

ЛЕКЦИИ МЕЖДУНАРОДНОЙ ШКОЛЫ «KAZCAS-16»

---

**Д.П. Голоскоков**

**Прямые методы математической физики с  
использованием системы Maple**

---



---

Казань, Казанский федеральный университет, 5–7 ноября 2016

УДК 517.958:52/59

## ПРЯМЫЕ МЕТОДЫ МАТЕМАТИЧЕСКОЙ ФИЗИКИ С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ MAPLE

Д.П. Голоскоков<sup>1</sup>

<sup>1</sup> *dpg1954@mail.ru*; Государственный университет морского и речного флота им. адм. С.О. Макарова

*Использование системы символьных вычислений Maple для реализации методов приближенного решения дифференциальных уравнений, связанных с вариационным исчислением.*

**Ключевые слова:** вариационная задача, конечно-разностный метод Эйлера, метод Рунге, задача Дирихле в прямоугольнике, метод Бубнова-Галеркина, конформное отображение, примеры построения конформных отображений.

Под прямыми методами приближенного решения задач математической физики понимают такие методы, которые сводят эти задачи к конечным системам алгебраических уравнений. Такое определение дает С. Г. Михлин в своей книге “Прямые методы в математической физике” (ГИТТЛ, М.-Л., 1950).

Прямые методы широко используются в практике инженерных расчетов, значительна роль этих методов и в теоретических исследованиях.

Известно, что многие проблемы механики наряду с дифференциальными уравнениями управляются так называемыми минимальными принципами. Например, положение равновесия механической системы есть положение, отвечающее минимуму ее потенциальной энергии. Ввиду этого проблема решения граничной задачи для дифференциального уравнения данной механической системы оказывается эквивалентной проблеме нахождения функции, дающей минимум интеграла, которым выражается потенциальная энергия системы.

Во многих случаях краевую задачу для дифференциального уравнения можно заменить равносильной задачей об отыскании функции, доставляющей некоторому интегралу наименьшее значение. Задачи такого типа называются вариационными, а методы, позволяющие свести задачу об интегрировании дифференциального уравнения к равносильной вариационной задаче, называются вариационными методами.

Математически проблема решения краевой задачи для дифференциального уравнения эквивалентна задаче вариационного исчисления — о минимуме интеграла, для которого данное дифференциальное уравнение служит уравнением Эйлера. Для решения задач вариационного исчисления наряду с уравнением Эйлера могут применяться непосредственные, прямые методы, которые обходят применение дифференциальных уравнений; благодаря указанной эквивалентности, эти методы являются одновременно и методами решения краевых задач для дифференциальных уравнений. В основе этих методов лежит идея построить искомую функцию, дающую экстремум функционала, при помощи некоторого предельного процесса, исходя непосредственно из вида того интеграла, экстремум которого ищется. Наиболее известными среди таких методов являются методы Рунге-Ритца и Бубнова-Галеркина.

В настоящей лекции мы покажем использование системы символьных вычислений Maple для реализации методов приближенного решения дифференциальных уравнений, связанных с вариационным исчислением. Использование пакета Maple при решении задач математической физики рассмотрено в книгах автора [1], [2], [3]. В связи с этим хочется также отметить книгу Ю. Г. Игнатьева по математическому моделированию с использованием системы Maple [4].

### 1. Простейшая вариационная задача. Конечно-разностный метод Эйлера

Мы начнем рассмотрение с простейшей вариационной задачи для одномерного интеграла: найти функцию  $y(x)$ , доставляющую экстремум (или просто стационарное значение) функционалу

$$J[y(x)] = \int_a^b F(x, y(x), y'(x)) dx,$$

где  $F$  — заданная функция трех аргументов, непрерывная вместе с производными второго порядка в некоторой области. Значения искомой функции  $y(x)$  на концах промежутка интегрирования заданы:

$$y(a) = y_a, y(b) = y_b.$$

Будем решать задачу одним из прямых методов — конечно-разностным методом Эйлера. Значения функционала  $J[y(x)]$  станем рассматривать не на произвольных, допустимых в данной вариационной задаче кривых, а лишь на ломаных, составленных из заданного числа  $N$  прямолинейных звеньев, с заданными абсциссами вершин

$$a, a+h, a+2h, \dots, a+(N-1)h, a+Nh, h = \frac{b-a}{N}.$$

На этих ломаных функционал превращается в функцию  $\Phi(Y_1, Y_2, \dots, Y_{N-1})$  ординат  $Y_1, Y_2, \dots, Y_{N-1}$  вершин ломаной. Ординаты  $Y_1, Y_2, \dots, Y_{N-1}$  выбираются так, чтобы функция  $\Phi(Y_1, Y_2, \dots, Y_{N-1})$  достигала экстремума, т. е. они определяются из системы уравнений

$$\frac{\partial \Phi}{\partial Y_k} = 0, k = 1, 2, \dots, N-1,$$

ординаты  $Y_0$  и  $Y_N$  известны из граничных условий  $y(a) = y_a, y(b) = y_b$ . Это типичный метод дискретизации, который состоит в приближенной замене функционала  $J[y(x)]$  на сумму

$$I_N = h \sum_{k=0}^{N-1} F\left(X_k, Y_k, \frac{Y_{k+1} - Y_k}{h}\right),$$

зависящую от конечного числа дискретных значений  $Y_k = y(X_k)$  искомой функции в заданных узлах  $X_k = a + kh$ .

**Пример 1.** Найти экстремаль функционала

$$J[y] = \int_{-1}^1 (\cos(x) - 2y'e^x + y'^2) dx; \quad \begin{cases} y(-1) = 2; \\ y(1) = 3. \end{cases}$$

**Решение.** Реализацию намеченного алгоритма решения поставленной задачи выполним в системе аналитических вычислений Maple.

```
> restart;
> interface(displayprecision=3):
```

Аппроксимация подынтегральной функции конечными разностями:

```
> F:=proc(Y,m,h)
  (Y[m+1]-Y[m])^2/h^2-2*(Y[m+1]-Y[m])*exp(X[m])/h+cos(X[m])
end proc:
```

Интеграл заменяем суммой по формуле прямоугольников:

```
> JN:=proc(h,F,N) options operator, arrow;
  h*(sum(F(Y,i,h),i=0..N-1))
end proc:
```

Задаем пределы интегрирования:  $a = -1, b = 1$ .

```
> a := -1: b := 1:
```

Выбираем число узловых точек и определяем шаг интегрирования; сначала решим задачу при небольшом числе интервалов разбиения, например,  $N = 7$ :

```
> N:=7: h:=(b-a)/N:
```

Вычисляем абсциссы вершин ломаной:

```
> for j from 0 to N do X[j] := h*j+a end do:
```

Функционал как функция ординат вершин ломаной:

```
> Phi:=JN(h,F,N):
```

Учет граничных условий:

```
> Y[0]:=2: Y[N]:=3:
```

Составление минимизирующей системы уравнений:

```
> for k to N-1 do
  eq[k]:=evalf(diff(Phi,Y[k]))=0:
end do:
> var:={}: eqns:={}:
  for k to N-1 do
  var:=var union {Y[k]}:
  eqns:=eqns union {eq[k]}:
end do:
```

Решение системы

```
> res:=solve(eqns,var):
> assign(res):
```

Формирование списка точек вершин ломаной:

```
> for j from 0 to N do
  P[j]:=[X[j],Y[j]]
end do:
> L:=[seq(P[k-1],k=1..N+1)]:
```

