ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Bologna

INFN/TC-88/14
11 Aprile 1988

M.L. Luvisetto e E. Ugolini:

INTRODUZIONE ALL'USO DI TEX : LINGUAGGIO, TOOLS, INSTALLAZIONE.

Servizio Documentazione
dei Laboratori Nazionali di Frascati

Istituto Nazionale di Fisica Nucleare
Centro Nazionale Analisi Fotogrammi

INFN/TC-88/14
11 Aprile 1988

INTRODUZIONE ALL'USO DI TEX: LINGUAGGIO, TOOLS, INSTALLAZIONE

M.L.Luvisetto, E.Ugolini
I.N.F.N.–C.N.A.F., Bologna

Il presente articolo è concepito come introduzione all'uso di TEX in funzione della preparazione di articoli destinati alla pubblicazione sulle riviste specializzate. Sono illustrati esempi di impaginazione, di costruzione di macro istruzioni, formule matematiche e grafica. L'articolo comprende anche una descrizione dei "tools" complementari e delle relative funzioni. Vengono fornite indicazioni utili all'installazione e alla relativa personalizzazione in rapporto all'ambiente di utilizzo e in funzione della qualità di stampa.

1. Introduzione

TEX è un programma di elaborazione testi matematici e scientifici sviluppato da D.E.Knuth dell'Università di Stanford [1] e distribuito dalla $\mathcal{AMS}$ (American Mathematical Society) che ne detiene il "copyright". Rispetto ad altri prodotti di gestione testi, TEX fornisce i seguenti vantaggi: la qualità del testo stampato, la trasportabilità dei testi, il costo modesto di distribuzione e la disponibilità del prodotto per molti elaboratori e sistemi operativi con garanzia di omogeneità dei risultati sulle diverse macchine.

TEX è stato concepito per la creazione di libri di elevata qualità specialmente per quanto riguarda la composizione di formule matematiche. Per questo motivo il prodotto è molto complesso e richiede la conoscenza di alcune nozioni e termini di composizione tipografica, che verranno introdotti in maniera graduale nei paragrafi seguenti.

2. Generalità

TEX è un compositore tipografico dedicato alla produzione di testi scientifici con particolare riferimento alla matematica. La gestione dell'impaginazione è automatica e le dimensioni di default sono relative al formato A4. I comandi di TEX sono introdotti sotto forma di sequenze di escape in cui il carattere \ (backslash) svolge la funzione di "escape character". I comandi possono essere

mescolati al testo purchè separati da uno o più blank. Non sono ammessi caratteri non stampabili (ad es. <TAB>). I comandi di TEX ***devono*** essere battuti rispettando il carattere della definizione che in generale è minuscolo, tranne alcune eccezioni quali \S per il simbolo di paragrafo.

Per la produzione di testi non specializzati non occorre specificare alcun comando, eccetto \bye per terminare l'esame del file. Al primo \bye incontrato TEX termina l'elaborazione. È sufficiente creare il testo con un editor e invocare TEX, che dovrebbe essere inserito come simbolo globale di sistema.

Ad esempio si abbia il file MYREPORT.TEX (dove .TEX è il tipo di default), per eseguire TEX e visualizzare l'uscita su di un terminale grafico, ad esempio DEC VT240, si avrà:

```
$ TEX MYREPORT
This is TeX, Vax/VMS Version 2.0.0 (preloaded format=plain 86.5.17)
DUA0:[MYDIR.PRIVATE]MYREPORT.DVI;4 [1] [2] [3]
Output written on DUA0:[MYDIR.PRIVATE]MYREPORT.DVI;4 (3 pages, 882 bytes).
Transcript written on DUA0:[MYDIR.PRIVATE]MYREPORT.LIS;4
$ TXMAPPER MYREPORT/OUT=240
....display grafico testo....
$
```

TEX scrive un file di log di tipo .LIS e il file di output di tipo .DVI. Il file .LIS contiene la stessa informazione che compare su video. Nel caso in cui non siano stati commessi errori, l'informazione indica il numero di pagine prodotte da TEX, sia come totale che singolarmente tra parentesi quadre durante l'elaborazione.

L'uso di TEX per documenti non specializzati è estremamente semplice, ma può diventare molto complesso se si vogliono gestire tabelle, impaginazioni particolari o creare macro complesse. Le note che seguono sono concepite come riferimento rapido per l'uso più comune di TEX e come introduzione immediata per il principiante, perciò alcuni concetti sono introdotti in modo semplificato. Per una conoscenza più completa occorre fare riferimento al manuale "The TEXbook".

3. Criteri funzionali di TEX.

Per l'uso correto di TEX sono necessari alcuni concetti base. Come si è già detto TEX non è un formatter ma un compositore tipografico, quindi opera secondo le regole della tipografia. Cercheremo di riassumere per punti le regole principali o che comunque devono essere rispettate per ottenere pagine di buona qualità.

TEX usa la terminologia tipografica per descrivere gli oggetti su cui opera; i termini più usati sono i seguenti. Si chiama ***stile*** la forma dei caratteri appartenenti ad una stessa famiglia, che quindi sono graficamente omogenei, per esempio i caratteri corsivi. Si chiamano ***fonti*** i caratteri che appartengono ad uno stesso stile tipografico. Si chiama ***corpo*** la dimensione dei caratteri. Come default viene usato il corpo 10, dove 10 indica il numero di ***punti*** tipografici di cui è composto il carattere più largo (o più alto) in quel fonte (ad es. la lettera M), con 10 punti tipografici pari circa a 3.5 mm. Dato che una pagina in formato A4 ha una dimensione utile di 6.5in in larghezza e 8.9in in altezza ci sono 469.755 punti per riga e 643.2 punti per pagina.

Per quanto detto sopra, dovendo produrre pagine di tipo tipografico, TEX non utilizza i caratteri esistenti nel device di output , possiede invece una vasta scelta di alfabeti e di simboli, cioè di fonti in diversi corpi. Tutti gli alfabeti vengono creati in forma grafica via software, perciò sono espandibili a piacere. Quando TEX elabora un testo, fa ricorso ad opportuni files (di tipo .TFM) che contengono

la descrizione geometrica dei caratteri relativi all'alfabeto in uso. Generalmente l'utente non deve avere nozione dei files .TFM a meno che non voglia alterare le scelte di default. In questo caso dovrà consultare le tabelle dei fonti disponibili e relativi corpi ed eventualmente usare un tool di visualizzazione su VT o LP non grafici. L'esempio "campionario fonti" invoca gran parte dei fonti disponibili e stampa il testo: ***ABC abc 123*** creando un campionario di fonti. Un caso tipico di uso esplicito di fonte è la scelta di un corpo più piccolo per le note a fondo pagina.

TEX considera il testo formato da "parole" separate le une dalle altre da blank o <LF>. Le parole vegono raggruppate in righe in modo tale da evitare di spezzare la parola su due righe (sillabazione–hyphenation). Se ciò non è possibile, le parole vengono spezzate secondo le regole inglesi che in molti casi funzionano bene anche in Italiano. Esistono comunque dei metodi che vedremo oltre per una sillabazione corretta. Un insieme di parole costituisce un paragrafo se separato dal resto del testo da una o più righe bianche o se termina con la sequenza `\par`. La dimensione della pagina determina il numero di righe che TEX riesce a "scrivere" su ciascuna pagina. Il comando `\vfill\eject` forza il salto pagina. È ovvio perciò che TEX gestisce l'impaginazione in maniera automatica, quindi quando si batte il testo le righe possono essere spezzate in modo qualsiasi e ***non si devono spezzare*** le parole. Occorre ricordare che l'unità minima a livello TEX è il capitolo.

TEX produce un file di output "device independent" (di tipo .DVI) che descrive con opportuni criteri la disposizione grafica dei caratteri nella pagina, perciò per produrre una stampa occorre un programma "device dependent" per ciascun device, il programma in questione è chiamato genericamente "spooler". A livello spooler l'unità minima è la pagina creata da TEX.

In tipografia i segni di punteggiatura appartengono alla parola che li precede perciò usando TEX ci si deve attenere a questa regola, altrimenti si rischia di trovare qualche riga che inizia con un segno di interpunzione.

Per quanto riguarda gli accenti, le virgolette etc. dato che non esistono tali caratteri nel set ASCII standard, si farà ricorso a particolari sequenze. Le più frequenti sono indicate nel paragrafo "Caratteri speciali".

Le pagine prodotte da TEX sono numerate in cifre arabe in basso al centro della pagina.

I paragrafi vengono "indentati", cioè la prima riga è spostata leggeremente a destra con spaziatura interlinea maggiorata.

TEX opera essenzialmente in 3 modi diversi: ***orizzontale***, quando agisce sulla riga, ***verticale***, quando agisce sugli spazi interlinea, ***matematico***, quando elabora formule matematiche. I due primi modi vengono gestiti in modo automatico mentre il terzo viene invocato esplicitamente.

Nei paragrafi seguenti compare un sommario dei comandi di uso più frequente, raggrupati per tipo.

4. Terminologia.

In TEX un comando viene definito come ***parola di controllo*** se è costituito da due o più caratteri, come ***simbolo di controllo*** se è costituito da un solo carattere. Nel primo caso il comando deve essere separato da ciò che segue con uno o più spazi, mentre nel secondo caso il blank non è richiesto. Dato che TEX non ammette numeri nei nomi dei comandi, in realtà il blank è necessario solo se il comando è seguito da una lettera, può essere omesso se il comando è seguito da un numero o da un altro comando. TEX possiede circa 300 comandi di basso livello detti ***primitive*** con i quali è possibile costruire comandi di livello superiore detti ***macro***. TEX possiede circa 600 macro alle quali si aggiungono tutte le definizioni dell'utente. Quando si invoca una parola di controllo i

di controllo i caratteri blank che seguono non vengono interpretati da TEX come spaziature tra parole ma come fine del comando, perciò se il comando stesso non prevede una spaziatura sarà cura dell'utente specificarla in uno dei modi indicati nel paragrafo "Spaziatura". La spaziatura può essere forzata anche racchiudendo la sequenza in un gruppo con l'uso delle graffe ({...}), imponendo un piccolo spazio (vedi "italic correction") con il comando (\/), imponendo un blank con il comando (\ ) oppure facendola seguire da un gruppo vuoto ({}). Un esempio è costituito dal logo TEX che viene invocato con la macro `\TeX`. Se nel testo compare la frase: "l'uso di TEX è...." per la gestione della spaziatura si potrà adottare una delle soluzioni seguenti:

a) `- l'uso di {\TeX} è...`
b) `- l'uso di \TeX\/ è...`
c) `- l'uso di \TeX\ è...`
d) `- l'uso di \TeX{} è...`

Come abbiamo visto nel paragrafo precedente, i caratteri usati da TEX sono definiti sotto forma di fonti. I nomi attribuiti agli stessi sono: ***roman*** per i caratteri normali, ***slanted*** per i caratteri normali con inclinazione, ***italic*** per i caratteri corsivi, ***typewriter*** per i caratteri del tipo macchina da scrivere, ***bold*** per il neretto, ***sans–serif*** per caratteri simili a quelli delle stampanti dei calcolatori.

Si chiama ***italic correction*** l'aumento di spaziatura che è necessario quando si passa da un fonte inclinato ad un fonte diritto per evitare il contatto tra l'ultima lettera obliqua e la prima diritta.

Si chiama ***magnification*** l'ingrandimento applicato a un singolo fonte o all'intero testo. Può essere specificata in vari modi ad esempio come `\magstep` (vedi il paragrafo "Unità di misura").

Per convenzione tipografica alcune lettere vengono scritte come unico carattere quando si trovano consecutive in una parola. Questa convenzione si applica alle sequenze `ff fi fl ffi ffl` e si chiama ***legatura***. Altre lettere invece, come ad esempio la coppia `AV` vengono scritte con una spaziatura inferiore rispetto a quella usuale, questa convenzione si chiama ***kerning***.

Con i comandi di TEX si possono costruire ***strutture a blocchi*** analogamente a quanto si fa con il Fortran nelle strutture IF...THEN...ELSE. I delimitatori di struttura sono: {...}, $...$, $$...$$. Le parentesi graffe sono usate per le strutture generiche mentre i dollari vengono usati in modo matematico. Viene definito modo ***text*** il modo matematico con cui si scrivono formule nella stessa riga del testo (delimitatori $...$). Viene definito modo ***display*** il modo matematico con cui vengono scritte le formule come righe a se stanti (delimitatori $$...$$). Quando si invoca il modo matematico le lettere ***A–Z*** e ***a–z*** si chiamano ***simboli ordinari*** e vengono scritte in corsivo. Gli operatori del tipo integrale, sommatoria, etc. si chiamano ***large operators***. Inoltre si chiama ***asse della formula***, la linea orizzontale immaginaria rispetto alla quale vengono allineati simmetricamente in verticale gli elementi che compongono la formula.

Si chiama ***macro*** una definizione costituita da primitive o da altre macro. L'insieme delle macro si chiama ***format***. Le macro ammettono dei parametri formali che vengono sostituiti dai valori degli argomenti al momento della chiamata.

Si chiamano ***registri*** quei comandi ai quali TEX accede per funzioni specifiche, quali il contatore delle pagine, il conteggio delle tolleranze, etc. I registri sono accessibili anche all'utente. I registri di TEX sono 256 per tipo. I tipi disponibili sono:

\count0	–	\count255	per interi con valore assoluto minore di 2^{31},
\dimen0	–	\dimen255	per le dimensioni (es. \hsize etc),
\skip0	–	\skip255	per la colla, (vedi impaginazione)
\muskip0	–	\muskip255	per la colla in unità matematiche,
\box0	–	\box255	per gestire le scatole. (vedi impaginazione)

I registri possono venire assegnati sia esplicitamente che implicitamente con i comandi \newif, \newcount, etc. il cui uso verrà illustrato nel paragrafo dedicato alla programmabilità di TEX.

Si chiama ***output routine*** l'insieme di macro preposte alla gestione del formato di impaginazione e cioè numerazione delle pagine, eventuale intestazione, etc.

5. Unità di Misura – Impaginazione

La dimensione della pagina standard definita per TEX è il formato A4 con 1 inch di margine su ogni lato, quindi le dimensioni utili orizzontale e verticale sono definite rispettivamente come:

```
\hsize=6.5in
\vsize=8.9in
```

Le lettere che vengono impaginate in queste dimensioni sono definite secondo le unità tipografiche riportate nella tabella seguente.

pt	punto tipografico (10pt = 3.5mm)
pc	pica (1pc = 12pt)
in	pollice (1in = 72.27pt)
cm	centimetro (2.54cm = 1in)

I caratteri vengono definti in punti tipografici. La dimensione di stampa usuale è di 10pt e viene anche definita come corpo 10. Con il corpo 10 lo spazio interlinea di default è di 12pt. I caratteri di TEX sono stati costruiti per una risoluzione grafica di almeno 200 dots/inch. Se si usa un device di stampa con formato diverso da A4 o con una risoluzione minore, la dimensione della pagina deve essere ridefinita. Ad esempio per un device con risoluzione di 780 dots/inch in orizzontale la larghezza della pagina non può superare i 3.9in, alrimenti le righe vengono troncate.

TEX consente di ridefinire i fattori di ingrandimento, sia a livello ***globale*** che a livello ***locale***. Il ***fattore globale*** viene definito con il comando `\magnification=valore`, dove valore è un numero del tipo 1200, 1500 etc. e viene calcolato moltiplicando per 1000 il fattore di ingrandimento, per esempio un ingrandimento di fonti del 20% rappresenta un fattore moltiplicativo di 1.2 quindi il valore della magnificazione è 1200. Il fattore locale viene applicato solo al fonte in questione ed è espresso come ***magstepn*** dove n è un numero compreso tra 0 e 6, compresi i valori 1/2 e 3/2. L'ingradimento effettivo è dato da $(\sqrt{1.2}^{2n} \times 1000$ perciò si avrà:

scaled \magstep0	$\longrightarrow$	1000
scaled \magstephalf	$\longrightarrow$	1095
scaled \magstep1	$\longrightarrow$	1200
scaled \magstep2	$\longrightarrow$	1440

etc., l'effetto della magnificazione è addittivo, perciò un font per cui sia stato specificato \magstep1 con una magnificazione globale pari a \magstep2 viene riprodotto con il valore \magstep3. I fattori di ingrandimento possono essere espressi in molti modi, i più comodi sono i seguenti:

```
\magnification=1440                    %ingrandimento di tutto il testo
\font\xxx=cmr10 scaled\magstep3        %ingrandimento del fonte xxx
```

La magnificazione iniziale *mi* (quella cioè relativa a \magstep0) assume valori diversi a seconda della risoluzione in dots/inch del device di stampa ed è espressa come $risoluzione/2 \times 1000$. I valori iniziali disponibili sono:

mi =	1000 per	200 dots/inch
mi =	1200 per	240 dots/inch
mi =	1500 per	300 dots/inch

quindi la formula dell'ingrandimento effettivo diventa $(\sqrt{1.2}^{2n} mi)$.

