Multivariable Taylor Expansions, Newton's Method and Optimization

1 Introduction

Optimization problems arise throughout science and engineering. Typical questions are

- Where does a function attain its minimum?
- How can nonlinear equations be solved efficiently?
- How can we approximate complicated functions locally?

The key mathematical tools are

1. Taylor expansions,
2. the gradient,
3. the Hessian matrix,
4. Newton's method,
5. steepest descent.

These lecture notes assume familiarity with

- partial derivatives,
- the chain rule,
- gradients,
- divergence,
- matrix multiplication.

The emphasis is on geometric understanding together with practical numerical algorithms.

2 Notation

Throughout these notes we consider functions

$$f : \mathbb{R}^n \to \mathbb{R}.$$

For visualization we mostly use the two-dimensional case

$$f(x, y).$$

Points are written as vectors

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

For functions of two variables we simply write

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}.$$

2.1 Gradient

Definition 2.1. The *gradient* of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \dfrac{\partial f}{\partial x_1} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{pmatrix}.$$

The gradient points in the direction of maximal increase of the function.
Its negative therefore points in the direction of steepest decrease.

2.2 The Hessian Matrix

Definition 2.2. Suppose all second-order partial derivatives exist.
The Hessian matrix of f is

$$H(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

If all second derivatives are continuous, then

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i},$$

so that the Hessian is symmetric.

2.3 Vector notation

Using vectors greatly simplifies many formulas.
Instead of writing

$$f(x + h, y + k),$$

we write

$$f(\mathbf{x} + \mathbf{h}).$$

Likewise,

$$\mathbf{h} = \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix}.$$

Many multivariable formulas then become elegant generalizations of the one-dimensional case.

2.4 A motivating example

Consider the function

$$f(x, y) = x^2 + 4y^2.$$

Its gradient equals

$$\nabla f(x, y) = \begin{pmatrix} 2x \\ 8y \end{pmatrix},$$

while its Hessian is

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 8 \end{pmatrix}.$$

Observe that

- the gradient depends on the position;
- the Hessian is constant;
- the function is curved four times more strongly in the y-direction than in the x-direction.

The Hessian therefore contains information about the local curvature of the surface.

In the next chapter we shall see that the Hessian naturally appears in the second-order Taylor expansion.

3 Taylor Expansions in Several Variables

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is sufficiently smooth. Just as in one dimension, we would like to approximate the function near a given point by a polynomial.

The approximation is obtained by matching the value of the function and its derivatives at the expansion point.

3.1 Review: Taylor expansion in one dimension

For a sufficiently smooth function $f : \mathbb{R} \to \mathbb{R}$,

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \mathcal{O}(h^3).$$

The linear term gives the tangent line.

The quadratic term describes the local curvature.

Our goal is to derive the analogous formula for functions of several variables.

3.2 First-order approximation

Consider a function

$$f(x, y).$$

Suppose we move from the point

$$(x, y)$$

to

$$(x + h,\ y + k),$$

where both h and k are small.

The first-order Taylor approximation is

$$f(x + h, y + k) \approx f(x, y) + h\, f_x(x, y) + k\, f_y(x, y).$$

In vector notation,

$$\mathbf{h} = \begin{pmatrix} h \\ k \end{pmatrix},$$

and therefore

$$f(\mathbf{x} + \mathbf{h}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{h}.$$

Notice that the correction is simply the dot product of the gradient with the displacement vector.

Example 3.1. Let

$$f(x, y) = x^2 + y^2.$$

Then

$$\nabla f = \begin{pmatrix} 2x \\ 2y \end{pmatrix}.$$

Near the point

$$(1,2),$$

the linear approximation becomes

$$f(1+h,2+k) \approx 5 + 2h + 4k.$$

Thus the graph of the function is approximated by its tangent plane.

3.3 Second-order approximation

The first-order approximation ignores curvature.

To obtain a more accurate approximation we include the second derivatives.

For two variables,

$$\begin{aligned} f(x+h,y+k) =& f(x,y) + hf_x + kf_y \\ &+ \frac{1}{2}\left(h^2 f_{xx} + 2hkf_{xy} + k^2 f_{yy}\right) \\ &+ \mathcal{O}(\|(h,k)\|^3). \end{aligned}$$

Notice the mixed derivative term

$$2hkf_{xy},$$

which measures how the variables interact.

3.4 The Hessian

The quadratic terms can be written compactly by introducing the Hessian matrix

$$H = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix}.$$

Since

$$f_{xy} = f_{yx},$$

the Hessian is symmetric whenever the second derivatives are continuous.

