Lecture 24: Trust-Region Methods

Yudong Chen

So far, we have been looking at methods of the form

$$x_{k+1} = x_k - \alpha_k \underbrace{B_k^{-1} \nabla f(x_k)}_{-p_k},$$

where $B_k \succ 0$. Examples:

- $B_k = I$: steepest descent;
- $B_k = \nabla^2 f(x_k)$: (damped) Newton's method
- B_k approximates $\nabla^2 f(x_k)$: quasi-Newton method.

In all these methods, we first determine the search direction p_k, and then choose the stepsize α_k.

In Trust Region (TR) methods, we first determine the size of the step, then the direction.

1 Trust region method

We want to compute the step p_k that gives the next iterate $x_{k+1} = x_k + p_k$.

Let $B_k \in \mathbb{R}^{d \times d}$ be given. Typically, B_k equals $\nabla^2 f(x_k)$ or an approximation thereof obtained by a Quasi-Newton method (say SR1). We use B_k to construct the following quadratic approximate model of f around x_k:

$$m_k(p) := f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} p^\top B_k p.$$

Basic idea of TR: to compute the direction p_k, we minimize $m_k(p)$ over a region (a ball centered at x_k) within which we trust that m_k is a good approximation of f.

Note that we do *not* require $B_k \succ 0$. In particular, we can use an indefinite $\nabla^2 f(x_k)$ without modification.

Formally, the (exact) TR direction is given by solving the following *constrained* subproblem:

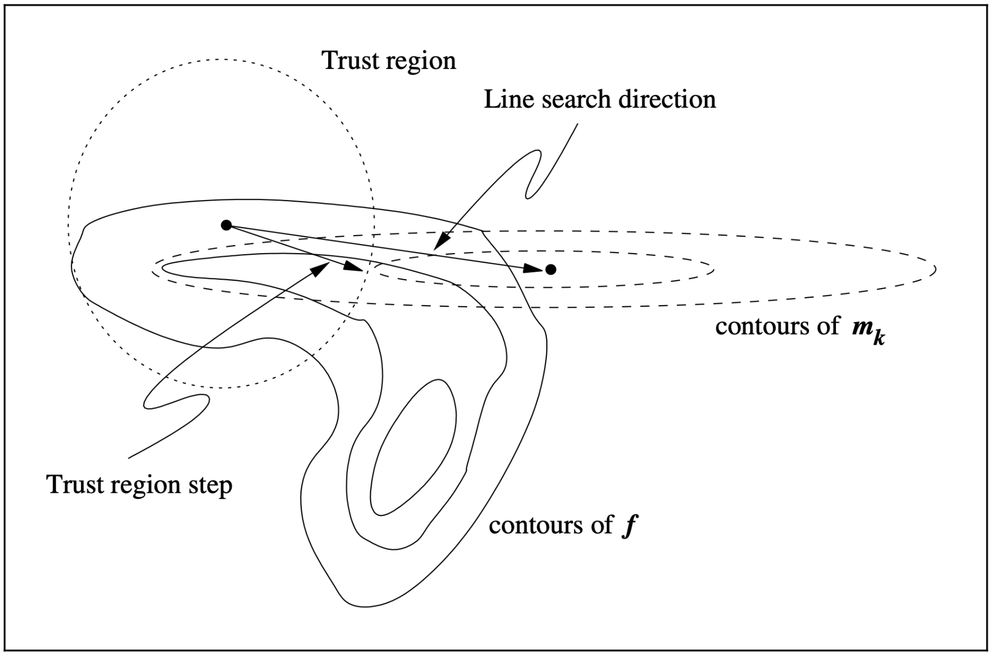
$$p_k := \underset{p \in \mathbb{R}^d : \|p\| \leq \Delta_k}{\operatorname{argmin}} \ m_k(p),$$

where Δ_k is the radius of the trust region.

