Министерство образования и науки Российской Федерации

Новосибирский государственный технический университет

Кафедра прикладной математики

Курсовая работа по дисциплине:

«Методы конечно-элементного анализа»

Факультет: ПМИ

Группа: ПММ-42

Студентка: Сулейманова К.А.

Преподаватель: Персова М.Г.

Новосибирск

2015

1. **Задание**

Реализовать численное решение задачи МКЭ с гармоническим источником.

1. **Теоретическая часть**
2. *Вид уравнения*

Рассмотрим уравнение (1):

|  |  |
| --- | --- |
| . | (1) |

Будем считать, что изучаемое электромагнитное поле полностью описываемое вектор-потециалом может быть представлено в виде суммы двух полей, каждое из которых описывается своим вектор-потенциалом , – аномальное поле, – нормальное поле.

|  |  |
| --- | --- |
| *.* | (2) |

Будем считать, что поле – может быть получено из решения двумерной задачи и будет достаточно близко к полю близко :

|  |  |
| --- | --- |
| . | (3) |

Вычтем (2) из (3) получим:

|  |  |
| --- | --- |
| . | (4) |
|  |  |

Пусть аномалия отличается от нормального поля только электропроводностью, а магнитные проницаемости равны, тогда первое слагаемое в правой части уравнения (14) будет равно нулю:

|  |  |
| --- | --- |
| . | (5) |
|  |  |

1. *Вариационная постановка*

Вариационная постановка для уравнения (14) с однородными краевыми условиями первого рода примет вид [11, c.806]

|  |  |
| --- | --- |
| . | (6) |

1. *Дискретизация по времени*

Для решения уравнения (6) воспользуемся двухслойной и трехслойной схемами по времени:

|  |  |
| --- | --- |
| ,  . | (7) |

|  |  |
| --- | --- |
| .  . | (8) |

1. *Конечноэлементная дискретизация*

При построении дискретного аналога приближенное решение будем представлять в виде линейной комбинации n некоторых известных линейно-независимых базисных функций :

|  |  |
| --- | --- |
|  | (9) |

Для уравнения с выделением поля (8) подставим

|  |  |
| --- | --- |
| , | (10) |

|  |  |
| --- | --- |
| . | (11) |

Разобьём область  на непересекающиеся подобласти – конечные элементы: . В трехмерной области возьмем - это параллелепипеды.

Подставим (10) и (11) в уравнение (8):

|  |  |
| --- | --- |
| *,* | (12) |

Тогда матричное уравнение относительно весов , выраженное через веса выглядит так:

|  |  |
| --- | --- |
| . | (13) |

1. *Базисные функции*

Рассмотрим – это параллелепипед. Определим на нем двенадцать (ассоциированных с ребрами) базисных вектор функций таких, что каждая из них имеет направление строго вдоль оси координат, параллельной ребру, с которым она ассоциирована. При этом модуль каждой базисной вектор-функции внутри является билинейной функций двух других координат, направлению осей которых она перпендикулярна.

Локальные базисные функции параллелепипеда можно получить, отображая на него базисные функций, полученные на шаблонном кубе:

Рассмотрим шаблонный элемент .

Введем одномерные скалярные иерархические функции:

|  |  |
| --- | --- |
| , , . | (14) |

Базис первого порядка на содержит 12 вектор-функций, которые через одномерные и могут быть записаны в виде:

|  |  |
| --- | --- |
| . | (15) |

И все эти вектор-функции ассоциированы с ребрами конечного элемента

Тогда на элементе

|  |  |
| --- | --- |
| ,  ,  . | (16) |

1. *Вид локальных матриц и вектора правой части*

Формулы для вычисления компонент глобальных матриц жесткости и массы , определяющие глобальную матрицу конечной элементной СЛАУ, имеют вид:

|  |  |
| --- | --- |
| , . | (17) |

Соответственно вклады в G и M от конечного элемента определяются соотношениями:

|  |  |
| --- | --- |
|  | (18) |

Где – локальные матрицы жесткости и массы конечного элемента , а , его локальные базисные вектор-функции.

