

УДК 004.42

ПРАКТИКУМ ВИЗУАЛЬНОГО РЕШЕНИЯ КРАЕВЫХ ЗАДАЧ МЕТОДОМ ПРОГОНКИ НА VBA MICROSOFT EXCEL

О. А. Широкова¹, [0000-0002-2883-0415]

Казанский (Приволжский) федеральный университет

¹shirokova2602@mail.ru

Аннотация

Рассмотрено построение визуального решения краевых задач для линейного дифференциального уравнения второго порядка методом трехдиагональной прогонки на VBA в Excel. Создание макросов на VBA позволяет визуализировать решения таких краевых задач.

Ключевые слова: краевые задачи, дифференциальное уравнение, трехдиагональная прогонка, макрос, Visual Basic Application, Excel

Дисциплина «Программирование» является важной в системе подготовки студентов математических факультетов. Сейчас существует очень много программ для всех сфер деятельности, но есть один универсальный пакет программ, которого достаточно для решения большого круга задач. Это пакет Microsoft Office производства корпорации Майкрософт. Изучение пакета Microsoft Office входит в учебные программы школ и вузов. Одной из важных частей данного пакета, расширяющих его возможности, являются макросы. В своей основе они используют программирование на языке Visual Basic Application (VBA). Для чего нужны макросы? Часто бывает, что в используемой программе нет нужной функции, поэтому приходится искать другие программы, способные ее реализовать. Макросы – это способ автоматизации и расширения функций приложения, в нашем случае приложения Excel. Таким образом, используя макросы, мы можем автоматизировать однотипные действия, добавив в приложение новые функции, не используя другие программы, которые могут стоить больших денег.

Практически всегда можно утверждать, что приложение, которое легко модифицировать, чтобы оно отвечало нужным требованиям, обладает большой привлекательностью для пользователей. Создание макросов возможно во всех приложениях Microsoft Office, и это является одной из причин, почему этот пакет

настолько популярен. Ведь практически все продукты, работающие с данными, имеют функцию экспорта в продукты этого пакета. Даже названия продуктов этого пакета (Word, Excel, PowerPoint, Access, Outlook) становятся нарицательными.

В статье рассмотрено построение визуального решения краевых задач для линейного дифференциального уравнения второго порядка методом трехдиагональной прогонки на VBA в Excel. Создание макросов на VBA позволяет визуализировать решения таких краевых задач.

При исследовании вопросов, связанных с теорией сопряженных уравнений [4], и при разработке алгоритмов возмущения [9] для решения неоднородных стационарных задач возникает насущная потребность в качественном решении краевых задач для линейных дифференциальных уравнений второго порядка. Обычно при этом прибегают к численному решению с использованием методов прогонки [1].

Краевая задача

Рассмотрим краевую задачу для линейного дифференциального уравнения второго порядка

$$y'' + p(x)y' + q(x)y = F(x),$$

где $p(x)$, $q(x)$ и $F(x)$ – заданные функции при $q(x) < 0$ на отрезке $[a, b]$.

Заданы краевые условия: $y(a) = \alpha$, $y(b) = \beta$.

Запишем краевую задачу для линейного дифференциального уравнения второго порядка в каноническом виде:

$$\begin{aligned} A_i y_{i-1} + B_i y_i + C_i y_i &= f_i; \\ A_i &= 1 - \frac{p_i h}{2}; \\ B_i &= -2 + q_i h^2; \\ C_i &= 1 + \frac{p_i h}{2}, \quad f_i = h^2 F_i. \end{aligned} \tag{1}$$

Метод прогонки основан на предположении, что искомые значения функции y_i связаны рекуррентным соотношением

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = n-1, n-2, \dots, 1, \tag{2}$$

которое называется обратной прогонкой. Здесь коэффициенты α_{i+1} и β_{i+1} вычисляются методом прямой прогонки

$$\alpha_{i+1} = \frac{C_i}{b_i + a_i \alpha_i}, \quad i = 0, \dots, n,$$

$$\beta_{i+1} = \frac{a_i \beta_i - d_i}{b_i - a_i \alpha_i}. \quad (3)$$

Алгоритм вычисления

Одним из самых простых способов решения задачи является сведение ее к системе конечно-разностных уравнений.

Для этого необходимо разбить отрезок $[a, b]$ на n равных частей длины h , где $h = (b - a)/n$. Точками разбиения абсциссы будут $x_i = x_0 + ih$, где $x_0 = a$, $x_n = b$.

