REPORT ON OpenDreamKit DELIVERABLE D2.9

Demonstrator: interactive books on Linear Algebra and Nonlinear Processes in Biology

MARCIN KOSTUR, JERZY ŁUCZKA, JAN AKSAMIT, JOLANTA MARZEC

Due on	31/08/2018 (M36)
Delivered on	31/08/2018
Lead	University of Silesia (USlaski)
Progress on and finalization of this deliverable has been tracked publicly at: `https://github.com/OpenDreamKit/OpenDreamKit/issues/49`	

CONTENTS

1. INTRODUCTION

Interactive web pages have always been an attractive tool in education. It has started in the age of Adobe-Flash technology, when a lot of educational material was created. The advent of JavaScript has led to significant improvements in that field: in the first place it created a possibility to author the content using Open Source tools and secondly it increased the portability of the created documents to practically all devices equipped with a modern web browser.

The problem which was persistent in those solutions was the big gap between the authoring process of interactive documents and their usage. Usually the code behind an interactive examples was not very educational and was not supposed to be edited by the student.

Then emerged the concept of interactive widgets in the notebook, appearing first in Mathematica, then being popularized by the SAGE notebook, and recently getting huge traction with the JUPYTER notebook thanks to its wide audience. Interactive widgets, and in particular the very simple to use `@interact` decorator, delivered a versatile solution to this problem: they made it possible to turn with little efforts a mere scientific calculation into a visually appealing interactive application, all within the original environment where the mathematical explorations took place.

The need of including computer technology for teaching at University of Silesia was sparked by interdisciplinary courses. In those fields it is common to master numerical tools for modeling and at the same time to gain an intuition. Integration of modern computer technology allows students to more efficiently analyse a system without tedious symbolic calculations. This has led to systematic integration of SAGE system with science education at the Institute of Physics since 2011. In this deliverable we have created texbooks which consists of c.a. 500 pages with over hundred interactive code cells and code examples and tens of figures. The audience for those books are students of physics, biophysics, econophysics and medical physics. This work took 25 PM of engagement of four people.

The source code of the interactive books is hosted on github within a dedicated repository of the OpenDreamKit organization: `https://github.com/OpenDreamKit/iODKbook2`.

2. Interactive book technology

We have used SPHINX as main authoring tool: SPHINX is originally a documentation generator which is popular in the Python world where it's used for the documentation of PYTHON itself, and many other projects. More generally it can be used to author large structured documents which are then exported to various formats, including HTML, PDF, EPub, and now even Jupyter notebooks. SPHINX contains a plugin system enabling to taylor it for particular applications. We used plugins for

- embedding SAGE code as live cells which can be edited, executed and interacted with (see Figure 1); under the hood the computations are run on a remote service thanks to the SageMathCell technology (`https://github.com/sagemath/sphinx-sagecell-ext`); an alternative would be to use Thebelab (see D4.7: "Full featured JUPYTER interface for GAP, PARI/GP, Singular");
- using conditional compilation of parts depending for example on whether the target medium is interactive of not, or whether solutions should be included,
- automatically including JUPYTER notebooks with embedded output;
- producing high quality pdf via LaTeXsystem.

In the authoring process one can join interactive material in the form of notebooks (based on sagenb or JUPYTER) that can be embedded in full or in part into the SPHINX book. In the HTML export, the material is then rendered as live SageCellServer code that the user can edit, execute, or interact with. It has to be stressed that not only interactive widgets written with `@interact` are useful. Practically all of exercises inside the book are illustrated by such interactive material. It's our experience however that such interactive material should not require or encourage the reader to do anything but small edits to the provided code; indeed such edits are volatile by essence; they take place anonymously in the browser with no storage behind the scene.

The authoring process is very similar to writing a LaTeXpaper. It is based on edit-compile-view cycle which is known to most of physicists and mathematicians. In some cases we have used a shared account on a JUPYTER system for this process, where compilation commands where pre-coded in a JUPYTER notebook, which also included a link to web, and pdf versions. In such case is was very easy to help academics less familar to computer science to just edit and modify the content.

3. Nonlinear Processes in Biology

The book is in large extend based on coursework taught at University of Silesia. It was primarily based on sagenb system and gradually was converted into the interactive book with live examples.

