# A recursive model-based trust-region method for derivative-free bound-constrained optimization.

ANKE TRÖLTZSCH [CERFACS, TOULOUSE, FRANCE]

JOINT WORK WITH:

SERGE GRATTON [ENSEEIHT, TOULOUSE, FRANCE]
PHILIPPE L. TOINT [FUNDP, NAMUR, BELGIUM]

FOIE 2010, CERFACS, Toulouse

$7^{th}$ of July 2010

# Outline

1. Introduction
2. Interpolation models and poisedness
3. Geometry control in DFO trust region methods
4. Extension to bounds
5. Numerical experiments
6. Conclusions and Perspectives

# Outline

1 Introduction
2 Interpolation models and poisedness
3 Geometry control in DFO trust region methods
4 Extension to bounds
5 Numerical experiments
6 Conclusions and Perspectives

# Why using derivative-free optimization?

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

- Derivatives are unavailable
- 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 in 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

1 Introduction
2 Interpolation models and poisedness
3 Geometry control in DFO trust region methods
4 Extension to bounds
5 Numerical experiments
6 Conclusions and Perspectives

# 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 $\phi$, 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 $\alpha_j$ 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

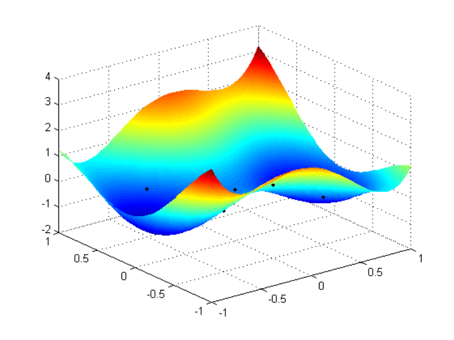



Figure: Six-hump camel back function with sample points

# Lagrange polynomials - Illustration

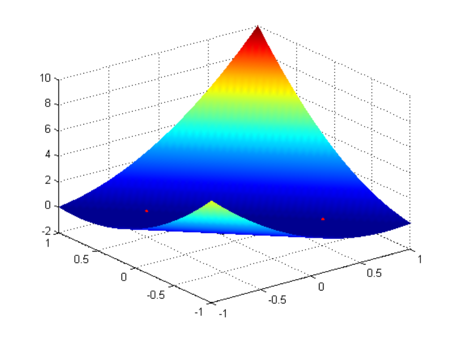



Figure: First Lagrange polynomial

# Lagrange polynomials - Illustration

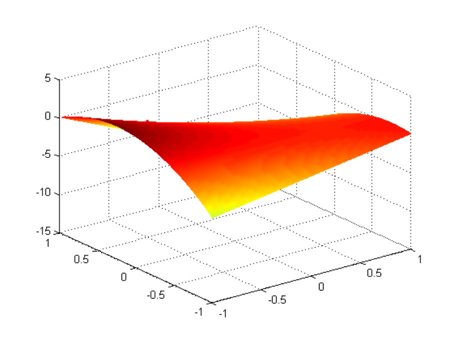



Figure: Second Lagrange polynomial

# Lagrange polynomials - Illustration

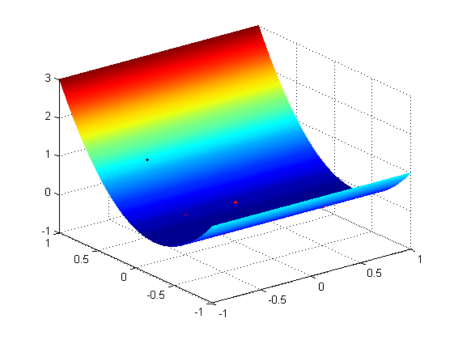



Figure: Third Lagrange polynomial

# Lagrange polynomials - Illustration

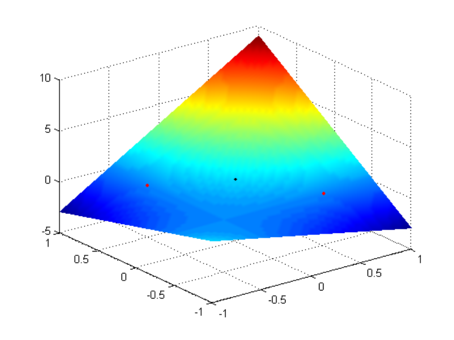



Figure: Fourth Lagrange polynomial

# Lagrange polynomials - Illustration

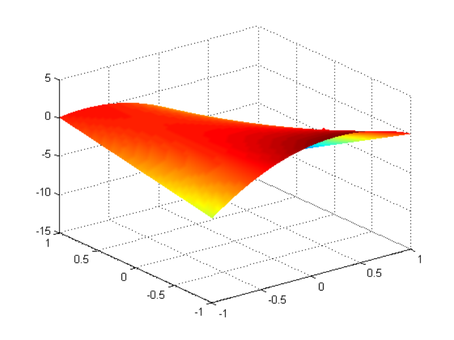



Figure: Fifth Lagrange polynomial

# Lagrange polynomials - Illustration

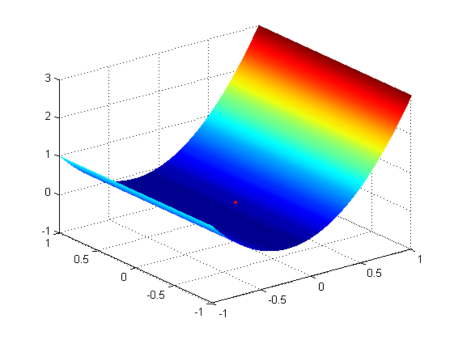



Figure: Sixth Lagrange polynomial

# Lagrange polynomials - Illustration

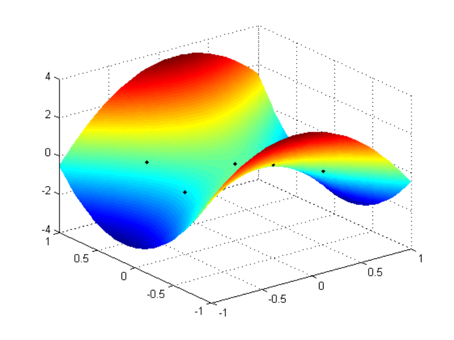



