An active-set trust-region method for derivative-free bound-constrained optimization.

ANKE TRÖLTZSCH [CERFACS]

JOINT WORK WITH:

SERGE GRATTON [INPT/IRIT]
PHILIPPE L. TOINT [FUNDP, NAMUR, BELGIUM]

Journee des doctorants

ENSEEIHT, 7 September 2010

Outline

Outline

Why using derivative-free optimization?

Some reasons to apply Derivative-Free Optimization (DFO):

- Derivatives are not available
  - Function evaluations are costly and/or noisy - Accurate approximation of derivatives by finite differences is prohibitive
  - Source code not available or owned by a company - Automatic differentiation impossible to apply
- Growing sophistication of computer hardware and mathematical algorithms and software (opens new possibilities for optimization)

Applications:

- Tuning of algorithmic parameters,
- Medical image registration,
- Engineering design optimization, ...

Problem formulation

We consider the bound-constrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \qquad \text{s.t. } xl(i) \leq x(i) \leq xu(i), i = 1, ..., n$$

where the first derivatives of the objective function are assumed to exist and be Lipschitz continuous, although explicit evaluation of these derivatives is assumed to be impossible.

We consider a model-based trust-region algorithm for computing local solutions of the minimization problem.

The method iteratively uses a local interpolation model of the objective function $f(x)$ to define a descent step, and adaptively adjusts the region in which this model is trusted.

Bibliography on developments in model-based DFO

Numerical optimization using local models:

- Powell, ”A direct search optimization method that models the objective function by quadratic interpolation”, 1994
- Conn, Scheinberg, and Toint, ”On the convergence of derivative-free methods for unconstrained optimization”, 1996
- Powell, ”The NEWUOA software for unconstrained optimization without derivatives”, 2004
- Conn, Scheinberg, and Vicente, ”Introduction to Derivative-free Optimization”, 2008
- Fasano, Nocedal, and Morales, ”On the geometry phase in model-based algorithms for derivative-free optimization”, 2009
- Scheinberg and Toint, ”Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization”, 2009

Outline

Polynomial interpolation

Consider $\mathcal{P}_n^d$, the space of polynomials of degree $\leq d$ in $\mathbb{R}^n$.

A polynomial basis $\phi = \{\phi_1(x), \phi_2(x), ..., \phi_p(x)\}$ of $\mathcal{P}_n^d$ is a set of p polynomials of degree $\leq d$ that span $\mathcal{P}_n^d$.

For any basis ϕ, any polynomial $m(x) \in \mathcal{P}_n^d$ can be written as

$$m(x) = \sum_{j=1}^{p} \alpha_j \phi_j(x),$$

where α_j are real coefficients.

Polynomial interpolation

Given a sample set $Y = \{y^1, y^2, ..., y^p\} \subset \mathbb{R}^n$ and a polynomial $m(x)$ of degree d in $\mathbb{R}^n$ that interpolates $f(x)$ at the points Y, the coefficients $\alpha_1, ..., \alpha_p$ can be determined by solving the linear system

$$M(\phi, Y)\alpha_\phi = f(Y),$$

where

$$M(\phi, Y) = \begin{bmatrix} \phi_1(y^1) & \phi_2(y^1) & \cdots & \phi_p(y^1) \\ \phi_1(y^2) & \phi_2(y^2) & \cdots & \phi_p(y^2) \\ \vdots & \vdots & & \vdots \\ \phi_1(y^p) & \phi_2(y^p) & \cdots & \phi_p(y^p) \end{bmatrix}, \quad f(Y) = \begin{bmatrix} f(y^1) \\ f(y^2) \\ \vdots \\ f(y^p) \end{bmatrix}.$$

Poisedness

If the coefficient matrix $M(\phi, Y)$ of the system is nonsingular, the set of points Y is called poised for polynomial interpolation in $\mathbb{R}^n$, otherwise the set Y is called non-poised.

As poisedness alone doesn't define the distance from singularity, there exists a measure of well-poisedness.

The most commonly used measure of well-poisedness in the polynomial interpolation literature is based on Lagrange polynomials [Powell, 1994].

Lagrange polynomials

If the sample set Y is poised, the basis of Lagrange polynomials exists and is uniquely defined (and vice versa).

The unique polynomial $m(x)$ that interpolates $f(x)$ on Y using the basis of Lagrange polynomials for Y can be expressed as

$$m(x) = \sum_{i=1}^{p} f(y^i)\ell_i(x),$$

where

$$\ell_j(y^i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

is the basis of Lagrange polynomials.

