УДК 519.688

Г. Г. Акниев, Р. М. Гаджимирзаев

Быстрый алгоритм приближенного нахождения решения задачи Коши для ОДУ

Настоящая статья посвящена алгоритму быстрого приближенного нахождения решения задачи Коши для ОДУ, путем вычисления коэффициентов разложения этого решения в ряд по системе функций $\{\varphi_{1,n}(x)\}_{n=0}^{\infty}$, где $\varphi_{1,0}(x) = 1$, $\varphi_{1,1}(x) = x$, $\varphi_{1,n+1}(x) = \frac{\sqrt{2}}{\pi n}\sin(\pi n x)$, $n = 1, 2, \ldots$, ортонормированной относительно скалярного произведения Соболева $\langle f, g\rangle = f(0)g(0) + \int_0^1 f'(x)g'(x)dx$ и порожденной косинусами $\varphi_0(x) = 1$, $\{\varphi_n(x) = \sqrt{2}\cos(\pi n x)\}_{n=1}^{\infty}$. Вычисление коэффициентов осуществляется посредством итерационного процесса, основанного на быстром преобразовании Фурье.

Библиография: 4 названия.

The present article considers the quick algorithm for finding an approximate solution for the Cauchy problem for ODE by calculating the coefficients of expansion of this solution in terms of the system $\{\varphi_{1,n}(x)\}_{n=0}^{\infty}$, where $\varphi_{1,0}(x) = 1$, $\varphi_{1,1}(x) = x$, $\varphi_{1,n+1}(x) = \frac{\sqrt{2}}{\pi n}\sin(\pi n x)$, $n = 1, 2, \ldots$. This system is orthonormal with respect to the Sobolev scalar product $\langle f, g\rangle = f(0)g(0) + \int_0^1 f'(x)g'(x)dx$ and generated by cosines $\varphi_0(x) = 1$, $\{\varphi_n(x) = \sqrt{2}\cos(\pi n x)\}_{n=1}^{\infty}$. The calculation of these coefficients is performed by an iterative process based on the fast Fourier transform.

Bibliography: 4 items.

Ключевые слова: обыкновенное дифференциальное уравнение, задача Коши, скалярное произведение типа Соболева, ортонормированные по Соболеву функции, быстрое преобразование Фурье, дискретное косинус-преобразование

Keywords: ordinary differential equation, Cauchy problem, inner product of Sobolev type, Sobolev orthonormal function, fast Fourier transform, discrete cosine transform

Введение

Через $W^1_{L^2(0,1)}$ обозначим пространство Соболева, которое состоит из абсолютно непрерывных на $[0, 1]$ функций f и таких, что $f' \in L^2(0, 1)$. В работе [1] была введена система функций

$$\varphi_{1,0}(x) = 1, \quad \varphi_{1,1}(x) = x, \quad \varphi_{1,n+1}(x) = \frac{\sqrt{2}}{\pi n}\sin(\pi n x), \quad n = 1, 2, \ldots, \tag{1}$$

которая ортонормирована относительно скалярного произведения Соболева

$$\langle \varphi_{1,n}, \varphi_{1,m}\rangle = \varphi_{1,n}(0)\varphi_{1,m}(0) + \int_0^1 \varphi'_{1,n}(x)\varphi'_{1,m}(x)dx$$

и порождена системой косинусов

$$\varphi_0(x) = 1, \quad \{\varphi_n(x) = \sqrt{2}\cos(\pi n x)\}_{n=1}^{\infty} \tag{2}$$

посредством равенств

$$\varphi_{1,0}(x) = 1, \quad \varphi_{1,n}(x) = \int_0^x \varphi_{n-1}(t)dt, \quad n = 1, 2, \dots$$

В этой же работе было показано, что системы (1) и (2), взятые вместе являются весьма эффективным инструментом численного решения систем линейных и нелинейных дифференциальных уравнений с помощью итерационного процесса, основанного на построении сжимающего оператора A. С подробным описанием этого итерационого процесса и построением сжимающего оператора можно ознакомиться в работе [2]. В настоящей работе мы рассмотрим задачу Коши

$$y' = f(x, y), \quad y(0) = y^0, \quad x \in [0, h], \tag{3}$$

где h выбирается из условия (8), $f(x, y)$ — функция двух переменных, которая непрерывна в некоторой замкнутой области $\bar{G}$ переменных (x, y), содержащей точку $(0, y^0)$ и такой, что $[0, h] \times \mathbb{R} \subset \bar{G}$. Кроме того, функция $f(x, y)$ удовлетворяет условию Липшица по второй переменной, т.е.

