A Subspace Decomposition Framework for Nonlinear Optimization: Global Convergence and Global Rate

S. Gratton

CERFACS and ENSEEIHT, France

May 19, 2014 — SIOPT, San Diego

http://www.cerfacs.fr/~gratton

Outline

In this talk, to make things simple:

- we consider an unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x);$$

In this talk, to make things simple:

- we consider an unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x);$$

- we suppose that minimising f directly is out of reach.

Derivative-free optimization

In this talk, to make things simple:

- we consider an unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x);$$

- we suppose that minimising f directly is out of reach.
- we aim for methods that are based on decomposition of the variable space derivative-free methods, or large scale optimisation with derivatives.

In this talk, to make things simple:

- we consider an unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x);$$

- we suppose that minimising f directly is out of reach.
- we aim for methods that are based on decomposition of the variable space derivative-free methods, or large scale optimisation with derivatives.
- These methods are naturally adapted to massively parallel computers. But: convergence, fault tolerence

Subspace techniques (I)

- Domain decomposition for PDE's based on Schwarz techniques. The full space is decomposed.
  - Conditions of convergence exist (fixed point theory using a generalized Cauchy-Schwarz inequality)
  - Crucial role of the problem-based decomposition (overlapping, coarse space,..)
- ASPIN from Cai and Keyes (for nonliear equaitons)
- N. Gould, A. Sartenaer, and Ph. L. Toint, On iterated-subspace minimization methods for nonlinear optimization. Rutherford Appleton Laboratory, 1994
- Y. Yuan, Subspace techniques for nonlinear optimization. Some topics in industrial and applied mathematics 8 (2007): 206–218

Subspace techniques (II)

- Block Jacobi (linear/nonlinear equations), block coordinate descent
- M. C. Ferris, O. L. Mangasarian, Parallel variable distribution. SIAM Journal on Optimization 4, no. 4 (1994): 815–832
- M. Fukushima, Parallel variable transformation in unconstrained optimization. SIAM Journal on Optimization 8, no. 3 (1998): 658–672
- S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, Notes on decomposition methods. Notes for EE364B, Stanford University, 2007
- C. Audet, J. E. Dennis Jr, and S. Le Digabel, Parallel space decomposition of the mesh adaptive direct search algorithm. SIAM Journal on Optimization 19, no. 3 (2008): 1150–1170

A subspace decomposition framework

- Suppose that the current iterate is x_k.

A subspace decomposition framework

- Suppose that the current iterate is x_k.
- Decomposition:

A subspace decomposition framework

- Suppose that the current iterate is x_k.
- Decomposition:
  - select spaces $\mathcal{S}_k^{(1)}, \mathcal{S}_k^{(2)}, \ldots, \mathcal{S}_k^{(m_k)}$ such that

$$\mathbb{R}^n = \sum_{i=0}^{m_k} \mathcal{S}_k^{(i)};$$

A subspace decomposition framework

- Suppose that the current iterate is x_k.
- Decomposition:
  - select spaces $\mathcal{S}_k^{(1)}, \mathcal{S}_k^{(2)}, \ldots, \mathcal{S}_k^{(m_k)}$ such that

$$\mathbb{R}^n = \sum_{i=0}^{m_k} \mathcal{S}_k^{(i)};$$

    or use (at least one) gradient related subspace(s) in a truncated sum
  - minimize $f(x_k + d)$ with respect to d on $\mathcal{S}_k^{(i)}$, and obtain $d_k^{(i)}$ $(i = 1, 2, \ldots, m_k)$.

A subspace decomposition framework

- Suppose that the current iterate is x_k.
- Decomposition:
  - select spaces $\mathcal{S}_k^{(1)}, \mathcal{S}_k^{(2)}, \ldots, \mathcal{S}_k^{(m_k)}$ such that

$$\mathbb{R}^n = \sum_{i=0}^{m_k} \mathcal{S}_k^{(i)};$$

    or use (at least one) gradient related subspace(s) in a truncated sum
  - minimize $f(x_k + d)$ with respect to d on $\mathcal{S}_k^{(i)}$, and obtain $d_k^{(i)}$ $(i = 1, 2, \ldots, m_k)$.
- How to compose the step d_k ?

