PRO 5961 Métodos de Otimização Não Linear

Celma de Oliveira Ribeiro

2023

Departamento de Engenharia de Produção
Universidade de São Paulo

Unconstrained optimization

Linear and Quadratic Approximations

Construct the linear and quadratic approximations for the function

$$f(x) = 3x_2 - \frac{x_1}{x_2}$$

at a point $x_0 = [2, 1]^t$.

The gradient is given by

Unconstrained optimization

Linear and Quadratic Approximations

Construct the linear and quadratic approximations for the function

$$f(x) = 3x_2 - \frac{x_1}{x_2}$$

at a point $x_0 = [2, 1]^t$.

The gradient is given by

$$\nabla f(x) = \left[\begin{array}{c} -\frac{1}{x_2} \\ 3 + \frac{x_1}{x_2^2} \end{array} \right] \Rightarrow \nabla f(x_0) = \left[\begin{array}{r} -1 \\ 5 \end{array} \right]$$

The hessian is

Unconstrained optimization

Linear and Quadratic Approximations

Construct the linear and quadratic approximations for the function

$$f(x) = 3x_2 - \frac{x_1}{x_2}$$

at a point $x_0 = [2, 1]^t$.

The gradient is given by

$$\nabla f(x) = \begin{bmatrix} -\frac{1}{x_2} \\ 3 + \frac{x_1}{x_2^2} \end{bmatrix} \Rightarrow \nabla f(x_0) = \begin{bmatrix} -1 \\ 5 \end{bmatrix}$$

The hessian is

$$\nabla^2 f(x) = \begin{bmatrix} 0 & \frac{1}{x_2^2} \\ \frac{1}{x_2^2} & -\frac{2x_1}{x_2^3} \end{bmatrix} \Rightarrow H(x_0) = \nabla^2 f(x_0) = \begin{bmatrix} 0 & 1 \\ 1 & -4 \end{bmatrix}$$

Unconstrained optimization

The linear approximation of the function is given by

$$l(x) = f(x_0) + \nabla(f(x_0))^t(x - x_0)$$

Unconstrained optimization

The linear approximation of the function is given by

$$l(x) = f(x_0) + \nabla(f(x_0))^t(x - x_0)$$

$$= 1 + \begin{bmatrix} -1 & 5 \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix} = -x_1 + 5x_2 - 2$$

The quadratic approximation is

$$q(x) = f(x_0) + \nabla f(x_0)^{'}(x - x_0) + \frac{1}{2}(x - x_0)^{'} H(x_0)(x - x_0)$$

Unconstrained optimization

The linear approximation of the function is given by

$$l(x) = f(x_0) + \nabla(f(x_0))^t(x - x_0)$$

$$= 1 + \begin{bmatrix} -1 & 5 \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix} = -x_1 + 5x_2 - 2$$

The quadratic approximation is

$$q(x) = f(x_0) + \nabla f(x_0)'(x - x_0) + \frac{1}{2}(x - x_0)' H(x_0)(x - x_0)$$

Verify!!!

$$q(x) = -x_1 + 5x_2 - 2 + \begin{bmatrix} x_1 - 2 & x_2 - 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -4 \end{bmatrix} \begin{bmatrix} x_1 - 2 \\ x_2 - 1 \end{bmatrix}$$

$$= -x_1 + 5x_2 - 2 + (x_1 - 2)(x_2 - 1) + (x_2 - 1)(x_1 - 4x_2 + 2)$$

A quadratic approximation of a function is often desired in optimization, as certain solution methods such as Newton's method show faster convergence for these functions

Unconstrained optimization

Newton's method

Consider unconstrained optimization problem:

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

with f twice continuously differentiable.

Motivation for Newton:

- Steepest descend is easy ... but can be slow
- Quadratics approximate nonlinear $f(x)$ better
- Faster local convergence
- More *robust* methods

Unconstrained optimization

Newton's method

- Steepest descent uses only first derivatives in selecting a suitable search direction.

