A robust combined trust region–line search exact penalty structured approach for constrained nonlinear least squares problems

N. Mahdavi-Amiri
Sharif University of Technology
Tehran, Iran

M. R. Ansari
IKI University
Qazvin, Iran

The 8th International Conference of Iranian Operations Research Society
Outline

- Problem description and structure
- Exact penalty method
- Trust region and line search methods
- Step directions: trust region projected quadratic subproblems
- Penalty parameter updating strategy
- Projected structured Hessian approximation
- Special nonsmooth line search
- Computational results
- Conclusions
- References
Problem description and structure

Constrained NonLinear Least Squares Problem (CNLLSP):

\[
\min_x \phi(x) = \frac{1}{2} F(x)^T F(x) = \frac{1}{2} \sum_{k=1}^{l} f_k^2 \quad \rightarrow \text{Objective function}
\]

\[
\text{s. t. } \begin{align*}
    c_i(x) &= 0, \quad i \in M_1, \quad \rightarrow \text{Equality constraints} \\
    c_j(x) &\geq 0, \quad j \in M_2, \quad \rightarrow \text{Inequality constraints}
\end{align*}
\]

where,

\[
F(x) = [f_1(x), \ldots, f_l(x)]^T,
\]

\[
c_i, c_j \text{ and } f_k : (\text{smooth functions}) \mathbb{R}^n \rightarrow \mathbb{R}
\]
Notations:

\[ G(x) = [\cdots \nabla f_k(x) \cdots]_{k=1,\ldots,l} \]  

(Transpose of Jacobian)

\[ \nabla \phi(x) = G(x)F(x), \]  

(Gradient of objective function)

\[ \nabla^2 \phi(x) = G(x)G(x)^T + S(x), \]  

(Hessian of objective function)

where,

\[ S(x) = \sum_{k=1}^{l} f_k(x) \nabla^2 f_k(x). \]

Note:

By knowing \( G(x) \), we can conveniently compute \( \nabla \phi(x) \) and part \( 1 \) of the Hessian \( \nabla^2 \phi(x) \).
Exact penalty method

Aim:  

CNLP  \rightarrow  UCNLP

Continuously differentiable:

• Fletcher (1973): equality constrained.
• Glad and Polak: (1979) equality and inequality constrained.

Nondifferentiable:

• Zangwill: (1967).
• Pietrzykowski: (1969).
Some attractive features:

- They overcome difficulties posed by degeneracy.
- They overcome difficulties posed by inconsistent constraint linearization.
- They are successful in solving certain classes of problems in which standard constraint qualifications do not hold.
- They are used to ensure feasibility of subproblems and improve robustness.

Coleman and Conn (1982): CNLP implementation
Mahdavi-Amiri and Bartels (1989): CNLLS implementation
Pertinent to our scheme:

- **Dennis, Gay and Welsh (1981):** structured DFP update for UNLP and for UNLLS, in particular.
- **Nocedal and Overton (1985):** quasi-Newton update of projected Hessian for CNLP.
- **Mahdavi-Amiri and Bartels (1989):** projected structured update for CNLLS.
- **Dennis and Walker (1981):** local Q-superlinear convergence for the structured PSB and DFP methods, in general, and for UNLLS, in particular.
- **Dennis, Martinez and Tapia (1989):** structured principle, and local Q-superlinear convergence of structured BFGS update for UNLP.
- **Mahdavi-Amiri and Ansari (2012):** Global convergent exact penalty projected structured Hessian updating schemes.
- **Mahdavi-Amiri and Ansari (2013):** Local superlinearly convergent exact penalty projected structured Hessian updating schemes.
\textbf{\( \ell_1 \)-exact penalty function} (violation adds penalty)

\[ \psi(x, \mu) = \mu \phi(x) + \chi(x), \]

where,

\[ \chi(x) = \sum_{i \in M_1} |c_i(x)| - \sum_{j \in M_2} \min(0, c_j(x)). \] (1-norm of the constraint violations)

\textbf{Note:} There exists fixed \( \bar{\mu} > 0 \):

for any \( 0 < \mu < \bar{\mu} \),

local minimizer of \( \psi(x, \mu) \) \quad \Rightarrow \quad \text{local minimizer of (CNLP)}

