resulting C code. Like processed cheese, the food value of the original is there, but the flavor is changed and the texture is gone.

Distributing the Product

We have elected to make TurboTEX a "semi-commercial" product within the US, that is, we will charge a modest license fee for each copyrighted copy of the binary and/or source code. However, unlike other commercial versions of TEX, the source code will still cost less than the other's binaries. We will distribute a complete package including TurboTEX, utilities, and printer drivers.

Late-breaking News. We have completed some preliminary benchmarks on the VAX BSD version of TurboTEX, with encouraging results. We compared TurboTEX in C to the public-domain Unix TEX distribution in Pascal on a VAX 750. We observed an execution speed-up factor of between 1.6 to 3.0 compared to the Stanford distribution TEX (the factor varies depending on the type of document being formatted). The size of the TurboTEX executable code is about 60% of the distribution version.

easyTEX

Ester Crisanti
Alberto Formigoni
Paco La Bruna

1 Introduction

1.1 easyTEX$_{1.0}$

TEX has introduced new powerful tools for scientific documents typesetting, allowing formulae to be easily built up through a linear language. As a new tool using TEX, a project was born in 1984 at the Istituto di Cibernetica (now Dipartimento di Scienze dell'Informazione) dell'Università degli Studi di Milano, Italy.

That project has produced easyTEX$_{1.0}$ that we propose as a new powerful tool for TEX documents typesetting.

easyTEX is an interactive Formula Processor, developed from the initial idea of Prof. Gianni Degli Antoni, Dipartimento di Scienze dell'Informazione, planned and implemented by TE CO GRAF with the collaboration of Dipartimento di Scienze dell'Informazione dell'Università degli Studi di Milano.

It allows the interactive typewriting of mathematical formulae on IBM-compatible Personal Computers. The formulae produced by easyTEX are memorized in ASCII standard files, prepared in order to be processed by TEX, either including such files in other ones by means of the TEX command "`\input`", or using usual editor commands for file merge.

The formula being built up is displayed on the screen through the fonts created with METAFONT and it is also possible to use every symbol and mathematical font.

The use of easyTEX is very simple, since the user is driven in his work by a pop-up menu interface, by means of which the choice of operators and mathematical symbols is easily made. It is also possible to select some virtual keyboards which, because they can be displayed on the screen, achieve a correspondence with the physical keyboard, allowing insertion of characters belonging to different alphabets, like the greek, or a wide selection of mathematical symbols.

Also, complex mathematical formulae can be typeset in an easy way, similar to the one used in writing by hand the same formula. Both for the foregoing reasons, and because the positioning of the cursor is automatically obtained through an interactive construction of the formula on the screen, easyTEX offers to the user a good facility for the preparation of a TEX document.

easyTEX has been implemented using attributed grammar techniques, as developed by D.E. Knuth. Programs have been written in C language.

2 Functional characteristics

2.1 User interface

The user communicates with easyTEX using pop-up menus making the selection of commands simple and fast. Using easyTEX, it is not necessary to know editing languages or to learn a particular syntax for the commands, because everything is done in an interactive way.

2.1.1 The screen layout

The screen handled by easyTEX is structurally divided into three separated areas named:

- **Menu line**
- **Work area**
- **Status line**

The *Menu-line* is on the upper part of the screen and displays a sequence of names each representing a **Menu name**.

The *Status-line* is on the lower part of the screen and contains a set of information concerning easyTEX' state.

The *Work-area* occupies the whole part of the screen extending between the *Menu-line* and the *Status-line* and contains the formula the user is working on.

2.1.2 The pop–up menus

Every pop–up menu contains a sequence of elements:

- *Menu-name*: its selection displays a different pop–up menu
- *Command-name*: its selection executes the associated command

Selecting commands inside the pop–up menus is very easy. You have first to activate the "menu mode" through the **F1** key. On the screen you will see the pop–up menu and any command or menu may be selected by stroking the name capital letter or positioning the cursor on the name (through the two keys <↓> and <↑>) and then stroking the < ↵ > key. If you have chosen a *Command-name*, the operation is immediately performed; on the contrary, if you have chosen a *Menu-name*, on the screen you will see the corresponding pop–up menu.

