4TEX: a TEX Workbench for MS-DOS PC's

Wietse Dol, Erik Frambach, Maarten van der Vlerk

Department of Econometrics
University of Groningen
P.O. Box 800
9700 AV Groningen
W.Dol@eco.rug.nl, E.H.M.Frambach@eco.rug.nl, M.H.van.der.Vlerk@eco.rug.nl

Abstract

TEX and all its companions offer an enormous amount of possibilities. This is both an advantage and a disadvantage. The advantage is that almost anything is possible; the disadvantage is that you need detailed knowledge of all related programs to fully exploit the possibilities.
The MS-DOS program 4TEX is an attempt to integrate all major TEX related programs in a shell that shields you from the tedious and frustrating job of setting environment variables and program parameters.

1 Introduction

Preparing documents with TEX or LATEX is by no means a simple job. However, those who persevere are rewarded with a beautifully typeset document. There are at least two reasons why writing with the 'aid' of TEX may cause some problems.

First of all there is the enormous amount of possibilities that TEX offers. While this is the reason to use TEX in the first place, it has the disadvantage that many—at first puzzling and maybe hard to remember—commands have to be used. In principle this is a problem that you will have to live with. A thought to comfort you: many people have tried before you and most of them are by now enthusiastic TEX users.

The second reason why TEX may seem difficult to work with is of a more technical nature. Since TEX itself is only(!) a compiler, it needs several other programs and utilities to make a fully operational text preparation system. For example, at some time between conception and delivery of your document, you might need an editor, a previewer or a printing program. All these programs need setting of environment variables and/or some parameters to cooperate the way you want them to. This is of course just the type of thing you do not want to bother about.

So here is the good news: you don't have to! If you do not like to spend valuable time discovering all these technicalities, try 4TEX. While in its infant days, it did nothing more than constitute a simple menu for calling the TEX compiler or an editor and the like. Although it is a relatively young program and is still growing, we feel that by now it is powerful enough to justify the title *workbench*. In the next sections you find a description.

2 Principles of 4TEX

The 4TEX system consists of a host of separate programs, e.g. it uses emTEX as TEX implementation, which is public domain but state–of–the–art. It is named after its author, Eberhard Mattes, from the Stuttgart University. You must prepare your TEX input file with an ASCII editor (we suggest you use QEDIT); the actual TEX compiler converts this input file into a `.dvi` file (from DeVice Independent); separate programs generate printed output from the `.dvi` file and allow you to preview the typeset page on your screen. In addition, there are a great many add-on utilities: a spell-checker; a database program for maintaining bibliographic references; an index generator; a utility to remove TEX codes; a font generation program, extended graphics support etc.

Some of these programs, especially the various versions of the compiler and of the print and preview programs, require *lots* of parameters and/or environment variables. 4TEX is designed to shield you from the dirty bits.

As mentioned above, the first objective in creating 4TEX was to have some sort of integrating menu to call TEX and the other main programs without the fuss of setting, let alone remembering parameters.

One obvious way to do this is by a *batch file*. However, plain `command.com` batch files tend to be slow, since only one command is read at a time. A much bigger draw-back is its lack of even the most basic commands for an interactive system. A very attractive alternative is provided by 4DOS (shareware by JP Software Inc.), nowadays a well known replacement of `command.com`.

What once started as a small and simple batch file, grew into a collection of batch files that currently comprises over 7000 lines of sometimes fairly sophisticated code. The main reason to program 4TEX in the 4DOS batch language and not in a higher programming language (e.g. Pascal or C) is that by using 4DOS we could create an *open system*, i.e. everyone can modify the workbench to suit personal needs and taste without the need of special tools or extra compilers. Another reason to use the 4DOS batch language is the availability of environment variables and variable functions that enabled us to do things that would require very tedious programming in a higher programming language. Since it is also fast (the complete batch file is read into memory at once), it was an easy decision to implement 4TEX as a 4DOS batch file.

One might object that using 4DOS batch files deprives the old-fashioned `command.com` users from the benefits of 4TEX. We happen to think that this would only be a mild punishment for not recognizing how good 4DOS really is. However, for those who have a good reason not to use 4DOS we also implemented 4TEX to run under `command.com`. This is of course done by loading 4DOS as a secondary shell.

Though 4DOS is very powerful, some routines had to be written in other programming languages, e.g. to support mouse operation and to select files. Some *public domain* and *shareware* programs are used, e.g. MARKNET.EXE, RELNET.EXE and PKUNZIP.EXE.