$$|f(x, y_1) - f(x, y_2)| \leqslant \lambda |y_1 - y_2|, \quad 0 \leqslant x \leqslant h.$$

Выполнив замену переменной $x = h\sin \pi t$ в (3), мы перейдем к следующей задаче

$$\eta'(t) = h\pi \cos(\pi t) f(h\sin \pi t, \eta(t)), \quad \eta(0) = y^0, \quad 0 \leqslant t \leqslant \frac{1}{2}, \tag{4}$$

где $\eta(t) = y(h\sin \pi t)$. Заметим, что после такой замены функция $y(h\sin \pi t)$ на отрезке $[0, 1]$ будет симметричной относительно прямой $t = \frac{1}{2}$. Поэтому, в дальнейшем мы будем считать, что $t \in [0, 1]$. Так как функция $f(x, y)$ непрерывна в области $\bar{G}$, то $\eta(t) \in W^1_{L^2(0,1)}$. Тогда в силу теоремы 2.2 из работы [3] следует, что функцию $\eta(t)$ можно разложить в равномерно сходящийся ряд Фурье по системе (1), т.е.

$$\eta(t) = \eta(0) + \sum_{k=0}^{\infty} c_{1,k+1}(\eta)\varphi_{1,k+1}(t), \tag{5}$$

где

$$c_{1,k+1}(\eta) = \int\limits_0^1 \eta'(\tau)\varphi_k(\tau)d\tau.$$

Кроме того, в силу полноты системы (2) в $L^2(0, 1)$, для функций $\eta'(t)$ и $q(t) = \cos(\pi t) f(h\sin \pi t, \eta(t))$ в метрике пространства $L^2(0, 1)$, имеют место следующие разложения в ряд Фурье по системе (2):

$$\eta'(t) = \sum_{k=0}^{\infty} c_{1,k+1}(\eta)\varphi_k(t), \tag{6}$$

$$q(t) = \cos(\pi t) f(h \sin \pi t, \eta(t)) = \sum_{k=0}^{\infty} c_k(q) \varphi_k(t). \tag{7}$$

Тогда из (4)–(7) для $k \geqslant 0$ имеем

$$c_{1,k+1}(\eta) = h\pi c_k(q) = h\pi \int\limits_0^1 f(h \sin \pi\tau, \eta(\tau)) \cos(\pi\tau) \varphi_k(\tau) d\tau =$$

$$h\pi \int\limits_0^1 f\left(h \sin \pi\tau, \eta(0) + \sum_{i=0}^{\infty} c_{1,i+1}(\eta) \varphi_{1,i+1}(\tau)\right) \cos(\pi\tau) \varphi_k(\tau) d\tau.$$

Правая часть последнего равенства и есть вышеупомянутый оператор A, сопоставляющий точке $(c_0, c_1, \ldots) \in l^2$ точку $(c'_0, c'_1, \ldots) \in l^2$ по правилу

$$c'_k = h\pi \int\limits_0^1 f\left(h \sin \pi\tau, \eta(0) + \sum_{i=0}^{\infty} c_i \varphi_{1,i+1}(\tau)\right) \cos(\pi\tau) \varphi_k(\tau) d\tau.$$

Если вместо бесконечной суммы в правой части предыдущего равенства рассмотреть ее частичную сумму порядка $N-1$, то мы получим конечномерный аналог оператора A, сопоставляющий точке $(c_0, c_1, \ldots, c_{N-1}) \in \mathbb{R}^N$ точку $(c'_0, c'_1, \ldots, c'_{N-1}) \in \mathbb{R}^N$ по правилу

$$c'_k = h\pi \int\limits_0^1 f\left(h \sin \pi\tau, \eta(0) + \sum_{i=0}^{N-1} c_i \varphi_{1,i+1}(\tau)\right) \cos(\pi\tau) \varphi_k(\tau) d\tau, \ k = \overline{0, N-1}.$$