Для сравнения найдем точное решение задачи:

```
> with(VariationalCalculus):
> f:=(diff(y(x),x))^2-2*(diff(y(x),x))*exp(x)+cos(x):
```

$$f := \left( \frac{dy(x)}{dx} \right)^2 - 2 \left( \frac{dy(x)}{dx} \right) e^x + \cos(x)$$

```
> ode:=EulerLagrange(f,x,y(x));
```

$$ode := \left\{ -2 \left( \frac{dy(x)}{dx} \right)^2 + 2e^x, \left( \frac{dy(x)}{dx} \right) - 2e^x = K_1 \right\}$$

```
> problem:={ode[1]} union {y(-1)=2,y(1)=3};
```

$$problem := \left\{ -2 \left( \frac{dy(x)}{dx} \right)^2 + 2e^x, y(-1) = 2, y(1) = 3 \right\}$$

```
> dsolve(problem,y(x));
```

$$y(x) = e^x + \left( \frac{1}{2} + \frac{1}{2}e^{-1} - \frac{1}{2}e \right) x + \frac{5}{2} - \frac{1}{2}e - \frac{1}{2}e^{-1}$$

```
> simplify(convert(%,trig));
```

$$y(x) = -x \sinh(1) - \cosh(1) + \cosh(x) + \sinh(x) + \frac{1}{2}x + \frac{5}{2}$$

```
> y:=unapply(rhs(%),x);
```

$$y := x \rightarrow -x \sinh(1) - \cosh(1) + \cosh(x) + \sinh(x) + \frac{1}{2}x + \frac{5}{2}$$

Построим графики приближенного и точного решений (рис. 1):

```
> plot([L,y(x)],x=-1..1,
  title=cat("Число интервалов N = ",convert(N,string)),
  titlefont=[Helvetica,15],font=[Helvetica,14],
  linestyle=[dash,solid],gridlines=true,
  legend=["метод Эйлера","Точное решение"]);
```

Численное решение удобно представить в виде таблицы значений:

```
> fd_table1:=
  eval([seq([X[k],Y[k],evalf(y(X[k]))],k=0..N)],res):
> z:=['x','Y','y'];
> linalg[stackmatrix](z,fd_table1);
```

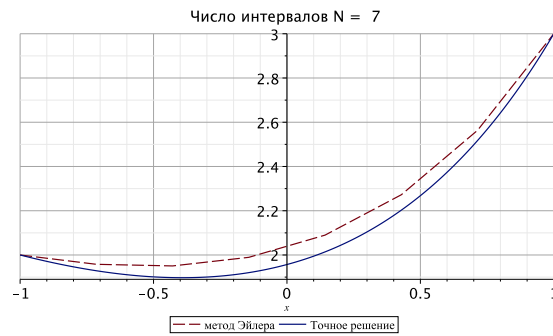


Рис. 1. Сравнение приближенного и точного решений при  $N = 7$ .

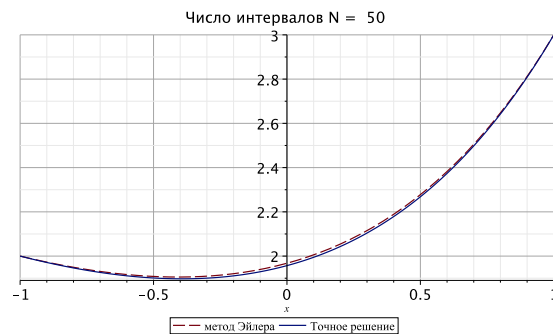


Рис. 2. Сравнение приближенного и точного решений при  $N = 50$ .

$x$	$Y$	$y$
-1	2	2.000
$-\frac{5}{7}$	1.958	1.929
$-\frac{3}{7}$	1.951	1.898
$-\frac{1}{7}$	1.989	1.920
$\frac{1}{7}$	2.090	2.014
$\frac{3}{7}$	2.272	2.203
$\frac{5}{7}$	2.564	2.517
1	3	3.000

Уточнить результат можно, увеличив  $N$ . Например, на рис. 2 показан результат при  $N = 50$ .

Как видим, результат не плох. Однако отметим, что приведенные рассуждения имеют скорее методическое значение, так как в вычислительном отношении процедура получается в точности такая же, как если от вариационной задачи перейти к уравнению Эйлера для исследуемого функционала, а затем решать для этого уравнения краевую задачу методом сеток.

## 2. Простейшая вариационная задача. Метод Ритца

Метод Ритца представляет собой один из методов построения минимизирующей последовательности.

Пусть функция  $u(P)$  удовлетворяет в некоторой области  $\Omega$  неоднородному урав-

нению

$$Au = f(P), \quad (1)$$

и некоторым однородным граничным условиям.

Мы знаем, что решение уравнения (1), где  $A$  — положительный оператор, сводится к нахождению минимума функционала

$$F[u(P)] = (Au, u) - 2(u, f) = \int_{\Omega} [u(P) Au(P) - 2u(P) f(P)] d\Omega \quad (2)$$

Эту последнюю задачу будем приближённо решать следующим образом. Выберем последовательность функций

$$\varphi_1(P), \varphi_2(P), \dots, \varphi_n(P), \dots, \quad (3)$$

принадлежащих области определения оператора  $A$ . Подчиним эти функции двум условиям: 1) последовательность (3) полна по энергии; 2) для любого  $n$  функции  $\varphi_1(P), \varphi_2(P), \dots, \varphi_n$  линейно независимы. Функции (3) называют *координатными*, или *базисными* функциями.

Возьмем линейную комбинацию первых  $n$  координатных функций:

$$u_n(P) = \sum_{j=1}^n a_j \varphi_j(P), \quad (4)$$

где  $a_j$  — произвольные числа.

Подставим (4) вместо  $u(P)$  в функционал (2); это превратит функционал  $F[u(P)]$  в функцию  $n$  независимых переменных  $a_1, a_2, \dots, a_n$ :

$$F[u_n] = \sum_{j=1}^n \sum_{k=1}^n (A\varphi_j, \varphi_k) a_j a_k - 2 \sum_{j=1}^n (\varphi_j, f) a_j \quad (5)$$

Выберем коэффициенты  $a_j$  так, чтобы функция (5) приняла минимальное значение. Необходимые условия минимума функции (5), как известно, такие:

$$\frac{\partial F[u_n]}{\partial a_i} = 0, \quad i = 1, 2, \dots, n. \quad (6)$$

Отметим, что используя положительность оператора  $A$ , можно доказать, что коэффициенты  $a_j$ , удовлетворяющие системе (6), реализуют минимум величины  $F[u_n]$ . Выпишем явно систему (6)

$$\sum_{k=1}^n (A\varphi_i, \varphi_k) a_k = (\varphi_i, f), \quad i = 1, 2, \dots, n. \quad (7)$$

Определитель системы (7) есть так называемый определитель Грамма линейно независимых функций, поэтому он отличен от нуля. Отсюда следует, что система (7) всегда разрешима, если оператор  $A$  — положительный. Найдя коэффициенты  $a_k$  из системы (7) и подставив их в (4), получим функцию  $u_n(P)$ , которую называют приближенным решением уравнения  $Au = f(P)$  по Ритцу.

Справедлива **теорема**: приближённые решения уравнения (1) по Ритцу образуют минимизирующую последовательность для функционала (2), если только это уравнение имеет решение с конечной энергией.