Figure: The resulting interpolation polynomial

# 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 $\Lambda$-poised in $\mathcal{B}$ if one has that

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

The smaller $\Lambda$, 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 asscociated 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

1. Introduction
2. Interpolation models and poisedness
3. Geometry control in DFO trust region methods
4. Extension to bounds
5. Numerical experiments
6. Conclusions and Perspectives

# A simple DFO trust-region algorithm

- Compute an initial poised interpolation set $Y_0$
- Test for convergence
- Build a quadratic model $m_k(x_k + s)$ of the objective function around an iterate $x_k$

$$m_k(x_k + s) = f(x_k) + g(x_k)^T s + \frac{1}{2} s^T H s$$

  based on well-poised sample sets.
- Calculate a new trial point $x_k^+$ by solving

$$\min_{s \in B(x_k;\Delta_k)} m_k(x_k + s).$$

  in the trust region $B(x_k; \Delta_k)$.

# A simple DFO trust-region algorithm

- Evaluate $f(x_k^+)$ and compute the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m(x_k) - m(x_k + s_k)} = \frac{\text{achieved reduction}}{\text{predicted reduction}}.$$

- Define the next iterate

  • **case 1)** Successful iteration: set $x_{k+1} = x_k^+$, increase $\Delta_k$ and include point in the set $Y_{k+1}$

  • **case 2)** Unsuccessful iteration: set $x_{k+1} = x_k$, decrease $\Delta_k$ and include point in the set if its closer to $x_k$ than the furthest in $Y_k$

- Compute the new interpolation model $m_{k+1}$ around $x_{k+1}$ using interpolation set $Y_{k+1}$ if $Y_{k+1} \neq Y_k$, increment $k$

# Geometry improving steps

- Fasano, Nocedal, and Morales [2009] observed that an algorithm which simply ignores the 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].
- Failure of current iteration might be due to a too large trust region or a bad quality of the interpolation model (set not well-poised).
- Shows that we cannot afford to do without a geometry phase (need to maintain quality of the geometry of the interpolation set).
- Improvement is usually carried out at special ”geometry improving” steps by computing additional function values at well-chosen points.

