

**БИРИНЧИ ДАРАЖАДАГЫ СЫЗЫКТУУ ДИФФЕРЕНЦИАЛДЫК
ТЕҢДЕМЕЛЕРИ ЧЫГАРУУ УЧУН ИНТЕРАКТИВДҮҮ БИР ЖОЛ
AN INTERACTIVE WAY TO SOLVE A FIRST-ORDER LINEAR
DIFFERENTIAL EQUATIONS**

MUSLU I.

**University of International Ataturk-Alatoo, The Department of Applied
Mathematics & Computers
izvestiya@ktu.aknet.kg**

Maplets are Java applets written with Maple that keep all the algebraic computational power of Maple. A Maplet consists of one or more windows which interact with the user by means of buttons, checkboxes, text fields, and other standard graphical controls. Most of the Maplets available with Maple either display something or compute something avoiding the standard Maple syntax. Our Maplet for First-Order Linear Differential Equations (FOLDE) is more pedagogical that they ask a student to do something, provide some hints and guidance along the way, and inform the student if their responses are correct or incorrect. The Maplets try to improve the students' intuitive and computational abilities while covering graphical, numerical, and analytical aspects. In this paper we represent a Maplet of FOLDE (First-Order Linear Differential Equations)

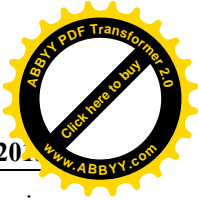
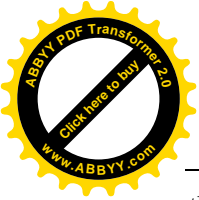
Бул макалада Маплет программасын биринчи даражадагы сызыктуу дифференциалдык теңдемелерди чыгаруу жана алардын чыгарылыштарын изилдөөдө колдонуу методикасы берилген. Маплет программасы колдонуучу менен өз ара байланышта жашайт. Колдонуучуга каралган маселесине карата керектиги маалыматтарды киргизүү үчүн суроо берет, колдонулган метод жөнүндө маалымат берет, маселенин туура же туура эмес чыгарылышы жөнүндө билтирет. Маплеттер маселенин чыгарылышын графикалык, сандык жана аналитикалык тирде көрсөтөт. Бул окуучулардын тишиңиңиңиң жана эсептөөчүлүк жөндөмдөрүн жакшыртууда зор салымын көрсөтөт.

Introduction

Maplets enable users access to the parts of Maple relevant to their needs while shielding them from any problems associated with Maple syntax.

What the student sees in using Maplets: Maplets and Java Swing Classes

Maplets use calls to the Java Swing Classes to handle Graphic User Interface (GUI) details. The screen image of the First-Order Linear Differential Equations example provided is typical enough. The elements from the Swing Classes used include dialog boxes (TextBox), and buttons. Items not illustrated



there are pull-down menus, checkboxes, etc. (One can imagine pedagogic variations of the example in which these might be used. For example, in teaching, it is often appropriate to ask students to do tasks, and here it might be appropriate to ask them to select from a list of intervals provided via an appropriate element where the function is increasing, etc. Maplets and the Swing Classes cope with this easily.) Many other GUI elements are illustrated in the Maplet examples publicly available at themapleapps.com repository. Maplets are launched from Maple. Once Maple knows the definition of myMaplet (say), the Maple command for running it is: `Maplets [Display](myMaplet)`: One could hide the Maple - both that defining myMaplet and also the callout to it - in a script or, on Windows, a .bat file. The script would be one which started Maple, read in Maple code which in turn launched the Maplet. (For better methods that will come with Maple 13, see the section, "How Maplets work".)

The instructor: authoring Maplets

The writing of code for complex code is a fairly detailed business. The layout, etc. has to be specified. The Maplets package for Maple 13 enables the writing of the GUI code to be done from Maple. In writing a Maplet, the first command to Maple is `with(Maplets[Elements]);`

In my view, the easiest way to begin writing Maplets is to start with the examples, and modify them. I had no difficulty writing simple Maplets. The Maplets package makes intelligent decisions on layout, and for a fairly simple GUI, writing a Maplet is easy; it is also capable of writing more complex and sophisticated GUIs.