Значение функции $y(x)$ в точках x_i и ее производных обозначим соответственно $y(x_i) = y_i$, $y''(x_i) = y_i''$, $y'(x_i) = y_i'$. Введем следующие обозначения $p_i = p(x_i)$, $q_i = q(x_i)$, $f_i = f(x_i)$. Заменяем производные центрально-разностными отношениями. Для внутренних точек отрезка $[a, b]$ будем иметь

$$y_i' = (y_{i+1} - y_{i-1})/2h;$$

$$y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}; \quad i = 1, 2, \dots, n - 1.$$

Можно представить краевую задачу в виде

$$\alpha_1 y_0 + \alpha_2 ((y_1 - y_0)/h) = A;$$

$$\beta_1 y_n + \beta_2 ((y_n - y_{n-1})/h) = B.$$

Таким образом, получили систему $n + 1$ линейных уравнений с $n + 1$ неизвестными y_0, y_1, \dots, y_n , решение которой позволяют найти значение искомой функции $y(x)$ в точках x_0, x_1, \dots, x_n . Система является трехдиагональной, и для ее решения используется метод прогонки. В ходе прямого хода определяем прогоночные коэффициенты α_k и β_k . В ходе обратного хода определяем все неизвестные y_i последовательно, начиная с y_n .

Визуальный проект

Рассмотрим создание визуального проекта [6–8, 10–13] решения краевых задач (1) методом трехдиагональной прогонки. Для решения создаются макросы в VBA Excel [2, 3, 5].

Отличительной особенностью VBA является использование наряду с обычными переменными и константами языка также и тех объектов, которые имеются

в конкретном приложении Microsoft Office. Например, в Microsoft Office Excel это могут быть рабочие книги, рабочие листы, диаграммы, диапазоны ячеек и т. д. С помощью VBA можно разработать приложения, которые включают различные компоненты нескольких приложений Microsoft Office и способствуют тем самым интеграции и совместному использованию данных.

Язык VBA использует технологию визуального программирования, т. е. конструирования рабочей поверхности приложения и элементов его управления непосредственно на экране, а также запись всей программы или ее частей при помощи MacroRecorder. Для создания диалоговых окон в приложениях VBA используются формы. Редактор форм является одним из основных инструментов визуального программирования. Используя панель элементов из незаполненной формы, можно сконструировать любое диалоговое окно, требуемое для приложения.

VBA обладает встроенным набором элементов управления. Элементы управления являются объектами. Поэтому, как любые объекты, они обладают свойствами, методами, событиями. На начальном этапе конструирования приложения происходит размещение элементов управления на форме. Большинство элементов управления можно располагать как на форме, так и на рабочем листе (Рис. 1).

Макрос на языке VBA создается с использованием MacroRecorder. MacroRecorder представляет собой транслятор, создающий программу (макрос) на языке VBA, которая является результатом трансляции на язык VBA действий пользователя с момента запуска макрорекордера до окончания записи макроса.

