

ЛАБОРАТОРНАЯ РАБОТА №2

Работа с графикой

ВАРИАНТ 1

В этом задании символьные вычисления **не используются**.

Двумерная графика, часть первая.

Написать функцию `compareInterp(x, xx, f)`, которая принимает на вход две сетки, `xx` (более мелкую) и `x` (более крупную), $x \subset xx$, и указатель на функцию f (`function handle`). Эта функция рисует графики f на сетке `xx` и графики функций, получающихся интерполированием f с сетки `x` на сетку `xx` различными методами (флаги команды `interp1: nearest, linear, spline, cubic`). График оборудовать легендой, а также заголовком с наименованием метода интерполяции.

1 [1]. Написать функцию `compareInterp(x, xx, f)` в соответствии с Требованиями к Написанию Программ Практикума.

2 [2]. Подобрать набор функций, на котором продемонстрировать преимущества и недостатки каждого метода интерполирования (хотя бы 4 нетривиальных примера).

3 [2]. Оценить априорную погрешность интерполирования методом ближайшего соседа. Для построения оценки считать известными максимумы производных. Построить график априорной и получившейся погрешности для двух функций: на одной с большим отклонением от априорной погрешности, на другом — с малым.

Двумерная графика, часть вторая.

4 [1]. Написать функцию `convergenceFunc(fn, f, a, b, n, convType)`, принимающую на вход аргументы: функцию `fn(n, x)`, такую, что $fn(n, x) = f_n(x)$, и функцию f , считающуюся пределом последовательности $f_n(x)$ на $[a, b]$ в смысле, задаваемом аргументом-строкой `convType`: это может быть поточечная сходимость, равномерная сходимость, среднеквадратичная сходимость. Функция рисует анимацию из n кадров, на каждом i -м из которых нарисованы f_i и f на отрезке $[a, b]$. В заголовке графика стоит вывести значения метрики разности для всех сходимостей, кроме поточечной. Подготовить несколько примеров, когда есть один вид сходимости, но нет другого.

5 [2]. Написать функции `fourierApprox(f, a, b, n, meth)`, принимающую на вход аргументы: функцию f , и рисующую анимацию из n кадров, на каждом из которых рисуется i -я частичная сумма ряда Фурье для этой функции на $[a, b]$ по системе функций, задаваемой параметром `meth`. Должна быть реализована стандартная тригонометрическая система функций, система функций Волша и любая другая полная система функций, не встречающаяся ни в одном из вариантов этого задания. Каждая система должна порождаться функцией вида `getFunc(n)`, возвращающую анонимную функцию номер n в той или иной системе.

6 [1]. Создать скрипт, в первом блоке которого задаётся переменная-функция и некоторая сетка. Вторым блоком рисует график функции на этой сетке, отмечает на все точки локального минимума, отмечает один глобальный максимум, и запускает от него до ближайшего минимума комету (команда `comet`). Подобрать примеры функций со многими экстремумами. В разных примерах комета должна иметь возможность двигаться как вправо, так и влево.

7 [2]. Написать функцию `getEqual(f, g, t0, t1, N)`, которая принимает на вход две функции, описывающие параметрическую кривую на плоскости,

$$l = \{(x, y) : x = f(t), y = g(t), t = [t_0, t_1]\},$$

и возвращает N точек $\{p_k\}_{k=1}^N$ таких, что $p_k \in l$ и $\|p_i - p_{i+1}\| = \text{const}$ для всех k , причём $p_1 = (f(t_0), g(t_0))$, $p_N = (f(t_1), g(t_1))$. Сравнить среднее расстояние между точками, полученных из равномерной по t параметризации ($p_k = (f(t_0 + k \cdot \Delta t), g(t_0 + k \cdot \Delta t))$) с получившимся расстоянием между точками. Продемонстрировать работу на фигурах Лиссажу:

$$x(t) = A \sin(at + \delta), y(t) = B \sin(bt), A, B, a, b, \delta - \text{const}.$$

8 [2]. Написать функцию `drawSet(rho, N)` которая получает на вход опорную функцию некоторого плоского множества `[val, point] = rho(x)`, возвращающую значение опорной функции в направлении x и соответствующий опорный вектор. Функция `drawSet` рисует внутреннюю и внешнюю кусочно-линейные аппроксимации границы множества с N точками. Функцией `convhull` пользоваться нельзя. Подготовить 3 примера с аналитически рассчитанными опорными функциями: эллипс, квадрат, ромб (в каждом случае центр не обязательно нулевой; центр и полуоси являются параметрами).

