

Baskerville

The Annals of the UK T_EX Users Group
ISSN 1354-5930

Editor: Sebastian Rahtz

Vol. 8 No. 6
May 1999

Baskerville is set in Monotype Baskerville, with Computer Modern Typewriter for literal text. Editing, production and distribution are undertaken by members of the Committee. Contributions and correspondence should be sent to Baskerville@TeX.Ac.Uk

Contents

Editorial	1
Chairman's Report	2
Minutes of the UK-T _U G AGM 21st September 1998 ...	2
Accounts for 1997-1998	4
The 19th annual T _E X Users Group Meeting	5
* * * * *	
L^AT_EX for the rest of us (part 2)	
Now to maths	8
Ups and downs	9
More simple operators	10
Mâthématicâl accents	11
We have many ways of doing this... ..	11
Prime	12
Above and beyond	12
More uses of sub- and super-scripts	13
Display, text, script and scriptscript	13
Other operators	14
Relations	15
Tablets from the mountain	16
Tables	16
A worked example	16
Classic example	17
Some variations	18
More tabular esoteria	18
Boxing (and paragraphs)	19
Floating away	21
Footnotes	21
Marginal returns	21
Tables and Figures	22
To whom it refers	22
* * * * *	
UK-T _U G Information	24

Editorial

This is, sadly, my last *Baskerville*. I have worked on the journal since 1993, and although I stopped being full-time editor in 1996, I remained in overall charge, *faut de mieux*. But this time I really retire, since I am no longer a member of the UK-T_UG committee, and have other tasks ahead.

And so, farewell then,
Baskerville.
You were a pain
Six times a year
Was there any gain?

s rahtz aged 43 3/4

Yes, of course there was gain! I learned a lot about T_EX over the years — having to garner articles for a regular publication is an excellent way of finding out what is “going on”, as is having to write space-fillers yourself.

It was the best of times, the worst of times... I want to thank in particular Robin Fairbairns and Malcolm Clark for their unfailing work throughout; and my colleagues from NTG in the Netherlands, who provided some of the most stimulating material. I am particularly proud to have brought you Rosemary Bailey's mathematical typesetting series, Malcolm's *Gleanings*, the UK-T_UG FAQ, and (most recently) *L^AT_EX for the rest of us*. I hope the editors to come can continue to tap the rich vein of creativity which is so marked a feature of the T_EX world.

TUG 2000

We are very glad to announce that the T_EX Users Group has accepted a proposal from UK-T_UG to host the TUG Annual General Meeting and Conference for the year 2000. This will take place from Saturday 12th to Friday 18th August.

The whole conference will be held in Wadham College. This is a traditional Oxford college in the centre of the town, riddled with ancient buildings and beautiful gardens.

We propose the following schedule:

Sat	12	eve.	arrival for overseas delegates
Sun	13	a.m.	free tutorial
			walking tour of Oxford
			arrival of delegates
Sun	13	p.m.	full opening Conference session
Sun	13	eve.	Reception in College gardens
Mon	14	a.m.	full Conference session
Mon	14	p.m.	free for tourism
Mon	14	eve.	optional Shakespeare play in College gardens
Tue	15	a.m.	full Conference session
Tue	15	p.m.	full Conference session
Tue	15	eve.	reception and banquet
Wed	16	a.m.	full Conference session
Wed	16	p.m.	half Conference session, completion by 3.30p.m.
Thu	17		Tutorials
Fri	18		Tutorials

The conference chairman will be Sebastian Rahtz (Oxford University), and the head of local organisation will be Kim Roberts (OUP). Steering and programme committees have already been set up; anyone else who is interested in helping should contact Sebastian.

Sebastian Rahtz

Chairman's Report

It was a great honour to be elected as Chairman of UK-TUG this time last year, for I have been associated with the User Group ever since its inception. Having now completed my first year in office, it is both a duty and a pleasure to report on what has transpired during that year.

From the point of view of membership, we continue to grow, albeit rather slowly. We have gained a number of individual members, and at least one (possibly more) Institutional member (the London Mathematical Society springs to mind), the latter by courtesy of my old friend and colleague Wilf Hodges with whom I served at Bedford College many years ago. Our finances are also in a healthy state, with around £15 000 in the bank.

In terms of services to members, I think we have made steady progress: *Baskerville* has come out on schedule, and those meetings which we have organised have been both well attended and well received. We have purchased and distributed licences for WinEdt, and of course we have prepared, burned, pressed and distributed the T_EX Live CDs.

So far, so good, but all has not been quite as rosy as the above might suggest: we have wasted an *inordinate* amount of time in Committee, debating ('arguing' might be a more accurate term) various minutiae concerning (for example) the exact interpretation to be afforded to a clause in the Constitution. Doubtless all members of the Committee (including myself) are to a greater or lesser extent responsible for this sad waste of time and resources, but matters reached a head at the last Committee meeting before this AGM, and the Committee passed (nem. con.) a vote of no confidence in the then-Secretary, Jonathan Fine. Jonathan did the decent thing by tendering his resignation, which he then backed up with a letter of resignation as Secretary. He has also indicated to the Committee that, with effect from this AGM, he will be resigning from the Committee as well. Although I believe that Jonathan's resignation is for the best, and that the Committee next year will be able to make far more rapid progress through each agenda as a result, we do owe a considerable debt of gratitude to Jonathan for his organisational skills when it comes to meetings, and on behalf of the Committee I would like to publically record our thanks to him for the work he has put into organising meetings (including this one) during the past year.

One real problem which besets us (and for which no simple solution can be found) is the unchangeable nature of the Constitution. This document, which of necessity governs every action the Committee makes, is worded in such a way that it can be changed only by the membership in general meeting. To achieve such a change, a quorum of forty must be achieved. And despite the increasing popularity of our meetings, and the increase in membership numbers, we seem unfortunately unable ever to achieve that quorum. As a result, we are stuck with a Constitution which none of the Committee believe is workable, and which continues to thwart us whenever we try to do something which its creators had failed to envisage. As just one example of its weaknesses, 2/3 of the Committee are required to stand down each year; yet the quorum for the Committee is five. Had it not been for the

extreme good fortune that a single member offered himself for election to the Committee with effect from this meeting, the Committee would have consisted of only four members, and would have therefore been inquorate. Being inquorate, it would have been unable to co-opt, and therefore it would have remained inquorate with absolutely no way of rectifying the situation. And an inquorate Committee can make no decisions, nor take any action involving (for example) the finances of UK-TUG. This would essentially have spelled the end of UK-TUG, for without a quorate Committee, nothing further could have taken place.

I had hoped that at this AGM, we might achieve the necessary quorum to change the Constitution, which is why I proposed the motion that you will see in the circulated papers. However, all the evidence suggests that we will not achieve the necessary figure of forty members present, and we will therefore be forced to live with a virtually unworkable Constitution for at least another year. This situation is so problematic that serious consideration must be given to a far more radical proposal: that we wind up UK-TUG as it is currently constituted, and immediately form a new organisation (which might just happen to be called the UK T_EX Users' Group) with a more rational Constitution and with inbuilt safeguards to effect changes to the Constitution without requiring an unachievable quorum. It is worth noting, in passing, that TUG itself has just such a Constitution, and the powers to change that Constitution are vested in the Board of Directors rather than in the membership at large.

I would welcome comments from as many members as possible as to whether this idea of winding up, and immediately reforming, UK-TUG might be acceptable to them.

Finally I would like to record my own very sincere thanks to those long-standing Committee members who are leaving the Committee with effect from this meeting. It would be invidious to name names, but I can say with complete confidence and certainty that they will be sadly missed on the Committee, and I very much hope that they will rejoin it in the not too distant future. Working with them has been a great pleasure, and even though four members remain, we lose six and gain one: I hope that at least some of those retiring this year will allow themselves to be co-opted, for the Committee genuinely needs their skills and experience.

Philip Taylor

Minutes of the UK-TUG AGM 21st September 1998

The meeting commenced at 16.10. Attended by 27 members including the Chair. Apologies were received from D. Rhead.

Approval of minutes from last AGM

Some changes were requested:

- Point 12: Jonathan Fine raised question regarding UK-TUG finance Robin Fairbairns answered.
- Point 11: Delete names involved in discussion.

The Minutes accepted by meeting with above changes.

Matters arising

There were no matters arising.

Chairman's Report

The Chairman read briefly from his report. Public thanks was given to Jonathan Fine for his input regarding meetings. He spoke on the constraints of the constitution and the possibility of dissolving and immediately reforming the group; this he asked for views on.

Treasurer's report

This year's — but not last year's — accounts are being inspected at the moment. One meeting made a profit and one made a loss. Jonathan Fine asked about balance for L^AT_EX₃ and the treasurer referred him to the notes on the back of the accounts.

Malcolm Clark asked if the balance was felt to be stable and we were told it was. Roy Everett asked for an explanation of the difference between an audit and an inspection of the accounts.

The treasurer proposed that the membership fees stay the same for the coming year. After general discussion between the members it was agreed that there would be a slight increase to £22 for single membership and £110 for institutional membership.

Committee membership

Peter Abbott stands under a special clause; Kaveh Bazargan stands down; Malcolm Clark stands down; Roy Everett continues; Robin Fairbairns stands down; Jonathan Fine stands down; David Hardy continues; Sebastian Rahtz stands down; Kim Roberts continues; Philip Taylor continues as Chairman; Mark Wooding stands down; Dominik Wujastyk continues.

One nomination was received from Phil Molyneux and so he was elected.

The Chair asked for volunteers to be co-opted by the new committee. The following showed interest: Kaveh Bazargan, Malcolm Clark, James Foster and John Palmer.

Report on Baskerville

Sebastian Rahtz apologised for the erratic production of editions this year. The year was to be finished off with the second part of Malcolm Clark's series. A new editor and production editor will be required for next year and the Chair asked for volunteers. Our thanks were given to Sebastian and Robin for all their work on past editions.

Motions

Details of motions proposed had been sent to all members.

JF1 This was not considered

JF2 Robin Fairbairns proposed an opt-in rather than an opt-out system. Jonathan took note of the proposal as a friendly amendment. Dick Nickalls seconded the motion. The motion was carried with a vote of 15 For and 0 Against with 7 Abstentions.

JF3 This was withdrawn

JF4 Jonathan read from the minutes of the June 98 meeting. Roy Everett seconded the motion.

Malcolm Clark clarified why he had stated the bursary award to be a shambles; he explained that this was due to the late award of the bursary and not due to the issues addressed by Jonathan's statement. Carol Hewlett spoke against the motion, feeling that it was a matter for the committee to decide and that the motion was not worded correctly.

Dominik Wujastyk reported that Jonathan had sent letters that were not agreed by the committee, which helped lead to the shambles. Also that the motion was obsolete as at the last committee meeting a working party was elected to look at the problems of awarding a bursary.

Dick Nickalls showed support for the motion.

The motion was lost with a vote of 6 For and 12 Against with 4 Abstentions

JF5 This was withdrawn

PT1 This was not considered

Future meetings

Sebastian Rahtz asked if we were to host TUG 2000.

The general opinion was that members would like to see TUG 2000 hosted in the UK. The incoming committee are to make the final decision.

Robin Fairbairns said that we still have suggestions from members for future meetings in 1999. But it was felt that if we were to go ahead with TUG 2000 there would be some curtailment of meetings in 1999.

Any other business

The committee had decided by single transferrable vote that the winner of this year's Bibby Cartoon was Sebastian Rahtz and so the cartoon was presented.

Sebastian appealed for help with the T_EX Live 4 CD-ROM.

Carol Hewlett asked about the random distribution of TUG CDs. We were informed that a mistake had been made when TUG was sent out and an insert was in the new TUG-Boat explaining this.

