

Анализ точности численных решений краевых задач на основе аналитических решений

Карпушкина Светлана Анатольевна

1. Введение

Метод конечных элементов (МКЭ) в настоящее время является одним из наиболее применяемых численных методов для решения уравнений в частных производных. Широкому распространению метода во многом способствовало создание на его основе многочисленных программных комплексов для решения широкого круга задач из различных областей физики. Коротко рассмотрим основные положения метода конечных элементов [1].

Пусть в области рассматривается некоторая краевая задача для уравнения в частных производных (предполагается, что решение задачи существует, и оно единственно). При использовании МКЭ исходная область разбивается на подобласти, называемые конечными элементами (КЭ), которые могут иметь произвольную форму. На каждом КЭ выбирается несколько точек – узлов КЭ. Затем внутри каждого конечного элемента искомая функция аппроксимируется полиномом, коэффициенты которого зависят от координат узлов КЭ и значений искомой функции в них (узловых значений). После определения узловых значений одним из известных методов (метод минимизации функционала, метод Галеркина и т.д.) задача считается решенной, так как аппроксимирующие полиномы для приближенного решения будут определены во всей области.

1.1. Основные задачи

Как и любой другой численный метод, МКЭ находит приближенное решение задачи, и степень близости такого решения к точному аналитическому решению во многом зависит от качества разбиения исходной области на КЭ, числа узлов КЭ-сетки, степени аппроксимирующего полинома и т. д. Поэтому в качестве основных задач данной публикации можно выделить:

1. Сравнение аналитического решения некоторой краевой задачи с результатами расчета той же задачи в одном из программных комплексов, реализованных на основе МКЭ;
2. Сравнение решений как в узлах КЭ-сетки, так и во внутренних точках элементов, так как степень аппроксимирующего полинома может существенно увеличить погрешность внутри КЭ по сравнению с погрешностью в узловых точках;
3. Исследование погрешности решения не только для функции, относительно которой записано исходное дифференциальное уравнение в частных производных, но и для её градиента и некоторых интегральных величин, зависящих от неё; так как для ряда задач (например, при расчете электромагнитного поля) больший интерес представляет именно градиент функции, а не она сама.
4. Исследование сходимости приближенного решения к аналитическому при увеличении плотности КЭ-сетки.

1.2. Выбор тестируемого программного комплекса

Для решения поставленных задач в качестве тестируемого программного комплекса выбран ПК ELCUT 5.1 Профессиональная версия (разработчик – Производственный кооператив TOP, Санкт-Петербург). Такой выбор обусловлен несколькими причинами:

- ПК ELCUT уже известен читателям журнала по ряду недавних публикаций (см. [2]-[4]);
- Он в усеченном варианте (количество узлов сетки не превышает 200) доступен всем желающим на сайте <http://elcut.ru/>;
- ПК ELCUT с успехом используется автором статьи в учебном процессе для обучения студентов различных специальностей основам метода конечных элементов [5].

Как следует из документации [6], ELCUT решает краевые задачи для уравнения Пуассона в двумерном случае:

$$\frac{\partial}{\partial x} \left(\lambda_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda_y \frac{\partial T}{\partial y} \right) = -q,$$

где в линейной постановке λ_x, λ_y – константы, $q = q(x, y)$; в нелинейной – $\lambda_x = \lambda_y = \lambda(T)$, $q = q(T)$. В качестве конечного элемента выбран треугольник 1-го порядка, т.е. КЭ-сетка состоит из треугольников, а аппроксимирующий полином имеет вид: $T(x, y) \approx a_0 + a_1x + a_2y$, где (x, y) – координаты внутренней точки треугольника, a_0, a_1, a_2 – константы. Для нахождения узловых значений функции выбран метод Галеркина.

1.3. Выбор тестовой задачи

В качестве тестовой задачи выберем задачу расчета стационарного температурного поля без внутренних источников тепла; такое температурное поле описывается уравнением Лапласа, которое является частным случаем уравнения Пуассона. В прямоугольной декартовой системе координат уравнение Лапласа имеет вид:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0. \quad (1)$$

Решение будем искать внутри прямоугольной области

$$0 \leq x \leq a, 0 \leq y \leq b, \quad (2)$$

на границе которой зададим условия Дирихле (рис. 1):

$$T(0, y) = f_1(y), T(a, y) = f_2(y), T(x, 0) = f_3(x), T(x, b) = f_4(x) \quad (3)$$

Предполагается, что

$$f_1(0) = f_3(0), f_1(b) = f_4(0), f_2(0) = f_3(a), f_2(b) = f_4(a) \quad (4)$$

Таким образом, тестовая задача – это первая краевая задача для уравнения Лапласа в прямоугольной области. Её выбор обусловлен, прежде всего, тем, что она является самой простой из всего многообразия задач, решаемых в ПК ELCUT. На основе результатов, полученных для нее, можно будет легко перейти к исследованию более сложных задач.

Результирующими величинами, получаемыми в ELCUT при расчете стационарного температурного поля, являются:

1. температура T ;
2. вектор плотности теплового потока $F = -\lambda \cdot \mathbf{grad}T$, где λ – коэффициент теплопроводности;
3. поток тепла через заданную поверхность $\Phi = \int_S \mathbf{F} \cdot \mathbf{n} ds$, где \mathbf{n} – единичный вектор нормали к поверхности S (поверхность задается контуром в плоскости модели).

В дальнейшем ELCUT-решение будет сравниваться с аналитическим решением именно по этим показателям.