Con alcuni spoolers, la magnificazione globale può essere specificata, esattamente con gli stessi criteri, anche esternamente a TEX a livello di visualizzazione o di stampa. Si può specificare ***una sola magnificazione globale***, valida per tutto il testo, in entrambi i programmi. Una magnificazione può essere usata solo se esiste la famiglia di fonti relativa, perciò prima di specificare valori di ingrandimento ***fare sempre riferimento*** alle tabelle dei fonti.

Una considerazione importante che deve essere fatta usando la magnificazione globale è che tutte le dimensioni vengono ingrandite: lettere, spazi bianchi, dimensione della pagina. Se la magnificazione viene usata all'interno di TEX si può evitare la propagazione dell'ingrandimento usando il prefisso "true" per l'unità relativa alla dimensione che non deve essere ingrandita, perciò se si è indicata una larghezza pagina con il comando \hsize=5cm e si è applicato il fattore di ingrandimento 1.2 con il comando \magnification=1200, la dimensione reale della pagina sarà di $5 \times 1.2 = 6cm$, se si vuole conservare invece la larghezza di 5cm si userà il comando \hsize=5truecm. Ovviamente queste distinzioni non sono possibili se il fattore 1200 viene specificato a livello di stampa, dato che TEX ha già calcolato tutte le dimensioni in valori assoluti.

Esistono comandi per gestire la spaziatura sia verticale che orizzontale, ad esempio \vskip e \hskip nei quali si deve specificare l'entità della spaziatura usando una delle unità ammesse da TEX. L'unità scelta dipenderà dall'entità della spaziatura stessa, perciò se si vuole aumentare lo spazio interlinea si useranno i punti tipografici, se invece si vuole lasciare dello spazio libero per una figura si useranno i cm.

La sintassi dei comandi di spaziatura è la stessa per ogni comando e precisamente il comando viene seguito dall'entità di spostamento espressa in una delle unità di misura valide per TEX ed eventualmente accompagnata dai valori di tolleranza come illustrato di seguito.

Esistono inoltre unità relative al fonte in uso che servono per la gestione di piccoli spazi bianchi sia in verticale che in orizzontale. Per una descrizione completa vedere il paragrafo "Spaziatura". Le definizioni base sono ***em*** ed ***ex***, rispettivamente per orizzontale e verticale. I valori assoluti dipendono dal fonte in uso sia come stile che come corpo e sono calibrati per conservare l'estetica della pagina. Indicativamente nel fonte \rm i valori sono: 1 $em = 10\ pt$ (storicamente: larghezza della lettera M), 1 $ex = 4.3\ pt$ (storicamente: altezza della lettera x), i numeri sono larghi .5 em.

Il comando \quad inserisce tra le parole uno spazio bianco di 1em, \qquad corrisponde a 2em. Qualsiasi altra spaziatura orizzontale può essere inserita con il comando \hskip.

L'impaginatura è il problema più arduo da risolvere se si vuole ottenere un risultato di buona qualità, perciò TEX fornisce moltissime possibiltà per la gestione dello spazio bianco tra le parole e tra le righe. Inoltre questo spazio può essere definito come valore assoluto, cioè TEX deve lasciare lo spazio specificato ***inalterato***, oppure come valore limite, sul quale cioè TEX può applicare delle tolleranze in più o in meno per poter gestire la pagina in modo da evitare inestetismi quali righe o parole isolate. Per specificare questi due tipi di spaziatura, ***fissa*** o ***variabile***, occorre assegnare oppurtunamente i limiti di tolleranza al comando relativo. Vediamo un esempio con il comando \vskip. La definizione più generale è: `\vskip 12pt plus2pt minus3pt` con cui si specifica una spaziatura verticale pari all'interlinea (12pt) che può essere aumentata di 2pt o diminuita di 3pt se una tale variazione si rendesse necessaria per una migliore impaginatura. Se uno o entrambi i termini di tolleranza mancano, si inibisce la corrispondente variazione. Per un sommario dei vari comandi di spaziatura disponibili si rimanda al paragrafo specifico.

Per la gestione delle righe e delle pagine TEX fa uso del concetto piuttosto complesso di ***scatole*** e di ***colla***, che qui illustreremo solo nelle linee essenziali. Una scatola viene definita da tre dimensioni che sono: altezza, larghezza e profondità. TEX considera un'immaginaria linea di base sulla quale porre le "scatole", le cui tre dimensioni rappresentano, rispetto alla linea di base, rispettivamente: la parte sopra alla linea di base (altezza), la parte sotto alla linea di base (profondità), mentre la larghezza non ha legame diretto con la linea di base. La colla viene definita da tre entità che sono: spazio, estensibilità e comprimibilità. Il testo gestito da TEX è quindi visto come un insieme di scatole collegate tra loro dalla colla. Ogni carattere è di per sè una scatola. Le righe formano scatole orizzontali, le pagine costituiscono scatole verticali. Le scatole sono unite fra loro da colla orizzontale e verticale. Per una disposizione ottimale, la colla gode delle due proprietà di estensibilità e comprimibilità, per cui lo spazio tra "scatole" può essere aumentato o diminuito secondo necessità. Se i valori di estensibilità e/o comprimibilità vengono annullati, la colla si riduce al valore rigido dello spazio. Per una gestione corretta di questi valori TEX fa uso di parametri quali la tolleranza, le penalità etc. per i quali si rimanda al manuale di Knuth.

Per esemplificare il concetto di scatola, esaminiamo la definizione del logo TEX:

```
\hbox{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125em X}
```

con questa macro si istruisce il programma a definire una scatola orizzontale (comando \hbox più esterno) in cui scrivere la lettera T, diminuire lo spazio tra lettere di .1667em (\kern), inserire una ulteriore scatola orizzontale (comando \hbox{E}) la cui linea di riferimento deve essere abbassata di .5ex (comando \lower), ripristinare la posizione della linea di riferimento (ciò avviene perchè \lower si applica ad \hbox e i delimitatori {} indicano che la scatola contiene solo la lettera E) ed inserire la lettera X avvicinata di .125em.

Sempre basati sul concetto di colla e di scatole esistono i due comandi \llap e \rlap che letteralmente significano "left overlap" e "right overlap" e che consentono di alterare la spaziatura orizzontale tra singole lettere. Per esemplificarne l'uso, consideriamo il problema di scrivere il simbolo $\hbar$ in un testo senza ricorrere ai fonti matematici. Potremo ottenere il risultato voluto in uno dei due modi: `\rlap/h` oppure `/\llap{h}` con il seguente risultato $\not h$.

Per esemplificare il concetto di colla riprendiamo in esame il comando di \vskip, che è stato definito come: `\vskip 12pt plus2pt minus3pt`. In questo modo viene specificata una colla verticale con spazio 12pt, estensibilità 2pt e comprimibilità 3pt. Quando TEX incontra un comando del

genere inserisce uno spazio di 12pt (colla verticale), sapendo che tale spazio può essere ridotto di 3pt oppure ampliato di 2pt, a seconda delle necessità di impaginazione.

Un paragrafo viene definito come un insieme di righe che terminano con un comando di colla verticale. In generale la fine del paragrafo viene designata dal comando \par oppure da una o più righe bianche. Esistono inoltre moltissimi comandi di spostamento verticale che emulano la fine del paragrafo. Un paragrafo può essere anche generato con l'uso di opportuni comandi di spostamento orizzontale e precisamente \parindent, \noindent, $$ che determinano la creazione di nuove scatole verticali. I comandi di indentazione fanno sì che il testo che precede termini il paragrafo in corso e si apra un nuovo paragrafo con indentazione specificata o nulla. I caratteri $$ invece aprono una sequenza matematica che deve essere scritta come paragrafo nuovo fino a che non vengono incontrati i due caratteri $$ di termine della sequenza matematica. TEX ha un limite di memoria di circa 200 righe piene per paragrafo. Se si termina il paragrafo con \par\filbreak si chiede a TEX di spezzare la pagina in questo punto a meno che nella pagina ci sia abbastanza spazio per inserire il testo fino al prossimo \filbreak, perciò, se i paragrafi non sono troppo lunghi, tutte le pagine prodotte termineranno alla fine di un paragrafo.

La divisione delle parole in righe viene gestita in modo automatico e TEX comprende di default delle regole di sillabazione per la lingua inglese e fornisce dei metodi per specificare una sillabazione diversa. Se si usa la dimensione della pagina stabilita da TEX, normalmente non vi sono problemi. Se, invece, si deve specificare una dimensione ridotta per poter utilizzare un device di stampa a bassa risoluzione, specie scrivendo in Italiano, si ottiene il warning "overfull box" che indica che una riga è troppo lunga per essere spezzata, cioè TEX non è riuscito a calcolare la colla in modo tale da spezzare le parole in accordo ai valori in corso per le tolleranze. Inoltre il file di output contiene un tassello nero in fondo ad ogni riga che ha dato origine a "overfull". Un sistema per evitare tali messaggi è l'aumento della tolleranza al valore \tolerance=1600. In questo modo i casi di overfull vengono ridotti drasticamente ma non eliminati. Si potrebbe porre allora il parametro \tolerance=10000 (praticamente infinito). In questo caso però compaiono dei warning di "underfull box" che indicano una riga troppo vuota. Per eliminare anche questi messaggi si può optare per il comando \raggedright con il quale le righe risultano disallineate sul margine destro. Se si vogliono conservare le righe del file sorgente, si usa il comando \obeylines.

Se nel testo compaiono delle parole che non devono essere divise su due righe, si sostituisce lo spazio bianco con il carattere di legatura ~ (tilde), si avrà ad es. Appendice~A, asse~x, etc. Se si vuole definire come inseparabile un insieme di caratteri in cui non compaiano spazi, si ricorre al comando \hbox come segue: Fig.~\hbox{8-A}, pagg.~\hbox{15-33}, etc.

Esiste il comando \hfil per la gestione della spaziatura lungo la riga. Ad esempio, se si vuole terminare una riga senza terminare il paragrafo si userà \hfil\break. Per un esempio delle possibilità offerte da \hfil si rimanda agli esempi di impaginazione. L'uso più frequente di \hfil si ha nella centratura degli elementi che costituiscono le colonne di un tabella. Vediamone un esempio in combinazione con il comando \line. Questo comando fa sì che il testo che segue occupi una riga a sè. Se \hfil segue il testo, si ha allineamento a sinistra, se \hfil precede si ha allineamento a destra, se \hfil precede e segue il testo viene centrato. In pratica \hfil indica completamento orizzontale della riga con spazi bianchi. Nel nostro esempio, il sorgente TEX è il seguente:

```
{\bf
\line{Text on the left.\hfil}
\line{\hfil Text on the right.}
```

```
\line{\hfil Text in the middle.\hfil}
}
```

Il testo prodotto è:

Text on the left.

Text on the right.

Text in the middle.

Per lasciare dello spazio libero per eventuali figure esistono tre comandi e precisamente:

`\topinsert`	per inserire lo spazio all'inizio della pagina,
`\midinsert`	per inserire lo spazio a metà pagina,
`\pageinsert`	per lasciare libera l'intera pagina,
`\endinsert`	per terminare il comando di inserimento.

All'interno del comando possono essere inserite le didascalie relative alla figura da inserire. Ad esempio se si vogliono lasciare 5cm di spazio bianco all'inizio di una pagina si avrà:

```
\topinsert\vskip 5cm\endinsert.
```

Le note in fondo alla pagina sono gestite in modo molto semplice dal comando `\footnote` che gestisce sia il numero della nota che il testo della medesima. La nota viene scritta in fondo alla pagina prima della numerazione separata dal testo da una riga come indicato di seguito. Se si desidera un corpo diverso da 10, lo si deve invocare esplicitamente nel corpo della nota stessa. La forma più generale è: `\footnote{no_nota}{testo_nota}`. Se invece di un numero si usa * , detto simbolo viene posto automaticamente come apice. I numeri devono essere espressi in modo matematico. Ad esempio:

```
\footnote*{Nota in fondo pagina}          %simbolo di richiamo *
\footnote{**}{Nota in fondo pagina}       %simbolo di richiamo **
\footnote{$^{12}$}{Nota in fondo pagina}  %no.  di richiamo 12
```

Il comando `\footnote` si invoca nel punto del testo in cui si vuole inserire il richiamo. Ad esempio il seguente frammento di TEX:

```
... il progetto di ricerca XYZW \footnote{$^{12}$}{supportato da KRML}
\'e di grande importanza ....
```

produrrà la stampa che segue:

.... il progetto di ricerca XYZW[12] è di grande importanza ....
....
....
....

[12] supportato da KRML

Una gestione altrettanto automatica è fornita per la creazione di liste enumerate di argomenti. Si usa il comando \item per la lista principale e il comando \itemitem per i sotto casi. Ad esempio, per produrre la lista:

1. Questa è una lista di argomenti con una sottolista, di cui questa è la prima voce.
   a) Caso 1.a della lista
   b) Caso 1.b della lista
2. Questa è la seconda voce della lista.

sono stati usati i comandi che seguono:

```
\medskip
\item{1.} Questa \'e una lista di argomenti con una sottolista, di cui
questa \'e la prima voce.
\itemitem{a)} Caso 1.a della lista
\itemitem{b)} Caso 1.b della lista
\item{2.} Questa \'e la seconda voce della lista.  \medskip
```

Per una spaziatura simmetrica tra l'inizio e la fine della lista enumerata si consiglia di iniziare e terminare il comando \item con la spaziatura verticale generata dalla sequenza \medskip. Se non si termina il comando \item con un comando di spaziatura verticale, il testo che segue continua ad essere gestito come argomento di \item.

L'aspetto finale della pagina viene determinato dai valori assegnati a \hsize e \vsize e dalle funzioni svolte dalla routine di output in uso. Come già detto la dimensione di default è relativa al formato A4 con 6.5in di larghezza utile e 8.9in di altezza utile con un margine di 1in su ogni lato. Se si vogliono alterare questi due parametri, lo si deve fare una sola volta per tutto il testo prima dell'impaginazione, altrimenti si potrebbero avere sorprese nella stampa finale (ad es. pagine vuote, etc.). Esistono altri due parametri con cui si può variare la marginatura e precisamente \hoffset e \voffset. Ad esempio se si pone \hoffset=.5in e \voffset=1.5in, si sposta la pagina di .5in a destra e di 1.5in verso il basso rispetto alla sua posizione normale.

Se non si vuole la numerazione delle pagine esiste il comando \nopagenumbers. I comandi \headline e \footline consentono di inserire righe di testata e di fine pagina personalizzate. Il comando \pageno consente di specificare il valore iniziale del numero della pagina, per esempio \pageno=100 assegnerà il no. 100 alla prossima pagina stampata. Se si vogliono numerare le pagine in cifre romane si inizia il manoscritto con \pageno=-1. Si consiglia di consultare il TEXbook per esempi di impaginazione diversa dallo standard e per l'uso delle routine di output per la gestione di indici, tavole sinottiche, etc. Il seguente codice TEX produce una intestazione delle pagine diversa per destra e sinistra.

```
\nopagenumbers   %elimina la numerazione standard delle pagine
\headline={\ifodd\pageno\rightheadline \else\leftheadline\fi}
% definisce \headline come comando che invoca \rightheadline per no.  pagina
% dispari, altrimenti invoca \leftheadline
\def\rightheadline{\tenrm\hfil INTESTAZIONE DESTRA\hfil\folio}
\def\leftheadline{\tenrm\folio\hfil INTESTAZIONE SINISTRA\hfil}
% il comando \folio inserisce il numero della pagina, \tenrm invoca il fonte
% \rm, i comandi \hfil allineano l'intestazione a destra e a sinistra
\voffset=2\baselineskip    %marginatura iniziale pagina
```

6. Composizione di Formule Matematiche

TEX ha la capacità di elaborare formule matematiche complesse descritte in modo semplice usando il linguaggio naturale matematico. Il modo matematico viene invocato dal carattere $ per mescolare formule al testo, oppure da $$ per scrivere le formule come nuovo paragrafo centrato nella riga. In modo matematico la spaziatura tra i termini viene gestita automaticamente, quindi le formule possono venire scritte nello stile preferito dall'autore. Ad esempio $a+b$ è gestito da TEX nello stesso modo di $ a + b $ e di $a + b$, quindi l'autore potrà adottare la spaziatura che preferisce. Analogamente il fonte scelto è quello matematico, cioè corsivo in corpo 10 per le lettere, "roman" per i numeri e corpi più piccoli per gli indici, gli esponenti, etc. Esistono moltissimi comandi per la gestione delle formule, di cui i seguenti rappresentano un insieme scelto fra quelli di uso più frequente.

Le lettere greche vengono indicate dal loro nome rispettigamente minuscolo o maiuscolo preceduto da \, cioè ad esempio \alpha e \Gamma. Le lettere phi, theta, epsilon e rho esistono in due forme diverse, di cui una si invoca con il prefisso var. Esistono comandi per i simboli matematici quali radice, sommatoria, integrale etc. per i quali si rimanda al paragrafo "Simboli Matematici" e che sono listati nelle apposite tabelle del TEXbook. L'esponente si indica con ^, l'indice con _ cioè il termine a_{ij}^2 è scritto in TEX come a_{ij}^2, notare che gli argomenti costituiti da un'espressione vanno indicati come gruppo tra {}. Ad esempio il segno di derivata è invocato con il comando \prime, per la radice quadrata esiste il comando \sqrt, per le radici n–esime il comando \root seguito dall'ordine della radice, per la sommatoria il comando \sum, etc.