Lagrange polynomials - Illustration

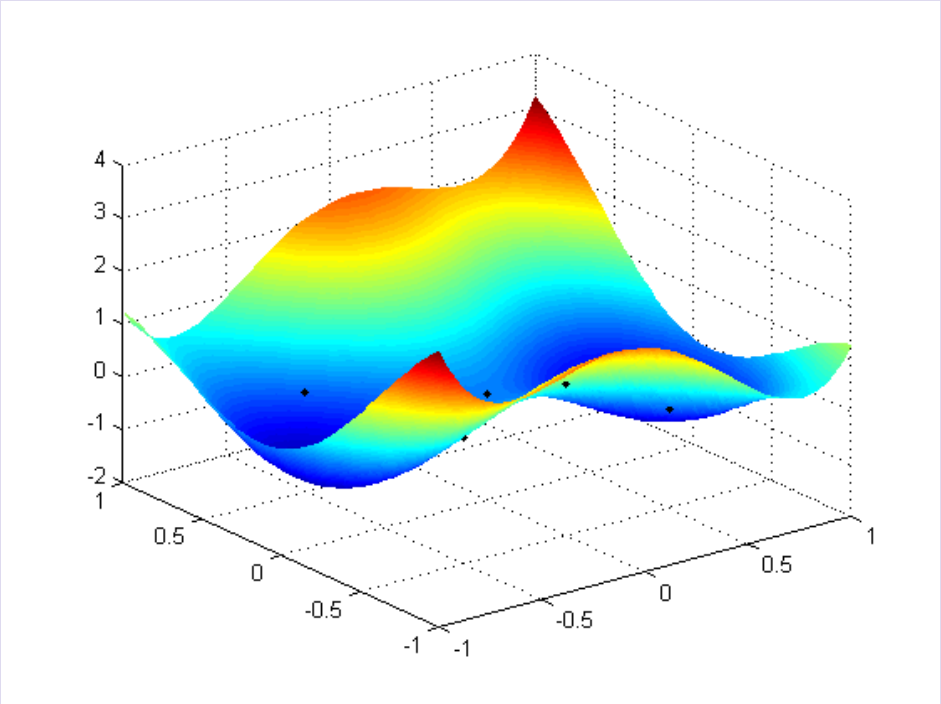


Figure: Six-hump camel back function with sample points

Lagrange polynomials - Illustration

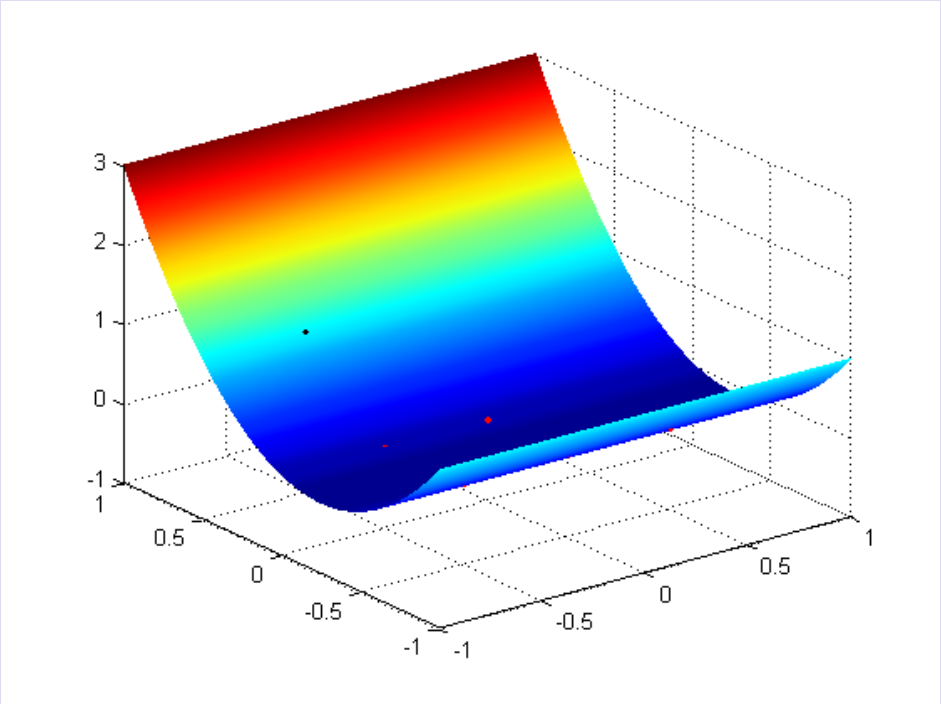


Figure: Lagrange polynomial associated with the 3. point

Lagrange polynomials - Illustration

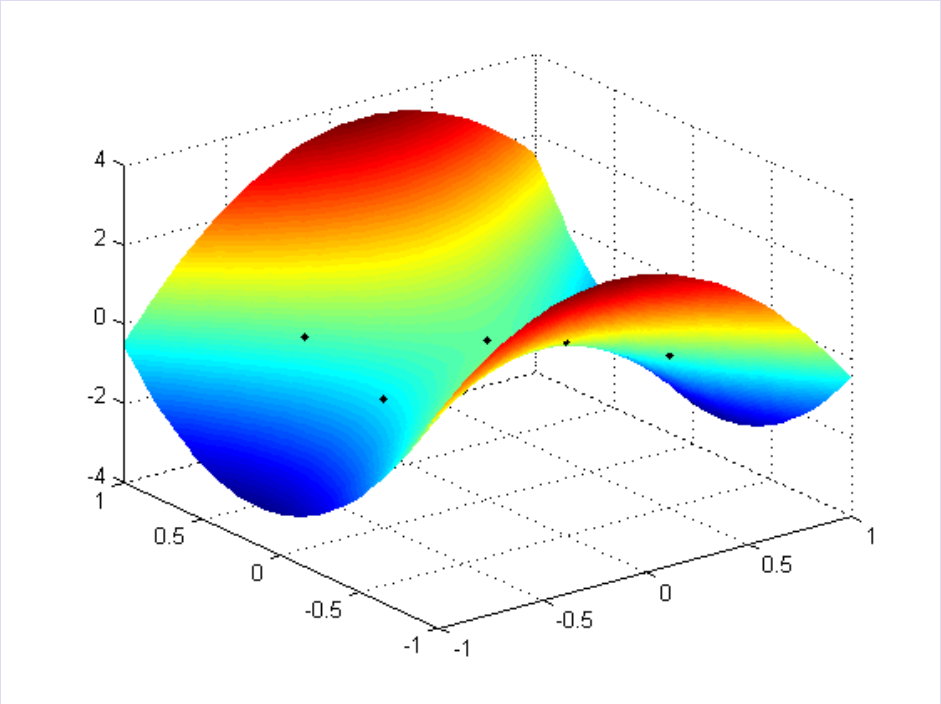


Figure: The resulting interpolation polynomial

Well poisedness

Very useful feature of Lagrange polynomials:

The upper bound on their absolute value in a region $\mathcal{B}$ is a classical measure of well-poisedness of the interpolation set Y in the ball $\mathcal{B}$.

A poised set Y is said to be Λ-poised in $\mathcal{B}$ if one has that

$$\max_{1 \le i \le p} \max_{x \in \mathcal{B}} |\ell_i(x)| \le \Lambda.$$

The smaller Λ, the better the quality of the geometry of the interpolation set.

Error bounds on model value and model gradient value

Given a ball $\mathcal{B}(x, \Delta)$, a poised interpolation set $Y \in \mathcal{B}(x, \Delta)$ and its associated basis of Lagrange polynomials $\ell_i(x), i = 0, ..., p$, there exists constants $\kappa_{ef} > 0$ and $\kappa_{eg} > 0$ such that, for any interpolation polynomial $m(x)$ of degree one or higher and any given point $y \in \mathcal{B}(x, \Delta)$,

$$||f(x) - m(x)|| \leq \kappa_{ef} \sum_{i=1}^{p} ||y_i - x||^2 |\ell_i(x)|$$

and

$$||\nabla_x f(x) - \nabla_x m(x)|| \leq \kappa_{eg} \Lambda \Delta,$$

where $\Lambda = max_{i=1,...,p} max_{x \in \mathcal{B}(x,\Delta)} |\ell_i(x)|$.