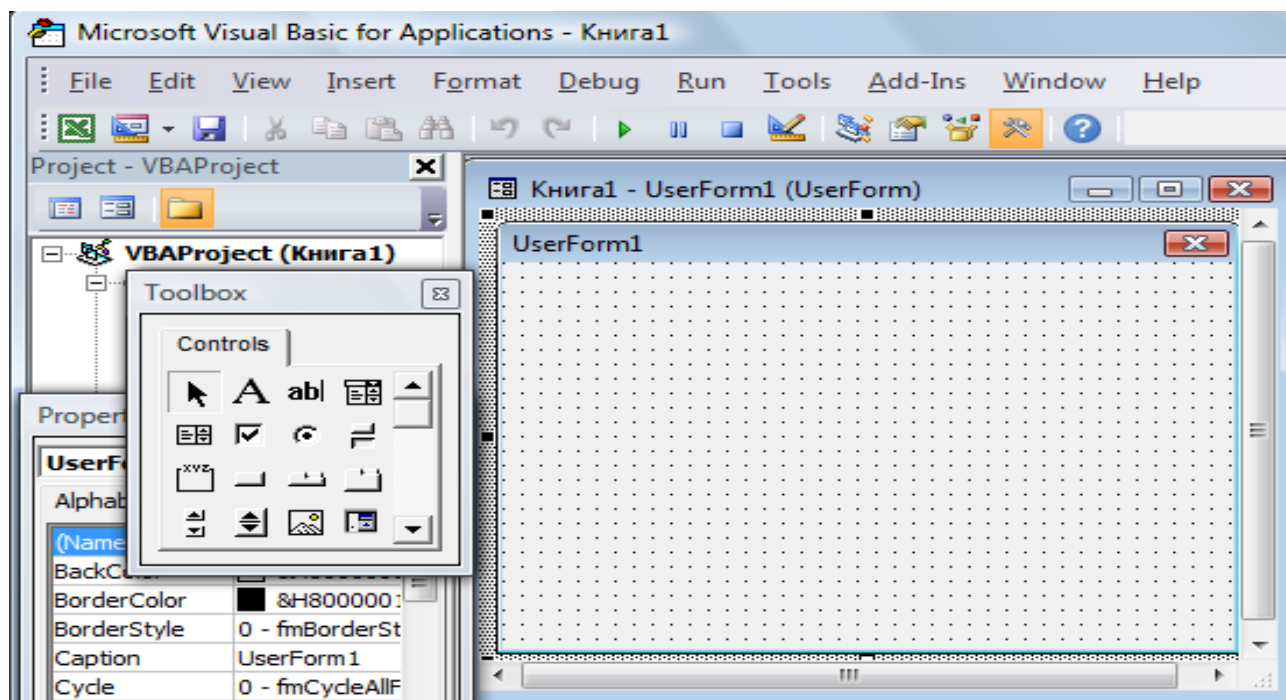


Рис. 1. Окно редактирования форм и панель элементов




Основное назначение макросов — автоматизация работы в конкретном приложении Microsoft Office. Кроме этого, созданный код макроса может служить основой для дальнейших разработок пользователя [5].

При записи макроса запоминаются все действия пользователя, которые автоматически преобразуются в программный код на языке VBA.


Каждому макросу дается имя, а для быстрого запуска макроса можно создать или присвоить ему “горячую” клавишу (клавишу, при нажатии на которую будет производиться запуск макроса). После запуска макрос будет автоматически выполнен тем приложением, в котором он создан и запущен. При выполнении макроса компьютер воспроизводит все действия пользователя.

В рассматриваемом случае с помощью макроса необходимо визуально разработать интерфейс главного окна проекта решения краевых задач методом трехдиагональной прогонки. Интерфейс проекта в виде диалогового окна с кнопками, полями ввода и другими компонентами позволяет решать поставленную задачу (1).

Для ввода исходных данных и отображения результата окно приложения должно содержать окна ввода коэффициентов линейного дифференциального уравнения второго порядка и краевых условий.

Добавим нужные нам поля на форму, используя TextBox , подписи, используя Label , и кнопки, используя CommandButton .

Для отделения части формы можно использовать Frame .

Для выбора вида результатов решения, а именно, либо только значений x и y , либо графика искомой функции, добавим CheckBox .