A subspace decomposition framework

- Suppose that the current iterate is x_k.
- Decomposition:
  - select spaces $\mathcal{S}_k^{(1)}, \mathcal{S}_k^{(2)}, \ldots, \mathcal{S}_k^{(m_k)}$ such that

$$\mathbb{R}^n = \sum_{i=0}^{m_k} \mathcal{S}_k^{(i)};$$

    or use (at least one) gradient related subspace(s) in a truncated sum
  - minimize $f(x_k + d)$ with respect to d on $\mathcal{S}_k^{(i)}$, and obtain $d_k^{(i)}$ $(i = 1, 2, \ldots, m_k)$.
- How to compose the step d_k ?
  - Set

$$d_k = \sum_{i=0}^{m_k} d_k^{(i)} \; ?$$

A subspace decomposition framework

- Suppose that the current iterate is x_k.
- Decomposition:
  - select spaces $\mathcal{S}_k^{(1)}, \mathcal{S}_k^{(2)}, \ldots, \mathcal{S}_k^{(m_k)}$ such that
  $$\mathbb{R}^n = \sum_{i=0}^{m_k} \mathcal{S}_k^{(i)};$$
  - minimize $f(x_k + d)$ with respect to d on $\mathcal{S}_k^{(i)}$, and obtain $d_k^{(i)}$ $(i = 1, 2, \ldots, m_k)$.
- Composition (M. Fukushima, 1998)

A subspace decomposition framework

- Suppose that the current iterate is x_k.
- Decomposition:
  - select spaces $\mathcal{S}_k^{(1)}, \mathcal{S}_k^{(2)}, \ldots, \mathcal{S}_k^{(m_k)}$ such that
$$\mathbb{R}^n = \sum_{i=0}^{m_k} \mathcal{S}_k^{(i)};$$
  - minimize $f(x_k + d)$ with respect to d on $\mathcal{S}_k^{(i)}$, and obtain $d_k^{(i)}$ $(i = 1, 2, \ldots, m_k)$.
- Composition (M. Fukushima, 1998)
  - set
$$\mathcal{S}_k = \operatorname{span}\left\{d_k^{(1)}, d_k^{(2)}, \ldots, d_k^{(m_k)}\right\};$$

- Suppose that the current iterate is x_k.
- Decomposition:
  - select spaces $\mathcal{S}_k^{(1)}, \mathcal{S}_k^{(2)}, \ldots, \mathcal{S}_k^{(m_k)}$ such that
$$\mathbb{R}^n = \sum_{i=0}^{m_k} \mathcal{S}_k^{(i)};$$
  - minimize $f(x_k + d)$ with respect to d on $\mathcal{S}_k^{(i)}$, and obtain $d_k^{(i)}$ $(i = 1, 2, \ldots, m_k)$.
- Composition (M. Fukushima, 1998)
  - set
$$\mathcal{S}_k = \operatorname{span}\left\{ d_k^{(1)}, d_k^{(2)}, \ldots, d_k^{(m_k)} \right\};$$
  - minimize $f(x_k + d)$ with respect to d on $\mathcal{S}_k$, and obtain d_k.

Globalization technique

Let $h_k^{(i)}$ be a model of f that "satisfies" $P_k^{(i)} \nabla h_k^{(i)}(x_k) = P_k^{(i)} \nabla f(x_k)$.

Globalization technique

Let $h_k^{(i)}$ be a model of f that "satisfies" $P_k^{(i)}\nabla h_k^{(i)}(x_k) = P_k^{(i)}\nabla f(x_k)$.

- Levenberg-Marquardt (natural for least-squares, nonlinear filtering):

$$\min_{d\in\mathcal{S}_k^{(i)}} h_k^{(i)}(x_k+d) + \frac{1}{2}\sigma_k\|d\|^2$$

Globalization technique

Let $h_k^{(i)}$ be a model of f that "satisfies" $P_k^{(i)} \nabla h_k^{(i)}(x_k) = P_k^{(i)} \nabla f(x_k)$.