Now observe that

$$\mathbf{h}^T H \mathbf{h} = \begin{pmatrix} h & k \end{pmatrix} \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix} \begin{pmatrix} h \\ k \end{pmatrix},$$

which expands to

$$h^2 f_{xx} + 2hkf_{xy} + k^2 f_{yy}.$$

Hence the Taylor expansion becomes

$$\boxed{f(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{h} + \frac{1}{2}\mathbf{h}^T H(\mathbf{x})\mathbf{h} + \mathcal{O}(\|\mathbf{h}\|^3).}$$

This compact formula is one of the most important equations in multivariable calculus.

3.5 Interpretation

The Taylor expansion has a clear geometric interpretation.

Constant term	Height of the surface
Gradient	Slope of the surface
Hessian	Local curvature

Thus

$$\boxed{\text{Gradient} = \text{first-order information}}$$

while

$$\boxed{\text{Hessian} = \text{second-order information.}}$$

3.6 General formula in $\mathbb{R}^n$

Everything extends naturally to n dimensions.

For

$$\mathbf{x}, \mathbf{h} \in \mathbb{R}^n,$$

the second-order Taylor expansion is

$$\boxed{f(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{h} + \frac{1}{2}\mathbf{h}^T H(\mathbf{x})\mathbf{h} + \mathcal{O}(\|\mathbf{h}\|^3).}$$

This approximation forms the mathematical foundation of many optimization algorithms, including Newton's method.

Example 3.2. Consider

$$f(x, y) = x^2 + xy + 2y^2.$$

The gradient is

$$\nabla f = \begin{pmatrix} 2x + y \\ x + 4y \end{pmatrix},$$

and the Hessian equals

$$H = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}.$$

Expanding about the origin gives

$$f(h, k) = h^2 + hk + 2k^2.$$

Since the function is already quadratic, its Taylor polynomial is exact.

3.7 Summary

The Taylor expansion replaces a complicated function by a local polynomial.

The approximation consists of three ingredients:

Function value	$f(\mathbf{x})$,
Slope	$\nabla f(\mathbf{x})$,
Curvature	$H(\mathbf{x})$.

These quantities are sufficient to construct powerful numerical algorithms for

- solving nonlinear equations,
- locating extrema,
- minimizing nonlinear functions.

In the next section we study how the Hessian determines whether a stationary point is a minimum, maximum or saddle point.

4 Stationary Points and the Hessian

In optimization we are interested in finding points where a function attains a local minimum or a local maximum.

For a differentiable function these points can only occur when the gradient vanishes.

4.1 Stationary points

Definition 4.1. A point $\mathbf{x}^* \in \mathbb{R}^n$ is called a *stationary point* (or *critical point*) if

$$\nabla f(\mathbf{x}^*) = \mathbf{0}.$$

At a stationary point the tangent plane is horizontal.

Notice that every local minimum and every local maximum is a stationary point, provided the function is differentiable.

The converse, however, is not true: a stationary point need not be a minimum or a maximum.

4.2 Why the Hessian matters

Suppose that

$$\nabla f(\mathbf{x}^*) = \mathbf{0}.$$

The second-order Taylor expansion becomes

$$f(\mathbf{x}^* + \mathbf{h}) = f(\mathbf{x}^*) + \frac{1}{2}\mathbf{h}^T H(\mathbf{x}^*)\mathbf{h} + \mathcal{O}(\|\mathbf{h}\|^3).$$

The linear term disappears because the gradient is zero.

Consequently, the local behaviour is completely determined by the quadratic form

$$\mathbf{h}^T H \mathbf{h}.$$

This observation is the basis of the second derivative test.

4.3 Positive and negative definite matrices

Definition 4.2. A symmetric matrix A is called

- *positive definite* if

$$\mathbf{x}^T A \mathbf{x} > 0 \qquad \text{for every } \mathbf{x} \neq 0;$$

- *negative definite* if

$$\mathbf{x}^T A \mathbf{x} < 0 \qquad \text{for every } \mathbf{x} \neq 0;$$

- *indefinite* if

$$\mathbf{x}^T A \mathbf{x}$$

takes both positive and negative values.

The definiteness of the Hessian determines the type of stationary point.

4.4 Second derivative test

Theorem 4.3 (Second Derivative Test)**.** *Suppose*

$$\nabla f(\mathbf{x}^*) = 0.$$

Then

1. *If $H(\mathbf{x}^*)$ is positive definite, then $\mathbf{x}^*$ is a strict local minimum.*
2. *If $H(\mathbf{x}^*)$ is negative definite, then $\mathbf{x}^*$ is a strict local maximum.*
3. *If $H(\mathbf{x}^*)$ is indefinite, then $\mathbf{x}^*$ is a saddle point.*

Proof. Since

$$\nabla f(\mathbf{x}^*) = 0,$$

Taylor's theorem gives

$$f(\mathbf{x}^* + \mathbf{h}) = f(\mathbf{x}^*) + \frac{1}{2}\mathbf{h}^T H \mathbf{h} + \mathcal{O}(\|\mathbf{h}\|^3).$$

If the Hessian is positive definite,

$$\mathbf{h}^T H \mathbf{h} > 0$$

for every nonzero $\mathbf{h}$.
Hence

$$f(\mathbf{x}^* + \mathbf{h}) > f(\mathbf{x}^*)$$

for sufficiently small $\mathbf{h}$.
Therefore $\mathbf{x}^*$ is a local minimum.
The remaining cases follow analogously.