2.1 Freeware, shareware and commercial software

4TEX uses several public domain (*freeware*) and *shareware* programs.

You may use a shareware copy for an evaluation period of 21 to 30 days. The purpose of this evaluation period is to allow you to determine whether the program meets your needs before purchasing it. Once the evaluation period ends, you agree to either purchase a registered copy of the program, or to stop using it.

Using freeware and shareware only, we are able to distribute 4TEX without violating any law or agreement. As a 4TEX *user* you are supposed to pay for the shareware programs that are listed in the documentation (see Chapter 13, page 79). Remember that 4TEX could never have been build without these programs.

Commercial software is not used, though interfaces to a few programs (e.g. EUROGLOT, VERTAAL! and WORDFINDER) are implemented. Naturally commercial fonts can be added. All DVI-drivers are installed in such a way that this should be easy.

2.2 Setting up 4TEX

After installing 4TEX from the distribution files, some customization is required before 4TEX will run.

The most important files are `texuser.set` and `system.set`. In these files all parameters that 4TEX needs are set. The file `system.set` contains values for the parameters that are needed for any user (e.g. what options does the compiler need, where are the fonts located), whereas `texuser.set` contains values for the parameters that are user specific (e.g. where TEX files are stored, what screen colors should be used).

As you might have guessed, 4TEX runs perfectly on a network. Because NOVELL NETWARE is the most popular type, 4TEX has an interface to run from such a network. However, only a few lines of code in one of the batch files need to be changed to make 4TEX run on any other type of network.

3 Features

In this section we will describe some features of the 4TEX workbench.

3.1 Add, delete, modify

4TEX aims to be an open system such that every user can add, delete or modify the 4TEX workbench to suit personal needs and taste. For instance 4TEX uses Babel: a simple way to generate TEX format files with multiple languages, and some control sequences to switch languages (and hyphenation patterns) within one TEX document. It is easy to adapt Babel for other/more languages than already used by 4TEX (i.e. English, German, French and Dutch). It is easy to add, modify or delete format files (e.g. 4TEX supports the New Font Selection Scheme).

Of course any user or department has specific printers. 4TEX currently supports matrix printers, LaserJets, DeskJets and PostScript printers. Adding or deleting a printer type should not cause problems. 4TEX lets you choose between local printers, network printers and printing to a file.

At this moment the spell-checker supports the languages Dutch, US English, English, French and German, but any other language can easily be added.

3.2 Help in 4TEX

There are several types of automated assistance available. For each item in every 4TEX menu there is a help screen.

The memory resident program TEXHELP is meant as partial replacement for the LATEX and TEX manuals. It is a hypertext system that can be called from the editor. For example, if the cursor is at the word `\documentstyle`, pressing the TEXHELP key results in a help screen that refers you to the topic `Document Styles`. Select this topic and you get a help screen that gives information about all valid LATEX document styles and options. Moreover, there are cross references to all options and, among others, the commands `\flushbottom` and `\twocolumn`. It is also possible to jump directly to the index of TEXHELP or to review the last help screen. For most topics, the text is taken from the LaTeX help system for MS-WINDOWS by Michael F. Reid, which is based on George Greenwade's help system for Vax-VMS computers. At this moment there are about 315 documented topics in the TEXHELP database.

Several *example* files exist, varying from a standard LaTeX file to more complex subjects, such as tables with asymmetric columns or creating multiple indexes. Example files can be viewed from the editor and, if desired, be inserted in the current document.

Finally, there is on-line TeX documentation. This consists of documentation of e.g. style files. To save disk space, the documentation is kept in archives that are temporarily decompressed on selection.

3.3 Editing

We have chosen QEDIT (shareware by Semware) as the default editor in 4TeX. For this editor, we have developed many macros. For example, you can enter a LaTeX environment defined by the commands `\begin{env}` ... `\end{env}` by picking it from a list; insert `\index{this}` behind the word `this` at the cursor position; or call the spell-checker (see below) to check the word at the cursor. Moreover, on loading a text file, the cursor will be placed at the exact location it was when you last edited the file (also available in the View-Block facility, see below).

3.4 Block-View: quick partial compilation

One of the most often heard objections to TeX is that it is not a so called WYSIWYG system. On the other hand, TeX users often reply that the acronym WYSIWYG is misleading: it should be WYSIAYG (What You See Is *All* You Get). We do not wish to solve this problem. Instead, 4TeX offers a feature that we think both sides may like.