- Levenberg-Marquardt (natural for least-squares, nonlinear filtering):

$$\min_{d \in \mathcal{S}_k^{(i)}} h_k^{(i)}(x_k + d) + \frac{1}{2}\sigma_k \|d\|^2$$

- Trust-region:

$$\begin{aligned} \min\ & h_k^{(i)}(x_k + d) \\ \text{s.t. } & d \in \mathcal{S}_k^{(i)} \\ & \|d\| \leq \Delta_k \end{aligned}$$

Globalization technique

Let $h_k^{(i)}$ be a model of f that "satisfies" $P_k^{(i)}\nabla h_k^{(i)}(x_k) = P_k^{(i)}\nabla f(x_k)$.

- Levenberg-Marquardt (natural for least-squares, nonlinear filtering):

$$\min_{d\in\mathcal{S}_k^{(i)}} h_k^{(i)}(x_k+d) + \frac{1}{2}\sigma_k\|d\|^2$$

- Trust-region:

$$\begin{aligned} &\min\ h_k^{(i)}(x_k+d) \\ &\text{s.t. } d \in \mathcal{S}_k^{(i)} \\ &\qquad \|d\| \le \Delta_k \end{aligned}$$

- How to update Δ_k or σ_k?

Globalization technique

Let $h_k^{(i)}$ be a model of f that "satisfies" $P_k^{(i)} \nabla h_k^{(i)}(x_k) = P_k^{(i)} \nabla f(x_k)$.

- Levenberg-Marquardt (natural for least-squares, nonlinear filtering):

$$\min_{d \in \mathcal{S}_k^{(i)}} h_k^{(i)}(x_k + d) + \frac{1}{2}\sigma_k \|d\|^2$$

- Trust-region:

$$\begin{aligned} \min\ & h_k^{(i)}(x_k + d) \\ \text{s.t. } & d \in \mathcal{S}_k^{(i)} \\ & \|d\| \le \Delta_k \end{aligned}$$

- How to update Δ_k or σ_k?

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{\sum_{i=1}^{m_k} \left[f(x_k) - h_k^{(i)}(x_k + d_k^{(i)}) \right]}$$

Globalization technique

Let $h_k^{(i)}$ be a model of f that "satisfies" $P_k^{(i)} \nabla h_k^{(i)}(x_k) = P_k^{(i)} \nabla f(x_k)$.

- Levenberg-Marquardt:

$$\min_{d \in \mathcal{S}_k^{(i)}} h_k^{(i)}(x_k + d) + \frac{1}{2}\sigma_k \|d\|^2$$

- Trust-region:

$$\begin{aligned} \min\ & h_k^{(i)}(x_k + d) \\ \text{s.t. } & d \in \mathcal{S}_k^{(i)} \\ & \|d\| \le \Delta_k \end{aligned}$$

- How to update Δ_k or σ_k?

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{\sum_{i=1}^{m_k} \left[f(x_k) - h_k^{(i)}(x_k + d_k^{(i)}) \right]}$$

Levenberg-Marquardt framework

Algorithm (Levenberg-Marquardt framework)

Algorithm (Levenberg-Marquardt framework)

Step 1. Select a constant $\eta \in [0, 1)$, pick a starting point $x_0 \in \mathbb{R}^n$, choose a positive number σ_0, and select constants η_1, η_2, η_3, θ, and γ so that

$$0 \le \eta_1 \le \eta_2 \le \eta_3, \text{ and } 0 < \theta < 1 < \gamma. \tag{1}$$

Set $k = 0$.

Algorithm (Levenberg-Marquardt framework)

Step 1. Select a constant $\eta \in [0, 1)$, pick a starting point $x_0 \in \mathbb{R}^n$, choose a positive number σ_0, and select constants η_1, η_2, η_3, θ, and γ so that

$$0 \leq \eta_1 \leq \eta_2 \leq \eta_3, \text{ and } 0 < \theta < 1 < \gamma. \tag{1}$$

Set $k = 0$.

