Ladislav Kavan's Physically Based Simulation 2017ed

Edited by wxgopher

May 11, 2019

Contents

This lecture note is based on Dr. Ladislav Kavan's Physics-based Animation course CS6660, originally taught at the University of Utah. I also try to include some supplementary materials from other resources.

Warning: Although as I strive to make this material useful, there are certain bugs, use this material at your own risk. I would also be grateful to hear feedbacks.

1 Classical Mechanics

1.1 Basics: a harmonic oscillator

2 Time Integration

3 Optimization

3.1 Implicit Newmark

$$x_{n+1} = x_n + \frac{h}{2}(v_n + v_{n+1}), \tag{1a}$$

$$v_{n+1} = v_n = \frac{h}{2}(f_n + f_{n+1}), \tag{1b}$$

where x and v are positions and velocities, and $f_n \equiv f(x_n)$. Thus we have

$$x_{n+1} - x_n - hv_n = \frac{h^2}{4}(f_n + f_{n+1}). \tag{2}$$

Let $y = x_n + h_n v_n + \frac{h^2}{4} f_n$, $x_{n+1} = x$, note that $\nabla E = -f$, then (2) becomes

$$x - y = \frac{h^2}{4} f \Rightarrow x - y + \frac{h^2}{4} f \nabla E = 0. \tag{3}$$

Let $g(x) = \frac{1}{2}\|x - y|^2 + \frac{h^2}{4}E$, then solving g equals to solve x for

$$\min_x \quad g(x). \tag{4}$$

3.2 Optimization Problems

Problem formulation:

$$\min_x \quad g(x), \quad x \in R^n, \quad g \in R. \tag{5}$$

Optimization problems can be categorized into constrained or unconstrained problems, or convex or non-convex problems.

Theorem 1. *For a convex problem (where both objective and feasible set are convex), if the objective is* C^2*, then the Hessian* $H \succeq 0$*, and the local minimum is the global minimum.*

Definition 1. *A **linear programming (LP)** is a problem with linear objective and linear equality or inequality constraints. A **quadratic programming (QP)** is* **the same as LP** *except with a quadratic objective.*

Note 1. *Convex QP has polynomial time solver but non-convex QP is NP-hard.*

Example 1. *A non-convex QP:*

$$\min_x \quad \frac{1}{2}\|Ax\|^2, \tag{6a}$$
$$s.t. \quad \|x\|_2 = 1. \tag{6b}$$

Note 2. *Software package for solving non-convex problem:* IpOPT, KNITRO, *or NEOS-Guide*

3.2.1 Solving unconstrained problems

There are two ways to solve an unconstrained problem: descent method or trust-region method.

Descent method refers to pick a descent direction d and do an exact or inexact line search (LS) to determine descent distance αd.

Definition 2. ***Descent direction:*** $\forall \alpha \in (0, \alpha_0)$, $\alpha_0 > 0$, $g(x + \alpha d) < g(x)$.

Definition 3. ***Exact LS*** *refers to solving the following problem:*

$$\arg\min_{\alpha > 0} g(x + \alpha d).$$

Backtracking LS

3.2.2 Newton's method

A usual descent direction is called **gradient descent** (GD), denoted by $d = -\nabla g$. It doesn't work well sometime, for example:

Example 2.

$$\min_{x_1, x_2} g(x1, x2) = \frac{1}{2}(x_1^2 + \gamma x_2^2), \quad \gamma > 0. \tag{7}$$

If we do exact LS, we get

$$x_1^k = \gamma(\frac{\gamma - 1}{\gamma + 1})^k, \tag{8a}$$
$$x_2^k = (-\frac{\gamma - 1}{\gamma + 1})^k. \tag{8b}$$

If gamma becomes huge, we can tell GD converges slowly (as two GD directions are perpendicular).

Newton's method refers to do a quadratic approximation to problem (5), given by

$$g(x + p) = g(x) + \nabla g^{\mathrm{T}}(x)p + \frac{1}{2}p^{\mathrm{T}}\nabla^2 g(x)p + O(\|p\|^2) \equiv \tilde{g}(x) + O. \tag{9}$$

Thus we have

$$\frac{\partial \tilde{g}}{\partial p} = 0 \Longleftrightarrow \nabla g + \nabla^2 g p = 0.$$

