A Simple Hint For Final Project Of Numerical Analysis.

Shuang Hu

School of Mathematical Science,
Zhejiang University

2022.11.24

Content

- How to grade this project assignment?
- Something about two previous programming assignments.
- A simple design for spline-fitting problem.
- How to achieve a better design?

Grading Scale

The total points of this final project is 100, and two additional requirements weigh 20 points (10 points each). And if your assignment is good enough, we will give 10% bonus for you.
Here are some grading scales for this project:

- Compile Error: 0 points.
- Wrong answer in all test cases: 30 points.
- Implement the S_0^1 interpolation formula: at least 60 points.
- Pass all test cases: at least 80 points.
- Pass all test cases and give enough discussions: at least 90 points.
- Pass all test cases with a good design: 100 points.
- Without report file: -10 points.
- Without makefile/CMake to build your project: -10 points.
- Without test files: -10 points.
- Shortcomings in your design: -1 point each one, at most 10 points.

Typical design shortcomings

In the past two programming assignments, we discover some typical design shortcomings:

- Using unnecessary global variables and functions.
- Abuse `public` elements.
- Memory leak.
- Unsuitable input and output formats.
- Incorrect interface design.
- Some misunderstanding of combination and inheritance.

How to avoid these design shortcomings?

Signature

First, we should clarify the **signature** of spline-fitting problem.

S_1^0 spline:

- Input: $\{(x_i, y_i)\}$.
- Precondition: $\forall i \neq j$, $x_i \neq x_j$.
- Output: Piecewise linear function $y = s(x)$.
- Postcondition: $s(x_i) = y_i$, $s|_{(x_i, x_{i+1})}$ is a linear function.

S_3^2 spline:

- Input: $\{(x_i, y_i)\}$, boundary condition type, boundary condition c_1, c_2.
- Precondition: $\forall i \neq j$, $x_i \neq x_j$.
- Output: piecewise cubic function $y = s(x)$.
- Postcondition: $s(x_i) = y_i$, $s|_{(x_i, x_{i+1})}$ is a cubic function, $s \in C^2$, s satisfies the boundary conditions.

Data Structures

Programming = data structure + algorithm. Now, we should design some suitable ADT(abstract data type) to solve this problem.
Input: Some discrete points in 2-dimensional vector space. So we can design `class Point`.
Output: Piecewise polynomials. Based on `class Polynomial` in the second assignment, we can design `class PiecewisePoly`.
BSpline approximation process: We should design a class to store `BSpline Basis.`
Linear System: We should design or choose classes to store the information of linear systems.
How to store the extra interpolation basis for S_3^2 fitting problem.
...

Algorithms

We should implement spline-fitting algorithms on the above data structures.

- Combine the polynomial pieces with piecewise polynomials.
- Operators on piecewise polynomials.
- Generate Bspline basis recursively.
- Generate the linear system for PP-form.
- Generate the linear system for the coefficients of Bspline.
- Deal with the boundary conditions.
- Solve the linear system.
- $\cdots$

Discussions

Here are some discussions about our design:

- How to store the sparse matrix?
- How to choose the linear equation solver?
- If we choose n as template parameter, how to generate Bspline basis of S_n^{n-1} recursively by meta-programming?
- How to reuse the spline-fitting algorithm on the problem of curve-fitting?
- (*) If we need arc length parameter for our fitted curve, how to design the curve-fitting algorithm?
- How to test the convergent rates?
- $\cdots$

Better design

To make your design better, you can:

1. Make sure you understand the lecture notes.
2. Write your design document.
3. Read it again and again, focus on each data structures and algorithms.
4. Clarify the relationships of each different parts.
5. Coding.
6. Test each parts and debug.
7. Test and use your final cplusplus package.

Enjoy Your Programming!