

Derivative-free Optimization (DFO)

Optimizing complex numerical models is one of the most common problems found in the industry (finance, multi-physics simulations, engineering, etc.). To solve these optimization problems with a standard optimization algorithm such as Gauss–Newton (for problems with a nonlinear least squares structure) or CG (for unstructured nonlinear objective) requires good estimates of the model’s derivatives. They can be computed by:

- explicitly written derivatives
- algorithmic differentiation (see NAG AD tools)
- finite differences (bumping), $\frac{\partial \phi}{\partial x_i} \approx \frac{\phi(x+he_i) - \phi(x)}{h}$

If exact derivatives are easy to compute then using derivative-based methods is preferable. However, explicitly writing the derivatives or applying AD methods might be impossible if the model is a Black Box. The alternative, estimating derivatives via finite differences, can quickly become impractical or too computationally expensive as it presents several issues:

- **Expensive**, one gradient evaluation requires at least $n + 1$ model evaluations;
- **Inaccurate**, the size of the model perturbations h influences greatly the quality of the derivative estimations and is not easy to choose;
- **Sensitive to noise**, if the model is subject to some randomness (e.g. Monte Carlo simulations) or is computed to low accuracy to save computing time, then finite difference estimations will be highly inaccurate;
- **Poor utilization of model evaluations**, each evaluation is only used for one element of one gradient and the information is discarded as soon as that gradient is no longer useful to the solver.

These issues can greatly slow down the convergence of the optimization solver or even completely prevent it. Conversely, DFO solvers are designed to get good improvements of the objective in these situations. They are able to reach convergence with a lot fewer function evaluations and are naturally quite robust to noise in the model evaluations.

NAG introduces, at Mark 27, a complete update of its model-based DFO solvers for nonlinear least squares problems (e04ff, e04fg) and unstructured nonlinear problems (e04jd, e04je). They present a number of attractive features:

- Improved performance;
- Integrated to the *NAG Optimization Modelling Suite* with simple interfaces to the solvers and related routines;
- Optional reverse communication interface (e04fg, e04je);
- Ability to start making progress with as few as 2 objective evaluations;
- Improved resilience to noise.

Problems structure

One frequent problem in practice is tuning a model’s parameters to fit real-world observations as well as possible. Let us consider a process that is observed at times t_i and measured with results y_i , for $i = 1, 2, \dots, m$. Furthermore, the process is assumed to behave according to a numerical model $\phi(t, x)$ where x are parameters of the model. Given the fact that the measurements might be inaccurate and the process might not exactly follow the model, it is beneficial to find model parameters x so that the error of the fit of the model to the measurements is minimized. This can be formulated as an optimization problem in which x is the decision variables and the objective function is the sum of squared errors of the fit at each measurement, thus:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \sum_{i=1}^m (\phi(t_i, x) - y_i)^2 \\ \text{s.t. } l \leq x \leq u \end{aligned} \quad (1)$$

When the optimization problem cannot be written as nonlinear least squares, a more general formulation has to be used:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } l \leq x \leq u \end{aligned} \quad (2)$$

The NAG solvers accommodate both of these formulations.

Noise robustness: illustration

Consider the following unbounded problem where ϵ is some random uniform noise in the interval $[-\nu, \nu]$ and r_i are the residuals of the Rosenbrock test function.

$$\min_{x \in \mathbb{R}^n} f(x) = \sum_{i=1}^m (r_i(x) + \epsilon)^2 \quad (3)$$

Let us solve this problem with a Gauss–Newton method combined with finite differences (e04fc) and the corresponding derivative-free solver (e04ff). For various noise level ν , we present in the following table the number of model evaluations needed to find a solution with an ‘achievable’ precision with respect to the noise level:

$$f(x) < \max(10^{-8}, 10 \times \nu^2)$$

The figures show the average on 50 runs, the number in brackets represent the number of failed runs.

level of noise ν	0.0e00	1.0e−10	1.0e−08	1.0e−06	1.0e−04	1.0e−02	1.0e−01
e04fc	90 (0)	93 (0)	240 (23)	∞ (50)	∞ (50)	∞ (50)	∞ (50)
e04ff	22 (0)	22 (0)	22 (0)	22 (0)	22 (0)	17 (0)	15 (0)

Table 1: the number of objective evaluations required to reach the desired accuracy

In this example, the new derivative-free solver is both cheaper in term of model evaluations and far more robust with respect to noise.

References

- [1] C. Cartis, J. Fiala, B. Marteau, and L. Roberts (2019) *Improving the Flexibility and robustness of model-based derivative-free optimization solvers* ACM Transactions On Numerical Software.
- [2] C. Cartis and L. Roberts (2017) *A derivative-free Gauss-Newton method* Mathematical Programming Computation.
- [3] Powell M. J. D. (2009) *The BOBYQA algorithm for bound constrained optimization without derivatives* Report DAMTP 2009/NA06 University of Cambridge.