□
□

4.5 The two-dimensional case

For

$$f(x, y),$$

the Hessian is

$$H = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}.$$

Instead of computing eigenvalues, we may use the determinant

$$D = f_{xx} f_{yy} - f_{xy}^2.$$

The classification becomes

$D > 0,\ f_{xx} > 0$	$\Longrightarrow$	local minimum,
$D > 0,\ f_{xx} < 0$	$\Longrightarrow$	local maximum,
$D < 0$	$\Longrightarrow$	saddle point.

If

$$D = 0,$$

the test is inconclusive.

Example 4.4. Consider

$$f(x, y) = x^2 + y^2.$$

Then

$$\nabla f = \begin{pmatrix} 2x \\ 2y \end{pmatrix},$$

so the only stationary point is

$$(0, 0).$$

Furthermore,

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix},$$

which is positive definite.
Hence

$$(0, 0)$$

is a strict local minimum.

Example 4.5. Consider

$$f(x, y) = x^2 - y^2.$$

Again,

$$\nabla f = \begin{pmatrix} 2x \\ -2y \end{pmatrix},$$

so the stationary point is

$$(0, 0).$$

The Hessian equals

$$H = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix},$$

which is indefinite.
Therefore

$$(0, 0)$$

is a saddle point.
Indeed,

$$f(x, 0) = x^2 > 0,$$

whereas

$$f(0, y) = -y^2 < 0.$$

The function increases in one direction and decreases in another.

4.6 Geometric interpretation

The Hessian measures the local curvature of the graph

$$z = f(x, y).$$

Positive definite Hessian	Bowl-shaped surface
Negative definite Hessian	Upside-down bowl
Indefinite Hessian	Saddle surface

The eigenvalues of the Hessian determine the principal curvatures of the surface.

Large eigenvalues correspond to steep curvature, while small eigenvalues indicate relatively flat directions.

4.7 Summary

To classify a stationary point, proceed as follows.

1. Compute the gradient.
2. Solve

$$\nabla f = \mathbf{0}.$$

3. Compute the Hessian.
4. Determine whether the Hessian is positive definite, negative definite or indefinite.
5. Conclude whether the stationary point is a minimum, maximum or saddle point.

The Hessian will also play a central role in Newton's method for optimization, which will be discussed later.

5 Quadratic Functions and Positive Definite Matrices

The second-order Taylor expansion shows that every sufficiently smooth function can locally be approximated by a quadratic function.

Understanding quadratic functions is therefore essential for the study of optimization algorithms.

5.1 Quadratic functions

A quadratic function on $\mathbb{R}^n$ has the form

$$q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c,$$

where

$$A \in \mathbb{R}^{n\times n}, \qquad \mathbf{b} \in \mathbb{R}^n, \qquad c \in \mathbb{R}.$$

Without loss of generality we assume that A is symmetric, since

$$\mathbf{x}^T A\mathbf{x} = \mathbf{x}^T \left(\frac{1}{2}(A + A^T)\right)\mathbf{x}.$$

Hence only the symmetric part contributes to the quadratic form.

5.2 Gradient and Hessian

For the quadratic function

$$q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c,$$

the gradient is

$$\nabla q(\mathbf{x}) = A\mathbf{x} + \mathbf{b},$$

while the Hessian equals

$$H(\mathbf{x}) = A.$$

Thus the Hessian of a quadratic function is constant.

Proof. Using matrix differentiation,

$$\frac{\partial}{\partial \mathbf{x}}\left(\frac{1}{2}\mathbf{x}^T A\mathbf{x}\right) = A\mathbf{x},$$

provided that A is symmetric.
Differentiating once more immediately gives

$$H = A.$$

□

□

5.3 Stationary point

A stationary point satisfies

$$\nabla q(\mathbf{x}) = 0,$$

that is,

$$A\mathbf{x} = -\mathbf{b}.$$

Hence minimizing a quadratic function is equivalent to solving a linear system.
If A is invertible,

$$\boxed{\mathbf{x}^* = -A^{-1}\mathbf{b}.}$$

5.4 Positive definite matrices

The nature of the stationary point depends entirely on the matrix A.

