Numerical Optimization: Project List

1. Rich and Jason: Implement and test a variety of nonlinear conjugate gradient algorithms.
2. Particle Swarm Optimization:
   - **2.1.** Jack, Alex, and Jason: Implement and test a particle swarm algorithm. CUDA etc.
   - **2.2.** Robin, Wen, and Chad: Implement and test a particle swarm algorithm. Quasi-Newton interactions.
3. Quasi-Newton schemes:
   - **3.1.** Implement and test Quasi-Newton schemes (single vector and block vector, and limited memory - BFGS, DFP, SR1, Broyden Class): Convergence of B_k and H_k; recovery from "bad" values, possibilities for safe guarding.
   - **3.2.** Implement and test Quasi-Newton schemes (single vector and limited memory - BFGS, DFP, SR1, Broyden Class): Convergence of B_k and H_k; recovery from "bad" values, possibilities for safe guarding.
   - **3.3.** Implement and test a few Quasi-Newton schemes for contrained optimization.
4. Solving Nonlinear Equations
   - **4.1.** Finding transfer orbits
   - **4.2.** ??
5. Parallel Stuff
   - **5.1.** Implement a web search algorithm that performs a "p-direction" line search simultaneoulsy:

     Compute $f_{i,j} = f(\alpha^j p_i)$ for $-m \le j \le m$

     It should slect the smallest and return suitable update values.

     Attempt to prove a thm that showing that the Z lemma holds provided the lowest value $f_{i,j}$ satisfies Wolfe 1 and $j < m$. I think the proof should look like Backtracking

     Test various implementations of the p-directions. For instance: you can zero out any entry you like, you can positively scale any individual entry. You can interchange any two components and flip one of the signs, ...

     Contemplate what this would mean for constrained optimization.
6. Implement a p-directional positivity scheme. Roughly speaking the Cholesky decomposition trick for +def is to find a minimal change to the Hessian which preserves +def. i.e. something like Min $\| \nabla^2 f - \hat{A} \|$ and with $\hat{A}$ SPD and then solve $\hat{A}.p_n = -\nabla f$ to get our modifed Newton like direction. To do this we need to generate the complete Hessian and then throw some of it away. Maybe instead it might be a good idea to generate some bits of the Hessian and use them all. Maybe something like: min $\| \nabla^2 f.p + \nabla f \|$ subject to the descent constraint $p.\nabla f < 0$. The bad news about this is that in most cases in high dimensions minimally constrained problems have tight constraints so one might need to modify the constraint. We could also do the minimization over a restricted set of directions. Maybe something like $p = -\alpha_0 \nabla f + \alpha_1 r_1 + \alpha_2 r_2 + \ldots$ where we construct the rs so that $r_i.\nabla f = 0$. test the behavior of various possibilities. Contemplate what this would mean for constrained optimzation. As an additional feature of this sort of problem we would not need all the components of $\nabla^2 f$ for such a computation.
7. AMBER Models:
   - **7.1.** Build interesting AMBER models of a range of sizes in matlab and test the behavior of the various optimization algorithms. Ours and built-in
   - **7.2.** Build interesting AMBER models in mathematica of a range of sizes and test the behavior of the various optimization algorithms. Ours and built-in