в XVIIв. (основываясь на разборе многочисленных частных случаев), но лишь на рубеже XVIII–XIXвв. упомянутая теорема была доказана Гауссом.

В течение XVIIв. продолжалось обсуждение арифметической природы мнимых чисел, возможности дать им геометрическое обоснование. Постепенно развивалась техника операций над мнимыми числами.

Матрицей размеров $m \cdot n$ называют прямоугольную таблицу чисел или буквенных выражений, содержащую $m$ строк и $n$ столбцов. Квадратная матрица порядка $n$ называется матрица ($A_{ij}$), у которой число строк равно числу столбцов $i = j$. По определенному правилу каждую квадратную матрицу $A$ можно поставить в соответствие с числом, которое называется определителем. Результатом умножения квадратной матрицы на число является матрица, каждый элемент которой умножен на это число.

Для вычисления нелинейных уравнений 2–4 степени используется метод обратной матрицы, а также метод Крамера.

Созданный проект позволяет производить:

1. Вычисления с комплексными числами, записанными в формах вида:

$$zi = a + bi;$$
$$zt = r(\cos(f) + \sin(f)i);$$

$$r = sqrt(a^2 + b^2);$$
$$f = artg(b/a);$$
$$a = r \cdot \cos(f);$$
$$b = r \cdot \sin(f).$$

Калькулятор выдает результаты как в алгебраической, так и в тригонометрической форме записи. В калькуляторе работает созданный модуль "complex", в котором происходят все операции, связанные с вычислением результата любого комплексного выражения, а также переводы в строку и заполнения полей.

Основная цель программы – это опознавание любого вычисления и налаживание связи с созданным модулем "complex". Процедура procedure StrToComplex(val:string; var z:TComplexEx) анализирует строку *val* на соответствие формам записи комплексных чисел и заполняет структуру *z*.

В начале проверяется форма записи: если в строке присутствуют 'cos' и 'sin', то считаем, что строка соответствует тригонометрической форме записи; если нет ни того ни другого, то алгебраической форме; если есть одно, но нет другого, например 'cos' без 'sin', то выдаем сообщение об ошибке формы записи.

Далее, в зависимости от формы записи, разбиваем строку на компоненты, и в случае удачной интерпретации, заполняем структуру *z*.

2. Простые математические вычисления.

Раздел простых математических вычислений включает в себя:
- выполнения всех действий над простыми числами;
- вычисления квадратного корня и числа в $n$-ой степени;
- вычисления тригонометрических функций.

В этой части проекта автоматически выводится справочная информация о правилах вычисления;

3. Действие над матрицей и вычисления нелинейных уравнений 2–4 степени.

Вычисления нелинейных уравнений 2–4 степени производится с помощью метода Крамара или методом Обратной матрицы.

В проекте созданы Procedure и Function для вычисления определителя матрицы 2–4 степени, а также вычисления корней нелинейных уравнений 2–4 степени. В процедурах кнопок вычисления корней сразу же вычисляется определитель и выводится его результат, но если определитель равен 0, то в результативном окне выводится сообщение о «Бесконечном количестве корней». После чего проверяется условие размера матрицы и выполняется тело программы и выводится результат.

В проекте производятся действия над матрицами: нахождение определителя и ранга матрицы, нахождение обратной матрицы, умножение матрицы на число. Результат умножения матрицы на число выводится в дополнительно созданную таблицу.

Минимальные технические требования для использования разработанного приложения следующие: Pentium 2, RAM 64, VC 32, Windows 9x, 2000, XP.

Разработанный проект нашел практическое применение на занятиях по высшей математике для управляемой самостоятельной работы студентов I курса инженерного факультета УО БарГУ.

## ЛИНИИ УРОВНЯ

**А.В. Фридрик**

Научный руководитель: О.И. Наранович

Цель проекта – создать уникальную программу, которая бы строила линии уровня поверхностей, заданных формулами.

В последнее время резко возрос интерес к программированию. Это связано с развитием и внедрением в повседневную жизнь информационно-коммуникационных технологий. Если человек имеет дело с компьютером, то рано или поздно у него возникает желание, а иногда и необходимость программировать.