The meeting closed at 17.46pm

Kim Roberts

Accounts for 1997–1998**Income and Expenditure for the period 01/08/1997 to 31/07/1998**

Income		Expenditure	
TUG		Baskerville costs	1322.06
	1997 130.00	TUG India	700.00
	1998 3322.50	Committee Meetings	
UKTUG		97 AGM	559.59
	1997 277.00	07/11/1997	148.45
	1998 4091.00	17/12/1997	354.31
Handling	324.50	23/04/1998	149.80
Invoice Charges	70.00	23/06/1998	110.05
Book sales	546.51	Book purchases	704.23
4All \TeX sales	25.00	4All \TeX purchases	652.99
La \TeX 3 fund	15755.91	La \TeX 3 payments	11369.33
Bursary	61.02	Bursary outgoings	620.00
Bank Interest	174.14	Bank Charges	2.00
\TeX Live CDs	5207.07	\TeX Live CDs	4416.83
EDMAC manual	40.00		
Meetings		Meetings	
	26/11/1997 550.00		26/11/1997 1291.06
	18/05/1998 670.00		18/05/1998 204.09
		Treasurer Expenses	
		Telephone	306.09
		Membership	159.87
		Book costs	63.51
		CDs discs etc	72.84
		Other	11.94
		Secretary Expenses	64.38
		Group Purchases	
		Hardware	1648.50
		Software	310.00
		Miscellaneous	164.90
		Excess Income	5837.83
Total	31244.65	Total	31244.65

Balance sheet

	Opening	Closing
TUG	1527.10	4979.60
Em \TeX /Oz \TeX	480.00	0.00
Discs	521.86	521.86
Books	316.71	281.59
4All \TeX CDs	1254.88	626.89
La \TeX 3	55140.42	127.00
Bursary	81.00	0.00
\TeX Live CDs	280.56	622.30
T Shirts	238.75	0.00
EDMAC Manuals	111.00	151.00
Holding A/C	23.00	0.00
General Funds	14544.92	14810.88
Total	74520.20	22121.12
Represented by		
Current A/C		7819.58
High Interest A/C		14301.54
Total Value		22121.12

Notes to the Accounts

The Account Heads EmT_EX/OzT_EX and T Shirts have shown no activity during 1997/98 and will not appear in 1998/99.

Bursary Income for 1996/97 was not used, and so appears as an opening balance.

During 1997/98 L^AT_EX3 funds were moved to special L^AT_EX3 accounts and UK-TUG transfers any balances accrued to the L^AT_EX3 account at periodic intervals. £500 was paid from UK-TUG accounts following the division of funds, being the calculated interest on the L^AT_EX3 funds whilst there were part of UK-TUG funds.

TUG is due £657.10 for 1997 and this is still held by UK-TUG. As at 31/07/98 £322.50 (1998 income) held by UK-TUG will be due to TUG on 01/01/1999.

Peter Abbott

The 19th annual T_EX Users Group Meeting

The 19th TUG meeting took place in Toruń, Poland, in August of this year. The following is the impression of the authors who tried to attend all the talks. Unfortunately a couple of the talks were not attended by either of the authors. Apologies to those speakers.

Daniel Taupin described his L^AT_EX to RTF converter, *ltx2rtf*. His problem is that he wants to transfer L^AT_EX documents to colleagues who may not have L^AT_EX, and not even a PostScript printer. His conclusion is that the great majority have MicroSoft Word installed, and therefore that should form the basis for the format in which the files are transferred. Since RTF can be read in directly by Word and by other word processors, and the specifications are published, it makes a logical choice for the conversion process. Of course the limitations of Word cannot be overcome. For example equations are saved and placed as bitmaps. Daniel's colourful personality and his vociferous comments at most presentations livened up the proceedings enormously.

RWD Nickalls is a consultant anaesthetist in a hospital. He uses L^AT_EX in conjunction with GNUPLOT to display and print charts in the operating theatre. The use of ASCII input for both GNUPLOT and L^AT_EX makes for a reliable system.

Marcin Woliński presented a L^AT_EX(2_ε) package for 'pretty-printing' program codes such as Pascal and Prolog directly through T_EX. The results are quite amazing, with the code printing with the usual conventions of bold, italic, correct indentation, and cleverly using maths mode to position mathematical constructs. It is a good use of the unique nature of T_EX, being a text processor and a programming language in one.

METAPOST seems at last to have opened the door to the untapped power of Knuth's METAFONT. METAPOST uses only a small subset of PostScript operators. In particular, there is no access to the PostScript pattern operators through METAPOST. Piotr Bolek's solution is to allow the user to use routines in METAPOST to define patterns, but then to post-process the file using a PERL script to insert the correct code according to PostScript level 2. His *mpattern* package consists of the METAPOST code to define the patterns, and the PERL script for post-processing.

Taco Hoekwater, of Kluwer Academic, laid out the procedure to generate PostScript Type 1 fonts from METAFONT sources. The major incentive for this work was to allow the inclusion of these fonts in PDF files generated by PdfT_EX (Adobe Acrobat is not good at viewing bitmap

fonts). The translation is not straightforward. For one thing, METAFONT has a richer set of primitives for describing font shapes and characteristics than the definitions of Type 1 fonts. Secondly, METAFONT allows the use of the equivalent of PostScript stroked paths, whereas Adobe Type 1 fonts can only be described as outlines. A reliable translator would open the way to more widespread use of METAFONT, which is undoubtedly the most powerful tool for generating typefaces. At the moment the procedure is quite long, as several manual steps are needed. Taco's paper, incidentally, is an excellent introduction to METAFONT, and to Type 1 fonts.

Using his Calculator demo, Hans Hagen posed a philosophical question which emphasized the changing world of document production. 'calculator.pdf' (available from <http://www.ntg.nl/context>) is a PDF file, viewable in Adobe Acrobat Reader, which looks and behaves like a scientific calculator. So is it a program or is it a document? Well, it is a kind of intelligent document, which goes one step further than having hypertext links. Its usefulness may be trivial, but the method used to produce it is extremely significant. The 'document' was produced by allowing T_EX to embed METAPOST and JavaScript code, and produce one composite PDF file. METAPOST produces graphics parametrically on the fly, JavaScript does the calculations behind the buttons, and T_EX does the overall typesetting and management. The whole process is processed by Hans's powerful Context package, which uses PdfT_EX to produce the output directly (rather than taking the DVI-PostScript-Distiller route). The Calculator demo is perhaps the single most powerful example of how relevant T_EX still is in the world of electronic publishing.

Another little gem from Hans Hagen is the idea of 'visual debugging'. T_EX has a subtle, and sometimes unconventional way of using vertical and horizontal space. When writing macros, a programmer often has to get into the mind of T_EX to work out why certain spaces have appeared in strange places, or why a paragraph is indented, etc. This can take quite a while to work out, and the answer is usually obvious in hindsight! It turns out that by carefully redefining a lot of T_EX's placement commands, such as `\hfil`, `\vfil`, `\kern`, `penalty` etc, T_EX can be persuaded to show graphically all spaces, fills, glue, and boxes. Hans has designed different solid and dashed lines to signify each type of box or glue placed by T_EX, thus providing a visual picture of what he calls 'an endoscopic view of T_EX's stomach'.

Bogusław Jackowski and his colleagues in BOP s.c. are masters of PostScript, especially when applied to \TeX . In the first of his most entertaining and gripping deliveries, Bogusław described his CEP or COP packages for compressing PostScript files. Most people who deal with PostScript files are aware of the enormous file sizes. What is not well known is that it is possible to reduce these files substantially by using PostScript's own compression algorithms. Most applications do not bother to use this compression when generating PS files. The most efficient compression is achieved when the type of data in the PS file is known *a priori*. Depending on the type of data, one of the compression programs (CEP and COP) is chosen.

In his second talk, Bogusław described a suite of unusual PostScript tools for use with \TeX . `Tiff2ps` is a program that converts Tiff files to PostScript files, the program itself having been written in PostScript. `Pf2afm` tries to extract AFM (Adobe Font Metrics) files from PFM (Printer Font Metrics) files. AFM files are needed for \TeX users, normally converted to TFM. PFM files are subsets of AFM files, and are used in Windows. Another utility, `Ttf2pf` converts TrueType fonts to AFM and Type 42 (PostScript's equivalent of TrueType). Finally, `Colormap` is a \TeX macro package that makes simple colour and grey scale changes to EPS files that are called by the \TeX file, without modifying the EPS files themselves. The two papers have a lot of useful material regarding PostScript and fonts, and are recommended reading. One thing that comes through is the power of the freeware Ghostscript program which is used in all this work.

Hàn Thê Thành, author of Pdf \TeX was not present, but his paper was read by Jiří Zlatuska. His idea is to improve the look of text set on a narrow measure by horizontally scaling the text in Acrobat after \TeX has finished its work, thus reducing the large variation in word spacing. This scaling has to be limited to only a small percentage, as otherwise the different shapes of the letters become noticeable. Also, the differing stem widths make the letters appear lighter or bolder, a generally undesirable effect.

Miroslava Misáková takes this idea of postprocessing one ingenious step further. (In fact she got the idea by looking at Gutenberg's 42-line Bible!) She solves the problem of the differing stem width by regenerating the font using METAFONT, and varying the width of the font but keeping the stem width constant—something that probably can only be done with METAFONT. This makes for a more uniform looking block of text. The procedure is first to set a paragraph by allowing the right margin to vary by some 5%. The right margin is therefore ragged, but the word spacing is more uniform. Using \TeX macros and `\specials`, marks are embedded in the DVI file, signifying the positions of the ends of the lines. A Perl script then interrogates the DVI file, and for each line, works out the correct font to use for setting. Usually, it would look at a set of 10 fonts already generated, and choose the closest match. The results presented are most impressive. Even when adjoining lines vary in letter width by some 7%, the text looks uniform, and is comfortable to read. Of course the ideal solution would be to add this capability of variable font width to \TeX as one of its parameters, but that becomes a more complicated procedure.

Richard Kinch's contribution was a new maths font, Belleek, that he put into the public domain. The font was produced manually, using Fontographer, and designed to work with Times as the body text. His conclusion from the experience of creating this font is that METAFONT is the best tool for font creation, but that the main stumbling block for its more popular use is that it creates bitmap output, as opposed to outlines. He discussed some methods for getting outlines from METAFONT.

The presentation that created the most heated discussion was kept for the last day. This was the joint presentation by the Dutch contingent (NTG), who had arrived in an orange van at the conference. The Dutch team, led by Hans Hagen, presented some radical ideas. These ideas were a culmination of lively discussions among Dutch \TeX users during the months prior to the conference, and led to the 'NTG \TeX future working group'. They felt that they had to share these ideas with the \TeX community. Here are some of the highlights of their views:

- It is proposed that the three projects currently under way to extend the capabilities of \TeX , namely Omega, Pdf \TeX and e- \TeX , should merge into one project. NTS (the New Typesetting System) is a project to rewrite \TeX in Java, with a guaranteed backward compatibility with \TeX 82, at least for the first 5 years. The Dutch team feel that this is a mistake, and that backward compatibility should be discarded.
- The group feels strongly that current distribution and implementation methods for \TeX are too complicated, incompatible, and involve too many files. A typical complete \TeX system can comprise some 30,000 files, which the Dutch group feel should be reduced by a factor of 100. This would be achieved by setting up an intelligent central database which could be queried, and which would install only the necessary files according to a user's specifications. This database could have access to CTAN to obtain any files not available locally. It is also proposed that CTAN be more closely regulated with files and packages being put up with more information accompanying them.
- Another suggestion is to extend the capabilities of \TeX to allow the production of electronic documentation. These include more support for colour and hypertext linking within \TeX , rather than relying on `\specials` to produce these capabilities at the back end. To allow the direct inclusion of METAPOST graphics, a set of `\specials` is proposed which allows the use a larger subset of PostScript commands than is used in METAPOST.
- Fonts have always caused problems for all \TeX users apart from real experts. The NTG \TeX working group recommend an overhaul of the font system, including replacing the way TFM and VF formats are defined, separation of ligatures and kerning information from the character metrics of fonts, and the facility for direct output of outline fonts from METAFONT. (The latter echoes the wishes of Richard Kinch.) The group also call for a friendly user interface for mathematics, and a general cleaning up of the \TeX syntax.
- The final proposal is a standard for the specification of

`\specials` across all systems. The proposal is based on that of Nelson Beebe, and the aim is to make T_EX truly portable.

Laurence Finston presented what should have been an extremely interesting paper on the generation of concordances using a combination of T_EX & Lisp. This paper was rather difficult to understand by reading it, so we looked forward to Laurence's oral presentation. Unfortunately, he chose to *read* his paper rather than presenting it *extempore*, as a result of which his audience (including your humble scribes) started to sneak out after the first five minutes. We learn that M. Taupin asked at the end if Laurence considered TUG'98 merely a practice run for his dissertation. . . !