# A DFO trust-region alg. with geometry restoration

- Compute an initial poised interpolation set $Y_0$
- Test for convergence
- Build a quadratic model $m_k(x_k + s)$ of the objective function around an iterate $x_k$

$$m_k(x_k + s) = f(x_k) + g(x_k)^T s + \frac{1}{2} s^T H s$$

  based on well-poised sample sets.
- Calculate a new trial point $x_k^+$ by solving

$$\min_{s \in B(x_k;\Delta_k)} m_k(x_k + s).$$

  in the trust region $B(x_k; \Delta_k)$.

# A DFO trust-region alg. with geometry restoration

- Evaluate $f(x_k^+)$ and compute the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m(x_k) - m(x_k + s_k)}.$$

- Define the next iterate

  • **case 1)** Successful iteration: set $x_{k+1} = x_k^+$, increase $\Delta_k$ and include point in the set $Y_{k+1}$

  • **case 2)** Unsuccessful iteration: set $x_{k+1} = x_k$, decrease $\Delta_k$ and include point in the set if its closer to $x_k$ than the furthest in $Y_k$

- Improve interpolation set by a geometry improving step
- Compute the new interpolation model $m_{k+1}$ around $x_{k+1}$ using interpolation set $Y_{k+1}$ if $Y_{k+1} \neq Y_k$, increment $k$

# Geometry improving steps

- As those geometry restoration steps are expensive, one may ask if they are really necessary.
- Idea is now to reduce the frequency and cost of the necessary tests as much as possible, while maintaining a mechanism for taking geometry into account.
- Design and convergence properties of new algorithm depend on a self-correction mechanism combining trust-region mechanism with polynomial interpolation setting.

# The new DFO trust-region algorithm

- Compute an initial poised interpolation set $Y_0$
- Test for convergence and improve geometry if necessary
- Build a quadratic model $m_k(x_k + s)$ of the objective function around an iterate $x_k$

$$m_k(x_k + s) = f(x_k) + g(x_k)^T s + \frac{1}{2} s^T H s$$

  based on the current interpolation set.
- Calculate a new trial point $x_k^+$ by solving

$$\min_{s \in B(x_k; \Delta_k)} m_k(x_k + s).$$

  in the trust region $B(x_k; \Delta_k)$.

# The new DFO trust-region algorithm

- Evaluate $f(x_k^+)$ and compute the ratio $\rho_k$
- Define the next iterate

  • **case 1)** Successful iteration: include point in the set $Y_{k+1}$, adjust $\Delta$ and define $x_{k+1} = x_k^+$

  • **case 2)** Try to replace a far interpolation point: if set $F_k$ is non-empty, include point in the set $Y_{k+1}$, set $\Delta_{k+1} = \Delta_k$

  • **case 3)** Try to replace a close interpolation point: if set $F_k = \emptyset$ and set $C_k$ is non-empty, include point in the set $Y_{k+1}$, set $\Delta_{k+1} = \Delta_k$

  • **case 4)** Reduce trust-region radius and set $Y_{k+1} = Y_k$.

# Self-correcting geometry

Set of far points:

$$F_k = \{y_{k,j} \in Y_k \text{ such that } ||y_{k,j} - x_{best}|| > \beta\Delta \text{ and } \ell_{k,j}(x_k^+) \neq 0\}$$

Set of close points:

$$C_k = \{y_{k,j} \in Y_k \text{ such that } ||y_{k,j} - x_{best}|| \leq \beta\Delta \text{ and } \ell_{k,j}(x_k^+) > \Lambda\}$$

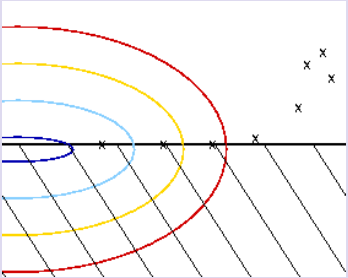
Self-correcting property:

If iteration $k$ is unsuccessful, $F_k = \emptyset$ and $\Delta_k \leq \kappa_\Lambda ||\nabla m_k||$, then $C_k \neq \emptyset$, and so, every unsuccessful iteration must result in an improvement of the interpolation set geometry. [Scheinberg and Toint, 2009]