\textbf{Note:} \( \psi(x, \mu) \) is not differentiable everywhere.
“$\varepsilon$-active” merit function:

$$\psi_\varepsilon(x, \mu) = \mu \phi(x) + \chi_\varepsilon(x),$$

where,

$$\chi_\varepsilon(x) = \sum_{i \in VE(x, \varepsilon)} \text{sgn}(c_i(x))c_i(x) - \sum_{j \in VI(x, \varepsilon)} c_j(x).$$

$VE(x, \varepsilon) = \{i : |c_i(x)| > \varepsilon, i \in M_1 \}$  \hspace{1cm} ($\varepsilon$-violated equalities)

$VI(x, \varepsilon) = \{j : c_j(x) < -\varepsilon, j \in M_2 \}$  \hspace{1cm} ($\varepsilon$-violated inequalities)

Notes:

1) $\psi_\varepsilon(x, \mu)$ is differentiable in a neighborhood of $x$,
2) $\psi(x, \mu) = \psi_0(x, \mu)$. 
More notations:

• active equality constraint indices:

\[ AE(x, \varepsilon) = \{ i : |c_i(x)| \leq \varepsilon, i \in M_1 \} \]

• active inequality constraint indices:

\[ AI(x, \varepsilon) = \{ j : |c_j(x)| \leq \varepsilon, j \in M_2 \} \]

• active constraint indices:

\[ AC(x, \varepsilon) = AE(x, \varepsilon) \cup AI(x, \varepsilon) \]

• active constraints gradient matrix:

\[ A(x) = [\cdots \nabla c_k(x) \cdots]_{k \in AC(x, \varepsilon)} \]
First-order necessary conditions: If $x^*$ is a local minimizer of $\psi$ then there exist “multipliers” $\lambda_k^*$, for $k \in AC(x^*, 0)$, such that

(a) $x^*$ is a stationary point:

$$\nabla \psi_0(x^*, \mu) = \sum_{k \in AC(x^*, 0)} \lambda_k^* \nabla c_k(x^*) = A(x^*)\lambda^*, \quad \lambda_k^* \in null(A^{*T}).$$

(b) $\lambda^*$ is in-kilter:

$$\begin{cases} -1 \leq \lambda_k^* \leq 1, & k \in AE(x^*, 0), \\
0 \leq \lambda_k^* \leq 1, & k \in AI(x^*, 0). \end{cases}$$
Second-order sufficiency conditions: If first-order necessary conditions are satisfied and

\[ \forall z: \begin{cases} 
  z^T \nabla c_i(x^*) = 0, & i \in AE \\
  z^T \nabla c_i(x^*) = 0, & i \in AI, \quad \lambda_j > 0 \\
  z^T \nabla c_i(x^*) \geq 0, & i \in AI, \quad \lambda_j = 0 
\end{cases} \]

\[ z^T \left[ \mu \nabla^2 \phi + \sum_{i \in VE} \text{sgn}(c_i) \nabla^2 c_i - \sum_{j \in VI} \nabla^2 c_i - \sum_{i \in AE} \lambda_i \nabla^2 c_i - \sum_{j \in AI} \lambda_j \nabla^2 c_j \right] z > 0, \]

then \( x^* \) is an isolated local minimizer of \( \psi \).
Using QR decomposition of $A(x)$,

$$A(x) = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Y \quad Z] \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where

- $Y$ : basis for $R(A)$,
- $Z$ : basis for $null(A^T)$,
- $A^T Z = 0$, $Y^T Y$ and $Z^T Z$ as identities,

(projected or reduced gradient) $Z^T \nabla \psi_0 = 0 \implies$ stationarity

**Note:** Computationally, $\|Z^T \nabla \psi_\varepsilon\| \leq \tau$ for a small $\tau > 0$ is accepted.
Regions of interest: \( (\tau > 0, \text{ stationary tolerance parameter}) \)

Multiplier estimates in Region 2:

\[
\lambda = \arg\min_{\lambda} \| A(x) \lambda - \nabla \psi_\varepsilon \|_2,
\]

using QR decomposition of \( A(x) \).
Trust Region and Line Search Methods

Trust region: for smooth functions

- are reliable and robust.
- can be applied to ill-conditioned problems.
- have strong global convergence properties.
- can be used for both convex and non-convex approximate models.
Near a nondifferentiable point the quadratic subproblem cannot approximate \( \psi(x, \mu) \) properly in a large interval (this may happen in any iterate, whenever \( x \) moves to a point where an inactive constraint becomes active). Therefore, the algorithm may have some inadequate iterations.