Следовательно, приближённые решения по Ритцу сходятся по энергии к точному решению. Они сходятся к точному решению и в среднем, если  $A$  — положительно определённый оператор.

**Пример 2.** Для сравнения решим методом Ритца задачу из примера 1.

```
> restart;
```

Выбираем базисные функции и определяем аппроксимирующую функцию

```
> phi0:=x->y1+(y2-y1)*(x-x1)/(x2-x1):
> phi:=(x,n)->sin(n*Pi*(x-x1)/(x2-x1)):
> Us:=proc(x,N)option operator,arrow; local n;
  phi0(x)+sum(a[n]*phi(x,n),'n'=1..N);
end proc:
```

Разработаем процедуру для составления уравнений метода Ритца:

```
> Ritz:=proc(F,u,i0,N,a)local Fu,eqns,var,eq,i,res;
  global x1,x2;
  Fu:=simplify(int(subs(y(x)=u,F),x=x1..x2));
  eqns:={}:var:={}:
  for i from i0 to N do
    var:=var union {a[i]}:
    eq[i]:=diff(Fu,a[i])=0:
    eqns:=eqns union {eq[i]}:
  od:
  res:=solve(eqns,var);
  assign(res);
end proc:
> interface(displayprecision=3):
```

Вводим граничные точки, задаем количество членов ряда, задаем подинтегральную функцию и выполняем расчеты:

```
> x1:=-1:x2:=1:y1:=2:y2:=3:
> N:=5:c1:='cross':c2:='circle':
  c3:='box':c4:='point':c5:='diamond':
  y:='y':
  F:=(diff(y(x),x))^2-2*(diff(y(x),x))*exp(x)+cos(x):
> for j from 1 to N do
  a:=array(1..j):
  Ritz(F,Us(x,j),1,j,a);
  pu_||j:=plot(Us(x,j),x=x1..x2,color=blue,
  style=point,symbol=c||j,
  legend=cat('Метод Ритца N = ',convert(j,string)),
  gridlines=true,font=[Helvetica,14],
```

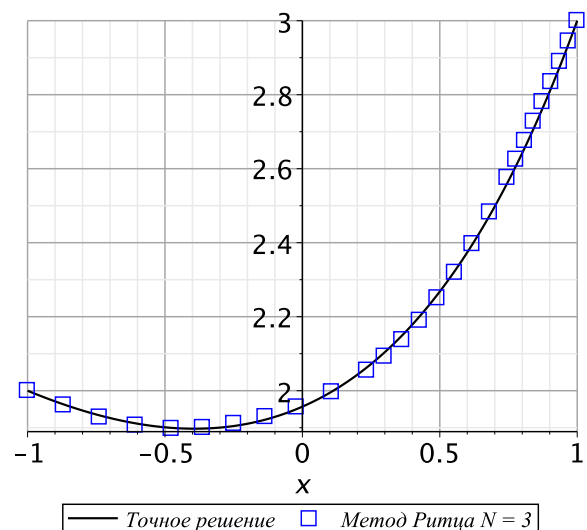
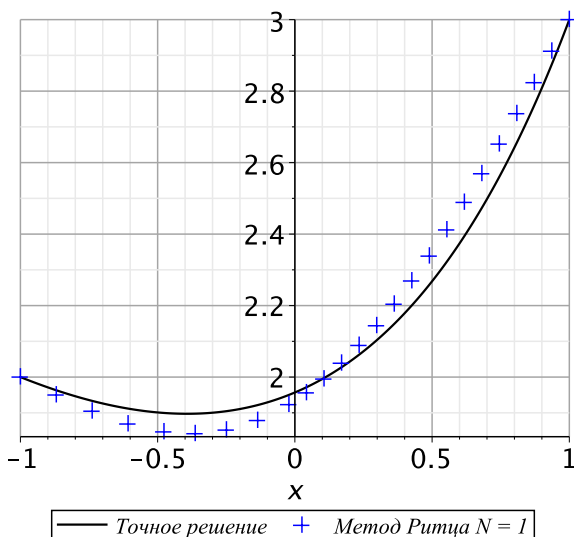
```
labelfont=[Helvetica,14],symbolsize=20,numpoints=5):
end do:
```

Точное решение мы уже получили в примере 1; определим это решение в виде процедуры-функции:

```
> y:=proc(x) options operator,arrow;
-x*sinh(1)-cosh(1)+cosh(x)+sinh(x)+(1/2)*x+5/2
end proc;
```

Построим графики приближенных решений по Рунту при разном количестве удерживаемых членов и точного решения (рис. 3 - 5):

```
> pr3:=plot(y(x),x=-1..1,
  legend='Точное решение',color=black):
> plots[display]({pu_1,pr3});
```



**Рис. 3.** Сравнение приближенного и точного решений при  $N=1$ .

**Рис. 4.** Сравнение приближенного и точного решений при  $N=3$ .

```
> plots[display]({pu_3,pr3});
> plots[display]({pu_5,pr3});
```

Для количественного сравнения решений удобно иметь таблицу значений в конкретных точках. Подготовим такую таблицу:

```
> M := 7:
> h := (x2-x1)/M:
> for j from 0 to M do
  X[j]:=h*j+x1:Y[j]:=evalf(Us(X[j],5)):
end do:
> ftab:=eval([seq([X[k],Y[k],evalf(y(X[k]))],k=0..M)]):
> z:=['x','Y','y']:
> linalg[stackmatrix](z,ftab);
```

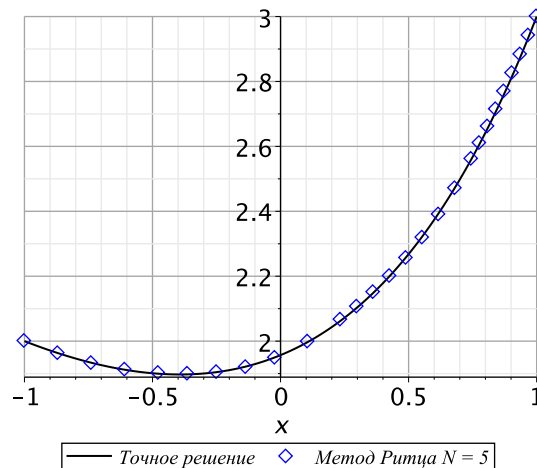


Рис. 5. Сравнение приближенного и точного решений при  $N = 5$ .

$x$	$Y$	$y$
-1	2.000	2.000
$-\frac{5}{7}$	1.927	1.929
$-\frac{3}{7}$	1.900	1.898
$-\frac{1}{7}$	1.918	1.920
$\frac{1}{7}$	2.016	2.014
$\frac{3}{7}$	2.202	2.203
$\frac{5}{7}$	2.516	2.517
1	3	3.000

Как видим, здесь результат лучше, чем в примере 1 — лучшее соответствие точному решению при значительно меньшем числе  $N$ .