При помощи системы `Latex` в отдельном `pdf`-файле выписать вывод опорных функций для трех указанных выше множеств.

9 [1]. Используя функцию `fmincon`, написать функцию `supportLebesgue(f, opts)`, которая выдает (приближенную) опорную функцию множества $X = \{x : f(x) \leq 0\}$, которую можно использовать в предыдущем задании. Функция f предполагается выпуклой. Параметры `opts` являются параметрами функции `fmincon`.

10 [3]. Написать функцию `drawPolar(rho, N)`, которая получает на вход опорную функцию `[val, point] = rho(x)`, возвращающую значение опорной функции некоторого плоского множества \mathcal{X} в направлении x и соответствующий опорный вектор. Функция `drawPolar` рисует полюру множества \mathcal{X} и само \mathcal{X} . Подобрать 3–4 примера, когда вид полюры известен заранее, в том числе, когда $0 \notin \mathcal{X}$.

Трёхмерная графика.

Создать блочный скрипт. В первом блоке задается функция, зависящая от двух переменных и скалярного параметра, двумерная сетка (см. команду `meshgrid`) и границы изменения параметра.

11 [1]. Создать блок, рисующий анимацию с эволюцией поверхности по параметру (см. `surf`) и сохраняющий анимацию в переменную. На каждом кадре необходимо отметить локальные максимумы и минимумы, подобрать примеры, где их несколько и где они с течением времени меняют своё положение и число. В следующем блоке воспроизвести эту анимацию командой `movie`. Написать блок, где фиксируется некоторое значение параметра и при помощи команды `contour` строится проекция сечения функции на некотором фиксированном уровне на плоскость Oxy .

12 [1]. Создать блоки, сохраняющие анимацию в файл на диске в форматах `.mat` и `.avi`.

13 [2]. Известно, что за время выгула корова Мурка выедает всю траву, до которой может добраться. Для её выгула кот Матроскин использует систему из N колышков, каждый из которых соединяется с Муркой отдельной цепью, причём так, что сумма расстояний до всех колышков до Мурки не может превышать L метров в p -й норме Гёльдера. Написать функцию `viewEaten(points,L,p)`, принимающую массив из координат N точек на плоскости, расстояние L , и показатель нормы, и выводящую на экран закрашенную двумерную область, которую Мурка съест за выгул. Определить минимальное возможное значение L как функцию координат колышков.

Изучение четырехмерной графики.

См. команды `patch`, `isosurface`, `isonormals`, `camlight`, `shading` и `lighting`.

14 [1]. Написать функцию `drawBall(alpha, params)`, которая создаёт трехмерную сетку и рисует на ней линию уровня функции

$$f(x, y, z) = \begin{cases} |x|^\alpha + |y|^\alpha + |z|^\alpha, & \alpha \in (0, +\infty) \\ \max(|x|, |y|, |z|), & \alpha = +\infty \end{cases}$$

В `params` (это может быть список параметров или структура) передать параметры отрисовки (как минимум, цвет, диапазоны изменения переменных и число точек в сетке). Если исследуемое множество пустое, то функция должна выводить соответствующее сообщение об ошибке.

15 [1]. Написать функцию `drawManyBalls(alphas, colors, edges)`, которая рисует единичные шары в метриках, указанных в векторе `alphas`, с цветами, задаваемыми в векторе `colors`. Параметр `edges` отвечает за цвет граней и может принимать значение `'None'`.