Janusz Nowacki, speaking through a translator, presented his work on Antykwa Toruńska, a modern digital adaptation of a traditional Polish font. Janusz commenced his talk by recounting the motivation for his work: the simple fact that since the DTP/EP revolution, no Polish text had been produced using a Polish font! Despite the problems of translation, Janusz kept his audience enthralled, and his talk was one of the most interesting and stimulating delivered. Antykwa Toruńska in its current version originates as scanned bitmaps; it is then post-processed in Corel Draw and fine-tuned in Fontographer. It is available in regular, italic and bold versions.

Karel Skoupý presented a report on the current status of NTS ("a New Typesetting System"). This project, which was first mooted in 1992 and which has only recently sprung into life, involves a complete re-implementation of T_EX using Sun's JAVA programming language. Rather than simply being a re-implementation, however, the project's real aim is to completely restructure T_EX/NTS internally so as to make the inter-module interfaces far clearer and to provide a framework within which current (for example e-T_EX, Omega and pdfT_EX) and future developments can be easily and straightforwardly embedded. Karel emphasized that the choice of JAVA had many benefits, of which guaranteed portability and network awareness were the greatest, but also had its drawbacks (for example, the lack of rigorous compile-time type-checking for derived types).

Although the majority of papers took place on schedule, the unanticipated absence of John Plaice (originator of the Omega project) caused some gaps in the schedule. These were filled by *extempore* talks by Kaveh Bazargan ("Can T_EX

create Farsi scripts?"), Daniel Taupin ("L^AT_EX2HTML for the i'x86 family of operating systems), and Taco Hoekwater ("DVIVIEW : a reference previewer").

The four days surrounding the conference proper were occupied by tutorials. Hans Hagen had bravely volunteered to give a split two-day tutorial entitled "Actually making an electronic document"; Philip Taylor & Jiří Zlatuška offered a preview of a tutorial originally prepared for IFIP'98 ("Document design, document markup, and the converging worlds of computer typesetting and electronic publishing"); and Bogusław Jackowski offered a one-day tutorial on "T_EX and PostScript integration). All three were well attended, but the Taylor/Zlatuska tutorial rather fell apart when the participants agreed that it was either too advanced or too basic but certainly not the right level. . . Hans' and Bogusław's tutorials, on the other hand, both proved very successful.

Of course, a conference is considerably more than a series of tutorials and talks: the social aspects are in many ways as important if not more so. From this perspective TUG'98 was a great success: the organizers had carefully planned the social activities, and the gala concert and banquet at Artus' Hall ("King Arthur's Hall") was one of the high spots. Equally memorable was the outing to the mediæval castle at Golub-Dobrzyn, where the guest were greeted by the sight of an ancient and wizened serf, clad in chain mail, slowly roasting a pig on the spit. There was limitless beer to accompany this feast, and a guided tour of the castle; guests of a weak disposition were probably somewhat discomfitted to learn of the mildly unpleasant ways in which "guests" of an earlier age were accommodated: simple pleasures such as being tied down under a slow but continuous water drip, or having salt rubbed into fresh wounds, were among the many ways in which their hosts passed away the weary hours. . . Despite (or perhaps because of) this, the sight of the pig being slowly roasted conjured up in several minds an analogous scene in which the central participant was rather more anthropoid than porcine. . . Simpler pleasures were also to be found every evening, when beer, vodka, guitars, whistle and voice mingled together to create an atmosphere of great conviviality and warmth. All in all we were impressed by the genuine hospitality that we received by from the local organizing party.

Kaveh Bazargan and Philip Taylor

L^AT_EX For The Rest Of Us

by
Malcolm Clark

Chapter 5. Now to maths

In many respects this is where L^AT_EX gets to be lots more fun. There is something extraordinarily satisfying about being able to set mathematical formulæ. L^AT_EX permits this both within the text (*text* style) and displayed on lines by themselves (*display* style). The same expression can be given in text or display style, but when processed by L^AT_EX, there will be a few stylistic changes which are taken care of automatically.

All mathematical formulæ are enclosed in their own environment. There are several different mathematical environments. A few examples soon convey most of the ideas and conventions.

```
\begin{math}
x = y + z - 10
\end{math}
```

will result in: $x = y + z - 10$; where L^AT_EX sets the alphabetic characters in italics, and copes quite gamely with the =, + and -. Note that here we have a true minus sign (which looks different from en-dash, em-dash and the hyphen). L^AT_EX also takes care of all the spacing. It matters not one whit whether you type

```
\begin{math}
x=y+z-10
\end{math}
```

or even

```
\begin{math} x= y +z - 1 0 \end{math}
```

L^AT_EX will produce the same thing, and generally correctly: $x = y + z - 10$.

The `\begin{math}` and `\end{math}` environment delimiters are used to signify the ‘text style’, that is to say, maths within text. For a displayed equation (‘display style’), we use `\begin{displaymath}` and `\end{displaymath}`, so that

```
\begin{displaymath}
x=y+z-10
\end{displaymath}
```

gives us

$$x = y + z - 10$$

In very general terms, you would tend to use the text style for fairly simple relationships, and the display style for more complicated expressions. L^AT_EX will break a text style equation between lines, if it requires to do so (according to some inbuilt rules which favour breaks at particular places within a

text style equation), but it will make no attempt at all to break a display style equation, even if it is far too long for the page width. It will report an overfull box in the log file.

Since `\begin{math}` is rather a lot to have to type, L^AT_EX allows you to abbreviate this to `\(`, while the `\end{math}` may be shortened to `\)`. Both these sets of delimiters could also be replaced by the `$` delimiter. This is a fundamental T_EX instruction, which Lamport deliberately replaced. It is especially intriguing that in T_EX the same symbol could be used to both enter and leave an environment. We could therefore have written the text style equations as

```
$x=y+z-10$
```

On the one hand, there are some advantages to having a different instruction initiate and terminate the maths segments; on the other hand, I find that `\(` soon gets lost in a complicated expression, while dollar signs are quite prominent, and easy to pick out. I have very rarely, if ever, seen anyone write out `\begin{math}` and `\end{math}`, except in a tutorial text like this.

In display style, we could substitute the `\[` and `\]` delimiters for the `displaymath` environment. Again the same sorts of arguments apply, but perhaps more strongly. As a leftover from T_EX, and an extension of the use of `$`, you may see `$$` used to limit display maths. This is to be deprecated firmly, although it may appear to work in many situations. As Lamport [2] says (p233) ‘T_EX’s `$$` does not work properly [in L^AT_EX]’. In any case, we shall see that retaining the L^AT_EX forms in display style confers certain advantages which the bald T_EX form of `$$` could not have provided. In the rest of this chapter, we will tend to use the ‘verbose’ forms rather than the terse ones, mainly as emphasis, but also to be consistent with all the other sorts of L^AT_EX environments.

But do note that there are a very few occasions when the `$` form is actually to be preferred (we’ll look at this later).

QUESTION 5.1 *Can you mix `\begin{math}` and `\)`? or any of the other equivalent environment delimiters?*

This chapter is about mathematics. Assume that almost all expressions are surrounded with some environment defining instructions, whether for *text* or *display* style maths. Where there is likely to be any ambiguity, the environment will be

defined explicitly. Note that mathematical instructions do not work outside a maths environment.

There are a host of instructions which are defined for maths environments. In particular, you may access the Greek alphabet (see Table 1). Note that this is not a Greek font as such, in the sense that you would not use it in order to set Greek text. It is best considered as just another set of symbols. The general form of access is much as you would expect: a Greek α is obtained by `\alpha`, a γ by `\gamma` – the ‘English’ name for the letter, preceded by the backslash. To obtain upper case Greek, use `\Gamma` and so on (Table 2). Since some upper case Greek letters are the same as their Roman equivalents, there are no entries for these overlaps. Similarly, there is no lower case omicron.

There is an extension to the Greek fonts which may sometimes be useful. You can obtain ‘italic’ Greek (an oxymoron if ever there was one) by `\mathit{\Lambda}` – Λ – and so on. Note that upper case Greek is ‘upright’, while lower case is ‘slanted’ or ‘italic’.

QUESTION 5.2 *Why is upper case Greek upright, and lower case slanted or italicised?*

There is also another alphabet available – the *calligraphic* alphabet. Again it applies only to upper case letters. Preceding an upper case letter by `\mathcal` gives the calligraphic equivalent, so that `\mathcal{AMS}` gives \mathcal{AMS} . Using a lower case letter with `\mathcal` produces the weirdest results. You cannot obtain calligraphic Greek at all.

QUESTION 5.3 *Generate a list of the italic Greek upper case letters; and another one with the calligraphic characters.*

Ups and downs

One of the things which mathematicians do is to use subscripts and super-scripts. These are so fundamental that L^AT_EX makes them very easy. We employ two keyboard symbols to trigger these events: the caret or circumflex, `^`, signals that the next character (or group of characters) is to be super-scripted, while the underline symbol, `_`, is used for subscripts. The underline (or underscore or low bar) symbol is different from all the other dashes, hyphens and minuses we have met so far.

Let us try this out:

```
\begin{displaymath}
10x^3 - 3x^2 + 4x - 2 = 0
\end{displaymath}
\begin{displaymath}
(x) = g(x) + bf^{-1}(x)
\end{displaymath}
\[ x_{n+1} = 1 - ax_n^2 + bz_n \]
```

You may find it easier if you space out your formulæ into ‘logical’ groupings, just letting L^AT_EX take care of the spacing. Jamming all the symbols together leads to problems. The result of these last three examples should be:

$$10x^3 - 3x^2 + 4x - 2 = 0$$

$$(x) = g(x) + bf^{-1}(x)$$

$$x_{n+1} = 1 - ax_n^2 + bz_n$$

Notice that in order to sub- or super-script more than one character, we group the relevant symbols within braces.

You can sub- or super-script to your heart’s content, and L^AT_EX will reduce the size of the first level; any further sub- or super-scripting will be at a yet smaller size. This is the limit, and however deeply you sub- or super-script, the symbols will get no smaller. The actual sizes depend on the document styles and options you have chosen. These reductions in size are set for you, but they *could* be changed to some other value.

`a^{b^{c^{d^e}}}`

gives an expression where c , d and e are all the same point size: $a^{b^{c^{d^e}}}$. They are all super-scripted correctly, but don’t become any smaller. Do note that the sequence

`a^b^c^d^e`

is illegal. L^AT_EX will complain

! Double superscript.

```
1.1 \begin{math}a^b^
c^d^e\end{math}
```

? h

I treat ‘`x^1^2`’ essentially like ‘`x^{1^2}`’.

and if you merely hit `return` to continue, L^AT_EX would present you with

$$a^{bcde}$$

As an error message it is pretty accurate. You probably meant `a^{bcde}` or `a^{b^{c}}` or something similar. Note too that when you ‘simultaneously’ sub- and super-script `x_n^2`

the sub and super-scripts will usually appear lined up at their left hand edge. We could have written that as `x^2_n`, and obtained the same result: x_n^2 . Table 3 accounts for most situations.

QUESTION 5.4 *The following are written differently? How should they be expressed in L^AT_EX?*

$$(((a^4)^3)^2)^1 \quad (((a^4)^3)^2)^1$$

Before absorbing L^AT_EX’s handling of sub- and super-scripts as ‘a good thing’, have a look at the following:

$$a_2^2 \quad a_2 \quad a^2$$

or even

$$a_1 a_2^2 c^4$$

Although it is possible to see the reasoning behind placing sub- and super-scripts differently, depending on context, it does tend to make these particular expressions untidy.

QUESTION 5.5 *How might we align the subscripts? Remember that nothing has its uses.*

Sub- and super-scripts can mean many things, depending on context. For example, what is the function of the 2 in SU_2 , $\log_2 x$, x^2 and T_2^2 ? In SU_2 it is the number of dimensions of the Lie group; in $\log_2 x$ it is the base of the logarithm; if x is a vector, the 2 in x_2 is an index; the 2 in x^2 could be a power, but if T is a tensor, the 2 in T_2^2 is a contrainvariant tensor index[4]. The superscript might also indicate a footnote!