Per sottolineare o sovrasegnare un carattere o una formula esistono i due comandi \underline e \overline. TEX calibra la dimensione del segno di radice, come lunghezza e altezza in proporzione alle dimensioni delle variabili implicate. Se si scrive una formula in cui compaiono molte radici, i cui argomenti sono costituiti da lettere di diversa altezza, si può allineare verticalemente il segno di radice con il comando \mathstrut preposto all'argomento, si avrà quindi \sqrt{\mathstrut a}. Negli esempi sono riportate formule complesse, frazioni, coefficienti binomiali, sommatorie, integrali, etc.

Quando TEX scrive le formule sceglie i fonti e la posizione degli esponenti secondo un certo numero di stili, che sono: ***display*** per le formule isolate, ***text*** per quelle che compaiono in mezzo al testo, ***script*** per esponenti ed indici e ***scriptscript*** per esponenti e indici del secondo ordine. L'uso è automatico ma può essere invocato esplicitamente con le sequenze \displaystyle, \textstyle, \scriptstyle, \scriptscriptstyle. TEX costruisce le formule matematiche in modo grafico e allinea gli elementi che vi compaiono verticalmente e orizzontalemnte rispetto all' asse della formula.

Esistono sequenze per designare i vari tipi di delimitatori matematici (per una lista completa consultare il manuale TEXbook). Se occorrono parentesi di dimensione maggiore del normale si possono usare le sequenze \big \Big \bigg \Bigg come prefisso della parentesi, in questo modo si creano delle parentesi di misura via via crescente. TEX è però in grado di gestire automaticamente la dimensione del delimitatore se si usano i due comandi \left e \right come prefisso, ad esempio $1+\left(.....\right)$. I due prefissi devono essere bilanciati, esiste però il delimitatore nullo e precisamente {\left.} oppure {\right.} per rispettare il bilanciamento senza scrivere il delimitatore corrispondente. Il delimitatore nullo viene usato nella sequenza \cases con cui si indicano valori alternativi di una stessa variabile (Vedi esempi). TEX è in grado di gestire le dimensioni dei simboli matematici in funzione degli argomenti degli stessi. Ad esempio è in grado di variare la misura del

simbolo di radice in formule del tipo $\sqrt{1+\sqrt{1+\sqrt{1+x}}}$. Analogamente centra verticalmente le formule, ad esempio in caso di frazioni composte.

La gestione dei limiti delle sommatorie e degli integrali in modo automatico produce i seguenti risultati. Nello stile "text" i limiti vengono scritti a lato del simbolo, nello stile "display" vengono scritti sopra e sotto solo per la sommatoria. Se si vuole alterare la disposizione dei limiti esistono i due comandi `\limits` (limiti sopra e sotto) e `\nolimits` (limiti a lato).

In matematica è consuetudine scrivere con caratteri corsivi le variabili, le funzioni generiche e i simboli quali $x, y, f(x), g(x)$ e simili, mentre funzioni specifiche come seno, coseno, etc. sono scritte con i caratteri normali. TEX fornisce una serie di comandi per gestire questa convenzione, perciò per esprimere la funzione seno non si scriverà sinx ma `\sin x`. Inoltre esistono due comandi per inserire dei puntini lungo una formula (ad es. $x_1 x_2 \ldots xn$). Con il comando `\ldots` i puntini vengono posti in basso, con `\cdots` i puntini sono centrati (ad es. nella formula $a_1 + a_2 + \cdots + a_n$ si userà `\cdots`). Esiste infine il comando `\ddots` per scrivere i puntini in diagonale e `\vdots` per scriverli in verticale. Questi comandi sono particolarmente utili per indicare elementi omessi nella scrittura di matrici.

Per la gestione dello spazio bianco in modo matematico oltre alle sequenze `\quad`, `\qquad`, etc. sono disponibili i seguenti comandi:

`\,`	spazio piccolo = 1/6 di quad
`\>`	spazio medio = 2/9 di quad
`\;`	spazio grande = 5/18 di quad
`\!`	spazio piccolo negativo = -1/6 di quad.

Oltre a questi simboli, esistono dei comandi di spaziatura (da invocare ***solo*** in modo matematico) che fanno uso delle unità matematiche ***mu*** e precisamente:

`\mskip`	spazio definito dall'utente (es. \mskip 9mu plus 2mu)
`\thinmskip`	pari a 3mu
`\medmskip`	pari a 4mu plus 2mu minus 4mu
`\thickmskip`	5mu plus 5mu

dove `\quad` è pari a 1em, mu (mathematical unit) è l'unità tipografica matematica corrispondente nell'uso al punto tipografico e vale ***18mu=1em*** nel fonte dei simboli matematici.

In generale TEX cerca di non spezzare le formule su più righe. Se ciò è inevitabile, la formula verrà spezzata solo dopo simboli come $= + < \times$, etc.

Il comando `\cases` gestisce in modo automatico la parantesi { che indica la scelta fra casi diversi. Si abbia ad esempio:

$$|x| = \begin{cases} x, & \text{per } x \geq 0; \\ -x, & \text{altrimenti.} \end{cases}$$

la sintassi di TEX per `\cases` è:

```
$$|x|=\cases{x,   &per $x\ge0$;\cr
-x,   &altrimenti.\cr}$$
```

dove \cr delimita le righe da elaborare come casi diversi, & indica la tabulazione. Dato che la sequenza esce dal modo matematico quando incontra il segno & di tabulazione,per le formule seguenti la tabulazione si deve invocare il modo text.

Per scrivere ***matrici***, TEX dispone delle due sequenze \matrix in cui le parentesi vengono indicate esplicitamente e \pmatrix in cui le parentesi sono implicite. Sia ad esempio:

$$A=\begin{pmatrix} x-1 & 1 & 0 \\ 0 & x-1 & 1 \\ 0 & 0 & x-1 \end{pmatrix}$$

usando la sequenza \matrix avremo:

```
$$A=\left(\matrix{x-1&1&0\cr
0&x-1&1\cr
0&0&x-1\cr}\right)$$
```

dove i comandi \left e \right fanno sì che TEX calcoli la dimensione corretta per le parentesi () scelte come delimitatori della matrice. Anche qui notare che & determina al tempo stesso la tabulazione e il contenuto di ciascun campo, mentre \cr chiude ciascuna riga della matrice. Se si usa il comando \pmatrix non si devono più specificare \left(e \right).

TEX è in grado di gestire in maniera molto semplice testo mescolato alla matematica, equazioni e formule complesse e variamente correlate, come illustrato negli esempi.

Se si vuole ***inserire del testo in una formula***, si definisce una scatola orizzontale contenente il testo voluto. Lo spazio bianco fra testo e formule deve essere gestito esplicitamente, mentre la spaziatura fra le parole del testo viene gestita in modo automatico. Se, ad esempio, si deve scrivere una frase del tipo:

$$x=na+b \qquad \text{per ogni} \quad a>b, \quad \text{altrimenti} \qquad x=na-b \qquad \text{per ogni } n\ge 0.$$

si avrà in TEX:

```
$$x=na+b\qquad\hbox{per ogni}\quad a>b,\quad\hbox{altrimenti}\qquad
x=na-b\qquad\hbox{per ogni $n\ge0$.}$$
```

In questo esempio sono riassunte le norme principali per la gestione di costrutti misti testo–matematica. La spaziatura tra testo e matematica viene gestita da \quad per spazi piccoli e \qquad per spazi maggiori. Il testo viene scritto come \hbox{....}, infine se nel testo compaiono formule che non debbano essere obbligatoriamente scritte in displaystyle (nel nostro caso $n \ge 0$) si possono scrivere all'interno di \hbox invocando il modo matematico text nella forma $.....$.

Altra norma utile è l'uso di \textstyle all'interno del modo display quando la formula è troppo piccola per giustificare l'uso di displaystyle. Ad esempio se si deve scrivere $y = 1/2x$, se ne può migliorare l'aspetto estetico scrivendo la formula come $${\textstyle y={1\over2}x$$}. Questo metodo può essere anche usato per impaginare più razionalmente formule molto complesse ed equazioni.

Per la gestione delle ***equazioni*** TEX fornisce le seguenti sequenze: \eqno, \leqno, \eqalign, \eqalignno, \leqalignno, dove le sequenze che iniziano per lettera "l" indicano allineamento con il numero della formula scritto a sinistra. Le sequenze \eqno e \leqno devono essere invocate alla fine della formula a cui assegnano il numero indicato. Ad esempio le due equazioni seguenti:

$$x^2+y^2=2x+y-7. \tag{3}$$

(3) $$x^2+y^2=2x+y-7.$$

sono state scritte con i seguenti comandi:

```
$$x^2+y^2=2x+y-7.\eqno(3)$$
$$x^2+y^2=2x+y-7.\leqno(3)$$
```

Il comando \eqalign produce una lista di equazioni allineate sul simbolo di uguaglianza o disuguaglianza. Se si hanno due gruppi di equazioni che si vogliono allineare su due colonne basta centrare le due scatole verticali definendo le formule con comandi separati di \eqalign con una spaziatura relativa di \qquad. Si possono inoltre inserire i delimitatori della giusta dimensione con l'uso dei comandi \left e \right. Gli esempi che seguono illustrano l'uso di \eqalign. Anche qui si fa ricorso al simbolo di tabulazione & e al comando \cr per la specifica dei campi da allineare.
Formula:

$$\begin{aligned} 4x^2+3y^2 &= 16, \\ 5x+y &= 7. \end{aligned}$$

TEX:

```
$$\eqalign{4x^2+3y^2&=16,\cr
5x+y&=7.\cr}$$
```

Formula:

$$\left\{\begin{aligned} x &= a+b \\ y &= z \\ w &= xy \\ v &= a-b \end{aligned}\right\} \qquad \left\{\begin{aligned} a &= 7.5\pi \\ b &= 1/2a \end{aligned}\right\}$$

TEX:

```
$$\left\{\eqalign{x&=a+b\cr y&=z\cr w&=xy\cr v&=a-b\cr
}\right\}
\qquad\left\{\eqalign{a&=7.5\pi\cr b&=1/2a\cr
}\right\}$$
```

Se durante un comando \eqalign si vuole alterare la spaziatura tra le righe, si può usare (***solo dopo un comando*** \cr) il comando \noalign seguito dal comando di colla verticale. Con \noalign si interrompe temporaneamente il processo di allineamento. Per saltare 3pt, ad esempio, si userà \noalign{\vskip3pt}.

I comandi \eqalignno e \leqalignno combinano l'azione di \eqno e \eqalign con numerazione della formula rispettivamente a sinistra o a destra. Infine esiste il comando \displaylines che consente di scrivere formule allineate arbitrariamente con o senza numerazione. La numerazione deve essere indicata prima del comando \cr della riga che si vuole numerare. Nel caso di \displaylines il posizionamento corretto viene fatto usando \llap perchè la riga viene centrata con il comando \hfil. Negli esempi che seguono abbiamo evidenziato la numerazione nella codifica TEX.
Di seguito diamo alcuni esempi di sequenze ***eqalignno*** e ***leqalignno***.

Formula:

$$2(x^4 - 6x^3 + 9x^2) = -3x^2 + 9x - 2$$
$$(x^2 - 3x)^2 = -3(x^2 - 3x) - 2; \tag{3}$$
$$x^2 - 3x + 1 = 0. \tag{4}$$

TEX:

```
$$\eqalignno{2(x^4-6x^3+9x^2)&=-3x^2+9x-2\cr    %no. a destra
{(x^2-3x)}^2&=-3(x^2-3x)-2;&(3)\cr               %no. 3
x^2-3x+1&=0.&(4)\cr}$$                           %no. 4
```

Se invece si vuole scrivere la numerazione a sinistra si sostituisce \eqalignno con \leqalignno, come segue:

```
$$\leqalignno{2(x^4-6x^3+9x^2)&=-3x^2+9x-2\cr   %idem no.  a sinistra
{(x^2-3x)}^2&=-3(x^2-3x)-2;&(3)\cr                   %no.  3
x^2-3x+1&=0.&(4)\cr}$$                               %no.  4
```

L'uso della sequenza ***displaylines*** e' indicato di seguito.

Formula:

$$x = y; \tag{1}$$
$$\text{se} \quad y = a \quad \text{allora} \quad x = a;$$
$$\text{e se} \quad z = x \quad \text{allora} \quad x = y = z = a. \tag{2}$$

TEX:

```
$$\displaylines{\hfill x=y;\hfill\llap(1)\cr
\hbox{se}\quad y=a\quad\hbox{ allora}\quad x=a;\cr
\hfill\hbox{e se}\quad z=x\quad\hbox{ allora}\quad x=y=z=a.
\hfill\llap(2)\cr}$$
```

La sequenza \eqalignno consente un uso più vasto di \eqalign. Con \eqalignno è possibile infatti spezzare la formula su due pagine e inserire testo tra le formule. Per segnalare a TEX un punto in cui dividere la formula ai fini dell'impaginazione si inserisce dopo un \cr il comando \noalign{\break}. Viceversa si può impedire la divisione specificando \noalign{\nobreak}. Inoltre si può inibire la divisione di tutta la sequenza definendola come \vbox e precisamente $$\vbox{\eqalignno{.....}}$$. Per inserire del testo nel corpo di una sequenza \eqalignno, si interrompe la sequenza dopo un \cr e si definisce \hbox per il testo, ad esempio la formula:

$$x = y + z + f(x)$$

dove

$$f(x) = \sin y.$$

viene scritta in TEX:

```
$$\eqalignno{x&=y+z+f(x)\cr
```

```
\noalign{\hbox{dove}}
f(x)&=\sin y.\cr}$$
```

cioè il testo viene inserito nella forma `\noalign{\hbox{....}}`.

Se una formula è tanto lunga da non poter essere spezzata su più righe in modo automatico, si può imporre la divisione inserendo i comandi `\cr&\qquad` nel punto voluto. Ad esempio:

$$x(i,j,k) = a_1(i,j,k)b_1(i,j,k) + a_2(i,j,k)b_2(i,j,k) + a_3(i,j,k)b_3(i,j,k) + a_4(i,j,k)b_4(i,j,k) + a_5(i,j,k)b_5(i,j,k)$$

viene scritta in TEX:

```
$$\eqalignno{x(i,j,k)&=a_1(i,j,k)b_1(i,j,k)+a_2(i,j,k)b_2(i,j,k)
+a_3(i,j,k)b_3(i,j,k)\cr
&\qquad+a_4(i,j,k)b_4(i,j,k)+a_5(i,j,k)b_5(i,j,k)\cr}$$
```

Non tutti i simboli matematici esistenti nei fonti hanno una macro predefinita con cui invocarli, tali simboli matematici speciali possono essere definiti con il comando `\mathchardef` seguito dal nome del simbolo e da un valore esadecimale di 4 cifre che è così composto: classe–famiglia–codice. La famiglia indica il tipo di fonte, la classe è definita secondo la tabella seguente, il codice è espresso in una tabella di simboli del TEXbook con l'indicazione della famiglia.

Tabella delle Classi di Fonti		
Classe	Significato	Esempio
0	ordinary	/
1	large operator	$\sum$
2	binary operation	+
3	relation	=
4	opening	(
5	closing	)
6	punctuation	,
7	variable family	x

La classe determina la gestione della spaziatura e la dimensione del carattere nella gestione automatica delle formule matematiche. Ad esempio un fonte definito "opening" o "closing" verrà ingrandito in funzione della formula su cui opera. Normalmente la classe viene assegnata a seconda della funzione svolta dal simbolo (es. ordinary), dalle tavole di fonti in TEXbook si deduce la famiglia che è indicata nella didascalia della tavola nella forma `\texfontn` dove n è il numero della famiglia (es. 2), il valore numerico del fonte è formato da una cifra esadecimale espressa nella colonna di destra nella forma "nx (es. "7x dove il carattere " sta per esadecimale), il valore di x viene letto nella colonna corrispondente al carattere in alto o in basso a seconda della riga in cui compare il carattere (riga in alto o in basso della doppia casella). Riportiamo una tabella di fonti realtivi a simboli speciali [2] a titolo esemplificativo.