Unconstrained optimization

Newton's method

- Steepest descent uses only first derivatives in selecting a suitable search direction.
- Newton's method uses first and second derivatives .

Unconstrained optimization

Newton's method

- Steepest descent uses only first derivatives in selecting a suitable search direction.
- Newton's method uses first and second derivatives .
- Given a starting point, construct a quadratic approximation to the objective function that matches the first and second derivative values at that point.

Unconstrained optimization

Newton's method

- Steepest descent uses only first derivatives in selecting a suitable search direction.
- Newton's method uses first and second derivatives .
- Given a starting point, construct a quadratic approximation to the objective function that matches the first and second derivative values at that point.
- Minimize the approximation (quadratic function).

Unconstrained optimization

Newton's method

- Steepest descent uses only first derivatives in selecting a suitable search direction.
- Newton's method uses first and second derivatives .
- Given a starting point, construct a quadratic approximation to the objective function that matches the first and second derivative values at that point.
- Minimize the approximation (quadratic function).
- The minimizer of the approximate function is used as the starting point in the next step and repeat the procedure iteratively.

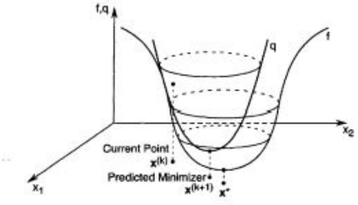
Multidimensional search

Newton's method

Consider the approximation of f at a given point x_k

$$q(x) = f(x_k) + \nabla f(x_k)'(x - x_k) + \frac{1}{2}(x - x_k)' H(x_k)(x - x_k)$$

with $H(x_k)$ the Hessian matrix of f at x_k.



Necessary condition for a minimum of q: $\nabla q(x) = 0$

Unconstrained Optimization

Newton's Method

The method is based on the quadratic approximation of the function f at a given point x_k

$$f(x) \approx q(x) =$$

$$\underbrace{f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)' H(x_k)(x - x_k)^2}_{\text{derivative equals to zero}}$$

Main step

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k)$$

Multidimensional search

Newton's method

Necessary condition for a minimum of q: $\nabla q(x) = 0$

$$\nabla q(x) = \nabla f(x_k) + H(x_k)(x - x_k)$$

$$\nabla q(x) = 0 \Rightarrow \nabla f(x_k)^{'} = -H(x_k)(x - x_k)$$

Solve the linear system $\nabla f(x_k)^{'} = -H(x_k)d$ to find the search direction

$$-H(x_k)^{-1}\nabla f(x_k)^{'} = (x - x_k)$$

Multidimensional search

Method of Newton

Then $\nabla f(x_k) = -H(x_k)(x - x_k)$

Assuming that the inverse of $H(x_k)$ exists, the sucessor point is

$$x_{k+1} = x_k - H(x_k)^{-1}\nabla f(x_k)$$

Algorithm - Newton's Method:

```
Step 0 Given $x_0$, set $k \leftarrow 0$
Step 1 $d_k = -H(x_k)^{-1}\nabla f(x_k)$. If $d_k = 0$, then stop.
Step 2 Choose step-size $\alpha_k = 1$.
Step 3 Set $x_{k+1} \leftarrow x_k + \alpha_k d_k$,
       $k \leftarrow k + 1$. Go to Step 1.
```

Example 1 Entrega

Consider $f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8\,(x_1 - 1)\,(x_2 - 3)$

$$\nabla f(x) = \begin{bmatrix} (x_1 - 1) - 1.8\,(x_2 - 3) \\ (x_2 - 3) - 1.8\,(x_1 - 1) \end{bmatrix}$$

$$\nabla^2 f(x) = \begin{bmatrix} 2 & -1.8 \\ -1.8 & 2 \end{bmatrix} \nabla^2 f(x)^{-1} = \begin{bmatrix} 2.6316 & 2.3684 \\ -2.3684 & 2.6316 \end{bmatrix}$$

$$x^* = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

- Write $f(x)$ as $f(x) = \frac{1}{2} x^t Q x + c^t x$
- Beginning with $x^{(0)} = \begin{bmatrix} 3 \\ -5 \end{bmatrix}$ find the next three steps of the Newton's method $\left(x^{(j)}\right), j = 1, 2, 3$