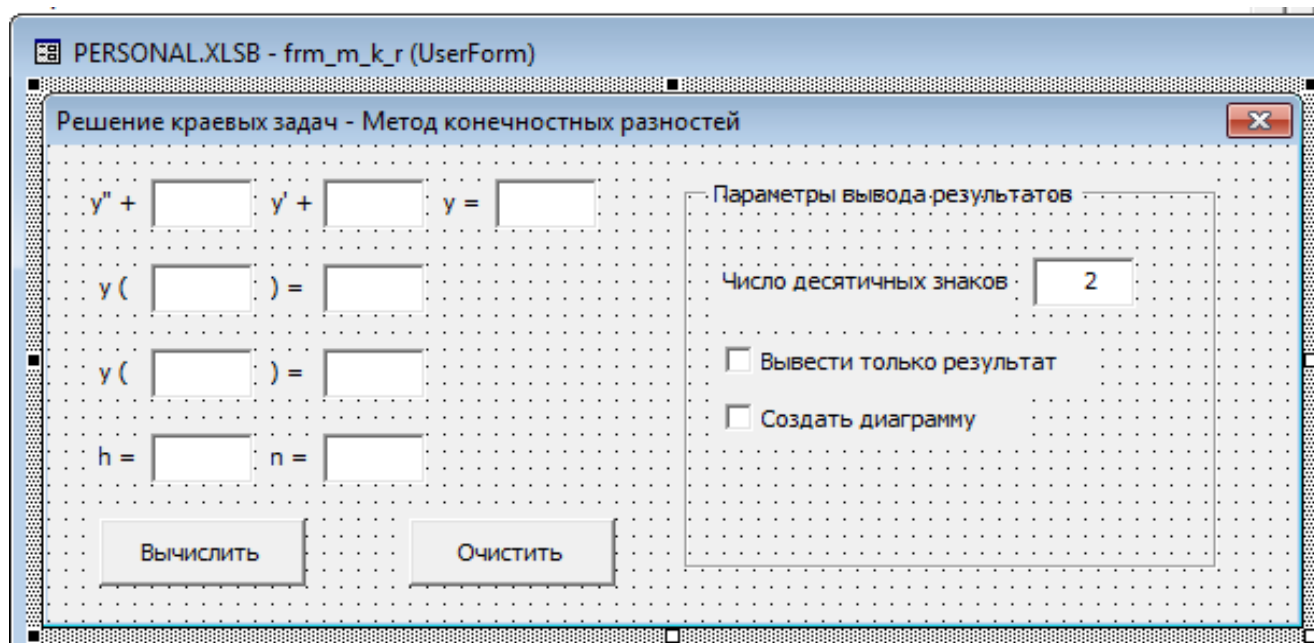


Рис. 2. Форма визуального проекта

В поля TextBox можно ввести начальный текст, вписав его в поле «Text» свойств. Для остальных объектов текст задается в поле «Caption» свойств. Каждый объект на форме, как и сама форма, имеет свойство (Name). Имена объектов можно изменить на явные и понятные.

Так как при вычислении мы строим таблицу, то в программе используется двумерный массив вещественных чисел. Для вычисления A, B, C, F используем формулы (1). В программе запишем:

```
' Вычисляем A, B, C, F
For i = 1 To n - 1
    masResh(i, 6) = 1 - masResh(i, 3) * h / 2 'A
    masResh(i, 7) = 1 + masResh(i, 3) * h / 2 'B
    masResh(i, 8) = 2 - masResh(i, 4) * h * h 'C
    masResh(i, 9) = masResh(i, 5) * h * h 'F
```

Next i

Теперь, когда есть система с трехдиагональной матрицей коэффициентов, можно применить прямой ход прогонки. Вычисляем прогоночные коэффициенты по формулам (3):

' Прямой ход прогонки

$masResh(1, 10) = masResh(1, 7) / masResh(1, 8)$

$masResh(1, 11) = -1 * masResh(1, 9) / masResh(1, 8)$

For i = 2 To n - 1

$masResh(i, 10) = masResh(i, 7) / (masResh(i, 8) - masResh(i - 1, 10) * masResh(i, 6))$

$masResh(i, 11) = (masResh(i, 6) * masResh(i - 1, 11) - masResh(i, 9)) / (masResh(i, 8) - masResh(i - 1, 10) * masResh(i, 6))$

Next i

' Конец прямого хода прогонки

Используем обратный ход прогонки для вычисления y_i по формуле (2)

' Обратный ход прогонки

For i = n - 1 To 1 Step -1

$masResh(i, 2) = masResh(i, 10) * masResh(i + 1, 2) + masResh(i, 11)$

Next i

' Конец обратного хода прогонки

Далее применим настройки вывода, указанные на форме, для вывода либо только значений x и y , либо графика искомой функции. Здесь используются специфические функции VBA для возврата полученных результатов на рабочий лист MS Excel.

В VBA есть удобный объект Range – диапазон ячеек. В паре со специальным запросом для определения диапазона можно получить и вставить данные на любое место любого листа любой открытой книги. Создадим переменную типа Range:

Dim rngResh As Range

Теперь создадим запрос пользователю на определение позиции вывода результатов. Но нужно предусмотреть нажатие кнопки «Отмена».