Definition 5.1. A symmetric matrix A is called *positive definite* if

$$\mathbf{x}^T A\mathbf{x} > 0 \qquad \text{for every } \mathbf{x} \neq 0.$$

Likewise,

$$\mathbf{x}^T A\mathbf{x} < 0$$

for all nonzero vectors if A is negative definite.

5.5 Eigenvalue characterization

The following theorem provides a practical criterion.

Theorem 5.2. *Let A be symmetric.*
Then

$$A \text{ is positive definite}$$

if and only if
all eigenvalues of A are positive.
Similarly,

$$A \text{ is negative definite}$$

if and only if
all eigenvalues are negative.

Idea of the proof. Since A is symmetric, it possesses an orthonormal eigenbasis.
Hence

$$A = Q\Lambda Q^T,$$

where

$$\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n).$$

Writing

$$\mathbf{x} = Q\mathbf{y},$$

gives

$$\mathbf{x}^T A\mathbf{x} = \mathbf{y}^T \Lambda \mathbf{y} = \sum_{i=1}^{n} \lambda_i y_i^2.$$

This expression is positive for every nonzero vector if and only if every eigenvalue is positive. □

5.6 Example

Consider

$$q(x, y) = x^2 + xy + 2y^2.$$

Its Hessian equals

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}.$$

The characteristic polynomial is

$$\lambda^2 - 6\lambda + 7 = 0,$$

so that

$$\lambda_{1,2} = 3 \pm \sqrt{2}.$$

Both eigenvalues are positive.
Therefore the quadratic function possesses a unique global minimum.

5.7 Geometry

The graph of a quadratic function

$$z = q(x, y)$$

is called a paraboloid.
Several different shapes may occur.

Positive definite Hessian	Elliptic paraboloid
Negative definite Hessian	Inverted paraboloid
Indefinite Hessian	Hyperbolic paraboloid (saddle)

The eigenvectors determine the principal directions of curvature, while the eigenvalues determine the amount of curvature.

5.8 Connection with Taylor's theorem

Suppose

$$\nabla f(\mathbf{x}_0) = 0.$$

The second-order Taylor approximation becomes

$$f(\mathbf{x}_0 + \mathbf{h}) \approx f(\mathbf{x}_0) + \frac{1}{2}\mathbf{h}^T H(\mathbf{x}_0)\mathbf{h}.$$

Thus every smooth function behaves locally like a quadratic function whose Hessian equals

$$A = H(\mathbf{x}_0).$$

Consequently,
all local optimization algorithms are based on repeatedly constructing and minimizing local quadratic models.

5.9 Summary

Quadratic functions play a central role in optimization because

- every smooth function is locally quadratic;
- the Hessian is constant;
- minimizing a quadratic function reduces to solving

$$A\mathbf{x} = -\mathbf{b};$$

- Newton's method is obtained by replacing a nonlinear function by its local quadratic approximation.

In the next section we use this idea to derive Newton's method for solving nonlinear systems of equations.

6 Newton's Method for Nonlinear Equations

Many problems in science and engineering require solving nonlinear equations.

A general nonlinear system can be written as

$$F(\mathbf{x}) = \mathbf{0},$$

where

$$F : \mathbb{R}^n \to \mathbb{R}^n.$$

Newton's method is one of the most important algorithms for solving such problems.
The main idea is simple:

replace the nonlinear function by its local linear approximation.

6.1 Newton's method in one dimension

Consider the scalar equation

$$f(x) = 0.$$

Suppose that x_k is an approximation of the solution.
The Taylor expansion around x_k gives

$$f(x_k + h) \approx f(x_k) + hf'(x_k).$$

The next approximation is obtained by requiring the linear approximation to be zero:

$$f(x_k) + hf'(x_k) = 0.$$

Solving for h gives

$$h = -\frac{f(x_k)}{f'(x_k)}.$$

Since

$$x_{k+1} = x_k + h,$$

we obtain Newton's iteration formula

$$\boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.}$$

6.2 Geometric interpretation

Newton's method replaces the nonlinear curve by its tangent line.

The tangent line at x_k is

$$y = f(x_k) + f'(x_k)(x - x_k).$$

The next approximation is the intersection of this tangent line with the x-axis.
Thus Newton's method can be interpreted as a repeated tangent construction.

6.3 Newton's method in several dimensions

We now consider a system

$$F(\mathbf{x}) = \mathbf{0},$$

where

$$F(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix}.$$

For two variables,

$$F(x, y) = \begin{pmatrix} f(x, y) \\ g(x, y) \end{pmatrix}.$$

The Taylor expansion becomes

$$F(\mathbf{x} + \mathbf{h}) \approx F(\mathbf{x}) + J(\mathbf{x})\mathbf{h},$$

where

$$J(\mathbf{x})$$

is the Jacobian matrix.