Note the following:

- The method assumes the Hessian $H(x_k)$ is nonsingular at each iteration.
- There is no guarantee that $f(x_{k+1}) \leq f(x_k)$
- Step 2 could be improved by a line-search of $f(x_k + \alpha d_k)$ to find an optimal value of the step-size parameter α.

Example 2

Consider $f(x) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$

Begins with $x^{(0)} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$

Newton's Method

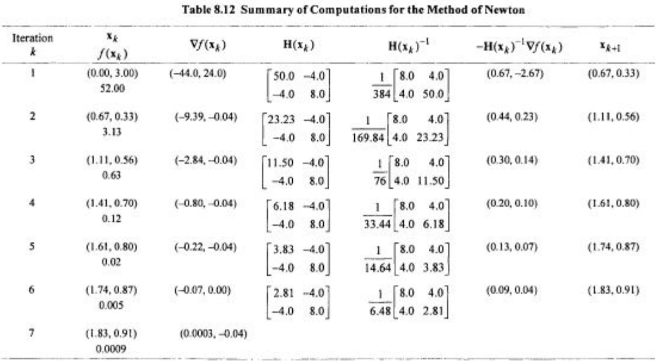
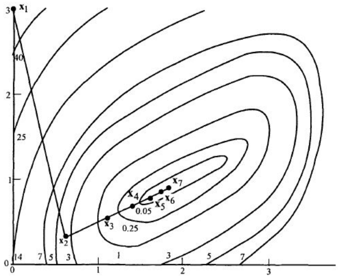


Table 8.12 Summary of Computations for the Method of Newton

Iteration k	$\mathbf{x}_k$ $f(\mathbf{x}_k)$	$\nabla f(\mathbf{x}_k)$	$\mathbf{H}(\mathbf{x}_k)$	$\mathbf{H}(\mathbf{x}_k)^{-1}$	$-\mathbf{H}(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$	$\mathbf{x}_{k+1}$
1	(0.00, 3.00) 52.00	(–44.0, 24.0)	$\begin{bmatrix} 50.0 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{384}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 50.0 \end{bmatrix}$	(0.67, –2.67)	(0.67, 0.33)
2	(0.67, 0.33) 3.13	(–9.39, –0.04)	$\begin{bmatrix} 23.23 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{169.84}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 23.23 \end{bmatrix}$	(0.44, 0.23)	(1.11, 0.56)
3	(1.11, 0.56) 0.63	(–2.84, –0.04)	$\begin{bmatrix} 11.50 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{76}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 11.50 \end{bmatrix}$	(0.30, 0.14)	(1.41, 0.70)
4	(1.41, 0.70) 0.12	(–0.80, –0.04)	$\begin{bmatrix} 6.18 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{33.44}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 6.18 \end{bmatrix}$	(0.20, 0.10)	(1.61, 0.80)
5	(1.61, 0.80) 0.02	(–0.22, –0.04)	$\begin{bmatrix} 3.83 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{14.64}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 3.83 \end{bmatrix}$	(0.13, 0.07)	(1.74, 0.87)
6	(1.74, 0.87) 0.005	(–0.07, 0.00)	$\begin{bmatrix} 2.81 & -4.0 \\ -4.0 & 8.0 \end{bmatrix}$	$\frac{1}{6.48}\begin{bmatrix} 8.0 & 4.0 \\ 4.0 & 2.81 \end{bmatrix}$	(0.09, 0.04)	(1.83, 0.91)
7	(1.83, 0.91) 0.0009	(0.0003, –0.04)				

