**Описание алгоритма формирования матриц весовых коэффициентов**

**для получения решения СДУ в полиномиальном матричном виде**

**(v.1 29.08.2021)**

Алгоритм предназначен для расчёта матриц весовых коэффициентов до третьего порядка включительно решения СДУ с полиномиальной правой частью до третьего порядка включительно с двумя неизвестными

Рассматривается СДУ

;

с начальными условиями .

Здесь - вторая и третья кронекеровские степени вектора X соответственно:

, .

Решение строится в виде полинома третьей степени

по кронекеровским степеням вектора .

Матрицы полиномиального разложения решения представляются формулами:

, где

, где

, матрица находится из решения матричного уравнения

, под звёздочкой понимается тензорное умножение.

РАЗДЕЛ 0. ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ

0.1. Функция расчёта второй кронекеровской степени символьной матрицы 2х2

sym\_matrix\_kronecker\_2\_deg2(A)

Вход: A - квадратная символьная матрица размерности 2х2

Выход: В - матрица размерности 3х3 в символьном виде

Алгоритм:

0.2. Функция расчёта третьей кронекеровской степени символьной матрицы 2х2

sym\_matrix\_kronecker\_2\_deg3(A)

Вход: A - квадратная символьная матрица размерности 2х2

Выход: В - матрица размерности 4х4 в символьном виде

Алгоритм:

0.3 Функция вычисления матричной экспоненты для матрицы-константы размерности 2х2

sym\_matrix\_exponenta(A, t0)

Вход: A - квадратная матрица - константа размерности 2х2

t0 - символьное значение

Выход: матричная экспонента размерности 2х2 в символьном виде

Алгоритм: расчёт матрицы - применение библиотечной функции sympy.solvers.ode.systems.matrix\_exp(А, t-t0)

0.4 Функция вычисления тензорного произведения двух матриц

sym\_matrix\_tenzor\_multiplicatioin(A, B)

Вход: A - матрица размерности х

B - матрица размерности х

Выход: С - матрица размерности х в символьном виде

Алгоритм

0.5 Функция нахождения формы матрицы в символьных переменных представляющих элементы матриц и

из уравнения

sym\_matrix\_r23\_formal\_calculation()

Алгоритм:

0.5.1. Объявление символьных переменных матрицы для построения матрицы

r11 = Symbol('r11')

r12 = Symbol('r12')

r21 = Symbol('r21')

r22 = Symbol('r22')

* + 1. Объявление символьных переменных для построения матрицы

p11 = Symbol('p11')

p12 = Symbol('p12')

p13 = Symbol('p13')

p21 = Symbol('p21')

p22 = Symbol('p22')

p23 = Symbol('p23')

0.5.3 Объявление символьных переменных для построения векторов , ,

x = Symbol('x')

y = Symbol('y')

0.5.4. Объявление вспомогательной символьной переменной

t = Symbol('t')

0.5.5. Формирование матриц R= и P=

R = Matrix([[r11, r12], [r21, r22]])

P = Matrix([[p11, p12, p13], [p21, p22, p23]])

0.5.6 Объявление вспомогательной матрицы SR и матрицы результата R23

SR = Matrix([[t,t,t,t], [t,t,t,t], [t,t,t,t], [t,t,t,t]])

R23 = Matrix([[t,t,t,t], [t,t,t,t], [t,t,t,t]])

0.5.7. Формирование векторов , ,

X = Matrix([[x], [y]])

XX = Matrix([[x\*\*2], [x\*y], [y\*\*2]])

XXX = Matrix([[x\*\*3], [(x\*\*2)\*y], [x\*(y\*\*2)], [y\*\*3]])

0.5.8 Вычисление произведения

RX = R \* X

0.5.9 Вычисление произведения

PXX = P \* XX

0.5.10 Вычисление тензорного произведения

Q1 = sym\_matrix\_tenzor\_multiplicatioin(RX, PXX)

0.5.11 Вычисление тензорного произведения

Q2 = sym\_matrix\_tenzor\_multiplicatioin(PXX, RX)

0.5.12 Вычисление суммы

Q = Q1 + Q2

0.5.13. В двойном цикле по числу строк и столбцов матрицы Q нахождение коэффициентов при элементах вектора из уравнения :

temp = sympy.collect(sympy.expand(Q[ind\_i, ind\_j]), x)

sol\_x3 = sympy.limit(temp/(x\*\*3), x, sympy.oo)