Imagine that you have just typed a very complicated formula. The natural way to check whether you did not make a mistake, is to see what it looks like. Or maybe you are just curious. However, to see this single formula you would have to leave the editor, compile the entire document, start the previewer, and find the formula. Then you need to return to the editor, and find the exact position where you left it. If you made a mistake, you have to do this all over again.

In 4TeX this whole procedure can be performed by the touch of a few keys. Even better, only the preamble of your document and the formula are compiled, which makes it as fast as possible. To view a certain part of a document, mark it as a block in the editor and press the 'view-block' key. Alternatively, if you want to view an entire file, just press the 'view-file' key. If you are making slides, it makes sense to view a complete slide at a time. Therefore, in this case 4TeX automatically defines the before mentioned block to be the current slide. If TeX finds an error in the block, the user has the option to be returned to the editor at the line that contains the mistake.

3.5 Spell-checking

In 4TeX we use AMSPELL, a public domain program by A. Merckens, to check and correct spelling in TeX documents. AMSPELL is basically a spell-checker for plain ASCII files, with some special features for dealing with TeX files.

The basic idea behind this program is to make spell-checking easier by

- providing the context of the possibly misspelled word;
- offering alternatives;
- offering facilities for editing the word;
- automatically replacing misspelled words in your document;
- learning new words.

AMSPELL does not require TeX commands to be removed from your document (deTeXing). In fact, it will even interpret the standard accenting commands like `\"`, `` \` ``, `\'` and will automatically use them while replacing misspelled words. When checking a TeX file, AMSPELL will ignore all text between `$`'s and `$$`'s. Furthermore, AMSPELL will ignore parameters of the LaTeX commands `\ref`, `\pageref`, `\cite`, `\nocite`, `\label`, and all text between `\begin{equation}`, `\begin{eqnarray}`, `\begin{eqnarray*}`, `\[` and their counterparts like `\]`. You can change or expand the lists by means of environment variables.

3.6 Extended graphics support

TeX has been developed with the idea that it should be possible to have a TeX implementation for every operating system (MS-DOS, VMS, UNIX etc.). Another important feature of TeX is that documents can be freely exchanged between operating systems (because documents are written in standard ASCII). Graphics, however, are machine dependent and the possibility to include graphics in TeX or LaTeX depend on the operating system and the DVI-driver you are using.

A solution often used for including graphics in TeX documents is to insert PostScript pictures in the document through `\special` commands. The `\special` command is ignored/passed on by the TeX compiler but the PostScript DVI-driver knows how to insert the PostScript picture at the right place and in the right size in your document. The disadvantage of this method is of course that you can only include *PostScript* pictures in your document and that you need a PostScript printer to produce output.

The emTeX DVI-drivers support a `\special` command to include black-and-white bitmapped pictures. Both this feature and the PostScript possibilities are used by 4TeX to incorporate pictures in TeX documents.

Graphic files come in many flavors. 4TeX allows you to view, manipulate and include the following types of pictures in your TeX documents:

- bitmapped pictures: GIF, TIFF, PCX, BMP, IFF, LBM, IMG, CUT, and PCL;
- vector pictures: HPGL and PostScript.

This is done by using the following freeware software: BM2FONT, HP2XX, PCLTOMSP, and GHOSTSCRIPT.

4TEX automatically chooses the appropriate conversion program, depending on the type of graphic file, or rather the file extension. Any conversion that 4TEX performs will result in either TEX fonts (accepted by any DVI-driver) or both a PCX and EPS file (for emTEX and PostScript resp.). Furthermore, a small TEX file is produced that contains all the necessary commands to incorporate the picture in your document.

4 4TEX in practice

At the Department of Econometrics of the University of Groningen 4TEX is installed on a network, and its usage is logged. The time that 4TEX was running is logged and the number of calls to functions are counted.

To give you an idea of the activity we made a summary for the period of 21 January to 12 March.

# calls	*function*
17202	editor
4273	compiler
3984	block-view
204	4TEX help
942	LaserJet print
9	PostScript print
13	matrix print
2501	preview (not block-view)
230	view log-file
189	spell-check
26	oneword spell-check
689	shell to DOS
175	BibTEX
47	BiBDbase
216	view picture
6	bitmap to TEX font
47	HPGL to PCX + EPS
54	PS/EPS to PCX
0	PCL to PCX
5	GraphicsWork Shop
45	MakeIndex
1	DeTEX
0	UnTEX
5	Wordlist
4	TEXcheck
11	TEXcad
0	submit TEX batch
22	automatic font generation
196	user installed utilities

In principle, these figures speak for themselves. However, let us make a few comments.