[Ciarlet and Raviart, 1972]

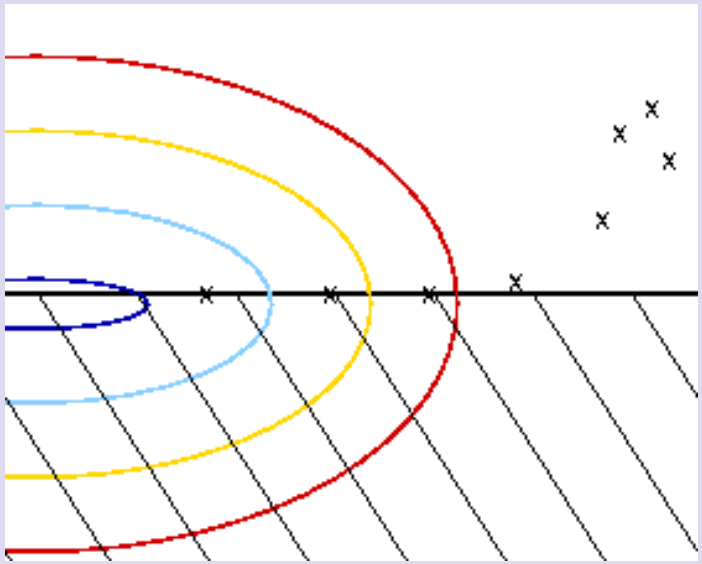
Outline

Geometry control

- Fasano, Nocedal, and Morales [2009] observed that an algorithm which simply ignores any geometry considerations may in fact perform quite well in practice.
- But it may lose the property of provable global convergence to first-order critical points [Scheinberg and Toint, 2009].
- Shows that we cannot afford to do without maintaining quality of the geometry of the interpolation set.
- Improvement is usually carried out at special "geometry improving" steps $\longrightarrow$ expensive!
- We know: failure of current iteration might be due to a too large trust region or a bad quality of the interpolation model.
- Suitable choice of interpolation points yields a self-correcting geometry

Outline

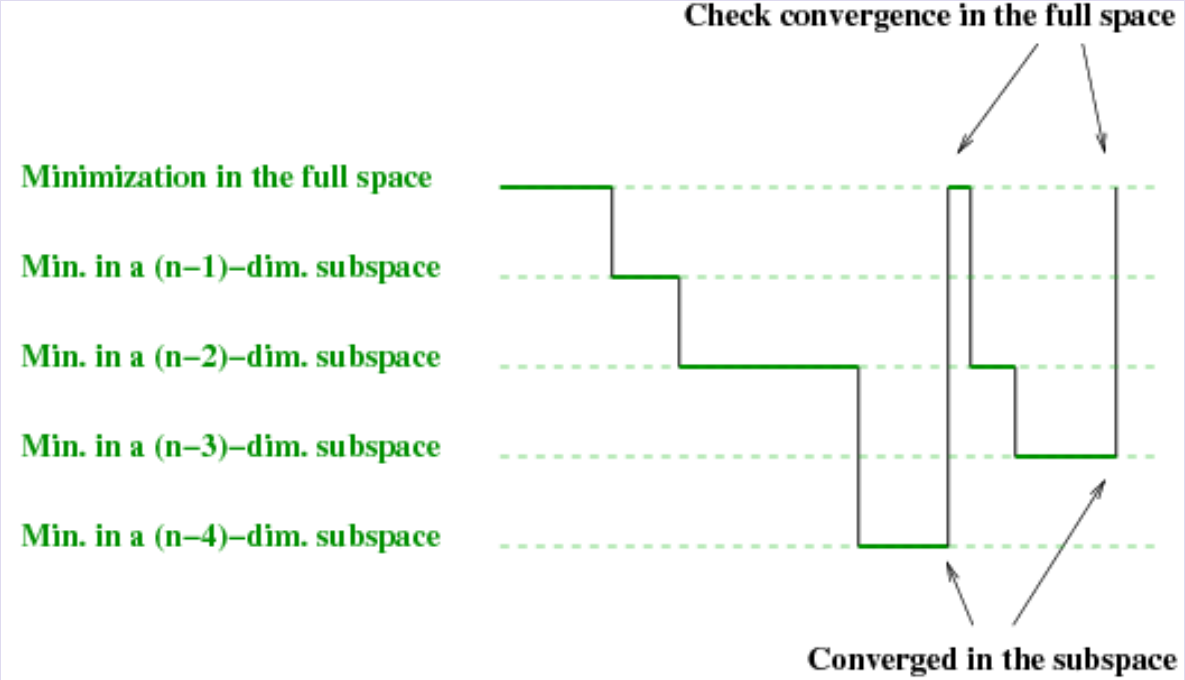
Extension to bounds



- Situation: algorithm converges towards a minimum
- Problem: iterates get aligned along the bound
- Results in a degenerate set of points due to the bounds!
- Λ-poisedness no suitable measure anymore, because maximum of Lagrange polynomials lies outside of the bounds
- Thus, self correcting property not working