Как было показано в работе [2], оператор A будет сжимающим, если

$$h \kappa_\varphi \lambda < 1, \tag{8}$$

где $\kappa_\varphi = \left(\int_0^1 \sum_{k=1}^{\infty} (\varphi_{1,k}(t))^2 dt\right)^{\frac{1}{2}} = \frac{1}{\sqrt{2}}$, и, следовательно, итерационный процесс будет сходится к его неподвижной точке, которую мы обозначим через $(\overline{c}_0, \overline{c}_1, \ldots, \overline{c}_{N-1})$. Координаты этой точки представляют собой приближенные значения коэффициентов $\{c_{1,k+1}(\eta)\}_{k=0}^{N-1}$. В данной работе разработан алгоритм и написана программа приближенного вычисления коэффициентов $\overline{c}_k$, $k = \overline{0, N-1}$, используя алгоритм быстрого преобразования Фурье. Путем подстановки найденных коэффициентов в частичную сумму порядка $N-1$ ряда (5), получены значения приближенного решения в узлах сетки $t_j = \frac{j}{N}$, $j = \overline{0, N-1}$. При этом мы также использовали алгоритм быстрого преобразования Фурье.

1. Описание алгоритма

Опишем подробнее алгоритм приближенного вычисления c'_k $(k = \overline{0, N-1})$. Входными данными являются:

- N — порядок частичной суммы ряда Фурье по системе, порожденной косинусами,

- p — первое приближение коэффициентов искомого решения,
- y_0 — начальное значение задачи Коши в точке 0,
- ε — порог для остановки итерационного процесса,
- f — функция двух переменных, удовлетворяющая условию Липшица по второй переменной.

Алгоритм вычисления суммы $\sum_{i=0}^{N-1} c_i \varphi_{1,i+1}(\tau)$ для некоторых заданных $c_i \in \mathbb{R}$ на сетке $\tau_j = \frac{j}{N}$ был описан в работе авторов [4].

Массив $c = \{c'_k\}_{k=0}^{N-1}$ вычисляется итерационным методом. В качестве первого приближения принимается $c^{(0)} = p$. После этого по формуле

$$c_k^{(m+1)} = \frac{h\pi}{N} \sum_{j=0}^{N-1} f\left(h\sin(\pi t_j), \eta(0) + \sum_{i=0}^{N-1} c_i^{(m)} \varphi_{1,i+1}(t_j)\right) \cos(\pi t_j) \varphi_k(t_j) \qquad (9)$$

вычисляются значения $c^{(m+1)}$ пока не выполнится

$$\sqrt{\sum_{k=0}^{N-1} (c_k^{(m+1)} - c_k^{(m)})^2} < \varepsilon.$$

Отметим, что при вычислении $c_k^{(m+1)}$ при $k = 0$ требуется дополнительно умножить коэффициент на $\frac{1}{\sqrt{2}}$, так как $\varphi_0(x) = 1$. Кроме того, сумма (9) представляет собой дискретное косинус-преобразование функции $f\left(h\sin(\pi t_j), \eta(0) + \sum_{i=0}^{N-1} c_i \varphi_{1,i+1}(t_j)\right) \cos(\pi t_j)$ на сетке $\left\{t_j = \frac{j}{N}\right\}_{j=0}^{N-1}$. Используя этот факт, мы можем применить быстрое преобразование Фурье для вычисления (9). Последнее полученное значение $c_k^{(m+1)}$ принимается за массив c. Далее взяв частичную сумму порядка $N-1$ равенства (5) и приняв в качестве значений $c_{1,1+k}$ значения массива c, мы получим приближенное решение задачи Коши

$$\eta(t) \approx \eta(0) + \sum_{k=0}^{N-1} c'_k \varphi_{1,k+1}(t).$$

Обратим внимание, что при $t = t_j$ это приближенное равенство представляет собой обратное синус-преобразование Фурье и может также быть посчитана посредством алгоритма быстрого преобразования Фурье.