The number of calls to the functions editor, compiler, print, and preview do not give information on how 4TEX is used in practice: they indicate actions that have to be taken whether you work in 4TEX or not. They do give information on the amount of TEX-ing that is going on at our department.

Among the other figures, the number of calls to the function block-view is rather striking. It indicates that this function is a feature indeed.

Another much used facility is shell-to-DOS. We suspect that many people start 4TEX only once per day, and from then on start any other program they need from the DOS-shell (at the expence of only 36K conventional memory). Note that the calls to user installed utilities represent activities that would have to be done in a DOS-shell in other programs.

Whereas without 4TEX most users would not even try to incorporate graphics in their document (other than by using scissors and glue), the total number of calls (328) to functions related to graphics bears witness to how easy this has become in 4TEX.

5 Availability and support

4TEX can be obtained through *anonymous ftp* from the file servers `obelix.icce.ruu.nl`, directory `/pub/erikjan/4tex`, and `ftp.cs.ruu.nl`, directory `/pub/TEX/4tex`. The system has a modular structure that allows you to install only those parts that you need. Therefore it is important that you read the file `install.txt` first. The installation files are compressed using ARJ.EXE and fit on 1.44 MB 3.5" high density diskettes. You may also take a look at the documentation file before you install anything. `4texdoc.dvi` and `4texdoc.ps` can be found in the same directory.

If you have trouble installing 4TEX or need more information you can send E-mail to

`4TeX-support@eco.rug.nl`.

However, don't expect an answer within the hour—we will try to help you as soon and as best as we can, but 4TEX is an after hours project.

4TEX users can join the 4TEX mailing list. On this list users can pose/answer questions regarding 4TEX usage. New or desired developments and features are also announced and discussed on this list.

TEX zonder omhaal voor Atari ST en andere PC's

Robert Best

best@zeus.rijnh.nl

Abstract

Deze cursus is bedoeld als eerste kennismaking met TEX op een eenvoudige PC. Een harde schijf is niet nodig. De cursus is gebaseerd op de PD-TEX van Christoph Strunk voor Atari ST. De installatie en de functie van de basis bestanden van TEX worden behandeld.
Deze cursus is een gecorrigeerde herdruk van een serie artikelen in het blad ST[1], uitgegeven door: Stichting ST [2] *Daar zijn de in de tekst genoemde schijfjes te verkrijgen.*

1 Inleiding

TEX automatiseert het werk van de zetter in een drukkerij. TEX software draait op vrijwel iedere computer. TEX kan simpel en goedkoop gebruikt worden; met de ca. 900 commando's zijn ook programmatische hoogstandjes te maken.
Dit artikel is een inleiding tot het gebruik van TEX. We beginnen zo eenvoudig mogelijk. Er hoort een schijf B118 bij[3] met een gebruiksklare selektie uit de PD TEX van Christoph Strunk.

1.1 Wat doet TEX?

In een ouderwetse drukkerij nam een zetter letterblokjes uit een bak en plaatste die op regels. Hij zette sommige letters in groepen (ligaturen) dicht tegen elkaar, b.v. ff fl fi. Sommige zetterijen hadden een schier eindeloze verzameling letters uit vele schriftsoorten en (wiskundige) tekens in vele formaten. De zetter verdeelde het 'wit' tussen de woorden zodat de regels even lang werden en een gelijkmatige verdeling van 'wit' in de alinea ontstond. Alleen als het niet anders kon, brak hij woorden af aan het eind van een regel.
Het pure handwerk bestaat nauwelijks meer. In alle commerciële drukkerijen is dit overgenomen door de computer. Bovendien gebeurt het zetten vaak helemaal niet meer op de drukkerij.

En dat is te merken. Sinds de meeste gebruikers hun materiaal 'camera ready' bij een drukker afleveren, is de kwaliteit van het zetwerk achteruit gegaan. Gewone tekstverwerkers leveren een zetsel op dat door de ouderwetse drukker afgekeurd zou worden: onjuiste woord- en letter afstand, incomplete fonts. Vooral voor wiskundige formules laat de tekstverwerker het afweten. Zelfs als de nodige tekens beschikbaar zijn, is het intikken moeizaam en het resultaat op papier abominabel.

TEX automatiseert het zetwerk met een resultaat dat vakkundig zetwerk benadert. Werken met TEX is moeilijker dan met een gewone tekstverwerker, maar het loont de moeite als je er oog voor hebt (gekregen). TEX is vooral geschikt om tekst met wiskundige formules mooi op papier te krijgen, maar met TEX kunnen ook teksten in b.v. Grieks, Hebreeuws of Russisch gezet worden.

