Optimization toolbox of Matlab™

I. Optimisation algorithms

For more details, use the command `help` (e.g. `help fminbnd`) or `doc` (e.g. `doc fminbnd`)

	Mathematical formulation	Matlab solver
One variable unconstrained optimization problem	$\min_{x\in\mathbb{R}} f(x)$	`fminbnd` (Golden search or parabolic interpolation)
One variable unconstrained optimization problem	$\min_{x\in\mathbb{R}^n} J(x)$	`Fminunc` (quasi-Newton) `Fminsearch` (Nelder and Mead)
Constrained optimization	$\min_{x\in\mathbb{R}^n} J(x)$ s.t. $\begin{cases} c(x)\le 0, c_{eq}(x)=0 \\ Ax\le b, A_{eq}x=b_{eq} \\ l\le x\le u \end{cases}$	`Fmincon` Several algorithms (SQP, interior point,…)
Multi-objective optimization problems	$\min_{x,\gamma} \gamma$ $s.t. \begin{cases} \lvert J(x)-goal\rvert \le w\gamma \\ c(x)\le 0, c_{eq}(x)=0 \\ Ax\le b, A_{eq}x=b_{eq} \\ l\le x\le u \end{cases}$	`fgoalattain`
Linear Programming	$\min_x c^T x$ $s.t. \begin{cases} Ax\le b, A_{eq}x=b_{eq} \\ l\le x\le u \end{cases}$	`Linprog` Simplex or interior points
Integer Linear Programming	$\min_x c^T x$ t.q. $\begin{cases} Ax\le b, A_{eq}x=b_{eq} \\ x_i \text{ integer} \end{cases}$	`intlinprog`
Quadratic Programmin	$\min_x \frac{1}{2}x^T Hx + f^T x$ $s.t. \begin{cases} Ax\le b, A_{eq}x=b_{eq} \\ l\le x\le u \end{cases}$	`quadprog`
Linear Least Square problem	$\min_x \lVert Cx-d\rVert^2$ $s.t. \begin{cases} Ax\le b, A_{eq}x=b_{eq} \\ l\le x\le u \end{cases}$	`lsqlin`
NonLinear Least Square problem	$\min_x \lVert F(x)\rVert^2$ t.q. $l\le x\le u$	`lsqnonlin`

II. Main stopping conditions and optimization options (via `optimset` or `optimoptions`)

Important remark: Depending on the used version of Matlab, use either `optimset` (previous versions) or `optimoptions` (last versions). See matlab document to choose the adequate command. `optimset` still works in last version of Matlab, but for some solvers, `optimoptions` must be used to change options.
Try to use `optimoptions`.
Use `help optimoptions` for more details on the parameters.

Parameter	Description and value
`Display`	Level of display. Possible values: 'notify' (default) | 'final' | 'off' | 'none' | 'iter'
`MaxIter`	Maximum number of iterations
`TolFun, TolX, TolCon/ ConstraintTolerance` (depending on the Matlab version)	Termination tolerance on (respectively): the function value, the decision variable, constraints.
`GradObj`	The gradient of the objective function is provided by the user (if set to `on`).
`GradConstr`	The gradient of the constraints is provided by the user (if set to `on`).
`LargeScale`	To set to `on` for large scale problems
`Algorithm`	To choose the algorithm to be used (depending on the used optimization algorithm)
`HessianApproximation`	Method of Hessian approximation: `'bfgs','lbfgs', {'lbfgs',PositiveInteger}`, or `'finite-difference'`.
`OptimalityTolerance`	Termination tolerance on the first-order optimality.

III. Some examples to call solvers

Hereafter, options is set using optimoptions (or optimset).
If a parameter is equal to the default value, it can be chosen as [].

- [x, fval, exitflag, output] =
  `fminsearch`(fun, x0, options, user parameters)

- [x, fval, exitflag, output] =
  `fminunc`(fun, x0, options, user parameters)

- [x, fval, exitflag, output] =
  `fmincon`(fun, x0, A, b, Aeq, beq, lb, ub, nl_con, options, user parameters)

- [x, fval, attaingoal, exitflag, output] =
  `fgoalattain`(fun, x0, goal, w, A, b, Aeq, beq, lb, ub, nl_con, options, user parameters)

- [x, fval, exitflag, output] =
  `linprog`(f, A, b, Aeq, beq, lb, ub, x0, options, user parameters)

- [x, fval, exitflag, output] =
  `quadprog`(f, A, b, Aeq, beq, lb, ub, x0, options, user parameters)

- [x, resnorm, residual, exitflag, output] =
  `lsqlin`(C, d, A, b, Aeq, beq, lb, ub, x0, options, user parameters)

- [x, resnorm, residual, exitflag, output] =
  `lsqnonlin`(fun, x0, lb, ub, options, user parameters)