Example 1. Suppose $f(x) = x_1^2 - x_2^2$, which is a nonconvex quadratic function. The quadratic model is the function itself: $m_k(p) = f(x_k + p)$. Suppose we are current at $x_k = \mathbf{0}$. Then $\nabla f(x_k) = 0$, so gradient descent (GD) and Newton's method will stay at $\mathbf{0}$ (a stationary point). In contrast, TR method will take the step

$$\begin{aligned} p_k &= \underset{p : \|p\| \leq \Delta_k}{\operatorname{argmin}} \, m_k(p) \\ &= \underset{p : p_1^2 + p_2^2 \leq \Delta_k^2}{\operatorname{argmin}} \left\{ (0 + p_1)^2 - (0 + p_2)^2 \right\} = (0, \Delta_k) \text{ or } (0, -\Delta_k). \end{aligned}$$

For TR applied to more general functions, see the illustration below from Nocedal-Wright:



To completely specify the TR method, we need to decide:

1. how to choose the radius Δ_k,
2. how and to what accuracy to solve the subproblem $\min_{p\in\mathbb{R}^d:\|p\|\le\Delta_k} m_k(p)$.

2 Choosing the radius Δ_k

Define

$$\rho_k := \frac{\overbrace{f(x_k) - f(x_k+p_k)}^{\text{actual reduction}}}{\underbrace{m_k(0) - m_k(p_k)}_{\text{predicted reduction,}\ge 0}}.$$

The ratio ρ_k tells us whether we are making progress, and if so, how much.

General idea:

1. If ρ_k is positive and large, then f and m_k agree well within the trust region $\|p\| \le \Delta_k$. We can try increasing Δ_k in next iteration.
2. If ρ_k is small or negative, we should consider decreasing Δ_k (shrink the trust region).
   (a) In particular, if ρ_k is negative, then f has increased. We should reject the step p_k and stay at x_k.

The following algorithm describes the process.

Algorithm 1 Trust Region

Input: $\hat{\Delta} > 0$ (largest radius), $\Delta_0 \in (0, \hat{\Delta})$ (initial radius), $\eta \in [0, 1/4)$ (acceptance threshold)
for $k = 0, 1, 2, \ldots$
  $p_k = \operatorname{argmin}_{p:\|p\| \le \Delta_k} m_k(p)$ (or compute an approximate minimizer)
  $\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$
  if $\rho_k < \frac{1}{4}$: `\\ insufficient progress`

$$\Delta_{k+1} = \frac{1}{4}\Delta_k \quad \texttt{\\ reduce radius}$$

  else:
    if $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$: `\\ sufficient progress, active trust region`

$$\Delta_{k+1} = \min\left\{2\Delta_k, \hat{\Delta}\right\} \quad \texttt{\\ increase radius}$$

    else: `\\ sufficient progress, inactive trust region`

$$\Delta_{k+1} = \Delta \quad \texttt{\\ keep radius}$$

  if $\rho_k > \eta$: `\\ sufficient progress`

$$x_{k+1} = x_k + p_k \quad \texttt{\\ accept step}$$

  else: `\\ insufficient progress`

$$x_{k+1} = x_k \quad \texttt{\\ reject step}$$

end for

3 Exact minimization of m_k

In each iteration of Algorithm 1, we need to solve the TR sub-problem

$$\min_{p:\|p\| \le \Delta_k} m_k(p) := f_k + g_k^\top p + \frac{1}{2} p^\top B_k p, \tag{P_{m_k}}$$

where we introduce the shorthands $f_k := f(x_k)$ and $g_k := \nabla f(x_k)$. This is a quadratic minimization problem over an Eucludean ball.

The theorem below characterizes the exact minimizer $p_k^* = \operatorname{argmin}_{p:\|p\| \le \Delta_k} m_k(p)$.

Theorem 1 (Characterizing the solution to (P_{m_k}))**.** *The vector $p^* \in \mathbb{R}^d$ is a global solution to the problem (P_{m_k}) <u>if and only if</u> p^* is feasible (i.e., $\|p^*\| \le \Delta_k$) and there exists $\lambda \ge 0$ such that the following condition holds:*

1. $(B_k + \lambda I)p^* = -g_k$,
2. $\lambda(\Delta_k - \|p^*\|) = 0$ *(complementary slackness),*
3. $B_k + \lambda I \succcurlyeq 0$.

The complete proof of Theorem 1 makes use of Lagrangian multipliers, which we will not delve into.

Exercise 1. Prove the necessity of part 1 above using the first-order optimality condition for constrained optimization (Lecture 14, Theorem 1).