TEX is géén direkt WYSIWYG (What You See Is What You Get) systeem. Je maakt een TEX-bestand, waarbij je op het scherm nog niet de opmaak in alinea's, de vreemde tekens, de verschillende schriften enz. ziet. TEX maakt daar een DVI-bestand van. 'DVI' staat voor 'device independent', oftewel uitvoerapparaat onafhankelijk. Deze uitvoer kan nu m.b.v. een DVI-programma bekeken of geprint worden. Dan zie je het resultaat pas. Je kunt een DVI-file ook van de ene computer naar de andere meenemen. Er is geen verschil tussen een Atari-, Mac-, UNIX- of DOS-DVI file.

TEX bestaat nu ca. 15 jaar. Al 10 jaar is TEX een voorbeeld van een vrijwel foutloos programma. In 1989 is een nieuwe versie verschenen met uitbreidingen maar geen wezenlijke wijzigingen. De schrijver van het pro-

[1] Uitgaven: ST37, ST39, ST41, ST42, ST43.

[2] Stichting ST, Postbus 11129, 2301 EC Leiden, tel. 071–130045. ST is een onafhankelijk tijdschrift van en voor gebruikers van Atari ST computers.

[3] Alle in dit artikel genoemde schijfjes zijn te verkrijgen via de Stichting ST.

gramma, Don. E. Knuth, hoogleraar informatica aan de Stanford Universiteit (USA), heeft het programma nu 'bevroren'.

Op schijf B118 staat de oude versie, omdat de nieuwe alleen compleet verspreid mag worden.

1.2 Wat heb je nodig voor TEX?

Minimaal een ST met 1 Megabyte RAM, dus een 520ST+ of een 1040ST. Beter is een ST met harde schijf, of 2 Mb RAM, zodat er ruimte is voor een 1 Mb RAM-disk om redelijk snel te kunnen wisselen tussen de editor en de TEX programma's. Verder een dubbelzijdige diskdrive en een SM124 of SM125 beeldscherm. Op een 9-naalds printer zijn al goede afdrukken te maken, maar de TEX kwaliteit komt beter tot zijn recht op een inkjet- of laser-printer.

En natuurlijk TEX software. Er is goede PD TEX voor de ST, maar het installeren is niet simpel zonder harddisk. Daarom heb ik uit de PD een schijfje samengesteld waarmee je zonder veel omhaal aan de slag kunt.

1.3 Installatie

Als er minstens 2 Mb RAM is, start de ST dan op met schijf B118 in drive A. Er wordt een RAM-disk van 1 Mb aangemaakt waarin je de hele schijf copiëert behalve de AUTO-folder. De RAM-disk is de werkschijf. Als er maar 1 Mb RAM is, copiëer dan de hele schijf behalve de AUTO-folder naar een lege schijf; dat wordt dan de werkschijf. Als er een harde schijf is, kan die als werkschijf gebruikt worden.

Open de folder FONT0115. Hier staan de 16 standaard fonts van TEX verpakt in 115.ARC (omdat ze zo veel beter te copiëren zijn). Start UNARC.TTP, tik

```
115.arc
```

en een `a` om alle fonts uit te pakken. Als je een Epson-compatibele printer hebt (b.v. een Star), pak dan ook 288.ARC uit in FONT0288. Verwijder de folders FONT0240 en FONT0288 en kijk onder het kopje 'Wat nu', als je niet zo'n printer hebt.

Zet (zo mogelijk) ook een editor op de werkschijf. Dat kan b.v. Tempus zijn of een tekstverwerker waarbij de wordprocessing uitgeschakeld is (dus geen haak voor WP in het menu 'Edit' van WordPlus).

1.4 Eerste stap

Start de editor en laad STAP_1.TEX of maak zelf een TEX-bestand op de werkschijf. Elke tekst die alleen de 26 letters van het alfabet (groot en klein), de cijfers en de gewone leestekens bevat, is goed voor een TEX-bestand. Breek de woorden niet af en gebruik een lege regel om alinea's te scheiden. Tik aan het eind van de tekst de 4 tekens `\bye` om TEX te laten weten dat er niets meer komt, en zet het bestand met extensie .TEX op de werkschijf.

