want: \( \tilde{y} \), \( \sigma \rightarrow \) we want.
Algorithm:
\[
\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (f(x_i, \ldots x_{i-1}, x_i + \hat{u}_i, x_i + 1, \ldots x_n) - \tilde{y})^2}
\]
We did: small \( n \)
- we used numerical differentiation as main tool.
- we used simulations (which run much longer) to check on results.

Monte Carlo Simulations:
Fix some no of iterations \( N \)
for each iteration \( k = 1, \ldots N \)
* simulate meas. errors \( \Delta x_i^{(k)} = \Delta u_i \cdot \text{gauss}() \);  
* simulate the actual values \( x_i^{(k)} = \hat{x}_i - \Delta x_i^{(k)} \)
* simulate \( y \); \( y^{(k)} = f(x_1^{(k)}, \ldots x_n^{(k)}) \)
* simulate \( \Delta y^{(k)} = \tilde{y} - y^{(k)} \)
\[
\sigma = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (\Delta y^{(k)})^2}
\]
Accuracy \( \frac{1}{\sqrt{N}} \)  
- 3% \(, N = 1000 \)
- 10% \(, N = 100 \)

HW: To write a program to generate \( \sigma \) using Monte-Carlo method.
Main time: computing $f$.

Numerical diff: $(n+1)$ calls to $f$.

Monte Carlo: $(N+1)$ calls to $f$.

If we have more # of variables $> N \Rightarrow$ Monte Carlo

" " " " " " " " " " " " $< N \Rightarrow$ numerical diff

Geosciences Experiment:

200 km
600 m

300, $\theta = 2000$

7-8 shots

Optimal Curve:

Monte Carlo Asymptote.