ЛАБОРАТОРНАЯ РАБОТА №2

Работа с графикой

ВАРИАНТ 2

В этом задании символьные вычисления **не используются**.

Двумерная графика, часть первая.

Написать функцию `compareInterp(x, xx, f)`, которая принимает на вход две сетки, `xx` (более мелкую) и `x` (более крупную), $x \subset xx$, и указатель на функцию f (`function handle`). Эта функция рисует графики f на сетке `xx` и графики функций, получающихся интерполированием f с сетки `x` на сетку `xx` различными методами (флаги команды `interp1: nearest, linear, spline, cubic`). График оборудовать легендой, а также заголовком с наименованием метода интерполяции.

1 [1]. Написать функцию `compareInterp(x, xx, f)` в соответствии с Требованиями к Написанию Программ Практикума.

2 [2]. Подобрать набор функций, на котором продемонстрировать преимущества и недостатки каждого метода интерполирования (хотя бы 4 нетривиальных примера).

3 [2]. Оценить априорную погрешность интерполирования линейным методом. Для построения оценки считать известными максимумы производных. Построить график априорной и получившейся погрешности для двух функций: на одной с большим отклонением от априорной погрешности, на другом — с малым.

Двумерная графика, часть вторая.

4 [1]. Написать функцию `convergenceFunc(fn, f, a, b, n, convType)`, принимающую на вход аргументы: функцию `fn(n, x)`, такую, что $fn(n, x) = f_n(x)$, и функцию f , считающуюся пределом последовательности $f_n(x)$ на $[a, b]$ в смысле, задаваемом аргументом-строкой `convType`: это может быть поточечная сходимость, равномерная сходимость, среднеквадратичная сходимость. Функция рисует анимацию из n кадров, на каждом i -м из которых нарисованы f_i и f на отрезке $[a, b]$. В заголовке графика стоит вывести значения метрики разности для всех сходимостей, кроме поточечной.

5 [2]. Написать функции `fourierApprox(f, a, b, n, meth)`, принимающую на вход аргументы: функцию f , и рисующую анимацию из n кадров, на каждом из которых рисуется i -я частичная сумма ряда Фурье для этой функции на $[a, b]$ по системе функций, задаваемой параметром `meth`. Должна быть реализована стандартная тригонометрическая система функций, система многочленов Чебышёва и любая другая полная система функций, не встречающаяся ни в одном из вариантов этого задания. Каждая система должна порождаться функцией вида `getFunc(n)`, возвращающую анонимную функцию номер n в той или иной системе.

6 [1]. Создать скрипт, в первом блоке которого задаётся переменная-функция и некоторая сетка. Вторым блоком рисует график функции на этой сетке, отмечает на все точки локального минимума, отмечает один глобальный максимум, и запускает от него до ближайшего минимума комету (команда `comet`). Подобрать примеры функций со многими экстремумами. В разных примерах комета должна иметь возможность двигаться как вправо, так и влево.

7 [2]. Написать функцию `getEqual(f, g, t0, t1, N)`, которая принимает на вход две функции, описывающие параметрическую кривую на плоскости,

$$l = \{(x, y) : x = f(t), y = g(t), t = [t_0, t_1]\},$$

и возвращает N точек $\{p_k\}_{k=1}^N$ таких, что $p_k \in l$ и $\|p_i - p_{i+1}\| = \text{const}$ для всех k , причём $p_1 = (f(t_0), g(t_0))$, $p_N = (f(t_1), g(t_1))$. Сравнить среднее расстояние между точками, полученных из равномерной по t параметризации ($p_k = (f(t_0 + k \cdot \Delta t), g(t_0 + k \cdot \Delta t))$) с получившимся расстоянием между точками. Продемонстрировать работу на фигурах Лиссажу:

$$x(t) = A \sin(at + \delta), y(t) = B \sin(bt), A, B, a, b, \delta = \text{const}.$$

8 [2]. Написать функцию `drawSet(rho, N)` которая получает на вход опорную функцию некоторого плоского множества `[val, point] = rho(x)`, возвращающую значение опорной функции в направлении x и соответствующий опорный вектор. Функция `drawSet` рисует внутреннюю и внешнюю кусочно-линейные аппроксимации границы множества с N точками. Функцией `convhull` пользоваться нельзя. Подготовить 3 примера с аналитически рассчитанными опорными функциями: эллипс, квадрат, ромб (в каждом случае центр не обязательно нулевой; центр и полуоси являются параметрами).