Start het programma TEX.TTP en tik de naam van het TEX-bestand in; de extensie .TEX mag weggelaten worden. Zoals gezegd vertaalt TEX het bestand in een 'device independent' bestand en zet dit onder dezelfde naam met de extensie .DVI op schijf. (Als TEX dit niet automatisch doet, stopt en met onbegrijpelijke vragen komt, kijk dan onder het kopje 'TEX vraagt hulp'.)

Start het programma DVI_ST.TTP en tik weer de zelfde naam (zonder .DVI) om het 'zetwerk' op het scherm te krijgen. Er moet hier flink gerekend worden, dus het duurt even voor er iets te zien valt. Met de pijltjestoetsen kan je door de tekst wandelen. Probeer [Help], en tik [Esc] of [Undo] om de 'preview' op het scherm te beëindigen.

Met DVI_FXHD kan een afdruk op een Epson-compatibele printer gemaakt worden.

1.5 TEX vraagt hulp

TEX zet tijdens het werk altijd wat op het scherm. Als het goed gaat verdwijnt dat vanzelf weer; je kunt het nalezen in een bestand met de extensie .LOG. Als er iets mis gaat, stopt TEX en vraagt hulp. Een vreemd teken in het bestand, b.v. een ë, doet TEX stoppen. Je kunt TEX afschepen met [Return], vaak werkt TEX dan door. Je kunt ook TEX antwoorden met e voor exit. Dat werkt ook als TEX een bestand niet kan vinden (I can't find file ···) dank zij een bestandje E.TEX op de werkschijf.

TEX meldt 'Overfull \hbox ···' als een regel te lang wordt, omdat het laatste woord niet afgebroken kon worden. Vaak kan je TEX helpen door \- in te lassen op een geschikte plaats, b.v.

```
com\-mando
```

TEX kent nl. veel afbreekplaatsen, maar niet alle.

1.6 Fonts

Het font (tekenverzameling) dat TEX bij de eerste stap gebruikt heeft heet cmr10, afkorting voor 'computer modern roman 10 point'. Dit is het 'default' font dat TEX neemt als de gebruiker 'verzuimt' een font te specificeren. Dit font staat in de folder FONT0096 voor het scherm en in FONT0240 voor de printer.

10 punts fonts zijn geschikt voor boeken en tijdschriften. Ze zijn eigenlijk te klein voor brieven en rapporten. Daarom zullen we verder 20% grotere fonts gebruiken. (TEX maakt onderscheid tussen een 20% vergroot 10-punts font en een 12-punts font. Het laatste is iets dunner.) De volgende stap leert een paar commando's om TEX een grotere versie van het zelfde font of een ander font te laten nemen.

Een TEX commando bestaat uit een backslash \ en één of meer letters, of een \ en één teken dat geen letter is.

1.7 Stap 2

Start de editer en tik aan het begin van het TEX-bestand van stap 1:

```
\magnification 1200
```

om TEX te vertellen dat de tekens 1.2 maal groter moeten worden. Voor de preview neemt TEX dan cmr10 uit FONT0115. Kies een tekstdeel in het TEX-bestand om schuin te drukken, b.v.

`gelijkmatige verdeling`

Verander dit tekstdeel als volgt:

`{\it gelijkmatige verdeling}`

`\it` betekent italic. De werking van dit commando wordt door de accolades begrensd. (Als je de {} weglaat, wordt *alle tekst* na `\it` cursief gezet.) Op de zelfde manier kan een tekstdeel vet gezet worden met het commando `\bf` voor boldface. TEX neemt dan voor de betreffende tekstdelen cmti10 (computer modern text italic) of cmbx10 (bold extended) i.p.v. cmr10 uit FONT0115. De tabel geeft nog een paar mogelijkheden.

roman	`\rm`	cmr10
italic	`\it`	cmti10
boldface	`\bf`	cmbx10
slanted	`\sl`	cmsl10
`typewriter type`	`\tt`	cmtt10

Bekijk het resultaat met TEX.TTP en DVI_ST.TTP.

1.8 Vreemde tekens

De volgende tabel toont enkele vreemde tekens die TEX kan maken, met de bijbehorende commando's. Let op de spatie tussen de 2 c's voor de cédille. Na een commando dat eindigt met een letter mag geen letter staan, want dan denkt TEX dat die letter ook nog tot het commando behoort.

trema, umlaut	ö	`\"o`
accent grave	ò	``\`o``
accent aigu	ó	`\'o`
circonflexe	ô	`\^o`
cédille	ç	`\c c`
eszet	ß	`\ss`
hekje	#	`\#`
dollar	$	`\$`
procent	%	`\%`
ampersand	&	`\&`

Voor 'großartig' wordt `gro\ss artig` getikt met een 'loze' spatie. Maar om 'groß und großartig' te krijgen kan je

`gro\ss\ und gro\ss artig`

tikken; de backslash voor de spatie maakt er een 'echte' spatie van. Het kan ook met

`gro{\ss} und gro{\ss}artig`

ook niet echt handig. In stap 3 leren we TEX dat `\3` een ß is, zodat je dan kan volstaan met

`gro\3 und gro\3artig`

zonder echte of loze spaties (omdat `\3` niet op een letter eindigt).