On Error Resume Next

```
Set rngResh = Application.InputBox("Выберите мышкой расположение резуль-  
тата",  
"Вывод решения", Type:=8)  
If rngResh Is Nothing Then  
    MsgBox "Вывод результатов отменен!"  
    Exit Sub  
End If
```

Это диалоговое окно позволяет выбрать диапазон ячеек или одну ячейку для вывода полученных результатов. Выводим полученное нами решение:

```
For i = 0 To n  
    For j = 1 To 11  
        rngResh.Cells(i + 1, j) = masResh(i, j)  
    Next j  
Next i
```

Следующий пункт настроек «Вывести только результат», т. е. только значения x и y . Для применения настройки будем проверять галочку на компоненте CheckBox1.

```
If CheckBox1.Value Then  
    For i = 0 To n  
        For j = 1 To 2  
            rngResh.Cells(i + 1, j) = Round(masResh(i, j), CInt(txtOptDrob.Text))  
        Next j  
    Next i  
Else  
    For i = 0 To n  
        For j = 1 To 11  
            rngResh.Cells(i + 1, j) = Round(masResh(i, j), CInt(txtOptDrob.Text))  
        Next j  
    Next i  
End If
```

Переходим к последнему пункту настроек «Создание диаграммы». Здесь используются только специфические функции VBA MS Excel. Суть в следующем:

программно выделяются значения y , создается диаграмма в отдельной вкладке, выбирается тип диаграммы «График», убирается легенда, добавляются подписи на оси x из значений x :

```
If CheckBox2.Value Then
    rngResh.Range("C2:C" + CStr(n + 2)).Select
    Set Diagramma = ActiveWorkbook.Charts.Add
    Diagramma.ChartType = xlLine
    Diagramma.SetElement (msoElementLegendNone)
    Diagramma.SeriesCollection(1).XValues = rngResh.Range("B2:B" + CStr(n + 2))
End If
```

Записываем программные действия для очистки формы:

```
txtP.Text = ""
txtQ.Text = ""
txtF.Text = ""
txtA.Text = ""
txtB.Text = ""
txtY0.Text = ""
txtYn.Text = "" ....
```

Также можно добавить проверку ввода данных. Для этого добавим проверку на пустые поля ввода. Исключение составляют поля $P(x)$ и $Q(x)$. Если они пустые, будем считать, что они равны 1.

```
If (txtA.Text = "") Or (txtB.Text = "") Or (txtY0.Text = "") Or (txtYn.Text = "") Or (txtN.Text = "") Or (txtF.Text = "") Then
```

```
    MsgBox "Заполнены не все поля!"
```

```
    Exit Sub
```

```
End If
```

```
If txtP.Text = "" Then txtP.Text = "1"
```

```
If txtQ.Text = "" Then txtQ.Text = "1"
```

```
If txtOptDrob.Text = "" Then txtOptDrob.Text = "2"
```

Добавим автоматический выход, по завершению всех действий.

```
' Автоматический выход из программы
```

```
MsgBox "Готово!"
```

End

' Конец автоматического выхода

На следующем этапе нам следует полностью оптимизировать код, объединив несколько циклов и все другие действия. Этот файл можно передать другому пользователю через экспорт формы и кода.