To overcome this difficulty, we propose a special line search strategy that makes use of the structure of least squares and examines all nondifferentiable points in the trust region along a given descent direction \( h \) for a proper minimizer of \( \psi(x, \mu) \).
Step Directions:

   a) global horizontal step direction: $h_G$.
   b) global vertical step direction: $v_G$.

2) dropping step direction: $d$.

3) Newton step direction: $d_N = h_A + v_A$.
   a) asymptotic horizontal step direction: $h_A$.
   b) vertical step direction: $v_A$.
Use of step direction:

**Region 1:** \( \|Z^T \nabla \psi_\varepsilon \| > \tau \)
- compute \( d_G \);
- perform a line search on \( \psi_0(x, \mu) \) to get \( \alpha > 0 \);
- update \( x \): \( \bar{x} \leftarrow x + \alpha d_G \);

**Region 2:** \( \|Z^T \nabla \psi_\varepsilon \| \leq \tau \)
- compute \( \lambda_k, k \in AC(x, \varepsilon) \);
- if \( \lambda_k \) is out-of-kilter for any \( k \in AC(x, \varepsilon) \) then
  - compute dropping step direction \( d \);
  - perform a line search on \( \psi_0(x, \mu) \) to get \( \alpha > 0 \);
  - update \( x \): \( \bar{x} \leftarrow x + \alpha d \);
- else (the \( \lambda_k \) are in-kilter)
  - compute Newton step direction \( d_N \);
  - update \( x \): \( \bar{x} \leftarrow x + d_N \) (no line search).
Trust Region Projected Quadratic Subproblems
Computing horizontal steps: (global and local)

Define

\[ Q_z = Z^T G G^T Z \]

\[ H_z = \mu Q_z + \mu B_z + D_z \]

(approximation of projected Hessian)

\[ \approx B_z \]

\[ \approx B_z \]

\[ \approx \mu Q_z + \mu Z^T [S(x) - \sum_{k \in AC(x, \varepsilon)} \frac{\lambda_k}{\mu} \nabla^2 c_k ] Z \]

\[ + Z^T \nabla^2 \chi_\varepsilon(x) Z \]

\[ \approx D_z \]

\[ \min_w (Z^T \nabla \psi_\varepsilon)^T w + w^T H_z w \]

s.t. \[ \|w\| \leq \beta \Delta \]

(\( \Delta \) is trust region radius and \( \beta \in (0, 1) \) is a prescribed constant)
Region 1 (global)  \[\text{Compute } \mathbf{w} : \text{use } \lambda_k = 0, \; k \in AC(x, \varepsilon), \; \text{and set } h_G \leftarrow Z \mathbf{w}.\]

Region 2 (local)  \[\text{Compute } \mathbf{w} : \text{use estimated } \lambda, \; \text{and set } h_A \leftarrow Z \mathbf{w}.\]

Solution of this subproblem is shown by \(\mathbf{w}(\mu)\).

Computing vertical steps: enforce activity

A Newton iterate

\[C_{AC}(x, \varepsilon)(x + h + \nu) = 0 \quad \Rightarrow \quad A^T(x)\nu = -C_{AC}(x, \varepsilon)(x + h)\]

Notes:

- \(h\) stands for \(h_G\) or \(h_A\).
- if necessary, we cut back \(\nu_G\) to ensure that \(\|\alpha d_G\| \leq \Delta\).
Computing dropping step:

1) Find an index $k$ corresponding to most out-of-kilter $\lambda_k$,
2) Solve:

$$A(x)^T d = \sigma_k e_k$$

where, $e_k$ is the $k$th unit vector and

$$\sigma_k = \begin{cases} 
-1, & \text{if } \lambda_k > +1, \\
1, & \text{if } \lambda_k < 0, \quad k \in M_1, \\
+1, & \text{if } \lambda_k < -1, \quad k \in M_2.
\end{cases}$$

