Лабораторная работа 3.3

Сервис PInvoke. Пакеты сплайн-интерполяции и нелинейной оптимизации математической библиотеки Intel MKL

В лабораторной работе 3.3 для первой компоненты векторного поля из класса **VDataArray** (из лабораторной работы 2) с помощью методов, реализованных в библиотеке **Intel MKL**, строится сглаживающий кубический сплайн.

В классе **VDataArray** дискретные значения векторного поля хранятся в виде массива значений координаты и массива со значениями компонент поля. Обозначим через массив значений первой компоненты, для которой будет строиться сглаживающий кубический сплайн. Предполагается, что массив узлов исходной сетки из **VDataArray** не содержит равных значений и упорядочен в порядке возрастания. Таким образом, все значения координат находятся на отрезке .

Сглаживающий кубический сплайн строится на равномерной сетке с заданным числом узлов. Число узлов сглаживающего кубического сплайна равно . Концы отрезка являются узлами сглаживающего сплайна. Обозначим множество узлов сглаживающего сплайна через . Для кубического сплайна задаются свободные граничные условия (вторая производная сплайна на концах отрезка равна нулю). Определим квадрат нормы невязки как , т.е. как сумму квадратов разности значений сглаживающего сплайна и заданных значений первой компоненты поля в узлах исходной сетки на которой заданы значения поля. Сглаживающий сплайн будет строиться из условия, что норма невязки принимает минимальное значение.

При построении сглаживающего кубического сплайна надо использовать методы из пакетов сплайн-интерполяции и нелинейной оптимизации математической библиотеки **Intel MKL.** При численном решении задачи минимизации неизвестными являются значения, которые сплайн принимает в узлах .

В среде VisualStudio надо cоздать решение (solution) с двумя проектами:

* тип одного проекта – консольное приложение **C#**;
* тип второго проекта – Dll-библиотека **C++**.

В Dll−библиотеке **C++** находится глобальная функция, которая вызывается из кода **C#** с использованием механизма **PInvoke.** Из этой глобальной функции вызываются функции из библиотеки **Intel MKL**.

В коде C# надо определить cтруктуру **SplineDataItem** и класс **SplineData**.Структура **SplineDataItem** содержит данные для одного узла сетки, на которой заданы значения поля. Класс **SplineData** содержит входные данные и результаты сплайн-аппроксимации.

В структуре **SplineDataItem** надо определить следующие открытые автореализуемые свойства

* типа **double** с координатой узла , в котором задано значение компоненты поля;
* два свойства типа **double** для заданного значения в этом узле и вычисленного значения сглаживающего сплайна.

Структура **SplineDataItem** содержит открытые

* конструктор с тремя параметрами типа **double** для инициализации всех автореализуемых свойств структуры;
* метод **string ToString(string format),** возвращающий строку, которая содержит значения всех свойств структуры**;** параметр **format** задает формат вывода чисел с плавающей запятой;
* перегруженную (**override**) версию виртуального метода **string ToString().**

Класс **SplineData** содержит входные данные для сплайн-аппроксимации, метод, в котором для построения сглаживающего сплайна вызываются функции из библиотеки **Intel** **MKL**, и результаты сплайн-аппроксимации. Класс **SplineData** содержит ссылку на класс **VDataArray**. В классе **SplineData** определяются поля и свойства, необходимые для построения и хранения результатов сплайн-аппроксимации.

В классе **SplineData** надо определить следующие открытые автореализуемые свойства

* типа **VDataArray** дляссылки на объект **VDataArray**;
* типа **int** для числа узлов равномерной сетки, которые являются узлами сглаживающего сплайна;
* свойство типа **double[]** для массива значений сглаживающего кубического сплайна в узлах сетки , на которой заданы значения поля;
* типа **int** для максимального числа итераций в процессе решения задачи минимизации невязки;
* типа **int** для информации о причине остановки итераций при решении задачи минимизации невязки;
* типа **double** для минимального значения невязки;
* свойство типа **List<SplineDataItem>** с результатами сплайн-аппроксимации, которые хранятся в виде коллекции элементов **SplineDataItem**.

В классе **SplineData** надо определить открытый конструктор с параметрами

* типа **VDataArray** для ссылки на объект **VDataArray**;
* типа **int** для числа узлов равномерной сетки, которые являются узлами сглаживающего сплайна;
* типа **int** для максимального числа итераций в процессе решения задачи минимизации.

В классе **SplineData** определить открытые методы

* метод, в котором вызывается глобальная функция изDll−библиотеки **C++**.Из глобальной функции **C++** вызываются функции из пакетов сплайн-интерполяции и нелинейной оптимизации математической библиотеки **Intel MKL.** Через параметры глобальной функции из кода **C#** передаются данные, которые необходимы для построения сглаживающего сплайна. Через параметры глобальной функции в код **C#** возвращается информация о причине остановки итерационного процесса и результаты сплайн-аппроксимации.
* метод **ToLongString(string format)**, который формирует строку, содержащую
* данные объекта **VDataArray** в виде строки, полученной при вызове метода **ToLongString(string format)** класса **VDataArray**;
* результаты сплайн-аппроксимации для каждого узла сетки из **VDataArray** ‒ координату узла, заданное значение первой компоненты поля в этом узле и вычисленное значение сглаживающего сплайна;
* минимальное значение невязки;
* информацию о причине остановки итераций при решении задачи минимизации невязки;
* число итераций, выполненных в процессе решения задачи минимизации.
* метод **Save (string filename, string format)**, который записывает в файл **filename** строку, сформированную в методе **ToLongString(string format)** (в текстовом формате);

Отладка программы

В процессе вычисления сглаживающего сплайна будет многократно вызываться метод для построения интерполяционного кубического сплайна из библиотеки Intel MKL. Необходимо сначала отладить эту часть программы. Надо проверить, что построенный сплайн является интерполяционным, то есть проходит через точки и значения второй производной на концах отрезка равны нулю.

Для отладки всей программы в методе **Main()** надосоздать объект **VDataArray**,создать объект **SplineData** с небольшим числом узлов в сглаживающем сплайне**,**  вызвать метод, вычисляющий сглаживающий сплайн, и сохранить результаты вычислений в файле.

**Срок сдачи лабораторной работы** **20 декабря (для всех групп).**

Решение с лабораторной работой надо прислать мне на почту, не только исполняемые файлы, но и исходные коды C# и C++. Можно выложить на GitHub и прислать мне ссылку. Можно разместить в вашем облаке и прислать ссылку. Можно упаковать все решение и переслать как вложение в письмо.

**Последний день для отправки лабораторной работы 20 декабря (для всех групп).**