How Maplets work

The following is a brief, description of the Maplets architecture based on that provided by Jason Schattman of Waterloo Maple. The user (student user, or staff author) interacts with the Maplets package within Maple. The `Maplets[Elements]` part of the package generates an XML description of a Maplet. Using `Maplets[Display]`, this description is sent to the Maplets runtime engine, which is written in Java. This runtime engine interprets the XML description of the Maplet, dynamically creates the necessary user interface (Java Swing) components and displays the Maplet. There is no standalone Java code associated to a specific Maplet. Maplets are XML descriptions that are taken as input by a generic runtime engine. We now quote verbatim from the Waterloo Maple source: "That said, in Maple 13, users will be able to save a Maplet to a standalone file, which can be shared with other users. Using "Save as ... Maplet" stores all the information necessary to reconstruct a Maplet (defined in a worksheet) to a file. Executing (double-clicking) this file on any machine that has a Maple 13 installation launches the Maplet."

Pedagogic purposes?

As a small incremental change?

One view of software is that it can come in two main sorts - "appliance software" as one sort, "environments" as another. "Appliance software" does just one task. Appliance software is easy to use. However, if different appliances come in too many different styles, it can be confusing. Maplets, by using Java Swing classes, should feel familiar to our students. (This is not to say that one cannot succeed in writing a confusing Maplet if one set out to do so.) "Computing environments" are, for us, big

packages like Maple, Mathematica or Matlab or similar. Such software which can do lots of different tasks. The obvious problem with environments is that they can become bloated and hence difficult to maintain as reliable software. Maple has correctly avoided too much in-package word processing (as LaTeX does the job) and also too much in-package float numerics (as Matlab/Octave) does the job.

Maple correctly aims at extending its functionality by intelligent connection or inter-relation with other software. Adding to its functionality the production of (Java-Swing-based) Maplets seems a reasonable increment. In the near future, perhaps most 1st year students in maths departments where Maple is the dominant package in teaching will use Maplets (or similar) rather than Maple. There should then be a smooth transition at 2nd year to students using pre-prepared Worksheets more than Maplets, with some Maplets around both for those who really don't want to program and to achieve continuity with the first year experience. At 3rd year it seems reasonable to expect all maths students to be able to program in Maple (or similar) to solve appropriate tasks of their own.

Pedagogic purposes for Maplets associated with larger-scale teaching software?

Many Calculus, etc. CDs have been written. Some even have Maple as an algebra engine underpinning them. I imagine that Maplets could provide an attractive way of getting a consistent feel to pedagogic materials of this kind. Perhaps if the Mathwise project were to have started now rather than a decade earlier, Maplets would have been seriously considered to underpin it. In reference [3] Douglas Quinney describes "Calculus Machina". This builds on Java, and is underpinned by its own Computer Algebra system. Once again, a reasonable fraction of the work described there could be implemented effectively in Maplets if one were to choose to do so.

The future? Java, Computer Algebra and the Web

I know of at least three Java-Maple combinations, all Web-server related. The first is AIM. AIM is reviewed in [4]. AIM uses Java servlets. The next two use JSP pages and