Solution: an active-set method

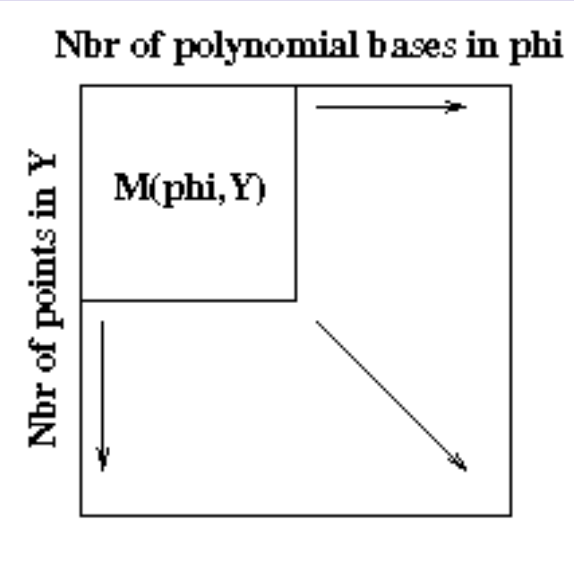
Continue minimization in a smaller dimensional subspace



- If encounter an active bound, reduce dimension of the problem
- If converged in the subspace, going back to check convergence in the full-space

Features of the algorithm

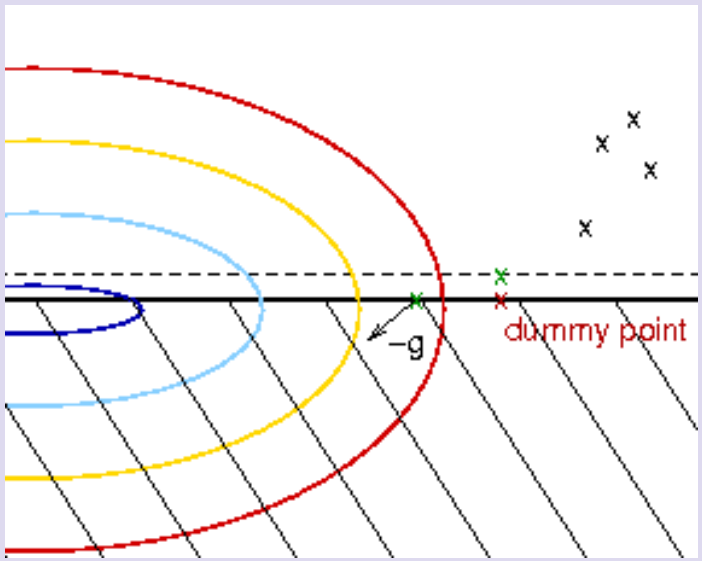
Variable size models



- Applying determined interpolation, $p = q$, number of points equals number of elements of the polynomial basis
- Degree of initial interpolation model user-defined: linear, diagonal, quadratic (Default: linear)
- Fill up the diagonal of the Hessian first and then its subdiagonals consecutively
- Geometry is monitored using the condition number of the shifted and scaled system matrix $M(\bar{\phi}, \hat{\mathcal{Y}})$

Features of the algorithm

Attempt to save function evaluations by creating dummy points



- New active bound: need to build a model in the subspace
- Consider points lying close to the active bound(s), create dummy points by projecting onto the subspace
- Compute the model values at the dummy points
- Consider real points and dummy points lying in the subspace to build the model
- Dummy points are then replaced by the new iterates

Features of the algorithm

Stopping criterion using the model gradient

- We have that

$$\|P_F(x_k - \nabla f_k(x_k)) - x_k\|_\infty \leq \|P_F(x_k - \nabla m_k(x_k)) - x_k\|_\infty + \|\nabla m_k(x_k) - \nabla f(x_k)\|_2$$

- Check convergence to a first-order critical point by verifying

$$\|P_F(x_k - \nabla m_k(x_k)) - x_k\|_\infty \leq \epsilon$$

and checking the bound on the gradient error

$$\|\nabla m_k(x_k) - \nabla f(x_k)\|_2 \leq \kappa_{eg} \Lambda \Delta \leq \epsilon$$

for some user-defined constant κ_{eg}

Outline

Methodology

CUTEr testing environment

- 53 bound-constrained test cases from CUTEr test environment
- Nbr. of variables varies from 1 to 30 dimensions

Competitor: BOBYQA

- Developed by M.J.D. Powell [2009]
- State-of-the-art code for bound-constrained minimization without derivatives

Stopping criteria are different

- Using optimal objective function value computed by TRON (using first and second derivatives) as a reference
- We terminate when a defined number of correct significant figures in f were attained

Numerical results

Performance profile in terms of nbr. of function eval.