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	⊡	⊞	⊠	□	■	▪	◊	⧫	"0x
'01x	⥀	⥁	⇌	⇋	⊟	⊩	⫴	⊨	
'02x	⇻	⇺	⇇	⇉	⇈	⇊	↾	⇂	"1x
'03x	↿	⇃	↦	↤	⇆	⇄	↰	↱	
'04x	⇝	⇜⇝	↢	↣	≑	≳	≳	⋛	"2x
'05x	⊸	∴	∵	÷	≜	≳	≲	⋚	
'06x	⩽	⩾	≼	≽	⋖	≦	⩽	⋚	"3x
'07x	\		⋥	⋤	⋗	≧	⩾	⋛	
'10x	⊏	⊐	▷	◁	⊵	⊴	★	⧫	"4x
'11x	▼	▶	◀			△	▲	▽	
'12x	⥱	⋚	⋛	⋚	⋛	¥	⇛	⇚	"5x
'13x	✓	⩡	⩜	⩜	∠	∡	⊲	∝	
'14x	⌣	⌢	⋐	⋑	⋓	⋒	⅄	⅄	"6x
'15x	⋋	⋌	⊆	⊇	≐	≑	⋘	⋙	
'16x	⌜	⌝	Ⓡ	Ⓢ	⫛	⨣	∽	⩲	"7x
'17x	⌞	⌟	✠	℄	⊺	⊙	⊛	⊖	
	"8	"9	"A	"B	"C	"D	"E	"F	

Simboli Speciali: Fonte **MSXM10**

perciò se vogliamo definire il simbolo `\bigstar` come appartenete alla classe 0 secondo la tabella TEXbook per la quale sia indicata la famiglia 2 (simbolo matematico), vediamo che il simbolo si trova nel gruppo 4x nella riga in alto in corrispondenza al valore 6, quindi la definizione sarà `\mathchardef\bigstar="0246`.

Se vogliamo definire nello stesso modo il simbolo `\blacktriangle`, vediamo che appartiene sempre al gruppo 4x ma nella riga in basso relativa al valore `E` perciò avremo `\mathchardef\AA="024E`.

7. Creazioni di Macro – Tabulazione – Capacità Grafiche di TEX

TEX consente di creare nuovi comandi personalizzati a partire dalle sequenze predefinite (primitve). Questi nuovi comandi si chiamano ***macro*** e vengono creati con la sequenza `\def`. Le macro possono essere ***annidate*** (cioè una macro può contenere a sua volta altre macro) e ***recursive*** (cioè una macro può chiamare se stessa). Le macro possono essere rigide o contenere parametri. La creazione di una macro è consigliata quando un'espressione compare molte volte lungo il testo. Esempi di macro di vario tipo sono illustrati nel paragrafo "Un Esempio Completo". Se le macro sono molte possono venire raggrupate in un unico file, ad esempio `MYMACROS.TEX` che viene inserito nel testo con il comando: `\input mymacros`. In questo modo è possibile creare librerie di macro.

La forma più generale di definizione è:

```
\def\macro_name#1#2#3...{macro_def con parametri#1#2#3...}
```

dove il simbolo # definisce un parametro formale con un massimo di 9 parametri. Sono ammessi solo i numeri e devono essere consecutivi, cioè avendo 2 parametri non è permesso dare loro le assegnazioni #3#5. Vediamo un esempio. Supponiamo che nel testo compaia molte volte la formula $ax + b$, si potrà definire la macro:

```
\def\abx{ax+b}
```

Perciò quando si incontrerà un testo del tipo "con $f(x) = ax + b$" si potrà scrivere "con `$f(x)=\abx$`". Se si vuole esprimere la stessa sequenza \abx in forma parametrica perchè ricorrono forme analoghe quali $a_1x + b_1$ etc. si può modificare la definizione precedente come segue:

```
\def\abx#1{a_#1x+b_#1}
```

In questo modo un testo del tipo "con $y_1 = a_1x + b_1$ e $y_2 = a_2x + b_2$" diventa "con `$y_1=\abx 1$ e $y_2=\abx 2$`". Si può ulteriormente complicare la macro rendendo parametrici anche a e b per poter gestire forme come $ax + b$, $cx + d$ etc. In questo caso la definizione diventerà:

```
\def\abx#1#2{#1x+#2}
```

La chiamata per il testo seguente "ponendo $ax + b$, $cx + d$, ..." diventa perciò "`ponendo $\abx ab$, $\abx cd$, ...`"

Nella dichiarazione della macro non possono comparire caratteri bianchi tra i parametri formali perchè qualsiasi carattere assume valore di delimitatore di argomento. Infatti se gli argomenti di chiamata sono più lunghi di un carattere devono essere chiusi tra graffe o terminare con il delimitatore definito nella dichiarazione della macro in questione. Ad esempio la definizione seguente:

```
\def\macrodef#1.  #2\par
```

indica che il primo argomento termina alla comparsa di ".", mentre il secondo argomento termina alla fine del paragrafo. Se una macro richiama un'altra macro che contiene a sua volta argomenti, questi ultimi devono essere indicati tra graffe. Ad esempio la sequenza predefinita \pmatrix fa uso di \matrix ed è definita come segue:

```
\def\pmatrix#1{\left( \matrix{#1} \right)}
```

La definizione di una macro può comparire in qualsiasi punto del testo purchè venga definita ***prima di essere invocata.*** Se in una macro compaiono dei raggruppamenti non bilanciati invece di usare le parantesi graffe che danno origine ad errore di sintassi si fa uso dei due comandi \begingroup e \endgroup, come nell'esempio seguente:

```
\def\begfortran{ \begingroup \tt\begingroup $$\vbox }
\def\endfortran{ $$ \endgroup \endgroup }
```

Questo problema si presenta quando si opera su una struttura del tipo:

```
\apertura_di_gruppo {corpo_del_gruppo} \chiusura_di_gruppo
```

in cui si voglia trasformare in macro l'insieme delle sequenze che costituiscono l'apertura e la chiusura del gruppo. Nel nostro esempio, la macro \begfortran definisce il corpo tipografico e la scatola verticale in cui andrà inserito il testo che rappresenta un frammento di codice Fortran, perciò in questa macro compaiono i raggruppamenti di nuovo fonte, inizio struttura verticale e modo matematico che dovranno essere bilanciati solo dopo aver inserito il testo, quindi in questa fase si hanno solo gruppi aperti. Il bilanciamento degli stessi avviene con la chiamata della macro \endfortran, in cui si hanno solo gruppi chiusi. Se, ad esempio, scrivessimo la definizione \endfortran utilizzando le parentesi graffe, avremmo:

```
\def\endfortran {$$}}}
```

per cui TEX considererebbe il gruppo { $$ } come struttura completa da assegnare alla definizione e troverebbe due parentesi graffe chiuse non bilanciate dopo avere completato la macro \endfortran con conseguente segnalazione di errore. Tale inconveniente viene evitato usando invece \endgroup.

Se compaiono più definizioni della stessa macro il valore assunto alla chiamata è quello dell' ultima definizione incontrata.

Esistono altri comandi di definizione oltre a \def e precisamente \font per l'assegnazione di un nuovo fonte (vedi "Unità di Misura"), \mathchardef e \chardef per definire il codice di un carattere che non compare sulla tastiera (vedi TEXbook e paragrafo "Composizione di Formule Matematiche"). A questi si aggiunge il comando \let che consente di cambiare il nome o di assegnare nomi alternativi ad una macro ad esempio \let\abx=\nuovaabx, fa sì che \nuovaabx produca gli stessi risultati di \abx. Infine esiste la possibilità di eseguire le macro condizionalemente. La sintassi è:

```
\if<condizione> <operazione per .true.>
\else <operazione per .false>\fi
```

dove \if inizia la struttura condizionale \fi la chiude e \else rappresenta una eventuale alternativa. Ovviamente \if e \fi devono essere bilanciati. Oltre a \if esistono forme composte quali \ifodd, \iftrue, etc. In generale il test opera su uno dei registri interni di TEX (vedi "Terminologia"). Ad esempio se il registro \count0 contiene il numero della pagina e se esistono due procedure \rightpage e \leftpage per la numerazione delle pagine dispari in alto a sinistra e delle pagine pari in alto a destra si avrà la seguente chiamata condizionale:

```
\ifodd\count0 \rightpage \else\leftpage \fi
```

Da quanto sopra è evidente che la generazione di macro complesse che richiamano a loro volta altre macro può essere un problema di difficile soluzione. Inoltre non tutte le sequenze di TEX possono essere richiamate all'interno di una macro, di conseguenza la scrittura di macro richiede spesso una fase di correzione. Per evitare che una macro scritta male o con gruppi non bilanciati determini la lettura del file fino alla fine, TEX considera illegale il comando \par nel corpo di una macro e ne termina l'elaborazione dando un messaggio d'errore. Esistono dei meccanismi che consentono di alterare questa modalità operativa per i quali rimandiamo all'apposito capitolo del TEXbook.

TEX consente la scrittura di ***tabelle*** in maniera abbastanza semplice. Per la gestione della tabulazione TEX possiede il comando \settabs e gli operatori \+ per la definizione del primo termine di ciascuna riga della tabella, & per la definizione delle tabulazioni successive e \cr per designare la fine di ciascuna riga. La tabulazione può essere definita per colonne o secondo prototipo. Ad esempio si voglia scrivere una tabella del tipo:

dato1.1	dato1.2	dato1.3
	dato2.2	dato2.3
dato3.1		dato3.3
dato4.1	dato4.2	dato4.3

il comando di TEX sarà:

```
{\settabs 3 \columns   %crea tabella a tre colonne
\+dato1.1&dato1.2&dato1.3\cr
\+&dato2.2&dato2.3\cr
\+dato3.1&&dato3.3\cr
\+dato4.1&dato4.2&dato4.3\cr }
```

Se un termine della tabella manca si indica solamente l'operatore di tabulazione. Usualmente per una spaziatura corretta tra la tabella e il resto del testo si definisce la tabella come scatola verticale, perciò il comando `\settabs` sarà espresso all'interno di una struttura del tipo `$$\vbox{\settabs ..... }$$` in cui il modo matematico (`$$`) gestisce automaticamente la spaziatura verticale tra testo e tabella.

Se la tabella non è formata da colonne tutte uguali si fa seguire il comando `\settabs` dal prototipo della riga più ampia. Nell'esempio seguente si è definito un prototipo per inserire istruzioni Fortran nel testo.

```
\tt{%usa i caratteri di tipo typewriter
$$\vbox
{\settabs\+\qquad\qquad&mmmmmmmmm\qquad&\cr   %Fortran sample line
\+&FUNCTION VDOT (X,Y,N)\cr
\+C\cr
\+&DIMENSION X(9),Y(9)\cr
\+C\cr
\+& XX= 0.\cr
\+& IF (N.LE.0)  GO TO 100\cr
\+& DO 9 I= 1,N\cr
\+9 &XX= XX + X(I)*Y(I)\cr
\+C\cr
\+100&VDOT= XX\cr
\+&RETURN\cr
\+&END\cr
}$$}
```

L'intera sequenza precedente deve essere chiusa in una struttura, se non si vuole che il fonte `\tt` rimanga attivo. Il comando `\settabs` ha effetto globale analogamente ai comandi di scelta fonte. Il comando `\cleartabs` annulla le tabulazioni eventualmente attive.

Oltre a `\settabs` che setta un tabulatore in modo simile a quanto avviene in una macchina da scrivere, TEX consente anche la generazione di tabelle secondo un prototipo dinamico tramite il comando `\halign`. Nel caso di `\settabs` il posizionamento di ciascuna colonna è rigido e TEX elabora una riga alla volta. Nel caso di `\halign` TEX deve leggere l'intera tabella per scoprire la dimensione delle colonne perciò `\halign` non è consigliabile per tabelle troppo lunghe. Inoltre `\settabs` setta una tabulazione ripetibile sotto controllo dell'autore, mentre tabelle generate con comandi separati di `\halign` hanno colonne di dimensione diversa, dato che le dimensioni vengono calcolate in modo dinamico per ogni nuovo `\halign`. La sintassi di `\halign` è del seguente tipo:

```
\halign{\indent#\hfil&
\quad#\hfil\cr
Uno & Due\cr  Tre & Quattro\cr}
```

in cui # indica la presenza di un argomento, & agisce come separatore di argomenti, `\cr` indica la fine della riga. La sequenza `\indent#\hfil&` indica il primo campo con indentazione e allineamento a sinistra, `\quad#\hfil\cr` indica il secondo e ultimo campo con spaziatura di 1 `\quad` e allineamento a sinistra. In questo modo si definisce un prototipo e questa parte dichiarativa della struttura della tabella è chiamata "preambolo".

Esistono altri comandi che servono per la creazione di tabelle molto complesse. Di seguito ne diamo un elenco con relativo significato e rimandiamo al paragrafo "Creazione della Tabella TABLE.TEX" per un esempio.

Il comando `\tabskip` gestisce la spaziatura tra le colonne della tabella, `\span` espande una colonna, `\offinterlineskip` elimina la spaziatura interlinea, `\omit` omette un termine o altera la struttura del preambolo, `\multispan` espande più colonne, `\hidewidth` altera l'allineamento descritto nel preambolo, `\noalign` consente di inserire spaziatura extra tra le righe nell'ambito della struttura della tabella (scatola verticale).

Infine TEX consente di aprire e chiudere files esterni su cui può leggere e scrivere. Con questo sistema si possono perciò preparare tavole sinottiche, indici, etc. I comandi sono `\openin \read \closin \openout \write \closout`.

TEX possiede capacità ***grafiche*** intrinseche che consentono di generare tasselli pieni e vuoti, di tracciare righe e di disegnare rettangoli. Inoltre figure più complesse possono essere generate sotto forma di bit map ed essere inserite come fonti speciali. Le figure grafiche di TEX vengono generate tramite la gestione dei due comandi `\vrule` e `\hrule` e la creazione di opportune scatole. Il paragrafo "Esempi di Grafica" dà indicazioni su ciò che si può fare. I comandi `\vrule` e e `\hrule` disegnano una riga rispettivamente verticale e orizzontale delle dimensioni specificate, ad esempio per produrre un tassello nero si avrà `\vrule height4pt width3pt depth2pt`. Se non si specificano dimensioni i valori di default sono:

	width	height	depth
`\hrule`	-	0.4pt	0.4pt
`\vrule`	0.4pt	-	-

Le dimensioni omesse nella tabella precedente non sono fisse ma dipendono dal contesto. Essenzialmente `\vrule` e `\hrule` producono lo stesso effetto, ma nelle scatole orizzontali si deve usare `\vrule`, mentre in quelle verticali si deve usare `\hrule`. Altra possibilità grafica è costituita dal comando `\leaders` che consente di generare scritte del tipo:

Begin . . . . . End
Begin _____ End

con comandi del tipo:

```
\line{Begin\leaderfill End}        %serie di puntini
\line{Begin\rulefill End}          %riga continua
```

dove `\leaderfill` e `\rulefill` sono definiti in termini di `\leaders` come:

```
\def\leaderfill{\leaders\hbox to 1em{\hss.\hss}\hfill}
\def\rulefill{ \leaders\hrule\hfill\ }      %blanks at beg & end of leaders
```

Il comando `\hss` gestisce la colla verticale per allargarla quanto serve a riempire lo spazio tra le parole di testa e di coda. Oltre a `\leaders` esistono anche `\cleaders` (puntini centrati) e `\xleaders` (puntini espansi).

Diamo infine un esempio di creazione di scatole in modo che il testo fornito come argomento del comando `\boxit` sia circondato da un rettangolo i cui lati distano di tre punti tipografici dal testo stesso.

Chiamata:

```
\boxit{Create box surrounded by ruled lines}
```

Create box surrounded by ruled lines

Definizione di \boxit:

```
\def\boxit#1{\vbox{\hrule\hbox{\vrule\kern3pt
\vbox{\kern3pt#1\kern3pt}\kern3pt\vrule}\hrule}}
```

Analizziamo la definizione di \boxit. La macro viene dichiarata con un argomento: il testo da inscrivere nel rettangolo. Prendiamo in considerazione il raggruppamento più interno:

```
\kern3pt#1\kern3pt
```

in questo modo si specifica che l'argomento (il nostro testo) deve essere preceduto e seguito da una spaziatura di 3 punti tipografici. Il testo è formato da scatole orizzontali, quindi per conservare tale spaziatura per tutte le righe di detto testo, metteremo questa definizione in una scatola verticale. Questa scatola sarà a sua volta circondata da una spaziatura superiore ed inferiore di 3pt. Due righe verticali vengono tracciate con i comandi \vrule all'inizio e alla fine della scatola. Questi comandi vengono a loro volta inseriti in una scatola orizzontale, prima e dopo la quale vengono tracciate le due righe orizzontali (\hrule). Il tutto viene infine chiuso in una scatola verticale.

In pratica la \vbox più esterna contiene tutta la struttura che è formata da un'unica scatola orizzontale, mentre la \vbox più interna presiede alla suddivisione in righe del testo. Ad un primo esame questa definizione può sembrare inutilmente complicata ma con un poco di pratica l'utilizzo delle scatole risulta ovvio. Per meglio comprendere i meccanismi interni di TEX è consigliabile fare delle prove eliminando alcuni degli elementi di \boxit, ad esempio i comandi di \vrule per vedere cosa cambia in stampa.

Un altro esempio grafico è fornito dalla sequenza che segue, in cui le definizioni \square e \upsquare fanno entrambe uso della definizione \sqr. Con questa macro si costruisce un quadrato con lato uguale al primo argomento e spessore del medesimo pari a 1/10 del secondo argomento (.#2) centrato verticalmente sulla linea di base. L'uso dei comandi \hrule, \vrule e della gestione delle scatole è strettamente analogo a quanto fatto per \boxit.