1.4. Аналитическое решение тестовой задачи

Аналитическое решение задачи (1), (3) в области (2) можно представить в виде [7]:

$$T(x,y) = \sum_{n=1}^{\infty} A_n \operatorname{sh}\left(\frac{\pi n(a-x)}{b}\right) \sin\left(\frac{\pi ny}{b}\right) + \sum_{n=1}^{\infty} B_n \operatorname{sh}\left(\frac{\pi nx}{b}\right) \sin\left(\frac{\pi ny}{b}\right) + \sum_{n=1}^{\infty} C_n \operatorname{sh}\left(\frac{\pi n(b-y)}{a}\right) \sin\left(\frac{\pi nx}{a}\right) + \sum_{n=1}^{\infty} D_n \operatorname{sh}\left(\frac{\pi ny}{a}\right) \sin\left(\frac{\pi nx}{a}\right), \quad (5)$$

где коэффициенты A_n, B_n, C_n, D_n определяются формулами:

$$A_n = \frac{2}{\lambda_n} \int_0^b f_1(\xi) \sin\left(\frac{\pi n \xi}{b}\right) d\xi, \quad B_n = \frac{2}{\lambda_n} \int_0^b f_2(\xi) \sin\left(\frac{\pi n \xi}{b}\right) d\xi, \quad C_n = \frac{2}{\mu_n} \int_0^a f_3(\xi) \sin\left(\frac{\pi n \xi}{a}\right) d\xi, \\ D_n = \frac{2}{\mu_n} \int_0^a f_4(\xi) \sin\left(\frac{\pi n \xi}{a}\right) d\xi, \quad \lambda_n = b \operatorname{sh}\left(\frac{\pi na}{b}\right), \quad \mu_n = a \operatorname{sh}\left(\frac{\pi nb}{a}\right) \quad (6)$$

Из (5) непосредственным дифференцированием по x и y легко получить компоненты вектора $\mathbf{grad}T$, по которым можно будет определить и компоненты вектора \mathbf{F} . Для нахождения потока тепла Φ возьмем за контур интегрирования ломанную из двух звеньев: первое – отрезок прямой, параллельной оси Ox , с концами в точках (a_1, b_1) и (a_2, b_1) , второе – отрезок прямой, параллельной оси Oy , с концами в точках (a_2, b_1) и (a_2, b_2) (рис. 1). Тогда

$$\Phi = -\int_{a_1}^{a_2} F_y(x, b_1) dx + \int_{b_1}^{b_2} F_x(a_2, y) dy. \quad (7)$$

2. Решение тестовой задачи

2.1. Реализация аналитического решения тестовой задачи

Для нахождения аналитического решения тестовой задачи по формулам (5)-(6) будем использовать систему Mathcad 2001 (Professional Edition).

В начале зададим величины a, b для границ прямоугольника, а также две константы ε и ε_1 . Первая константа будет использоваться для сравнения числовых значений с нулем, т.е., если значение по модулю меньше ε , то считается, что оно равно нулю. Вторая константа задает

точность нахождения членов бесконечных рядов, т.е. ряды в формуле (5) будут считаться до тех пор, пока некоторый член ряда не станет по модулю меньше $\varepsilon 1$. Далее введем функции $f1, f2, f3, f4$ для граничных условий (3) и произведем их проверку в соответствии с условиями (4) (рис. 2).

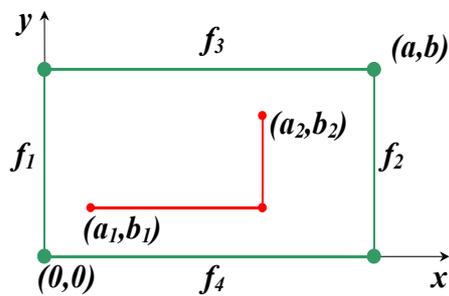


Рис. 1. Область решения задачи

```

Первая краевая задача для уравнения Лапласа в прямоугольной области
Границы прямоугольника   Точность для сравнения с нулем   Точность для члена ряда
a := 2   b := 1           ε := 10-10                               ε1 := 10-7
Граничные условия
f1(y) := a · sin(π · y / b)   f2(y) := y4 - b · y3 + y2 - b · y   f3(x) := x2 - a · x   f4(x) := b · sin(π · x / a)
Проверка граничных условий
prov := | 1 if |f1(0) - f3(0)| < ε ∧ |f1(b) - f4(0)| < ε ∧ |f2(0) - f3(a)| < ε ∧ |f2(b) - f4(a)| < ε
        | 0 otherwise
prov = 1
    
```

Рис. 2. Константы и граничные условия

Создадим функции для n -ых членов рядов, входящих в формулу (5) (рис. 3). В этих функциях используется функция $sh_sh(a,b,n,m)$ (рис. 4), которая реализует отношение гиперболических функций, например, $sh_sh(x,y,1,1) = sh(x)/sh(y)$, $sh_sh(x,y,1,0) = ch(x)/ch(y)$. Необходимость введения такой функции обусловлена тем, что непосредственная реализация в Mathcad отношений вида $sh(x)/sh(y)$ через встроенные функции $\sinh()$ и $\cosh()$ (при больших номерах n членов рядов) часто приводит к получению чисел, превышающих 10^{308} , что вызывает ошибку выполнения кода. Функция sh_sh , анализируя аргументы гиперболических функций, производит тождественное преобразование исходного отношения, что в большинстве случаев позволяет выполнить нужный расчет.

```

Интегралы из An, Bn, Cn, Dn
IntA(n) := ∫0b f1(ξ) · sin(π · n · ξ / b) dξ   IntB(n) := ∫0b f2(ξ) · sin(π · n · ξ / b) dξ
IntC(n) := ∫0a f3(ξ) · sin(π · n · ξ / a) dξ   IntD(n) := ∫0a f4(ξ) · sin(π · n · ξ / a) dξ
n-ые члены рядов
S1(n, x, y) := 2 · IntA(n) · b-1 · sin(π · n · y / b) · sh_sh[π · n · (a - x) / b, π · n · a / b, 1, 1]
S2(n, x, y) := 2 · IntB(n) · b-1 · sin(π · n · y / b) · sh_sh[π · n · x / b, π · n · a / b, 1, 1]
S3(n, x, y) := 2 · IntC(n) · a-1 · sin(π · n · x / a) · sh_sh[π · n · (b - y) / a, π · n · b / a, 1, 1]
S4(n, x, y) := 2 · IntD(n) · a-1 · sin(π · n · x / a) · sh_sh[π · n · y / a, π · n · b / a, 1, 1]
    
```