Some observations about Theorem 1:

- If $\|p^*\| < \Delta_k$, then the trust region constraint is inactive/irrelevant. In this case, part 2 implies $\lambda = 0$, part 1 implies $B_k p^* = -g_k$, and part 3 implies $B_k \succcurlyeq 0$. See p^{*3} in the figure below.
- In the other case where $\|p^*\| = \Delta_k$, we have $\lambda > 0$. Part 1 of Theorem 1 gives:

$$\lambda p^* = -B_k p^* - g_k = -\nabla m_k(p^*),$$

hence p^* is parallel to $-\nabla m_k(p^*)$ and thus normal to contours of m_k; equivalently, $-\nabla m_k(p^*) \in N_{\mathcal{X}}(p^*)$, where $\mathcal{X} = \{p : \|p\| \leq \Delta_k\}$. See p^{*1} and p^{*2} in the figure below.

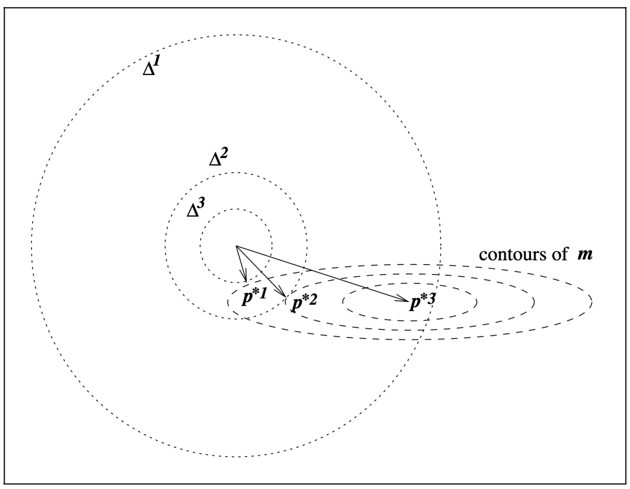


Figure 4.2 Solution of trust-region subproblem for different radii Δ^1, Δ^2, Δ^3.

To find the exact minimizer p_k^*, one may use an iterative method to search for the λ that satisfies the conditions in Theorem 1.

4 Approximate methods for minimizing m_k

Solving the TR subproblem (P_{m_k}) exactly is usually unnecessary. After all, m_k is only a local approximation of actual objective function f.

4.1 Algorithms based on the Cauchy point

The *Cauchy point* p_k^C is defined by the following procedure.

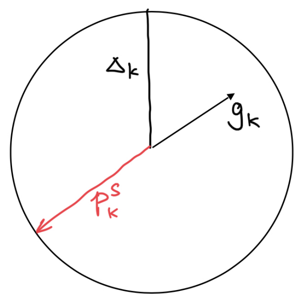
Algorithm 2 Cauchy Point Calculation

Compute

$$p_k^S = \operatorname*{argmin}_{p:\|p\|\le\Delta_k} \left\{ f_k + g_k^\top p \right\},$$
$$\tau_k = \operatorname*{argmin}_{\tau\ge 0: \|\tau p_k^S\|\le\Delta_k} m_k(\tau p_k^S).$$

Return $p_k^C = \tau_k p_k^S$

Note that p_k^S is the minimizer of the *linear* model $f_k + g_k^\top p$ within the trust region; that is, p_k^S solves the linear version of the TR subproblem (P_{m_k}). The scalar τ_k is obtained by minimizing the *quadratic* model m_k along the direction of p_k^S.



Linear version, ignoring the quadratic part

The Cauchy point can be easily computed.

Lemma 1. *The Cauchy point $p_k^C = \tau_k p_k^S$ is given explicitly by*

$$p_k^S = -\frac{\Delta_k}{\|g_k\|} g_k, \qquad \tau_k = \begin{cases} 1 & g_k^\top B_k g_k \le 0, \\ \min\left\{1, \frac{\|g_k\|^3}{\Delta_k g_k^\top B_k g_k}\right\}, & g_k^\top B_k g_k > 0. \end{cases}$$

