Chapter 1

Introduction to the course. Function classes for optimization.

Abstract:
We summarize first-order and second-order conditions for local minima, recall differential notation, discuss error sources in numerical differentiation, and collect basic convexity, strong convexity, and Lipschitz smoothness facts that will be used throughout the course.

1.1 Optimization problem

We will start our course with a study of unconstrained optimization problems of the form

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

with the practical goal of finding minima. Gradients and Hessians are denoted by

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \in \mathbb{R}^n, \qquad \nabla^2 f(x) = \left\{ \frac{\partial^2 f}{\partial x_i \partial x_j} \right\}_{i,j=1}^{n} \in \mathbb{R}^{n \times n}.$$

Many ML losses are highly nonconvex, with numerous sharp minima (see fig. 1.1). The sharp global minimum can make the final model sensitive to inputs and noise, while flatter local minima often yield more robust solutions even if their objective value is slightly higher. In general, finding a good local minimum can be more valuable than finding the global minimum. Moreover, ML problems are typically high-dimensional, which makes finding a global minimum very complicated, if it is even viable. Therefore, our first goal is not necessarily to rush toward the global minimum, but to find a good local minimum: this is often easier and can lead to better results. We start by reviewing basic optimality conditions for local minima.

1.2 Local optimality conditions

In many problems, the first step is to write down optimality conditions: in simple cases they can lead to a closed-form solution, and in iterative algorithms they often serve as

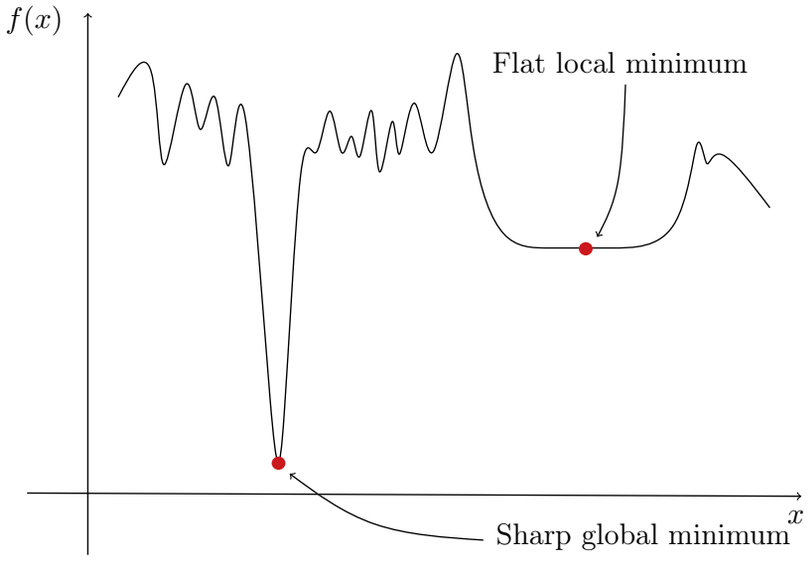

Figure 1.1: Rugged nonconvex loss with many minima and a sharp global minimum.

stopping criteria.

Definition 1.1 (Local minimum)**.** A point x is a (non-strict) local minimum of f if there exists $\varepsilon > 0$ such that

$$f(y) \geq f(x) \quad \forall y \in O_\varepsilon(x) := \{y : \|y - x\| < \varepsilon\}.$$

If the inequality is strict for all $y \neq x$ in $O_\varepsilon(x)$, then x is a strict local minimum.

Claim 1.1 (Necessary condition of a local minimum)**.** If $f \in C^1$, $x \in \operatorname{int}(\operatorname{dom} f)$, and x is a local minimum, then $\nabla f(x) = 0$. If in addition $f \in C^2$, then $\nabla^2 f(x) \succeq 0$.