2. Листинг программы

```
private static double[] Ettas(IReadOnlyList<double> a)
{
  var n = a.Count;
  var b = new double[2 * n];
  b[0] = 0;
  for (var i = 1; i < n; i++)
  {
    b[i] = a[i] / i;
  }
```

```
  b[n] = 0;
  for (var i = 1; i < n; i++)
  {
    b[2 * n - i] = -a[i] / i;
  }
  var h = new Complex[n];
  for (int i = 0; i < n; i++)
  {
    h[i] = new Complex(b[2 * i], b[2 * i + 1]);
  }

  var a1 = InvFftForRealInput(h);
  var c = new Complex[n];

  for (var i = 0; i < c.Length; i++)
  {
    c[i] = Sqrt(2.0) / PI * a1[i] / ImagNum;
  }

  return c.Select(x => x.Real).ToArray();
}

private static Complex[] InvFftForRealInput(Complex[] a)
{
  var n = a.Length;

  if (_expArray == null || _expArray.Length != n)
  {
    _expArray = new Complex[n];

    for (var k = 0; k < n; k++)
    {
      _expArray[k] = new Complex(Cos(PI * k / n), Sin(PI * k / n));
    }
  }

  var imag = new Complex(0, 1);

  var b = new Complex[n];

  var a1 = ifft(a);

  b[0] = new Complex(a1[0].Real, -a1[0].Imaginary);
  for (var i = 1; i < n; i++)
  {
    b[i] = new Complex(a1[n - i].Real, -a1[n - i].Imaginary);
  }
  var g1 = new Complex[n];
  var g2 = new Complex[n];
  var b1 = new Complex[2 * n];
  for (int i = 0; i < n; i++)
  {
    g1[i] = (a1[i] + b[i]) / 2;
```

```
    g2[i] = -imag * (a1[i] - b[i]) / 2;
  }
  for (var i = 0; i < n; i++)
  {
    b1[i] = g1[i] + _expArray[i] * g2[i];
  }
  return b1;
}

private static Complex[] FftForRealInput(Complex[] a)
{
  var n = a.Length;

  var expArray = new Complex[n];

  for (var k = 0; k < n; k++)
  {
    expArray[k] = new Complex(Cos(PI * k / n), -Sin(PI * k / n));
  }
  var imag = new Complex(0, 1);

  var b = new Complex[n];

  var a1 = fft(a);

  b[0] = new Complex(a1[0].Real, -a1[0].Imaginary);
  for (int i = 1; i < n; i++)
  {
    b[i] = new Complex(a1[n - i].Real, -a1[n - i].Imaginary);
  }
  var g1 = new Complex[n];
  var g2 = new Complex[n];
  var b1 = new Complex[n];
  for (int i = 0; i < n; i++)
  {
    g1[i] = (a1[i] + b[i]) / 2;
    g2[i] = -imag * (a1[i] - b[i]) / 2;
  }
  for (int i = 0; i < n; i = i + 1)
  {
    b1[i] = g1[i] + expArray[i] * g2[i];
  }
  return b1;
}

private static double[] EttasWithInitialValues
                (IReadOnlyList<double> p, double y0)
{
  return Ettas(p).Select(x => x + y0).ToArray();
}

private static double[] _tau;
private static double[] _hsin;
```

```
private static double[] _cos;
private static int[] _indexes;
private static double sqrt2 = Sqrt(2);

private static double tau(int i, int N)
{
  return 1.0 * i / N;
}

// Метод, вычисляющий значения c^(m+1)_k из c^(m)_k
private static double[] Cks
        (IReadOnlyList<double> p, double h,
        double y0, Func<double, double, double> f)
{
  var N = p.Count;
  var ettas = EttasWithInitialValues(p, y0);

  if (_indexes == null || _indexes.Length != N)
  {
    _indexes = Enumerable.Range(0, N).ToArray();

    _tau = _indexes.Select(i => tau(i, N)).ToArray();

    _hsin = _indexes.Select(j => h * Sin(PI * _tau[j]))
      .ToArray();

    _cos = _indexes.Select(j => Cos(PI * _tau[j])).ToArray();
  }

  var dicreteF = _indexes
    .Select(j => f(_hsin[j], ettas[j]) * _cos[j]).ToArray();

  var f1 =
    dicreteF
    .Append(0).Concat(dicreteF.Skip(1).Reverse())
    .ToArray();

  var f1Complex = _indexes
    .Select(i => new Complex(f1[2 * i], f1[2 * i + 1]))
    .ToArray();

  var a1 = FftForRealInput(f1Complex);
  a1[0] = h * PI * (a1[0] + f1[0]) / (2.0 * N);
  for (var i = 1; i < a1.Length; i++)
  {
    a1[i] = h * PI * (a1[i] + f1[0]) / (sqrt2 * N);
  }
  return a1.Select(x => x.Real).ToArray();
}

// Метод, находящий евклидово расстояние между двумя массивами
static double Diff(double[] a, double[] b)
{
```

```
  return Sqrt(a.Zip(b, (x, y) => (x - y) * (x - y)).Sum());
}

// Метод, решающий задачу Коши для входных данных
public static double[] Calculate
  (double[] p, Func<double, double, double> f,
  double h, double y0, double eps)
{
  var N = p.Length;

  for (int i = 0; i < N; i++)
  {
    p[i] = 1 / (i + 1);
  }

  var cks = Cks(p, h, y0, f);
  var k = new double[p.Length];

  do
  {
    k = cks;
    cks = Cks(cks, h, y0, f);
  } while (Diff(cks, k) >= eps);

  return cks.Select(x => y0 + x).ToArray();
}
```

3. Численные эксперименты

На конкретных примерах покажем результат применения вышеприведенной программы.