SR[ind\_i, 0] = sol\_x3

sol\_x2y = sympy.limit((temp - sol\_x3\*x\*\*3)/(y\*x\*\*2), x, sympy.oo)

SR[ind\_i, 1] = sol\_x2y

sol\_xy2 = sympy.limit((temp - sol\_x3\*x\*\*3 - sol\_x2y\*y\*x\*\*2)/(x\*y\*\*2), x, sympy.oo)

SR[ind\_i, 2] = sol\_xy2

sol\_y3 = sympy.limit(temp/(y\*\*3), y, sympy.oo)

SR[ind\_i, 3] = sol\_y3

0.5.14 Формирование матрицы результата R23

R23[0,:] = SR[0,:]

R23[1,:] = SR[1,:]

R23[2,:] = SR[3,:]

РАЗДЕЛ 1. Расчёт матриц для фиксированного значения t

Функция weight\_calculation(matrix\_p11, s\_matrix\_p12, s\_matrix\_p13, t0, tt)

Вход: matrix\_p11 - квадратная матрица - константа размерности 2х2

s\_matrix\_p12 - матрица - константа размерности 2х3

s\_matrix\_p13 - матрица - константа размерности 2х4

t0 - значение

tt - значение t

Выход: матрицы

Алгоритм:

* 1. Определение символьных переменных t и z:

z = Symbol('z') // z в коде обозначает переменную в расчётных формулах

t = Symbol('t')

* 1. Расчёт матрицы в символьном виде

Вызов функции s\_matrix\_r11\_calculation(matrix\_p11, t0)

1.2.1. Функция s\_matrix\_r11\_calculation(matrix\_p11, t0)

Вход: matrix\_p11 - квадратная матрица - константа размерности 2х2

t0 - численное значение

Выход: матрица s\_matrix\_r11 в символьном виде

Алгоритм: вызов функции sym\_matrix\_exponenta(matrix\_p11, t0)

* 1. Расчёт матрицы в символьном виде

Вызов функции s\_matrix\_r11\_calculation(matrix\_p11, z)

Выход: матрица s\_matrix\_r11\_z в символьном виде

* 1. Расчёт матрицы в символьном виде

Вызов функции sym\_matrix\_kronecker\_2\_deg2(s\_matrix\_r11)

Выход: матрица s\_matrix\_r22 в символьном виде

* 1. Расчёт матрицы в символьном виде

Вызов функции sym\_matrix\_kronecker\_2\_deg3(s\_matrix\_r11)

Выход: матрица s\_matrix\_r33 в символьном виде

* 1. Расчёт матриц в символьном виде

Замена символьной переменной t на символьную переменную z в символьных матрицах

s\_matrix\_r22 и s\_matrix\_r33:

s\_matrix\_r22\_z = s\_matrix\_r22.subs(t, z)

s\_matrix\_r33\_z = s\_matrix\_r33.subs(t, z)

* 1. Расчёт матрицы  в численном виде (t = tt)
     1. Расчёт подынтегрального выражения – матрицы

Вычисление произведения матриц:

s\_matrix\_r12\_int\_z = s\_matrix\_r11\_z \* s\_matrix\_p12 \* s\_matrix\_r22\_z

* + 1. Замена символьной переменной t на заданное значение tt:

s\_matrix\_r12\_int = s\_matrix\_r12\_int\_z.subs(t, tt)

* + 1. Численное интегрирование подынтегрального выражения:

Вызов функции s\_matrix\_r12 = matrix\_r12\_integrate(s\_matrix\_r12\_int, t0, tt)

Функция matrix\_r12\_integrate(s\_matrix\_r12\_int, t0, tt)

Вход: s\_matrix\_r12\_int - подынтегральная матрица с символьной переменной z

t0 - значение

tt - значение t

Выход: матрица s\_matrix\_r12 в численном виде

Алгоритм:

* + - 1. Вычисление числа строк и столбцов подынтегральной матрицы s\_matrix\_r12\_int
      2. В двойном цикле по числу строк и столбцов вызов библиотечной функции численного интегрирования scipy.integrate.quad():

matrix[index\_1, index\_2] = quad(s\_matrix\_r12\_val\_calculation, t0, tt, args=(index\_1, index\_2, s\_matrix\_r12\_int))[0]

В качестве подынтегральной функции для численного интегрирования используется функция s\_matrix\_r12\_val\_calculation с аргументами (index\_1, index\_2, s\_matrix\_r12\_int))

* + - * 1. Расчёт значения элемента матрицы подынтегрального выражения при конкретном значении z