Unconstrained Optimization

Newton's Method



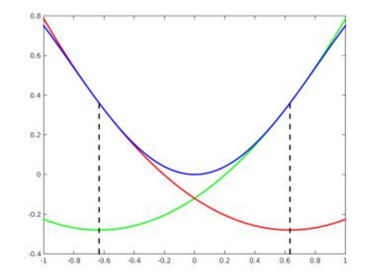
Multidimensional search

Some problems 1

Full Newton ($\alpha = 1$) step may fail to reduce $f(x)$, e.g.

$$min_x f(x) = x^2 - \frac{1}{4}x^4$$

$x_0 = \sqrt{\frac{2}{5}}$ creates the following s alternating iterates: $-\sqrt{\frac{2}{5}}$ and $\sqrt{\frac{2}{5}}$



Multidimensional search

Some problems 2

Hessian has indefinite curvature

Consider

$$min_x f(x) = x_1^4 + x_1 x_2 + (1 + x_2)^2$$

Starting Newton at $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

, $\nabla f(x_0) = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $\nabla^2 f(x_0) = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \Rightarrow d_0 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$

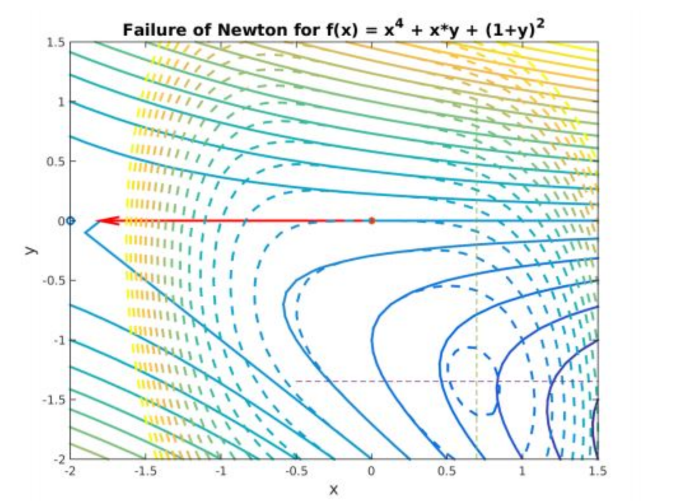
Line-search from x_0 in direction d_0

$x_0 + \alpha d_0 = \begin{bmatrix} -2\alpha \\ 0 \end{bmatrix} \Rightarrow f(x_0 + \alpha d_0) = (-2\alpha)^4 + 1 = 16\alpha + 1 > 0$

$\forall \alpha$. Then $\alpha = 0$ Then Newton's method stalls

Remedy: Modify Hessian to make it positive definite.

Multidimensional search



Multidimensional search

Newton's method - Convergence

Important question: how many iteration is needed to make sure x is ϵ-close to the minimizer x^*? That is

$$d(x, x^*) = \|x - x^*\| < \epsilon$$

where d is a distance function.

This refer to the convergence to the minimizer x^* it does not specify how many minimizers there are

Multidimensional search

Newton's method - Convergence

Important question: how many iteration is needed to make sure x is ϵ-close to the minimizer x^*? That is

$$d(x, x^*) = \|x - x^*\| < \epsilon$$

where d is a distance function.

This refer to the convergence to the minimizer x^* it does not specify how many minimizers there are

- Linear convergence

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|, c > 0$$

- quadratic convergence

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|^2, c > 0$$

Newton's method - Convergence

Important question: how many iteration is needed to make sure x is ϵ-close to the minimizer x^*? That is

$$d(x, x^*) = \|x - x^*\| < \epsilon$$

where d is a distance function.

This refer to the convergence to the minimizer x^* it does not specify how many minimizers there are

- Linear convergence

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|, c > 0$$

- quadratic convergence

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|^2, c > 0$$

Theorem (Newton's Method). Let $f \in C^3$ on $\mathbb{R}^n$, and assume that at the local minimum point x^*, the Hessian $H(x^*)$ is positive definite. Then if started sufficiently close to x^*, the points generated by Newton's method converge to x^*. The order of convergence is at least two.