The organization of pop–up menus in easyTEX is hierarchical. At the first level there is the *Menu-line* and at the second the **O**perator and **G**eneral pop–up menus. In the **O**perator pop–up menu there are some *Menu-name*s, whose selection performs the displaying of other pop–up menus forming the third level.

2.1.3 Virtual keyboards

The selection of the characters belonging to different character fonts is made through a system of **virtual keyboards**.

easyTEX associates to every character font a **virtual keyboard**, in which a symbol of the current character font corresponds to each key of the physical keyboard. During the working session of easyTEX it is possible to display the **virtual keyboard**. easyTEX provides five **virtual keyboards**:

- Italic
- Romanic
- Boldface
- Greek
- Symbols

2.2 How to build up a formula

Suppose you to want to input the formula:

$$V(x) = \int_0^1 F(x)dx$$

You first have to write the "$V(x) =$" string; this is done very easily: the association between the **virtual keyboard** and the physical one has not changed the meaning of the keys. We have therefore the following result, with the position of the cursor represented by the box:

$$V(x) = \square$$

Now you have to write the integral. After activating the "menu mode" by the function key **F1**, from the *Menu-line* you select the *Menu-name* **O**perator. On the screen you will see, under the element selected from the *Menu-line*, the corresponding pop–up menu:

Operator	General
Triple	
Fraction	
Root	
Exponent	
Index	
Block	
Matrix	
tri**G**	
Accent	
Dots	

Let's now select the *Menu-name* **T**riple and you will see now on the screen, by the side of the element selected by the pop–up menu, the window containing the corresponding pop–up menu:

Operator	General
Triple	**I**ntegral
Fraction	**S**um
Root	limi**T**
Exponent	lo**G**
Index	**L**n
Block	**M**in
Matrix	ma**X**
tri**G**	uni**O**n
Accent	interse**C**t
Dots	pro**D**
	i**N**f
	s**U**p
	limin**F**
	limsu**P**
	overbr**A**ce
	under**B**race

Let's finally select from the **T**riple pop–up menu the *Command-name* **I**ntegral, obtaining:

$$V(x) = \int_{\square}$$

Not only the integral symbol has been displayed, but also the cursor has been correctly positioned to write the lower limit and the font in use has been reduced from size 10 pt to size 7 pt. This size reduction has also changed the cursor dimension. It is now possible to insert the lower limit, obtaining:

$$V(x) = \int_{0\square}$$

In order to set the end of the lower limit you stroke $< \hookleftarrow >$.

$$V(x) = \int_{0}^{\square}$$

The cursor is automatically positioned on the upper limit. This is written as the lower limit and, when closing it through the $< \hookleftarrow >$, you obtain:

$$V(x) = \int_{0}^{1} \square$$

At last you input the "$F(x)dx$" string, obtaining the complete formula:

$$V(x) = \int_{0}^{1} F(x)dx \,\square$$

2.3 The editing commands

2.3.1 The cursor movements

A mathematical formula consists of strings (i.e., strings of alphanumeric characters and symbols) and of a class of structures, such as fraction, triple, root and so on.

Every complex mathematical formula may therefore be decomposed into simple formulae, which are in turn reduced to single characters. For instance, in the same way as the string consists of single characters

$$\boxed{a}\boxed{+}\boxed{b}\boxed{=}\boxed{c}$$

you can single out in a fraction the numerator and the denominator; each of these may in turn consist of other mathematical structures or of strings.

$$\frac{\boxed{a+b}}{\boxed{\sin\alpha}} \Rightarrow \frac{\boxed{numerator}}{\boxed{denominator}}$$

$$\frac{\boxed{a+b}}{\sin\boxed{\alpha}} \Rightarrow \frac{\boxed{string}}{\boxed{trigonometric,\ string}}$$

The splitting of a formula into strings and structures simplifies the movements across a mathematical text. easyTEX allows two movement modes:

- by character
- by structure

The initial movement mode is by character.