The book covers the classical topics in modeling in biology. Methodologically it can be split into the following topics:

(1) Methods of mathematical modelling of processes based on ODE and PDE.
(2) Qualitative methods in ODE and PDE with application of Sage. It includes:
    - stationary states of the system,
    - the linear stability of fix points,
    - phase curves,
    - bifurcation diagrams,
    - time-dependent solutions,

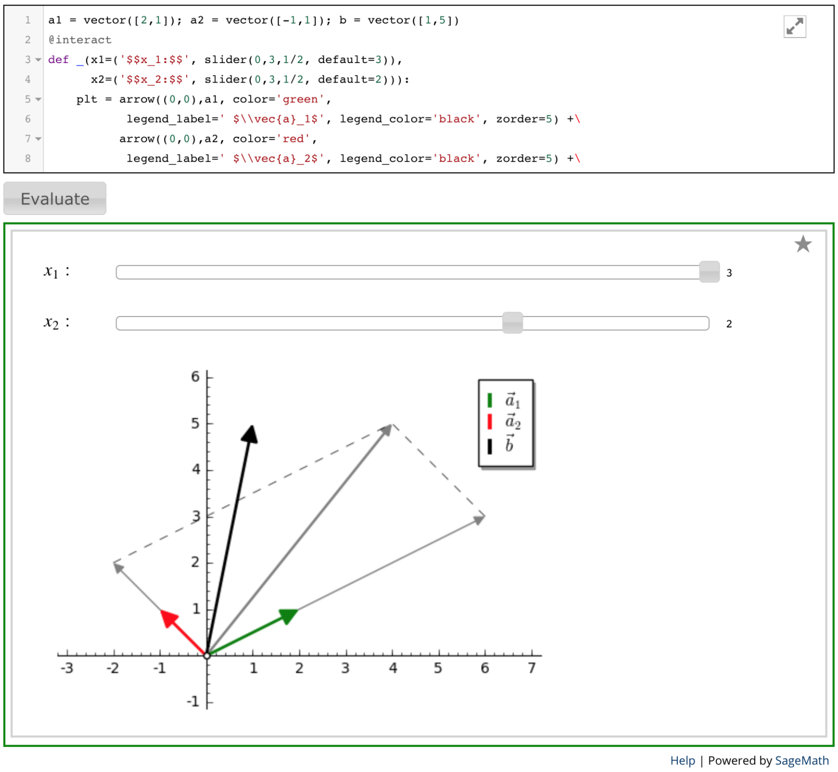

FIGURE 1. An example of `@interact` in SPHINX generated web version of interactive book. The upper box shows the code and the lower box shows the output.

- a graphical presentation of all above elements.

  For each subject there is a list of related problems and supplementary tasks as homework. All requires symbolic manipulations as well as certain techniques from numerical analysis like root finding.

(3) Numerical solving of ODEs. It is complementary to analytical and qualitative methods for ODE.

(4) Numerical solving of PDE-s - diffusion equation, reaction-diffusion systems and similar. It requires good tools in data processing, usually uses numpy for calculations.

It turned out that conceptually simple methods - like plotting a function of one variable, which depends on a parameter (see Figure. 2), can be very useful for the analysis of models containing analytically untractable expressions. In this particular example, students construct just a simple plot of a function; then, with the help of `@interact` construction, the plot is easily turned intro an applet for the analysis of the system.

The book covers the following models:

- one-dimensional systems (Malthus, Verhulst, Ludwig, Alee models);
- two-dimensional systems (Lotka-Volterra, May models);
- kinetics of chemical reactions;
- Belousov-Zhabotinsky reactions;
- models of epidemics.

4. LECTURES ON LINEAR ALGEBRA

The Linear Algebra book is dedicated to Dr Jan Aksamit for his devotion, and endless patience in work with students. He has lectured linear algebra for physics students for almost 40 years of

```
In [34]: @interact
def _(k=slider(round(0.001,3),20,0.1, 12.0),r=slider(round(0.001,4),1,0.05,.4)):
    p1 = plot(x/(x^2+1),(x,0,15),legend_label=r'$y=\frac{x}{x^2+1}$')
    p2 = plot(-r/k*x+r,(x,0,15),color='red',legend_label=r'$y=r(1-\frac{x}{k})$')
    p3 = plot( -(1/6*(3*k*x^2 - 2*x^3)*r/k - x + arctan(x)),(x,0,11),\
          legend_label=r'$y=-\frac{(3kx^2-2x^3)}{6k}-x+ \arctan(x)$')
    show(p3,figsize=(6,2),axes_labels=[r'$x$',r'$y$'],tick_formatter='latex')
```