Математическое моделирование – это стремительно развивающаяся область науки, касающаяся, на сегодняшний день, едва ли не любого аспекта нашей жизни. Результаты исследований и прочие научные достижения, особенно носящие прикладной характер, предстают перед нами в совершенно иной форме, удобной для восприятия и интуитивно более понятной. Визуализация делает модель, с одной стороны, более доступной для анализа, представляя ее в виде последовательности образов и метафорических конструкций, а с другой – более гибкой и даже интерактивной, обеспечивая некое подобие обратной связи между человеком, пользующимся результатами моделирования и собственно моделью. Для представления информации о предмете в разных осях и плоскостях используют линии уровня, которые отображают более полное, развернутое изображение.

Линиями уровня называются линии на плоскости *XOY*, в точках которой функция сохраняет постоянное значение. На практике линии уровня используются во многих областях [1]:

- топографии – для отображения рельефа местности;
- почвоведении – для изображения слоев почвы;
- метеорологии – для описания слоев атмосферы.

Программа выполняет построение линий уровня одно- и двуполостного гиперболоида, конуса, эллипсоида – поверхностей, которые задаются уравнениями [1]. В проекте для отображения изображения используется класс TCanvas [2], который предоставляет возможность рисовать чертежными инструментами: пером, кистью и шрифтом.

Минимальные технические требования для использования разработанного приложения следующие: Pentium 2, RAM 64, VC 32, Windows 9x, 2000, XP.

Разработанный проект можно использовать для визуализации решения дифференциальных уравнений в частных производных, а также на занятиях по высшей математике при изучении темы «Линии уровня».

**Литература**


1. Гусак А.А. Высшая математика: учеб. для студентов вузов: в 2 т. Т. 1.– 5-е изд.– Мн.: ТетраСистемс, 2004.– 544 с.
2. Фаронов В.В. Ф24 Delphi. Программирование на языке высокого уровня: учеб. для вузов.– СПб.: Питер, 2005.– 640 с.: ил.


## ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ КРАЕВОЙ ЗАДАЧИ ДЛЯ СИСТЕМЫ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ (ОДУ) ВТОРОГО ПОРЯДКА


**М.С. Чернышев, А.М. Филипчик**

Научный руководитель: О.И. Наранович


При решении многих задач математической физики дискретизация дифференциальных уравнений в частных производных приводит к системам линейных алгебраических уравнений (СЛАУ), содержащих десятки и сотни уравнений, при этом матрицы коэффициентов содержат большое количество нулей.

Для решения поставленной задачи предпочтительнее использовать итерационные методы, если размерность системы велика для имеющегося объема оперативной памяти компьютера, на котором предполагается решать задачу.

Эти методы позволяют оптимизировать процесс решения в зависимости от име-ющихся вычислительных ресурсов компьютера. При этом получается приближенное решение. Максимальная точность ограничивается допустимым временем решения задачи и разрядной сеткой компьютера.

Предлагается эффективный алгоритм решения краевой задачи для системы ОДУ второго порядка методом Зейделя. В качестве численного исследования полученного алгоритма рассмотрена задача Пуассона:

$$\frac{d^2U}{dz^2}+\frac{d^2U}{dr^2}=f,$$

добавим простые граничные условия (для теста): $U(z, r=0)=0$, $u(z, r=1)=0$, $u(z=L, r)=0$, $u(z=0, r)=\sin(\pi r)$.

На интервале $0 < r < 1$ выберем равномерную сетку:

$\{r_j=h_r(j-1),\ h_r=1/M, j=1..M=1\}$.

### Метод итераций Зейделя

$$\frac{U_{ij-1}-2U_{ij}+U_{ij+1}}{h_r^2}+\frac{U_{i-1j}-2U_{ij}+U_{i+1j}}{h_z^2}=f,$$

$$h_z^{\,2}(U_{ij-1}-2U_{ij}+U_{ij+1})+h_r^{\,2}(U_{i-1j}-2U_{ij}+U_{i+1j})=(h_rh_z)^2 f_{ij},$$