При помощи системы `Latex` в отдельном `pdf`-файле выписать вывод опорных функций для трех указанных выше множеств.

9 [1]. Используя функцию `fmincon`, написать функцию `supportLebesgue(f, opts)`, которая выдает (приближенную) опорную функцию множества $X = \{x : f(x) \leq 0\}$, которую можно использовать в предыдущем задании. Функция f предполагается выпуклой. Параметры `opts` являются параметрами функции `fmincon`.

10 [3]. Написать функцию `drawPolar(rho, N)`, которая получает на вход опорную функцию `[val, point] = rho(x)`, возвращающую значение опорной функции некоторого плоского множества \mathcal{X} в направлении x и соответствующий опорный вектор. Функция `drawPolar` рисует полярную функцию множества \mathcal{X} и само \mathcal{X} . Подобрать 3–4 примера, когда вид полярной функции известен заранее, в том числе, когда $0 \notin \mathcal{X}$.

Трёхмерная графика.

Создать блочный скрипт. В первом блоке задается функция, зависящая от двух переменных и скалярного параметра, двумерная сетка (см. команду `meshgrid`) и границы изменения параметра.

11 [1]. Создать блок, рисующий анимацию с эволюцией поверхности по параметру (см. `surf`) и сохраняющий анимацию в переменную. На каждом кадре необходимо отметить локальные максимумы и минимумы, подобрать примеры, где их несколько и где они с течением времени меняют своё положение и число. В следующем блоке воспроизвести эту анимацию командой `movie`. Написать блок, где фиксируется некоторое значение параметра и при помощи команды `contour` строится проекция сечения функции на некотором фиксированном уровне на плоскость Oxy .

12 [1]. Создать блоки, сохраняющие анимацию в файл на диске в форматах `.mat` и `.avi`.

13 [2]. Колонисты установили на поверхности Марса N независимых антенных станций (поля от них складываются), каждая из которых генерирует вокруг себя беспроводную сеть с уровнем сигнала $V/(1 + d(p_k, p))$, где p_k — точка, где находится k -я станция, p — точка, где проводится замер, $d(\cdot, \cdot)$ — евклидово расстояние, V — исходный уровень сигнала. Для уверенной работы марсохода требуется сигнал с уровнем, не меньшим L . Написать функцию `viewPossible(points, P, L)`, принимающую массив из координат N точек на плоскости, уровни сигналов L и P , и выводящую на экран область, в которой можно уверенно управлять марсоходом. Определить, будет ли полученная область односвязной.

Изучение четырехмерной графики.

См. команды `patch`, `isosurface`, `isonormals`, `camlight`, `shading` и `lighting`.

14 [1]. Написать функцию `drawBall(alpha, params)`, которая создаёт трехмерную сетку и рисует на ней линию уровня функции

$$f(x, y, z) = \begin{cases} |x|^\alpha + |y|^\alpha + |z|^\alpha, & \alpha \in (0, +\infty) \\ \max(|x|, |y|, |z|), & \alpha = +\infty \end{cases}$$

В `params` (это может быть список параметров или структура) передать параметры отрисовки (как минимум, цвет, диапазоны изменения переменных и число точек в сетке). Если исследуемое множество пустое, то функция должна выводить соответствующее сообщение об ошибке.