Let's now see how the structure movements take place, considering a simple formula:

$$\sum_{\boxed{n}=n_0}^{\infty} (-1)^n u_n$$

As you want to move in structure mode, you have to stroke the $< home >$ key. On the screen you will now see:

$$\sum_{\boxed{n=n}_0}^{\infty} (-1)^n u_n$$

The cursor contains the "string" structure "$n = n$". This is consistent with what we said above; actually the lower limit consists of a string and of an index, and the cursor was positioned on one of its characters. By again stroking $< home >$ the cursor gets the dimensions of the entire summation lower limit:

$$\sum_{\boxed{n=n_0}}^{\infty} (-1)^n u_n$$

By stroking $< home >$ again the cursor gets the dimension of the entire "Triple Summation" structure:

$$\boxed{\sum_{n=n_0}^{\infty}} (-1)^n u_n$$

and at last, by stroking $< home >$ again the cursor gets the size of the entire formula:

$$\boxed{\sum_{n=n_0}^{\infty} (-1)^n u_n}$$

Let's now stroke the $< end >$ key. The cursor gets the size of the summation lower limit:

$$\sum_{\boxed{n=n_0}}^{\infty} (-1)^n u_n$$

and stroking again the $< end >$ key, the cursor gets the size of the first structure of the lower limit:

$$\sum_{\boxed{n=n}_0}^{\infty} (-1)^n u_n$$

and by typing another $< end >$, the cursor gets the size of the first string character:

$$\sum_{\boxed{n}=n_0}^{\infty} (-1)^n u_n$$

This explanation may appear very complex, but using the $< home >$ and $< end >$ keys is actually very easy and allows the crossing of a formula in a faster way than the one provided by the character mode. If any mistake is made, for example by including a structure more external than the one you want, this is immediately visible, as the cursor gets always the

size of its content, and the two keys < *home* > and < *end* > let the user re-establish the desired situation.

2.3.2 Deletion of characters and structures

easyTEX provides the possibility of deleting any character or mathematical structure that the cursor movements can visit. Two keys are available for deletion:

- < *BS* > (backspace key)
- < *del* > (delete key)

Backspace The usefulness of this key is evident during the input of a formula in case of a stroking mistake. For example, in the formula:

$$\sum_{n=n_a\square}$$

We have a wrong index. By stroking < *BS* > you obtain:

$$\sum_{n=n_\square} .$$

Delete By means of this command, it is possible to delete the portion of the formula included by the cursor. The combined use of < *home* > and < *end* > makes easier the selection of what is to be deleted.

2.3.3 Insertion of characters and structures

To add a new part to the formula, it is sufficient to stroke what you want to input, with the normal procedure. On the contrary, if the new part is to be inserted in the middle of the formula, you have first to move the cursor to the position immediately after where the new part is to be placed. Considering our example, if you want to vary the lower limit by adding the "$j = 0$," string, you have to stroke the key <←>, positioning the cursor as follows:

$$x_i = -\sum_{\boxed{k}=1}^{n} \lambda_{jk} x_k$$

and then to input the string, obtaining:

$$x_i = -\sum_{j=0,\boxed{k}=1}^{n} \lambda_{jk} x_k$$

If you want to input a mathematical structure, you have to use the same method. For example, let's insert a fraction before the summation. You have to move the cursor to the Summation, by using the keys <←> and < *home* >, and you get to this situation:

$$x_i = -\boxed{\sum_{j=0,k=1}^{m,n}} \lambda_{jk} x_k$$

Let's now select from the **O**perator pop–up menu the *Command–name* **F**raction, yielding:

$$x_i = -\frac{\square}{\square} \sum_{j=0,k=1}^{m,n} \lambda_{jk} x_k$$

Let's then insert the numerator and the denominator, yielding:

$$x_i = -\frac{\omega}{\varphi\square} \sum_{j=0,k=1}^{m,n} \lambda_{jk} x_k$$

easyTEX provides also the insertion of mathematical structures and of characters after the cursor positioning.

2.4 The TEX–interface command

easyTEX$_{1.0}$ is designed only to produce formulae to be inserted into TEX documents. Therefore, after building up the formula, easyTEX has to properly organize it so that it can be processed by TEX, i.e. easyTEX has to translate it into TEX source format. TEX is able to produce a formula in two different ways:

- in text mode: when the formula is on a line with normal text
- in display mode: when the formula is alone on a line.

