
L^AT_EX training through spoken tutorials

Kannan M. Moudgalya

Abstract

Spoken Tutorials, a combination of screencast and voice over, are meant for self-learning of free and open source software (FOSS) systems. The pedagogy involved in creating spoken tutorials for L^AT_EX is explained. A checklist, instructions for conductors and activity-based instructions for learners, along with the spoken tutorial are what is needed to conduct the two hour SELF FOSS Study Workshops. As one finds out during the workshop how to learn from these tutorials, one can complete the learning at home if two hours are insufficient.

This method of learning has been shown to be effective. As it allows the conducting of these workshops without the domain experts, this methodology is scalable: we expect to conduct 500 workshops in a period of six months. The students are trained free of cost. The honorarium expenses for the conductors of these workshops work out to Rs. 25 per student per software package.

1 Introduction

A spoken tutorial is an audio-video tutorial that explains an activity performed on the computer. An expert explains the working of the software by demonstrating it on the screen, along with a running commentary. Screencast software makes a movie of the entire activity, both the screen and the spoken part. This movie is the spoken tutorial. It is of ten minutes duration. One can reproduce the commands shown in the tutorial side by side and thus use it as an effective instructional tool.

We have been using this methodology to create a series of tutorials in open source software families, such as L^AT_EX, Scilab, GNU/Linux, ORCA, Python, LibreOffice and PHP/MySQL. We have selected the duration of a typical spoken tutorial to be about ten minutes long. Although only a small topic can be covered in ten minutes, by stringing them together, one can come up with study plans that are capable of teaching advanced topics as well.

Our approach involves the creation of a script before creating the video. It is possible to translate the script into other languages and use them for dubbing, while screen shots continue to be in English. For example, see a tutorial with Tamil audio at [1]. This will help those who are weak in English, while not compromising on the employability.

Spoken tutorials can also be used to bridge the digital divide: topics such as buying train tickets on-

line, locating low cost agricultural loans, and locating information on first aid and primary health care can be covered. One target audience for a spoken tutorial is a remote child, working alone at midnight without anyone to help her.

As the spoken tutorials are created for self-learning, it is possible to conduct workshops even without domain experts. This allows scaling up workshops using additional instructional methodologies.

The above mentioned points have been explained in [4]. The motivation behind this effort is available at [6]. An early work in this area is [3].

The initial part of this article focuses on creation of spoken tutorials on L^AT_EX and Xfig. The rest of this paper is devoted to the methodology of conducting workshops using these tutorials.

2 Creation of L^AT_EX Spoken Tutorials

We created the first few Spoken Tutorials in early 2007 to teach L^AT_EX to our students. We had just completed a textbook [2]. Had we not used L^AT_EX, it would have taken several more years to complete it. We wanted our students also to benefit from L^AT_EX. We also wanted to contribute something back to the community. We created the following tutorials:

1. What is compilation (9:17)
2. Letter writing (8:19)
3. Report writing (16:14)
4. Mathematical typesetting (24:46)
5. Equations (23:42)
6. Tables and figures (25:12)
7. Bibliographies (8:20)
8. Inside story of bibliographies (24:44)
9. L^AT_EX on Windows (27:20)
10. Updating MiK_TE_X on Windows (15:31)
11. Beamer (34:00)

The numbers inside the brackets indicate the time duration in minutes and seconds. Although we wanted the tutorials to be 10 minutes long, some of them turned out to be longer. In those days, we did not have a method to estimate the length of a spoken tutorial before it was made. The original tutorial on bibliography was longer than we wanted, so we renamed it as the “inside story” and re-did a shorter one lasting only 8:20.

In order to make L^AT_EX available to Windows users, we made tutorials 9 and 10 in the above list. As our group had not used L^AT_EX on Windows, we had to learn it first before making a tutorial. Finally, we created the spoken tutorial on Beamer. We now insist on L^AT_EX-created slides to accompany all new spoken tutorials.

Block diagrams are an important requirement for any scientific writing. For this purpose, we cre-

ated: introduction to Xfig (12:38), feedback diagram in Xfig (12:02) and mathematics in Xfig (15:20). The last tutorial explains a procedure to embed mathematical formulae in figures, using Xfig and L^AT_EX.