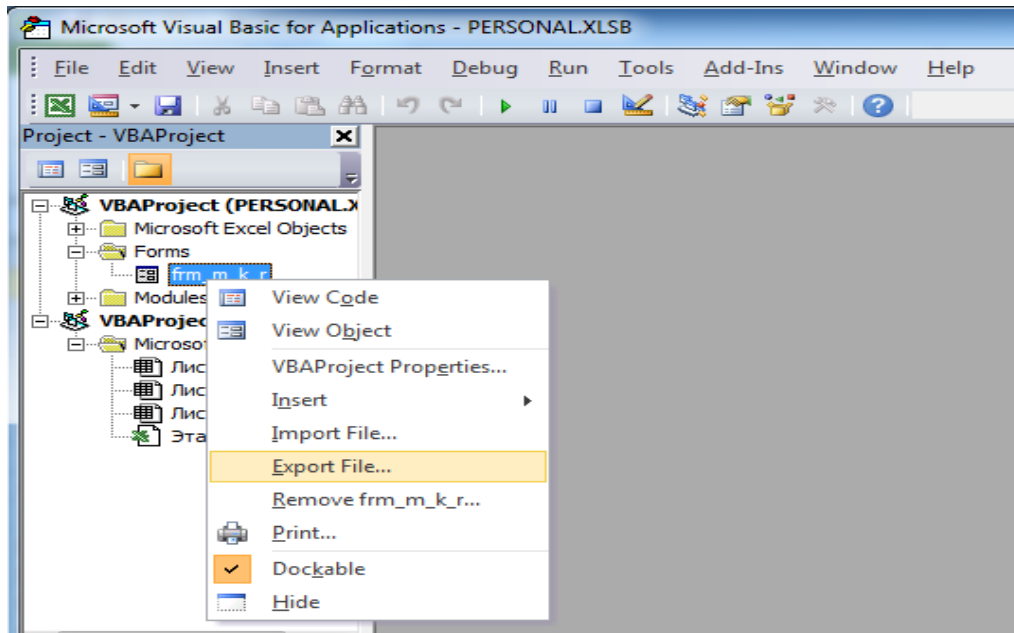


Рис. 3. Экспорт формы

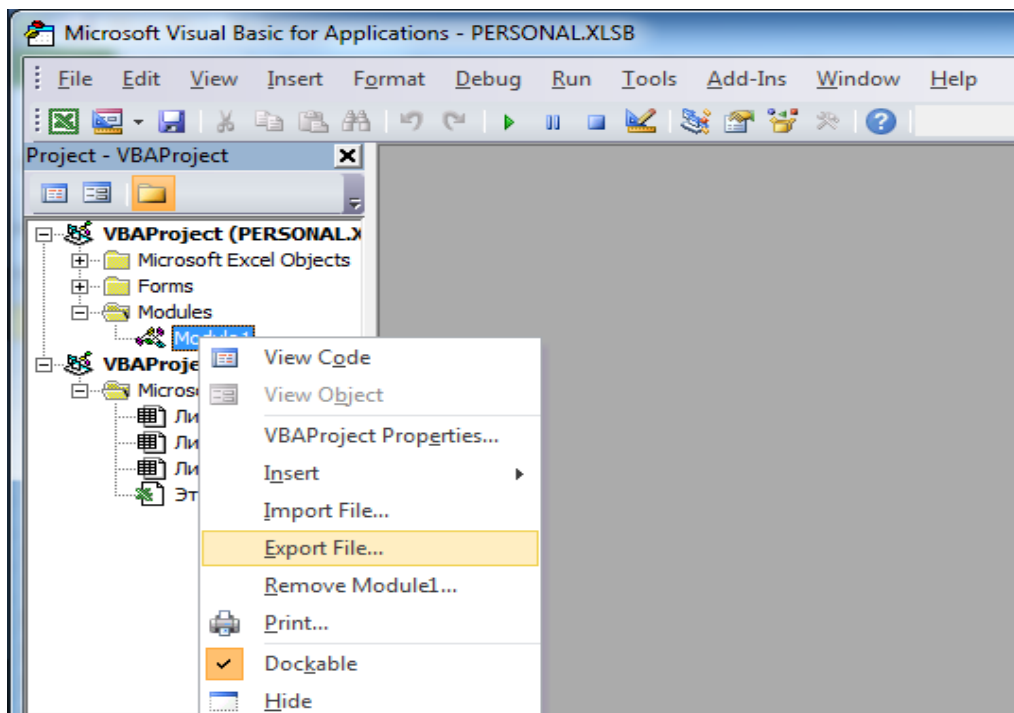


Рис. 4. Экспорт кода

Данным способом можно передать макрос в другие приложения, поддерживающие макросы VBA, например, в MS Word или CorelDraw. Однако возможно, что для других приложений код нужно будет изменить.

Поскольку использована программа со своей формой, то запуск формы имеет вид:

