

Лабораторная работа №3 «Интерполяция эрмитовыми кубическими сплайнами»

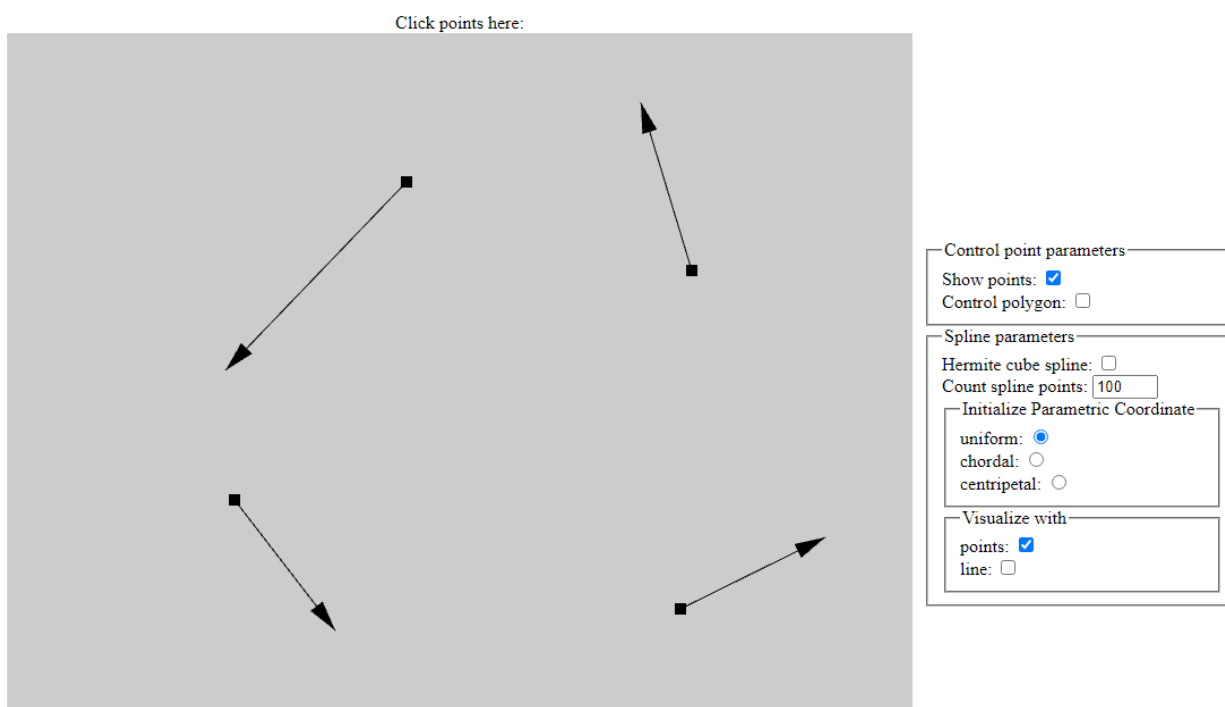
Цель работы: изучение методов интерполяции эрмитовыми кубическими сплайнами.

Задание: по заданным двумерным координатам контрольных точек и касательных векторов построить эрмитов кубический сплайн. Ввод координат контрольных точек и касательных векторов осуществляется мышью в окне браузера. Результатом работы программы должен являться вывод точек эрмитового кубического сплайна с заданным шагом.

Продолжительность: 2 часа.

Исходные данные: шаблон программы, написанный на языке JavaScript с использованием библиотеки WebGL. Файлы содержатся в архиве 3.zip.

Описание. Шаблон программы содержит функции, позволяющие пользователю вводить координаты контрольных точек и касательных векторов в окне браузера и изменять их при помощи мыши (изменение длин и направлений касательных векторов осуществляется путем перемещения концов стрелок векторов):



Флаг «Hermite cube spline» позволяет запустить расчет эрмитового кубического сплайна и вывести полученные точки с заданным шагом (расчитывается исходя из количества точек, задаваемого параметром «Count spline points»).

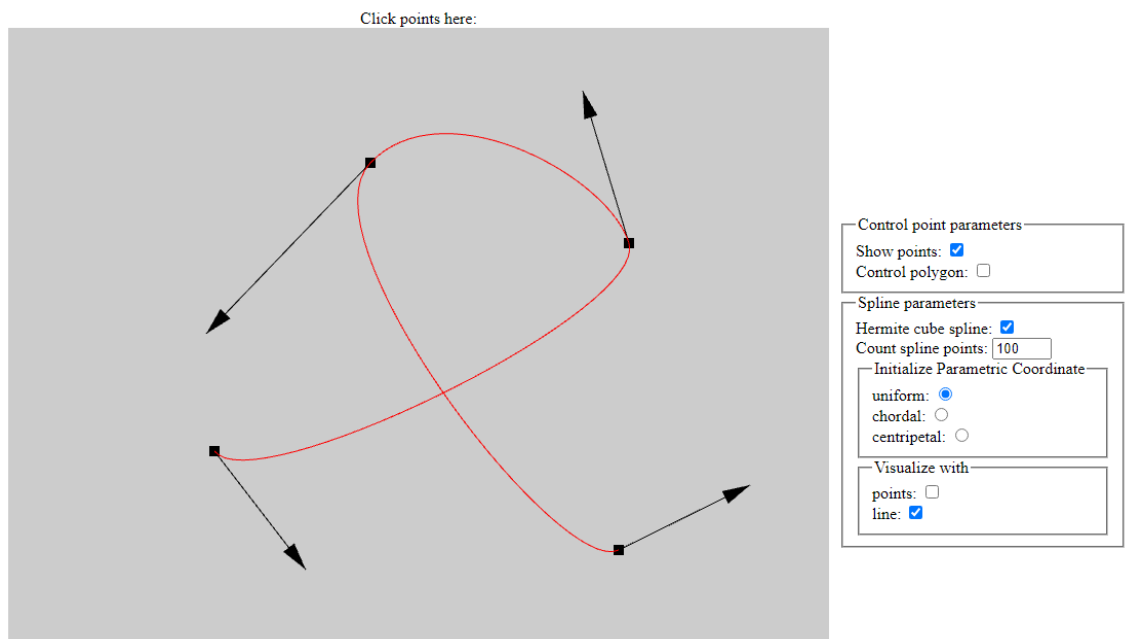
Флаг «Control polygon» позволяет провести ломанную линию через контрольные точки.