Матрицу можно записать в компактном виде, для этого введем подматрицу

|  |  |
| --- | --- |
| , тогда . | (19) |

Где O – состоящая из нулей матрица размера 4 на 4, – размеры параллелепипеда.

Локальная матрица жесткости определяется через три подматрицы:

|  |  |
| --- | --- |
| , ,  . | (20) |

|  |  |
| --- | --- |
| . | (21) |

|  |  |
| --- | --- |
|  |  |

А локальный вектор правой части примет вид:

|  |  |
| --- | --- |
|  | (22) |

|  |  |
| --- | --- |
| . | (23) |

1. *Краевые условия*

Первые краевые условия: . Ребра , Где N – количество ребер с первыми краевыми условиями.

На j-место диагонали в глобальной матрице ставим 1, всю остальную строку зануляем, в правую часть ставим значение функции. Для того, чтобы сохранить симметричность матрицы сделаем один шаг методом Гаусса.

На i-место диагонали в глобальной матриц ставим 1, всю остальную строку зануляем, в правую часть ставим значение функции, умноженное на единичный вектор в направления (по x, по y, по z) ребра. Для симметризации матрицы делаем один шаг методом Гауссом.

1. **Практическая часть**
2. *Тест 1*

Была выбрана область [-5000; 5000] x [-5000; 5000] x [-5000; 5000] где проводимость равна , объект размерами [-5; 5] x [-5; 5] x [-2;-5] с проводимостью . Задали гармонический источник равный , где частота = 10, период равен . Время, когда процесс установился равно 94c

Аномальное поле в области: [-5; 5] x [-5; 5] при z= -3:

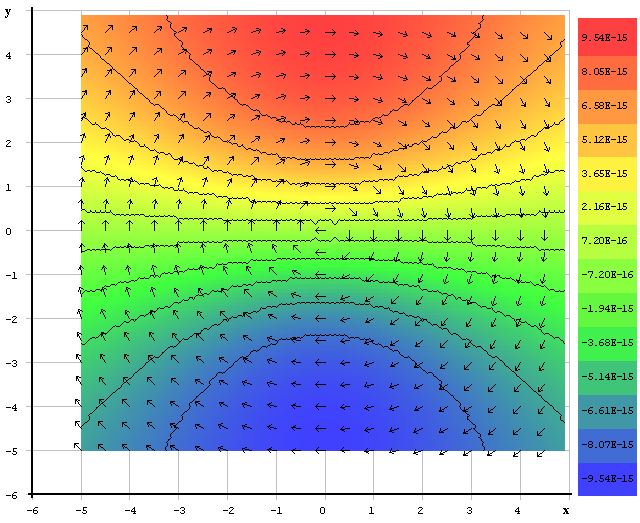


Рисунок 1

1. *Тест 2*

Была выбрана область [-5000; 5000] x [-5000; 5000] x [-5000; 5000] где проводимость равна , объект размерами [-3; 3] x [-3; 3] x [-2;-5] с проводимостью . Задали гармонический источник равный , где частота = 10, период равен . Время, когда процесс установился равно 94c

Аномальное поле в области: [-5; 5] x [-5; 5] при z= -3:

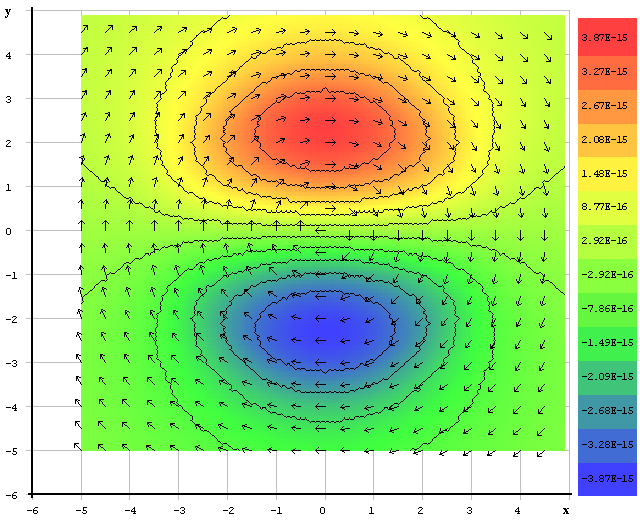


Рисунок 2

1. *Тест 3*

Была выбрана область [-5000; 5000] x [-5000; 5000] x [-5000; 5000] где проводимость равна , объект размерами [-2; 2] x [-2 ;2] x [-2;-5] с проводимостью . Задали гармонический источник равный , где частота = 10, период равен . Время, когда процесс установился равно 94c

Аномальное поле в области: [-5; 5] x [-5; 5] при z= -3:

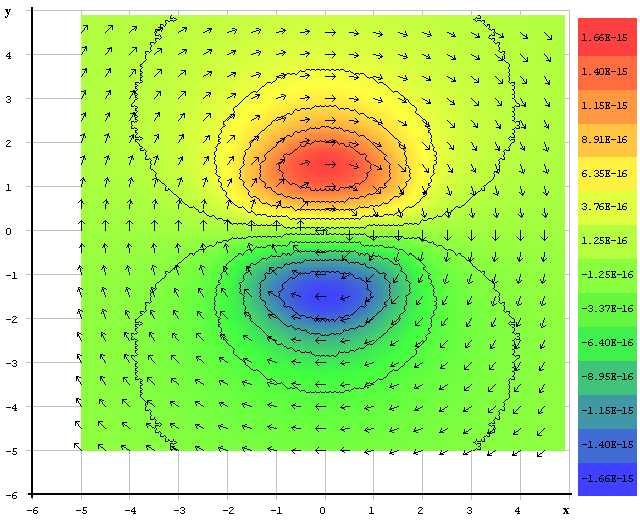
**

Рисунок 3

1. **Выводы**

Проанализировав тесты, видим, что размер объекта влияет на поле. В первом случае объект большой, и поле в нем сильнее. По изолиниям рисунка 2, видим, что объект находится в области [-3; 3] x [-3; 3]. По изолиниям рисунка 3 видим, что объект находится в области [-2;2] x [-2;2], т.к. там оно самое сильное, а в остальной области слабое.

1. **Текст программы**

void normal\_field\_2D::controle(int t\_i)

{

double zhach = cos(w\*time\_mas[t\_i]) + sin(w\*time\_mas[t\_i]);

bool not\_end = true;

for (int i = 0; i<N && not\_end; i++)

{

if ((nodes[el[i].node[0]].r <= r) &&

(nodes[el[i].node[1]].r >= r) &&

(nodes[el[i].node[0]].z <= z) &&

(nodes[el[i].node[2]].z >= z))

{

pr[el[i].node[0]] += zhach\*fi\_1(i, r, z);

pr[el[i].node[1]] += zhach\*fi\_2(i, r, z);

pr[el[i].node[2]] += zhach\*fi\_3(i, r, z);

pr[el[i].node[3]] += zhach\*fi\_4(i, r, z);

not\_end = false;

}

}

}

void normal\_field\_2D::time\_calc\_matrix()

{

FILE \*f = fopen("rec\_xyz.txt", "r");

n\_rec = 10000;

node\_xyz \*nodes\_xyz = new node\_xyz[n\_rec];

for (int i = 0; i < n\_rec; i++)

{

fscanf(f, "%lf %lf %lf", &nodes\_xyz[i].x, &nodes\_xyz[i].y, &nodes\_xyz[i].z);

nodes\_xyz[i].r = sqrt(nodes\_xyz[i].x\*nodes\_xyz[i].x + nodes\_xyz[i].y\*nodes\_xyz[i].y);