Definition 6.1. The Jacobian matrix of

$$F : \mathbb{R}^n \to \mathbb{R}^n$$

is

$$J(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}.$$

6.4 Derivation of the Newton step

We want to find

$$\mathbf{x}^* = \mathbf{x} + \mathbf{h}$$

such that

$$F(\mathbf{x}^*) = 0.$$

Using the linear approximation,

$$F(\mathbf{x}) + J(\mathbf{x})\mathbf{h} \approx 0.$$

Therefore the Newton correction satisfies

$$J(\mathbf{x})\mathbf{h} = -F(\mathbf{x}).$$

After solving this linear system,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k.$$

Hence Newton's method is

$$\boxed{J(\mathbf{x}_k)\mathbf{h}_k = -F(\mathbf{x}_k),}$$

followed by

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k.}$$

6.5 Two-dimensional system

Consider

$$\begin{cases} f(x,y) = 0, \\ g(x,y) = 0. \end{cases}$$

The Jacobian is

$$J(x,y) = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}.$$

At iteration k we solve

$$\begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}_{(x_k,y_k)} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} f(x_k,y_k) \\ g(x_k,y_k) \end{pmatrix}.$$

The new approximation becomes

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}.$$

6.6 Algorithm

Algorithm 1 Newton method for nonlinear systems

Choose initial guess $\mathbf{x}_0$
for $k = 0, 1, 2, \ldots$ **do**
  Compute $F(\mathbf{x}_k)$
  Compute Jacobian $J(\mathbf{x}_k)$
  Solve

$$J(\mathbf{x}_k)\mathbf{h}_k = -F(\mathbf{x}_k)$$

  Set

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k$$

  if $\|\mathbf{h}_k\|$ is sufficiently small **then**
    stop
  end if
end for

6.7 Example

Consider the system

$$\begin{cases} x^2 + y^2 - 4 = 0, \\ x - y = 0. \end{cases}$$

The Jacobian is

$$J(x, y) = \begin{pmatrix} 2x & 2y \\ 1 & -1 \end{pmatrix}.$$

Starting from an initial guess, Newton's method repeatedly solves a linear system to approach the intersection of the two curves.

6.8 Convergence

Under suitable assumptions, Newton's method converges quadratically.

This means that once the iteration is sufficiently close to the solution,

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq C\|\mathbf{x}_k - \mathbf{x}^*\|^2.$$

Quadratic convergence is extremely fast.

However, Newton's method is a local method:

- a good initial guess is important;
- the Jacobian must be nonsingular;
- far away from the solution convergence is not guaranteed.

6.9 Summary

Newton's method is based on one fundamental idea:

Replace a nonlinear function by its local linear approximation.

For scalar equations this uses the derivative.

For systems it uses the Jacobian.

For optimization, the same philosophy will lead to Newton's method based on the Hessian.

7 Newton's Method for Optimization

In the previous section Newton's method was derived for solving nonlinear equations.

Optimization problems can also be transformed into a nonlinear equation.

For a differentiable function

$$f : \mathbb{R}^n \to \mathbb{R},$$

a necessary condition for a local minimum is

$$\nabla f(\mathbf{x}^*) = 0.$$

Therefore finding a minimum is equivalent to solving the nonlinear system

$$\nabla f(\mathbf{x}) = \mathbf{0}.$$

Newton's method applied to this system leads to Newton optimization.

7.1 From minimization to root finding

Define

$$F(\mathbf{x}) = \nabla f(\mathbf{x}).$$

Then the optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x})$$

requires solving

$$F(\mathbf{x}) = 0.$$

The Jacobian of F is

$$J_F(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = H(\mathbf{x}),$$

the Hessian matrix.
Therefore Newton's method for the equation

$$\nabla f(\mathbf{x}) = 0$$

uses the Hessian.

7.2 Derivation from Taylor expansion

Let $\mathbf{x}_k$ be the current approximation.
The second-order Taylor approximation is

$$f(\mathbf{x}_k + \mathbf{h}) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{h} + \frac{1}{2}\mathbf{h}^T H(\mathbf{x}_k)\mathbf{h}.$$

The idea of Newton optimization is:

minimize the quadratic approximation instead of the original function.

Define

$$m_k(\mathbf{h}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{h} + \frac{1}{2}\mathbf{h}^T H(\mathbf{x}_k)\mathbf{h}.$$

The minimum of this quadratic model satisfies

$$\nabla_{\mathbf{h}} m_k(\mathbf{h}) = 0.$$

Differentiating gives

$$\nabla f(\mathbf{x}_k) + H(\mathbf{x}_k)\mathbf{h} = 0.$$

Therefore the Newton step satisfies

$$\boxed{H(\mathbf{x}_k)\mathbf{h}_k = -\nabla f(\mathbf{x}_k).}$$

The new point is

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k.}$$

Equivalently,

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - H(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k).}$$

7.3 Geometric interpretation

Gradient descent uses only the slope:

$$-\nabla f(\mathbf{x}).$$

Newton's method also uses the curvature:

$$H(\mathbf{x}).$$

The Hessian changes the direction and length of the step.