**Note:** $d$

- gives a local, first-order decrease in $\psi_\varepsilon(x, \mu)$, and
- moves the $c_k$ away from zero up to the first-order.
Quadratic subproblems:

1) When $x$ is $\varepsilon$ – feasible:

$$\min_w (Z^T \nabla \phi)^T w + w^T (Q_z + B_z) w$$

s. t. $\|w\| \leq \beta \Delta,$

(local quadratic subproblem)

which is a penalty parameter free quadratic subproblem (thus, no computational difficulties in the local phase).

2) When $\mu = 0$:

$$\min_w (Z^T \nabla \chi_\varepsilon)^T w + w^T D_z w$$

s. t. $\|w\| \leq \beta \Delta,$

(auxiliary quadratic subproblem)
Note: The last auxiliary quadratic subproblem

- determines the optimal improvement in quadratic feasibility achievable inside the trust region.
- helps to balance the progress towards both optimality and feasibility in the exact penalty approach.

Solution of auxiliary subproblem is shown by $w(0)$.

Byrd, Nocedal and Waltz (2008)  Linear subproblem

Here  Quadratic subproblem
Penalty parameter updating strategy

Essential for practical success (robustness) in the context of penalty methods:

- too small a $\mu$ → inefficient behavior and slow convergence
- a $\mu$ too large → possible convergence to an infeasible stationary point.

Note:
We allow the possibility of both increasing (at times with a safeguard) or decreasing value of $\mu$. 
a weak progress towards feasibility by $w(\mu)$ in comparison with the optimal achievable reduction predicted by $w(0)$

a very good progress towards feasibility by $w(\mu)$, but a poor improvement in optimality

**Note:** $\mu$ is updated

- only in Region 1 (global phase), and
- when, $VE(x,\epsilon) \cup VI(x,\epsilon) \neq \emptyset$ (that is, $x$ is not $\epsilon$-feasible).
Notations:

— the predicted decrease in the objective function \( \phi(x) \):
\[
Pred_\phi(w) = - \left( (Z^T \nabla \phi)^T w + \frac{1}{2} w^T (Q_z + B_z) w \right),
\]

— the predicted improvement in measure of infeasibility:
\[
Pred_\chi(w) = - \left( (Z^T \nabla \chi_\varepsilon)^T w + \frac{1}{2} w^T D_z w \right),
\]

— the predicted decrease in \( \psi \):
\[
Pred_\psi(w) = \mu Pred_\phi(w) + Pred_\chi(w).
\]

\[
Pred_\chi(w(0)) \geq 0 \quad \text{(best predicted improvement in measure of infeasibility achievable inside the trust region)}
\]

\[
Pred_\chi(w(\mu)) \quad \text{(predicted improvement in measure of infeasibility \( w(\mu) \))}
\]
Reduce $\mu$ : (e.g., $\mu \leftarrow \frac{\mu}{8}$)

$\mu$ is reduced (if necessary) to satisfy followings:

1) predicted reduction in infeasibility of constraints is a fraction of the best possible predicted reduction; that is,

$$\text{Pred}_\chi(w(\mu)) \geq \varepsilon_1 \text{Pred}_\chi(w(0)) \quad \text{(e.g., } \varepsilon_1 = 0.15\text{)}$$

2) penalty function provides a sufficient improvement of feasibility; that is,

$$\text{Pred}_\psi(w(\mu)) \geq \varepsilon_2 \text{Pred}_\chi(w(\mu)) \quad \text{(e.g., } \varepsilon_2 = 0.5\text{)}$$
Increase $\mu : (\text{e.g., } \mu \leftarrow 8 \times \mu)$

In practice (on a few problems), to ensure satisfaction of (1) and (2) above, $\mu$ may become too small too quickly.