}

fclose(f);

AxAy \*\*rec = new AxAy\*[n\_time];

for (int i = 0; i < n\_time; i++)

rec[i] = new AxAy[n\_rec];

printf("nu begin\n");

for (int j = 0; j<n\_gg; j++)

gg[j] = 0;

for (int j = 0; j<m; j++)

{

di[j] = 0;

pr[j] = 0;

}

char name[40];

for (int i = 0; i<n\_time; i++)

{

sprintf(name, "Afinorm%d.txt", i);

f = fopen(name, "w");

if (i == 0)

{

for (int j = 0; j<m; j++)

q0[j] = 0.0;

form\_global\_matrix(i);

s\_MSG.give\_data(ig, jg, gg, di, pr, q0, m, n\_gg);

for (int j = 0; j < m; j++)

{

x0[j] = q0[j];

// fprintf(f, "%.15lf\n", q0[j]);

}

for (int j = 0; j<n\_rec; j++)

{

double val = output\_in\_point(nodes\_xyz[j].r, nodes\_xyz[j].z);

rec[i][j].Ax = -nodes\_xyz[j].y \*val / nodes\_xyz[j].r;

rec[i][j].Ay = nodes\_xyz[j].x \*val / nodes\_xyz[j].r;

}

printf("2D calc q0\n");

}

if (i > 1)

{

for (int j = 0; j<m; j++)

{

q2[j] = 0.0;

x0[j] = 0.0;

}

form\_global\_matrix(i);

s\_MSG.give\_data(ig, jg, gg, di, pr, q2, m, n\_gg);

for (int j = 0; j < m; j++)

{

x0[j] = q2[j];

//fprintf(f, "%.15lf\n", q2[j]);

}

for (int j = 0; j<n\_rec; j++)

{

double val = output\_in\_point(nodes\_xyz[j].r, nodes\_xyz[j].z);

rec[i][j].Ax = -nodes\_xyz[j].y \*val / nodes\_xyz[j].r;

rec[i][j].Ay = nodes\_xyz[j].x \*val / nodes\_xyz[j].r;

}

for (int ii = 0; ii<m; ii++)

{

q0[ii] = q1[ii];

q1[ii] = q2[ii];

q2[ii] = 0;

}

printf("2D calc q%d\n", i);

}

if (i == 1)

{

for (int j = 0; j<m; j++)

{

q1[j] = 0;

x0[j] = 0;

}

form\_global\_matrix(i);

s\_MSG.give\_data(ig, jg, gg, di, pr, q1, m, n\_gg);

for (int j = 0; j < m; j++)

{

x0[j] = q1[j];

//fprintf(f, "%.15lf\n", q1[j]);

}

for (int j = 0; j<n\_rec; j++)

{

double val = output\_in\_point(nodes\_xyz[j].r, nodes\_xyz[j].z);

rec[i][j].Ax = -nodes\_xyz[j].y \*val / nodes\_xyz[j].r;

rec[i][j].Ay = nodes\_xyz[j].x \*val / nodes\_xyz[j].r;

}

printf("2D calc q1\n");

}

fclose(f);

}

printf("nu end\n");

string file\_name = "res.txt";

ofstream outpf(file\_name.c\_str());

outpf << "TITLE = \"Slice\" \n";

outpf << "VARIABLES = \"x\" \"y\" \"Ax\" \"Ay\" \n";

size\_t n\_x = 100, n\_y = 100;

outpf << "ZONE I= " << n\_x << ", J= " << n\_y << ", F=POINT\n";

for (int j = 0; j < n\_rec; j++)

{

outpf << nodes\_xyz[j].x << " " << nodes\_xyz[j].y << " " << rec[1500][j].Ax << " " << rec[1500][j].Ay << endl;

}

outpf << endl;

outpf.close();

}