Step 2. Choose nonzero subspaces $\mathcal{S}_k^{(i)}$ $(i = 1, 2, \cdots, m_k)$ of $\mathbb{R}^n$ so that

$$\sum_{i=1}^{m_k} \mathcal{S}_k^{(i)} = \mathbb{R}^n.$$

Algorithm (Levenberg-Marquardt framework)

Step 1. Select a constant $\eta \in [0,1)$, pick a starting point $x_0 \in \mathbb{R}^n$, choose a positive number σ_0, and select constants η_1, η_2, η_3, θ, and γ so that

$$0 \le \eta_1 \le \eta_2 \le \eta_3, \text{ and } 0 < \theta < 1 < \gamma. \tag{1}$$

Set $k = 0$.

Step 2. Choose nonzero subspaces $\mathcal{S}_k^{(i)}$ $(i = 1, 2, \cdots, m_k)$ of $\mathbb{R}^n$ so that

$$\sum_{i=1}^{m_k} \mathcal{S}_k^{(i)} = \mathbb{R}^n.$$

Step 3. For $i = 1, 2, \ldots, m_k$, approximately solve the following problem to get $d_k^{(i)}$:

$$\min_{d \in \mathcal{S}_k^{(i)}} h_k^{(i)}(x_k + d) + \frac{1}{2}\sigma_k \|d\|^2.$$

Algorithm (Levenberg-Marquardt framework cont.)

Step 4. Determine $d_k \in \mathrm{span}\{d_k^{(1)}, d_k^{(2)}, \cdots, d_k^{(m_k)}\}$ so that

$$f(x_k + d_k) \leq f(x_k + \sum_{i=1}^{m_k} d_k^{(i)}).$$

Algorithm (Levenberg-Marquardt framework cont.)

Step 4. Determine $d_k \in \mathrm{span}\{d_k^{(1)}, d_k^{(2)}, \cdots, d_k^{(m_k)}\}$ so that

$$f(x_k + d_k) \le f(x_k + \sum_{i=1}^{m_k} d_k^{(i)}).$$

Step 5. Let

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{\sum_{i=1}^{m_k} \left[f(x_k) - h_k^{(i)}(x_k + d_k^{(i)}) \right]},$$

and set

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } \rho_k \ge \eta_1 \\ x_k & \text{if } \rho_k < \eta_1. \end{cases}$$

Algorithm (Levenberg-Marquardt framework cont.)

Step 6. Set

$$\sigma_{k+1} = \begin{cases} \theta\sigma_k & \text{if } \rho_k \geq \eta_3 \\ \sigma_k & \text{if } \eta_2 \leq \rho_k < \eta_3 \\ \gamma\sigma_k & \text{if } \rho_k < \eta_2. \end{cases}$$

Increment k by 1, and go to **Step 2**.

Trust-region framework

Algorithm (Trust-region framework)

Trust-region framework

Algorithm (Trust-region framework)

Step 1. Select a constant $\eta \in [0, 1)$, pick a starting point $x_0 \in \mathbb{R}^n$, choose $\Delta_0 > 0$, and select constants η_1, η_2, η_3, θ, and γ so that

$$0 \leq \eta_1 \leq \eta_2 \leq \eta_3, \text{ and } 0 < \theta < 1 < \gamma. \tag{2}$$

Set $k = 0$.

Trust-region framework

Algorithm (Trust-region framework)

Step 1. Select a constant $\eta \in [0, 1)$, pick a starting point $x_0 \in \mathbb{R}^n$, choose $\Delta_0 > 0$, and select constants η_1, η_2, η_3, θ, and γ so that

$$0 \leq \eta_1 \leq \eta_2 \leq \eta_3, \text{ and } 0 < \theta < 1 < \gamma. \tag{2}$$

Set $k = 0$.