α	<code>\alpha</code>	β	<code>\beta</code>
γ	<code>\gamma</code>	δ	<code>\delta</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>
ζ	<code>\zeta</code>	η	<code>\eta</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>
ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>
ν	<code>\nu</code>	ξ	<code>\xi</code>
o	<code>o</code>	π	<code>\pi</code>
ϖ	<code>\varpi</code>	ρ	<code>\rho</code>
ϱ	<code>\varrho</code>	σ	<code>\sigma</code>
ς	<code>\varsigma</code>	τ	<code>\tau</code>
υ	<code>\upsilon</code>	ϕ	<code>\phi</code>
φ	<code>\varphi</code>	χ	<code>\chi</code>
ψ	<code>\psi</code>	ω	<code>\omega</code>

Table 1. lower case Greek symbols available in maths environments

Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>
Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>
Ξ	<code>\Xi</code>	Σ	<code>\Sigma</code>
Π	<code>\Pi</code>	Υ	<code>\Upsilon</code>
Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>
Ω	<code>\Omega</code>		

Table 2. upper case Greek symbols available in maths environments

$x - 2y^2$	<code>x_2y^2</code>
x^2	<code>x^2</code>
x_2	<code>x_2</code>
2^x	<code>2^x</code>
x^2y^2	<code>x^2y^2</code>
x^2y^2	<code>x^2y^2</code>
x_2y_2	<code>x_2y_2</code>
x_2y^2	<code>x_2y^2</code>
$2x^2$	<code>2x^2</code>
$x^{\{2y\}}$	<code>x^{2y}</code>
$x^{\{2^y\}}$	<code>x^{2^y}</code>
$x_{\{2^y\}}$	<code>x_{2^y}</code>
$x_{\{2^y\}^2}$	<code>x_{2^y}^2</code>
$\{x_{\{2\}}\}^y$	<code>x_{2^y}^y</code>

Table 3. sub- and super-scripts

<code>\hat{a}</code>	\hat{a}
<code>\check{a}</code>	\check{a}
<code>\tilde{a}</code>	\tilde{a}
<code>\acute{a}</code>	\acute{a}
<code>\grave{a}</code>	\grave{a}
<code>\dot{a}</code>	\dot{a}
<code>\ddot{a}</code>	\ddot{a}
<code>\breve{a}</code>	\breve{a}
<code>\bar{a}</code>	\bar{a}
<code>\vec{a}</code>	\vec{a}

Table 4. mathematical accents

<code>\tilde{x}</code>	\tilde{x}
<code>\widetilde{x}</code>	\widetilde{x}
<code>\widetilde{xy}</code>	\widetilde{xy}
<code>\widetilde{xyz}</code>	\widetilde{xyz}
<code>\widetilde{wxyz}</code>	\widetilde{wxyz}
<code>\hat{x}</code>	\hat{x}
<code>\widehat{x}</code>	\widehat{x}
<code>\widehat{xy}</code>	\widehat{xy}
<code>\widehat{xyz}</code>	\widehat{xyz}
<code>\widehat{wxyz}</code>	\widehat{wxyz}

Table 5. wider yet and wider

QUESTION 5.6 *Tackle the following:*

Pascal's triangle is based on the identity

$$n^{-1}C_k + n^{-1}C_{k-1} = nC_k$$

More simple operators

Square roots can be expressed quite simply by the `\sqrt` instruction:

```
\sqrt{2}
```

produces a 2 with a square root symbol around it – $\sqrt{2}$.

The bar over the square rooted expression will be extended if required: thus

```
\sqrt{b^2-4ac}
```

will have the bar over the entire grouped expression – $\sqrt{b^2 - 4ac}$. In order to get n th roots, use the expression

```
\sqrt[n]{x}
```

would produce something like $\sqrt[n]{x}$. This is quite a pleasing use of an optional parameter.

The instructions `\overline` and `\underline` work in a similar manner. (Another name for overline is *vinculum*, which sounds more specialised, and may convey the notion of grouping.) Thus:

```
\overline{\alpha} + \underline{\gamma^2}
```

should do just what it says: $\bar{\alpha} + \underline{\gamma^2}$.

QUESTION 5.7 *Regroup that last expression to obtain $\overline{\alpha + \underline{\gamma^2}}$*

QUESTION 5.8 *Flushed with your success so far, try*

Using the mean water depths over the integration range, $\overline{h(x_1)}$ and $\overline{h(x_2)}$ we obtain

$$y_1\overline{h(x_1; y_1 \geq y \geq 0)} = y_2\overline{h(x_2; y_2 \geq y \geq 0)}$$

If you really cannot guess the instruction for \geq peek at Table 9.

QUESTION 5.9 *How do you place an overline or accent over nothing?* If $\bar{}$ denotes complex conjugation, then

$$\overline{\xi + \zeta} = \bar{\xi} + \bar{\zeta}$$

There are two other ‘pairs’ of symbols: first, `\overleftarrow` and `\overrightarrow` which are fairly self-explanatory and produce

$$\overleftarrow{x+y} \quad \text{and} \quad \overrightarrow{y+x}$$

and secondly, horizontal braces may be placed above or below a group; so that

$$\overbrace{x_1 \dots x_n} \quad \text{and} \quad \underbrace{a^1 \dots a^n}$$

may be produced from `\overbrace` and `\underbrace`, respectively.

Mâthémâtîcāl accēnts

Mathematicians often like to put modifiers over symbols (so do other people, but we shall give them honorary mathematical status just for now). As we have already seen, L^AT_EX supports some accents (or diacriticals) in its ‘normal’ document environment. In maths environments, it also supports accents; there appears to be considerable overlap (Table 4).

QUESTION 5.10 *Typeset*

Then the relativistic four-tensor equation of motion of this particle takes the form

$$mc^2 \ddot{a}_\nu = eF_{\nu\mu}(a)\dot{a}^\mu$$

which is identical with the Lorentz equation.

There are differences however. These are genuinely maths accents; you cannot use them in ‘normal’ text – L^AT_EX will complain; nor can you use expressions such as `\^` to obtain a circumflex in maths environments. Treat them as different symbols. After all, the whole of mathematics depends on having symbols which look identical, but have different meanings, depending on context. It is partly this which makes typesetting maths rather tricky in the first place. A second difference lies in the fact that you may accent a whole formula, as well as a symbol – but beware, the accent appears centred over the whole group, whether a single character, or many. Thus `\vec{a+b}` gives us $\vec{a+b}$. Perhaps we really meant $\vec{a} + \vec{b}$.

QUESTION 5.11 *What should we have said to get $\vec{a} + \vec{b}$?*

QUESTION 5.12 *Try this:*

The average rate at which work is done over one wave period T , \bar{W} , is $\bar{E}c_g$.

From time to time it may be necessary to use `\overline` instead of `\bar`. Consider the following:

“The bottom frictional term $\bar{\tau}_{hy}$ may be given as $\rho f_w |\overline{u_{wh}}| v/2$ where C and f_w are both friction coefficients and $|\overline{u_{wh}}|$ is the absolute time average.”

In this case the author clearly wanted to indicate that the average was the average of u_{wh} and not \bar{u}_{wh} , or indeed u_{wh} .

QUESTION 5.13 *You are almost ready to attempt the following example. There are two extra pieces of information you need for \rightarrow and \equiv . They may be found in Tables 9 and 10.*

Consider the four-fermion interaction

$$g(\bar{\psi}_2 O \psi_1)(\bar{\psi}_4 O \psi_3)$$

Make the transformations

$$\psi_i \rightarrow \gamma_5 \psi_i$$

$$\bar{\psi}_j \rightarrow -\bar{\psi}_j \gamma_5$$

Then chirality invariance implies

$$\bar{\psi}_j O \psi_i \equiv -\bar{\psi}_j \gamma_5 O \gamma_5 \psi_i$$

where O is any operator. We have

$$O \gamma_5 = O$$

and

$$-\gamma_5 O = O$$

or

$$[O, \gamma_5]_+ = 0$$

That means O and γ_5 anticommute.

There are two other ‘grouping’ accents, `\widehat` and `\widetilde`, which grow to accommodate whatever is below them – within limits (Table 5).

From this table you can see that the smallest `\widetilde` or `\widehat` is nevertheless distinct from a `\tilde` or `\hat`, and that the maximum extent of these `\wide` mathematical accents is three ‘normal’ characters. Contrast this with the previous section, where so-called ‘operators’ such as `\overbrace` or even `\sqrt` can grow to fit whatever they group. In passing, consider what \widehat{x}_2 might mean: is it ‘the estimate of x_2 ’? It looks ugly and may be more likely to be written in the slightly ambiguous form \hat{x}_2 , where it could equally ‘mean’ ‘the second part of \hat{x} ’ [1].

We have many ways of doing this...

The language of mathematics is a very rich one, and the vocabulary expects a very wide range of symbols and operators, much larger than that encountered in normal text. L^AT_EX allows access to many symbols: Table 6 gives some useful new symbols, with their instruction. There are yet more, but exposure to the rich diet of L^AT_EX might lead to indigestion. Some symbols require a particular ‘package’ to be available to L^AT_EX. In a later chapter we’ll look at how packages may be loaded. Until then, you can’t use them, but since you are probably impatient, issue the command

```
\usepackage{latexsym}
```

in the preamble. While we might query the inclusion of `\b` and `\v` in the mathematical symbols, especially when useful chemical symbols are ignored, we must allow Knuth and Lamport some whimsical indulgences. After all, without them, we would have none of this. All of these new symbols can be used as sub- or superscripts, and they will behave as expected, *except* for `\angle`. It never gets any smaller.

QUESTION 5.14 *Set the following text and its included equations:*

The surface is $y = h_s(x)$, the bed is $y = h_b(x)$; the slopes, if small, are $\alpha = -dh_s/dx$, $\beta = -dh_b/dx$. If α , β are small, the equation can be transformed into this coordinate system to give the approximate relation

$$\tau_b = \rho g h \alpha + 2G - T$$

\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>	\hbar	<code>\hbar</code>
\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\imath	<code>\imath</code>	∇	<code>\nabla</code>
\neg	<code>\neg</code> or <code>\lnot</code>	\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\wp	<code>\wp</code>
\perp	<code>\perp</code>	\sharp	<code>\sharp</code>	\Re	<code>\Re</code>	\parallel	<code>\parallel</code>
\clubsuit	<code>\clubsuit</code>	\Im	<code>\Im</code>	\angle	<code>\angle</code>	\diamondsuit	<code>\diamondsuit</code>
∂	<code>\partial</code>	\triangle	<code>\triangle</code>	\heartsuit	<code>\heartsuit</code>	∞	<code>\infty</code>
\backslash	<code>\backslash</code>	\spadesuit	<code>\spadesuit</code>	\mho^*	<code>\mho^*</code>	\square	<code>\Box^*</code>
\diamond	<code>\Diamond^*</code>						

* available when the `latexsym` package is loaded

Table 6. an abundance of riches

where the ice thickness is $h = h_s - h_b$.

QUESTION 5.15 *These displayed equations provide some additional experience and practice:*

$$h_0 + \epsilon = h + p = h_0 + h_1 + p$$

$$(m + 1)h_1/h_0 + m\alpha_1/\alpha_0 = 0$$

$$h^{m+1}\alpha^m = h_0^{m+1}\alpha_0^m$$

$$\tau^2 = \tau_{xy}^2 + \tau_{zx}^2$$

$$\lambda = A\tau^{n-1}m = \rho$$

$$eM_\omega/PM_a = 0.623\rho e/P$$

Prime

The normal use of prime is as a superscript. The character you obtain by `\prime` is a rather large, non-superscript symbol. Thus we use it in the form x^\prime . But you may also use the right quote mark as an already superscripted prime. Thus $x^\text{'}$ and x^\prime are equivalent and give x^\prime . One of the reasons that `\prime` does not come ready superscripted is that that would limit its capabilities, give inconsistencies, and under certain circumstances may lead to strange spacing. It also means you can do something like this: x_\prime .

QUESTION 5.16 *A small example to try*

When the waves reach point a , the wave height is give by point l' . Point l' is transferred to point l'' on the 15 m sec^{-1} wave height curve. The wave height variation from point a to point d is given by curve $l'm''$, which is parallel to curve $l'm'$ on the 15 m sec^{-1} wave height curve. Therefore curve $Ol''m''$ is the determined wave height variation.

QUESTION 5.17 *More waves:*

The wave period increases to the value \overline{bt}_1 while the wave height increases to the value $\overline{ll'}$ by following the wave height curve Ol' .

This illustrates a minor problem, since what we really wanted was $\overline{ll'}$, and not $\overline{ll'}$, or even $\overline{ll'}$. Any suggestions?