Proof. It is easy to see that

$$p_k^S = -\frac{\Delta_k}{\|g_k\|} g_k,$$

which is in the direction of the negative gradient. Hence

$$\begin{aligned} m_k(\tau p_k^S) &= f_k + \tau \left\langle g_k, -\frac{\Delta_k}{\|g_k\|} g_k \right\rangle + \frac{\tau^2}{2} \left(\frac{\Delta_k}{\|g_k\|} g_k \right)^\top B_k \left(\frac{\Delta_k}{\|g_k\|} g_k \right) \\ &= f_k \underbrace{-\tau \Delta_k \|g_k\|}_{\le 0} + \frac{\tau^2}{2} \frac{\Delta_k^2}{\|g_k\|^2} g_k^\top B_k g_k. \end{aligned}$$

The RHS is a one-dimensional quadratic function of τ. Since $\|p_k^S\| = \Delta_k$, the trust-region constraint $\|\tau p_k^S\| \le \Delta_k$ is equivalent to $0 \le \tau \le 1$.

Case 1: $g_k^\top B_k g_k \le 0$. Then $m_k(\tau p_k^S)$ is decreasing in τ, so the minimizer is on the boundary of the trust region, that is, $\tau_k = \frac{\Delta_k}{\|p_k^S\|} = 1$.

Case 2: $g_k^\top B_k g_k > 0$. Then $m_k(\tau p_k^S)$ is a convex quadratic in τ, hence τ_k is either the unconstrained minimizer of $m_k\left(\tau p_k^S\right)$, or 1 (on the boundary), whichever is smaller.

Combining Case 1 + Case 2, we conclude that

$$\tau_k = \begin{cases} 1 & g_k^\top B_k g_k \le 0, \\ \min\left\{1, \frac{\|g_k\|^3}{\Delta_k g_k^\top B_k g_k}\right\}, & g_k^\top B_k g_k > 0. \end{cases}$$

□

4.2 Improving the Cauchy point

If we simply using the Cauchy point, $p_k = p_k^C$, then the TR method will move in the direction $-g_k = -\nabla f(x_k)$ and hence converge no faster than gradient descent (with exact line search).

The Cauchy point only uses the matrix B_k to determine the length of the step but not the direction. To achieve faster convergence, we need to make more substantial use of B_k.

Two ways to improve upon the Cauchy point are

- The dogleg method;
- Two-dimensional subspace minimization.

We will not go into the details. Please refer to the appendix (optional).

5 Convergence analysis of trust-region methods

In this section, we state without proof several convergence results for TR methods.

5.1 Global convergence to a stationary point

The Cauchy point p_k^C can be used as a benchmark. To assess the quality of another approximate solution p_k to the TR subproblem (P_{m_k}), we compare it with p_k^C. One can show that for a TR method to converge globally, it is sufficient if p_k reduces m_k by at least some constant times the decrease from the Cauchy point, i.e.,

$$m_k(p_k) - m_k(0) \le c\left(m_k\left(p_k^C\right) - m_k(0)\right). \tag{1}$$

Note that (1) is satisfied by the exact minimizer of the TR subproblem (P_{m_k}), the dogleg method and the 2D subspace minimization method with $c = 1$.

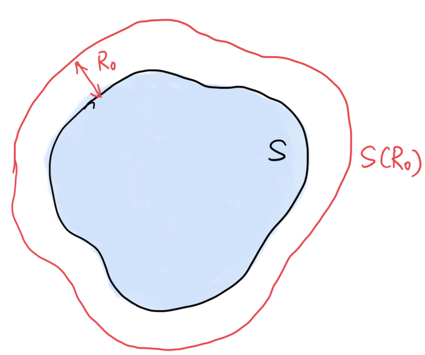
To state the formal theorem, we need some definitions and assumptions.

Consider the level set

$$S := \left\{x \in \mathbb{R}^d \mid f(x) \le f(x_0)\right\}.$$

Define an open neighborhood of S by

$$S(R_0) := \{x \mid \|x - y\| < R_0 \text{ for some } y \in S\}.$$



Assumptions:

1. $\forall k : \|B_k\|_2 \leq \beta < \infty$.
2. f is bounded below on S.
3. f is smooth (i.e., has Lipschitz continuous gradient) on $S(R_0)$ for some $R_0 > 0$.