Рис. 3. Функции для n -ых членов рядов

```

sh_sh(a, b, n, m) := s ←  $\frac{1 + (-1)^n \exp(-2 \cdot a)}{\exp(b - a) + (-1)^m \cdot \exp(-b - a)}$  if a > 0 ∧ b > 0 ∧ a > b
s ←  $\frac{\exp(a - b) + (-1)^n \cdot \exp(-a - b)}{1 + (-1)^m \cdot \exp(-2 \cdot b)}$  if a > 0 ∧ b > 0 ∧ a ≤ b
s ←  $\frac{\exp(2 \cdot a) + (-1)^n}{\exp(a + b) + (-1)^m \cdot \exp(-a + b)}$  if a < 0 ∧ b < 0 ∧ |a| > |b|
s ←  $\frac{\exp(a + b) + (-1)^n \cdot \exp(-a + b)}{\exp(2 \cdot b) + (-1)^m}$  if a < 0 ∧ b < 0 ∧ |a| ≤ |b|
s ←  $\frac{\exp(b) + (-1)^n \cdot \exp(b - 2 \cdot a)}{\exp(2 \cdot b - a) + (-1)^m \cdot \exp(-a)}$  if a > 0 ∧ b < 0
s ←  $\frac{\exp(2 \cdot a - b) + (-1)^n \cdot \exp(-b)}{\exp(a) + (-1)^m \cdot \exp(a - 2 \cdot b)}$  if a < 0 ∧ b > 0
return s
    
```

Рис. 4. Функция sh_sh

Создадим функции для нахождения частичных сумм рядов с заданной точностью $\varepsilon 1$ (рис. 5). В этом фрагменте кода находится частичная сумма первого ряда из формулы (5). Аналогично создаются функции $S2_1(x,y)$, $S3_1(x,y)$, $S4_1(x,y)$ для частичных сумм остальных рядов формулы (5). Построение решения для компонент вектора \mathbf{F} происходит точно так же, как и для T . Окончательно, для аналитического выражения T, \mathbf{F}, Φ получим (рис. 6):

```

Частичные суммы рядов с заданной точностью
S1_1(x, y) :=
  n ← 1
  S ← 0
  while |S1(n, x, y)| > ε1
    S ← S + S1(n, x, y)
    n ← n + 1
    while (|sin(π · n · y / b)| < ε ∨ |lnA(n)| < ε) ∧ n < 100
      n ← n + 1
  return S

```

Рис. 5. Частичная сумма ряда

```

Аналитическое решение для T
T(x, y) := S1_1(x, y) + S2_1(x, y) + S3_1(x, y) + S4_1(x, y)
Аналитическое решение для F
λ := 10
Fx(x, y) := -λ · (S1_1x(x, y) + S2_1x(x, y) + S3_1x(x, y) + S4_1x(x, y))
Fy(x, y) := -λ(S1_1y(x, y) + S2_1y(x, y) + S3_1y(x, y) + S4_1y(x, y))
Аналитическое решение для Φ
a1 := 0.1 b1 := 0.2 a2 := 1.8 b2 := 0.7
Φ := -∫a1a2 Fy(x, b1) dx + ∫b1b2 Fx(a2, y) dy

```

Рис. 6. Аналитическое решение

После расчетов по функциям $T(x,y)$, $F_x(x,y)$, $F_y(x,y)$ можно получить картины соответствующих полей (например, рис. 7). Значение потока тепла, найденное по функции Φ равно 37.555.

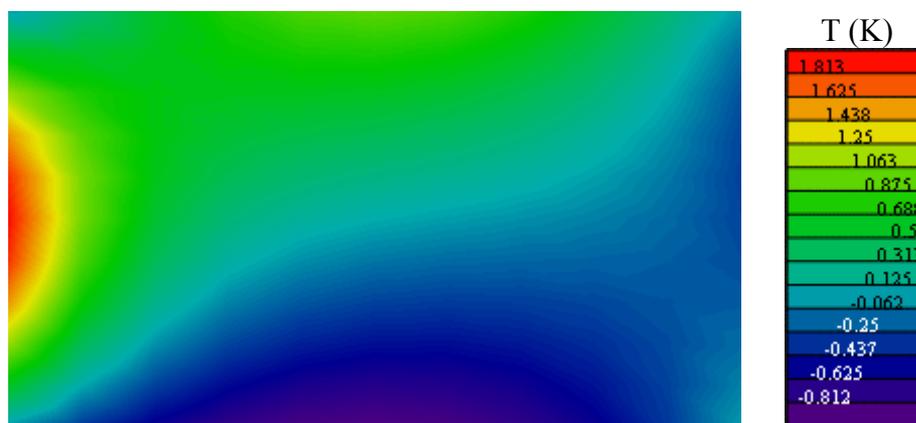


Рис. 7. Температура T на основе аналитического решения

2.2. Решение тестовой задачи в ПК ELCUT 5.1

Для решения тестовой задачи в ELCUT выполним следующие действия:

1. Создадим плоскую задачу расчета стационарного температурного поля `templo1.pbm`.
2. Для создания геометрической модели построим прямоугольник со сторонами 2 м и 1 м. В вершинах прямоугольника поставим шаг дискретизации, равный 0,2 м, разобьем геометрическую модель на КЭ (рис. 8).
3. Присвоим сторонам прямоугольника метки «f1», «f2», «f3», «f4», на которых зададим значения температуры соответственно функциям f_1, f_2, f_3, f_4 (здесь следует иметь в виду, что в ELCUT аргумент тригонометрических функций задается в градусах). Области, лежащей внутри прямоугольника, присвоим метку «Inside» и зададим на ней значение коэффициента теплопроводности, равное 10, а в вершинах прямоугольника поставим метку «Corner», значения параметров которой оставим по умолчанию.

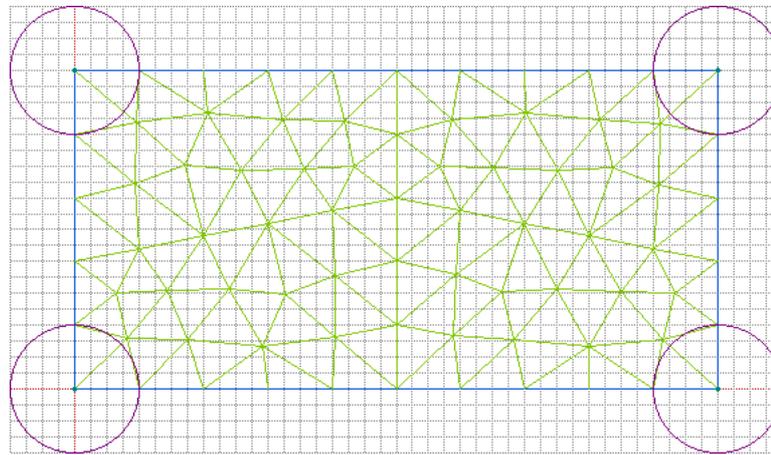


Рис. 8. Геометрическая модель и КЭ-сетка задачи

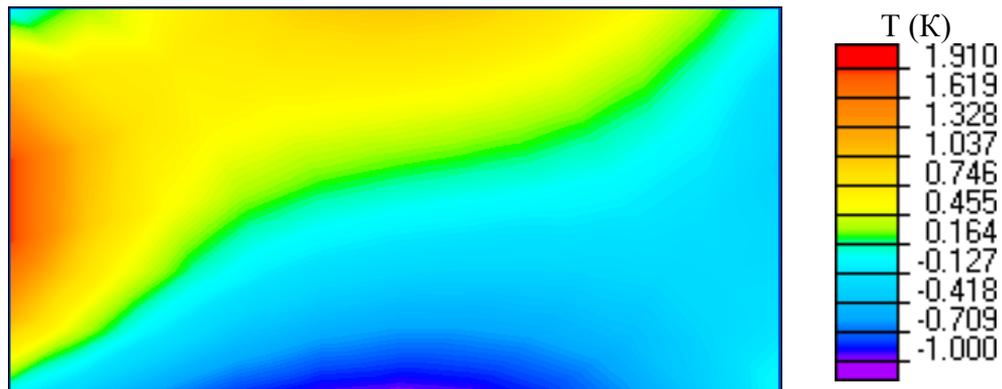


Рис. 9. T на основе ELCUT-решения

Созданная таким образом задача будет соответствовать первой краевой задаче для уравнения Лапласа. После решения, используя команду «Картина поля», получим картины полей T , F_x , F_y (например, рис. 9).

Нарисовав нужный контур и применяя команду «Интегральные значения», получим, что значение потока тепла через заданную поверхность равно 37.647.

3. Сравнение аналитического и численного решений

На рис. 7 и 9 видно, что картины полей, полученные на основе аналитического и численного решений, идентичны. Для более точного сравнения решений будем использовать норму разности между численными и аналитическими значениями величин T , F_x , F_y и модуль разности между значениями потока тепла Φ :

$$\|R^{ан} - R^{числ}\| = \sum_{i=1}^N [R^{ан}(x_i, y_i) - R^{числ}(x_i, y_i)]^2, \quad |\Phi^{ан} - \Phi^{числ}|, \quad (8)$$

где R – значение T или F_x , или F_y ; x_i, y_i – либо узлы КЭ-сетки, либо внутренние точки конечных элементов.

3.1. Сравнение решений во внутренних точках конечных элементов

Для получения значений локальных величин во внутренних точках КЭ в ELCUT можно использовать команду «Экспорт поля». В диалоговых окнах этой команды зададим сетку, охватывающую всю геометрическую модель, с размером ячейки 0,2 м × 0,1 м, а в качестве выводимых физических величин выберем температуру $T(x,y)$ и компоненты вектора теплового потока $F_x(x,y)$ и $F_y(x,y)$. Результаты запишем в текстовый файл *teplo2.prn*. Тогда этот файл будет содержать следующие величины:

1. координата x ;
2. координата y ;
3. $T(x,y)$;
4. $F_x(x,y)$;
5. $F_y(x,y)$.