Above and beyond

We place items 'over' other items with the instruction `\frac`. As this is intended for fractions, it also places a line between the items;

`\frac{a}{b}`

produces the fraction $\frac{a}{b}$ in text style, but it is really much better in display style, where it produces

$$\frac{a}{b}$$

This is a rather trivial example, but expressions such as

$$\frac{\partial F_{\nu\mu}}{\partial x_\mu} = 4\pi j_\nu(x)$$

start to demonstrate its utility. It was created from the following L^AT_EX:

```
\[
\frac{\partial F_{\nu\mu}}{\partial x_\mu}=4\pi j_\nu(x)
\]
```

The use of `\frac` is perhaps a little counter-intuitive since you have to remember to issue the command and then state the numerator and denominator, while in 'speech' you might be inclined to plough on and just say 'numerator over denominator'.

Set the following:

QUESTION 5.18

The coefficient $\frac{2}{3}e^2/c^3$ is related to the so-called relaxation time of the particle.

QUESTION 5.19

The zeros of the quadratic $ax^2 + bx + c$ are

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

QUESTION 5.20

If $\nu = n_1 + n_2 - 2$ and

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

then

$$\frac{\bar{X}_1 - \bar{X}_2}{s\sqrt{(\frac{1}{n_1} + \frac{1}{n_2})}}$$

is distributed as t_ν .

QUESTION 5.21

The ratio between the diffracted and incident wave heights is given by

$$\frac{H}{H_i} = |F(x, y)| = K_{\bar{a}}$$

where $K_{\bar{a}}$ is the diffraction coefficient.

QUESTION 5.22

$${}^6C_2 = \frac{6!}{4!2!}$$

More uses of sub- and super-scripts

The conventions already described for sub- and super-scripting may also be used for some of L^AT_EX's 'large operators'. Perhaps the two key operators are `\sum` for summation, and `\int` for integral sign. Note that `\Sigma`, Σ , and `\sum`, \sum or \sum , are different (they may look similar, but they behave differently).

In order to place the limits on `\sum` and `\int` simply sub- and/or super-script. L^AT_EX correctly understands that these two special operators must have their sub- or super-scripts placed above or below the symbol as appropriate.

For example,

```
\begin{displaymath}
S = \sum^7_{r=1} S_r
\end{displaymath}
\begin{displaymath}
\int_{V_i}^{V_g} P(V, T) dV = 0
\end{displaymath}
```

will produce the following:

$$S = \sum_{r=1}^7 S_r$$

$$\int_{V_i}^{V_g} P(V, T) dV = 0$$

QUESTION 5.23 *What differences occur when the large operators are used in text style and display style?*

There are several other large operators. All are listed in Table 7 in their text and display styles, showing the default positions of limits. In text style, the first column of the table, the limits are always placed to the right: in display style, they are placed to the right for integrals, but above and below in all other cases. You can alter this behaviour by the use of two instructions, `\limits` and `\nolimits`. The first of this pair instructs L^AT_EX to place limits above and below, while the other instructs it to place them to the right. There are probably no sound reasons for using `\limits` within an equation in text, since this would tend to space out lines irregularly, but

perhaps there are times when

$$\int_{V_i}^{V_g} P(V, T) dV = 0$$

might just be required. The modifier should follow the large operator instruction:

```
\int\limits_{V_1}^{V_g} P(V, T) dV = 0
```

QUESTION 5.24 *Prepare some examples where you use the large operators with limits. It does not matter that the operators or limits you choose are absurd. Who's to know? Pretend you are a theoretical meta-physicist. For example*

$$X \ominus B = \bigcup_i X_i \ominus B$$

$$X \ominus \check{B} = X \cap X_{-h}$$

$$\int_{-B/2}^{B/2} R''(y, y') \varphi''(y') dy' = \gamma'' \varphi''(y)$$

$$C^*(k) = \frac{\sum_{\lambda=1}^g A(Z_\lambda) C_\lambda^*(k)}{\sum_{\lambda=1}^g A(Z_\lambda)}$$

$$p(h|r) = \sum_z q(z|r) q(z + h|r)$$

$$\frac{e^{-m'/2}}{\sqrt{2\pi}\sigma_1^{m'}} \frac{e^{-m''/2}}{\sqrt{2\pi}\sigma_1^{m''}} = \frac{e^{-m/2}}{\sqrt{2\pi}^m \sigma_1^{m'} \sigma_2^{m''}}$$

Display, text, script and scriptscript

Display style is the normal form used by displayed mathematics, just as text style is the normal form for maths within text. The 'same' expression in display and text style will look different. This is especially noticeable in the large operators of Table 7. The differences are in the size and the position of the sub- and super-scripts. Fractions are also handled differently.

At times it is necessary to over-ride the default activity and force different behaviour. For example, within displayed maths it may be necessary to write

$$M_\phi = \frac{1}{2}(\phi_{84} - \phi_{16})$$

or

$$H_{1/n} = \frac{\int_{H_n}^\infty H_p(H) dH}{P(H_n)}$$

where the `\frac{1}{2}` and the `\int` are presented in their text style equivalents in order to compress them slightly. The instruction to do this is `\textstyle`. This is a genuine T_EX instruction and carries with it T_EX's legacy: it is therefore used as

```
M_{\phi} =
{\textstyle\frac{1}{2}}(\phi_{84} - \phi_{16})
```

and

```
H_{1/n} =
\frac{{\textstyle\int_{H_n}^\infty H_p(H) dH}}{P(H_n)}
```

\int_b^a	\int_b^a	<code>\int</code>	\oint_b^a	\oint_b^a	<code>\oint</code>			
\sum_b^a	\sum_b^a	<code>\sum</code>	\prod_b^a	\prod_b^a	<code>\prod</code>	\coprod_b^a	\coprod_b^a	<code>\coprod</code>
\bigcap_b^a	\bigcap_b^a	<code>\bigcap</code>	\bigcup_b^a	\bigcup_b^a	<code>\bigcup</code>	\bigsqcup_b^a	\bigsqcup_b^a	<code>\bigsqcup</code>
\bigvee_b^a	\bigvee_b^a	<code>\bigvee</code>	\bigwedge_b^a	\bigwedge_b^a	<code>\bigwedge</code>	\bigoplus_b^a	\bigoplus_b^a	<code>\bigoplus</code>
\bigodot_b^a	\bigodot_b^a	<code>\bigodot</code>	\bigotimes_b^a	\bigotimes_b^a	<code>\bigotimes</code>	\bigoplus_b^a	\bigoplus_b^a	<code>\bigoplus</code>

Table 7. Large operators, with their limits in the default positions

The corresponding instruction to use in text style when you want a truly large operator (or a larger fraction), is `\displaystyle`, but this is provided for symmetry rather than utility. Placing display style elements in text is seldom required and it does make line spacing rather irregular. However, when we use `\frac` the numerator and denominator will be handled as text style. Usually this goes unnoticed, but when a large operator is part of the fraction, it is rather obvious. For example,

`H_{1/n} = \frac{\int_{H_n}^{\infty} H_p(H) dH}{P(H_n)}`
gives

$$H_{1/n} = \frac{\int_{H_n}^{\infty} H_p(H) dH}{P(H_n)}$$

In the event that we wanted the integral to be its full size, we could have written:

`H_{1/n} = \frac{\displaystyle \int_{H_n}^{\infty} H_p(H) dH}{P(H_n)}`

Sub- and superscripts normally diminish in size: there are instructions which can give us some control here too: `\scriptstyle` and `\scriptscriptstyle`. Intuitively, script style refers to the first level of sub- or superscripts, while the script script style is the second (and subsequent level). Every now and again it is handy to be able to modify (or subvert) \TeX 's 'expert system' and change the size of these styles. Again, they are unreconstructed and are best used within their own group.

Other operators

\LaTeX also has an extensive suite of ordinary (and extraordinary) operators which are used in equations. Before presenting them all, it is useful to try to establish some order, which helps in the understanding of how \LaTeX treats these operators. The first group can be described as 'binary' or 'dyadic' operators since they are normally preceded and followed by other symbols, e.g. $x + y$, $a \times b$, $n \div m$ or $\alpha - \beta$. Where they might exist on their own they form a second group of 'monadic' or 'unary' operators: $x = -1$

Anyone who has experienced APL, the arcane programming language, would be happy to make a distinction between binary and unary operators, and so too is \LaTeX . But the difference is in terms of spacing. The spacing around the minus is different in the expressions $x = a - b$ and $x = -y$. Examine this example:

$$x = -y - z$$

where the spacing around the two minus signs is different, reflecting their different status.

Concentrating on the binary operators, \LaTeX recognises at least 32 'primitive' operators of this type (Table 8). There are even facilities to define your own if you wish. Once defined as a binary operator, \LaTeX will apply the rules for binary operations.

Binary operations normally have extra space between them (the whole question of how space in equations is treated will be dealt with later). At the moment it is sufficient to say that if the operator is not being used in a binary way, \LaTeX is 'smart' enough to treat it as a simple symbol (most times). Thus H^+ , f^* or even $g^\circ \mapsto g^*$ will use the 'operators' as simple symbols, leaving no extra space. Contrast this with $a + b * c \circ d * e$, where there is a little extra space between them as operators (believe me, or measure things very carefully!).

QUESTION 5.25 *This quark is a boson (or a fermion):*

Suppose the creation and annihilation operators of bosons are a and a^\dagger , and the corresponding creation and annihilation operators for fermions are b and b^\dagger , the the boson operators obey the commutation relation

$$[a^\dagger, a] = a^\dagger a - a a^\dagger = 1$$

whereas the fermion operators obey the anticommutation relation

$$\{b^\dagger, b\} = b^\dagger b + b b^\dagger = 1$$

where 1 denotes the identity operator.

+	+	-	-	/	/	±	\pm
∓	\mp	\	\setminus	·	\cdot	×	\times
*	\ast or *	*	\star	◇	\diamond	○	\circ
•	\bullet	÷	\div	∩	\cap	∪	\cup
⊕	\uplus	∩	\sqcap	⊔	\sqcup	◁	\triangleleft
▷	\triangleright	∨	\wr	○	\bigcirc	△	\bigtriangleup
▽	\bigtriangledown	∨	\vee or \lor	^	\wedge or \land	⊕	\oplus
⊖	\ominus	⊗	\otimes	⊘	\oslash	⊙	\odot
†	\dagger	‡	\ddagger	∏	\amalg	◁	\lhd*
▷	\rhd*	⊲	\unlhd*	▷	\unrhd*		

*available when latexsym package is loaded

Table 8. binary operators

=	=	>	>	<	<	≠	\ne or \neq
∝	\propto	⋈	\Join*	≤	\leq or \le	≥	\geq or \ge
≡	\equiv	⋞	\prec	⋟	\succ	≈	\sim
⋞	\preceq	⋟	\succeq	⋈	\simeq	≪	\ll
≫	\gg	⋞	\asymp	⊂	\subset	⊃	\supset
≈	\approx	⊂	\subseteq	⊃	\supseteq	≅	\cong
⊆	\sqsubseteq	⊃	\sqsupseteq	⋈	\bowtie	∈	\in
⊇	\ni or \owns	∉	\notin	⊥	\vdash	⊥	\dashv
⊨	\models	⌢	\smile	⊥	\mid	≐	\doteq
⌢	\frown	∥	\parallel	⊥	\perp	⊆	\sqsubset*
⊃	\sqsupset*						

*available when latexsym package is loaded

Table 9. relations

Relations

There are lots of relations in the family of L^AT_EX as Table 9 shows. If you ever require a negated version of one of these relations, such as $\not\sim$, or more likely, \neq , then you precede the relation by `\not`, that is, `\not\smile` or `\not\equiv`. The positioning is not always perfect, notably with `\in`, and there is another version `\notin` which is said to look better.