The **Newton descent direction** is given by

$$p = -(\nabla^2 g)^{-1}\nabla g. \tag{10}$$

Note 3. *Newton's method is affine invariant. Let $x = Ty$ with T being a nonsingular matrix, so problem (5) is equivalent to problem*

$$\min_y \tilde{g}(y). \tag{11}$$

We also have

$$\nabla\tilde{g} = T^2\nabla g, \quad \nabla^2\tilde{g} = T^{\mathrm{T}}\nabla^2 g T.$$

Newton direction Δy of $\tilde{g}$ and Δx of g is given by

$$\Delta y = -T^{-1}\nabla^2 g\nabla g = T^{-1}\Delta x.$$

Definition 4. ***Newton decrement**: substitute Newton descent direction (10) into (9), we get the **Newton decrement***

$$\lambda^2 = g - \frac{1}{2}\nabla g^{\mathrm{T}}(\nabla^2 g)^{-1}\nabla g. \tag{12}$$

A stopping criteria for Newton's method would be

$$\frac{\lambda^2}{2} \leq \epsilon.$$

Note that Newton decrement is also affine invariant.

Convergence analysis of the Newton's method is given as follows:

Theorem 2. ***Kantorovich**: if g is strictly convex and $\nabla^2 g$ is Lipschitz continuous, then $\exists\epsilon > 0, \eta > 0, \gamma > 0$, s.t.*

Damped phase I *If $\|\nabla g\| \geq \eta$, then $\|g^{k+1}\| \leq \|g^k\| - \gamma$;*

Quadratic phase II *If $\|\nabla g\| < \eta$, then $\|g^{k+1}\| \leq c\|g^k\|^2$.*

Some cases where Newton's method doesn't work well:

1. If g is highly nonlinear, the damped phase will be long;
2. If g is non-convex, solution might not converge to a minimum.

In practical implementations, the Hessian might not always be positive definite. A possible Hessian modification is:

Note 4. ***Hessian modification**: Consider $A = \nabla^2 g + cI$, where $c > 0$. It is shown that if $\nabla^2 g = Q\Lambda Q^{\mathrm{T}}$ ($\nabla^2 g$ is symmetric, hence it is always diagonalizable), then $A = Q(\Lambda + cI)Q^{\mathrm{T}}$, hence A is positive definite given sufficiently large $c > 0$.*

Solving Newton's direction (10) refers to solving a linear system. We introduce several techniques to tackle this.

3.3 Numerical Linear Algebra

We consider solving a linear system $Ax = b$.

3.3.1 Direct solvers

We can prefactor A into $A = UV^{\mathrm{T}}$, with U and V being low rank matrices.

Common factorization methods are

LU If A is nonsingular, then there exist L, P, U, where they are lower-triangular matrix, permutation matrix (hence orthogonal), and upper-triangular matrix, respectively, s.t. $A = PLU$.

Sparse LU $A = P_1 LUP_2$, where P_1 and P_2 are permutations to utilize sparse information of A.

Cholesky For symmetric positive-definite matrix, we have $A = LL^{\mathrm{T}}$ with unique L. Note that for positive semi-definite matrix, Cholesky is not unique.

LDLT

3.3.2 Iterative solvers

Iterative solvers includes:

1. Classical method: Jacobian, Gauss-Seidel, SOR (super over-relaxation), etc.;
2. Krylov subspace method: CG (conjugate gradient), GMRES (generalized minimal residual), etc.;
3. multigrid method.

Classical method Suppose P is easy to invert, let $A = P + A - P$, then

$$Ax = b \quad \Leftrightarrow \quad Px = (P - A)x + b.$$

$P = diag(A)$ gives us Jacobian method, and it's GPU-friendly; $P = lower(A)$ gives us GS method; $P = diag(A) + w \cdot lower(A)$ gives us SOR method where $w < 2$.

The residual term follows

$$r_k \equiv A(x^\star - x_k) = b - Ax_k \quad \Leftrightarrow \quad r_{k+1} = (I - P^{-1}A)r_k.$$

To make classical method converge, we need $\rho(I - P^{-1}A) < 1$, where $\rho(\cdot)$ is the spectrum radius.

Krylov subpace method

Note 5. *A good reference on CG:* `An Introduction to the Conjugate Gradient Method Without the Agonizing Pain` *by J. Shewchuk.*

Assuming $A^{\mathrm{T}} = A$ and $A \succeq 0$.

4 Elastic Materials and Finite Element Simulation

5 Something more...

6 Papers and books

1. Nonlinear Continuum Mechanics for Finite Element Analysis by J. Bonet
2. Convex optimization by S. Boyd
3. Numerical optimization by Nocedal