Количество строк файла равно 121.

Теперь перейдем в Mathcad и подсчитаем аналитические значения T , F_x , F_y в тех же точках, а затем найдем нормы разности (рис. 10, 11):

```

x_y_T := READPRN("C:\temp\T\teplo2.prn")
nr1 := rows(x_y_T)
i := 0..nr1-1
x_y_Ti,5 := T(x_y_Ti,0, x_y_Ti,1)    x_y_Ti,6 := Fx(x_y_Ti,0, x_y_Ti,1)    x_y_Ti,7 := Fy(x_y_Ti,0, x_y_Ti,1)
T_T := √(∑_{i=0}^{nr1-1} (x_y_Ti,2 - x_y_Ti,5)²)
Fx_Fx := √(∑_{i=0}^{nr1-1} (x_y_Ti,3 - x_y_Ti,6)²)
Fy_Fy := √(∑_{i=0}^{nr1-1} (x_y_Ti,4 - x_y_Ti,7)²)
dT := (max(x_y_T^{(2)}) - min(x_y_T^{(2)}))
dFx := (max(x_y_T^{(3)}) - min(x_y_T^{(3)}))
dFy := (max(x_y_T^{(4)}) - min(x_y_T^{(4)}))
    
```

Рис. 10. Нахождение норм разности во внутренних точках КЭ

$T_T = 0.08669425$	$F_x_F_x = 3.19059251$	$F_y_F_y = 3.60483321$
$\frac{T_T}{dT} = 0.02987397$	$\frac{F_x_F_x}{dFx} = 0.04797883$	$\frac{F_y_F_y}{dFy} = 0.03342761$

Рис. 11. Нормы разности между численным и аналитическим решениями во внутренних точках КЭ

Полученные нормы можно разделить на разницу между максимальным и минимальным значением соответствующей величины, чтобы учесть пределы её изменения. На рис. 11 видно, что точность результатов для температуры составляет 0.03, т.е. результаты численного и аналитического решения расходятся примерно на 3%, в то время как точность результатов для компонент вектора теплового потока составляет 0.048 и 0.033 (соответственно 4.8% и 3.3% расхождения). Таким образом, точность численных результатов для температуры приблизительно такая же, как и точность результатов для её производных. Для интегральной величины Φ было получено: 37.555 – при аналитическом расчете, 37.647 – в ELCUT. Тогда разница составит 0.092 или 0.245%. Меньшее значение погрешности для Φ , по сравнению с погрешностями T , F_x , F_y можно объяснить тем, что нормы этих величин считались в 121 узле, в то время как при нахождении Φ использовалось гораздо меньшее число узлов.

3.2. Сравнение решений в узлах конечно-элементной сетки

Во введении уже указывалось, что степень аппроксимирующего полинома может увеличить погрешность решения внутри КЭ по сравнению с погрешностью в узловых точках. Поэтому найдем нормы разности между аналитическими и численными решениями для T , F_x , F_y в узлах КЭ-сетки, а затем сравним полученные результаты с результатами предыдущего пункта.

Для получения координат узлов КЭ-сетки можно использовать утилиту *Mod2Txt.exe*, предоставляемую разработчиками ELCUT. Эта утилита считывает файл геометрии ELCUT (имеет расширение *.mod) с построенной КЭ-сеткой и на основе этого файла создает текстовый файл, содержащий информацию о координатах узлов и другие данные о разбиении.

Когда координаты узлов КЭ-сетки получены, необходимо вычислить значения T , F_x , F_y в них. Однако, в самом ELCUT нет способа их вывести, например, в текстовый файл или на экран для последующей обработки. Используемая в предыдущем пункте команда «Экспорт поля» здесь не применима, так как в ней можно задать расчет локальных величин только в узлах прямоугольной сетки. Единственной возможностью в данной ситуации остается написание собственного приложения, которое будет выполнять следующие действия:

1. открывать существующую задачу *teplo1.pbm* и решать её;
2. запускать из самого приложения утилиту *Mod2Txt.exe* для создания файла с координатами узлов КЭ-сетки;
3. организовывать цикл по записям созданного файла;
4. если запись содержит координаты узла, то вычислять значения T , F_x , F_y в них и записывать в некоторый файл *.prn для последующей работы с ним в Mathcad, иначе завершать цикл.

Написание подобного приложения оказалось бы невозможным без применения технологии ActiveField, предложенной разработчиками ELCUT в версии 5.1. Эта технология позволяет работать с ELCUT не только посредством графического интерфейса, но и разрабатывать собственные приложения, взаимодействующие с ядром ELCUT посредством COM Automation. Большинство современных средств разработки приложений, включая Microsoft Visual Basic, Visual Basic for Applications, Visual C++, Borland Delphi и многие другие, поддерживают эту технологию.

Код приложения написан на Microsoft Visual Basic 6.0, так как именно для этого языка в документации ELCUT содержится большое количество примеров по использованию технологии ActiveField; причем все примеры подробно разобраны, так что даже начинающий программист может применить их для своих разработок.

Приложение построено следующим образом:

- объявляются глобальные переменные и константы (прил. 1);

- работа приложения начинается с загрузки формы (прил. 2), где открывается уже существующая задача *teplo1.pbm*;
- расчет запускается нажатием кнопки *Command1*, встроенной в форму (прил. 3). При этом:
 - в функции *Solve_Problem* (прил. 4) решается открытая задача ELCUT;
 - в функции *FEGrid* (прил. 5) осуществляется расчет локальных значений в узлах КЭ-сетки. Эта функция возвращает число узлов КЭ-сетки для геометрической модели ELCUT, в качестве параметров берутся имя файла, который будет выходным файлом при работе утилиты *Mod2Txt.exe*, и имя файла, в который будут помещаться координаты узлов КЭ-сетки и значения локальных величин в них.
- работа завершается закрытием ELCUT-задач и выходом из ELCUT (прил. 2).