Conditions for such an occurrence:

- cases (1) and (2) above are satisfied ($\mu$ does not decrease),
- there is a very good improvement in feasibility; that is,
  \[ \text{Pred}_\chi(w(\mu)) \geq \eta_1 \text{Pred}_\chi(w(0)) \]  
  (e.g., $\eta_1 = 0.95$)
- objective function is not optimal and could be improved; that is,
  \[ \text{Pred}_\phi(w(\mu)) > 0 \]
- there is a poor improvement in optimality
  \[ \text{Pred}_\phi(w(\mu)) < \varepsilon_3 \text{Pred}_\chi(w(\mu)) \]  
  (e.g., $\varepsilon_3 = 0.1$)
Therefore,

while

\begin{itemize}
\item \[ \text{Pred}_\chi(w(\mu)) \geq \eta_2 \text{Pred}_\chi(w(0)) \]  
  \text{ (e.g., } \eta_2 = 0.9) \\
\item \[ \text{Pred}_\phi(w(\mu)) < \varepsilon_3 \text{Pred}_\chi(w(\mu)) \]  
  \text{ (e.g., } \varepsilon_3 = 0.1) \\
\end{itemize}

increase \( \mu \)

**Note:**

If the new \( w(\mu) \) does not satisfy (1) and (2), then \( \mu \) is set back to its previous value.

**safeguard**

(to avoid attaining too large a \( \mu \) too soon)

**increase \( \mu \) a limited number of times**

at most 2 times in one iteration and at most 6 times in all the iterations
Step acceptance

For a trial point $x_T$, define

$$A_{red} = \psi(x, \mu) - \psi(x_T, \mu)$$  \hspace{1cm} \text{(actual reduction)}$$

Also, for a step direction $h = \alpha v_G, v_A$ or $\alpha d$, define

$$\text{Pred}_l(h) = -\mu(GF)^T h + \sum_{i \in M_1} (|c_i| - |c_i + \nabla c_i^T h|)$$
$$- \sum_{j \in M_2} (\min(0, c_j) - \min(0, c_j + \nabla c_j^T h))$$
Predicted reduction:

- global step: \( \alpha d_G = \alpha Z w + \alpha v_G \),
  
  \[ \text{Pred} = \text{Pred}_\psi(\alpha w) + \text{Pred}_I(\alpha v_G) \]

- dropping step: \( \alpha d \),
  
  \[ \text{Pred} = \text{Pred}_I(\alpha d) \]

- Newton step: \( d_N = Z w + v_A \),
  
  \[ \text{Pred} = \text{Pred}_\psi(w) + \text{Pred}_I(v_A) \]

A trial point \( x_T \) is accepted if \( A_{red} > 0 \) and

\[
\rho = \frac{A_{red}}{\text{Pred}} > 10^{-6}.
\]
Updating the trust region radius:

For current step $h$:

$$
\Delta^+ = \begin{cases}
\max(\Delta, 3\|h\|), & \rho \geq 1.1, \\
\max(\Delta, 5\|h\|), & 0.9 \leq \rho < 1.1, \\
\max(\Delta, 2\|h\|), & 0.3 \leq \rho < 0.9, \\
\Delta, & 10^{-6} \leq \rho < 0.3, \\
\min(0.25\Delta, 0.25\|h\|), & \rho < 10^{-6}.
\end{cases}
$$

Note:

Locally (in Region 2):

- $\rho \to 1$.
- there exists a fixed $\Delta > 0$: $\Delta > \Delta$. 
Projected structured Hessian approximation

\[ Z^T (\nabla^2 \psi_\epsilon) - \sum_{k \in AC(x, \epsilon)} \lambda_k \nabla^2 c_k )Z \]

\[ \approx B_Z \]

\[ = \mu Q_Z + \mu Z^T [S(x) - \sum_{k \in AC(x, \epsilon)} \frac{\lambda_k}{\mu} \nabla^2 c_k ]Z \]

\[ + Z^T \nabla^2 \chi_\epsilon(x)Z \]

\[ \approx D_Z \]

\[ \approx \mu Q_Z + \mu B_Z + D_Z = H_Z. \]
Notations:

- violated equality constraints gradient matrix:
  \[ E(x) = [\cdots \nabla c_i(x) \cdots]_{i \in \mathcal{E}(x, \varepsilon)} \]