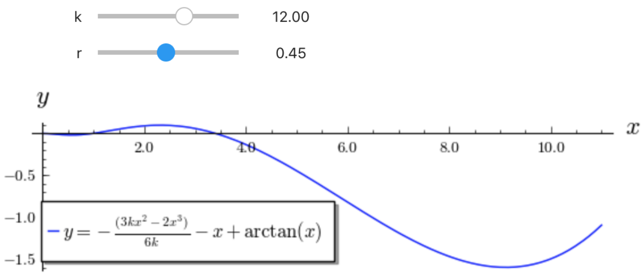

FIGURE 2. An example of `@interact` applied for graphical examination of roots of a function. It is a first step in qualitative analysis of model of tumor growth in a book.

his life. With restless passion he spent the last years of his life preparing an electronic version of his lectures notes.

This book is based on Linear Algebra courses for physics students taught in recent years at the University of Silesia. It has been gradually equipped with interactive materials and extended by sections illustrating application of linear algebra in economy, computer science, and other areas. The computations and the interactive part are performed by SAGE, which is embedded in the book.

Along the theory the user becomes familiar with the code used in SAGE. Their learning process is enhanced by various factors:

- numerous visualisations;
- interactive images which allows for the construction of a graphical solution to a problem; the problem may be changed in real time by the user changing the code;
- a place for experiments: the user may fill in the gaps in the code to produce graphical interpretation of a chosen operation;
- possibility to perform the calculations with SAGE within the book; in particular, computations with real data and, thus, real life conclusions are possible;
- some longer coding exercises are equipped with check points (without revealing the answer) to ensure that the discovered solution is correct.

Moreover, not only users benefit from interactive resources, but also learn how to produce them themselves.

5. FUTURE WORK

The two open textbooks are put to continuous use for the eponymous courses at university of Silesia; they are frequently updated with new material (new models, exercises, ...), and improved according to feedback from teachers and students. Also there are effort being done to encourage other lecturers to use them and contribute back.

An important improvement will be inclusion of automatic testing in the compilation procedure.

Currently the material exist in Polish and English versions, which are, however unsynchronized. In future we plan to update the English version only, mostly because of growing internationallity of studies. In some cases, mostly for theoretical parts of linear algebra book, the polish version will be kept in synchronisation.

Appendix A. Screenshots

Contents

i

FIGURE 3. Table of contents of Linear Algebra book.

ii

FIGURE 4. Table of contents of Linear Algebra book.

iii

FIGURE 5. Table of contents of Linear Algebra book.

Column Picture

The foregoing system of equations

$$\begin{aligned} 2x_1 - x_2 &= 1 \\ x_1 + x_2 &= 5 \end{aligned}$$

can be rewritten as equality of two column vectors:

$$\begin{bmatrix} 2x_1 - x_2 \\ x_1 + x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}.$$

The left-hand side can be expressed as a linear combination with coefficients x_1, x_2 :

$$x_1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + x_2 \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}.$$

Now the unknowns x_1, x_2 are weights for column vectors $\boldsymbol{a}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $\boldsymbol{a}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ in a linear combination, which should equal the column vector of constants $\boldsymbol{b} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$.

Experiment with Sage:

In the following program the column vectors $\boldsymbol{a}_1, \boldsymbol{a}_2$ and $\boldsymbol{b}$ are represented by the geometric vectors $\vec{a}_1, \vec{a}_2$ and $\vec{b}$, respectively. Using the sliders, set the values of coefficients x_1 and x_2 so that the vector $x_1\,\vec{a}_1 + x_2\,\vec{a}_2$ (coloured grey) is equal to the vector $\vec{b}$.

```
a1 = vector([2,1]); a2 = vector([-1,1]); b = vector([1,5])

@interact

def _(x1=('$$x_1:$$', slider(0,3,1/2, default=3)),
      x2=('$$x_2:$$', slider(0,3,1/2, default=2))):

    plt = arrow((0,0),a1, color='green',
```