В результате будет сформирован файл *TNodes.prn*, в котором записаны следующие величины:

1. координата x узла КЭ-сетки;
2. координата y ;
3. $T(x,y)$;
4. $F_x(x,y)$;
5. $F_y(x,y)$.

Количество строк файла равно 78 (по количеству узлов).

Перейдем в Mathcad и подсчитаем аналитические значения T , F_x , F_y в узлах той же КЭ-сетки, а затем найдем нормы разности. Соответствующий Mathcad-код ничем не отличается от кода на рис. 10, кроме имени считываемого файла (*teplo2.prn* следует заменить на *TNodes.prn*). В результате получим:

$T - T = 0.03297103$	$F_x - F_x = 1.76897629$	$F_y - F_y = 2.42366505$
$\frac{T - T}{dT} = 0.01136148$	$\frac{F_x - F_x}{dF_x} = 0.02660115$	$\frac{F_y - F_y}{dF_y} = 0.02247464$

Рис. 12. Нормы разности между численным и аналитическим решениями в узлах КЭ-сетки.

Сравнивая эти значения с соответствующими значениями на рис. 11, можно сказать, что погрешность для величин T , F_x , F_y в узлах КЭ-сетки меньше, чем во внутренних точках конечных элементов. Если учесть, что нормы разности для внутренних точек считались для 121 узла, а нормы для КЭ-сетки – для 78 узлов, то, разделив значения из рис. 11 на 121, а значения из рис. 12 на 78, получим более достоверные результаты для сравнения:

	Для внутренних точек КЭ	Для узлов КЭ-сетки
Для нормы T	$0.03 / 121 = 2.469 \cdot 10^{-4}$	$0.011 / 78 = 1.457 \cdot 10^{-4}$
Для нормы F_x	$0.048 / 121 = 3.965 \cdot 10^{-4}$	$0.027 / 78 = 3.410 \cdot 10^{-4}$
Для нормы F_y	$0.033 / 121 = 2.762 \cdot 10^{-4}$	$0.022 / 78 = 2.881 \cdot 10^{-4}$

Из этой таблицы видно, что если для температуры погрешность в узлах КЭ-сетки осталась по-прежнему меньше, чем погрешность во внутренних точках КЭ, то для F_x и F_y эти погрешности практически сравнялись.

3.3. Сравнение решений при увеличении числа узлов конечно-элементной сетки

Все приведенные выше расчеты были проделаны при сравнительно малом числе узлов КЭ-сетки геометрической модели ELCUT, а именно, 78 узлах. Варьируя шаг дискретизации в вершинах прямоугольника, можно попробовать уменьшить норму разности между численным и аналитическим решениями и тем самым определить влияние числа узлов на точность получаемых результатов.

Будем изменять шаг дискретизации от 0.2 до 0.002, причем каждое последующее значение шага получается из предыдущего значения делением на 1.5; для каждого шага дискретизации будем фиксировать соответствующее число узлов КЭ-сетки и рассчитывать нормы разности между численными и аналитическими значениями величин T , F_x , F_y во внутренних точках КЭ (аналогично п. 3.1) и в узлах КЭ-сетки (аналогично п. 3.2). Прделанные расчеты сведем в таблицу:

Шаг дискретизации	Число узлов КЭ-сетки	Для внутренних точек КЭ			Для узлов КЭ-сетки			Для модуля Φ
		Для нормы T	Для нормы F_x	Для нормы F_y	Для нормы T	Для нормы F_x	Для нормы F_y	
0.2	78	$2.469 \cdot 10^{-4}$	$3.965 \cdot 10^{-4}$	$2.762 \cdot 10^{-4}$	$1.457 \cdot 10^{-4}$	$3.410 \cdot 10^{-4}$	$2.881 \cdot 10^{-4}$	0.092
0.133	184	$1.037 \cdot 10^{-4}$	$2.424 \cdot 10^{-4}$	$1.302 \cdot 10^{-4}$	$4.735 \cdot 10^{-5}$	$1.649 \cdot 10^{-4}$	$1.131 \cdot 10^{-4}$	$1.277 \cdot 10^{-3}$
0.089	326	$7.155 \cdot 10^{-5}$	$2.024 \cdot 10^{-4}$	$7.509 \cdot 10^{-5}$	$1.976 \cdot 10^{-5}$	$1.341 \cdot 10^{-4}$	$7.443 \cdot 10^{-5}$	$4.859 \cdot 10^{-3}$
0.059	745	$3.449 \cdot 10^{-5}$	$1.014 \cdot 10^{-4}$	$5.887 \cdot 10^{-5}$	$7.227 \cdot 10^{-6}$	$5.790 \cdot 10^{-5}$	$4.293 \cdot 10^{-5}$	$6.192 \cdot 10^{-3}$
0.040	1600	$1.057 \cdot 10^{-5}$	$5.575 \cdot 10^{-5}$	$5.000 \cdot 10^{-5}$	$2.159 \cdot 10^{-6}$	$2.467 \cdot 10^{-5}$	$1.884 \cdot 10^{-5}$	0.018
0.026	3527	$8.687 \cdot 10^{-6}$	$4.539 \cdot 10^{-5}$	$3.527 \cdot 10^{-5}$	$6.906 \cdot 10^{-7}$	$1.265 \cdot 10^{-5}$	$8.713 \cdot 10^{-6}$	$6.051 \cdot 10^{-4}$
0.018	7990	$2.290 \cdot 10^{-6}$	$3.330 \cdot 10^{-5}$	$1.414 \cdot 10^{-5}$	$1.937 \cdot 10^{-7}$	$5.363 \cdot 10^{-6}$	$3.350 \cdot 10^{-6}$	$6.389 \cdot 10^{-4}$
0.012	17802	$1.037 \cdot 10^{-6}$	$1.298 \cdot 10^{-5}$	$9.628 \cdot 10^{-6}$	$9.842 \cdot 10^{-8}$	$3.241 \cdot 10^{-6}$	$2.583 \cdot 10^{-6}$	$2.275 \cdot 10^{-3}$
$7.804 \cdot 10^{-3}$	47489	$3.561 \cdot 10^{-7}$	$5.464 \cdot 10^{-6}$	$3.999 \cdot 10^{-6}$	$7.178 \cdot 10^{-8}$	$1.967 \cdot 10^{-6}$	$1.669 \cdot 10^{-6}$	$7.515 \cdot 10^{-3}$
$5.202 \cdot 10^{-3}$	109505	$1.951 \cdot 10^{-7}$	$4.769 \cdot 10^{-6}$	$3.017 \cdot 10^{-6}$				$1.935 \cdot 10^{-3}$
$3.468 \cdot 10^{-3}$	207068	$1.017 \cdot 10^{-7}$	$4.260 \cdot 10^{-6}$	$2.660 \cdot 10^{-6}$				$1.187 \cdot 10^{-3}$
$2.312 \cdot 10^{-3}$	507676	$3.506 \cdot 10^{-8}$	$1.355 \cdot 10^{-6}$	$9.669 \cdot 10^{-7}$				$9.941 \cdot 10^{-3}$