### 3. Задача Дирихле в прямоугольнике. Метод Бубнова-Галеркина

Пусть функция  $u(P)$  удовлетворяет в некоторой области  $\Omega$  неоднородному уравнению (1) и некоторым однородным граничным условиям. Здесь мы не будем предполагать положительность оператора  $A$ . Выберем бесконечную последовательность координатных функций (3), которые достаточное число раз (в соответствии с данными задачи) непрерывно дифференцируемы в замкнутой области  $\bar{\Omega} = \Omega + S$ , где  $S$  — граница области  $\Omega$ . Будем считать, что эти функции удовлетворяют всем краевым условиям задачи. Будем считать также, что как уравнение (1), так и краевые условия — линейные. Составим функцию (4), где  $a_j$  — произвольные числа. Эта функция удовлетворяет всем краевым условиям задачи. Подставим теперь (4) в (1); получим невязку в выполнении уравнения

$$Au_n - f(P) = 0. \quad (8)$$

По методу Бубнова-Галёркина коэффициенты  $a_j$  определяются из требования, чтобы левая часть (8) была ортогональна функциям  $\varphi_1(P), \varphi_2(P), \dots, \varphi_n$ . Таким образом, метод Бубнова-Галеркина приводит к системе линейных алгебраических уравнений (7), по виду тождественной системе, к которой приводит метод Рунца.

Отсюда заключаем, что методы Бубнова-Галеркина и Ритца совпадают, если оператор  $A$  — положительно определённый. В общем же случае метод Ритца неприменим, тогда как метод Бубнова-Галеркина сохраняет силу.

Рассмотрим следующую задачу Дирихле для уравнения Пуассона: требуется найти функцию  $u(x, y)$ , удовлетворяющую уравнению Пуассона  $-\Delta u = f(x, y)$  в прямоугольнике и условию  $u|_{\Gamma} = 0$  на границе  $\tilde{A}$  этого прямоугольника — граничному условию первого рода. Эта задача встречается во многих приложениях математической физики. В механике жидкости и газа она описывает стационарное течение вязкой жидкости в канале с поперечным сечением в форме прямоугольника; в электростатике — потенциал электростатического поля; в теории теплопроводности — стационарное распределение температуры; в статике упругих тел — равновесие прямоугольной мембраны, кручение стержня с поперечным сечением в форме прямоугольника и многие другие процессы.

**Пример 3.** Рассмотрим модельную задачу о стационарном течении вязкой жидкости в канале с поперечным сечением в форме квадрата. Уравнение для компоненты скорости, направленной вдоль канала, приведенное к безразмерной форме записи за счет надлежащего выбора масштабов, имеет вид  $\Delta u = -1$ ; граничные условия —  $u|_{x=\pm 1} = 0$ ,  $u|_{y=\pm 1} = 0$ .

**Решение.** Продемонстрируем применение метода Бубнова-Галеркина к решению этой задачи в системе Maple.

```
> restart;
> interface(displayprecision = 5):
```

Уравнение Пуассона определим как:

```
> pde:=diff(u(x,y),x$2)+diff(u(x,y),y$2)+1;
```

$$pde := \frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) + 1$$

Подготовим процедуру, реализующую алгоритм метода Бубнова-Галеркина:

```
> Bubnov_Gal := proc(pde, n, a, bol :: boolean)
  local eqns, vars, sol, i;
  eqns := {}; vars := {};
  for i from 1 to n do
    eqns := eqns union
    {int(eval(pde)*phi(i, x, y), [x=-1..1, y=-1..1])};
    vars := vars union {a[i]};
  end do;
  if bol then print(eval(eqns));
  print(eval(vars)); end if;
  sol := solve(eqns, vars);
  if bol then print(sol); end if;
  assign(sol);
end proc;
```

Определяем полиномиальные базисные функции, удовлетворяющие граничным условиям:

> phi := (n, x, y) -> (1 - x^2)^n \* (1 - y^2)^n;

$$\phi := (n, x, y) \rightarrow (1 - x^2)^n (1 - y^2)^n$$

Определяем аппроксимирующую функцию:

> W := (x, y, N) -> sum(a[i]\*phi(i, x, y), 'i' = 1..N);

$$W := (x, y, N) \rightarrow \sum_{i=1}^N a_i \phi(i, x, y)$$

Задаем количество членов в аппроксимации и выполняем расчет

> N := 2:

> eq := value(subs(u(x, y) = W(x, y, N), pde));

> Bubnov\_Gal(eq, N, a, true);

$$\left\{ -\frac{131072}{33075}a_2 - \frac{2048}{525}a_1 + \frac{256}{225}, -\frac{2048}{525}a_2 - \frac{256}{45}a_1 + \frac{16}{9} \right\}$$

$$\{a_1, a_2\}$$

$$\left\{ a_1 = \frac{37}{104}, a_2 = -\frac{105}{1664} \right\}$$

Вычислим скорость на оси канала и безразмерный расход (расход определяется как интеграл от функции скорости по области сечения)

> value(W(x, y, N)); evalf(W(0, 0, N));

0.29267

> Int(Int(u(x, y), x = -1..1), y = -1..1) =  
evalf(Int(Int(W(x, y, N), x = -1..1), y = -1..1));

$$\int_{-1}^1 \int_{-1}^1 u(x, y) dx dy = 0.56068$$

Результат совпадает с книгой К.Флетчера “Численные методы на основе метода Галеркина”, стр. 26, таб. 1.6. Точные значения приведены в книге [1]:

$$u(0, 0) = 0.2947, \int_{-1}^1 \int_{-1}^1 u(x, y) dx dy = 0.5623.$$

С помощью следующих команд изобразим несколько изолиний (см. рис. 6):

```
> W_b := value(W(x, y, N)):
myskyblue := ColorTools:-Color([0.8, 0.9, 1.]):
pic3 := implicitplot([W_b=0.01, W_b=0.1, W_b=0.2, W_b=0.25],
x=-1..1, y=-1..1, grid=[50, 50], color=khaki,
font=["Helvetica", 14], labelfont=["Courier", 14]),
polygonplot(ngon, font=["Helvetica", 14], color=myskyblue):
pict := textplot({[0., 1.1, "u = 0.01"],
[0., 0.9, "u = 0.1"], [0., 0.7, "u = 0.2"],
[0., 0.5, "u = 0.25"]}):
display([pic3, pict], axes=frame);
```

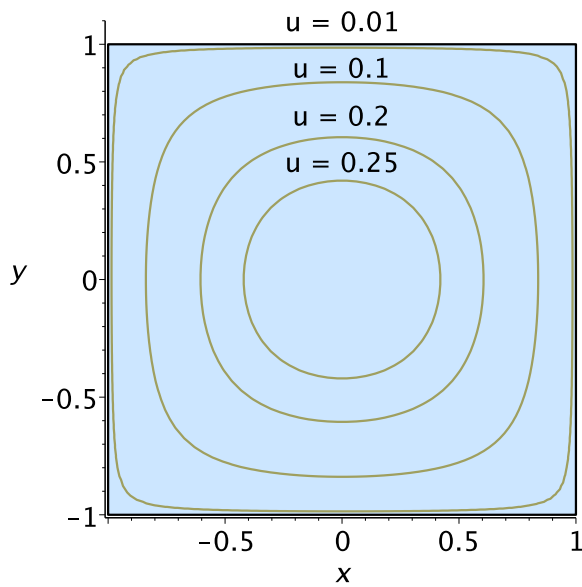


Рис. 6. Изолинии скорости.