The two representations differ remarkably only as to the size.

2.5 How to insert the formulae into the text

The combined use of easyTEX$_{1.0}$ and of TEX allows the creation of documents consisting of normal text and mathematical formulae.

You first have to recognize the parts of the document you have to produce with easyTEX and to associate a name to each one. Then, you have to build each formula up by easyTEX and to request its translation into TEX format.

Next, you build up the current text by means of the text–editor and then, in the proper positions, you input the mathematical formulae produced by easyTEX in one of the two possible ways:

- by using the TEX command:

  `\input` < *file name* >

- by using the text–editor command to *merge* two files (for every formula).

At this point, the document is ready to be processed by TEX.

3 Future developments

Two other releases will be carried out for easyTEX.

easyTEX$_{2.0}$ will have an interactive Word–Processor that will immediately show the action of all commands.

Besides all the usual word processing functions, easyTEX will allow the interactive and automatic pagination of the text and will perform wrap–arounds, applying, if necessary, an algorithm for hyphenation.

The document layout may be established by the author or selected from a library of standard document layouts. This library may be updated and extended by the author, thus allowing him to create his own document layouts library. The author may, anyway, locally change the document layout for special purposes.

A fundamental characteristic of the Word–Processor is the usage of fonts. easyTEX enables the use of several typographical fonts; the author may select different fonts within the text, and the resulting text image will be displayed interactively on the screen yielding a WYSIWYG interactive Word–Processor and Formula–Processor. Fonts can be selected from a library.

Since easyTEX, to set a page, looks up the different sizes of characters, the space between two lines is adjusted according to the biggest character (box) of the second line; the justification, on the other hand, is carried out re–arranging adequate units (pixels) of white space between words.

easyTEX$_{2.0}$ will have a Box–Processor that will allow text integration with "objects" (i.e., texts such as spread sheet tables, and images such as pictures and drawings) produced by other systems and whose file formats are known.

Using several commands, the author will be able to define, edit (copy, move, change, etc.) and format empty "boxes" within the text, which may be filled with the "objects" created by other systems, and contained in ASCII (for instance, PostScript files) or bit–map files whose file format is known; in the latter case, the images must already have all the characteristics necessary to make them printable on the target device, as easyTEX$_{2.0}$ performs scaling of images only between printer and screen formats. Anyway, easyTEX$_{2.0}$ is not an "Image Processor"; it is able, however, to give a text–image integration.

easyTEX$_{2.0}$ will also produce a source file for TEX, for more powerful processing through *passive* commands, i.e., commands ignored by easyTEX and passed to TEX for fine–tuning purposes.

We have received some other suggestions for extensions to easyTEX, such as the integration of another environment devoted to *graphs design*, useful in industrial project design; we are now evaluating the opportunity for such extensions.

We have also been requested to design a *Document Data Base*, based on a Local Area Network among PCs and a host system and using *CD–ROMS*, able to solve documentation (also technical) problems in industrial organizations. Such a system, based on TEX and easyTEX, builds on the experience we have gained with *SDDS*, together with Mondadori publishing company, CILEA and Università di Milano, Dipartimento di Scienze dell'Informazione, as one of the DOCDEL experiments supported by the Commission of the European Communities.

4 easyTEX hardware requirements

easyTEX runs on PC–IBM and compatibles equipped with the MS DOS operating system, release 2.0 or later.

easyTEX needs one of the following graphic cards:

- Hercules or Hercules–like graphic card (720x348 pixels),
- IBM Enhanced Graphic Adapter (640x350 pixels),
- OLIVETTI M24 Graphic Card (640x400 pixels),
- NCR PC6/8 Graphic Card (640x400 pixels),
- other graphic cards compatible with those described.

5 References

TE CO GRAF snc. is a company working on electronic publishing, in collaboration with the Dipartimento di Scienze dell'Informazione dell'Università degli Studi di Milano, Italy.

Refer to Paco La Bruna for any question.

TE CO GRAF snc
Via Plinio, 11
20129 MILANO, ITALIA
Phone: 2–20 81 50 and 2–27 80 63
Telex: "340160 PER TECOGRAF"