Therefore Newton's method automatically compensates for different scales in different directions.

7.4 Quadratic functions

For a quadratic function

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c,$$

we have

$$\nabla f(\mathbf{x}) = A\mathbf{x} + \mathbf{b},$$

and

$$H = A.$$

The Newton step satisfies

$$A\mathbf{h} = -(A\mathbf{x} + \mathbf{b}).$$

Hence

$$A(\mathbf{x} + \mathbf{h}) = -\mathbf{b}.$$

Therefore

$$\mathbf{x}_{k+1} = -A^{-1}\mathbf{b}.$$

Thus:

Newton's method finds the minimum of a quadratic function in one step.

This explains why quadratic functions are so important in optimization.

7.5 Example: two-dimensional quadratic

Consider

$$f(x, y) = x^2 + 2y^2 - 2x - 4y.$$

The gradient is

$$\nabla f = \begin{pmatrix} 2x - 2 \\ 4y - 4 \end{pmatrix}.$$

The Hessian is

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}.$$

The Newton equation is

$$\begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} 2x - 2 \\ 4y - 4 \end{pmatrix}.$$

Hence

$$\Delta x = 1 - x,$$

and

$$\Delta y = 1 - y.$$

Therefore

$$(x_{k+1}, y_{k+1}) = (1, 1)$$

after one iteration.
The minimum is found exactly.

7.6 Newton optimization algorithm

Algorithm 2 Newton method for unconstrained optimization

Choose initial point $\mathbf{x}_0$
for $k = 0, 1, 2, \ldots$ **do**
  Compute gradient

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k)$$

  Compute Hessian

$$H_k = H(\mathbf{x}_k)$$

  Solve

$$H_k \mathbf{h}_k = -\mathbf{g}_k$$

  Update

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k$$

  if $\|\mathbf{g}_k\|$ is sufficiently small **then**
    stop
  end if
end for

7.7 Advantages and disadvantages

Newton's method has important advantages:

- very fast convergence near a minimum;

- uses curvature information;
- independent of scaling of variables.

However:

- computing the Hessian can be expensive;
- solving the linear system can be costly;
- the Hessian may not be positive definite;
- convergence is not guaranteed from arbitrary starting points.

These disadvantages motivate simpler methods based only on the gradient.
The most important example is the steepest descent method.

7.8 Summary

Newton optimization follows from the second-order Taylor approximation:

$$\boxed{\text{gradient gives the direction, Hessian gives the correction.}}$$

The iteration is

$$\boxed{H(\mathbf{x}_k)\mathbf{h}_k = -\nabla f(\mathbf{x}_k),}$$

with

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k.}$$

In the next section we study steepest descent, which ignores curvature and uses only the gradient.

8 Steepest Descent Method

Newton's method uses both the gradient and the Hessian.
Although this leads to fast convergence, computing and storing the Hessian can be expensive.
A simpler approach is to use only first-order information.
This leads to the steepest descent method.

8.1 The direction of maximum decrease

Consider a differentiable function

$$f : \mathbb{R}^n \to \mathbb{R}.$$

The directional derivative in the direction $\mathbf{d}$ is

$$D_{\mathbf{d}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T \mathbf{d}.$$

Assume that

$$\|\mathbf{d}\| = 1.$$

We want to find the direction in which the function decreases most rapidly.
Therefore we minimize

$$\nabla f(\mathbf{x})^T \mathbf{d}$$

subject to

$$\|\mathbf{d}\| = 1.$$

By the Cauchy–Schwarz inequality,

$$\nabla f(\mathbf{x})^T \mathbf{d} \geq -\|\nabla f(\mathbf{x})\|.$$

Equality occurs for

$$\boxed{\mathbf{d} = -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}.}$$

Hence the direction of steepest descent is

$$\boxed{-\nabla f(\mathbf{x}).}$$

8.2 The steepest descent iteration

The next point is chosen as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k,$$

where

$$\mathbf{d}_k = -\nabla f(\mathbf{x}_k).$$

Therefore

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k).}$$

The parameter

$$\alpha_k > 0$$

is called the step size.

8.3 Choice of the step size

The step size determines how far we move in the descent direction.

There are several possibilities.

1. Fixed step size:

$$\alpha_k = \alpha.$$

2. Diminishing step size:

$$\alpha_k \to 0.$$

3. Exact line search:

   choose α_k such that

$$f(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k))$$

   is minimized.