Definizione di \sqr:

```
\def\sqr#1#2{{\vcenter{\hrule height.#2pt
\hbox{\vrule width.#2pt height#1pt \hskip#1pt
\vrule width.#2pt}
\hrule height.#2pt}}}
```

La definizione \square produce un quadrato da utilizzarsi in modo matematico e quindi di diversa misura a seconda che si richiami la definizione in modo display, text, script o scriptscript. Il comando \mathchoice fa sì che ciascuna delle definizioni seguenti sia usata nei modi suindicati nell'ordine. Nel nostro esempio il lato del quadrato è di 3pt in display e text, 2.1 in script e 1.5 in scriptscript. Lo spessore dei lati è di 0.4pt in display e text, di 0.3pt negli alri casi.

Definizione di \square:

```
\def\square{\mathchoice\sqr34\sqr34\sqr{2.1}3\sqr{1.5}3}
```

La definizione \upsquare fa riferimento a \sqr per la costruzione di un quadrato di 7pt di lato e 1pt di spessore seguito da uno spazio di \, e alzato di .250ex. Perchè il comando \raise possa funzionare è necessario definire l'intera struttura sotto forma di scatole.

Definizione di \upsquare:

```
\def\upsquare{\hbox{\raise.250ex \hbox{$\sqr7{10}\,$}}}
```

L'aspetto visivo delle macro descritte è illustrato in "Esempi di Grafica".

Infine TEX consente di inserire comandi grafici specifici del device di destinazione facendo uso del comando \special il cui argomento è formato dai comandi grafici. Ovviamente questo metodo può essere utilizzato solo se lo spooler relativo al device è in grado di elaborare i comandi così inseriti e se gli stessi rispettano il protocollo grafico di detto device.

8. Programmabilità di TEX

Come abbiamo già detto, TEX possiede un insieme di registri accessibili all'utilizzatore sui quali possono essere eseguite operazioni di calcolo, di confronto e di stampa. In questo modo è possibile scrivere dei programmi all'interno di TEX. Dato che TEX lavora come interpretere l'elaborazione è molto lenta. Inoltre elaborazioni molto complesse possono determinare overflow interno e abort di TEX (vedi note in "Esempi di Grafica"). Le operazioni consentite sono di basso livello, quindi anche programmi molto semplici richiedono un numero elevato di istruzioni. Le operazioni si basano sul concetto di macro, sono consentiti annidamento e recursione.

Per evitare di ridefinire dei registri, è opportuno ricorrere all'assegnazione automatica definendo gli oggetti su cui si opera in uno dei modi seguenti:

\newcount	definisce un intero
\newdimen	definisce una dimensione
\newif	definisce una variabile logica

Le nuove assegnazioni effettuate con \newcount e \newdimen vengono segnalate nel log file di TEX. Ad esempio se nel sorgente di TEX compaiono i comandi:

```
\newcount\k \newcount\n
\newcount\b \newcount\o
\newdimen\diskwide\diskwide=9pt
\newdimen\diskhigh\diskhigh=5pt
\newdimen\diskvskip\diskvskip=3pt
\newdimen\towerwide\towerwide=5\diskwide
\newdimen\towerhigh\towerhigh=5\diskhigh
```

Il file di log contiene le seguenti informazioni:

```
\k=\count25
\n=\count26
\b=\count27
\o=\count28
\diskwide=\dimen16
\diskhigh=\dimen17
```

```
\diskvskip=\dimen18
\towerwide=\dimen19
\towerhigh=\dimen20
```

Le operazioni aritmetiche consentite sono:

`\advance`	somma algebrica
`\multiply`	moltiplicazione
`\divide`	divisione

Le operazioni di controllo sono:

`\loop`	inizia una struttura ripetitiva
`\repeat`	termina la struttura ripetitiva
`\ifnum`	confronto tra due interi
`\ifdim`	confronto tra due dimensioni
`\ifodd`	verifica di intero dipari
`\iftrue`	test logico per true
`\iffalse`	test logico per false
`\ifcase`	inizio struttura con più casi
`\or`	caso alternativo
`\else`	tutti gli altri casi
`\fi`	fine struttura
`\the`	espansione del registro secondo il tipo
`\number`	espansione di un intero in cifre arabe
`\romannumeral`	espansione di un intero in cifre romane
`\message`	scrittura di messaggio sul log file e su VT

Seguono esempi d'uso dei comandi suddetti.

`\newif\ifprime` definisce una variabile logica `prime` su cui sono ammesse le operazioni di confronto logico (`\ifprime`) e di assegnazione logica (`\primetrue` e `\primefalse`)

`\newcount\n` e `\newcount\d` definiscono gli interi `\n` e `\d`

`\newdimen\linelength\linelength=50pt` definisce la nuova dimensione `\linelength` con il valore di 50 punti tipografici

`\ifnum\n>\d` esegue if confronto tra i due interi `\n` e `\d`

`\multiply \n by \d` esegue il prodotto di `\n` per `\d` e memorizza il risultato in `\n`

`\advance \n by -\1` memorizza in `\n` il valore di `\n -1`

`\divide \n by -\d` divide `\n` per `-\d` e memorizza il risultato in `\n`

`\the\linelength` espande il valore di `\linelength`

`\vskip\linelength` lascia uno spazio verticale di `\linelength` (50pt)

`{\loop\ifnum\n>0 \domacro \advance\n by -1 \repeat}` esegue il comando `\domacro` fino a che `\n` non diventa nullo.

Per meglio illustrare l'utilizzo delle funzioni programmabili di TEX vogliamo creare una macro per la stampa dinamica dei numeri di Fibonacci. Il testo in linguaggio TEX è il seguente:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%
% fibonacci M.L.L. & E.U. %
%%%%%%%%%%%%%%%%%%%%%%%%%%
%
\newcount\k \newcount\n
\newcount\b \newcount\o
%
\def\fibonacci#1{1,~1,  \k=3 \n=1 \b=1
\loop\ifnum\k<#1 \o=\n \advance \o by \b {\number\o~,}
\b=\n \n=\o \advance \k by1 \repeat
\o=\n \advance \o by \b { e \number\o}}
%
First 26 Fibonacci numbers are:\par
\fibonacci{26}.
```

Si tratta di costruire una macro che stampi i primi n numeri di Fibonacci che soddisfano la seguente condizione:

$$F_n = F_{n-1} + F_{n-2}, \qquad \text{con } n \geq 2.$$

Nel nostro esempio iniziamo da 2 e poniamo come argomento della macro il valore finale di n, perciò quando scriviamo in TEX la frase: "`I primi 7 numeri di Fibonacci sono \fibonacci{7}`". In stampa avremo: "I primi 7 numeri di Fibonacci sono: 1, 1, 2 , 3 , 5 , 8 , e 13".

Per il calcolo abbiamo bisogno di 4 registri: `\k` per il contatore di loop, `\o` per F_n, `\n` per F_{n-1}, `\b` per F_{n-2}. Questi registri vengono definiti come `\newcount`. I primi due valori vengono stampati subito. La tilde (~) serve per evitare che un numero e la virgola vengano stampati su due righe diverse. `\k` viene inizializzato a 3, `\n` e `\o` a 1. Viene poi costruito un loop fino a che `\k` è minore dell'argomento (#1) in cui si esegue il calcolo e si stampa `\o`. L'ultimo numero viene stampato fuori del loop perchè preceduto dalla congiunzione (e). Come esercizio si consiglia di modificare la macro (che è stata sviluppata originariamente in lingua inglese) in modo che non venga stampata la virgola dopo il penultimo numero.

9. Esempi di Formule Matematiche

Come primo esempio prediamo in considerazione i simboli di sommatoria, integrale, limite, etc. e facciamo un confronto tra il modo ***text*** e il modo ***display***.

In modo text le formule sono scritte usando corpi più piccoli e disposizione diversa per esponenti, indici, segno di frazione, etc. Ad esempio per le frazioni si avrà $\frac{n+1}{3}$, $\binom{n+1}{3}$ e $\frac{x/2}{\pi}$, per le potenze avremo x^{2^y}, per le espressioni si ha $x = b$, $y = c$, $a = b^2 + c^2$, $n/\log n$, mentre per le sommatorie e per gli integrali si ha $\sum_{i=1}^{10} a_{ij}$, $(\sum_{k=1}^{n} a_k)$ e $\int_0^x \int_0^y dF(u,v)$ con dimensioni calibrate per contenere la formula nelle righe di testo. In modo display invece le formule hanno dimensioni maggiori e la

disposizione di indici, limiti, segni di frazione è differente e può essere programmata con `\limits` e `\nolimits`, come evidenziato nel seguente gruppo di formule.

$$\sum_{\substack{0\le i\le m\\ 0<j<n}} P(i,j)$$

$$\int_{-\infty}^{+\infty} x\,dx + \int_0^{\pi} y\,dy - \sum\nolimits_{i=1}^{n} a_i y_i$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=l-n}^{l} A_i B_j C_k$$

$$\iint_D dx\,dy$$

$$y = \int_{-\infty}^{+\infty} \sin 2\theta d\theta + \sqrt{a}x^2$$

$$\lim_{x\to 0}\frac{\sin x}{x} = 1$$

$$\binom{n}{\frac{1}{2}k}$$

$$\gamma + \nu \in \Gamma$$

Per quanto riguarda il modo text non diamo la codifica TEX, dato che è identica al modo display eccetto il delimitatore ($ invece di $$). Il codice TEX per le formule stampate nell'esempio precedente è:

```
$$\sum_{\scriptstyle0\le i\le m\atop\scriptstyle0<j<n}P(i,j)$$
$$\int_{-\infty}^{+\infty}xdx + \int\limits_0^\pi ydy -
\sum\nolimits_{i=1}^n a_iy_i$$
$$ \sum^n_{i=1} \sum^m_{j=1} \sum^l_{k=l-n} A_i B_j C_k $$
$$\int\!\!\!\int_D dx\,dy$$
$$y = \int\limits_{-\infty}^{+\infty}\sin2\theta d \theta + \sqrt ax^2$$
$$\lim_{x\to 0}{\sin x\over x}=1$$
$$n \choose {1\over2}k$$
$$\gamma + \nu\in\Gamma$$
```

Gli esempi seguenti prendono in esame equazioni, matrici, formule allineate e illustrano i metodi di numerazione gestiti dalle seguenze `\eqno`, `\leqno`, `\eqalignno`, `\leqaligno` e `\displaylines`.

$$a_i = b_j \qquad \text{se e solo se} \quad b_n > 0 \qquad \text{per ogni n} \ge 0$$

$$x^{2y} = y_{x^2} + \left(\left(x^2\right)^3\right)^4$$

$$x^2 + y^2 = 2x + y - 7. \tag{3}$$

(3) $$x^2 + y^2 = 2x + y - 7.$$

$$4x^2 + 3y^2 = 16,$$
$$5x + y = 7.$$

$$2(x^4 - 6x^3 + 9x^2) = -3x^2 + 9x - 2$$
$$(x^2 - 3x)^2 = -3(x^2 - 3x) - 2; \tag{3}$$
$$x^2 - 3x + 1 = 0. \tag{4}$$

$$2(x^4 - 6x^3 + 9x^2) = -3x^2 + 9x - 2$$
(3) $$(x^2 - 3x)^2 = -3(x^2 - 3x) - 2;$$
(4) $$x^2 - 3x + 1 = 0.$$

$$x = y; \tag{1}$$
$$\text{se} \quad y = a \quad \text{allora} \quad x = a;$$
$$\text{e se} \quad z = x \quad \text{allora} \quad x = y = z = a. \tag{2}$$

$$A = \begin{pmatrix} x - \lambda & 1 & 0 \\ 0 & x - \lambda & 1 \\ 0 & 0 & x - \lambda \end{pmatrix}.$$

$$x^2 - y^2 = (x + y)(x - y). \tag{1}$$

$$\left\{ \begin{aligned} \alpha &= f(z) \\ \beta &= f(z^2) \\ \gamma &= f(z^3) \end{aligned} \right\} \qquad \left\{ \begin{aligned} x &= \alpha^2 - \beta \\ y &= 2\gamma \end{aligned} \right\}.$$

$$|y| = \begin{cases} f(x), & \text{if } 1 < x < \pi; \\ -f(x), & \text{if } \pi < x < 2\pi \\ 0 & \text{altrove.} \end{cases}$$

Le equazioni che precedono sono state prodotte con il codice:

```
$$a_i=b_j \qquad\hbox{se e solo se}\quad b_n>0\qquad\hbox{per ogni n $\ge0$}$$
$$x^{2y} = y_{x^2} + {({(x^2)}^3)}^4$$
$$x^2+y^2=2x+y-7.\eqno(3)$$
$$x^2+y^2=2x+y-7.\leqno(3)$$
$$\eqalign{4x^2+3y^2&=16,\cr
5x+y&=7.\cr}$$
$$\eqalignno{2(x^4-6x^3+9x^2)&=-3x^2+9x-2\cr
{(x^2-3x)}^2&=-3(x^2-3x)-2;&(3)\cr
x^2-3x+1&=0.&(4)\cr}$$
$$\leqalignno{2(x^4-6x^3+9x^2)&=-3x^2+9x-2\cr
{(x^2-3x)}^2&=-3(x^2-3x)-2;&(3)\cr
x^2-3x+1&=0.&(4)\cr}$$
$$\displaylines{\hfill x=y;\hfill\llap(1)\cr
\hbox{se}\quad y=a\quad\hbox{ allora}\quad x=a;\cr
\hfill\hbox{e se}\quad z=x\quad\hbox{ allora}\quad x=y=z=a.
\hfill\llap(2)\cr}$$
```

```
$$A=\left(\matrix{x-\lambda&1&0\cr
0&x-\lambda&1\cr
0&0&x-\lambda\cr}\right).$$
$$x^2-y^2 = (x+y)(x-y).\eqno(1)$$
$$\left\{
\eqalign{\alpha&=f(z)\cr \beta&=f(z^2)\cr \gamma&=f(z^3)\cr}
\right\}\qquad\left\{
\eqalign{x&=\alpha^2-\beta\cr y&=2\gamma\cr}\right\}.$$
$$|y|=\cases{f(x),&if $1<x<\pi$;\cr -f(x), &if $\pi<x<2\pi$\cr
0 &altrove.\cr}$$
```

Le formule che seguono evidenziano le capacità di variare in modo automatico allineamento e dimensioni nelle formule.

$$a_0+\cfrac{1}{a_1+\cfrac{1}{a_2+\cfrac{1}{a_3+\cfrac{1}{a_4}}}}$$

$$\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+x}}}}}}}$$

$$m+{m+{m}}.$$

Segue la codifica TEX relativa.

```
$$a_0+{1\over\displaystyle a_1 +{\strut 1\over\displaystyle a_2 +
{\strut 1\over\displaystyle a_3 + {\strut 1\over a_4}}}}$$
$$\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+x}}}}}}}$$
$$ m+\scriptstyle m+ \scriptscriptstyle m.$$
```

Da notare la differenza di dimensione del simbolo di radice se usato in combinazione con il comando \mathstrut come nell'esempio che segue. I termini a sinistra sono stati scritti con {\sqrt a ...}, quelli a destra con {\sqrt{\mathstrut a}+ ... }.

$$\sqrt{a}+\sqrt{d}+\sqrt{y}=\sqrt{a}+\sqrt{d}+\sqrt{y}$$

Le dimensioni delle parentesi in combinazione con i delimitatori "big" sono le seguenti:

$$()\big()\Big()\Big()$$

.

L'uso delle sequenze "dots" è indicato di seguito, per \cdots: $y_1 \cdots + y_n$, per \ldots $y_1 y_2 \ldots y_n$, nel testo e nelle matrici. Per il testo l'esempio è: "Indichiamo la velocità con il vettore $\vec{v}$ di modulo $\bar{v}$, allora ... ", per le matrici segue l'esempio con la relativa codifica TEX.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

```
$$A=\pmatrix{a_{11}&a_{12}&\ldots&a_{1n}\cr
a_{21}&a_{22}&\ldots&a_{2n}\cr
\vdots&\vdots&\ddots&\vdots\cr
a_{m1}&a_{m2}&\ldots&a_{mn}\cr}$$
```

Gli esempi seguenti illustrano le lettere greche, simboli vari (as es. lettere corsive calligrafiche come l e ℓ), disuguaglianze, radici, derivate, etc.

$$\alpha, \beta, \gamma, \Gamma;$$

$$\nu + \kappa - \phi * \emptyset - \epsilon \in \Gamma$$

$$\phi, \theta, \epsilon, \rho$$

$$\varphi, \vartheta, \varepsilon, \varrho$$

$$\approx \; \mapsto$$

$$x \times y \cdot z \quad \text{oppure} \quad x \circ y \bullet z \quad \text{oppure} \quad x \cup y \cap z$$

$$x \leq y \neq z \sim a + b \simeq \pi \not\equiv 0.0$$

$$y_1' = f'[g(x)]g'(x) + y_3''' - g'^2$$

$$\sqrt[n+1]{a}$$

La codifica TEX di questo esempio é:

```
$$\alpha, \beta, \gamma, \Gamma;$$
$$\nu + \kappa - \phi * \emptyset - \epsilon \in \Gamma$$
$$\phi,\theta,\epsilon,\rho$$
$$\varphi,\vartheta,\varepsilon,\varrho$$
$$\approx \, \> \; \mapsto$$
$$x\times y\cdot z \quad\hbox{ oppure }\quad x\circ y\bullet z
\quad\hbox{ oppure }\quad x\cup y\cap z$$
$$x\le y\ne z \sim a+b \simeq \pi \not\equiv 0{.}0$$
$$y_1^\prime = f'[g(x)]g'(x) + y'''_3 - g'{}^2$$
$$\root n+1 \of a$$
```

10. Esempi di Grafica

TEX è in grado di tracciare rette verticali e orizzontali con i comandi `\vrule` e `\hrule`. Poichè entrambi i comandi permettono di assegnare delle dimensioni al segmento tracciato, con i medesimi si possono creare dei tasselli e dei rettangoli neri. Con opportune combinazioni di `\hrule` e di `\vrule` si possono generare quadrati e rettangoli in cui inserire eventualmente del testo.

Le definizioni `\square` e `\boxit` illustrate nel paragrafo "Capacità Grafiche di TEX", producono i risultati illustrati di seguito. Il comando `\vrule height4pt width3pt depth2pt` produce un tassello nero circa delle stesse dimensioni del quadratino disegnato da `\square`. Il comando `\vrule width3cm height20pt` produce un rettangolo nero con il lato lungo di 3cm e il lato corto di 20pt sulla riga in corso. Se a `\vrule` si sostituisce `\hrule`, il rettangolo viene tracciato su di una nuova riga. Nell'ultimo esempio un intero paragrafo è racchiuso in una cornice rettangolare calcolata dinamicamente da `\boxit`.

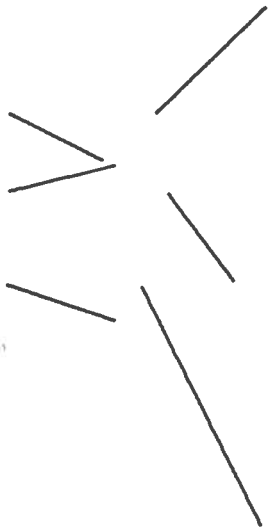
Questo paragrafo è contenuto in una cornice rettangolare le cui dimensioni sono gestite in modo automatico in funzione della lunghezza del testo che viene specificato come argomento della macro "boxit".

Sfruttando le capacità matematiche di TEX per calcolare piccoli spostamenti e scrivendo un punto alla volta si possono tracciare segmenti di inclinazione variabile. La definizione seguente ***newline*** [3] designa come parametri di spostamento i registri `\orignalvmove`, `\hmove`, `\vmove` e sfrutta come unità di misura il valore ***sp = scaled points*** che è definito da TEX come ***65536sp=1pt***, con ***72.27pt=1in***. La macro newline ha 3 argomenti: l'ampiezza dei movimenti orizzontale (#1) e verticale (#2) in ogni step e lo spostamento totale (#3). Muovendo un punto (`{\hss.\hss}`) un certo numero di volte (`\loop\ifnum \numtimes < \hwidth`) in orizzontale (`\hskip \the\hmove sp`) e abbassandolo (`\lower \the\vmove sp`) in unità sp, il movimento è tanto piccolo che il singolo punto non è più distinguibile e si produce una riga continua.

```
\newcount\originalvmove
\newcount\hwidth\newcount\numtimes
\newcount\hmove\newcount\vmove
%%%
\def\newline#1#2#3{\numtimes=0 \vmove=1000\hmove=1000
\multiply\hmove #1 \multiply\vmove #2 \originalvmove=\the\vmove
\hwidth=#3
\multiply\hwidth by65536
\divide\hwidth by\the\hmove
\loop\ifnum\numtimes<\hwidth
\hskip\the\hmove sp\lower\the\vmove sp\hbox
to 0pt{\hss.\hss}\advance\numtimes by 1\advance\vmove by\originalvmove
\repeat}
```

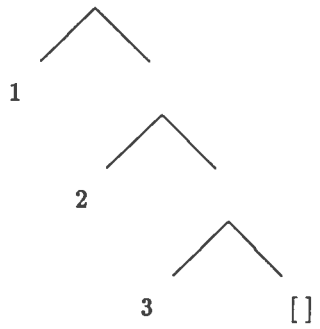
Variando il valore degli argomenti si possono produrre segmenti di diversa linghezza e orientamento. I segmenti di questo esempio sono stati generati con i comandi:

```
\newline{40}{20}{35}\qquad\newline{20}{-20}{40}\par
\newline{60}{-15}{40}\qquad\newline{30}{40}{25}\par
\newline{30}{10}{40}\quad\newline{30}{60}{45}
```



In modo analogo si possono creare figure complesse tenendo presente la limitazione di memoria di TEX, che non consente di creare più di 6 o 7 segmenti del tipo indicato per pagina. Per ovviare a questa limitazione si devono costruire fonti grafici opportuni. Con TEX sono distribuiti i fonti: ***line*** e ***circle*** per tracciare rette inclinate o cerchi come illustrato di seguito.

Diamo un esempio di struttura binaria ad albero. I fonti usati sono segmenti con angolo di $+45°$ (`\char00`) e di $-45°$ (`\char64`) e vengono posizionati in modo opportuno per produrre il grafico in figura.



La codifica TEX è la seguente:

```
\font\lfont=line10
{\bigskip \lfont \obeylines
%
\hskip 40pt \char00 \char64 \vskip -5pt
\hskip 30pt \char00 \hskip 20.25pt \char64
{\rm \hskip 18pt $1$}
%
\hskip 65pt \char00 \char64 \vskip -5pt
\hskip 55pt \char00 \hskip 20.0pt \char64
{\rm \hskip 43pt $2$}
%
\hskip 90pt \char00 \char64 \vskip -5pt
\hskip 80pt \char00 \hskip 20.0pt \char64
```

```
{\rm \hskip 68pt $3$ \hskip 47pt [ ]}\bigskip}
```

Si invita il lettore a riesaminare l'esempio stampato e a costruire una definizione in cui gli spostamenti siano relativi. Il nostro esempio infatti darebbe cattivi risultati se si variasse l'ingrandimento e/o la spaziatura delle righe. Esempi di questo tipo si prestano bene ad un'analisi delle capacità di TEX perchè possono essere elaborati in vari modi e trasformati in macro.

Usando il fonte "circle" si possono disegnare cerchi di varie misure. Nell'esempio sono stati usati i caratteri 39,36,38,37 (archi di cerchio) per disegnare il cerchio grande. I cerchi piccoli (pieno e vuoto) sono generati dai caratteri 110 e 126. In "circle" sono disponibili anche altre dimensioni, ad esempio i caratteri 11, 08, 10, 09 rappresentano archi di cerchio con circonferenza uguale a quella del cerchio `\char110`. Il grafico che si ottiene è:

ed è stato generato dalle istruzioni:

```
\font\cfont=circle10 {\cfont \obeylines
\char39\char36
\char38\char37 \hskip 30pt \char110 \hskip 30pt \char126}
```

11. Campionario Fonti

Abbiamo già visto che TEX possiede una vasta serie di fonti che l'utente può selezionare con il comando `\font`. I fonti esistenti devono essere memorizzati in opportune directories a seconda dell'installazione, che dovrebbe fornire anche le tabelle di descrizione dei fonti medesimi con i rispettivi corpi e ingrandimenti.

Tutte le installazioni dovrebbero avere i fonti normali (`\rm`), il neretto (`\bf`), il corsivo (`\sl` e `\it`), i caratteri speciali per la matematica e l'emulazione della macchina da scrivere (`\tt`).

Di seguito diamo un esempio di codifica TEX per la produzione di un campionario di fonti. Nella prima parte vengono definiti tutti i nomi dei fonti di cui si vuole fare uso. Nella seconda parte vengono richiamati e stampati rispettando le righe per consentire un confronto dei tipi di carattere.

```
\font\boldx=cmbx10 \font\boldc=cmcsc10   %definizione fonti
\font\boldu=cmu10      \font\boldunh=cmdunh10  \font\texdix=cmtex10
\font\tidix=cmti10     \font\ittpwr=cmitt10    \font\boldit=cmbi10
\font\boldsl=cmbxsl10 \font\sltpwr=cmsltt10   \font\sansdix=cmss10
\font\boldsdix=cmssbx10  \font\itsansdix=cmssi10
\font\boldsmcdix=cmssmc10 \font\sans=cmssq8   \font\itsans=cmssqi8
% fine definizioni
{\obeylines \parindent 0pt  %elimina indentazione & inizia campionario
\centerline{Campionario Fonti}
\vskip 15pt
{\bf ABC abc 123} --- cmb10 {\boldx ABC abc 123} --- cmbx10
```

```
{\boldc ABC abc 123} --- cmcsc10 {\boldu ABC abc 123} --- cmu10
{\boldunh ABC abc 123} --- cmdunh10 {\texdix ABC abc 123} --- cmtex10
{\rm ABC abc 123} --- cmr10  {\it ABC abc 123} --- cmi10
{\tidix ABC abc 123} --- cmti10 {\sl ABC abc 123} --- cmsl10
{\tt ABC abc 123} --- cmtt10  {\ittpwr ABC abc 123} --- cmitt10
{\boldit ABC abc 123} --- cmbi10  {\boldsl ABC abc 123} --- cmbxsl10
{\sltpwr ABC abc 123} --- cmsltt10  {\sansdix ABC abc 123} --- cmss10
{\boldsdix ABC abc 123} --- cmssbx10  {\itsansdix ABC abc 123} --- cmssi10
{\boldsmcdix ABC abc 123} --- cmssmc10  {\sans ABC abc 123} --- cmssq8
{\itsans ABC abc 123} --- cmssqi8
% fine campionario}
```

Dato che la produzione di un campionario del genere è un compito molto pesante sia per TEX che per lo spooler, a titolo esemplificativo, diamo di seguito la stampa dei soli fonti predefiniti.

ABCDEF abcdef 123456 — (Bold)
ABCDEF abcdef 123456 — (Roman)
ABCDEF abcdef 123456 — (Slanted)
ABCDEF abcdef 123456 — (Italic)
`ABCDEF abcdef 123456 --- (Typewriter)`

12. Un Esempio Completo

Con questo paragrafo inizia la descrizione di un esempio applicativo tratto dal mondo reale in cui la metodologia di utilizzo di TEX è illustrata in modo completo anche se applicata ad un testo semplificato. L'esempio è tratto da una relazione attinente ad un argomento di fisica delle alte energie da cui abbiamo estratto tutte le codifiche complesse incontrate. Nell'esempio abbiamo seguito la metodologia che usiamo abitualmente quando affrontiamo il compito, spesso gravoso, di produrre documenti tecnici lunghi e complessi in una veste tipografica che abbini un livello elevato di qualità e di leggibilità.

Oltre al file di testo, sono stati creati due files ausiliari, uno con le macro definizioni e uno per la tabella. La relazione è redatta in lingua inglese. Quando si usa TEX per un testo di questo tipo, il primo compito è la stesura di un file di macro definizioni per avere uno stile unificato nella gestione dei paragrafi e dell'impaginazione in generale e inoltre per la creazione di definizioni che possano semplificare la stesura del testo. Di seguito compaiono i testi sorgente in TEX con la spiegazione relativa. Alla fine del paragrafo, nei limiti imposti dalle dimensioni dell'articolo, compare la stampa finale del testo. L'impaginazione dipende dai parametri `\vsize` e `\hsize` posti all'inizio del testo. Normalmente per centrare meglio il testo sul formato A4 Europeo che è un poco più stretto e leggermente più lungo, definiamo `{\hsize = 6 truein}` e `{\vsize = 9.3 treuin}`, contemporaneamente modifichiamo o l'ingrandimento o lo spazio interlinea per migliorare la leggibilità del testo.

13. Creazione della libreria MACRO.TEX

Da un esame preliminare del manoscritto si annotano le frasi, i termini e le espressioni di uso più frequente, i fonti che potranno servire, e tutte le parti del testo che possono essere gestite vantaggiosamente da macro definizioni. Si crea quindi un primo file che verrà inserito con l'istruzione

\input, con il vantaggio di poter provare le definizioni con un testo molto semplice e di poter visualizzare rapidamente i risultati senza dover ricorrere alla stampa. Nel nostro caso il file che contiene le macro si chiama `MACRO.TEX` ed è stampato di seguito.

```
% file: MACRO.TEX - font definitions
%
\font\mysmall=cmr8                              % footnote, abstract, etc.
\font\titlefont=cmr10  scaled\magstep2          % title
\font\myfont=cmr10
\font\mybold=cmbx10
\font\myit=cmi10
\font\mylp=cmtt10
%-----------------------------------------------------
% macro definitions
%
\def\HEP{High Energy Physics }
\def\Hep{High energy physics }
\def\hep{high energy physics }
\def\FOR{Fortran }
%
\def\cftcode#1{\medskip {\tt\halign
{\indent{##}\hfil&&\qquad##\hfil\crcr#1}}
\medskip}
%
\def\begfortran{ \begingroup \tt\begingroup $$\vbox }
\def\codfortran{ \qquad\qquad&mmmmm\qquad&\cr }
\def\endfortran{ $$ \endgroup \endgroup }
\def\begtable{ \begingroup \smallskip \vbox }
\def\endtable{ \smallskip \endgroup}
%
\def\newpartitle#1{\vskip 12pt plus2pt minus2pt
{\mybold #1}
\vskip 12pt plus2pt minus2pt}
%
% end macros
%-----------------------------------------------
```

Esaminiamo le definizioni gruppo per gruppo.

1. Per prime compaiono le ***definizione dei fonti***. Vengono definiti i fonti per il titolo e per le note a fondo pagina. Con `\font\mysmall = amr8` si definisce il font `\mysmall` nel corpo 8 che viene usato sia per le note di "abstract" che per quelle a fondo pagina. Con `\font\titlefont = amr10 scaled\magstep2` si definisce un corpo ingrandito per il titolo. Si definisco altri fonti di uso generale.
2. Si definiscono le macro per la gestione tipografica omogenea delle ***frasi ricorrenti***. Nel testo compaiono molte volte le scritte "high energy physics" e "Fortran" quindi vengono definite le sequenze per "high energy physics" \hep, \HEP, \Hep in cui \hep è formato da tutte minuscole

per l'uso nel corso del test, \HEP ha tutte le iniziali maiuscole per il titolo, la bibliografia, etc. e \Hep ha solo la prima iniziale maiuscola per l'uso all'inizio di una frase, per "Fortran" \FOR.

3. Seguono le macro che gestiscono ***tabulazione e impaginazione***. Dato che il testo riporta numerosi esempi di codice Fortran abbiamo definito due possibili forme di tabulazione per riprodurre il formato del linguaggio: \cftcode e il gruppo \begfortran \codfortran \endfortran. Queste due macro sono parametriche. Entrambe definiscono una tabulazione su due colonne con fonte di tipo "typewriter" e spaziatura verticale addizionale prima e dopo l'inserimento del codice Fortran. La prima fa uso del comando \halign, la seconda di \settabs. Con la prima è più semplice la modifica del Fortran in formato TEX, la seconda è più conveniente per liste molto lunghe, dato che (come si è già detto) TEX allinea le righe una alla volta su una tabulazione fissa, mentre nel caso di \halign la dimensione delle colonne è determinata dinamicamente in base al testo richiedendo perciò la memorizzazione dell'intera tabella prima dell'elaborazione. Per tabulazioni generiche è stato definito il gruppo \begtable \endtable che crea la scatola verticale in cui verrà inserita la tabella e gestisce la spaziatura iniziale e finale. Il prototipo della tabella verrà descritto di volta in volta. Per gestire l'impaginazione è stata scritta la macro \newpartitle che gestisce l'intestazione dei paragrafi e precisamente spaziatura e fonte del testo che viene passato come parametro. La definizione di newpartitle potrebbe venire migliorata con l'aggiunta della numerazione automatica del paragrafi. Daremo una descrizione particolareggiata dell'uso di questa macro nell'ambito del testo in esame.

14. Creazione della Tabella TABLE.TEX

Nel testo dobbiamo inserire una tabella molto complessa. Invece di inserire la codifica completa TEX nel punto voluto, facciamo ricorso al comando \input seguito dal nome del file che contiene la tabella, nel nostro caso TABLE. Il tipo non è necessario (il default è .TEX) e in questo caso è indifferente l'uso di maiuscole o minuscole dato che il nome del file viene elaborato dal Rercord Manager del VMS. In questo modo possiamo verificare la correttezza della tabella separatamente.

Nel nostro caso la tabella è costituita da 5 colonne, con un'instestazione per ogni colonna e un'intestazione generale, il tutto racchiuso in un diagramma. Il codice TEX per la tabella è il seguente.