Step 2. Choose subspaces $\mathcal{S}_k^{(i)}$ $(i = 1, 2, \cdots, m_k)$ of $\mathbb{R}^n$ so that

$$\sum_{i=1}^{m_k} \mathcal{S}_k^{(i)} = \mathbb{R}^n.$$

Trust-region framework

Algorithm (Trust-region framework)

Step 1. Select a constant $\eta \in [0,1)$, pick a starting point $x_0 \in \mathbb{R}^n$, choose $\Delta_0 > 0$, and select constants η_1, η_2, η_3, θ, and γ so that

$$0 \leq \eta_1 \leq \eta_2 \leq \eta_3, \text{ and } 0 < \theta < 1 < \gamma. \tag{2}$$

Set $k = 0$.

Step 2. Choose subspaces $\mathcal{S}_k^{(i)}$ $(i = 1, 2, \cdots, m_k)$ of $\mathbb{R}^n$ so that

$$\sum_{i=1}^{m_k} \mathcal{S}_k^{(i)} = \mathbb{R}^n.$$

Step 3. For $i = 1, 2, \ldots, m_k$, approximately solve the following problem to obtain $d_k^{(i)}$:

$$\min_{d \in \mathcal{S}_k^{(i)}} \; h_k^{(i)}(x_k + d)$$
$$\text{s.t. } \|d\| \leq \Delta_k.$$

Algorithm (Trust-region framework cont.)

Step 4. Determine $d_k \in \mathrm{span}\{d_k^{(1)}, d_k^{(2)}, \cdots, d_k^{(m_k)}\}$ so that

$$f(x_k + d_k) \leq f(x_k + \sum_{i=1}^{m_k} d_k^{(i)}).$$

Trust-region framework

Algorithm (Trust-region framework cont.)

Step 4. Determine $d_k \in \mathrm{span}\{d_k^{(1)}, d_k^{(2)}, \cdots, d_k^{(m_k)}\}$ so that

$$f(x_k + d_k) \le f(x_k + \sum_{i=1}^{m_k} d_k^{(i)}).$$

Step 5. Let

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{\sum_{i=1}^{m_k} \left[f(x_k) - h_k^{(i)}(x_k + d_k^{(i)}) \right]},$$

and set

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } \rho_k \ge \eta_1 \\ x_k & \text{if } \rho_k < \eta_1. \end{cases}$$

Algorithm (Trust-region framework cont.)

Step 6. Set

$$\Delta_{k+1} = \begin{cases} \gamma\Delta_k & \text{if } \rho_k \geq \eta_3 \\ \Delta_k & \text{if } \eta_2 \leq \rho_k < \eta_3 \\ \theta\Delta_k & \text{if } \rho_k < \eta_2. \end{cases}$$

Increment k by 1, and go to **Step 2**.

Assumptions

Assumption

Assumptions

Assumption

1. *The function f is bounded from below and twice continuously differentiable, and $\nabla^2 f$ is bounded on $\mathbb{R}^n$.*

Assumptions

Assumption

1. *The function f is bounded from below and twice continuously differentiable, and $\nabla^2 f$ is bounded on $\mathbb{R}^n$.*

2. *The sequence $\{m_k\}$ is bounded.*

Assumptions

Assumption

1. *The function f is bounded from below and twice continuously differentiable, and $\nabla^2 f$ is bounded on $\mathbb{R}^n$.*
2. *The sequence $\{m_k\}$ is bounded.*
3. *The smallest eigenvalues of $\sum_{i=1}^{m_k} P_k^{(i)}$ are bounded away from zero, where $P_k^{(i)}$ is the orthogonal projection matrix from $\mathbb{R}^n$ onto $\mathcal{S}_k^{(i)}$.*

Assumption

1. *The function f is bounded from below and twice continuously differentiable, and $\nabla^2 f$ is bounded on $\mathbb{R}^n$.*
2. *The sequence $\{m_k\}$ is bounded.*
3. *The smallest eigenvalues of $\sum_{i=1}^{m_k} P_k^{(i)}$ are bounded away from zero, where $P_k^{(i)}$ is the orthogonal projection matrix from $\mathbb{R}^n$ onto $\mathcal{S}_k^{(i)}$.*
4. $\left\{\nabla^2 h_k^{(i)}\right\}$ *is uniformly bounded on $\mathbb{R}^n$.*