QUESTION 5.26 *More practice:*

$$\tau_1/\tau_0 \approx 0.5p/h_0$$

$$h/L \gg \partial h/\partial x$$

$$\frac{\Delta}{t} \geq \sqrt{2gh}$$

$$\nabla^2 f(i, j) \equiv \delta_x^2 f(i, j) + \delta_y^2 f(i, j)$$

QUESTION 5.27 *In order to redress the balance in favour of ‘text’ maths, try the following passage. The vertical bar in this instance may be obtained from | rather than \mid, since it is not a relational operator.*

The expected cost, $c(i|\vec{z})$ is given by $\sum_j p(j|\vec{z})\lambda(i|j)$; if both $\lambda(i|i) = 0$ and the $\lambda(i|j)$ ’s are equal when $j \neq i$, minimizing the expected cost is equivalent to minimizing $\sum_{j \neq i} p(j|\vec{z})$. The Laplacian $\partial^2 f/\partial x^2 + \partial^2 f/\partial y^2$ is an orientation-invariant derivative operator. The responses of $\sqrt{\Delta_+^2 + \Delta_-^2}$ are $h\sqrt{2}$, $h\sqrt{2}$, h and h .

You might like to try out the alternative of using \mid to see the impact it has. This variation in the behaviour of the vertical bar begins to illustrate the difference between the originator of the text, who would ‘know’ the meaning of the vertical bar symbol, and someone who has to typeset or type it without that background. Often the L^AT_EX version of an equation contains far more information than the typeset.

If the vast array of relations is not sufficient for you, there are also lots of arrows, shown in Table 10. `\Leftrightarrow` and `\iff` give the same symbol, but `\iff` has extra space around it.

`a\Leftrightarrow b` gives $a \iff b$

`a\iff b` gives $a \iff b$

As if this was not enough, you can place symbols over left and right arrows with the aid of the `\stackrel` instruction.

For example $a \xrightarrow{x'} b$ was obtained by

`a\stackrel{x'}{\longrightarrow}b`

The command is not restricted to arrows and anything can be placed on top of anything else by this means.

QUESTION 5.28 *I had problems finding examples of equations with arrows, but these will do:*

$$Q \in N(P) \iff P \in N(Q)$$

$$d(X_i, X) \rightarrow 0 \Rightarrow W_k^{(n)}(X_i) \rightarrow W_k^{(n)}(Y), \forall n, k$$

QUESTION 5.29 *Demonstrate that arrows may be negated, as well as relations.*

\leftarrow	<code>\leftarrow, \gets</code>	\longleftarrow	<code>\longleftarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\rightarrow	<code>\rightarrow, \to</code>	\longrightarrow	<code>\longrightarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Leftrightarrow (or \iff)	<code>\Leftrightarrow</code> (or <code>\iff</code>)
\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\leadsto^*	<code>\leadsto^*</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>

*available when latexsym package is loaded

Table 10. arrows, hooks and harpoons

Chapter 6. Tablets from the mountain

Tables

Tables belong in two environments in L^AT_EX. If we wish to create a ‘proper’ ruled table, we use the `tabular` environment. You might have supposed that we used the `table` environment, but as you will see in Section 7 this has already been claimed. It is always a problem in a system which claims to give ‘natural’ names to things, to find enough ‘natural’ names.

You know how to set up the environment, but `tabular` takes an additional argument. This argument is very compact, and very powerful. The argument allows the structure of the columns in the table to be specified. At its most basic it allows us to left-justify, centre, or right-justify the contents of a column. If we write

```
\begin{tabular}{ccc}
```

we are saying we wish three columns, each centred. If `c` is for centred, you should be able to guess what `r` and `l` do. I said we want a ‘proper’ ruled table. To say we want a vertical rule between any two entries in a row, we insert the vertical bar symbol `|`:

```
\begin{tabular}{|ccc|}
```

which would give us a table with a vertical rule around the left and right sides. This does not give us the rule around the top and bottom. We will do that ourselves in a moment.

A worked example

In general, tables are quite easy and straightforward in L^AT_EX. Part of the price paid for that ease is that very complicated tables may take excessive ingenuity and thought. There are several issues here, but it is often a good idea to take a less mechanistic approach to tables. What is the function of a table? Why are we bothering to create one in the first place? Sadly, many tables which appear in reports and books would be better left out. They are included to lend a (spurious) air of legitimacy and erudition. They ought to be there as part

of the development of the theme or argument. Sometimes tables are an archival mechanism, so that others have all the information needed to confirm or refute the arguments contained. But too often the tables are merely included to confirm that a great deal of worthy work was done, work which should receive some recognition. Most tables are probably unnecessary. Having said that, they will not go away.

The other point to make about tables is that they are very strongly visual. In a broad sense, L^AT_EX tries to separate the content of a document from its presentation, but with tables this is probably problematic in general, and to look at the specifics, choosing how columns are handled (or that the material is presented as a column rather than a row), is surely specifying many of the presentational aspects too. The same content may be presented several different ways in a table. Some ways will be confusing and difficult to understand. We have only to consider the example of a railway timetable to see just how difficult an apparently simple problem may be to solve. We can therefore expect that a table may require adjustment before it ‘works’.

Tables need not contain any rules at all, and still be useful and reasonably pleasing. Table 11 contains some useful typographic information. The ‘measure’ indicates a recommendation of desirable line length from a number of reputable sources. Although agreement is not exact, there is fair consensus, once one allows for some translation between ‘words’, ‘characters’ and ‘letters’. A common fault when authors are allowed to determine page width is to have too great a ‘measure’. People tend to read comfortably with the measure at the sorts of values in this table. And now, back to the plot...

Table 11 was created quite simply:

```
\begin{tabular}{rl}
guru           & & measure \\
Morison        & & 10--12 words \\
Karl Treebus   & & 10--12 words or \\
                & & 60--70 letters
```


guru	measure
Morison	10–12 words
Karl Treebus	10–12 words or 60–70 letters
John Miles	60–65 characters
Leslie Lamport	less than 76 characters
Linotype	7–10 words or 50–65 letters

Table 11. A simple unruled table

guru	measure
Morison	10–12 words
Karl Treebus	10–12 words or 60–70 letters
John Miles	60–65 characters
Leslie Lamport	less than 76 characters
Linotype	7–10 words or 50–65 letters

Table 12. A ruled example

```
John Miles & 60--65 characters \\
Leslie Lamport & less than 76 characters \\
Linotype & 7--10 words or\\
& 50--65 letters
\end{tabular}
```

The first point to be made is that the table contents have been laid out so that they can be read. How were the row entries separated? By inspection the & character is used. How do we terminate a row entry – after all, how does L^AT_EX know when we have finished a row? If it was very clever it could just count off each entry, but it isn't that smart. The instruction \\ is used to indicate the end of a row. This is one of the first instances where we have met an instruction which does different things in different circumstances. Since the end of the table is signalled, there is no need for a \\ terminator on the last line. Compare this with the similar use of \\ in the `verse` environment.

The example is actually incomplete. This table is centred across the page width. To achieve this, the whole `tabular` environment itself was placed within a `center` environment (note the American spelling – L^AT_EX is, after all, American). The `center` environment has nothing to do with the table itself, and can be applied to other situations or environments to give this effect. It is used in the following fashion:

```
\begin{center}
\begin{tabular}{...}
.
.
\end{tabular}
\end{center}
```

There are also the associated `flushright` and `flushleft` environments, which do much what they say.

It is very easy to take this last example and enclose it in a frame, simply by adding to the `{rl}` parameters:

```
\begin{tabular}{|rl|}
```

While this ensures that vertical rules are drawn on the left and right, it does not help with the horizontal rules. Another instruction, `\hline` will do this:

```
\begin{center}
\begin{tabular}{|rl|}
\hline
guru & measure \\
\hline
Morison & 10--12 words \\
Karl Treebus & 10--12 words or\\
& 60--70 letters \\
John Miles & 60--65 characters \\
Leslie Lamport & less than 76 characters \\
\end{tabular}
\end{center}
```

```
Linotype & 7--10 words or\\
& 50--65 letters \\
\hline
\end{tabular}
\end{center}
```

The horizontal rules used here separate the line containing the column descriptions too. Note that the horizontal lines do not require to be ‘terminated’ by the \\ instruction. But also note that we have had to introduce a \\ on the last ‘line’ of the table in order to separate it from the `\hline`.

Just because you can create horizontal and vertical rules, there is no need to box every entry. When you use vertical or horizontal rules depends on how you wish your reader to tackle the table, and what you are trying to convey. The use of rules in tables is partly a fashionable artefact. Looking at text books from different periods tends to show a greater or lesser tendency to include rules. Technologically, vertical rules have tended to fall out of favour because the typesetting equipment used found them difficult to include. By default, L^AT_EX does not give you any control over the thickness of either the vertical or horizontal rules. Placing two `\hline` instructions together merely gives two slightly separated horizontal lines.

QUESTION 6.1 Take Table 12 but make the ‘guru’ column left-justified.

QUESTION 6.2 Tables 11 and 12 contain the same information (recommended line widths or ‘measure’). Which do you find easier to read? and what further improvements might you like to make to increase readability?

Classic example

Let us take the classic example of the AT&T stock and see how that translates into L^AT_EX (Table 13). The basic structure of the table is of three columns, where the left column contains the year (right-justified), the middle column has the stock price (centred), and the third column has the dividend, centred again. There are rules between each column. The basic structure is described by

```
\begin{tabular}{|r|c|c|}
```

The problem that remains is how to handle the column titles (or stubs). Essentially we need a way of temporarily overriding the instructions we have just given. This may be done by the `\multicolumn` instruction. The overall title of the table can be given as:

```
\multicolumn{3}{|c|}{AT\&T Common Stock}
```

The first argument says we are taking 3 columns (in this case). So we are spanning three columns. The second argument is a position argument, to say how we want the information in the new column to be placed (using the same conventions as for `tabular`). In this instance we want it centred. We also have to say whether we need a vertical bars or not, since the `multicolumn` over-rides all the existing information. The last argument is the piece of text which we wish to be placed.

This seems fairly logical when several columns are involved. How is it used to apply to a single column? In this case the ‘multi’ is 1. Using this to over-ride a single column sounds like overkill, but it is just the limiting condition. To use `\multicolumn` in this case, we say something like

```
\multicolumn{1}{|c|}{Year}&
\multicolumn{1}{|c|}{Price}&
\multicolumn{1}{|c|}{Dividend}
```

Perhaps the remaining question is to wonder why the vertical bar information is presented the way it is, and not some other alternative.

We know that the horizontal lines are obtained from `\hline`, the column separator is `&` (which helps to explain the `\&` above), and that the line terminator is `\\`. We have all the bits we need:

```
\begin{tabular}{|r|c|c|}
\hline
\multicolumn{3}{|c|}{AT&T Common Stock}\\
\hline
\multicolumn{1}{|c|}{Year}&
\multicolumn{1}{|c|}{Price}&
\multicolumn{1}{|c|}{Dividend}\\
\hline
1971 & 41--54 & \$2.60 \\
2 & 41--54 & \$2.70 \\
3 & 46--55 & \$2.87 \\
4 & 40--53 & \$3.24 \\
5 & 45--52 & \$3.40 \\
\hline
\end{tabular}
```

This will give us Table 13. It would have looked better if it were centred on the column. To prove the point a duplicate version, Table 14, has been centred over its column, using the `center` environment.

QUESTION 6.3 *What are we going to do when AT&T stock rises to give a dividend of over \$10.00? Assume that by 1983 the dividend rose to \$11.11 and the price was 99–110. Generate a new table to reflect this new information.*

Some variations

Obviously, there is a lot more to tables, and L^AT_EX manages remarkably well. Table 13 is given its ‘natural’ width – that is, the width that L^AT_EX calculates by taking into account the widest column entry and the amounts by which it separates columns. We can stretch it out with an extra argument – but it is a slightly different environment – `tabular*`:

```
\begin{tabular*}{4in}{|r|c|c|}
```

A `tabular*` environment must be completed with an `\end{tabular*}`.

Unfortunately, although this ‘stretched’ form looks rather appealing, it is not without its pitfalls. The stretching is not performed uniformly across the columns. A simplified version of Table 13, stretched within the available column width:

```
\begin{tabular*}{\linewidth}{|r|c|c|}
\hline
1971 & 41--54 & \$2.60 \\
\hline
\end{tabular*}
```

will yield:

1971	41–54	\$2.60
------	-------	--------

This looks particularly ugly. Later we’ll see how to manipulate this more elegantly. At the moment, it perhaps best avoided. The use of `\linewidth` requires some elaboration. In a `multicolumn` environment or class, it takes the value of the text within a column. In ‘normal’, single column setting, it will also take a value corresponding to the current text width. However, any valid expression of a dimension, as defined in Chapter 3, is suitable.