Функция s\_matrix\_r12\_val\_calculation с аргументами (zz, index\_1, index\_2, s\_matrix\_r12\_int))

Вход: zz – значение символьной переменной z

index\_1 – номер строки в подынтегральной матрице

index\_2 – номер столбца в подынтегральной матрице

s\_matrix\_r12\_int – подынтегральная символьная матрица при конкретном значении z

Выход: значение s\_matrix\_r12\_int[index\_1, index\_2] при z=zz

Алгоритм: замена символьной переменной z на значение zz в матрице s\_matrix\_r12\_int[index\_1, index\_2]:

s\_matrix\_r12\_v = s\_matrix\_r12\_int.subs(z, zz)

* + - 1. Формирование результата – матрицы s\_matrix\_r12
  1. Расчёт матрицы в численном виде (t = tt)
     1. Расчёт первого интеграла – матрицы
        1. Вычисление произведения матриц :

s\_matrix\_r1311\_int = s\_matrix\_r11\_z \* s\_matrix\_p12

* + - 1. Замена символьной переменной t на заданное значение tt:

s\_matrix\_r1311\_int = s\_matrix\_r1311\_int\_z.subs(t, tt)

* + - 1. Численное интегрирование подынтегрального выражения:

Вызов функции s\_matrix\_r131 = matrix\_r131\_integrate(s\_matrix\_r1311\_int, s\_matrix\_r11, s\_matrix\_r12\_int\_z, t0, tt)

Функция matrix\_r131\_integrate(s\_matrix\_r1311\_int, s\_matrix\_r11, s\_matrix\_r12\_int\_z, t0, tt)

Вход:

s\_matrix\_r1311\_int – первые два сомножителя из подынтегральной матрицы с символьной переменной z

s\_matrix\_r11 – матрица в символьном виде с символьной переменной t

s\_matrix\_r12\_int\_z – матрица в символьном виде с символьными переменными t и z

t0 - значение

tt - значение t

Выход: матрица s\_matrix\_r131 в численном виде

Алгоритм:

* + - * 1. В двойном цикле по числу строк равных 2 и числу столбцов равных 4 вызов библиотечной функции численного интегрирования scipy.integrate.quad():

matrix[index\_1, index\_2] = quad(s\_matrix\_r131\_val\_calculation, t0, tt, args=(t0, index\_1, index\_2, s\_matrix\_r1311\_int, s\_matrix\_r12\_int\_z, s\_matrix\_r11))[0]

В качестве подынтегральной функции для численного интегрирования используется функция s\_matrix\_r131\_val\_calculation с аргументами (t0, index\_1, index\_2, s\_matrix\_r1311\_int, s\_matrix\_r12\_int\_z, s\_matrix\_r11))

* + - * 1. Расчёт значения элемента матрицы подынтегрального выражения при конкретном значении z

Функция s\_matrix\_r131\_val\_calculation(zz, t0, ind\_i, ind\_j, s\_matrix\_r1311\_int, s\_matrix\_r12\_int\_z, s\_matrix\_r11)

Вход: zz – значение символьной переменной z

t0 - значение

ind\_i – номер строки в подынтегральной матрице первого интеграла

ind\_j – номер столбца в подынтегральной матрице первого интеграла

s\_matrix\_r1311\_int – первые два сомножителя из подынтегральной матрицы с символьной переменной z

s\_matrix\_r12\_int\_z – матрица в символьном виде с символьными переменными t и z

s\_matrix\_r11 – матрица в символьном виде с символьной переменной t

Выход: значение s\_matrix\_r131[ind\_i, ind\_j] при z=zz

Алгоритм:

Определение символьных переменных t и z:

z = Symbol('z') // z в коде обозначает переменную τ в расчётных формулах

t = Symbol('t')

Вычисление матрицы в численном виде при z=zz

matrix\_r12\_v = s\_matrix\_r12\_calculation(zz, t0, s\_matrix\_r12\_int\_z)

Замена в матрице символьной переменной t на значение zz

matrix\_r11\_v = s\_matrix\_r11.subs(t, zz)

Вычисление численного значения матрицы при z=zz

Вызов функции s\_matrix\_r23 = sym\_matrix\_r23\_calculation(matrix\_r11\_v, matrix\_r12\_v)

Функция sym\_matrix\_r23\_calculation(matrix\_r11\_v, matrix\_r12\_v)