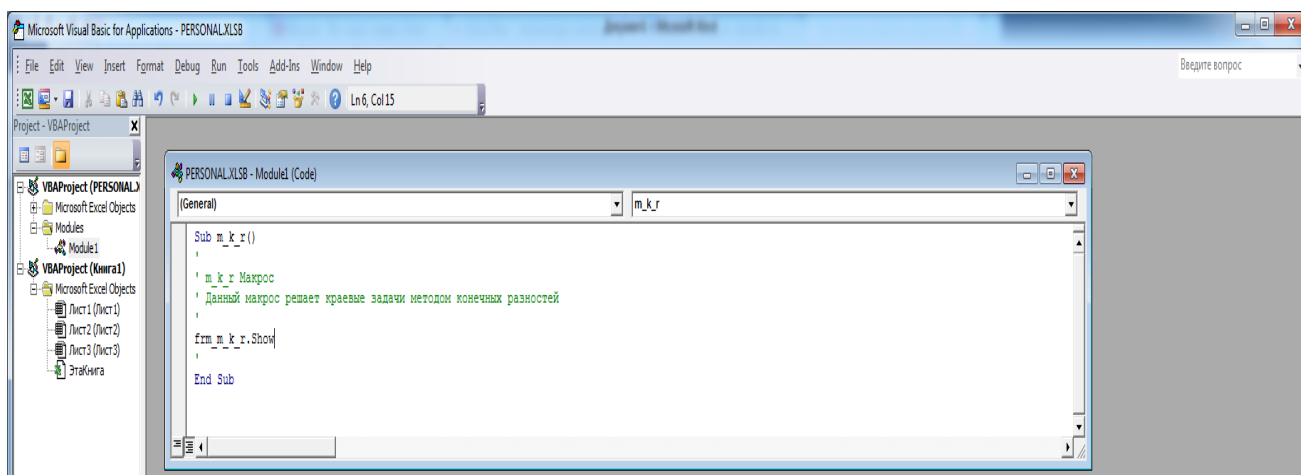


Рис. 5. Запуск формы

Запустить макрос на выполнение можно, используя командную кнопку. Вставить командную кнопку на рабочий лист Excel можно с помощью панели «Элементы управления». Затем необходимо назначить этой кнопке макрос. Выбрать команду «Свойства». Откроется окно «Properties», где нужно изменить надпись на кнопке. В окне кода рабочего листа после заголовка процедуры нужно ввести инструкцию Call с названием надписи на кнопке: «Решение краевых задач методом прогонки». Для запуска макроса достаточно будет нажать кнопку с этим названием.

Построение визуального решения краевых задач для линейного дифференциального уравнения второго порядка методом трехдиагональной прогонки с созданием макросов на VBA в Excel формирует важные профессиональные компетенции будущих учителей математики и информатики. Практикум построения визуальных решений с созданием макросов в Excel вырабатывает у студентов умения алгоритмизации математических задач с переводом их на язык компьютерных программ. Построение визуальных решений на языке Visual Basic for Applications в Excel учит ориентироваться в различных средах программирования,

проводить проверку, отладку и тестирование программных кодов на языках программирования высокого уровня.

СПИСОК ЛИТЕРАТУРЫ

1. *Вержбицкий В.М.* Основы численных методов: Учебник для вузов. М.: Высш. шк., 2002. 840 с.: ил.
2. *Гарнаев А.Ю., Рудикова Л.В.* Microsoft Office Excel 2010: разработка приложений. СПб.: БХВ-Петербург, 2011. 528 с.
3. *Ларсен Рональд.* Инженерные расчеты в Excel. М.: Изд-во «Вильямс», 2004. 544 с.
4. *Марчук Г.И.* Сопряженные уравнения и анализ сложных систем. М.: Наука. Гл. ред. физ.-мат. лит., 1992. 336 с.
5. *Гайнанова Р.Ш., Широкова О.А.* Программирование на Visual Basic for Applications в Excel: учебное пособие. Казань: КФУ, 2012. 153 с.
6. *Широкова О.А.* Особенности обучения программированию на основе общности и различия принципов // Современные проблемы науки и образования. 2015. № 1. С. 1757. URL: <http://www.science-education.ru/121-17896>
7. *Широкова О.А.* Объектно-ориентированные проекты решения математических задач // Материалы XI Международной науч.-практ. конф. «Объектные системы – 2015» (Ростов-на-Дону, 10–12 декабря 2015 г.) / Под общ. ред. П.П. Олейника. Ростов-на-Дону: ШИ (ф) ЮРГПУ (НПИ) им. М.И. Платова, 2015. С. 15–22.
8. *Широкова О.А.* Объектно-ориентированные проекты с созданием классов для математических объектов // Электронные библиотеки. Тематический выпуск «Математическое образование в школе и вузе». 2020. Т. 23, № 1-2. С. 216–223.
9. *Shirokova O.A.* Methods of demolition of the boundary conditions by perturbation methods for solving filtration problems with free boundaries // IOP Conference Series: Materials Science and Engineering, 2019.
10. *Gainutdinova T.Yu., Denisova M.Yu., Riazanova L.V., Shakirova Z.F., Shirokova O.A.* Modelling mathematical structures and object-oriented programming // DILEMAS CONTEMPORANEOS-EDUCACION POLITICA Y VALORES. 2019. Vol. 6. Art. No. 12.

11. Gainutdinova T.Y., Denisova M.Y., Shirokova O.A. The use of digital and information technologies in order to increase the effectiveness of mathematical education // Talent Development and Excellence. 2020. Vol. 12. P. 188–198.