There are a few reasons for not using a special-purpose L^AT_EX editor for Unix systems:

1. We wanted to teach L^AT_EX in the most pristine form, without getting bogged down by the editor's details.
2. There is a good amount of effort involved in learning any special-purpose L^AT_EX editor.
3. As we don't use one, it would have taken a considerable amount of our time and effort to explain the working of a specialised editor.
4. As Unix users are reasonably comfortable with the terminal, our approach would not create much difficulty.

The situation is quite different for a Windows user, who is generally not comfortable in working with the terminal. As most of our audience use Windows, we did not mind putting in some effort in learning a specialised L^AT_EX editor for Windows and creating tutorials 9 and 10 above.

All the spoken tutorials on L^AT_EX explain a three step process: 1) Creating a source file with an editor; 2) Compiling the source with `pdflatex`; 3) Viewing the resulting PDF file with a PDF browser.

The recorded area was divided to simultaneously show the editor, terminal and the PDF browser. These three were arranged in a non-overlapping manner in the early tutorials, such as letter writing, mathematics and equations. In the letter writing tutorial, a given source file was explained along with an illustration of the effects of changes in the commands. In the tutorial on Beamer presentations, however, additional commands and text were copied from another file and the effects demonstrated. A screen shot of this is shown in Fig. 1.

Showing three or four non-overlapping windows made the fonts small. This was addressed in the Xfig tutorial, where the overlapping requirement was done away with, so as to achieve large font sizes. A screen shot of this is shown in Fig. 2. In all of the above, only a small portion of the Desktop was recorded to keep the recording size small.

All the tutorials mentioned above are available at <http://spoken-tutorial.org>, along with all the required source, style, `bst` and PDF files.

Spoken tutorial is based on demonstrations. We want a learner to reproduce whatever is shown in the tutorial. As it is *active learning*, it is effective. To ensure that this happens, we insist that 75% of every tutorial is devoted to demonstrations and not

more than 25% is concerned with theory. These limits generally work well in all except perhaps the first tutorial in a series. To solve this problem, we recommend that the introductory tutorial be created after all other tutorials in the series are completed. Armed with these tutorials, one can also make the introductory tutorial consist of 75% demonstrations by playing these tutorials. This strategy has been followed in creating [6].

3 Workshops using Spoken Tutorials

Spoken tutorials are created for self-learning. These are also available for free download. Although one can learn from these tutorials without any external assistance, organised workshops are the most effective way if we want to train a large number of people in a short time.

Ours differ from conventional workshops in many ways: instructional material, conduct of the workshop, time duration of the workshop and the conductor of the workshop, to name a few. We will explain these in detail now.

3.1 Instructional material

In order to conduct the workshops one needs the spoken tutorials and the associated files, as mentioned earlier. One can download these from our website free of cost. If required, we also provide them on CDs, once again free. In addition to these, we provide the following three documents:

A checklist that has to be completed the day before the workshop is conducted. This forces the organiser of the workshop to verify for every PC (a) that the spoken tutorials and resource files are copied; (b) that the software to be taught in the workshop is installed correctly; (c) whether the headphones are working; (d) whether the spoken tutorial can be seen and heard through the headphones. In case of online tests, the organiser should also check the working of the Internet. A copy of such a list is given in Fig. 3.

The second document is the instruction sheet for the conductor of the workshop. This important activity is covered in detail in the next section.

The rest of this section is devoted to the third document, which contains instructions to the learners. This instruction sheet is meant to be used with the spoken tutorials. We have reproduced the initial part of the instruction sheet meant for learning L^AT_EX on GNU/Linux system in Fig. 5.