De reden dat sommige tekens niet rechtstreeks ingetikt kunnen worden is dat TEX de tekens `\#$%&_^{}` reserveert voor speciale doeleinden. De volgende stap leert het gebruik van `%` voor commentaar en `&` voor een tabel, en we maken een 'macro' om een reeks toetsaanslagen te verkorten.

1.9 Stap 3

Start de editer en laad STAP_3.TEX. De regels die met een % beginnen zijn commentaar; TEX slaat ze over. Er staan wat voorbeelden van woorden met accenten en een tabel gemaakt met `\settabs`. Bekijk dit met TEX.TTP en DVI_ST.TTP.

Nu de macro voor ß. Zet ergens in het begin van STAP_3.TEX

`\def\3{\ss}`

en tik na deze definitie ergens `gro\3 und gro\3artig` of iets anders met ß om te zien of het werkt. Probeer ook de macro

`\def\mvg{Met vriendelijke groeten,}`

die `\mvg` definiëert. Na

`\def\\{$\backslash$}`

kan je met `\\`een backslash maken.

1.10 Wiskunde en Grieks

Veel tekens, Griekse letters en wiskundige symbolen, maakt TEX niet in 'text mode' maar in 'math mode', d.w.z. tussen dollartekens. Wiskundige formules kunnen in de tekstregel staan, b.v. $a^2 + b^2 = c_o^2$. Deze formule wordt met

`$ a^2+b^2=c_o^2 $`

gemaakt. Met `^` en `_` zet je super- en subscripts. Een formule tussen dubbele dollars wordt apart en gecentreerd gezet:

```
$$ y = {a/b \over \sqrt{1+c}}
    \int_0^x {p \over 1+z^2} dz $$
```

levert

$$y = \frac{a/b}{\sqrt{1+c}} \int_0^x \frac{p}{1+z^2} dz$$

Let op het gebruik van accolades en de automatische aanpassing van de hoogte van het integraalteken aan de integrand! Ook het wortelteken past automatisch bij hoogte en breedte van het argument.

Voor gewone letters in formules gebruikt TEX font cmmi10 (math italic), voor cijfers cmr10. Sub- en superscript worden in cmmi7 of cmr7 gezet en subsubscript enz. in cmmi5 of cmr5. De speciale wiskundige symbolen komen uit de drie cmsy fonts en cmex10.

In stap 3 zagen we dat { en } gemaakt worden met `$\{$` en `$\}$`. De tekens < en > krijg je met `$<$` en `$>$`. De Griekse letters $\alpha\beta\gamma\Gamma\ldots$ zijn te maken met `$\alpha\beta\gamma\Gamma$`, enz.

`$\cal ABC$` levert calligrafische hoofdletters $\mathcal{ABC}$.

1.11 Opmaak

De regellengte zet TEX op 6.5 inch en de teksthoogte op 8.9 inch, tenzij je iets anders voorschrijft. (De drukker rekent graag met inches en punten; 1 inch = 72 punt = 25.4 mm.) De commando's voor b.v. 160 mm regellengte en 250 mm teksthoogte (voor A4 papier) luiden

```
\hsize 160 true mm
\vsize 250 true mm
```

Een `magnification` moet in het TEX-bestand vooraf gaan aan de size-commando's. Bovendien zijn dan de woordjes `true` nodig. Het woord `true` in een maataanduiding voorkomt dat die maat mee vergroot wordt.

TEX bepaalt zelf de afstand tussen woorden op een regel. Al zet je 10 spaties tussen twee woorden, het effekt is gelijk aan 1 spatie. Een afstand van b.v. 5 mm tussen twee woorden bereik je met het commando

```
\hskip 5 mm
```

TEX maakt ook zelf de vertikale afstand tussen regels. Al maak je met 10 Return-aanslagen een flinke ruimte tussen twee regels, het effekt is gelijk aan 1 lege regel. Een lege ruimte van b.v. 50 mm hoog wordt met

```
\vskip 50 mm
```

gemaakt.