12. Denisova M., Gainutdinova T., Shirokova O. Multimedia technologies as a means of introducing game elements in the process of teaching mathematics, EDULEARN19 Proceedings. 2019. P. 1189–1194.

13. Shirokova O., Gainutdinova T., Denisova M. Interdisciplinary integration as a condition for the modernization of teacher education, EDULEARN19 Proceedings. 2019. P. 1183–1188.

PRACTICE OF VISUAL SOLUTION OF BOUNDARY VALUE PROBLEMS BY RUNNING VBA IN MICROSOFT EXCEL

O. A. Shirokova¹, [0000-0002-2883-0415]

Kazan Federal University, Kazan

shirokova2602@mail.ru

Abstract

The article discusses the construction of a visual solution of boundary value problems for a linear differential equation of the second order by the tridiagonal sweep method in VBA in Excel. Creating macros in VBA allows you to visualize solutions to such boundary value problems

Keywords: *boundary value problems, differential equation, tridiagonal sweep, macro, Visual Basic Application, Excel*

REFERENCES

1. Verzhbitsky V.M. Fundamentals of numerical methods: Textbook for universities. M.: Vysh. shk., 2002. 840 p.: il.
 2. Garnaev A.Yu., Rudikova L.V. Microsoft Office Excel 2010: application development. St. Petersburg: BHV-Petersburg, 2011. 528 p.
 3. Larsen Ronald. Engineering calculations in Excel. M.: Publishing house "Williams," 2004. 544 p.
-

4. *Marchuk G.I.* Conjugated equations and analysis of complex systems. M.: Science. Ch. edition Fiz.-mat. Lit., 1992. 336 p.
5. *Gainanova R.Sh., Shirokova O.A.* Programming on Visual Basic for Applications in Excel: tutorial. Kazan: KFU, 2012. 153 p.
6. *Shirokova O.A.* Features of programming training based on commonality and differences of principles // Modern problems of science and education. 2015. No. 1, P. 1757. URL: <http://www.science-education.ru/121-17896>
7. *Shirokova O.A.* Object-oriented projects for solving mathematical problems//Materials of the XI International Scientific and Practical. conf. "Object Systems–2015" (Rostov-on-Don, December 10–12, 2015)/Under the common. Ed. P.P. Oleinik. Rostov-on-Don: SHI (f) YURGPU (NPI) named after M.I. Platova, 2015. P. 15–22.
8. *Shirokova O.A.* Object-oriented projects with the creation of classes for mathematical objects // Electronic libraries. Thematic issue "Mathematical education at school and university". 2020. Vol. 23, No. 1-2. P. 216–223.
9. *Shirokova O.A.* Methods of demolition of the boundary conditions by perturbation methods for solving filtration problems with free boundaries // IOP Conference Series: Materials Science and Engineering, 2019.
10. *Gainutdinova T.Yu., Denisova M.Yu., Riazanova L.V., Shakirova Z.F., Shirokova O.A.* Modelling mathematical structures and object-oriented programming // DILEMAS CONTEMPORANEOS-EDUCACION POLITICA Y VALORES. 2019. Vol. 6. Art. No. 12.
11. *Gainutdinova T.Y., Denisova M.Y., Shirokova O.A.* The use of digital and information technologies in order to increase the effectiveness of mathematical education // Talent Development and Excellence. 2020. Vol. 12. P. 188–198.
12. *Denisova M., Gainutdinova T., Shirokova O.* Multimedia technologies as a means of introducing game elements in the process of teaching mathematics, EDULEARN19 Proceedings. 2019. P. 1189–1194.
13. *Shirokova O., Gainutdinova T., Denisova M.* Interdisciplinary integration as a condition for the modernization of teacher education, EDULEARN19 Proceedings. 2019. P. 1183–1188.

СВЕДЕНИЯ ОБ АВТОРЕ



ШИРОКОВА Ольга Александровна – доцент, кафедра высшей математики и математического моделирования, Казанский федеральный университет, Казань.

Olga Aleksandrovna SHIROKOVA – Associate Professor of the Department of Higher Mathematics and Mathematical Modeling, Kazan Federal University, Kazan.

Email: shirokova2602@mail.ru

ORCID: 0000-0002-2883-0415

Материал поступил в редакцию 15 марта 2021 года