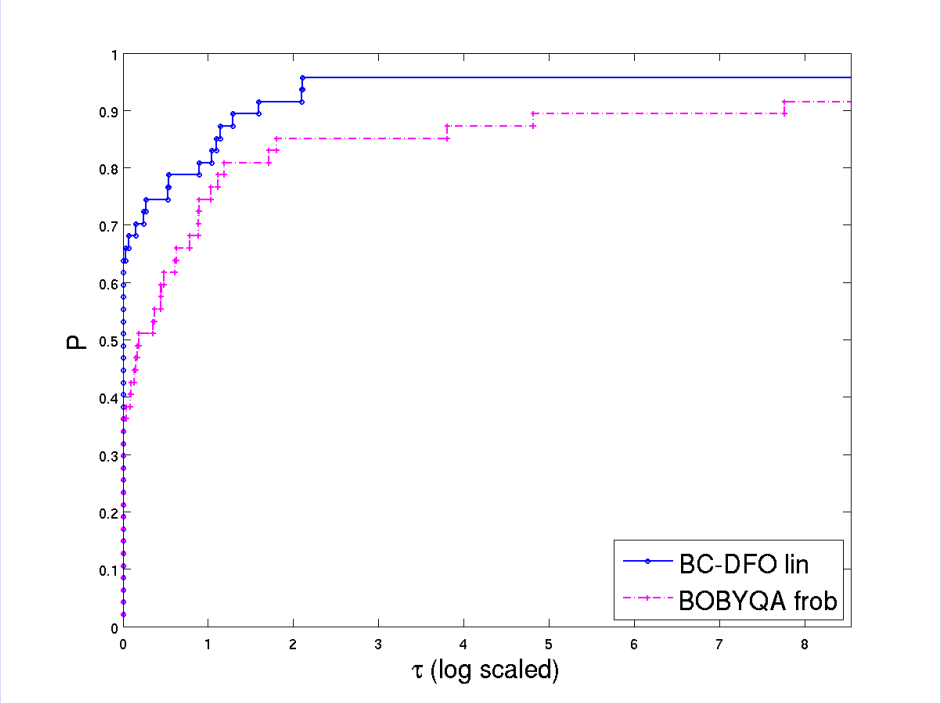


Figure: require 6 correct significant figures attained in f

Outline

Conclusions and Perspectives

Summary

- Presented a new model-based trust-region DFO algorithm with a self-correcting geometry property
- Extended the algorithm to handle bounds
- Implemented a robust version of the algorithm: BC-DFO
- Compared BC-DFO to BOBYQA with quite satisfying results

Perspectives

- Consider further enhancements on model Hessian update to improve performance
- Test the algorithm on real-life application (aerodynamic functions provided by Airbus)
- Implement the use of an inexact gradient

Thank you for your attention!

Condition number as a measure of well poisedness

In general, the condition number of the matrix $M(\phi, Y)$ of the interpolation conditions is a bad measure of poisedness of Y since it can be made arbitrarily large by changing the basis ϕ and scaling the interpolation set Y.

Considering the basis of monomials $\bar{\phi}$ and $\hat{Y}$, a shifted and scaled version of Y, so that $\hat{Y} \in \mathcal{B}(0, 1)$. Then a relation between the condition number of $M(\bar{\phi}, \hat{Y})$ and the measure of Λ-poisedness can be established. [Conn, Scheinberg, and Vicente, 2008]

Using the matrix $M(\bar{\phi}, \hat{Y})$ directly in a DFO algorithm enables to use $\kappa(M(\bar{\phi}, \hat{Y}))$ to monitor poisedness of the interpolation set without computing Lagrange polynomial bases and Λ.

Self-correcting geometry

Design and convergence properties of new algorithm depend on a self-correction mechanism combining trust-region mechanism with polynomial interpolation setting.

Self-correcting property:

If iteration k is unsuccessful, the interpolation set is included in the trust region and $\Delta_k \leq \kappa_\Lambda ||\nabla m_k||$, then there exists an interpolation point j which can be replaced with $\ell_{k,j}(x_k^+) > \Lambda$.

And so, every unsuccessful iteration must result in an improvement of the interpolation set geometry. [Scheinberg and Toint, 2009]