Пример 1. *Решить задачу Коши*

$$y' = xy, \quad y(0) = 1, \quad 0 \leqslant x \leqslant 1,2.$$

Решение. Точное решение этой задачи имеет вид: $y(x) = e^{\frac{x^2}{2}}$. Полагая $x = 1,2\sin(\pi t)$, $\eta(t) = y(1,2\sin(\pi t))$ и учитывая, что функция $\eta(t)$ симметрична относительно прямой $t = \frac{1}{2}$, исходная задача примет следующий вид:

$$\eta'(t) = 1,2^2\cos(\pi t)\sin(\pi t)\eta(t), \quad \eta(0) = 1, \quad 0 \leqslant t \leqslant 1. \tag{10}$$

Ниже на рисунке 1 приведены первые 4 итерации применения программы к задаче (10) при $N = 64$. Синим цветом обозначено точное решение, красным – приближенное.

Пример 2. *Решить задачу Коши*

$$y' = f(x, y), \tag{11}$$

при

$$f(x,y) = \begin{cases} \sqrt{1-y^2}\sin(x), |y| \leqslant 0,9, \\ \sqrt{1-0,9^2}\sin(x), y \in \mathbb{R} \setminus [-0,9; 0,9], \end{cases} \qquad y(0) = 0, \quad 0 \leqslant x \leqslant 1,5.$$

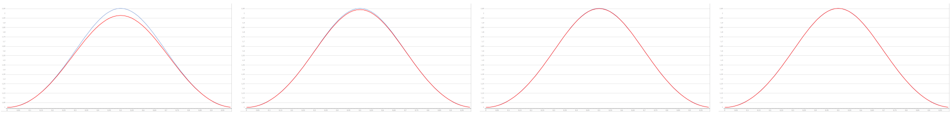

Рис. 1. Первые 4 итерации применения программы к задаче (10) при $N = 64$. Синим цветом обозначено точное решение, красным – приближенное.

Решение. Точное решение этой задачи имеет вид: $y(x) = \sin(1 - \cos(x))$. При $x = 1,5\sin(\pi t)$, $\eta(t) = y(1,5\sin(\pi t))$ и с учетом того, что функция $\eta(t)$ симметрична относительно прямой $t = \frac{1}{2}$, задача (11) примет следующий вид:

$$\eta'(t) = 1,5\cos(\pi t)\sin(1,5\sin(\pi t))\sqrt{1 - \eta^2(t)}, \quad \eta(0) = 0, \quad 0 \leqslant t \leqslant 1. \tag{12}$$

На рисунке 2 приведены первые 3 итерации применения программы к задаче (12) при $N = 64$. Синим цветом обозначено точное решение, красным – приближенное.

Рис. 2. Первые 3 итерации применения программы к задаче (12) при $N = 64$. Синим цветом обозначено точное решение, красным – приближенное.

Список литературы

[1] Шарапудинов И.И., Магомед-Касумов М.Г. Численный метод решения задачи Коши для систем обыкновенных дифференциальных уравнений с помощью ортогональной в смысле Соболева системы, порожденной системой косинусов // Дагестанские электронные математические известия. 2017. Вып. 8. С. 53–60.

[2] Шарапудинов И.И. О приближении решения задачи Коши для нелинейных систем ОДУ посредством рядов Фурье по функциям, ортогональным по Соболеву // Дагестанские электронные математические известия. 2017. Вып. 7. С. 66–76.

[3] Шарапудинов И.И. Системы функций, ортогональные по Соболеву, ассоциированные с ортогональной системой // Изв. РАН. Сер. матем. 2018. 82:1. С. 212–244.

[4] Акниев Г.Г., Гаджимирзаев Р.М. Алгоритм численной реализации полиномов по функциям, ортогональным по Соболеву и порожденным косинусами // Дагестанские электронные математические известия. 2018. Вып. 9. С. 1–6.

Г. Г. Акниев (G. G. Akniyev)
Дагестанский научный центр РАН
E-mail: hasan.akniyev@gmail.com

Р. М. Гаджимирзаев (R. M. Gadzhimirzaev)
Дагестанский научный центр РАН
E-mail: ramis3004@gmail.com

Поступила в редакцию 14.09.2018