Convex and Nonsmooth Optimization
HW6: Nesterov's Optimal Gradient Method

Michael Overton

Spring 2020

Write a MATLAB code `nesterov` that implements Nesterov's "optimal gradient method" (or "accelerated gradient method") given on p. LCB5 of these notes, which are taken from p. 66-68 and p.81 of Nesterov's book. (In his book, Nesterov uses μ and L for m and M. The "optimal method" is taken from (2.2.11), p. 81 of his book.)

Also implement the ordinary gradient method with two *fixed* step sizes, namely,

- $t = \frac{1}{M}$ (for which we have the bound on p. 4 of last week's notes; see also p. GD1 of my summary notes on gradient descent and (9.18) in BV)
- $t = \frac{2}{m+M}$ (for which we have the bound on p. GD2 of the summary notes, which is taken from Theorem 2.1.15 in Nesterov's book).

Run all three methods on four examples:

- the quadratic defined by the Hilbert matrix that you used in HW5 using the same starting point (the vector of all ones)
- the example of Exercise 9.30 (BV page 519) that you used in HW5, using the same starting point (zero)
- Nesterov's worst case example given on p. LCB1 of the complexity notes (and p. 67 of Nesterov's book) with the given starting point (zero) and with the order of the tridiagonal matrix truncated to $n = 10,000$, with $m = 1$ and $M = 100$. Be sure to code the tridiagonal matrix as a *sparse* matrix (type `help sparse` in MATLAB) so that matrix-vector multiplies with the tridiagonal matrix are efficient. If you don't do this properly, your code will be very slow and you may run out of memory.
- The same example with $M = 10,000$.

Summarize your results using `semilogy` as you did in HW5, carefully comparing the results you observe with the theoretical guarantees.