These instruction sheets are for self-learners. In view of this, the initial part of the instructions are activity based, or equivalently contain verbs. For instance, the first few instructions have the following

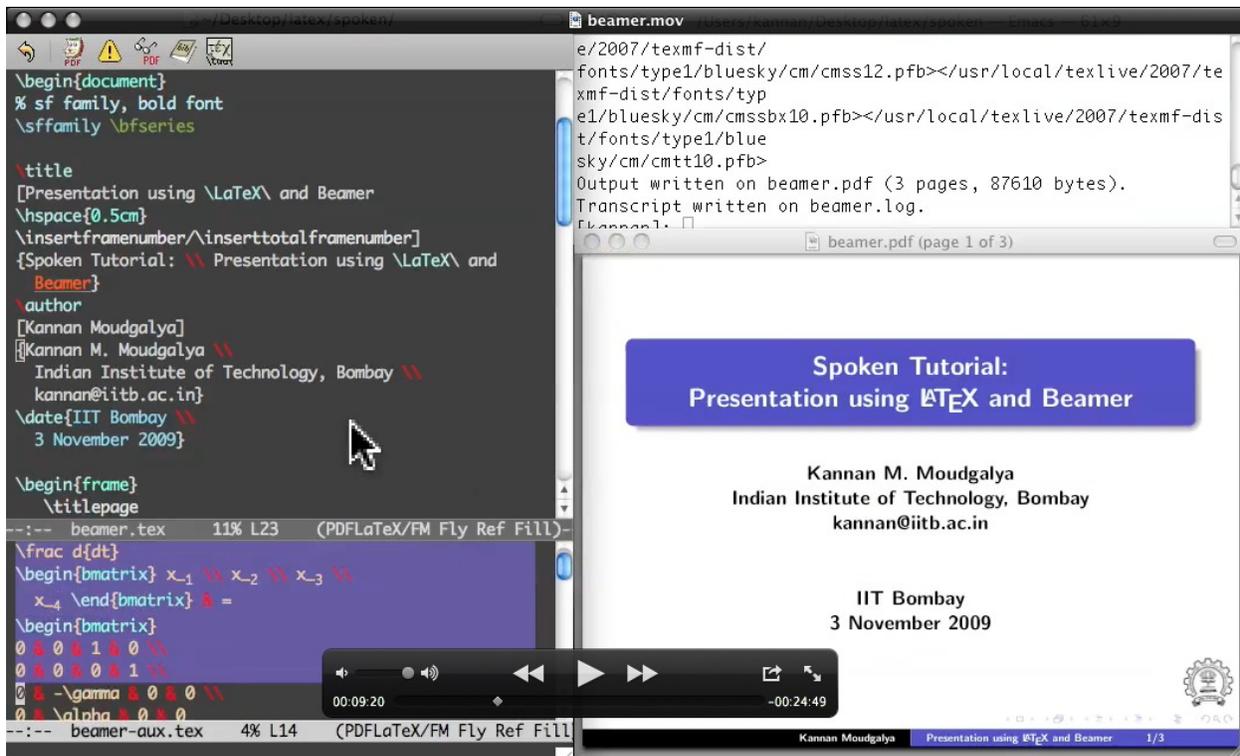


Figure 1: Two files, terminal, PDF browser shown in non-overlapping fashion in the Beamer Spoken Tutorial [5]