Proof: Let $d \in \mathbb{R}^n$ be arbitrary and define the one-dimensional function

$$\varphi(\alpha) = f(x + \alpha d).$$

Since x is a local minimum and $x \in \operatorname{int}(\operatorname{dom} f)$, for sufficiently small $|\alpha|$ we have $x + \alpha d \in \operatorname{dom} f$ and $\varphi(\alpha) \geq \varphi(0)$, so $\alpha = 0$ is a local minimum of φ. Because $f \in C^1$, we can use a Taylor expansion:

$$\varphi(\alpha) = \varphi(0) + \varphi'(0)\alpha + o(\alpha) \geq \varphi(0) \overset{:\alpha}{\Longrightarrow} \varphi'(0) + o(1) \geq 0 \overset{\alpha \to 0}{\Longrightarrow} \varphi'(0) \geq 0$$

Hence, we have $\varphi'(0) = \nabla f(x)^\top d \geq 0 \, \forall d$. This means that if we take d and $-d$, the inequality can hold only if $\nabla f(x) = 0$. If in addition $f \in C^2$, then for any d we have the second-order expansion

$$f(x + \alpha d) = f(x) + \alpha \nabla f(x)^\top d + \frac{\alpha^2}{2} d^\top \nabla^2 f(x)\, d + o(\alpha^2).$$

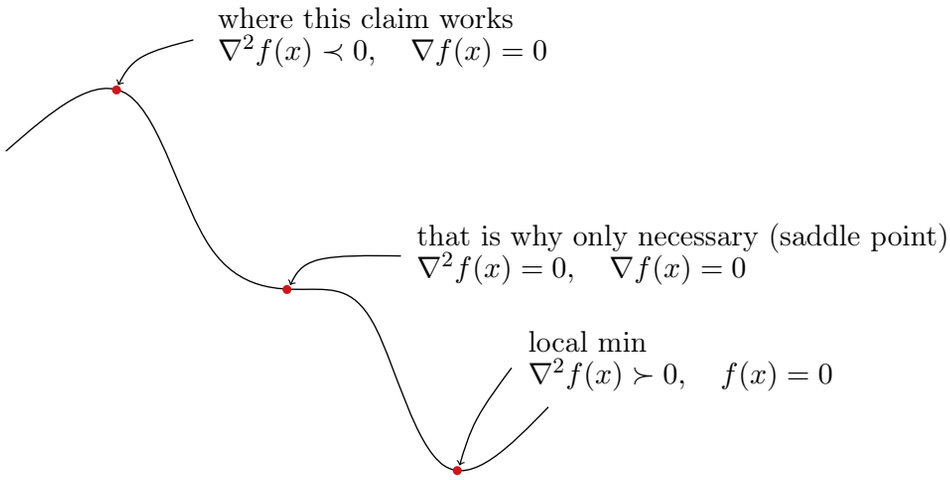

Figure 1.2: Explanation of why $\nabla^2 f \succeq 0$ and $\nabla f = 0$ are only necessary conditions.

Using $\nabla f(x) = 0$ and the same logic as above, we obtain that $d^\top \nabla^2 f(x)\, d \geq 0$ for all d, i.e. $\nabla^2 f(x) \succeq 0$. □

Why only necessary?
A stationary point with $\nabla f(x) = 0$ can be a local minimum, a local maximum, or a saddle point. The condition $\nabla^2 f(x) \succeq 0$ rules out strict local maxima, but still allows saddle points when the Hessian is only semidefinite (see Figure 1.2).

Claim 1.2 (Sufficient condition of a local minimum)**.** If $f \in C^2$, $x \in \operatorname{int}(\operatorname{dom} f)$, $\nabla f(x) = 0$, and $\nabla^2 f(x) \succ 0$, then x is a local minimum.

Proof: By the second-order Taylor expansion,

$$f(x+h) = f(x) + \nabla f(x)^\top h + \frac{1}{2} h^\top \nabla^2 f(x)\, h + o(\|h\|^2).$$

If $\nabla f(x) = 0$ and $\nabla^2 f(x) \succ 0$, then $h^\top \nabla^2 f(x)\, h > 0$ for all $h \neq 0$. Therefore, for sufficiently small $h \neq 0$ we have $f(x+h) > f(x)$. □

1.3 Differentials

Definition 1.2. The function f is differentiable at x if

$$f(x+h) - f(x) = \mathrm{d}f(x)[h] + o(\|h\|),$$

where $\mathrm{d}f(x)$ is a linear operator.