- violated inequality constraints gradient matrix:
  \[ I(x) = [\cdots \nabla c_j(x) \cdots]_{j \in \mathcal{I}(x, \varepsilon)} \]

- violated equality vector of signs:
  \[ \pi(x) = [\cdots \text{sgn} \left( c_j(x) \right) \cdots]^T_{j \in \mathcal{I}(x, \varepsilon)} \]

Let

\[ e = [1 \cdots 1 \cdots 1]^T \]
Assumption: \((x = x^k, \quad \bar{x} = x^{k+1})\)

\[AC(x^k, \varepsilon) = AC(x^*, 0)\]  Region 2 (locally)

Secant condition: Mahdavi-Amiri and Bartels (1989)

\[
(\mu \bar{Q}_z + \mu \bar{B}_z + \bar{D}_z) s = \mu \bar{Q}_z s + \mu \bar{B}_z s + \bar{D}_z s
\]

\[
= \mu \bar{Q}_z s + \mu y_1 + y_2,
\]

where,

\[
s = \bar{Z}^T (\bar{x} - x),
\]

\[
y_1 = \bar{Z}^T \left[ (\bar{G} - G) \bar{F} + A \hat{\lambda} \right], \quad \hat{\lambda} = \frac{\lambda}{\mu},
\]

\[
y_2 = \bar{Z}^T \left[ (\bar{E} - E) \bar{\pi} - (\bar{I} - I) e \right].
\]

Set

\[ u = \bar{Q}_z s + y_1, \]
\[ X = \bar{Q}_z + B_z. \]

**BFGS:**

\[ \bar{B}_z = B_z + \frac{uu^T}{u^T s} - \frac{Xs(Xs)^T}{s^T Xs} \]

\[ \bar{D}_z = D_z + \frac{y_2 y_2^T}{y_2^T s} - \frac{D_z s(D_z s)^T}{s^T D_z s} \]
DFP:

\[
\bar{B}_z = B_z + \frac{(u - Xs)u^T + u(u - Xs)^T}{s^T u} - \frac{s^T (u - Xs)}{(s^T u)^2} uu^T
\]

\[
\bar{D}_z = D_z + \frac{(y_2 - D_zs)y_2^T + y_2(y_2 - D_zs)^T}{s^T y_2} - \frac{s^T (y_2 - D_zs)}{(s^T y_2)^2} y_2 y_2^T
\]
Special nonsmooth Line Search Strategy
(for global or dropping direction)

\( h \) : a descent search direction.
\( \alpha > 0 \) : step length along \( h \).
\( \psi(\alpha) = \psi(x + \alpha h, \mu) \)

Breakpoint : A point where derivative of \( \psi \) does not exist (when a new constraint becomes active).

**Note:**
A minimizer exists and a minimum occurs either at a breakpoint, or at a point where \( \psi(\alpha) \) is zero.

Motivation: Consider the \( c_i \) and the \( f_k \) to be linear. Approximate \( \psi \) using linear approximations of the \( c_i \) and the \( f_k \), denoted by \( \tilde{\psi} \). Let
\[
\tilde{\psi}(\alpha) = \tilde{\psi}(x + \alpha h, \mu).
\]
\[ \tilde{\psi}' = \text{Derivative of } \tilde{\psi} \]

\[ \tilde{\psi}'_+ \quad \text{and} \quad \tilde{\psi}'_- = \text{Right and left derivatives of } \tilde{\psi} \]

Similar notations are used for \( c_i \) (\( \tilde{c}_i \), as a linear approximation of \( c_i \))

**Note:**

- Breakpoints occur at zeros of the \( c_i \).
- If \( \tilde{c}_i(x)\tilde{c}'_i(x) < 0 \) then (by linearity of \( \tilde{c}_i \)) \( \tilde{c}_i(x + \alpha h) = 0 \) gives a breakpoint.
- Breakpoints are computed approximately using linear approximations.
Strategy:

Examine breakpoints sequentially, and find a minimum at one of the following situations:

(i) a breakpoint $\gamma_i$ (Figure 1), where

\[ \tilde{\psi}'_+ (\gamma_i) > 0 \text{ and } \tilde{\psi}'_- (\gamma_i) \leq 0 \]
(ii) or at a point, where \( \tilde{\psi}'(\alpha) = 0 \).