Замечания:

1. Выделенные цветом строки таблицы относятся к результатам, которые можно получить, используя версию Student Edition.
2. Результаты для узлов КЭ-сетки получены не для всех рассматриваемых шагов дискретизации в связи с тем, что утилита Mod2Txt.exe не может обрабатывать сетки, содержащие более 100000 узлов.
3. Нормы всех величин были поделены на количество точек, по которым они были рассчитаны: для внутренних точек на 121, для узлов КЭ-сетки на соответствующее число узлов (см. 2-ой столбец таблицы).

На основе таблицы можно построить графики (рис. 13), из которых видно, что при увеличении числа узлов КЭ-сетки нормы разности для всех величин постепенно сходятся к аналитическому решению; причем нормы разности, рассчитанные для узлов КЭ, сходятся быстрее, чем нормы для внутренних точек (см. $T_{вн}$ и $T_{кэ}$, $F_{хвн}$ и $F_{хкэ}$, $F_{ывн}$ и $F_{укэ}$); также норма для температуры сходится быстрее, чем нормы для компонент вектора теплового потока (см. $T_{вн}$, $F_{хвн}$ и $F_{ывн}$, а также $T_{кэ}$, $F_{хкэ}$, $F_{укэ}$).

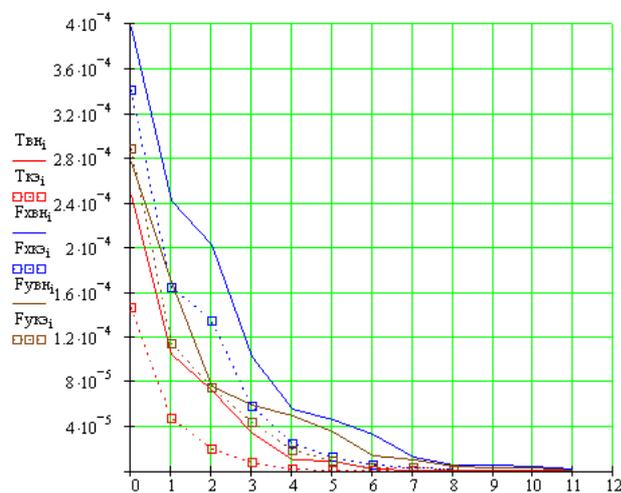


Рис. 13. Нормы разности для T , F_x , F_y при увеличении числа узлов КЭ-сетки

4. Основные выводы

В работе исследована точность численного решения, полученного методом конечных элементов в ПК ELCUT 5.1, первой краевой задачи для уравнения Лапласа по сравнению с аналитическим решением той же задачи. При этом сопоставлялась как сама функция, относительно которой решалось уравнение Лапласа, так и градиент этой функции, а также контурный интеграл от градиента. Отдельно исследована точность решения в узлах сетки конечных элементов и во внутренних точках элементов, причем все сравнения проводились при разной густоте конечно-элементной сетки.

Результат сопоставления показал следующее:

1. В студенческой версии ПК ELCUT 5.1 с размерностью КЭ-сетки менее 200 узлов достигается практически приемлемое решение (с погрешностью 3-4 %) для всех величин (T , F_x , F_y , Φ). Это позволяет использовать Студенческую версию ELCUT 5.1 для учебных и демонстрационных целей.
2. В профессиональной версии ELCUT 5.1 численное решение хорошо сходится к аналитическому при увеличении густоты сетки для исследованных величин T , F_x , F_y , Φ . Так, например, для T точность в 5 верных десятичных знаков можно получить при количестве узлов сетки равном ≈ 50000 , что соответствует времени решения на среднем персональном компьютере не более 15 секунд. Следует отметить, что для T скорость сходимости на порядок выше, чем для F_x , F_y и примерно на два порядка выше, чем для Φ .
3. Точность решения в узлах КЭ-сетки оказывается выше, чем во внутренних точках. Однако, для T точность будет несколько выше, чем для F_x , F_y .