15 [1]. Написать функцию `drawManyBalls(alphas, colors, edges)`, которая рисует единичные шары в метриках, указанных в векторе `alphas`, с цветами, задаваемыми в векторе `colors`. Параметр `edges` отвечает за цвет граней и может принимать значение `'None'`.

ЛАБОРАТОРНАЯ РАБОТА №2

Работа с графикой

ВАРИАНТ 3

В этом задании символьные вычисления **не используются**.

Двумерная графика, часть первая.

Написать функцию `compareInterp(x, xx, f)`, которая принимает на вход две сетки, `xx` (более мелкую) и `x` (более крупную), $x \subset xx$, и указатель на функцию f (`function handle`). Эта функция рисует графики f на сетке `xx` и графики функций, получающихся интерполированием f с сетки `x` на сетку `xx` различными методами (флаги команды `interp1: nearest, linear, spline, cubic`). График оборудовать легендой, а также заголовком с наименованием метода интерполяции.

1 [1]. Написать функцию `compareInterp(x, xx, f)` в соответствии с Требованиями к Написанию Программ Практикума.

2 [2]. Подобрать набор функций, на котором продемонстрировать преимущества и недостатки каждого метода интерполирования (хотя бы 4 нетривиальных примера).

3 [2]. Оценить априорную погрешность интерполирования кубическими сплайнами. Для построения оценки считать известными максимумы производных. Построить график априорной и получившейся погрешности для двух функций: на одной с большим отклонением от априорной погрешности, на другом — с малым.

Двумерная графика, часть вторая.

4 [1]. Написать функцию `convergenceFunc(fn, f, a, b, n, convType)`, принимающую на вход аргументы: функцию `fn(n, x)`, такую, что $fn(n, x) = f_n(x)$, и функцию f , считающуюся пределом последовательности $f_n(x)$ на $[a, b]$ в смысле, задаваемом аргументом-строкой `convType`: это может быть поточечная сходимость, равномерная сходимость, среднеквадратичная сходимость. Функция рисует анимацию из n кадров, на каждом i -м из которых нарисованы f_i и f на отрезке $[a, b]$. В заголовке графика стоит вывести значения метрики разности для всех сходимостей, кроме поточечной.

5 [2]. Написать функции `fourierApprox(f, a, b, n, meth)`, принимающую на вход аргументы: функцию f , и рисующую анимацию из n кадров, на каждом из которых рисуется i -я частичная сумма ряда Фурье для этой функции на $[a, b]$ по системе функций, задаваемой параметром `meth`. Должна быть реализована стандартная тригонометрическая система функций, система многочленов Эрмита и любая другая полная система функций, не встречающаяся ни в одном из вариантов этого задания. Каждая система должна порождаться функцией вида `getFunc(n)`, возвращающую анонимную функцию номер n в той или иной системе.

6 [1]. Создать скрипт, в первом блоке которого задаётся переменная-функция и некоторая сетка. Вторым блоком рисует график функции на этой сетке, отмечает на все точки локального минимума, отмечает один глобальный максимум, и запускает от него до ближайшего минимума комету (команда `comet`). Подобрать примеры функций со многими экстремумами. В разных примерах комета должна иметь возможность двигаться как вправо, так и влево.

7 [2]. Написать функцию `getEqual(f, g, t0, t1, N)`, которая принимает на вход две функции, описывающие параметрическую кривую на плоскости,

$$l = \{(x, y) : x = f(t), y = g(t), t = [t_0, t_1]\},$$

и возвращает N точек $\{p_k\}_{k=1}^N$ таких, что $p_k \in l$ и $\|p_i - p_{i+1}\| = \text{const}$ для всех k , причём $p_1 = (f(t_0), g(t_0))$, $p_N = (f(t_1), g(t_1))$. Сравнить среднее расстояние между точками, полученных из равномерной по t параметризации ($p_k = (f(t_0 + k \cdot \Delta t), g(t_0 + k \cdot \Delta t))$) с полученным расстоянием между точками. Продемонстрировать работу на фигурах Лиссажу:

$$x(t) = A \sin(at + \delta), y(t) = B \sin(bt), A, B, a, b, \delta = \text{const}.$$