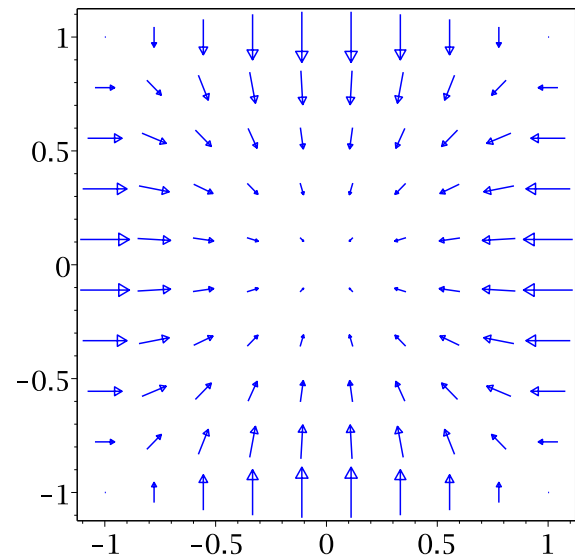


Рис. 7. Градиент векторного поля.

Изобразим график градиента векторного поля (см. рис. 7):

```
> plots[gradplot](W_b,x=-1..1,y=-1..1,grid=[10,10],style=PATCHNOGRID,
  arrows=SLIM,color=blue,axes=BOXED,symbolsize=14,
  font=[Times,roman,14],labelfont=[Helvetica,roman,14]);
```

#### 4. Конформное отображение. Метод Ритца

Конформные отображения имеют многочисленные приложения в механике, физике и технике, прежде всего для расчетов плоских гармонических векторных полей в гидро- и аэродинамике, теории фильтрации, теории электрических и магнитных полей, теории теплопередачи, теории упругости. Многие стационарные задачи математической физики сводятся к решению простейшего из уравнений эллиптического типа — уравнения Лапласа.

Искомая функция, решающая поставленную проблему, определяется областью, в которой она ищется, и граничными условиями, которые должны быть выполнены на контуре этой области. Для простых областей таких, как круг, прямоугольник, круговое кольцо и т. п., задача может быть сравнительно просто решена при относительно сложных граничных условиях. Если же область имеет сложное строение, то решение задачи непосредственно для этой области представляется затруднительным даже для такой простой задачи, как, например, задача Дирихле.

Один из подходов к решению такой задачи — попытаться предварительно преобразовать заданную область к простейшему виду. При таком преобразовании будет меняться, вообще говоря, область, для которой ищется решение, граничные условия, и дифференциальное уравнение, которому должна удовлетворять искомая функция. Поэтому, большое значение будет, очевидно, иметь такое преобразование, относительно которого само дифференциальное уравнение остается инвариантным. Как известно, уравнение Лапласа остается инвариантным относительно конформных преобразований, осуществляемых аналитическими функциями ком-

плексной переменной.

Задача построения функции, отображающей одну область на другую, даже в простом случае, когда одна из областей является кругом, часто вызывает непреодолимые затруднения. Поэтому вызывают интерес методы приближенного построения отображающей функции. Как правило, пытаются аппроксимировать искомую функцию линейной комбинацией хорошо изученных функций. Простейшими и часто используемыми функциями являются многочлены. Известно, что всегда можно построить многочлен, как угодно мало отличающийся от функции (в любой внутренней части области), отображающей область на круг. Кроме того, если граница области простая кривая, то можно построить многочлен, отличающийся от отображающей функции как угодно мало всюду в области, включая и границу. Известно, что среди функций, осуществляющих конформное преобразование заданной односвязной однолистной области  $D$  на некоторую другую область, функция, отображающая на круг, обладает экстремальными свойствами: 1) свойством минимума площади при преобразовании области на круг, 2) свойством минимума длины контура при преобразовании области на круг. Указанные свойства можно использовать при построении вариационного метода приближенного конформного преобразования области на круг.

#### 4.1. Свойство минимума площади при преобразовании области на круг.

Рассмотрим функцию

$$\zeta = f(z) = a_0 + a_1 z + a_2 z^2 + \dots \quad (9)$$

Пусть требуется построить конформное отображение однолистной области на круг. При этом потребуем, чтобы заданная внутренняя точка области (всегда можно считать эту точку началом координат) переходила в центр круга. Потребуем также, чтобы определенное направление в этой точке, принимаемое за положительное направление вещественной оси, совпадало бы с положительным направлением вещественной оси преобразованной области. Очевидно, эти условия равносильны требованиям:

$$f(0) = 0, f'(0) = a_1 > 0.$$

Можно считать, что  $a_1 = 1$ ; для этого достаточно рассмотреть функцию  $f(z)/a_1$ , которая будет так же давать отображение  $D$  на круг, но только может быть другого радиуса, что не существенно. Поэтому будем предполагать:

$$f(0) = 0, f'(0) = 1. \quad (10)$$

Функцию, дающую требуемое конформное преобразование, можно искать как решение следующей вариационной задачи: среди функций, аналитических в области  $D$  и удовлетворяющих условиям (10), найти ту, которая дает минимум функционалу

$$I = \int_D f'(z) \overline{f'(z)} dx dy \quad (11)$$

по сравнению со всеми другими функциями, которые удовлетворяют требованиям (10). Функционал (11) определяет площадь преобразованной области. Из-за трудностей, возникающих при реализации данного метода, полноценно использовать

его стало возможным лишь с появлением систем символьных вычислений таких, как Maple или Mathematica.

#### 4.2. Свойство минимума длины контура при преобразовании области на круг

Рассмотрим всевозможные конформные преобразования области  $D$ , совершаемые при помощи аналитических в области  $D$  функций  $\zeta = f(z)$ , удовлетворяющих условиям (10). Вариационную задачу можно сформулировать так: среди функций, аналитических в области  $D$  и удовлетворяющих условиям (10), найти ту, которая дает минимум функционалу

$$I = \int_L |f'(z)| ds. \quad (12)$$

Функционал (12) выражает длину контура  $L$ .

#### 4.3. Применение метода Рунца

Для решения задачи минимизации функционалов (11) и (12) будем использовать метод Рунца. При вычислениях по методу Рунца зачастую получают плохо обусловленные системы уравнений. Приближенное решение таких систем известными методами приводит к большим погрешностям, что делает результат вычислений бесполезным. Однако применение систем символьных вычислений позволяет эффективно использовать метод, устранить проблемы, связанные с плохой обусловленностью систем, и получить достаточно точное решение задачи. Среди таких систем символьных вычислений стоит отметить две наиболее мощные системы — это Maple и Mathematica.

Заметим, что используя свойство минимума площади при отображении области на круг, можно искать не непосредственно функцию  $f(z)$ , а ее производную  $f'(z)$ , так как под знак интеграла (11) функция  $f(z)$  не входит, и по найденной производной  $f'(z)$  восстановить функцию  $f(z)$  по формуле

$$f(z) = \int_0^z f'(z) dz.$$

Используя свойство минимума длины контура при отображении области на круг, при вычислениях удобнее искать приближения не непосредственно к производной  $f'(z)$ , в виду линейности подынтегрального выражения относительно  $|f'(z)|$ , а к величине  $\sqrt{f'(z)}$ . Допустимыми будут все функции  $\varphi(z)$ , обращающиеся в единицу в начале координат. Среди них будем искать ту, которая дает наименьшее значение функционалу

$$I = \int_L |\varphi(z)|^2 ds = \int_L \varphi(z) \overline{\varphi(z)} ds \quad (13)$$

Если удастся найти решение вариационной задачи для функционала (13), то

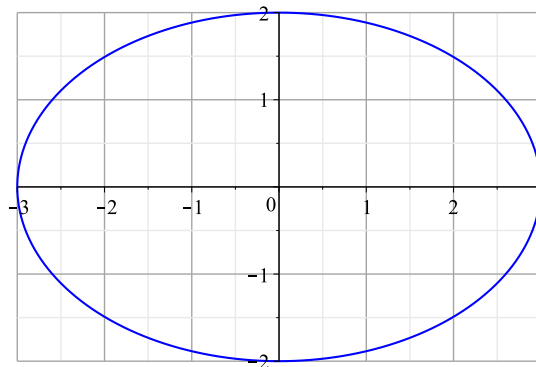


Рис. 8. Эллипс.