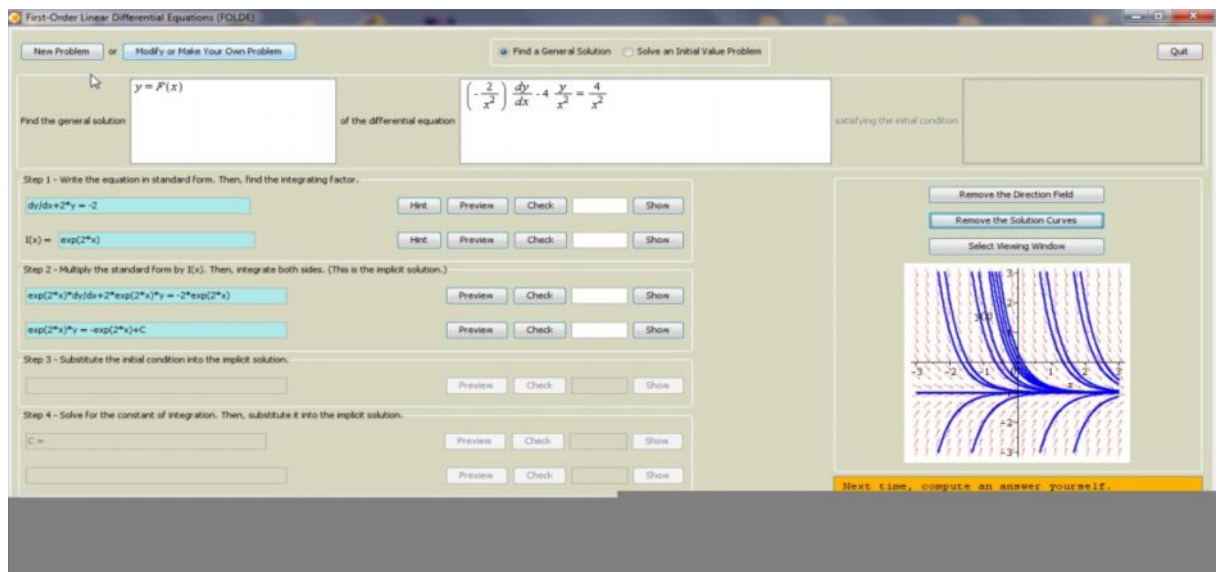


Fig. 1. General view of the FOLDE Maplet, Solution steps, Direction field and Solution Curves