Theorem 2 (Theorems 4.4 and 4.5 in Nocedal-Wright)**.** *Let $\eta = 0$ in Algorithm 1. Suppose that the assumptions stated above are satisfied, and the step p_k satisfies $\|p_k\| \leq \Delta_k$ and the comparison inequality (1) for all k. Then*

1. *p_k has sufficient progress:*

$$m_k(p_k) - m_k(0) \leq -\frac{c}{2} \|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}, \quad \forall k. \tag{2}$$

2. *The gradient sequence $\{g_k\}$ has a limit point at zero:*

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

Part 1 of Theorem 3 can be viewed as a "descent lemma" for TR methods and implies the convergence property in Part 2. This is similar to how the convergence of gradient descent follows from its descent lemma.

Theorem 3 assumes that $\eta = 0$ is used in the Algorithm 1; that is, we always accept the step if there is any progress. If we use $\eta > 0$ (rejects steps with low progress), we have the stronger result that $g_k \to 0$. See Theorem 4.6 in Nocedal-Wright.

5.2 Local convergence of TR-Newton method

The results discussed so far hold for a general B_k. We now specialize to TR methods that use the exact Hessian $B_k = \nabla^2 f(x_k)$ for all sufficiently large k. (We refer to these methods as TR-Newton.) In this case, we expect that the TR bound $\|p_k\| \leq \Delta_k$ becomes inactive near the minimizer of f and thus an approximate solution p_k to the TR subproblem (P_{m_k}) becomes similar to the Newton step $p_k^{\mathrm{N}} := -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$.

The theorem below establishes superlinear local convergence of TR-Newton.

Theorem 3 (Theorem 4.9 in Nocedal-Wright)**.** *Let f be twice continuously differentiable (with β_1-Lipschitz gradients and L-Lipschitz Hessians) in a neighborhood of a local minimizer x^* satisfying $\nabla f(x^*) = 0, \nabla^2 f(x^*) \succ 0$. Suppose that*

1. *$\{x_k\}$ converges to x^*;*
2. *for all k sufficiently large, the TR algorithm with $B_k = \nabla^2 f(x_k)$ chooses p_k such that*
   - *(a) the sufficient progress condition (2) holds, and*
   - *(b) p_k is asymptotically similar to $p_k^N = -\nabla^2 f(x_k)^{-1} g_k$ whenever $\|p_k^N\| \leq \frac{\Delta_k}{2}$, i.e.,*

$$\left\|p_k - p_k^N\right\| = o(\left\|p_k^N\right\|). \tag{3}$$

Then the TR bound becomes inactive for all sufficiently large k and the convergence of $\{x_k\}$ to x^* is superlinear.

Theorem 3 is proved by invoking the generic quasi-Newton result in Lecture 21, Theorem 2, which states that the condition (3) implies superlienar convergence.

Appendices

All the materials in this appendix are optional.

A The dogleg method

The Dogleg method is used only when $B_k \succ 0$.

Intuition: consider two extremes.

- If Δ_k is small, then $\Delta_k^2 \ll \Delta_k$. Hence for $\|p\| \le \Delta_k$, the quadratic model is approximately linear: $m_k(p) \approx f_k + g_k^\top p$. In this case, it is approximately optimal to use the Cauchy point, i.e., $p_k^* \approx p_k^{\mathrm{C}}$.
- If Δ_k is large, then the constraint $\|p_k\| \le \Delta_k$ becomes irrelevant. In this case, p_k^* approximately equals the unconstrained minimizer of m_k, i.e., $p_k^* \approx -B_k^{-1} p_k =: p_k^{\mathrm{B}}$.

The dogleg method interpolates between these two extremes.

Formally, define

$$p_k^{\mathrm{U}} := -\frac{g_k^\top g_k}{g_k^\top B_k g_k} g_k = \text{(unconstrained) GD step with exact line search}$$
$$p_k^{\mathrm{B}} := -B_k^{-1} g_k = \text{unconstrained minimizer of } m_k$$

