Animations in *PDF*

Petr Sojka
Masaryk University in Brno, Faculty of Informatics
Botanická 68a, 602 00 Brno, Czech Republic

sojka@fi.muni.cz

ABSTRACT

This paper describes a technique to create interactive teaching materials as animations that are stored and distributed in PDF file format. PdfLaTeX with small macropackage, Maple and Javascript are used and allow the development of interactive animations of high typographical quality that are fine-tuned for on-the-screen reading.

Categories and Subject Descriptors

J.1 [**Administrative Data Processing**]: Education; H.5.1 [**Multimedia Information Systems**]: Animations; J.7 [**Computers in Other Systems**]: Publishing; H.5.2 [**User Interfaces**]: Interaction styles—*forms*

General Terms

Documentation, Design, Human Factors

Keywords

interactive animations, electronic CS teaching materials, PDF, TeX, LaTeX, JavaScript, PostScript, Maple, Acrobat

1. INTRODUCTION

Portable Document Format (PDF) [1] is mostly considered only as a static hypertext alternative to widely used PostScript for the portable delivery of electronic documents (teaching materials, homework exercises, etc). It is not widely known that in PDF file one can embed animations and other programmable *interactive* content using JavaScript scripting language, as there is a built-in JavaScript engine in Acrobat Reader and other PDF-compliant viewers.

There is not one best and only way to produce teaching materials. They should differ for distance (E)-learning and standard courses. One way of enriching the diversity of exposition of topics taught is to take advantage of new portable technologies and formats and provide teaching materials *designed* and fine-tuned for *multiple modes of delivery*. *Interactive animations* designed for electronic delivery and reading on computer screen is one possibility, in our experience applauded and endorsed by the students. Interactivity allows for autotesting and similar means of student evaluation.

2. PREPARATION AND TYPESETTING OF ANIMATIONS

We have developed the technique of embedding interactive animations created by Maple into PDF, using pdfTeX program. With this technique we have created electronic teaching materials [2] for the course on *Multi-variable Calculus using Maple* for students of computer science. Lecture notes containing animations and their explanations are created by the teacher in the standard LaTeX notation using images generated by Maple and exported as a series of JPEG images. As compared to a solution using HTML/XML with CGI scripts and animated GIF, PDF allows full control of a high typographical quality of materials.

Even though animations created in PDF documents has been considered to be something of the "Holy Grail" of the PDF world, our method of creating hypertext teaching materials with highly portable animations for visualization is rather straightforward. For embedding JavaScript into PDF an approach similar to [3] is used, exploiting Acrobat forms [1] and Acrobat. For layout and typesetting a small macro-package has been written so that creation of animations is automated as much as possible. For Maple license owners, an elegant but less portable solution is possible too: by clicking in the PDF file, animations with parameters chosen can be created by calling Maple on the fly.

Animations can be viewed and studied step by step, stepped forward and backward, viewed at different speeds, zoomed, etc. The only software needed by a student to enjoy the materials is freely available Adobe Acrobat Reader or Ghostview.

Examples of animations can be downloaded from [2] as part of teaching materials (in Czech), or from http://www.fi.muni.cz/usr/sojka/animations/.

3. ACKNOWLEDGEMENTS

4. REFERENCES

[1] Adobe Systems Incorporated. *PDF reference: Adobe portable document format version 1.4*. Addison-Wesley, Reading, MA, USA, third edition, Dec. 2001.

[2] Z. Došlá, R. Plch, and P. Sojka. Mathematical Analysis with Maple: 2. Infinite Series (in Czech). CD-ROM, http://www.math.muni.cz/~plch/nkpm/, Dec. 2002.

[3] D. P. Story. Techniques of introducing document-level JavaScript into a PDF file from LaTeX source. *TUGboat*, 22(3):161–167, Sept. 2001.