Een alinea kan je laten inspringen met b.v. 10 mm extra kantlijnruimte met:

```
{\leftskip 10 mm
... alineatekst ....
\par}
```

In STAP_3.TEX staat een manier om een tabel te maken.

1.12 Initex, Metafont, LATEX

Een hulpbestand bij het TEX programma is PLAIN.FMT waarin o.a. de patronen verwerkt zijn die TEX gebruikt om woorden af te breken. Het bestand PLAIN.FMT op schijf B118 bevat Nederlandse afbreekpatronen, gemaakt door de Universiteit van Utrecht. Met onze taal werken die vrijwel foutloos, maar met een vreemde taal natuurlijk niet. INITEX.TTP dient o.a. om PLAIN.FMT te maken met andere afbreekpatronen.

Bij TEX wordt een selectie van fonts meegeleverd in vergrotingen die in stappen van 20% oplopen, dus voor magnification 1000, 1200, 1440, 1728, enz. Met METAFONT zijn fonts te maken in iedere gewenste grootte. Het heeft geen zin om de opdracht `\magnification 2000` te geven als je niet eerst de bijbehorende fonts hebt aangemaakt!

Er zijn vele macro's in omloop, b.v. om kopjes te maken, tekst in twee kolommen te zetten, enz. De meest bekende macroverzameling heet LATEX, naar de maker Leslie Lamport.

1.13 Wat nu?

In deze inleiding heb je kennis gemaakt met TEX. In vervolghoofdstukken zal dieper ingegaan worden op dit zetsysteem. De oude versie staat op de PD schijfjes B74–83. Daaruit is schijf B118 samengesteld. De nieuwe versie van TEX staat op B98–105. Deze is sneller, heeft mooiere fonts en kan afbreekpatronen voor meer talen tegelijk gebruiken. Zonder harde schijf is deze versie echter niet te installeren. Beide versies bevatten drivers voor vele printers, INITEX met afbreekpatronen voor Duits en Engels, METAFONT en LATEX.

1.14 Literatuur

Het standaardwerk is van de maker van TEX: Donald E. Knuth, *The TEXbook* (Addison-Wesley), een goudmijn voor gevorderde TEXers. Ruim voldoende voor gewoon gebruik. Meer geschikt om iets op te zoeken is: Norbert Schwarz, *Inleiding TEX* (Addison-Wesley), omdat dit boek een lijst van TEX-commando's bevat.

– – * – –

2 Boxen

Na de inleiding tot TEX in het vorige hoofdstuk, gaan we in dit hoofdstuk dieper in op het zetten van tekstblokken op een pagina. Als voorbeeld zetten we een brief en een tekst in twee kolommen.

2.1 Briefhoofd

De bedoeling is een bestand BRIEF.TEX te maken met een briefkader dat je eigen briefhoofd, de plaats voor het adres, de datum enz. bevat. Als je dan een brief aan P. Puk wilt schrijven, zet je BRIEF.TEX in de editor, je vult het adres in, schrijft de brieftekst en zet het bestand als PUK.TEX weer op de werkschijf. Alle namen en maten in dit verhaal zijn willekeurig.

BRIEF.TEX begint met een paar instellingen:

```
\magnification 1200
\vsize 250 true mm
\hsize 165 true mm
\nopagenumbers
\parindent 0 mm
```

De eerste drie commando's kennen we al uit de Inleiding (Stap 1 en Opmaak). Het vierde commando onderdrukt paginanummers; als je die in een lange brief toch wilt hebben, zet je een procentteken voor dit commando (Stap 3). Het laatste commando zorgt dat TEX niet bij elke nieuwe alinea inspringt (\parindent 0 werkt niet, TEX wil een maat zien).

Vervolgens het briefhoofd, voorlopig in de linker bovenhoek. Andere plaatsen komen later aan de orde (onder 'Meer boxen').

```
% briefhoofd
   {\obeylines
   { \bf A. Texnicus }
   { \sevenrm Tolstr.\ 88
      1234 TT Assen
      Tel 099-99999 }}
```

Het commando `\obeylines` maakt dat TEX niet zelf de tekst gaat opmaken in regels van gelijke lengte. Zonder dit commando zouden naam, straat, enz. niet onder maar achter elkaar gezet worden. De werking van `\obeylines` wordt begrensd door de eerste en laatste accolade. De naam is vet gezet met `\bf`, zoals we in Stap 2 gezien hebben. Adres enz. zetten we in een klein lettertje. Met `\sevenrm` wordt het cmr7