Хотя в статье речь шла только о стационарных тепловых полях без внутренних источников тепла, но эти же выводы можно отнести и к другим физическим задачам, решаемым в ПК ELCUT, которые сводятся к решению первой краевой задачи для уравнения Лапласа (например, задача расчета электростатического поля по заданным значениям потенциала на границе при отсутствии объемных зарядов, а также задача расчета упруго-деформированного состояния массива с закрепленными границами при отсутствии объемных сил).

Таким образом, можно заключить, что программный комплекс ELCUT продемонстрировал практическую применимость к решению задач описанного типа.

В качестве пожеланий к последующим версиям ELCUT можно отметить:

1. включение в состав ПК ELCUT средства для генерации координат узлов конечно-элементной сетки и значений искомой функции в них с выводом на экран или текстовый файл;
2. создание надстройки, обеспечивающей автоматизированное решение задач с разной плотностью КЭ-сеток и формирование сравнений для нужных величин на таких сетках.

Приложение 1. Глобальные переменные и константы

```
Public Const d = "C:\temp\TOR"  
Public elc As ELCUT.Application  
Public prb As ELCUT.Problem  
Public mdl As ELCUT.Model  
Public res As ELCUT.Result  
Public wnd As ELCUT.FieldWindow  
Public cnt As ELCUT.Contour  
Public Coords As ELCUT.Point  
Public T Coords As ELCUT.FieldPoint
```

Приложение 2. Процедуры загрузки и выгрузки формы

```
Private Sub Form_Load()  
Set elc = CreateObject("ELCUT.Application")  
elc.Problems.Close  
elc.DefaultFilePath = d  
Set prb = elc.Problems.Open("tepl01.pbm")  
prb.LoadModel:Set mdl = prb.Model  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
elc.Problems.Close:elc.Quit  
End Sub
```

Приложение 3. Процедура для запуска расчетов

```
Private Sub Command1_Click()  
Dim Nodes_Number As Long  
If Solve_Problem = 0 Then MsgBox  
    ("Задача не может быть решена")  
Nodes_Number = FEGrid("\Nodes.txt", \TNodes.prn")  
End Sub
```

Приложение 4. Функция Solve_Problem

```
Public Function Solve_Problem() As Byte  
If prb.CanSolve Then prb.SolveProblem  
If prb.Solved Then prb.AnalyzeResults  
Set res = prb.Result  
If res Is Nothing Then Solve_Problem = 0  
    Else Solve_Problem = 1  
End Function
```

Приложение 5. Функция FEGrid

```
Public Function FEGrid(Nodes_file As String, T_Nodes_file As String) As Long  
Dim s As String, X1 As Double, Y1 As Double  
Dim i As Long, Pos1 As Integer, RetVal As Integer  
RetVal = Shell("c:\WINDOWS\COMMAND.COM /C "&d"&\Mod2Txt.exe "&d"&\Teplo1_1.mod > "&d&Nodes_file)  
If RetVal = 0 Then Call MsgBox("Ошибка запуска")  
PauseTime = 10: Start = Timer  
Do While Timer < Start + PauseTime:DoEvents:Loop  
Open d & Nodes_file For Input Access Read As #2:Open d & T_Nodes_file For Output As #1  
i = 0  
Do  
    Input #2, s  
    Pos1 = InStr(1, Trim(s), " ", vbTextCompare)  
    If Pos1 < 1 Or InStr(1, Trim(Mid(s, Pos1)), " ", vbTextCompare) > 1 Then Exit Do  
    End If  
    X1 = Val(Trim(Left(s, Pos1 - 1))): Y1 = Val(Trim(Mid(s, Pos1)))  
    i = i + 1  
    Set Coords = elc.PointXY(X1, Y1): Set T_Coords = res.GetLocalValues(Coords)  
    Write #1, Coords.X, Coords.Y, T_Coords.Potential, T_Coords.KGrad.X, T_Coords.KGrad.Y  
Loop  
Close #1:Close #2  
FEGrid = i  
End Function
```

Литература

1. Сабоннадьер Ж.-К., Кулон Ж.-Л. Метод конечных элементов и САПР. – М.: Мир, 1989. – 190 с.
2. Дубицкий С. Д. ELCUT 5.1 – платформа разработки приложений анализа полей // EXponenta Pro. Математика в приложениях. – 2004. – №1. – С. 20-25.
3. Дубицкий С. Д., Поднос В. П. ELCUT – инженерная система моделирования двумерных физических полей // CADmaster. – 2001. - №1. – С. 17-21.
4. Черных И. В. Моделирование устройств индукционного нагрева с помощью пакета ELCUT // EXponenta Pro. Математика в приложениях. – 2003. – №2. – С. 4-8.
5. Белов В.Ф., Шабанов Г.И., Карпушкина С.А. Математическое моделирование. – Саранск: Изд-во Мордов. ун-та, 2001. – 340 с.
6. ELCUT. Моделирование двумерных полей методом конечных элементов. Версия 5.1. Руководство пользователя. – СПб.: Производственный кооператив ТОР, 2003. – 249 с.
7. Полянин А. Д. Справочник по линейным уравнениям математической физики. – М.: ФИЗМАТЛИТ, 2001. – 576 с.