Definition 1.3. The second differential is

$$\mathrm{d}^2 f(x)\,[h_1, h_2] = \mathrm{d}\left(\mathrm{d}f(x)\,[h_1]\right)(x)\,[h_2]\,.$$

For a scalar function $f : \mathbb{R}^n \to \mathbb{R}$, the first differential is the scalar product with the gradient:

$$\mathrm{d}f(x)[h] = \nabla f(x)^\top h.$$

The second differential is a bilinear form linked to the Hessian:

$$\mathrm{d}^2 f(x)[h_1, h_2] = h_1^\top \nabla^2 f(x)\, h_2.$$

Example 1.1 (A worked example: $f(x) = \frac{1}{3}\|x\|_2^3$)**.** Let $f(x) = \frac{1}{3}\|x\|_2^3 = \frac{1}{3}(x^\top x)^{3/2}$. For $x \neq 0$,

$$\nabla f(x) = \|x\|_2\, x, \qquad \nabla^2 f(x) = \frac{1}{\|x\|_2} x x^\top + \|x\|_2 I.$$

1.4 Error analysis and machine precision

In the algorithms covered in this book, we use only real numbers. However, modern computers cannot store and manipulate arbitrary real numbers. Instead, they work only with a finite subset of numbers known as floating-point numbers. This means that computations produce only approximations, so we want these approximations to be as close to the theoretical values as possible. In floating-point arithmetic, a real number is stored in the form $\mathrm{fl}(x) = sM2^E$. Machine precision ε_m bounds the relative representation error:

$$\left|\frac{\mathrm{fl}(x) - x}{x}\right| \leq \varepsilon_m \qquad (x \neq 0).$$

Typical values:

bits	ε_m
64	10^{-16}
32	10^{-7}
16	10^{-3}

In practice, this means that when we plot an error versus the number of iterations, we should stop our algorithm once the error is on the order of ε_m, because improvements beyond that are below machine precision (see Figure 1.3).

1.5 Classes of functions

Unfortunately, Taylor expansions are useful only in a neighborhood of a point. To prove convergence of optimization algorithms, we therefore need to introduce common classes of functions whose defining properties hold globally rather than locally.

1.5.1 Convexity and strong convexity

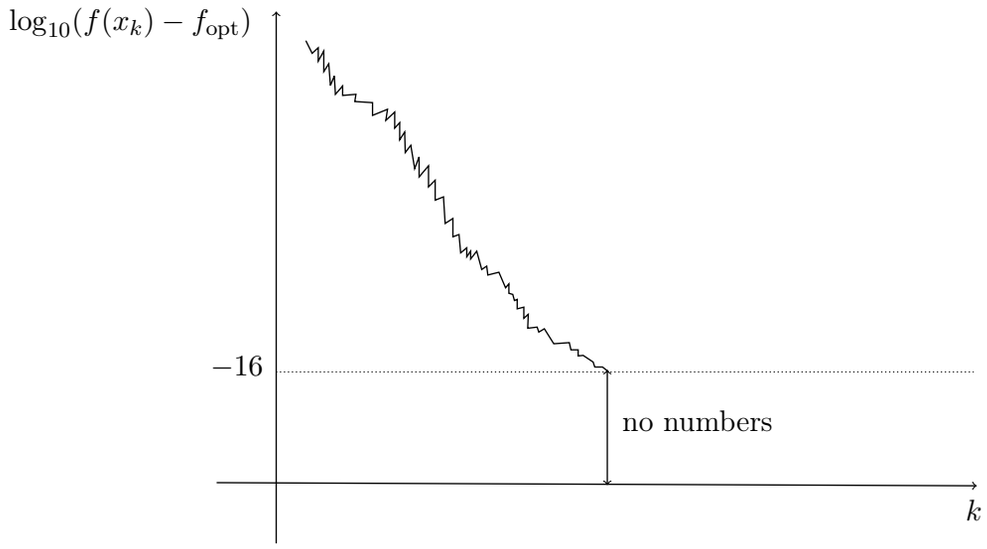

Figure 1.3: Finite-difference tradeoff in 64-bit computations.

Definition 1.4 (Convex function (see Figure 1.4))**.** A function f is convex if for all x, y and $\alpha \in [0, 1]$,

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y).$$

It is strictly convex if the inequality is strict for $x \neq y$ and $\alpha \in (0, 1)$.
Note: A function is concave if $-f$ is convex (strictly concave if $-f$ is strictly convex).