# Outline

1. Introduction
2. Interpolation models and poisedness
3. Geometry control in DFO trust region methods
4. Extension to bounds
5. Numerical experiments
6. Conclusions and Perspectives

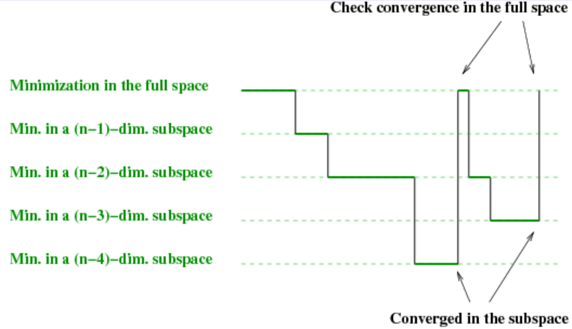
# Extension to bounds



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

# Solution: a subspace method

## Continue minimization in a smaller dimensional subspace



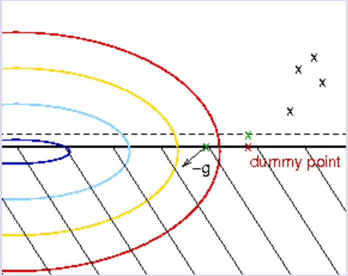

- If encounter an active bound, reduce dimensionality
- If converged in the subspace, going back to check convergence in the full space

# Further features of the algorithm (I)

- Degree of initial interpolation model user-defined: linear, diagonal, quadratic
- Adjust the initial trust region and shift the starting point to build the initial model inside the bounds
- Using variable size models: unless model is quadratic, new iterates augment the size of the interpolation set
- Initial degree of subspace-models is linear and is then augmented with the new iterates computed in the subspace
- Recursive technique: call the algorithm itself to solve the problem in the subspace(s)

# Further features of the algorithm (II)

## 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
- Compute the model values at the dummy points
- Take real points and dummy points lying in the subspace to build the model
- Dummy points are then replaced by the new iterates

# Outline

1 Introduction

2 Interpolation models and poisedness

3 Geometry control in DFO trust region methods

4 Extension to bounds

5 Numerical experiments

6 Conclusions and Perspectives

# Methodology

## CUTEr testing environment

- 50 bound-constrained test cases from CUTEr test environment
- Nbr. of variables varies from 1 to 25 dimensions

## Competitor: BOBYQA

- State of the art software developed by M.J.D. Powell [2006]
- Currently one of the best codes for bound-constrained minimization without derivatives

## Stopping criterion

- Stopping criteria are different
- Using optimal objective function value computed by TRON (using first and second derivatives) as a reference
- We terminate when 6 correct significant figures in $f$ were attained

# Numerical results

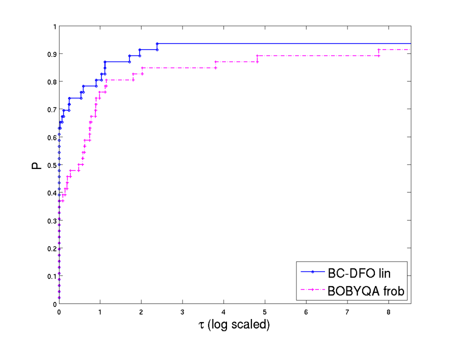



Figure: Performance profile in terms of nbr. of function eval.

# A success in solving a 25-dim. problem

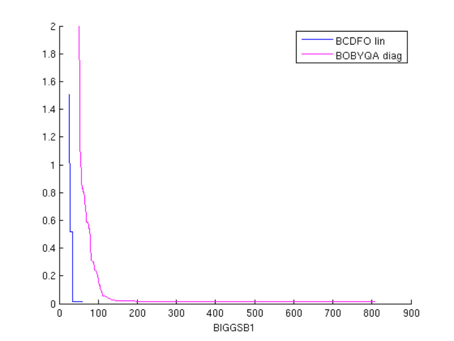



Figure: Convergence history of problem BIGGSB1

# Outline

1. Introduction
2. Interpolation models and poisedness
3. Geometry control in DFO trust region methods
4. Extension to bounds
5. Numerical experiments
6. Conclusions and Perspectives

# 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!