Примечание.

В шаблоне комментарии, написанные прописными буквами, указывают места, в которых нужно добавить свой код.

Последовательность выполнения работы:

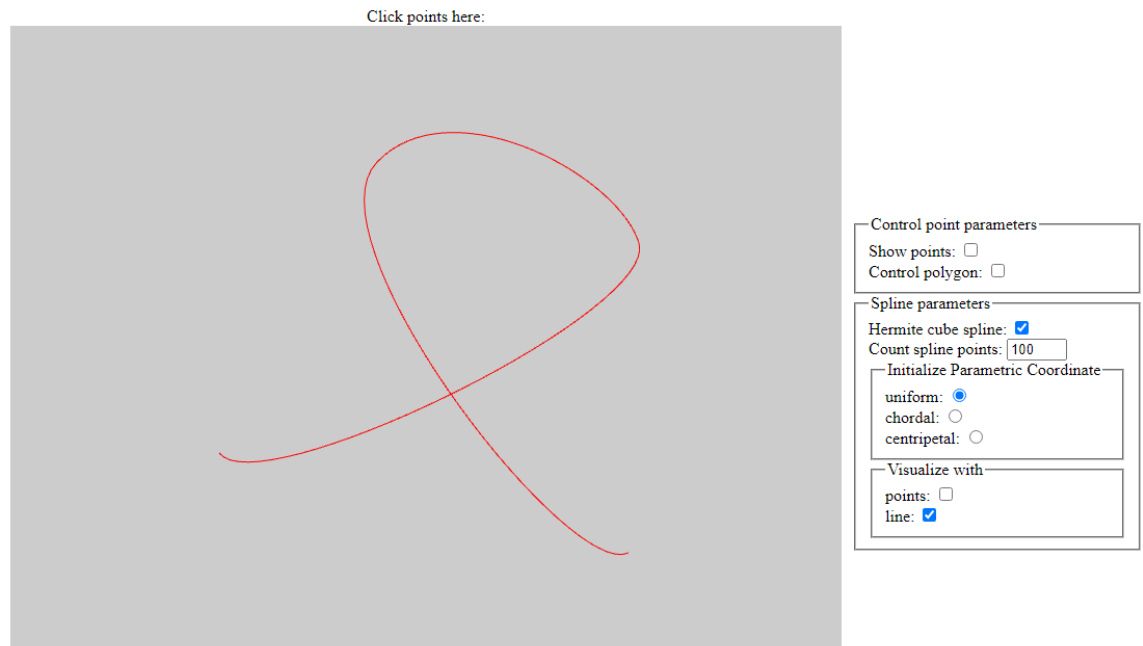
1. Открыть в браузере файл 3.html, убедиться, что описанные функции для создания и изменения координат контрольных точек и касательных векторов, а также опции формы работают.
2. Открыть файл 3.js для редактирования.
3. В класс Point добавить новое свойство – параметрическую координату t . Инициализировать ее значением по умолчанию (например, 0).
4. Добавить инициализацию параметрической координаты t для созданных контрольных точек в методе calculateHermiteSpline класса Data. Принцип ее инициализации полностью идентичен инициализации для линейной интерполяции, описанной в лабораторной работе №1 («uniform», «chordal» и «centripetal»). Для управления натяжением кривой, можно изменять диапазон изменения параметрической координаты t , например, умножить его на некоторый коэффициент $k > 1$.
5. В методе calculateHermiteSpline класса Data требуется написать код расчета координат точек эрмитового кубического сплайна pt по известным массивам контрольных точек `pointsCtr` и касательных векторов `mCtr`.
6. В результате должна получиться примерно следующая картина:



Сплайн показан красным цветом. Шаг равномерный (100 разбиений). Параметр $t \in [0, n]$, где n – количество контрольных точек минус 1, то есть:

```
k = this.pointsCtr.length-1.
```

С помощью флага “Show points” можно скрыть показ контрольных точек и векторов касательных для просмотра окончательного вида кривой:



7. Реализовать эффективную формулу для практического вычисления эрмита сплайна. Получить аналогичный результат.