The third option is theoretically attractive because it gives the best step along the chosen direction.

8.4 Exact line search

Define the one-dimensional function

$$\phi(\alpha) = f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)).$$

The optimal step satisfies

$$\phi'(\alpha) = 0.$$

Using the chain rule,

$$\phi'(\alpha) = -\nabla f(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k))^T \nabla f(\mathbf{x}_k).$$

Therefore the optimal step satisfies

$$\boxed{\nabla f(\mathbf{x}_{k+1})^T \nabla f(\mathbf{x}_k) = 0.}$$

The new gradient is orthogonal to the previous search direction.

8.5 Steepest descent for quadratic functions

Consider the quadratic function

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A \mathbf{x} - \mathbf{b}^T \mathbf{x}.$$

The gradient is

$$\nabla f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}.$$

The steepest descent iteration becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k (A\mathbf{x}_k - \mathbf{b}).$$

The error is

$$\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}^*.$$

Since

$$A\mathbf{x}^* = \mathbf{b},$$

we obtain

$$\mathbf{e}_{k+1} = (I - \alpha_k A)\mathbf{e}_k.$$

The convergence therefore depends on the eigenvalues of A.

8.6 Geometric interpretation

Consider a quadratic function with elliptical contour lines.

The gradient is perpendicular to the contour line.

Therefore steepest descent moves directly across contour lines.

However, if the ellipse is very elongated, the method repeatedly overshoots from one side to the other.

This produces the characteristic zig-zag behaviour.

8.7 Example

Consider

$$f(x, y) = x^2 + 10y^2.$$

The gradient is

$$\nabla f = \begin{pmatrix} 2x \\ 20y \end{pmatrix}.$$

Starting from

$$(x_0, y_0) = (1, 1),$$

the steepest descent direction is

$$\mathbf{d}_0 = \begin{pmatrix} -2 \\ -20 \end{pmatrix}.$$

The large difference between the two components is caused by the different curvatures in the two directions.

The method therefore tends to oscillate.

8.8 Algorithm

Algorithm 3 Steepest descent method

Choose initial point $\mathbf{x}_0$
for $k = 0, 1, 2, \ldots$ **do**
  Compute gradient

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k)$$

  Choose step size α_k
  Update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$$

  if $\|\mathbf{g}_k\|$ is sufficiently small **then**
    stop
  end if
end for

8.9 Newton versus steepest descent

The two methods can be compared as follows.

	Steepest descent	Newton
Information	∇f	$\nabla f,\ H$
Cost per step	low	high
Convergence	linear	quadratic
Curvature	ignored	included

Steepest descent is cheap but may converge slowly.
Newton is expensive but can converge extremely rapidly.

8.10 Summary

The steepest descent method is based on the idea

$$\boxed{\text{move in the direction of maximum decrease.}}$$

The iteration is

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k).}$$

The method requires only gradients and is therefore suitable for very large problems.
Newton's method improves this idea by correcting the gradient direction using the Hessian.

9 Worked Examples

This section illustrates the main algorithms with explicit examples.

The purpose is not only to obtain the numerical answer, but also to understand the structure of the algorithms.

9.1 Example 1: Newton's method for a nonlinear system

Consider the system

$$\begin{cases} f(x,y) = x^2 + y^2 - 4 = 0, \\ g(x,y) = x - y = 0. \end{cases}$$

We write

$$F(x,y) = \begin{pmatrix} x^2 + y^2 - 4 \\ x - y \end{pmatrix}.$$

The Jacobian matrix is

$$J(x,y) = \begin{pmatrix} 2x & 2y \\ 1 & -1 \end{pmatrix}.$$

Newton's method requires solving

$$J(x_k, y_k) \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = -F(x_k, y_k).$$

The new approximation is

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix}.$$

First iteration

Choose

$$(x_0, y_0) = (1, 1).$$

Then

$$F(1,1) = \begin{pmatrix} -2 \\ 0 \end{pmatrix}.$$

The Jacobian is

$$J(1,1) = \begin{pmatrix} 2 & 2 \\ 1 & -1 \end{pmatrix}.$$

Hence

$$\begin{pmatrix} 2 & 2 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

Solving gives

$$\Delta x = \frac{1}{2}, \qquad \Delta y = \frac{1}{2}.$$

Therefore

$$(x_1, y_1) = (1.5, 1.5).$$

Repeating this procedure converges to

$$(x^*, y^*) = (\sqrt{2}, \sqrt{2}).$$

9.2 Example 2: Newton optimization

Consider the function

$$f(x, y) = x^2 + 2y^2 - 2x - 4y.$$

The gradient is

$$\nabla f = \begin{pmatrix} 2x - 2 \\ 4y - 4 \end{pmatrix},$$

and the Hessian is

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}.$$

The Newton step satisfies

$$H \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -\nabla f.$$

Therefore,

$$\begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} 2x - 2 \\ 4y - 4 \end{pmatrix}.$$

This gives

$$\Delta x = 1 - x,$$

and

$$\Delta y = 1 - y.$$

Hence

$$(x_{k+1}, y_{k+1}) = (1, 1)$$

for every initial point.
The minimum is reached after one Newton step.