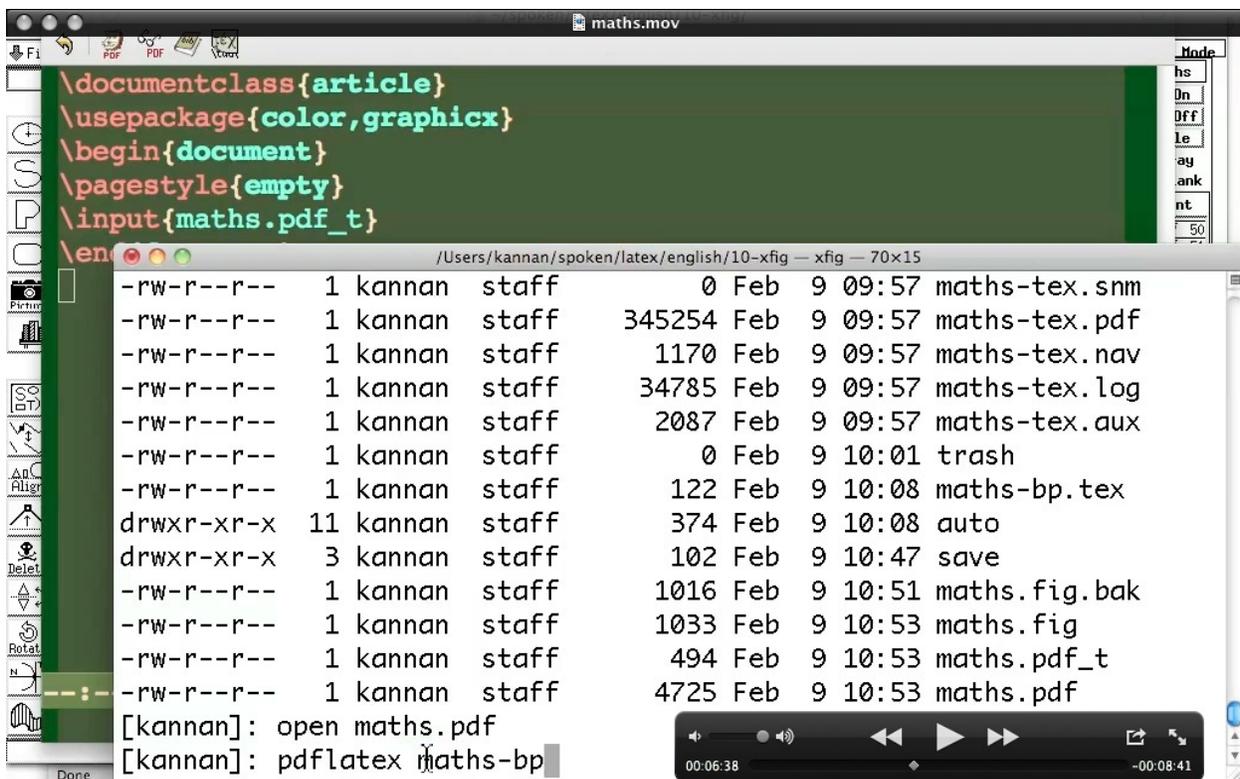


Figure 2: Overlapping of Emacs editor, terminal, and Xfig allows the use of large fonts [7]

PC No.	Is the PC booting?	Can log into PC? (if applicable)	These can be separated in 2 PCs, possibly in 2 rooms				
			PC for spoken tutorial				PC for test
			FOSS loaded?	Spoken Tut. copied?	Plays in VLC?	Audio works?	Internet works?
PC 1							
PC 2							
⋮							
PC 50							

Figure 3: A sample checklist to be completed before starting a SELF FOSS Study Workshop

Getting Started

- 04:17: Perform the following calculations on the scilab command line:

$$\text{phi} = \frac{\sqrt{5} + 1}{2} \quad \text{psi} = \frac{\sqrt{5} - 1}{2}$$

Find $1/\text{phi}$ and $1/\text{psi}$

- 6:06: Verify Euler's identity: Is $e^{\pi i} + 1$ close to zero? Compare with $\cos(\pi) + i \cdot \sin(\pi)$
-

Figure 4: Assignments with timing for the *Getting Started Spoken Tutorial on Scilab*

verbs: click, locate, copy, open, gedit, right click, etc. Theory is introduced only slowly.

These instructions are based on the time when a certain activity has to be carried out. For example, the instructions numbered 7, 8 and 9 in Fig. 5, respectively, are supposed to be carried out at 1:57, 2:04 and 3:04 min.

There could be instructions to point out the difference in the activity. For example, the instruction at 3:04 says not to invoke the command `skim`, but to use `evince`. This methodology can also be used to correct minor mistakes, if any, in the spoken tutorial.

These instructions are detailed. For example, in the invocation of `evince` there is a space before the ampersand (&) symbol to help emphasize it. Normally, such detailed instructions are unnecessary for Unix users. But for Windows users who may not know the difference between a terminal and an editor and the concept of background jobs, the instructions need to be detailed.

Finally, there are assignments that need to be done at a specific time. This is clear from a sample assignment question of a Scilab spoken tutorial workshop, given in Fig. 4.

3.2 Conductor of the workshop

We do not need a domain expert to conduct these workshops. In a large country like India, one may

have to conduct these workshops in thousands of places. It would be difficult to get domain experts to even visit such a large number of places, let alone *lecturing* in these workshops.