8 [2]. Написать функцию `drawSet(rho, N)` которая получает на вход опорную функцию некоторого плоского множества `[val, point] = rho(x)`, возвращающую значение опорной функции в направлении x и соответствующий опорный вектор. Функция `drawSet` рисует внутреннюю и внешнюю кусочно-линейные аппроксимации границы множества с N точками. Функцией `convhull` пользоваться нельзя. Подготовить 3 примера с аналитически рассчитанными опорными функциями: эллипс, квадрат, ромб (в каждом случае центр не обязательно нулевой; центр и полуоси являются параметрами).

При помощи системы `Latex` в отдельном `pdf`-файле выписать вывод опорных функций для трех указанных выше множеств.

9 [1]. Используя функцию `fmincon`, написать функцию `supportLebesgue(f, opts)`, которая выдает (приближенную) опорную функцию множества $X = \{x : f(x) \leq 0\}$, которую можно использовать в предыдущем задании. Функция f предполагается выпуклой. Параметры `opts` являются параметрами функции `fmincon`.

10 [3]. Написать функцию `drawPolar(rho, N)`, которая получает на вход опорную функцию `[val, point] = rho(x)`, возвращающую значение опорной функции некоторого плоского множества \mathcal{X} в направлении x и соответствующий опорный вектор. Функция `drawPolar` рисует полярную область множества \mathcal{X} и само \mathcal{X} . Подобрать 3–4 примера, когда вид полярной области известен заранее, в том числе, когда $0 \notin \mathcal{X}$.

Трёхмерная графика.

Создать блочный скрипт. В первом блоке задается функция, зависящая от двух переменных и скалярного параметра, двумерная сетка (см. команду `meshgrid`) и границы изменения параметра.

11 [1]. Создать блок, рисующий анимацию с эволюцией поверхности по параметру (см. `surf`) и сохраняющий анимацию в переменную. На каждом кадре необходимо отметить локальные максимумы и минимумы, подобрать примеры, где их несколько и где они с течением времени меняют своё положение и число. В следующем блоке воспроизвести эту анимацию командой `movie`. Написать блок, где фиксируется некоторое значение параметра и при помощи команды `contour` строится проекция сечения функции на некотором фиксированном уровне на плоскость Oxy .

12 [1]. Создать блоки, сохраняющие анимацию в файл на диске в форматах `.mat` и `.avi`.

13 [2]. Полярнику необходимо обойти N станций в бурю в полярную ночь. Каждая станция оборудована фонарём, в свете которого видна территория на L метров вокруг станции. Известно, что оказавшись в темноте, полярник неизбежно потеряется и погибнет. Написать функцию `viewViewable(points, L)`, принимающую массив из координат N точек на плоскости и дистанцию L , и выводящую границу освещённой области. Определить, сможет ли полярник обойти все станции (порядок обхода фиксирован).

Изучение четырехмерной графики.

См. команды `patch`, `isosurface`, `isonormals`, `camlight`, `shading` и `lighting`.

14 [1]. Написать функцию `drawBall(alpha, params)`, которая создаёт трехмерную сетку и рисует на ней линию уровня функции

$$f(x, y, z) = \begin{cases} |x|^\alpha + |y|^\alpha + |z|^\alpha, & \alpha \in (0, +\infty) \\ \max(|x|, |y|, |z|), & \alpha = +\infty \end{cases}$$

В `params` (это может быть список параметров или структура) передать параметры отрисовки (как минимум, цвет, диапазоны изменения переменных и число точек в сетке). Если исследуемое множество пустое, то функция должна выводить соответствующее сообщение об ошибке.

15 [1]. Написать функцию `drawManyBalls(alphas, colors, edges)`, которая рисует единичные шары в метриках, указанных в векторе `alphas`, с цветами, задаваемыми в векторе `colors`. Параметр `edges` отвечает за цвет граней и может принимать значение `'None'`.