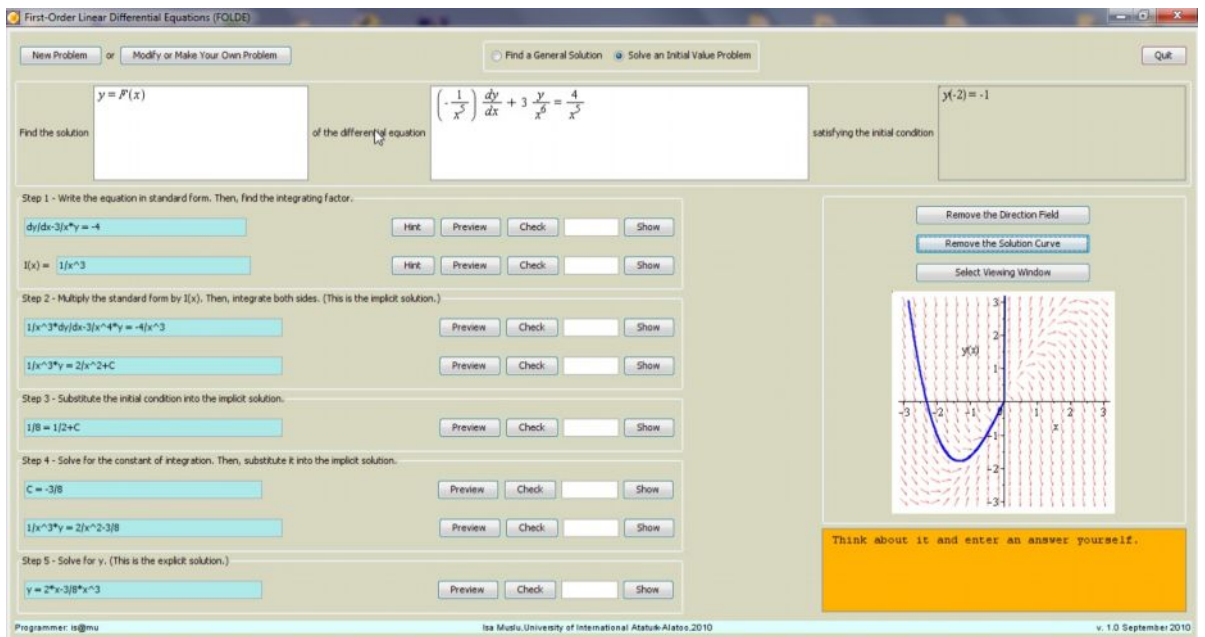


Fig.2. Modify or make your own Problem

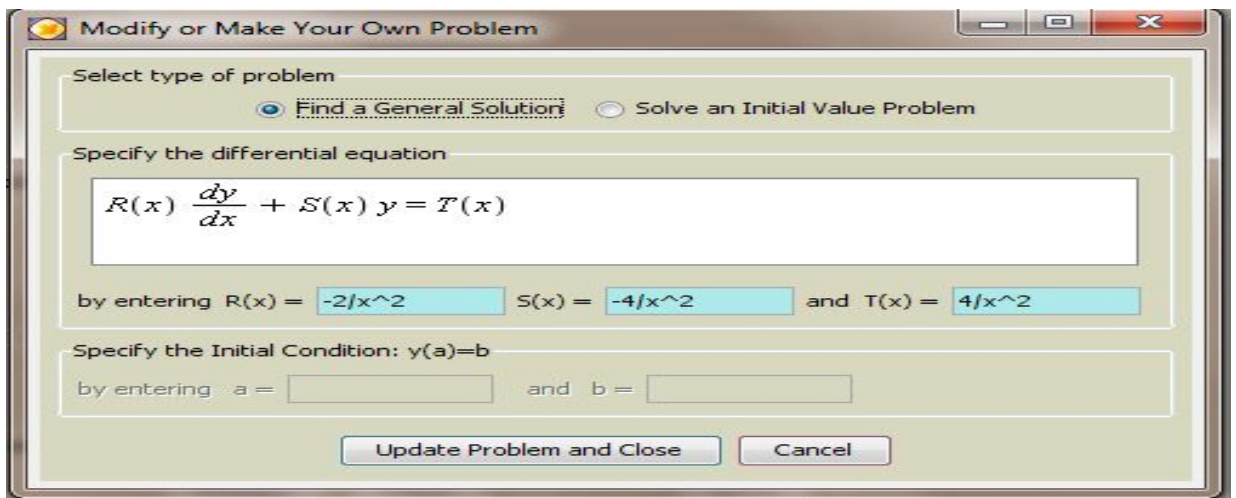
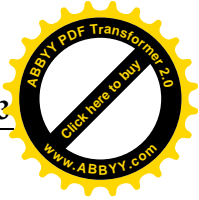
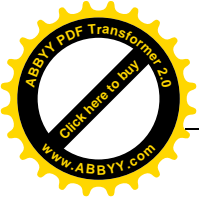


Fig.3. Solution steps of initial-value problem, Graphics of directed field and Solution Curves

Conclusion

In the presented paper an interactive way, FOLDE Maplet, of a first-order differential equations is given. This way is very pedagogic because here students or users could control if their solution steps are correct or not. They can check the steps if correct or not. If not they can see the hint about the step even if they still can't find then they can see the results. After solving the problem it is possible to plot directed field and some solution curves of the problem.



References

1. G Keady and D Hill, Maths&Stats, Nov 1997
2. T Kolokolnikov, Maple Forum, Feb 2002:
http://www.scg.uwaterloo.ca/~maple_gr/Digests/Digest00.02
3. D Quinney, <http://itsn.mathstore.ac.uk/articles/maths-caeries/nov2001/index.htm>
4. N Strickland, MSOR Connections, Feb 2002
5. Grant Keady, A review of Maplet, University of Western Australia
6. Maplets, A customizable Interface to Maple, Stephan Forrest, 2005