Evaluate

x_1 : 3

x_2 : 2

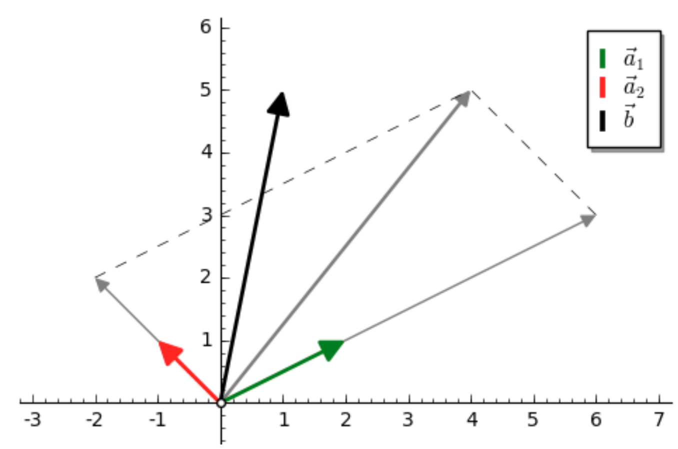

FIGURE 6. An example of interactive element in web version of Linear Algebra book.

```
Column Picture
~~~~~~~~~~~~~~

The foregoing system of equations

.. math::
   :nowrap:

   \begin{alignat*}{3}
   2\,x_1 & {\,} - {\,} & x_2 & {\;} = {\;} & 1 \\
   x_1 & {\,} + {\,}& x_2 & {\;} = {\;} & 5
   \end{alignat*}

can be rewritten as equality of two column vectors:

.. math::

   \left[\begin{array}{r} 2\,x_1\ -\ x_2 \\ x_1\ +\ x_2 \end{array}\right]
   \ =\
   \left[\begin{array}{c} 1 \\ 5 \end{array}\right]\,.

The left-hand side can be expressed as a linear combination
with coefficients :math:`\,x_1,\,x_2\,:`

.. math::

   x_1\ \left[\begin{array}{r} 2 \\ 1 \end{array}\right] \ + \
   x_2\ \left[\begin{array}{r} -1 \\ 1 \end{array}\right] \ = \
   \left[\begin{array}{r} 1 \\ 5 \end{array}\right]\,.

Now the unknowns :math:`\,x_1,\,x_2\ ` are weights for column vectors
:math:`\ \boldsymbol{a}_1 =
\left[\begin{array}{r} 2 \\ 1 \end{array}\right],\ `
:math:`\boldsymbol{a}_2 =
\left[\begin{array}{r} -1 \\ 1 \end{array}\right]\ ` :math:`\\` in a linear
combination, :math:`\,` which should equal the column vector of constants
:math:`\ \boldsymbol{b}\,=
\left[ \begin{array}{r} 1 \\5 \end{array} \right].`

|
.. admonition:: Experiment with Sage:

   In the following program the column vectors
   :math:`\;\boldsymbol{a}_1,\ \boldsymbol{a}_2\ `
   and :math:`\;\boldsymbol{b}\ ` are represented by the geometric vectors
   :math:`\;\vec{a}_1,\ \vec{a}_2\ ` and :math:`\;\vec{b},\ ` respectively.
   Using the sliders, set the values of coefficients
   :math:`\ x_1\ ` and :math:`\ x_2\ ` so that the vector
   :math:`\;x_1\ \vec{a}_1 + x_2\ \vec{a}_2\;` (coloured grey)
   is equal to the vector :math:`\,\vec{b}`.

.. sagecellserver::

   a1 = vector([2,1]); a2 = vector([-1,1]); b = vector([1,5])

   @interact

   def _(x1=('$$x_1:$$', slider(0,3,1/2, default=3)),
         x2=('$$x_2:$$', slider(0,3,1/2, default=2))):

       plt = arrow((0,0),a1, color='green',
              legend_label=' $\\vec{a}_1$', legend_color='black', zorder=5) +\
             arrow((0,0),a2, color='red',
              legend_label=' $\\vec{a}_2$', legend_color='black', zorder=5) +\
             arrow((0,0),b, color='black',
              legend_label=' $\\vec{b}$', legend_color='black', zorder=5) +\
             arrow((0,0),x1*a1, color='gray', width=1, arrowsize=3) +\
             arrow((0,0),x2*a2, color='gray', width=1, arrowsize=3) +\
             arrow((0,0),x1*a1+x2*a2, color='gray',
              width=1.75, arrowsize=3) +\
             line([x1*a1,x2*a2+x1*a1], color='black',
              linestyle='dashed', thickness=0.5) +\
             line([x2*a2,x2*a2+x1*a1], color='black',
              linestyle='dashed', thickness=0.5) +\
             point((0,0), color='white', faceted=True, size=18, zorder=7)
```

FIGURE 7. The source code of an example of interactive element in web version of Linear Algebra book.