функция совершающая искомое отображение, определится по формуле

$$f(z) = \int_0^z \varphi^2(z) dz.$$

## 5. Примеры построения конформных отображений

Рассмотрим применение метода Ритца на конкретных примерах в системе Maple [25].

**Задача 1.** Приблизительно построить конформное отображение эллипса с полуосями  $a = 3, b = 2$ ; начало координат — в точке пересечения осей эллипса (рис. 8). Все вычисления выполняем в Maple. Подключаем необходимые пакеты.

```
> with(VectorCalculus); with(Optimization)
```

Для наглядности можно изобразить отображаемую область (рис. 8); в данном случае это — эллипс. Набираем команду:

```
> plot([3*cos(t), 2*sin(t), t=0..2*Pi], color=blue, scaling=constrained,
      gridlines=true);
```

После нажатия клавиши “Enter” на экране монитора появится изображение эллипса (рис. 8). Задаем степень полинома  $n$  и выражения для  $f'(z)$  и  $\bar{f}'(z)$  — переменные  $P$  и  $P1$ , вычисляем подынтегральную функцию в функционале (11) — переменная  $Q$ .

```
n := 4:
P := 1+sum((a[i]+I*b[i])*(x+I*y)^i, i = 1 .. n):
P1 := 1+sum((a[i]-I*b[i])*(x-I*y)^i, i = 1 .. n):
Q := expand(P*P1):
```

Вычисляем функционал (11)

```
> J := int(Q, [x, y] = Ellipse((1/9)*x^2+(1/4)*y^2-1));
```

Минимизируем функционал:

```
> res := Minimize(J):
```

Результат выполнения последней команды появляется на экране монитора:

```
res := [16.7866843612400, [a1= 0., a2= -0.984942061510830e - 1, a3= 0.,
a4= 0.437728717667569e - 2, b1= 0., b2= 0., b3= 0., b4= 0.]]
```

Далее надо присвоить вычисленные значения коэффициентам  $a_i$  и  $b_i$ ; это осуществляется командами

```
> op(2, res): assign(%):
```

Таким образом, найдена производная отображающей функции  $\overline{f'(z)}$  — обозначим ее  $p1$ :

```
> p1 := 1+sum((a[i]+I*b[i])*z^i, i = 1 .. n);
```

$$p1 := 1 + (-0.0984942061510830 + 0.I)z^2 + (0.00437728717667569 + 0.I)z^4$$

Восстанавливаем отображающую функцию

```
> p := int(p1, z = 0 .. z);
```

$$p := z - 0.03283140205z^3 + 0.0008754574353z^5$$

Для последующего использования удобно преобразовать полученное выражение в функцию

```
> p := unapply(p, z):
```

и функцию — отображение на единичный круг

```
> w := unapply(p(z)/abs(p(3)), z):
```

Для проверки качества полученного решения параметризуем границу области  $D$  и построим график ее образа при отображении функцией  $w(z)$  (рис. 9)

```
> z1 := 3*cos(t)+(2*I)*sin(t):
```

```
pic1 := plot([evalc(Re(w(z1))), evalc(Im(w(z1))), t = 0 .. 2*Pi]):
```

```
pic2 := plot([cos(t), sin(t), t = 0 .. 2*Pi], color = blue):
```

```
plots[display]({pic1, pic2})
```

На рис. 9 голубым цветом показан точный единичный круг, красным цветом — приближение к нему. Как видим, уже многочлен пятой степени дает достаточно точное приближение к кругу. Можно оценить и погрешность вычислений

```
> assume(t, real):
```

```
absw1 := abs(expand(w(z1))):
```

```
Maximize(abs(absw1-1), {0 <= t, t <= 2*Pi});
```

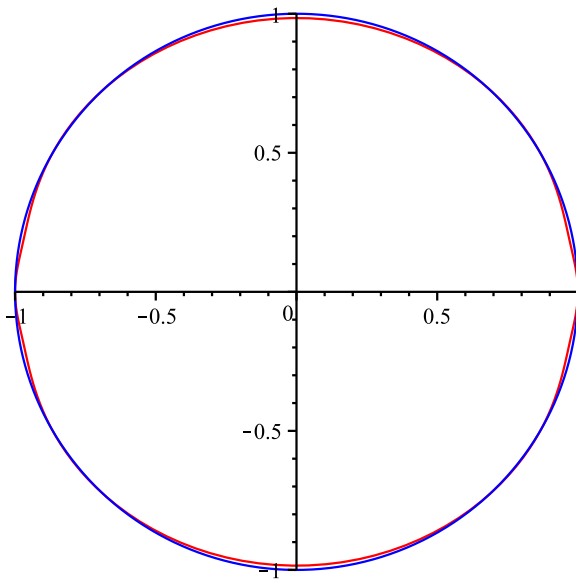


Рис. 9. Круг и образ эллипса.

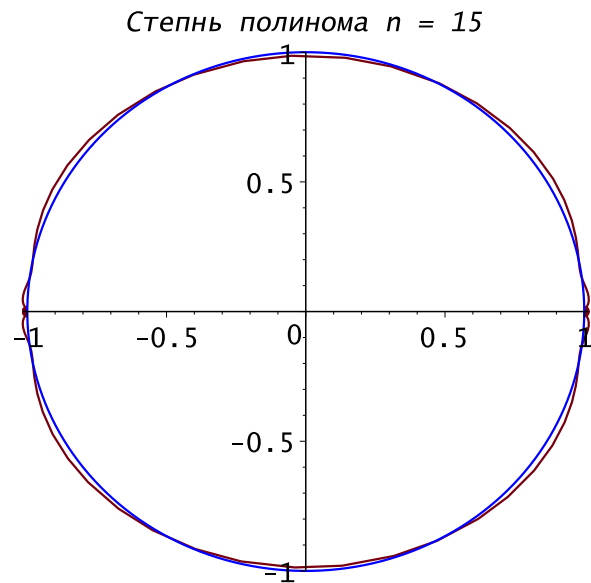


Рис. 10. Отображение эллипса с эксцентриситетом 0.96825.

$$[0.181516596201847236e-2, [t = 1.02918393864730]]$$

Погрешность в данном случае составляет не более, чем  $\varepsilon = 0.002$ .

Следует отметить, что точность и трудоемкость решения задачи зависит от характера отображаемой области. Так для отображения на единичный круг эллипса с большим эксцентриситетом, например, с полуосями  $a = 4, b = 1$  и эксцентриситетом  $\eta = \frac{\sqrt{a^2 - b^2}}{a} = \frac{\sqrt{15}}{4} \approx 0.96825$  приходится использовать комплексный многочлен более высокой степени. На рис. 10 показан результат решения задачи при степени многочлена  $n = 15$ . Погрешность при этом составляет не более чем  $\varepsilon = 0.02$ , т.е. на порядок хуже, чем в предыдущем случае.