9.3 Example 3: Steepest descent

Consider

$$f(x, y) = x^2 + 4y^2.$$

The gradient is

$$\nabla f = \begin{pmatrix} 2x \\ 8y \end{pmatrix}.$$

The steepest descent iteration is

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \alpha_k \begin{pmatrix} 2x_k \\ 8y_k \end{pmatrix}.$$

Choose

$$(x_0, y_0) = (1, 1)$$

and a fixed step size

$$\alpha = 0.1.$$

Then

$$\nabla f(1, 1) = \begin{pmatrix} 2 \\ 8 \end{pmatrix}.$$

The first step is

$$(x_1, y_1) = (1, 1) - 0.1(2, 8),$$

so

$$(x_1, y_1) = (0.8, 0.2).$$

The method continues by repeatedly moving opposite to the gradient.

9.4 Example 4: Newton versus steepest descent

Consider the quadratic function

$$f(x, y) = x^2 + 100y^2.$$

The contours are ellipses:

$$x^2 + 100y^2 = c.$$

The Hessian is

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 200 \end{pmatrix}.$$

The eigenvalues differ by a factor of

$$100.$$

This means the function has very different curvature in different directions.

Steepest descent

The gradient is

$$\nabla f = \begin{pmatrix} 2x \\ 200y \end{pmatrix}.$$

The large second component causes the method to overshoot in the y-direction. The trajectory therefore shows a zig-zag pattern.

Newton method

Newton uses

$$H^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/200 \end{pmatrix}.$$

The scaling difference is removed:

$$-H^{-1}\nabla f = -\begin{pmatrix} x \\ y \end{pmatrix}.$$

Therefore Newton moves directly towards the minimum.

9.5 Practical comparison

The examples show the main difference.

	Steepest descent	Newton
Gradient	✓	✓
Hessian	–	✓
Cheap step	✓	–
Fast near minimum	–	✓

9.6 Final perspective

All methods in these notes are based on approximating a nonlinear problem.

Taylor expansion
↓
local model
↓
solve simpler problem

For Newton's method:

linear approximation

is used for equations and

quadratic approximation

is used for optimization.
For steepest descent:

only the first-order information

is used.
This viewpoint unifies the methods.

10 Summary

This chapter summarizes the main ideas developed in these lectures.
The central theme is the use of local approximations to solve nonlinear problems.

10.1 The main idea

A smooth function can be approximated locally by its Taylor expansion.
For

$$f : \mathbb{R}^n \to \mathbb{R},$$

we have

$$f(\mathbf{x}+\mathbf{h}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{h} + \frac{1}{2}\mathbf{h}^T H(\mathbf{x})\mathbf{h}.$$

The three terms represent:

Quantity	Meaning
$f(\mathbf{x})$	value
$\nabla f(\mathbf{x})$	slope
$H(\mathbf{x})$	curvature

10.2 Optimization

A stationary point satisfies

$$\nabla f(\mathbf{x}^*) = 0.$$

The Hessian determines the local behaviour.

H	type of stationary point
positive definite	minimum
negative definite	maximum
indefinite	saddle point

10.3 Newton's method for equations

For

$$F(\mathbf{x}) = 0,$$

the Taylor approximation gives

$$F(\mathbf{x} + \mathbf{h}) \approx F(\mathbf{x}) + J(\mathbf{x})\mathbf{h}.$$

The Newton correction solves

$$\boxed{J(\mathbf{x}_k)\mathbf{h}_k = -F(\mathbf{x}_k).}$$

The iteration is

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k.}$$

10.4 Newton's method for optimization

For minimization,

$$\nabla f(\mathbf{x}) = 0.$$

Newton's equation becomes

$$\boxed{H(\mathbf{x}_k)\mathbf{h}_k = -\nabla f(\mathbf{x}_k).}$$

The update is

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k.}$$

Newton uses curvature information.

10.5 Steepest descent

The direction of maximum decrease is

$$-\nabla f(\mathbf{x}).$$

The iteration is

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k).}$$

The parameter α_k controls the step size.
Steepest descent is simple and robust, but may converge slowly.

10.6 Algorithm overview

Method	Information used	Main equation
Steepest descent	∇f	$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f$
Newton optimization	$\nabla f, H$	$H\mathbf{h} = -\nabla f$
Newton equations	J	$J\mathbf{h} = -F$