There is also an option argument (in square braces), which is another ‘positional’ argument, indicating where the table is in a vertical sense. By default a table is aligned horizontally along its centre, but you can align it on its top row, or its bottom row (using `t` and `b`). Why should you want to do this? Usually when you have several tables side by side. For the sake of an example,

```
\begin{tabular}[t]{|c|}
\hline A \\ \hline
\end{tabular}
\begin{tabular}[t]{|c|}
\hline A \\ B \\ \hline
\end{tabular}
```

will produce

A	A
	B

QUESTION 6.4 *Reproduce the last example, but aligned on the table bottom. For amusement you might like to experiment with all possible combinations of alignment.*

The full structure for `tabular` may therefore be

```
\begin{tabular*}{4in}[b]{|r|c|c|}
```

or

```
\begin{tabular}[t]{|r|c|c|}
```

Note the positions of the arguments. I find L^AT_EX’s handling of optional arguments arbitrary, but perhaps I have not yet seen the model properly.

More tabular esoteria

One recurrent theme with tables is the desire to align numerical information on the decimal point. This is quite easily achieved by use of one of the column specifiers, the `@` specifier. This allows text to be inserted at a given position in a column (in every column), and suppresses the space which is normally added between adjacent column entries. In this context the suppression of space is ‘a good thing’. We can specify the column formats like this:

AT&T Common Stock		
Year	Price	Dividend
1971	41-54	\$2.60
2	41-54	\$2.70
3	46-55	\$2.87
4	40-53	\$3.24
5	45-52	\$3.40

Table 13. example table

```
\begin{tabular}{r@{.}lr@{.}l}
```

to create two pairs of columns (that is, four columns in all). Each pair is really a column aligned on a decimal point. We therefore write the contents of the table like:

```
12&3      & 156&    \\
0&12345 &      &34 \\
```

in order to make 12.3 and 0.12345 align correctly. Note that we do not use the decimal point at all. We give the alignment position, and we have already specified the extra text to be added, the decimal point. The penalty we have paid is to introduce extra columns. This means that we have to be careful in specifying the column headings (or stubs), since they should span two columns.

The `\multicolumn` instruction makes this an easy problem to solve: a full table might therefore look like:

```
\begin{tabular}{|r|r@{.}l|r@{.}l|r@{.}l|}
\hline
$\lambda_{ij}$&&\multicolumn{2}{c|}{$L_{R-I}$}&&
\multicolumn{2}{c|}{$DAR$}&&
\multicolumn{2}{c|}{$L_{R-P}$}\\
\hline
70&4&60&6&80&5&10\\
80&10&70&12&10&11&20\\
\hline
\end{tabular}
```

λ_{ij}	L_{R-I}	DAR	L_{R-P}
70	4.60	6.80	5.10
80	10.70	12.10	11.20

This illustrates a deficiency. The three stub multicolumns should themselves be ‘gathered together’ under the heading ‘%BP’. Now this introduces an interesting point, since although we would like it boxed if we do it very simply:

```
\hline
&\multicolumn{6}{|c|}{\%BP}
```

we also obtain an extra empty stub cell in the top left corner:

	%BP					
--	-----	--	--	--	--	--

This does not look right. Instead, we can write

```
\cline{2-7}
\multicolumn{1}{c}{}&
\multicolumn{6}{|c|}{\%BP}
```

The new instruction is the `\cline`, which allows a horizontal line to be drawn from one column to another, in this case from column 2 through to 7. By adding the extra `\multicolumn` at the beginning, we suppress the first vertical rule.

Putting all this together yields:

AT&T Common Stock		
Year	Price	Dividend
1971	41-54	\$2.60
2	41-54	\$2.70
3	46-55	\$2.87
4	40-53	\$3.24
5	45-52	\$3.40

Table 14. example table, centred

	%BP		
λ_{ij}	L_{R-I}	DAR	L_{R-P}
70	4.60	6.80	5.10
80	10.70	12.10	11.20

QUESTION 6.5 Some would argue that T_EX and L^AT_EX users are over-fond of vertical and horizontal lines in their tables. Recreate the last table with the minimum number of lines you think is necessary.

L^AT_EX normally adds space to separate the columns, which includes adding space at the beginning and end of the row entries. When vertical rules are used this makes good sense, but when they are absent it does often make any horizontal rules a little longer, and sometimes ungainly. A way of suppressing the space at the ends is to use the `@` specifier and insert nothing: i.e. `@{}`. For example, Table 15 begins

```
\begin{tabular}{@{}lcc1@{}}
\hline
```

The alignment of the third and fourth columns may require some explanation. From the details given, it clearly does not attempt to align on the decimal points, and yet it manages to ‘centre’ the columns. Clearly something is going on. It is a feature of many (but not necessarily all) fonts that the width of numerical characters are the same. This is a boon for tables. Fortunately the width of a numeric happens to be `\enspace`, so this instructions was included where appropriate. We’ll look at the insertion of space later.

QUESTION 6.6 In the meantime, attempt to recreate Table 15, but do it by aligning on the decimal points. You need two new columns.

QUESTION 6.7 Tables 16 and 17 ‘contain’ the same information. Re-create both of them. Which is the easier to create? Which is the easier to interpret?

QUESTION 6.8 Mathematics can be included within tables. Table 18 is adapted from [3] (whence Knuth obtained the AT&T example). You ought to add a caption.

Boxing (and paragraphs)

Sometimes it is useful to box a chunk of text. We almost have the tools to do it. We have seen how we can have rules up the margins and across the top. All we need extra is a way of placing some text which will be treated in the ‘normal’ way, with respect to line breaking, paragraphing, and so on. There is an argument to `tabular` which will help, the `p`, or

	Symbol	Period (Solar hr)	Amplitude ($M_2 = 100$)	Description
Semidiurnal	M_2	12.42	100.0	Principal lunar
	S_2	12.00	46.6	Principal solar
	N_2	12.66	19.1	Larger lunar elliptic
	K_2	11.97	12.7	Luni-solar
Diurnal	K_1	23.93	58.4	Luni-solar
	O_1	25.82	41.5	Principal lunar
	P_1	24.07	19.3	Principal solar
Fortnightly	M_f	327.86	17.2	Lunar long-period

Table 15. principal tidal constituents

					Unemployed (1000s)			
					Total	Males	Females	
Thousands					1973	600	500	99
unemployed	1966	1968	1970	1973	1970	580	500	87
Total	330.9	549.5	582.2	597.9	1968	550	460	89
Males	259.6	460.7	495.3	499.4	1966	330	260	71
Females	71.3	88.8	86.9	98.5	Average	520	430	86

Table 16. exhibit A

Table 17. exhibit B

paragraph, option.¹ It requires an argument which describes the ‘width’ of the column into which the paragraph will be placed.

```
\begin{tabular}{|p{4in}|}
\hline
Letting a hundred flowers blossom and a
hundred schools of thought contend is
the policy for promoting the progress of
the arts and the sciences\dots\
\hline
\end{tabular}
```

Here we have specified a paragraph width of 4 inches, but we can choose any width we like (and if we `center` it, we can position it very nicely – Figure 1).

In Table 19 this paragraphing mechanism has been used for a rather long string of text. The important part runs:

```
\begin{tabular}{@{}p{0.5\textwidth}r1@{}}
```

The width I have used is `0.5\textwidth`. This needs a little elaboration. The width of the body text is known to L^AT_EX as `\textwidth`. The entry has been scaled to half of this width. Obviously here it is not taking up half the body text width, because I have enclosed this table in another environment (called `minipage` and which we’ll meet later) in order to be able to place the two tables side by side. The entry is scaled to half of the width of whatever width I let the `minipage` environment have. In this way it can adapt to whatever is the text width of the current style. The rest of it is rather nondescript, except perhaps to notice how the ‘paragraph’ lines up with the other row entry.

QUESTION 6.9 Since this gives an opportunity to experiment with a paragraph, and to incorporate some maths, tackle Table 19.

¹There is also a quite different mechanism, using `\fbox` or `\framebox` which will be examined elsewhere. Very often L^AT_EX provides more than one tool to do rather similar tasks.

Letting a hundred flowers blossom and a hundred schools of thought contend is the policy for promoting the progress of the arts and the sciences. . .

Figure 1. a helmsman speaks

Name	Definition
Gamma	$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$
Sine	$\sin(x) = \frac{1}{2i}(e^{ix} - e^{-ix})$
Bessel	$J_0(z) = \frac{1}{\pi} \int_0^\pi \cos(z \sin \theta) d\theta$

Table 18.

Diameter of sun	1.392×10^6	km
Diameter of earth	12,573	km
Diameter of moon	3,479	km
Mass of earth	5.98×10^{24}	kg
Mass of moon	7.34×10^{22}	kg
Mass of sun	1.96×10^{30}	kg
Average distance between centres of earth and moon	384,329	km
Average distance between centres of earth and sun	149,360,000	km

Table 19. earth, sun and moon dimensions

Chapter 7. Floating away

Consider structures which are not part of the ‘linear’ structure of a document. For example, footnotes are a sort of non-linear interpolation. If we alter the dimensions of our pages, the positions of footnotes change. In a similar sort of way, marginal notes run orthogonal to the text. And then we have the class of ‘floating’ material, such as tables and figures, which are commonly positioned at the top or bottom of pages, close to, but not exactly where they are referred to in the text. Again, this is because of the structure of physical pages. If we had a continuous scroll, then tables and figures could appear ‘anywhere’. Because they have to be an indivisible unit, there are only some places they may occur.

Footnotes

It must be obvious that footnotes are possible, since I have already used them². The instruction is fairly obvious:

It must be obvious that footnotes are possible, since I have already used them`\footnote{too frequently}`. The instruction is fairly obvious:

L^AT_EX takes care of the numbering, positioning and so on. If we inspect these notes, we will see that the footnotes are numbered consecutively within a chapter. Some document styles seem to want footnotes to be numbered sequentially on a per page basis. It is said that this can be done, but for all sorts of reasons (‘asynchronous output’ mainly), it is rather tricky. On the other hand, it is possible to have symbols used, rather than numerals. To do this issue, the instruction

```
\renewcommand{\thefootnote}
{\fnsymbol{footnote}}
```

in the preamble. This will be explained later. At the moment treat it as an incantation.

²too frequently

³please

Marginal returns

Less marginal in value are notes in the margins. These are obtained very simply with the instruction `\marginpar`: for example

```
\marginpar{Non-linear text}.
```

would place the text in the margin. There are a few additional instructions which allow a bit more control, notably for the width of the paragraph in which these marginal notes are set. There is also an interesting use of optional text. The default form, shown above will ‘normally’ place the text in the right margin. Changing the text to an option:

```
\marginpar[Non-linear text]{}.
```

would place the note in the left margin. However, this default activity is changed if the `twoside` option of the `\documentclass` is in use. Now the ‘normal’ placement of notes is in the left margin of even-numbered pages and the right margin of odd-numbered pages (quite logical, really). The other exceptions are that when `twocolumn` is in use, the notes go in the available margin (again, quite sensible). A further complication is that the default class used for *Baskerville* does not support marginal notes at all.

One other thing: don’t put too many footnotes³ and marginal notes in your text. They really do destroy the readability.

QUESTION 7.1 *Since I can’t demonstrate the use of `\marginpar` you should modify some of the previous examples to include marginal information, and place some in both margins.*

Tables and Figures

These two are so similar that they are best taken together. The only difference between

```
\begin{figure}
.
\end{figure}
and
\begin{table}
.
\end{table}
```

is in the nature of the caption which appears, and the fact that one appears in a list of figures, and the other in a list of tables.

Within these two environments, you may nest the positional environments such as `center`, and the table environment `tabular`. Most of the tables in these chapters start with something like:

```
\begin{table}[t]
\begin{center}
\begin{tabular}{|llp{3in}|}
```

so that we accumulate a tabular environment within a centered environment, within a table. In fact this isn't quite true: since we are in a multicolumn environment where we want the tables or figures to be positioned relative to the entire text width of the page, we have to use an asterisk form of the environment: `table*` or `figure*`. In fact, in the *Baskerville* class file used to typeset this article, the asterisk form is the only acceptable float, but in the standard `twocolumn` mode you may use either.