Assumptions

Assumption

1. *The function f is bounded from below and twice continuously differentiable, and $\nabla^2 f$ is bounded on $\mathbb{R}^n$.*
2. *The sequence $\{m_k\}$ is bounded.*
3. *The smallest eigenvalues of $\sum_{i=1}^{m_k} P_k^{(i)}$ are bounded away from zero, where $P_k^{(i)}$ is the orthogonal projection matrix from $\mathbb{R}^n$ onto $\mathcal{S}_k^{(i)}$.*
4. $\left\{\nabla^2 h_k^{(i)}\right\}$ *is uniformly bounded on $\mathbb{R}^n$.*
5. *On subspace $\mathcal{S}_k^{(i)}$, $d_k^{(i)}$ achieves a decrease proportional to the Cauchy decrease.*

Global convergence

Theorem

Suppose that the assumptions stated before hold, then the iterates $\{x_k\}$ generated by the Levenberg-Marquardt framework satisfy

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Sketch of the proof:

- The k-th iteration is successful if σ_k is large enough.
- $\{\sigma_k\}$ is bounded from above.
- Contradiction if $\liminf_{k\to\infty} \|\nabla f(x_k)\| > 0$.

Global convergence

Theorem

Suppose that the assumptions stated before hold, and $\eta_1 > 0$, then the iterates $\{x_k\}$ generated by either of the Levenberg-Marquardt framework satisfy

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Sketch of the proof:

- $\{\sigma_k\}$ is bounded from above.
- $\{f(x_k)\}$ is decreasing and bounded from below.
- The convergence follows from the decreases obtained when the iterates move.

Global rate

Theorem

Suppose that the assumptions stated before hold, then the iterates $\{x_k\}$ generated by the Levenberg-Marquardt framework satisfy

$$\min_{0 \leq \ell \leq k} \|\nabla f(x_\ell)\| \leq C \sqrt{\frac{1}{k}}.$$

Sketch of the proof:

- Because of the boundedness of $\{\sigma_k\}$, the number of unsuccessful iterations is linearly bounded by that of the successful ones.
- Because of the boundedness of $\{f(x_k)\}$ and the decreases achieved on the successful iterations, the number of successful iterations is bounded by $\mathcal{O}(\epsilon^{-2})$.

Similar results can be derived for the trust-region framework.

Properties of the framework

Properties of the framework

- *It can be applied to optimization with or without derivatives.*

Applications to derivative-free optimization

Properties of the framework

- *It can be applied to optimization with or without derivatives.*
- *It is naturally parallel.*

Applications to derivative-free optimization

Properties of the framework

- *It can be applied to optimization with or without derivatives.*
- *It is naturally parallel.*
- *It is naturally multilevel.*

Applications to derivative-free optimization

Properties of the framework

- *It can be applied to optimization with or without derivatives.*
- *It is naturally parallel.*
- *It is naturally multilevel.*

$$\Downarrow$$

Our goal

Parallel and multilevel algorithms without using derivatives and capable of solving relatively large problems.

Very preliminary numerical results

- Use the Levenberg-Marquardt framework
- Subproblem solver: `NEWUOA`
- Number of subspaces: $\sqrt{n/2}$
- Benchmark: `NEWUOA` (`NPT=2N+1; RHOEND=1.0E-6`)
- Very preliminary: not parallel, not multilevel, not large-scale . . .
- Dimension of test problems: 25, 30, 35, 40
- Denote our code as `SSD`

VARDIM

Table: Numerical results of VARDIM

n	25	30	35	40	
$\#f$	8343	8926	12689	17741	NEWUOA
	3592	6222	7507	16653	SSD
f_{final}	1.61E-11	4.08E-11	4.93E-11	1.76E-10	NEWUOA
	9.74E-11	6.85E-10	5.74E-11	7.89E-13	SSD

$$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 + \Big[\sum_{i=1}^{n} i(x_i - 1)\Big]^2 + \Big[\sum_{i=1}^{n} i(x_i - 1)\Big]^4$$