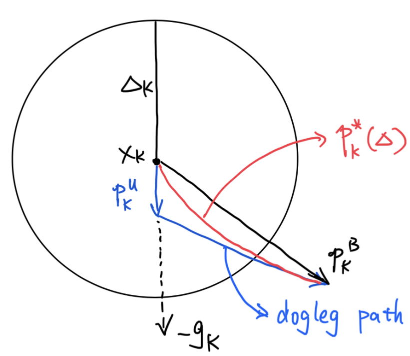
Consider the "dogleg path" defined below:

$$\tilde{p}_k(\tau) := \begin{cases} \tau p_k^{\mathrm{U}}, & 0 \le \tau \le 1, \\ p_k^{\mathrm{U}} + (\tau - 1)(p_k^{\mathrm{B}} - p_k^{\mathrm{U}}), & 1 \le \tau \le 2. \end{cases}$$

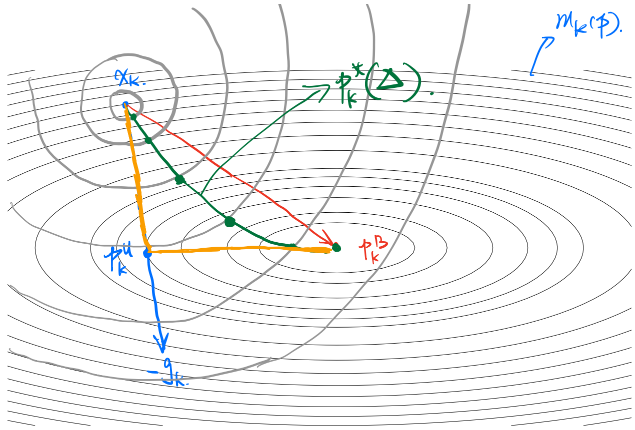
Note that $\tilde{p}_k(\tau)$ consists of two line segments and is an approximation of the optimal path $p_k^*(\Delta)$. The dogleg step is given by constrained minimizer over the path $\tilde{p}(\tau)$, i.e.,

$$p_k^{\mathrm{D}} := \min_{\substack{0 \le \tau \le 2 \\ \|\tilde{p}_k(\tau)\| \le \Delta}} m_k\left(\tilde{p}_k(\tau)\right).$$

Illustration:



Another illustration:



Thanks to the following lemma, it is easy to compute the minimizer p_k^{D} along the dogleg path.

Lemma 2 (Lemma 4.2 in Nocedal-Wright)**.** *Let B_k be positive definite. Then*

(i) $\|\tilde{p}_k(\tau)\|$ *is an increasing function of* τ*;*

(ii) $m_k(\tilde{p}_k(\tau))$ *is a decreasing function of* τ*.*

Consequently:

- If $\|p^{\mathrm{B}}\| < \Delta$, then the dogleg path does not intersect the TR boundary $\|p\| = \Delta$. Since m_k is decreasing in τ,we have $p_k^{\mathrm{D}} = \tilde{p}_k(2) = p^{\mathrm{B}}$.
- If $\|p^{\mathrm{B}}\| \geq \Delta$, then the dogleg path intersects the boundary at one point, which is p_k^{D}. The corresponding τ can be computed by solving the scalar equation $\|\tilde{p}_k(\tau)\| = \Delta$.

B Two-dimensional subspace minimization

The dogleg method minimizes over the one-dimensional path defined by p^{U} and p^{B}. This can generalized by minimizing over the 2-D subspace spanned by $p^{\mathrm{U}} \propto -g_k$ and $p^{\mathrm{B}} = -B_k^{-1} g_k$.

Formally:

$$p_k^{\mathrm{2D}} = \operatorname*{argmin}_{p \in \mathbb{R}^d} \left\{ m_k(p) : \|p\| \leq \Delta_k, p \in \operatorname{span}\{g_k, B_k^{-1} g_k\} \right\}.$$

The minimizer is relatively easy to compute (amounts to finding the roots of a fourth degree polynomial).

Unlike dogleg, 2D-subspace minimization can readily be adapted to handle indefinite B_k. In this case, there exists $\lambda > 0$ such that $p_k^* = -(B_k + \lambda I)^{-1} g_k$ (by Theorem 1 from the last lecture). Therefore, we can change the feasible 2D subspace to

$$\operatorname{span}\left\{ g_k, (B_k + \alpha_k I)^{-1} g_k \right\},$$

where $\alpha_k \in (-\lambda_{\min}(B_k), -2\lambda_{\min}(B_k))$.