вычисляет численное значение матрицы при z=zz

Вход: matrix\_r11\_v – матрица в численном виде matrix\_r12\_v - матрица в численном виде

Выход: значение s\_matrix\_r131[ind\_i, ind\_j] при z=zz

Алгоритм:

Объявление символьных переменных матрицы

r11 = Symbol('r11')

r12 = Symbol('r12')

r21 = Symbol('r21')

r22 = Symbol('r22')

Объявление символьных переменных матрицы

p11 = Symbol('p11')

p12 = Symbol('p12')

p13 = Symbol('p13')

p21 = Symbol('p21')

p22 = Symbol('p22')

p23 = Symbol('p23')

Вычисление матрицы в символьном виде с объявленными в предыдущем пункте символьными переменными

Вызов функции R23 = sym\_matrix\_r23\_formal\_calculation()

Замена символьных переменных в матрице R23 на численные значения матрицы

R23 = R23.subs(r11, R11[0, 0])

R23 = R23.subs(r12, R11[0, 1])

R23 = R23.subs(r21, R11[1, 0])

R23 = R23.subs(r22, R11[1, 1])

Замена символьных переменных в матрице R23 на численные значения матрицы

R23 = R23.subs(p11, R12[0, 0])

R23 = R23.subs(p12, R12[0, 1])

R23 = R23.subs(p13, R12[0, 2])

R23 = R23.subs(p21, R12[1, 0])

R23 = R23.subs(p22, R12[1, 1])

R23 = R23.subs(p23, R12[1, 2])

Вычисление подынтегральной матрицы в символьном виде с символьной переменной t:

s\_matrix\_r131\_v = s\_matrix\_r1311\_int \* s\_matrix\_r23

Замена в подынтегральной матрице символьной переменной z на значение zz:

s\_matrix\_r131\_v = s\_matrix\_r131\_v.subs(z, zz)

Формирование результата – элемента подынтегральной матрицы с номерами ind\_i, ind\_j

* + 1. Вычисление первого интеграла
       1. Расчёт подынтегрального выражения для второго интеграла – матрицы

Вычисление произведения матриц: s\_matrix\_r132\_int = s\_matrix\_r11\_z \* s\_matrix\_p13 \* s\_matrix\_r33\_z

* + - 1. Замена символьной переменной t на заданное значение tt:

s\_matrix\_r132\_int = s\_matrix\_r132\_int\_z.subs(t, tt)

* + - 1. Численное интегрирование подынтегрального выражения:

Вызов функции s\_matrix\_r132 = matrix\_r132\_integrate(s\_matrix\_r132\_int, t0, tt)

Функция matrix\_r132\_integrate(s\_matrix\_r132\_int, t0, tt)

Вход: s\_matrix\_r132\_int - подынтегральная матрица с символьной переменной z

t0 - значение

tt - значение t

Выход: матрица s\_matrix\_r132 в численном виде

Алгоритм:

* + - * 1. Вычисление числа строк и столбцов подынтегральной матрицы s\_matrix\_r12\_int
        2. В двойном цикле по числу строк и столбцов вызов библиотечной функции численного интегрирования scipy.integrate.quad():

matrix[index\_1, index\_2] = quad(s\_matrix\_r132\_val\_calculation, t0, tt, args=(index\_1, index\_2, s\_matrix\_r132\_int))[0]

В качестве подынтегральной функции для численного интегрирования используется функция s\_matrix\_r132\_val\_calculation с аргументами (index\_1, index\_2, s\_matrix\_r12\_int))

Расчёт значения элемента матрицы подынтегрального выражения при конкретном значении z

Функция s\_matrix\_r132\_val\_calculation с аргументами (zz, index\_1, index\_2, s\_matrix\_r12\_int))

Вход: zz – значение символьной переменной z

index\_1 – номер строки в подынтегральной матрице

index\_2 – номер столбца в подынтегральной матрице

s\_matrix\_r132\_int – подынтегральная символьная матрица при конкретном значении z

Выход: значение s\_matrix\_r132\_int[index\_1, index\_2] при z=zz

Алгоритм: замена символьной переменной z на значение zz в матрице s\_matrix\_r132\_int[index\_1, index\_2]:

s\_matrix\_r132\_v = s\_matrix\_r132\_int.subs(z, zz)

* + - * 1. Формирование результата – матрицы s\_matrix\_r132
    1. Вычисление матрицы как суммы двух рассчитанных интегралов:

s\_matrix\_r13 = s\_matrix\_r131 + s\_matrix\_r132