PENALTY1

Table: Numerical results of PENALTY1

n	25	30	35	40	
$\#f$	9532	10947	14427	13577	NEWUOA
	2089	2784	2348	2812	SSD
f_{final}	2.03E-04	2.48E-04	2.93E-04	3.39E-04	NEWUOA
	2.04E-04	2.50E-04	2.95E-04	3.41E-04	SSD

$$f(x) = 10^{-15} \sum_{i=1}^{n} (x_i - 1)^2 + \left(\frac{1}{4} - \sum_{i=1}^{n} x_i^2 \right)^2$$

Table: Numerical results of SBRYBND

n	25	30	35	40	
$\#f$	968	576	2052	2363	NEWUOA
	27889	53103	90304	206608	SSD
f_{final}	235	326	342	395	NEWUOA
	3.08	3.08	3.08	3.08	SSD
	134	284	233	229	

$$f(x) = \sum_{i=1}^{n} \left[(2 + 5p_i^2 x_i^2) p_i x_i + 1 - \sum_{j \in J_i} p_j x_j (1 + p_j x_j) \right]^2,$$

where $J_i = \{j \mid j \neq i, \max\{1, i-5\} \leq j \leq \min\{n, j+1\}\}$, and $p_i = \exp\left(6\frac{i-1}{n-1}\right)$.

CHROSEN

Table: Numerical results of CHROSEN

n	25	30	35	40	
$\#f$	1123	1445	1717	1859	NEWUOA
	96040	103296	127726	142272	SSD
f_{final}	8.94E-12	1.07E-11	1.13E-11	3.14E-11	NEWUOA
	2.95E-10	5.49E-10	7.26E-10	8.09E-10	SSD

$$f(x) = \sum_{i=1}^{n-1} \left[4(x_i - x_{i+1}^2)^2 + (1 - x_{i+1})^2 \right]$$

Concluding remarks

- A subspace decomposition framework (two versions) with global convergence and known convergence rate
- Possible to develop parallel and multilevel methods with or without derivative

Concluding remarks

- A subspace decomposition framework (two versions) with global convergence and known convergence rate
- Possible to develop parallel and multilevel methods with or without derivative
- "Clever" way of choosing subspaces . . .

Concluding remarks

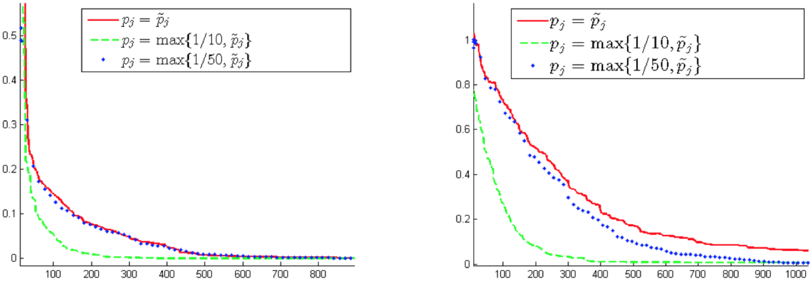
- A subspace decomposition framework (two versions) with global convergence and known convergence rate
- Possible to develop parallel and multilevel methods with or without derivative
- "Clever" way of choosing subspaces . . .
    - not try to cover the whole space, but . . .

Concluding remarks

- A subspace decomposition framework (two versions) with global convergence and known convergence rate
- Possible to develop parallel and multilevel methods with or without derivative
- “Clever” way of choosing subspaces . . .
  - not try to cover the whole space, but . . .
  - choose subspaces randomly

Concluding remarks

- A subspace decomposition framework (two versions) with global convergence and known convergence rate
- Possible to develop parallel and multilevel methods with or without derivative
- “Clever” way of choosing subspaces . . .
  - not try to cover the whole space, but . . .
  - choose subspaces randomly
  - take into account failures in the step computation



(a) Average of function values. (b) Average of absolute error of iterates.

Levenberg-Marquart with wrong gradient at a given iteration frequency on Rosebrock

Thanks!