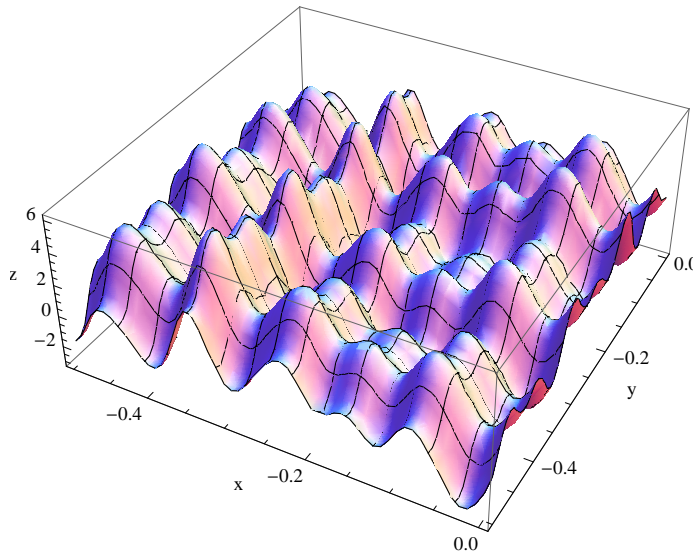


Numerical Optimization HW 1

1. Run a prebuilt numerical minimization routine in any programming environment (matlab, Mathematica, the GNU Scientific Library, MathCad, etc.) on the function defined and plotted below. FYI the global min is in the picture and is $-3.208139129534623580433366$. Do not worry if your code does not find it. Explain why this problem is hard.

```
f[x_, y_] := Exp[Sin[50 x]] + Sin[60 Exp[y]] +  
  Sin[70 Sin[x]] + Sin[Sin[80 y]] - Sin[10 (x + y)] + 1 / 4 (x^2 + y^2)  
Plot3D[f[x, y], {x, -0.5, 0}, {y, -0.5, 0}, AxesLabel -> {"x", "y", "z"}]
```



2. Repeat question one in a different programming environment.
3. Calculus texts (in the current MTU text it is Appendix A p1039) describe Newton's method and the Bisection method for numerically approximating solutions of equations $g(x) = 0$ in one dimension.
 - 3.1. Explain the connection between minimizing a function $g(x)$ and finding solutions of $g'(x) = 0$.
 - 3.2. Write down the Newton update formula for approximating the solution of $g'(x) = 0$
 - 3.3. Plot $g(x) = -(x-1)^2 (x-\frac{1}{2})^2 (x-\frac{1}{3})^2$ and use Newton's method with $x_0 = 0.8$ to find an approximate solution of $g'(x) = 0$. Repeat to find approximate $\{x, y\}$ coordinates of the local min between $1/3$ and $1/2$. Is the second problem harder or easier? Why? Show me your values.
 - 3.4. Repeat the above to using the Bisection method with reasonable initial values for the iteration.
4. Many Numerical Optimization algorithms look something like the following flow chart where \vec{p} is a direction and α is a scalar. Think about how to organize your computer program so that you can easily change various components. There is nothing to write down here. Just think about it.

Generic Numerical Optimization