Claim 1.3 (Differential criteria of convexity)**.** Let $f \in C^1$. Then f is convex if and only if for all x, y,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x).$$

If $f \in C^2$, then f is convex if and only if $\nabla^2 f(x) \succeq 0$ for all x (see Figure 1.4).

Geometrically, this claim means that the graph of the function must be no lower than the tangent plane drawn to it at any point.

Definition 1.5 (μ-strongly convex)**.** The function f is μ-strongly convex if for all x the function $f(x) - \frac{\mu}{2}\|x\|^2$ is convex.

The geometric view can be better understood from the following claim.

Claim 1.4 (Differential criteria of strong convexity)**.** If $f \in C^1$, then f is μ-strongly convex if and only if for all x, y,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|x - y\|^2.$$

If $f \in C^2$, then f is μ-strongly convex if and only if $\nabla^2 f(x) \succeq \mu I \succ 0$ for all x.
Note: This is also equivalent to $\lambda_{min}(\nabla^2 f(x)) \geq \mu$.

From this claim, it is easy to see that if f is μ-strongly convex, then it is convex. From a geometric perspective, the class of μ-strongly convex functions includes convex functions that are not too flat and have curvature bounded below (see Figure 1.4).

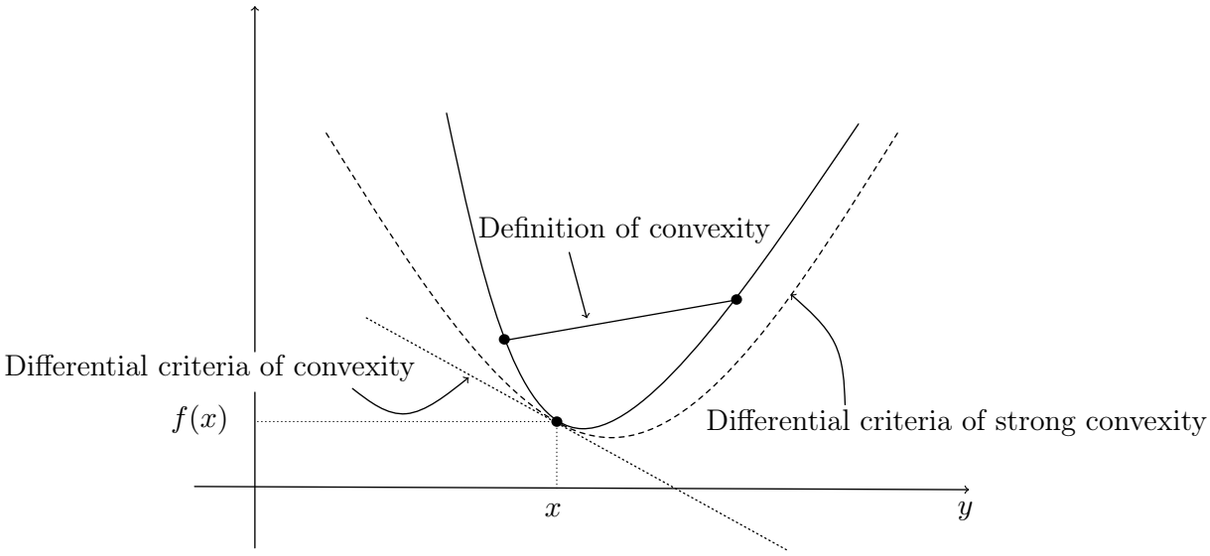

Figure 1.4: Kinds of convexity. The figure highlights the definitions and claims stated above. For the differential criterion of μ-strong convexity, the function $f(x)$ lies above a dashed quadratic lower bound. For the definition of convexity, the function lies below the solid chord. Finally, for the differential criterion of convexity, the function lies above the dotted tangent plane. All of these properties must hold for any x and y.

1.5.2 Lipschitz classes

Convexity describes the shape of f, but it does not quantify how rapidly the function (or its derivatives) can vary. For gradient-based methods, such quantitative control is crucial: it lets us bound Taylor remainders, obtain convergence rates. Lipschitz classes provide a compact way to encode these smoothness features.