**Задача 2.** Приблизительно построить конформное отображение пятиугольника с вершинами в точках  $(-3, -1), (1, 3), (4, 0), (4, -2), (0, -3)$ . Изобразим отображаемую область. Строим пятиугольник по его вершинам (рис. 11).

```
> display(polygon([[[-3, -1], [1, 3], [4, 0], [4, -2], [0, -3]]]), color=gray,
  thickness=2);
```

Задаем степень полинома  $n$  и выражения для  $f'(z)$  и  $\overline{f'(z)}$  — переменные  $P$  и  $P1$ , вычисляем подынтегральную функцию в функционале (11) — переменная  $Q$ . Учитывая замечание о точности и трудоемкости решения задачи в конце предыдущего примера, зададимся степенью полинома  $n = 40$  и увеличим число значащих цифр при выполнении расчетов  $Digits = 70$ .

```
> n := 40: Digits := 70:
  P := expand(1+sum((a[i]+I*b[i])*(x+I*y)^i, i = 1 .. n)):
  P1 := expand(1+sum((a[i]-I*b[i])*(x-I*y)^i, i = 1 .. n)):
  Q := expand(P*P1):
```

Будем вычислять функционал как сумму трех интегралов по трем треугольным областям (рис. 12):

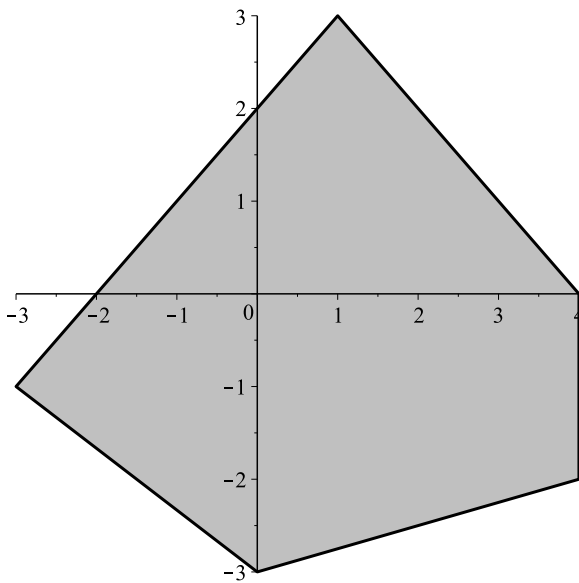


Рис. 11. Пятиугольник.

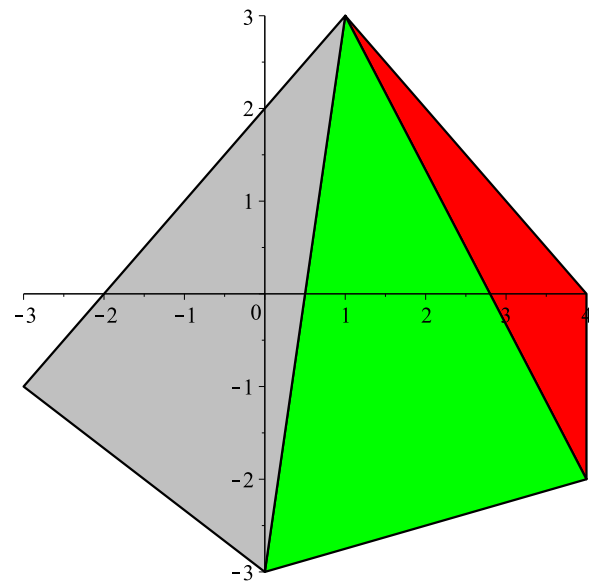


Рис. 12. Разбиение пятиугольника на треугольные области.

```
> T1 := polygon([[ -3, -1], [1, 3], [0, -3]], color = gray):
  T2 := polygon([[1, 3], [4, -2], [0, -3]], color = green):
  T3 := polygon([[1, 3], [4, 0], [4, -2]], color = red):
  display({T1, T2, T3});
```

Вычисляем функционал (11)

```
> J1:=int(Q,[x,y]=Triangle('<,>'(-3,-1), '<,>'(0,-3), '<,>'(1,3))):
  J2:=int(Q,[x,y]=Triangle('<,>'(0,-3), '<,>'(4,-2), '<,>'(1,3))):
  J3:=int(Q,[x,y]=Triangle('<,>'(1,3), '<,>'(4,-2), '<,>'(4,0))):
  J :=J1+J2+J3:
```

Минимизируем функционал и восстанавливаем отображающую функцию

```
> res := Minimize(J);
  op(2, res);
  assign(%);
  p1 := 1+sum((a[i]+I*b[i])*z^i, i = 1 .. n);
  p := int(p1, z = 0 .. z);
  p := unapply(p, z);
  w := unapply(p(z)/abs(p(4)), z);
```

Параметризуем границу области  $D$  и построим график ее образа при отображении функцией  $w(z)$  (рис. 13):

```
> z1:=t+I*(t+2):z2:=t+I*(4-t):z3:=4+I*t:z4:=t+I*((1/4)*t-3):
  z5:=t-I*(2*t*(1/3)+3):
  pic1:=plot([evalc(Re(w(z1))),evalc(Im(w(z1))),t=-3..1]):
  pic2:=plot([evalc(Re(w(z2))),evalc(Im(w(z2))),t=1..4]):
  pic3:=plot([evalc(Re(w(z3))),evalc(Im(w(z3))),t=-2..0]):
```

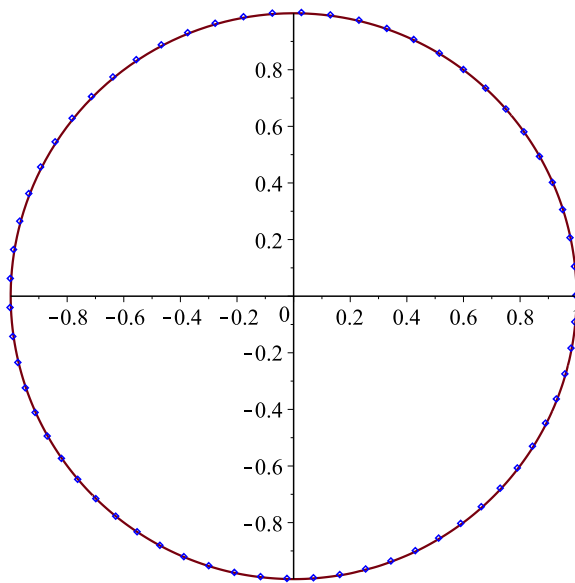


Рис. 13. Образ пятиугольника.

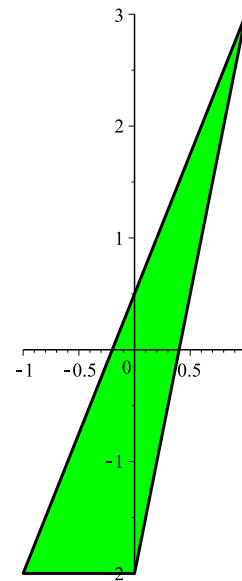


Рис. 14. Вытянутый треугольник.

```
pic4:=plot([evalc(Re(w(z4))),evalc(Im(w(z4))),t=0..4]1):
pic5:=plot([evalc(Re(w(z5))),evalc(Im(w(z5))),t=-3..0]1):
PIC:=plot([cos(t),sin(t),t=0..2*Pi],color=blue,style=point,numpoints=1):
plots[display]({PIC, pic1, pic2, pic3, pic4, pic5});
```

Как видно из рис. 13, аппроксимирующая кривая практически совпадает с окружностью! Аналогично примеру 1 можно оценить и погрешность вычислений. В данном случае она не превышает  $\varepsilon = 0.0005$ . Обратим внимание: для аппроксимации конформного отображения использовался полином степени  $n = 40$ ; при этом расчеты выполнялись с числом значащих цифр  $Digits = 70$ .

