	Gradient descent	**Subgrad method**	**Prox grad descent**	**Stochastic grad descent**
Criterion	smooth f	any f	smooth + simple, $f = g + h$	smooth + simple, $f = g + h$
Constraints	projection onto constraint set	projection onto constraint set	constrained prox operator	projection onto constraint set
Opti parameters	fixed step size ($t \leq 1/L$) or line search	diminishing step sizes	fixed step size ($t \leq 1/L$) or line search	fixed or diminishing step sizes, mini-batch size
Iteration cost	cheap (compute gradient)	cheap (compute subgradient)	moderately cheap (evaluate prox)	very cheap (compute stochastic gradient)
Rate	$O(1/\epsilon)$ (acceleration: $O(1/\sqrt{\epsilon})$, strong convexity: $O(\log(1/\epsilon))$)	$O(1/\epsilon^2)$	$O(1/\epsilon)$ (acceleration: $O(1/\sqrt{\epsilon})$, strong convexity: $O(\log(1/\epsilon))$)	$O(1/\epsilon^2)$, but practically converges rapidly at the start

	Newton	**Barrier method**	**Primal-dual interior-point**	**Quasi-Newton**
Criterion	twice smooth f	twice smooth f	twice smooth f	twice smooth f
Constraints	equality constraints	equality, twice smooth h_i (inequality constraints)	equality, twice smooth h_i (inequality constraints)	unconstrained
Opti parameters	pure step size ($t = 1$) or line search	inner: pure step size or line search; outer: barrier parameter	line search for step size, barrier parameter	line search
Iteration cost	moderate to expensive (compute Hessian and solve linear system)	expensive to very expensive (one iter solves one smoothed problem)	moderate to expensive (one iter performs one Newton step)	moderately cheap (compute gradients, inner products; no matrix inversion)
Rate	$O(\log\log(1/\epsilon))$ (local rate)	$O(\log(1/\epsilon))$ (also rate for total Newton steps)	$O(\log(1/\epsilon))$	local superlinear rate

	Prox Newton	**Coordinate descent**	**ADMM**	**Frank-Wolfe**
Criterion	twice smooth + simple, $f = g + h$	smooth + separable, $f = g + h$	block separable, $f(x, z) = g(x) + h(z)$	smooth f
Constraints	constrained H-prox	separable constraints	always have equality constraints; for inequalities: constrained prox	any compact constraint set for which we know linear minimization oracle
Opti parameters	pure step size or line search	none	augmented Lagrangian parameter	default step sizes or line search
Iteration cost	expensive to very expensive (evaluate H-prox)	cheap to expensive (one iteration performs a full cycle or coordinate minimizations)	cheap to expensive (one iteration solves g, h subproblems, makes a dual step)	moderately cheap (one iteration evalutes linear minimization oracle)
Rate	$O(\log\log(1/\epsilon))$ (local rate)	same as prox grad, but can be faster in practice	same as prox grad, similar in practice	same as prox grad, but can be slower in practice