We insist that the conductor of the workshop *not* answer any domain-dependent questions. There are two reasons for this:

- We are not sure about the capability of the conductors of these workshops. We definitely do not want them to give wrong answers.
- Answering domain-dependent questions could take up a lot of time. As a result, a conductor may not be able to handle more than a few students. It would be impossible even for an expert to answer all the questions of twenty students, the recommended ratio in our methodology.

The students are supposed to follow the steps exactly as given in the instruction sheet and the spoken tutorial. Two types of difficulties can arise:

- If the students have difficulty in following any instruction, the conductor of the workshop should point out the mistake made by the student. If the mistake is a serious one, the conductor could even ask the student to start the tutorial from scratch. As the tutorials are short, one will not have to spend a lot of time in repetition.
- It is possible for a student to try out some changes of their own. They can do this so long as they do not encounter any problems. If they experience any difficulty, they are recommended to go to the next tutorial. The idea is that there are enough things (that work) to learn, before trying something of one's own.

To handle these two difficulties, the conductor of the workshop need not be a domain expert. As a corollary, a person who is trained to conduct a workshop on a topic (say, \LaTeX), can conduct a workshop on another topic (say, Scilab) also.

The procedure indicated above suggests a definite set of things to learn during the workshop. The learner may not have the freedom to learn whatever



Spoken Tutorial Based \LaTeX
Workshop on Linux
Spoken Tutorial Team
IIT Bombay
5 August 2011



First tutorial: What is Compilation?

These detailed instructions are intended mainly for Windows users, who may have to use GNU/Linux for learning \LaTeX . GNU/Linux users will already know most of this.

1. Click the **Places** button in the top left hand corner and then click the **Home Folder**. The folder that opens is called your **home** folder.
2. Please locate the folder **LaTeXWorkshop** that is available on the desktop. The sub-folder **01-compilation** contains the following files that you need for this tutorial: **hello.tex** and **compiling.mov**.
3. Please copy **hello.tex** from this folder to your **home** folder.
4. Open the **terminal** using the command **Ctrl-Alt-t**, by pressing all these three keys simultaneously.
5. Open the file that you copied above into the editor using the command

```
gedit hello.tex &
```

Do not forget the symbol ampersand (&) at the end of the command, obtained by pressing **shift 7**. Please leave spaces exactly as given above.

6. Right click on **compiling.mov**, point the cursor on **Open With** and select **VLC Media Player**. Now listen to this spoken tutorial.
7. As shown in the video at 1:57min, compile from the terminal the file **hello.tex** using the command

```
pdflatex hello.tex
```

Note that **pdflatex** is *one* command. Please do not leave a space between **pdf** and **latex**.

8. Pause the video at 2:04min. You should now be able to give the command **pdflatex hello.tex** and get a file **hello.pdf**. If there is any difficulty in this step, please listen to the tutorial from 1:57min to 2:04min once again.
9. The video talks about a PDF viewer called **skim** at 3:04min.

- Please do not attempt to use **skim**—it is *not* available on GNU/Linux.

You have to use the PDF viewer **evince** instead. Give the following command from the **terminal** to open the PDF file:

```
evince hello.pdf &
```

Once again, do not forget the & symbol in the above command.

Figure 5: A sample of instructions for a SELF FOSS Study Workshop on \LaTeX . To make it suitable for self-study, all initial instructions are activity based.

they want. But there are enough new things to learn in any case during the course of the workshop. This topic is explained further in the next section.

3.3 Duration of the workshop

We recommend a duration of two hours for these workshops. A workshop typically has about ten spoken tutorials. In a two hour period, one can learn four tutorials. More importantly, students will figure out how to learn from spoken tutorials. Thus, students can learn the remaining tutorials on their own.

So, why do we restrict the workshop to two hours? There are many reasons for this:

1. This will allow the same facility to be used for other workshops or for other people or both.
2. As our workshops are conducted free of cost, it is not clear how many students are really interested in the workshop. We do not want to host any uninterested students longer than absolutely necessary.
3. The students who are interested in appearing in the online exams have to necessarily complete the tutorials on their own. This improves the quality of learning and hence can indeed help provide better learning than conventional workshops.
4. In government supported training programmes, not only do the students pay nothing, but there are organisational expenses as well. For example, the conductors of the workshop have to be paid an honorarium for their time and effort. Such expenses are reduced by minimising the workshop duration. If any workshop is conducted for a longer period, there are also demands for a break and financial support for refreshments.

