Algorithm zero

Intuitive original algorithm from theory:

⇒ Initial
$$X^{(0)}$$
, $P_0 = -\nabla \phi(x^{(0)})$ (where $P_0 = 0$)

⇒ D_0 Initial stepping in P_0 directions

$$CL_0 = \frac{-\nabla \phi(X^{(0)})^T P_0}{P_0^T A P_0} \quad CLn)$$

Step in initial direction
$$X^{(1)} = X^{(0)} + d_0 P_0$$

⇒ For $t = 0, ..., n-1$:

Pick a direction cut how much to step

$$P_{t-1} A P_{t-1}$$

Figure out where to step to lands next (new direction)
$$P_t = -\nabla \phi(x^{(t)}) + P_t P_{t-1}$$

when do it again with CL_t

$$CL_t = \frac{-\nabla \phi(x^{(t)})^T P_t}{P_t^T A P_t}$$

$$X^{(t+1)} = X^{(t)} + CL_t P_t$$

Step in new direction

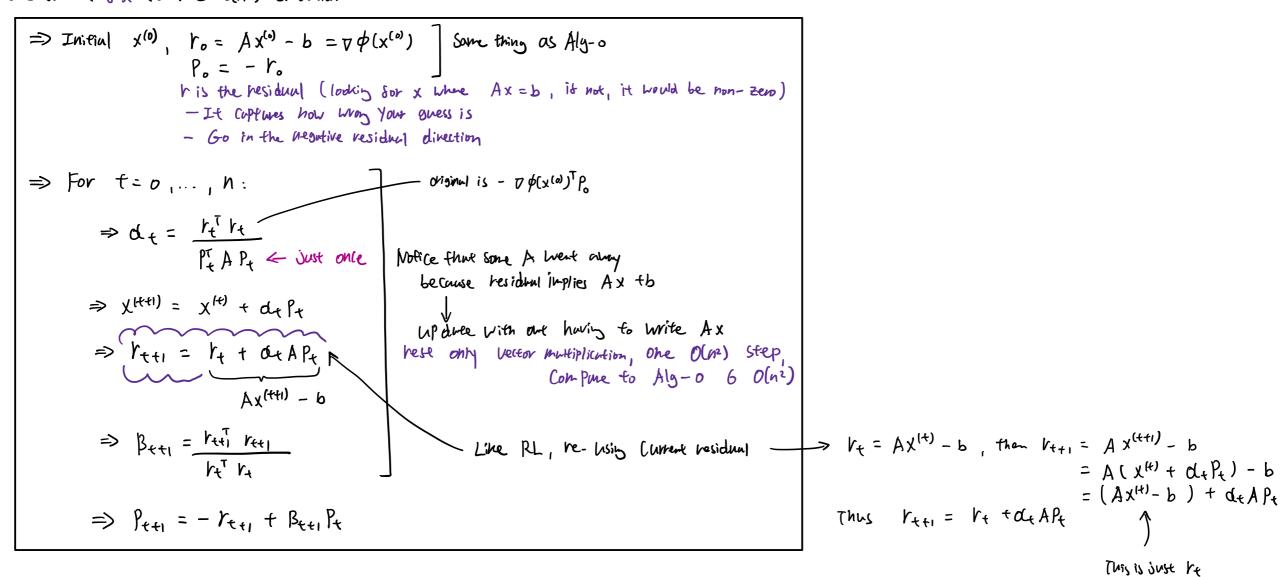
Pick new direction based on gradient or previous X^t and previous P_t

Theorem: Algorithm o Commences in at most n Step, Very Panewsul This is actually not that much more expensive than pheniously

Multiply A is the most expensive operation $O(n^2)$

Algorithm one

More efficient fix to the dn2) situation:



There can be extensions to general convex surflion, though hard to sind conjugate

Computational Complexity

Grammer of not personning worst than GD on Complexity's Penspective

GD on Convex requires Ax as well, so hothing here is more expensive $O(n^2)$ f 7 O(n)