Newton's method - Convergence

When f is a quadratic function Newton´s method reaches the point x^* such that $\nabla f(x^*) = 0$ in just one step starting from any initial point.

Suppose $Q = Q^t$ invertible and $f(x) = \frac{1}{2}x^tQx - x^tb$

$\nabla f(x) = Qx - b$ and $H(x) = \nabla^2 f(x) = Q$

Given any initial point, x_0, by Newton's algorithm:

$$x_1 = x_0 - H(x_0)^{-1}\nabla f(x_0)$$

$$x_1 = x_0 - Q^{-1}[Qx_0 - b])$$

$$x_1 = Q^{-1}b = x^*$$

$$\boxed{\textbf{Convergence}}$$

Entregar: Give a numerical example with $Q \in \mathbb{R}^3$

Multidimensional search

Newton's method - Convergence

Theorem: Let $\{x_k\}$ be the sequence generated by Newton's method for minimizing a given objective function $f(x)$.

If the Hessian $H(x_k)$ is positive definite and $\nabla f(x_k) \neq 0$,

then the search direction $d^k = -H(x_k)^{-1}\nabla f(x_k) = x_{k+1} - x_k$ is a descent direction for f in the sense that there exists an $\bar{\alpha}$ such that for all $\alpha \in (0, \bar{\alpha})$ $f(x_k + \alpha d^k) < f(x_k)$

Multidimensional search

Newton's method

- Advantages
  - The convergence to the solution of $\nabla f(x) = 0$ will be rapid, at least for initial guesses that are near to a solution
  - If f is quadratic the Newton's method takes us to a critical point in just one iteration

Multidimensional search

Newton's method

- Advantages
  - The convergence to the solution of $\nabla f(x) = 0$ will be rapid, at least for initial guesses that are near to a solution
  - If f is quadratic the Newton's method takes us to a critical point in just one iteration
- Disadvantages
  - Since the Hessian may not always be positive definite, $H(x_k)^{-1}\nabla f(x_k)$ may not always be a descent direction
  - For non-quadratic objectives and particularly at points far from the minimizer, the step direction is not necessarily better that the steepest descent step direction.
  - Factorization of the Hessian may require considerable effort if n is large or the Hessian is dense

Multidimensional search

Newton's method

- Convergence

  If f is twice differentiable and the Hessian is "well behaved" in a neighborhood of a solution x^* and if the sufficient conditions are satisfied,

  - If the starting point is sufficiently close to x^* the sequence of interates converges to x^*
  - the rate of convergence of x_k is quadratic ($\lim_{k\to\infty} \frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|^2} < M$)

Newton's method converges quadratically ... steepest descend only linearly

Newton's method - Modifications

Although Newton's method is very attractive in terms of its convergence properties near the solution, it requires modification before it can be used at points that are remote from the solution

- The theorem motivates the following modification of Newton's method:

$$x_{k+1} = x_k - \alpha_k H(x_k)^{-1} \nabla f(x_k)$$

  where α_k is selected to minimize $f(\alpha)$. that is, at each iteration, we perform a line search in the direction

- The Hessian matrix may not be positive definite!

  A basic consideration for Newton's method can be seen most clearly by a brief examination of the general class of algorithms

$$x_{k+1} = x_k - \alpha M_k g_k$$

  , where M_k is an $n \times n$ matrix, α is a positive search parameter, and $g_k = \nabla f(x_k)$.

  Both steepest descent $(M_k = I)$ and Newton's method $(M_k = H(x_k)^1)$ belong to this class.

Newton's method - Modifications

Ref: Luenberger

Levenberg–Marquardt method

Common approach: take $M_k = [\epsilon_k I + H(x_k)]^{-1}$ for some non-negative value of ϵ_k.