Definition 1.6 (Lipschitz class $C_L^{k,m}$)**.** The function f belongs to the Lipschitz class $C_L^{k,m}$ if (i) $f \in C^k$, (ii) $m \leq k$, and (iii)

$$\|\nabla^m f(x) - \nabla^m f(y)\| \leq L\|x - y\| \quad \forall x, y.$$

Two special cases are used throughout the book: $C_L^{0,0}$ (Lipschitz continuity of the function values) and $C_L^{1,1}$ (Lipschitz continuity of the gradient), also known as L-smoothness. We now state some claims that we will use later; we omit the proofs (the first one is straightforward and the second one is more complicated).

Claim 1.5. If $f \in C_L^{0,0}$, then $|f(x) - f(y)| \leq L\|x - y\|$ for all x, y, equivalently

$$f(y) - L\|x - y\| \leq f(x) \leq f(y) + L\|x - y\|.$$

Claim 1.6. If $f \in C_L^{1,1}$, then

$$f(x) \geq f(y) + \nabla f(y)^\top (x - y) - \frac{L}{2}\|x - y\|^2,$$

$$f(x) \leq f(y) + \nabla f(y)^\top (x - y) + \frac{L}{2}\|x - y\|^2.$$

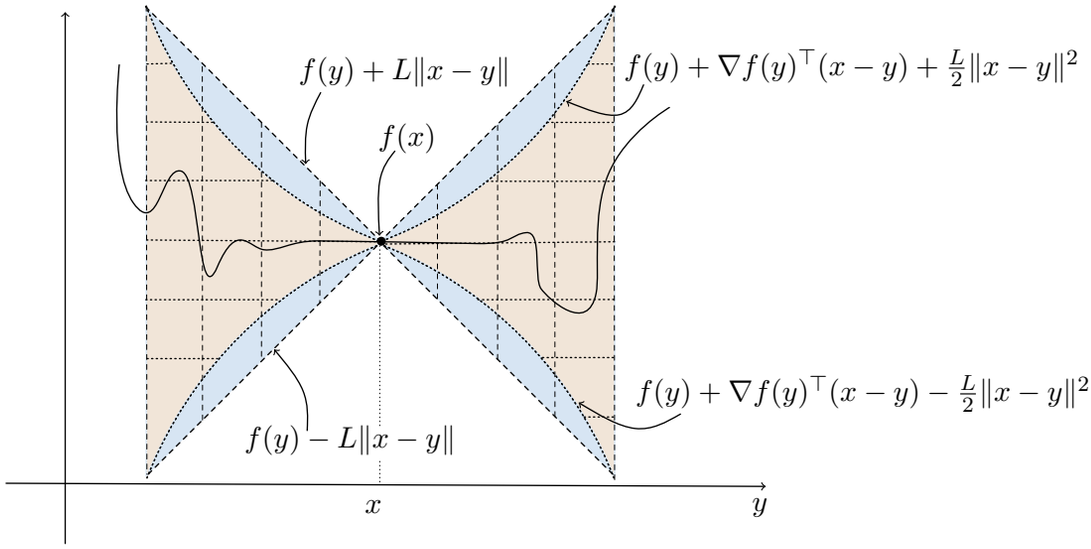

Figure 1.5: Examples of Lipschitz classes. The solid line corresponds to the function. The dashed region is the space of $C_L^{0,0}$ (blue), and the dotted region is the space of $C_L^{1,1}$ (orange).

How do we identify a Lipschitz class? One way to do it is using the definition, but the following claim simplifies this process.

Claim 1.7. $f \in C_L^{k,m-1} \iff \|\nabla^m f(x)\| \leq L.$

It can also be proved easily using the mean value theorem. Do it if you are an enthusiast!

1.6 Finite differentiation

Optimization algorithms often rely on gradients, which we sometimes compute analytically. Still, many of us have been in the situation where a complicated derivative turns out to be incorrect after spending 30 minutes (or more) deriving it. How can we check correctness when there is no answer at the end of the textbook? Finite-difference methods provide a simple sanity check; below we discuss several common schemes. At a point x, we sample a random direction d and compare the implemented directional derivative $\nabla f(x)^\top d$ with a numerical difference quotient in the same direction. We denote

$$g := \nabla f(x)^\top d,$$

and approximate g with step size ε in floating-point arithmetic. We write $\mathrm{fl}(\cdot)$ for computed quantities and use the model

$$|\mathrm{fl}(z) - z| \le |z|\,\varepsilon_m.$$

The key point is that we cannot simply take the smallest possible ε: in finite differences too small step amplifies roundoff errors (subtractive-cancellation effects), so ε must balance truncation and roundoff errors.