The optional argument for `table` is a positional indicator. The positional options are `h` for 'here' (right here on the page) – often rather tricky to achieve – `t` for 'top' (of a page), `b` for 'bottom' (of a page), and `p` for 'page' (that is, on a separate page, containing no text, but perhaps other figures and tables). The default is equivalent to

```
\begin{table}[tbp]
```

This is *not* interpreted in sequence. L^AT_EX attempts to place the floating table or figure where you specified (if the `h` specifier is used), then, if `t` is given, at the top of page; then, if `b` is specified, at the bottom of a page; failing that, if `p` is given, place on a page of other tables or figures (a 'float' page). Even if `p` is not specified, L^AT_EX sometimes fails to find somewhere satisfactory to accumulate the figures or tables, and positions them on a float page at the end of the 'chapter', or whatever is the appropriate logical entity. This usually means you have asked for something pretty demanding.

You do have some control over the floats, in that you can attempt to override L^AT_EX's selection. The first way is to emphasise the position by the specifier, `!` which is placed as a prefix to other positional specifications, `!b`, and is a sort of 'try harder' instruction. Now, this is visual formatting – that is to say, we are looking at the appearance of the document, and reacting. Normally you would not expect to insert this specifier until you were fine-tuning the document.

Another sort of control is given through the

```
\suppressfloats
```

instruction, which takes an optional placement argument (either `t` or `b`). This suppress *additional* floats on the current

page, deferring them to a later page. If a positional specifier is used it suppresses floats in that location, but not necessarily in the other one. If you have already used the `!` emphasis in your `table` or `figure` environment, then the `\suppressfloats` is suppressed at that time. Since you have no way of guessing where a float is likely to be positioned, this is again a reactive or fine-tuning instruction.

It is fine saying that you can override L^AT_EX's selection, but what is that likely to be, and what is it actually doing? L^AT_EX's float instructions do two sorts of things. They control the amount of floating material, in terms of the space it takes up on the page, and they control the total numbers of floats on a page. These controls are available for modification. The *Baskerville* class file contains instructions to change the default values to alternatives thought to work better for this newsletter. In a later chapter we will look at ways in which you can alter these values.

There is another mechanism which can be used for fine tuning. It might be used to assist in the placement of floats, but more likely it would be used to control smaller amounts of unwanted white space. The instruction

```
\enlargethispage{3ex}
```

will make the page height available for text 3 ex longer (about a line: if you knew the way of specifying the current distance between lines, or more correctly, baselines, you could have used that). You could use any other measurement in there too. There is a variant form which can be used to change the page height, but also has the effect of squeezing any available 'glue'. We have not really examined the notion of glue (it will be examined later), but often additional white space is left before sections and other structures. This additional space is notional 'glue'. These instructions must be used very cannily.

QUESTION 7.2 Collect all the examples you have done so far into a single file, but turn some of the tabular material into floating tables or figures. It is only when you have a few pages of material that you can start to experiment with the various positional specifiers and gain some feeling for the quirky but charming subtlety of L^AT_EX's positioning algorithm. You might also try to use the `twocolumn` option. If you do so, try using the asterisk form of the environments for some of the tables or figures.

To whom it refers

Table 19 in chapter 6 ends like this:

```
\end{tabular}
\caption{earth, sun and moon
dimensions}\label{solar}
\end{center}
\end{table}
```

There are important new elements introduced here – the `\caption` and the `\label`. These chapters make fairly extensive use of 'labels' to refer to the tables and figures. Until the document is run through L^AT_EX (a few times) I do not know what the table numbers will be. They are all symbolic references. Provided I remember the symbolic reference (`solar` here), I can use it elsewhere and we will get the correct numbering – 19. If you check back to Table 19 in

chapter 6 it should correspond correctly. You will also see how L^AT_EX inserts extra text, ‘Table’, which I have not included explicitly. In a `figure` environment the text would be ‘Figure’. In this situation, the `\label` must follow the `\caption`, or be part of it. Thus

```
\caption{earth, sun and moon
dimensions}\label{solar}}
```

should also be acceptable. Part of the reason for this is that we could easily have a Table or Figure with more than one caption. It therefore makes sense to have the caption and the label together, and no ambiguity arises. Careless L^AT_EXers disregard this at their peril. Your cross-references just won’t cross refer. Besides attaching a label to a caption, you may also attach it to a sectioning instruction. For example `\section{Introduction}\label{start-table}` would allow us to refer to the label, and generate the correct chapter and section number in the text. Again, the *Baskerville* class does not number sections, but many other classes do.

But how do we use the cross reference? The `\label` sets them up, and `\ref` places them in the text. Earlier I said

```
If you check back to
Table~\ref{solar} it should correspond
```

to have the cross reference used. Note the use of the tilde character `~`. This is a special instruction, one of the few which lack the introductory `\` character, which generates a ‘hard space’ (or a tie), one which may not be broken – that is, there may not be a line break after ‘Table’. This just looks a bit better. There are few things more distracting than a split between ‘Table’ and ‘19’, which could equally be over a page or column boundary. ‘Tie’ them together.

The `\label` and `\ref` strategy works more widely than this. A `\label` might be appended to a sectioning instruction

```
\section{Referring}\label{references}
```

but it can even be at some arbitrary point in your text. In such a case it will reflect the current sectional numbering. That the *Baskerville* class does not number sections will not however prevent me from referring to a section by its implied number.

Perhaps of more use is the `\pageref` instruction, which stores the page number where the reference is issued.

In these chapters I have labelled each table, figure, section, subsection, and question. I also stuck a label on the `\chapter` instruction which introduces each chapter. It should therefore be possible for me to say something like

“In Chapter 6, several tables, for example, numbers 16 and 17 occur in section 14, which starts on page 18. The tables themselves are on page 20.”

which in this instance was generated from

```
In Chapter~\ref{tablets}, several tables,
for example, numbers~\ref{hard}
and~\ref{easier} occur in
section~\ref{variations}, which
starts on page~\pageref{variations}. The
tables themselves are on
page~\pageref{hard}.
```

Note how I have managed to refer to a section which is itself unnumbered.

QUESTION 7.3 Revisit question 7.2 and incorporate captions and labels into your floating material. The bolder might like to refer to them in the text.

QUESTION 7.4 Place the instruction `\listoffigures` or `\listoftables` at the beginning (or end) of your text. You will have to run the file through L^AT_EX a couple of times before anything interesting happens.

Citations to bibliographic references are similar to references within the text, except that they use the instruction `\cite` and really require a separate (bibliographic) database to resolve. This is a subject we will attack later. Indexes may also be created, and will also be tackled later.

In the next block we’ll also find out how to cross reference equations. They follow the same sorts of rules, but have some special structures associated with them.

These labels are not at all like L^AT_EX instructions and do not follow the conventions for L^AT_EX instruction names: the text of the cross referencing labels may contain *any* sequence of alphabetic characters, numerals or punctuation characters, and we might therefore find the following:

```
\label{first}      \pageref{1st}
\pageref{1.2}     \ref{a3p}
\label{First}     \cite{ohms-law}
\label{1.5in}    \pageref{query?}
\ref{:end}       \pageref{*5}
```

Note that the text is case sensitive. Therefore `First` is different from `first`.

Since it is often difficult to remember all the labels you introduce, an auxiliary tool is provided with all L^AT_EX installations, `lablst.tex`. When you run this through L^AT_EX it will produce a list of all the labels you have used, with extra information which should allow you to find them in the text. This tool processes the `.aux` file. It produces a table of contents, interspersed with details of the labels used, the table or figure numbers to which they refer, and the page numbers on which they appear. This is essential to track down the position of those labels which the `.log` file keeps reporting as missing (usually because they were mistyped), and to remind you of the names you used.

Since it can also pick up other useful cross referencing information, such as equation numbering, we’ll defer a more detailed examination until we have covered that topic too.

References

- [1] R A Bailey. Maths in L^AT_EX: Part 1. Back to Basics. *Baskerville*, 4(4), 1994.
- [2] Leslie Lamport. *L^AT_EX A document preparation system*. Addison Wesley, Reading, Mass., 1985.
- [3] Michael Lesk. *Tbl – a program to format tables*. Bell Laboratories, 1979.
- [4] N A F M Poppelier, E van Herwijnen, and C A Rowley. Standard DTDs and scientific publishing. *Baskerville*, 7(1), 1997.

UK-TUG Information

The 1998–99 UK-TUG committee

Chairman

Philip Taylor: P.Taylor@mail.rhnc.ac.uk

Treasurer & Membership Secretary

Peter Abbott: peter.abbott@tex.ac.uk

Secretary

John Palmer: johnp@bcs.org.uk

Kaveh Bazargan: kaveh@focal.demon.co.uk

Malcolm Clark: malcolm.clark@kcl.ac.uk

Roy Everett: rpe@anglianet.co.uk

James Foster: j.foster@sussex.ac.uk

David Hardy: dhardy@sunrise-setting.co.uk

Hong Ji: hong@focal.demon.co.uk

Phil Molyneux: molyneux@kingston.ac.uk

Kim Roberts: robertsk@oup.co.uk

Dominik Wujastyk: ucgadkw@ucl.ac.uk

Contacting UK-TUG

Please send UK-TUG subscriptions, and book or software orders, to the Treasurer: Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Fax/telephone: 0121-476 2159.

General enquiries should be sent to the Secretary: John Palmer, 69 St Cross Road, Winchester SO23 9RE. Telephone: 01962 865261.

UK-TUG maintains pages on the World Wide Web at <http://uk.tug.org/> E-mail enquiries about UK-TUG may be sent to uktug-enquiries@tex.ac.uk

Baskerville

Articles should be submitted using electronic mail to baskerville@tex.ac.uk; contributors who cannot submit using e-mail may seek advice from the Secretary. Any correspondence concerning *Baskerville* may also be addressed in this way. Earlier issues of *Baskerville* can be found on CTAN in usergrps/uktug/baskervi.

Book Discounts for UK-TUG members

We have arrangements with Addison-Wesley for their well-known T_EX-related publications, with International Thomson Publishing for the O'Reilly & Associates Inc. series, and with Oxford University Press, to supply books to members at discounted prices.

Please send details (including ISBN if possible) of books required to Peter Abbott. Peter will supply a quotation.

We are allowed to offer this service only to **current** members of UK-TUG and/or members of TUG. Please send your order and cheque (in UK £) to Peter Abbott. Cheques should

be made payable to 'UKTUG'. All books will be routed through UK-TUG. *In all cases*, please notify Peter Abbott by e-mail, 'phone, fax or letter when books are delivered. Provided that the book(s) are in stock, it will normally take about a week from receipt of order to delivery of the book(s).

Obtaining T_EX from CTAN

The UK T_EX Archive at <ftp://ftp.tex.ac.uk/> is part of the CTAN (Comprehensive T_EX Archive Network) collaborating network of archives on the Internet organised under the aegis of the T_EX Users Group.

The CTAN archives run an enhanced *ftp* server which supports dynamic compression, uncompression, and archive-creation options. Fetch the top-level file [README.archive-features](#) for information. The server also supports site-defined commands to assist you. Please read [README.site-commands](#) for a brief overview.

For interfaces to, and catalogues of, CTAN, you are referred to Graham Williams' *T_EX and L^AT_EX Catalogue* which is available from CTAN as help/Catalogue/catalogue.html; other interfaces are listed in <http://www.tug.org/interest.html>.

Please report any problems with the CTAN archives to ctan@ctan.org

Obtaining T_EX on CD-ROM

UK-TUG, in collaboration with GUTenberg, TUG, NTG and DANTE e.V., have produced T_EX Live, a "plug-and-play" CD-ROM for Linux, Unix, and Windows 32, based on the Web2c T_EX setup. As it is formatted according to ISO 9660, with both Joliet and Rock Ridge extensions, the platform-independent files can, in principle, be read on all operating systems which can deal with that format. The CD is supplied free of charge to all members of UK-TUG.

Complete DOS, Windows 32 and Macintosh implementations are available as packages on the CD; floppy disk sets will now be supplied only on special request.

Future meetings of UK-TUG

It is planned to hold two meetings in 1999, one on May 28th and one, including the AGM, before the end of September. The spring meeting will be on "T_EX for Windows 32", and the autumn one on the relationships between T_EX and L^AT_EX on the one hand, and SGML, HTML, XML and related notations on the other. We plan to hold the AGM in London, as this seems likely to maximise attendance.

For further information please contact one of the Committee, or visit the [UK-TUG Web site](#).

The Committee of the UK T_EX Users' Group