- This can be regarded as a kind of compromise between steepest descent (ϵ_k very large) and Newton's method ($\epsilon_k = 0$).
- There is always an ϵ_k that makes M_k positive definite.

Fix a constant $\delta > 0$. Given x_k, calculate the eigenvalues of $H(x_k)$ and let ϵ_k and the smallest non negative constant for which the matrix $\epsilon_k I + H(x_k)$ has eigenvalues greater than or equal to δ

Then define

$$d_k = -[\epsilon_k I + H(x_k)]^{-1} * \nabla f(x_k)$$

and $x_{k+1} = x_k + \alpha_k d_k$ where α_k minimizes $f(x_k) + \alpha d_k$, $\alpha > 0$.

The selection of an appropriate δ is an art.

- δ small $\Rightarrow$ nearly singular matrices to be inverted
- δ large $\Rightarrow$ order two convergence may be lost

Conditioning

- Consider the system

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

  Solution $x^{'} = \begin{bmatrix} 1 & 1 \end{bmatrix}$
  Change the first element of right-hand-size from 3 to 3.00001. The exact solution is $x^{'} = \begin{bmatrix} 0.99999 & 1.00001 \end{bmatrix}$

- An ill conditioned matrix

$$\begin{bmatrix} 1.00001 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.00001 \\ 2 \end{bmatrix}$$

  Solution $x^{'} = \begin{bmatrix} 1 & 1 \end{bmatrix}$
  Change the first element of right-hand-size from 2.00001 to 2 . The exact solution is $x^{'} = \begin{bmatrix} 0 & 2 \end{bmatrix}$

Conditioning

Suppose that $A \in \mathbb{R}^{n \times n}$ is non singular. The **condition number** of A is defined as $\kappa(A) = \|A\| \|A^{-1}\|$. If A is singular the condition number is ∞ [1]

A matrix with a **large** condition number is said to be ill conditioned

Unconstrained Optimization

Scaling

- Scaling: transforming variables that is, change their units
- It has a significant influence on the performance of optimization methods
- Useful when the variables in the initial formulation of the problem have widely differing magnitudes

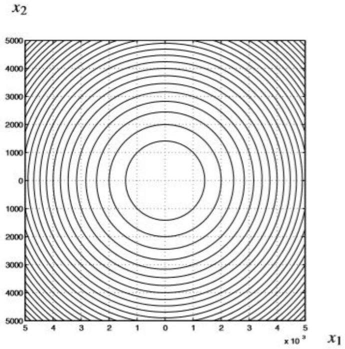
Unconstrained Optimization

Example scaling

$f : \mathbb{R}^2 \to R$, $f(x) = (1000x_1)^2 + (\frac{x_2}{1000})^2$

x_1 usually lies in $[-0.01; 0.01]$

x_2 usually lies in $[-10^4; 10^4]$



Unconstrained Optimization

Example scaling

If you want to obtain an solution that yields an objective that is within one unit of the minimum then we need to obtain a value of x such that (approximately):

$$|x_1 - x_1^*| < 0.001 \, |x_2 - x_2^*| < 1000$$

Errors in the different components of x have differing effects on the objective.

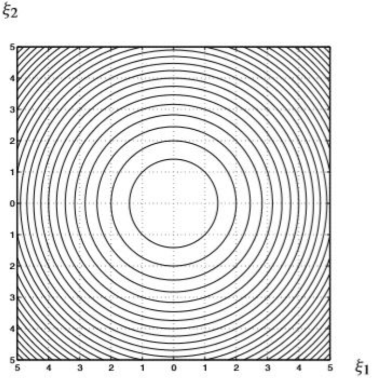
Define: $\epsilon_1 = 1000x_1$ and $\epsilon_2 = \frac{x_2}{1000}$

Unconstrained Optimization

Example scaling

$$\phi(x) = (\epsilon_1)^2 + (\epsilon_2)^2$$

Contour sets of $\phi(x)$


ξ_2
ξ_1