Forward difference. The Taylor expansion gives

$$f(x + \varepsilon d) = f(x) + g\,\varepsilon + O(\varepsilon^2),$$

so

$$g = \frac{f(x + \varepsilon d) - f(x)}{\varepsilon} + O(\varepsilon).$$

Define the computed estimator

$$\widehat{g}_{\mathrm{fd}}(\varepsilon) = \mathrm{fl}\left(\frac{f(x + \varepsilon d) - f(x)}{\varepsilon}\right).$$

Then

$$|g - \widehat{g}_{\mathrm{fd}}(\varepsilon)| \le |g - \tfrac{f(x+\varepsilon d)-f(x)}{\varepsilon}| + \left|\tfrac{f(x+\varepsilon d)-f(x)}{\varepsilon} - \widehat{g}_{\mathrm{fd}}(\varepsilon)\right|.$$

Assume $|f(\cdot)| \le L_0$ in a neighborhood of x, and the truncation error satisfies

$$\left|g - \tfrac{f(x+\varepsilon d)-f(x)}{\varepsilon}\right| \le L_2\,\varepsilon.$$

Then the roundoff term is bounded by

$$\left|\tfrac{f(x+\varepsilon d)-f(x)}{\varepsilon} - \widehat{g}_{\mathrm{fd}}(\varepsilon)\right| \le \varepsilon_m\,\tfrac{|f(x+\varepsilon d)|+|f(x)|}{\varepsilon} \le \tfrac{2L_0\,\varepsilon_m}{\varepsilon}.$$

Therefore

$$|g - \widehat{g}_{\mathrm{fd}}(\varepsilon)| \le L_2\,\varepsilon + \tfrac{2L_0\,\varepsilon_m}{\varepsilon}.$$

This shows that taking ε too small increases the error due to roundoff. Minimizing the bound gives

$$\varepsilon_{\mathrm{opt}} = \sqrt{\tfrac{2L_0\,\varepsilon_m}{L_2}} \approx \sqrt{\varepsilon_m}.$$

Central difference. Using the expansions

$$f(x + \varepsilon d) = f(x) + g\,\varepsilon + \tfrac{1}{2}d^\top \nabla^2 f(x) d\,\varepsilon^2 + O(\varepsilon^3),$$
$$f(x - \varepsilon d) = f(x) - g\,\varepsilon + \tfrac{1}{2}d^\top \nabla^2 f(x) d\,\varepsilon^2 + O(\varepsilon^3),$$

we get

$$g = \frac{f(x + \varepsilon d) - f(x - \varepsilon d)}{2\varepsilon} + O(\varepsilon^2).$$

With the computed estimator

$$\widehat{g}_{\mathrm{cd}}(\varepsilon) = \mathrm{fl}\left(\frac{f(x + \varepsilon d) - f(x - \varepsilon d)}{2\varepsilon}\right),$$

and assuming $|f(\cdot)| \le L_0$ in a neighborhood of x and a truncation bound $|O(\varepsilon^2)| \le L_3\varepsilon^2$, we obtain

$$|g - \widehat{g}_{\mathrm{cd}}(\varepsilon)| \le L_3\,\varepsilon^2 + \tfrac{L_0\,\varepsilon_m}{\varepsilon}.$$

Minimizing yields

$$\varepsilon_{\mathrm{opt}} = \sqrt[3]{\tfrac{L_0\,\varepsilon_m}{2L_3}} \approx \sqrt[3]{\varepsilon_m}.$$

Complex step. If f is an analytic function, then

$$f(x + i\varepsilon d) = f(x) + ig\,\varepsilon + O(\varepsilon^2) + iO(\varepsilon^3),$$

so

$$g = \frac{\operatorname{Im} f(x + i\varepsilon d)}{\varepsilon} + O(\varepsilon^2).$$

The previous schemes can be unstable because they subtract nearly equal numbers ($a - b$ with $a \approx b$) in floating-point arithmetic. This method avoids that subtraction, so we can take $\varepsilon_{\text{opt}} = \varepsilon_m$. This makes the scheme the most accurate.