В заключение приведем процедуру построения комплексного полинома, осуществляющего конформное отображение треугольника на круг и использующую свойство минимума длины контура:

```
> ConformPolTriangl:=proc(n,A1,B1,C1) local P,P1,Q,J1,J2,J3,J,res,A,B;
global p,w;
P:=1+sum((A[i]+I*B[i])*(x+I*y)^i,i=1..n);
P1:=1+sum((A[i]-I*B[i])*(x-I*y)^i,i=1..n);
Q:=expand(P*P1);
J1:=PathInt(Q,[x,y]=Line(A1,B1));
J2:=PathInt(Q,[x,y]=Line(B1,C1));
J3:=PathInt(Q,[x,y]=Line(C1,A1));
J:=J1+J2+J3;res:=Minimize(J);op(2,res);assign(%);
P:=subs(x+I*y=z,P);
p:=simplify(int(P^2,z=0..z));
p:=unapply(p,z);w:=unapply(p(z)/abs(p(A1[1]+I*A1[2])),z)
end proc;
```

С помощью этой процедуры построим отображение вытянутого треугольника на круг (рис. 14). Задаем вершины треугольника:

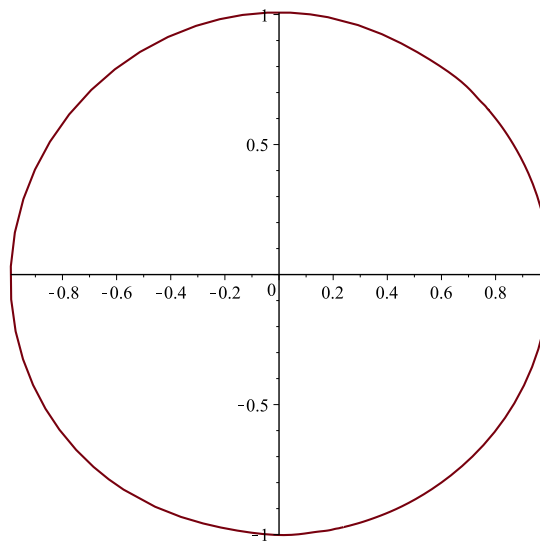
```
> x1 := -1; x2 := 1: x3 := 0: y1 := -2: y2 := 3: y3 := -2:
  A1 := '<,>'(x1, y1): B1 := '<,>'(x2, y2): C1 := '<,>'(x3, y3):
```

Изобразим этот треугольник с помощью команды (рис. 14):

```
> display(polygon([[[-1, -2],[1, 3],[0, -2]]], color=green, thickness=2,
  scaling=constrained);
```

Решаем задачу:

```
> Digits:=100:
  ConformPolTriangl(60, A1, B1, C1):
```



**Рис. 15.** Образ треугольника.

Параметризуем границу области  $D$  и построим график ее образа при отображении функцией  $w(z)$  (рис. 15):

```
> z1:=x1+I*y1+t*(x2-x1+I*(y2-y1)):
  z2:=x2+I*y2+t*(x3-x2+I*(y3-y2)):
  z3:=x3+I*y3+t*(x1-x3+I*(y1-y3)):
  wpic1:=plot([evalc(Re(w(z1))),evalc(Im(w(z1))),t=0..1]):
  wpic2:=plot([evalc(Re(w(z2))),evalc(Im(w(z2))),t=0..1]):
  wpic3:=plot([evalc(Re(w(z3))),evalc(Im(w(z3))),t=0..1]):
  plots[display]({wpic1,wpic2,wpic3});
```

Как видно из рис. 15, аппроксимирующая кривая практически совпадает с окружностью!

**Заключение.** Рассмотренные примеры приближенного построения конформных отображений областей на круг единичного радиуса показывают эффективность применения вариационного метода в совокупности с системой аналитических вычислений Maple к решению указанных задач. В системе Maple имеются эффективные средства вычисления кратных и контурных интегралов [3]. Система Maple эффективно используется для решения различных задач математической физики —

математического моделирования в механике деформируемого твердого тела при расчете различных судовых и гидротехнических конструкций, а также при решении задач теории упругости и теплопроводности [6] - [11].

## Литература

1. Голоскоков Д.П. Курс математической физики с использованием пакета Maple / Д.П. Голоскоков. - СПб.: Изд-во Лань, 2015. - 576 с.
2. Голоскоков Д.П. Уравнения математической физики. Решение задач в системе MAPLE / Д.П. Голоскоков. - СПб: Питер, 2004. - 539 с.
3. Голоскоков Д.П. Практический курс математической физики в системе Maple: учебн. пособие для вузов / Д.П. Голоскоков. - СПб: ООО "ПаркКом", 2010. - 643 с.
4. Игнатъев Ю.Г. Математическое и компьютерное моделирование фундаментальных объектов и явлений в системе компьютерной математики Maple. Лекции для школы по математическому моделированию / Ю.Г. Игнатъев. - Казань: Казанский университет, 2014. - 298 с.
5. Голоскоков Д.П. Моделирование конформных отображений вариационным методом в системе символьных вычислений Maple / Д.П. Голоскоков // Вестник государственного университета морского и речного флота имени адмирала С.О. Макарова. - 2015. - Вып. 5(33). - С. 207-216.
6. Голоскоков Д.П. Примеры реализации в Maple энергетического метода для задач изгиба тонких плит / Д.П. Голоскоков // международный семинар по математическому моделированию в системах компьютерной математики KAZCAS-2014 и международная школа по математическому моделированию в системах компьютерной математики KAZCAS-SCHOOL-2014: материалы конференции и труды семинара. - Казань, 2014. - С. 33-62.
7. Голоскоков Д.П. Расчет плоских затворов судоходных шлюзов на динамическое воздействие / Д.П. Голоскоков, А.В. Васин // международная научно-практическая конференция ИТОН-2014 и IV-й международный семинар и международная школа "Математическое и компьютерное моделирование фундаментальных объектов и явлений в системах компьютерной математики: материалы конференции и труды семинара. Казань, 2014. - С. 295-299.
8. Goloskokov D.P. Analyzing simply supported plates using Maple system / D.P. Goloskokov // International conference on computer technologies in physical and engineering applications: materials of conference. - Saint-Petersburg, 2014, P. 57-58.
9. Goloskokov D.P. Simulation of the Dynamic Loads and Calculation of Plane Lock Bypass Galleries / D.P. Goloskokov, A.V. Vasin // International conference on computer technologies in physical and engineering applications: materials of conference. - Saint-Petersburg, 2014, P. 201-202.
10. Голоскоков Д.П. Моделирование температурных полей при частичном нарушении теплоизоляции / Д.П. Голоскоков // Журнал университета водных коммуникаций. - 2010. - Вып. 4(8). - С. 53-56.
11. Голоскоков Д.П. Математическое моделирование нестационарных температурных полей / Д.П. Голоскоков // Журнал университета водных коммуникаций. - 2011. - Вып. 1(9). - С. 79-83.