Not all students who undergo the two hour workshops have computers at home. So, if a college wants to offer their premises for learning, we have no objection. We tell them, however, that such sessions are not a part of our training programme.

A comment about the quality of learning in the two hour workshop is in order. In the previous section, the highly regimented procedure of the workshop has been discussed. Although no green field type of learning is possible, in no other method can one learn in two hours:

1. how to write letters using \LaTeX ,
2. how to write reports,
3. writing mathematics and equations, and
4. and introduction to presentations using Beamer.

We guarantee all of the above in a two hour workshop.

The two hour workshops discussed above are called *Spoken Tutorial based education and learning*

through *Free FOSS Study Workshops* or *SELF FOSS Study Workshops*. The word *Free* in the above denotes *free of cost*, unlike the word *free* that comes in FOSS, which denotes freedom.

As mentioned earlier, the students who undergo training through these workshops do not pay anything. As of now, we actually *spend* about Rs. 500 for every twenty students as honorarium expenses for the organisers. Thus, the cost of training one student on one software package is Rs. 25. We are now planning to do away with this honorarium expense.

The SELF FOSS Study Workshops have been extremely effective and also popular. In a workshop on GNU/Linux conducted in an engineering college in Alwar, Rajasthan, by their own student volunteer, the average marks went up by 85% after the workshop, although the post workshop test was tougher than the pre-workshop test. As a matter of fact, every student passed the second test and received a certificate of completion.

4 Conclusion and future work

This article has presented an instructional methodology for conducting large number of FOSS workshops on software systems, such as L^AT_EX, Scilab, Python, PHP/MySQL and GNU/Linux. Using this scalable method, we expect to conduct about 500 workshops in a period of six months, with an average number of participants in each workshop of about 50.

Although the current workshops are organised only for college students, we hope to extend them to secondary schools as well. We will pursue this activity as soon as instructional material on LibreOffice is ready. Office seems to be the most important software for schools.

A more interesting question is whether it is possible to create instructional material to teach L^AT_EX for school students. It is likely that the methodology of conducting the workshop will work for schools also. What is not clear, however, is whether a special type of instructional material has to be created exclusively for schools.

We also desire to teach more advanced topics on L^AT_EX to the students who successfully complete the basic training explained in this work. This requires more spoken tutorials. We hope to get help from the T_EX community for this purpose.

Finally, it will be useful to create more rigorous evaluation methods to check the efficacy of learning. At present, we use only multiple choice questions. It is not clear how to administer exams to check the L^AT_EXing capability of the large number of students who may take the exam. To preserve scalability, we need automatic evaluation methods.

References

- [1] Tamil dubbing: T. Vasudevan and Priya. Report Writing in L^AT_EX, Seen on 4 Nov. 2011. Video available at <http://spoken-tutorial.org/How-to-write-a-Report-using-LaTeX-Tamil>.
- [2] K. M. Moudgalya. *Digital Control*. John Wiley & Sons Ltd., Chichester, 2007.
- [3] K. M. Moudgalya. Spoken tutorials. In *Technology for Education, T4E 2009*, pages 17–23, Bangalore, August 2009. IEEE.
- [4] K. M. Moudgalya. Spoken Tutorial: A Collaborative and Scalable Education Technology. *CSI Communications*, 35(6):10–12, September 2011. Available at <http://spoken-tutorial.org/CSI.pdf>.
- [5] K. M. Moudgalya. Presentation using L^AT_EX and Beamer, 3 November 2009. Video available at http://spoken-tutorial.org/Latex-beamer_english.
- [6] K. M. Moudgalya. What is a Spoken Tutorial, 8 March 2011. Video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial.
- [7] K. M. Moudgalya. Embedding Maths in Xfig, 9 Feb. 2011. Video available at http://spoken-tutorial.org/Xfig-Feedback_Diagram_with_Maths.

◇ Kannan M. Moudgalya
 Dept. of Chemical Engineering
 IIT Bombay, Powai
 Mumbai 400 076, India
 kannan (at) iitb dot ac dot in
<http://spoken-tutorial.org>