Note:

By the used linearizations, all derivatives are computed without any new objective or constraint function evaluation.
Computational considerations

If something goes wrong computationally, then

In region 1 (global $||Z^T \nabla \psi_\varepsilon|| > \tau$)

- $AC(x, \varepsilon) \neq AC(x, 0)$ (\varepsilon \text{ large})
  
  $AC(x, \varepsilon) = AC(x, 0)$
  or $\varepsilon$ too small

  $\varepsilon \leftarrow \frac{\varepsilon}{2}$ (to change $AC(x, \varepsilon)$)

  Failure

In region 2 (asymptotic $||Z^T \nabla \psi_\varepsilon|| \leq \tau$)

- $\tau$ too small
  
  Failure

- otherwise

  $\tau \leftarrow \frac{\tau}{2}$ (to have $||Z^T \nabla \psi_\varepsilon|| > \tau$, and transfer to global phase)
Computational results

32 CNLLS problems from Hock and Schittkowski (1981)

- number of variables: \(2 \leq |n| \leq 5\)
- number of constraints: \(1 \leq |M_1| + |M_2| \leq 13\)

7 CNLLS randomly generated test problems from Bartels and Mahdavi-Amiri (1986)

- number of variables: \(5 \leq |n| \leq 20\)
- number of constraints: \(5 \leq |M_1| + |M_2| \leq 20\)

Objective function and constraints: all nonlinear,
Lagrangian Hessian at solution: positive definite for problems 1, 4 and 6, and indefinite for problems 2, 3, 5 and 7.
6 CNLLS problems from Biegler, Nocedal, Schmid and Ternet: 2000

number of variables: \( 80 \leq |n| \leq 300 \)
number of constraints: \( 40 \leq |M_1| + |M_2| \leq 299 \)

objective function and constraints: all nonlinear.

12 CNLLS problems from Luksan and Vlcek (1986)

number of variables: \( 297 \leq |n| \leq 300 \)
number of constraints: \( 2 \leq |M_1| + |M_2| \leq 300 \)

objective function and constraints: all nonlinear.
Comparisons:

- 6 methods tested by Hock and Schittkowski (1981).
- Two algorithms provided by ‘fmincon’ in MATLAB.

Features of our code:

- Language: C++,
- Compiler: Microsoft visual C++ 6.0,
- Linear algebra: Simple C++ Numerical Toolkit,
- Computer: x86-based PC, 1667Mhz processor.
**Performance profile:**

Means of comparison: Number of objective function evaluations.

Define

\[ \rho_{p,i} = \frac{FE_{p,i}}{\min_{1 \leq j \leq n_a} \{FE_{p,j}\}} \]

*FE*<sub>*p,i*</sub>: Number of objective function evaluations in algorithm *i* for problem *p*.

*n<sub>a</sub>*: Number of algorithms.

Performance function for algorithm *i*:

\[ p_i(\tau) = \frac{|N_i(\tau)|}{n_p}, \quad \tau \geq 1, \]

where *n<sub>p</sub>* is the total number of problems and

\[ N_i(\tau) = \{j \mid 1 \leq j \leq n_p, \rho_{j,i} \leq \tau\} \]
Legends:

**Tr-LS:** our combined trust region-line search method.
**B:** method of Bidabadi and Mahdavi-Amiri.
**H:** best methods tested by Hock and Schittkowski.
**MA:** Method of Mahdavi-Amiri and Bartels.
**D:** Interior/Direct algorithm of KNITRO.
**CG:** Interior/CG algorithm of KNITRO.
**AS:** Active-set algorithm of KNITRO.
**FM1:** Active-set algorithm provided by ‘fmincon’ in MATLAB.
**FM2:** Interior point algorithm provided by ‘fmincon’ in MATLAB.
Conclusions

- two-step superlinear rate of convergence
- effective new scheme for approximating and updating projected structured Hessian
- practical adaptive scheme for the penalty parameter updates
- efficient combined trust region-line search strategy help in speeding up the global iterations
- Employment of an effective trust region radius adjustment
- competitive as compared to other methods
References


Thank you for your attention!