```
% file table.tex
%
\relax
\medskip
\vbox{\tabskip=0pt \offinterlineskip
\def\tablerule{\noalign{\hrule}}
\halign to260pt{\strut#& \vrule#\tabskip=1em plus2em&
\hfil#& \vrule#& \hfil#\hfil& \vrule#&
\hfil#\hfil& \vrule#&
\hfil#\hfil& \vrule#&
\hfil#\hfil& \vrule#\tabskip=0pt\cr\tablerule
&&\multispan9\hfil Cray Total Gain (millisec)\hfil&\cr\tablerule
&&\omit\hidewidth Level\hidewidth&&
\omit\hidewidth Time\hidewidth&&
\omit\hidewidth Speed--up\hidewidth&&
```

```
\omit\hidewidth Gain\hidewidth&&
\omit\hidewidth CDC/Cray\hidewidth&\cr\tablerule
&&0&&33.37&& -- && -- &&1.36&\cr\tablerule
&&1&&29.86&&10.5&&10.5&&1.52&\cr\tablerule
&&2&&24.19&&27.5&&17.0&&1.87&\cr\tablerule
&&3&&21.92&&34.3&& 6.8&&2.07&\cr\tablerule
\noalign{\medskip}\hfil\cr}}
\relax
```

I comandi `\relax` all'inizio e alla fine hanno la funzione di "no operation" e servono a TEX per sincronizzare l'output, è quindi buona norma farne uso prima e dopo tabulazioni complesse. Per maggiori informazioni fare riferimento al TEXbook.

La macro inizia con una spaziatura verticale supplementare all'inizio e alla fine della tabella (comandi `\medskip`). Ovviamente l'intera tabella deve costituire una struttura, quindi inizieremo con `\vbox`. La prima colonna è allineata con la marginatura naturale del foglio, quindi poniamo `\tabskip=0pt`. Dato che le righe verticali di separazione delle colonne devono essere continue, eliminiamo la spaziatura interlinea con `\offinterlineskip`. Per disegnare le righe orizzontali di separazione tra le varie voci della tabella definiamo la macro `\tablerule` (notare che la definizione è interna alla struttura) come `\hrule`, il comando `\noalign` è necessario per informare TEX che la riga deve estendersi su tutte le caselle inibendo i criteri di incasellamento attivi. Il comando `\halign to 260pt` definisce una tabulazione dinamica con una dimensione orizzontale massima di 260pt.

Ora possiamo definire la struttura (`\strut`) come un insieme dinamico di righe verticali (`\vrule` per un totale di 6) e di elementi variabili centrati all'interno delle singole caselle (`\hfil#\hfil` per un totale di 5) con separazione di `1em plus 2em` tra una colonna e l'altra tranne l'ultima (comandi `\tabskip`), per un totale di 9 colonne (5 con testo e 4 con le righe verticali di separazione). I comandi di tabulazione (avanzamento da una casella formale all'altra) sono indicati da `&`, la fine della riga è determinata da `\cr`.

Con `\tablerule` si traccia la prima riga orizzontale. Per inserire l'intestazione generale si deve inibire totalmente la tabulazione su 9 colonne formali con il comando `\multispan9`. Il testo è compreso tra 2 `\hfil` per la centratura e `\tablerule` traccia la riga orizzontale sotto l'intestazione. Seguono le intestazioni delle singole colonne in cui il comando `\omit` inibisce il calcolo di centratura tabulare della casella, cioè omette l'elaborazione del preambolo, mentre i 2 `\hidewidth` prima e dopo il testo determinano la centratura delle intestazioni indipendentemente dalla definizione tabulare descritta nel preambolo per evitare di influenzare l'incolonnamento dei dati numerci.

Dopo le intestazioni vengono battuti i dati da incolonnare. Notare che ogni riga ***deve*** terminare con `\cr`. I 2 comandi consecutivi di tabulazione (`&&`) indicano che una delle colonne è la linea verticale (`\vrule`) predeterminata nella descrizione del prototipo.

Infine compare la spaziatura verticale `\medskip`. Dato che è stata inserita all'interno della tabella va completata da `\noalign` e `\hfil`. Si rimanda al paragrafo seguente per la visualizzazione della tabella dopo l'elaborazione da parte di TEX.

15. Il Testo Completo

Di seguito riportiamo il testo completo in formato TEX, seguito dalle note sui punti salienti e dalla stampa finale in cui si devono tenere presenti le limitazioni dovute all'inserimento di detta stampa in un altro testo.

```
%-----------------------------------------------------------------------
%
\font\mysmall=cmr8                              % footnote, abstract, etc.
\input macro
\relax
\centerline{\titlefont \HEP}
\vskip 12pt
\noindent {A. AUTHOR}\par
\noindent {\it At Following Address:  \Hep Road, No.  $\pi$ }\par
\vskip 12pt plus2pt minus2pt
\hrule
\vskip 4pt plus 2pt
{\mysmall Abstract:  foolish report to show how to become \TeX experts and
Wizards for \hep type of pubblications}
\vskip 4pt plus 2pt
\hrule
\medskip
This is a sample of report structure using \TeX{} with tables, \FOR examples
and footnotes.
This is the first \footnote* {\mysmall show first footnote} line
of a two \footnote{**} {\mysmall show second footnote}
footnotes demo text.  To display \FOR code write down line as follows:
\cftcode{
&SUBROUTINE XYZ\cr
C&FORTRAN ROUTINE EXAMPLE\cr
&I=10\cr
10&CONTINUE\cr
&RETURN\cr
}%
And now some very complicated table with ruled delimiters.
\input table
\newpartitle{1.  Table settings}
\noindent
To write long tables, use the settabs command.\par
\noindent{\bf WCW}\smallskip
\item{1.} User Record Length $l1$ (18 bits $-$ bit position 0--17).
\item{2.} Unused Bit Count of Last Record (6 bits $-$ bit position 18--23).
\item{3.} Continuation flag (2 bits $-$ bit position 42--43), as follows:
\begtable
{\settabs\+\indent\quad&mm&mmmmmmmmmm\qquad&\cr     %table template
\+&0&Full Record &binary $=00$,\cr
\+&1&Beginning Portion &binary $=01$,\cr
\+&2&MiddlePortion  &binary $=10$,\cr
\+&3&End Portion  &binary $=11$.\cr
}
```

```
\endtable
The records can be spanned, that is only full records are present, therefore
from the above description, we can see that the record pattern is the
following.
Item list is ended, and print resumes correctly as we can verify on
\TeX{} output.This is another template for \FOR source code.
\par
\begfortran
{\settabs\+\codfortran
\+&FUNCTION DOT(A,B)\cr
\+999&RETURN\cr
\+&END\cr}
\endfortran
\newpartitle{Reference.}
\item{[1]}D.J.Paddon, Supercomputers and parallel computation, Claredon Press,
Oxford.
\item{[2]}J.S.Someone, Private Communication.
\par
\bye
%-------------------------------------------------------------------------------
```

16. Note Esplicative

1. Il titolo è centrato e scritto in un fonte ingrandito.
2. L'indirizzo dell'autore è in caratteri corsivi.
3. Le note di abstract, così come le note in fondo pagina sono in un corpo più piccolo.
4. Il primo frammento di codice Fortran è stato scritto con la macro `\cftcode` (corpo tipo macchina da scrivere, incolonnamento Fortran). Il testo deve comparire tra graffe, ogni riga deve terminare con `\cr`, il secondo campo deve essere preceduto da `&`.
5. È inserita la tabella con il comando `\input table`.
6. I titoli dei paragrafi sono scritti con il comando `\newpartitle` (neretto e spaziatura verticale aumentata). I numeri dei paragrafi sono gestiti manualmente. Se si volessero gestire in modo automatico si dovrebbe usare una definizione `\newpartitle` del tipo seguente:

```
\newcount\newparno
\def\initnewpar{\newparno=0}
\initnewpar
\def\newpartitle#1{\vskip 12pt plus2pt minus2pt
\advance\newparno by 1
{\noindent \bf \the\newparno .\ #1}
\vskip 9pt plus1pt minus1pt}
```

in cui `\newparno` è il numero del paragrafo e viene inizializzato da `\initnewpar` che viene chiamata all'inizio del testo. Ogni volta che si invoca `\newpartitle` il registro `\newparno` viene incrementato di uno e stampato assieme al titolo del paragrafo con il comando `\the\newparno`.

7. La parola WCW è stata scritta in neretto, segue una lista enumerata di argomenti che viene gestita con il comando `\item`. Alla voce 3. della lista compaiono delle sottoliste di argomenti incolonnati. Queste liste sono gestite con le macro `{\begtable \endtable}`, la tabulazione è gestita dal comando `\settabs` che genera un "template" per 3 colonne di diversa ampiezza. L'inizio di ogni riga è determinato da `\+` (tabulazione iniziale), ogni campo è preceduto da `&`, le righe terminano con `\cr`.
8. L'ultimo frammento di codice Fortran è gestito dalle macro `\begfortran` e `\endfortran` che delimitano la struttura, l'incasellamento è generato dal comando `\settabs` e il "template" è definito dalla macro `\codfortran`.
9. L'impaginazione della bibliografia è gestita da `\item`.

High Energy Physics

A. AUTHOR
At Following Address: High energy physics Road, No. π

Abstract: foolish report to show how to become TEXexperts and Wizards for high energy physics type of pubblications

This is a sample of report structure using TEX with tables, Fortran examples and footnotes. This is the first * line of a two ** footnotes demo text. To display Fortran code write down line as follows:

```
      SUBROUTINE XYZ
C     FORTRAN ROUTINE EXAMPLE
      I=10
10    CONTINUE
      RETURN
```

And now some very complicated table with ruled delimiters.

Cray Total Gain (millisec)				
Level	Time	Speed–up	Gain	CDC/Cray
0	33.37	–	–	1.36
1	29.86	10.5	10.5	1.52
2	24.19	27.5	17.0	1.87
3	21.92	34.3	6.8	2.07

This is a small paragraph another one follows with title.

1. Table settings

To write long tables, use the settabs command.

* show first footnote
** show second footnote

WCW

1. User Record Length $l1$ (18 bits – bit position 0–17).
2. Unused Bit Count of Last Record (6 bits – bit position 18–23).
3. Continuation flag (2 bits – bit position 42–43), as follows:

0	Full Record	binary = 00,
1	Beginning Portion	binary = 01,
2	MiddlePortion	binary = 10,
3	End Portion	binary = 11.

The records can be spanned, that is only full records are present, therefore from the above description, we can see that the record pattern is the following. Item list is ended, and print resumes correctly as we can verify on TEX output. This is another template for Fortran source code.

```
      FUNCTION DOT(A,B)
999   RETURN
      END
```

Reference.

[1] D.J.Paddon, Supercomputers and parallel computation, Claredon Press, Oxford.

[2] J.S.Someone, Private Communication.

17. Debug

Un compito anche molto complesso è rappresentato dalla correzione degli errori che TEX incontra durante l'elaborazione del testo. Nelle note che seguono sono illustrate le istruzioni di ***debug*** definite in TEX e vegono forniti suggerimenti e indicazioni utili alla rapida identificazione e correzione degli errori.

Gli errori che si possono commettere possono essere rivelati da TEX oppure possono influenzare solo l'aspetto del testo prodotto (ad es. cattiva impaginazione, fonte errato, etc.) e quindi venire evidenziati solo in fase di stampa.

Di solito TEX viene usato in modo interattivo, perciò se il programma incontra degli errori durante l'elaborazione del testo, scrive un messaggio con l'indicazione della riga in cui ha trovato l'errore e si ferma con il ***prompt ?***. Il messaggio è del tipo:

```
! Undefined control sequence
2.20 \sequence
?
```

dove `\sequence` è la sequenza che ha dato errore e `2.20` indica file e riga in cui l'errore è stato diagnosticato. Nel nostro esempio 2 è il secondo file letto da TEX con il comando `\input` e 20 indica la riga in cui l'elaborazione si è arrestata. Le risposte che si possono dare (oltre ovviamente a terminare ***brutalmente*** la seduta con `^Y`) sono indicate di seguito.

```
R per proseguire
I per inserire qualche cosa
H per help
```

`X per terminare`

Va notato che è molto difficile poter correggere l'errore in modo interattivo e le risposte listate possono servire solo come aiuto nella diagnosi. Se TEX viene invocato da una procedura "batch", per evitare problemi in caso di errore, è bene eliminare la richiesta di intervento con il comando `\batchmode`.

Nel caso in cui TEX dichiari degli errori in fase di esecuzione si possono ottenere informazioni supplementari inserendo opportuni comandi di debug. Ad esempio si può chiedere il significato di una sequenza con il comando `\show\nome_sequenza`, esistono altri comandi del genere, ad esempio `\showbox0` per vedere il contenuto del `box0`. Altri comandi di debug sono `\showlists` che elenca le liste di elaborazione, `\tracingcommands` e `\tracingparagraph` che producono un flow trace dei comandi e della suddivisione in righe. L'informazione dei comandi di "trace" viene scritta sul file di "log", perchè venga scritta anche su video occorre specificare `\tracingonline` con valore positivo, ad esempio `\tracingonline=1`. Analogamente `\tracingpages` produce un flow trace dell'impaginazione, mentre `\tracingmacros` traccia lo sviluppo delle macro. La stampa di debug così prodotta è molto complessa e per una corretta interpretazione richiede una conoscenza profonda del modo operativo di TEX per il quale si rimanda al TEXbook.

Segue un ricettario rapido di errori e rimedi per i problemi che si presentano con maggiore frequenza nell'uso di TEX.

Errori molto frequenti sono determinati da uso improprio della spaziatura e dei simboli di controllo. Generalmente gli spazi bianchi possono essere omessi in tutti i casi in cui non ci sia ambiguità, deve però essere presente il blank alla fine di una sequenza di controllo. I nomi dei comandi definiti dall'utente non possono contenere numeri o caratteri speciali. È errore comune mescolare male modo matematico e testo, specie la punteggiatura che non deve essere scritta in modo matematico.

Le parentesi di tutti i tipi devono essere bilanciate, cioè `{-}`, `$-$`, `$$-$$`, `\if-\fi`, altrimenti si possono ottenere messaggi d'errore difficili da decifrare. Inoltre una parentesi aperta che non abbia la corrispondente chiusa in genere viene rivelata molte righe dopo l'errore, talvolta anche solo alla fine del file. Se le parentesi sono bilanciate erroneamente, cioè mancano sia un'aperta che una chiusa, molto probabilmete TEX non sarà in grado di rivelare errori ma la stampa prodotta sarà inconsistente o illeggibile, specie nel caso in cui un testo normale venga elaborato in modo matematico o con un fonte grafico.

I nomi delle sequenze, macro etc. devono essere corretti e avere il delimitatore richiesto. Ad esempio la parola TEXbook deve essere scritta `\TeX book` e non `\TeXbook` perchè in questo secondo caso la sequenza risulterà mancante. Analogamente, se una macro non prevede la spaziatura finale che la separi dal testo che segue, sarà cura dell'utilizzatore ricordarsi di gestirla correttamente. Ad esempio se in stampa deve comparire la frase "TEX è ...", si deve scrivere `"\TeX{} \'e...`", se non si usa il gruppo vuoto `{}` e si scrive `"\TeX \'e ...`", in stampa si avrà "TEXè ..." indipendentemente dagli spazi bianchi lasciati dopo `\TeX` la cui sola funzione è di delimitatori di sequenza.

Invocando le macro ricordarsi di fornire gli argomenti, in particolare non dimenticare i valori numerici con le relative unità di misura per i parametri dei comandi di spostamento come `\vskip`, `\hskip`, etc. In modo matematico usare le `{}` per denotare esponenti complessi, ad esempio $a^{2\pi}$ deve essere scritto `$a^{2\pi}$`, altrimenti omettendo le parentesi graffe, si ottiene $a^2\pi$. Analogo discorso vale per elementi grafici quali il segno di radice, sommatoria, integrale, etc. Tenere sempre

presente la differenza tra i modi text e display nelle formule. Se una macro ha un comportamento anomalo e non si riesce a capire il tipo di errore commesso inserire \par per bloccarne l'espansione.

Il comando \showthe\definizione dà il valore della definizione, per esempio la sequenza seguente \showthe\parindent fornisce il valore corrente di \parindent. Se si invoca un nuovo fonte all'interno di una macro e si vuole che abbia effetto temporaneo, occorre metterlo in un gruppo altrimenti rimane attivo fino alla fine del file.

Se si definisce una macro e si dimentica di chiamarla con \, verrà stampato il nome della macro stessa. Ad esempio se viene definita la macro mathseq per stampare una sequenza matematica ma invocandola si usa il comando $mathseq$ invece di $\mathseq$ in stampa si avrà *mathseq*.

Come abbiamo visto TEX indenta ogni nuovo paragrafo, cioè ogni gruppo di righe che termini con \par o con una riga vuota. Per eliminare localmente l'indentazione esiste il comando \noindent che ha effeto solo se preceduto da \par o da qualsiasi altro comando che determini la fine del paragrafo o comunque il passaggio da modo orizzontale a verticale. Per eliminare globalmente l'indentazione si deve ridefinire il parametro \parindent ponendone il valore =0pt. Dato però che TEX usa \parindent per la gestione di \item, occorre ripristinare il suo valore a 20pt, prima di usare \item. In caso contrario, il testo di ciascun elemento di \item viene scritto sopra alla numerazione assegnata nella forma \item{1.}. Una lista enumerata creata con \item deve essere messa in una struttura o terminare con un comando \medskip, altrimenti tutto il testo che segue verrà elaborato in forma di lista enumerata.

Un caso particolare è rappresentato dalle tabelle. Oltre al problema di generare correttamente il prototipo di gestione occorre definire correttamente ogni colonna con il simbolo & e terminare sempre ogni riga \cr. Se si dimentica &, si avrà solo un cattivo incolonnamento ma se si dimentica \cr si potranno avere errori di TEX molto difficili di localizzare. Le stesse considerazioni valgono per matrici, equazioni, etc.

Un ultima avvertenza che non è direttamente connessa con TEX ma dipende dal device di stampa è la scelta delle dimensioni dei fonti e della pagina. Per mantenere una buona leggibiltà in funzione della risoluzione della stampante, può essere necessario modificare l'ingrandimento e/o i fonti usati oltre a ridefinire le dimensioni della pagina. Nel caso in cui la pagina abbia una dimensione molto ridotta rispetto al default (ad es. \hsize=4truein), è molto probabile che si abbiano frequenti segnalazioni di \overfull box, in questo caso è bene rivedere i valori assegnati a \penalty e \tolerance.

18. Parole Chiave

TEX possiede un set di parole chiave che non hanno bisogno di essere invocate con il \ dato che possono essere espresse solo in contesti ben definiti. Le parole chiave sono:

`at`	definizione di una dimensione
`bp`	unità tipografica – big point
`by`	denota step incrementale
`cc`	unità tipografica – cicero
`cm`	centimetro
`dd`	unità tipografica – didot
`depth`	profondità di scatola
`em`	unità tipografica – larghezza della lettera m
`ex`	unità tipografica – altezza della lettera x
`fil`	unità relativa di riempimento
`fill`	idem – doppio della precedente
`filll`	idem – doppio della precedente
`height`	altezza di scatola
`in`	inch
`minus`	tolleranza negativa
`mm`	millimetro
`mu`	unità di spaziatura delle formule matematiche – math unit di valore 18mu, circa 1em nel fonte dei simboli matematici
`pc`	unità tipografica – pica
`plus`	tolleranza positiva
`pt`	unità tipografica – punto
`scaled`	assegnazione di magnificazione
`sp`	unità tipografica – scaled point
`spread`	tolleranza di una scatola
`to`	assegnazione di dimensione di una scatola
`true`	dimensione vera indipendentemente dalla magnificazione
`width`	larghezza di una scatola

19. Caratteri di Controllo

TEX usa i seguenti caratteri come caratteri di controllo.

\	carattere di escape, ciò che segue è un comando
{	inizio struttura
}	fine struttura
$	inizio/fine modo matematico
&	tabulatore
#	parametro
^	apice
_	indice
%	commento
~	legatura cioè la parola preceduta da ~non può essere separata dalla parola che la precede (es. `Fig.~12`) ossia le due parole Fig. e 12 devono comparire sulla stessa riga, inoltre ~agisce anche da spazio tra i due termini

Se nel testo devono comparire i caratteri su indicati, questi verrano espressi dalle sequenze di controllo seguenti.

$	\$
#	\#
%	\%
&	\&
_	_
^	\^ seguito da {\ } per ^ isolato, altrimenti accento circonflesso
~	\~ seguito da {\ } per ~ isolato, altrimenti tilde
\	\
{	${$
}	$}$

20. Caratteri Speciali

TEX definisce una serie di caratteri con significato speciale per la gestione di accenti, virgolette, etc.

-	hyphen, trattino di sillabazione (autogestito)
--	en–dash = trattino inter parola (es. n–esimo)
---	em–dash = trattino inter frase
$-$	segno meno in matematica
\'	accento acuto sulla vocale seguente
\`	accento grave sulla vocale seguente
\"	dieresi sulla vocale seguente
``	virgolette aperte
\lq	virgolette aperte
''	virgolette chiuse
\rq	virgolette chiuse

Esistono altri caratteri di controllo per alfabeti con accenti speciali (es. Tedesco, Danese, Svedese, etc) per cui si rimanda al TEXbook.

21. Fonti Predefiniti e User

TEX possiede una serie di fonti predefiniti di l'uso corrente.

\rm	roman	caratteri tipografici
\sl	slanted	caratteri tipografici inclinati
\it	italic	caratteri corsivi
\tt	typewriter	caratteri macchina da scivere
\bf	bold face	caratteri tipografici in neretto

Tutti questi fonti sono in corpo 10. Se si usa `\sl` o `\it`, tornando a `\rm` ricordarsi di usare `\/` (italic correction) per recuperare lo spazio occupato dall'inclinazione dell'ultimo carattere obliquo.

Se l'utente vuole specificare dei fonti diversi, potrà definirli con il comando:

```
\font\mysml=cmr7     % lo stesso fonte di \rm ma in corpo 7.
```

Per i titoli è consigliabile usare un corpo ingrandito, ad esempio, `\font\titlefont=cmr10 scaled\magstep2`. che corrisponde al corpo 10 ingrandito di 2 steps tipografici (circa 144%).

22. Ragruppamento

Per definire un gruppo si usano le graffe `{}`. Le azioni definite all'interno di un gruppo sono locali, cioè il loro effetto cessa quando il gruppo finisce. Esistono due tipi di gruppo a seconda del tipo di comando a cui si riferiscono. Se il comando ha funzione globale, deve essere espresso all'interno del gruppo. Se il comando ha funzione locale, deve essere espresso all'esterno del gruppo. Ad esempio, la scelta del fonte è un comando globale, così se si vogliono scrivere in neretto ***solo*** le n parole seguenti, si avrà `{\bf A few words}`, se non si indica il raggruppamento, il testo che segue il comando `\bf` verrà stampato tutto in neretto. Invece `\centerline` è un comando locale che opera solo sulla parola ***immediatamente*** seguente, perciò se si vuole centrare la scritta `"This is a title"`, si avrà `\centerline{This is a title}`. In questo caso, se si omettono le graffe si centra solo la parola `"this"`. Sono delimitatori di gruppo anche $ e $$ per il modo matematico e le sequenze `\begingroup`, `\endgroup`.

23. Spaziatura

Ricordiamo che una pagina di TEX in formato standard (A4) è larga 6.5in e che 1in contiene 72.27pt tipografici per un totale di 469.755pt. In verticale la dimensione è 8.9in pari a 643.2pt. Per alterare la spaziatura esistono svariati comandi che agiscono rispettivamente in modo orizzontale, verticale o matematico.

I comandi che agiscono in ***modo orizzontale*** sono:

`\enskip`	spaziatura di .5em con em spaziatura pari a una m
`\enspace`	spaziatura di .5em
`\hskip`	spaziatura del valore e unità indicate (es. `\hskip20pt`)
`\kern`	spaziatura del valore e unità indicate (es. -.1667em)
`\quad`	spaziatura di 1em
`\qquad`	spaziatura di 2em
`\negthinspace`	piccola spaziatura negativa (-.1667em)
`\thinspace`	lascia un piccolo spazio predeterminato (-.1667em)
`\/`	italic correction
`\`	\ seguito da blank lascia dello spazio (indicato per la gestione della bibliografia, es. `Am.\ Mat.\ Soc.`)

I comandi che agiscono in ***modo verticale*** sono:

\bigskip	spaziatura di 2 medskip = 4 smallskip
\lower	abbassa i caratteri del valore e unità indicate (es .5ex) con ex spaziatura pari a una x
\medskip	spaziatura di 2 smallskip
\raise	operazione contraria a \lower, alza i caratteri del valore e unità specificati
\smallskip	spaziatura di 3pt
\vskip	lascia uno spazio pari al parametro fornito, es. 12pt

I comandi che agiscono in ***modo matematico*** sono:

\,	spazio piccolo = 1/6 di quad
\>	spazio medio = 2/9 di quad
\;	spazio grande = 5/18 di quad
\!	spazio piccolo negativo = -1/6 di quad.
\mskip	spazio definito dall'utente (es. \mskip 9mu plus 2mu)
\thinmskip	pari a 3mu
\medmskip	pari a 4mu plus 2mu minus 4mu
\thickmskip	5mu plus 5mu

Agiscono sulla spaziatura anche i comandi \hfill e \vfill che servono per alterare gli spazi tra le parole nella riga o agiscono sullo spazio interlinea. Se si vuole imporre un salto pagina con \eject mantenendo l'interlinea in vigore occorre invocare prima \vfill, in caso contrario l'interlinea viene alterata in modo da ridistribuire omogeneamente lo spazio bianco.

24. Sillabazione

La sillabazione viene gestita da TEX secondo le regole della lingua inglese in base a tabelle e ad un dizionario di sillabazione. Le regole inglesi funzionano discretamente per molte parole italiane ma tutte le parole con gn, quali magnificazione, etc. vengono sillabate mag–nificazione invece che ma–gnificazione. Per ovviare a questo inconveniente si può aggiungere una voce al dizionario di TEX con il comando \hyphenation perciò nel caso di magnificazione si avrà: \hyphenation{ma-gni-fi-ca-zio-ne}. Esiste il comando \patterns che serve a inserire il dizionario di sillabazione in TEX. La specifica di \hyphenation va inserita all'inizio del file, prima dell'uso.

Va tenuto presente che TEX non è in grado di distinguire apostrofo e segni di interpunzione dai caratteri alfabetici che compongono la parola perciò in italiano può capitare che TEX non riesca a sillabare sostantivi che iniziano con vocale quando questi sono preceduti da articoli e/o preposizioni articolate con apostrofo come dell'addizione. In questo caso si risolve il problema usando la tilde e cioè dell''~addizione.

25. Sequenze di Controllo

Le sequenze di controllo di uso più frequente sono illustrate di seguito. Per un elenco completo consultare il TEXbook.

\backslash	produce in stampa il carattere \

\baselineskip	distanza tra le righe (default=12pt)
\begingroup	denota l'inizio della struttura
\batchmode	per l'esecuzione in modo batch
\bigskip	spaziatura verticale di 12pt tipografici
\box	identifica i registri di tipo "scatola"
\break	segnala una possibile posizione in cui spezzare una riga o una pagina
\bye	fine elaborazione pari a \vfill\eject\end
\centerline	centra il testo seguente
\char	usa il carattere del valore indicato. Serve per emulare caratteri mancanti sulla tastiera. (es. \char98 produce b).
\chardef	definisce un carattere, come \char ma con nome
\cleaders	come \leaders ma i puntini sono centrati
\cleartabs	annulla tutti i tabulatori settati con \settabs
\dotfill	completa la riga di testo con puntini
\closin	chiude un file in lettura
\closeout	chiude un file in scrittura
\columns	determina il numero di colonne nel comando \settabs
\count	identifica i registri di tipo count
\cr	carriage return, determina la fine della riga nelle tabulazioni
\def	definisce nuovi comandi
\dimen	identifica i registri di tipo dimensione
\divide	divisione fra interi
\eject	forza salto pagina con interlinea modificata
\else	alternativa in una struttura \if
\enskip	spaziatura orizzontale di .5em
\enspace	spaziatura orizzontale di .5em
\end	fine elaborazione
\endgroup	denota la fine della struttura
\endinsert	termina i comandi di inserimento spazio bianco
\filbreak	spezza la pagina compatibilmente con \vsize
\fi	fine struttura \if
\folio	stampa il numero della pagina
\font	definisce un nuovo fonte
\footline	scrive righe in fondo alle pagine
\footnote	gestisce le note a fondo pagina
\halign	crea tabella secondo prototipo dinamico
\hangafter	no. di righe per cui deve durare \hangindent. Se il no. specificato è positivo. Si applica alle n righe seguenti, se negativo alle n righe precedenti (default \hangafter=1)
\hangindent	determina la lunghezza totale della riga con un'indentazione del margine pari al valore specificato. Agisce sul margine sinistro se il valore è positivo, su quello destro se negativo (default \hangindent=0pt)
\hbox	definisce una scatola orizzontale
\headline	setta l'intestazione in cima alle pagine
\hfil	completa la riga con spazio bianco

`\hfill`	completa la riga con spazio bianco (tolleranza maggiore)
`\hidewidth`	altera la centratura delle caselle in `\halign`
`\hoffset`	altera margine orizzontale
`\hrule`	scrive una riga verticale
`\hrulefill`	completa la riga di testo con una linea orizzontale
`\hsize`	determina la dimensione orizzontale secodo valore e unità (default=6.5in)
`\hskip`	spaziatura orizzontale del valore e unità specificate
`\hss`	crea colla orizzontale con estensibilità infinita
`\hyphenation`	detemina le regole di sillabazione
`\if`	inizio struttura condizionale
`\ifcase`	selezione case
`\ifdim`	test su dimensione
`\iffalse`	test se falso
`\ifnum`	test su interi
`\ifodd`	test su dispari
`\iftrue`	test se vero
`\indent`	indenta il nuovo paragrafo
`\input`	lettura da file specificato
`\item`	genera una lista di voci con indentazione corretta
`\itemitem`	genera una sottolista nell'ambito di `\item`
`\kern`	spaziatura orizzontale secondo valore e unità
`$\ldots$`	scrive con spaziatura corretta la stringa ...
`\let`	assegnazione di alias a una definizione
`\leaders`	viene utilizzato per inserire dei puntini in una riga
`\leftline`	come `\line` ma con allineamento a sinistra
`\leftskip`	aumenta la marginatura sinistra
`\line`	crea una scatola orizzontale di dimensione `\hsize`
`\llap`	left overlap, sovrappone caratteri con spostamento a sinistra
`\loop`	costruzione di loop nel calcolo interno ad una macro
`\lower`	spaziatura verticale secondo valore e unità (verso il basso)
`\lq`	virgoletta sinistra (left quote)
`\magnification`	altera la dimensione del foglio stampato globalmente
`\magstep`	specifica lo step di ingrandimento (valori ammessi: half, 0:6)
`\mathchardef`	definizione di simbolo matematico
`\mathchoice`	definizione di macro per i 4 stili matematici (display–text-script-scriptscript)
`\mathstrut`	struttura metematica (crea box di allineamento formule)
`\medskip`	spaziatura verticale di 6pt tipografici
`\message`	scrive righe di messaggio su video
`\midinsert`	inserisce spazio per figure a metà pagina
`\multiply`	moltiplicazione tra interi
`\multispan`	espande una riga su più tabelle gestite da `\halign`
`\narrower`	stringe le righe del prossimo paragrafo
`\negthinspace`	piccola spaziatura orizzontale negativa (-.1667em)
`\newcount`	assegnazione registro interno per interi
`\newdimen`	assegnazione registro interno per dimensioni

\newif	assegnazione registro interno per logici
\noalign	inibisce l'allineamento durante una tabulazione
\nobreak	evita che una riga o una pagina vengano spezzate in quel punto
\noindent	inizia un nuovo paragrafo senza indentarlo
\nopagenumbers	non numera le pagine
\number	scrive un valore intero
\obeylines	rispetta le righe del file sorgente
\obeyspaces	rispetta gli spazi bianchi tra le parole
\omit	omette termini di tabulazione con \halign
\openin	apre un file in lettura
\openout	apre un file in scrittura
\offinterlineskip	elimina la spaziatura interlinea
\pageinsert	inserisce spazio per figure su tutta la pagina
\pageno	altera il no. iniziale delle pagine
\par	delimita un paragrafo
\parfillskip	definisce la spaziatura della riga finale del paragrafo. Se posto = 0, fa sì che l'ultima riga del paragrafo sia lunga come le altre
\parindent	definisce il valore dell'indentazione per i paragrafi (default)
\parshape	determina la forma del paragrafo in base alle dimensioni specificate (default)
\parskip	tolleranza tra paragrafi (default opt plus1pt)
\patterns	definisce la struttura delle sillabe
\quad	spaziatura orizzontale di 1em
\qquad	spaziatura orizzontale di 2em
\raggedbottom	elimina overfull verticale (analogo a \raggedright)
\raggedright	elimina overfull con bordi destri disallineati
\raise	spaziatura verticale secondo valore e unità (verso l'alto)
\read	legge da file esterno o da video
\relax	no operation (serve come termine di comandi per risolvere ambiguità e sincronismo)
\repeat	termine dell'iterazione iniziata da \loop
\rightline	come \line ma con allineamento a destra
\rightskip	aumenta la margimatura destra
\rlap	sovrapposizione caratteri con spostamento a destra
\romannumeral	stampa un intero in cifre romane
\rq	virgoletta destra (right quote)
\S	stampa il simbolo di paragrafo
\settabs	setta le posizioni di tabulazione
\show	stampa il valore di una sequenza (debug)
\showbox	stampa scatole per debug
\showlists	stampa liste per debug
\showthe	visualizza il valore del parametro specificato
\skip	definisce i registri di "skip"
\smallskip	spaziatura verticale di 3pt tipografici
\span	estende il testo in una colonna di una tabulazione
\special	inserisce comandi speciali

\tabskip	definisce l'entità di spazio libero tra le colonne di una tabella
\TeX	stampa il logo di TEX
\the	visualizza un registro secondo il tipo
\thinspace	spaziatura orizzontale di .1667em
\tracingcommands	traccia i comandi elaborati (debug)
\tracingmacros	traccia le macro elaborate (debug)
\tracingpages	traccia le pagine prodotte (debug)
\tracingparagraphs	traccia i pargrafi prodotti (debug)
\tolerance	aumenta la tolleranza di impaginazione orizzontale. Valore tipico=1600 (possibili overfull), =10000 valore infinito (possibili underfull)
\topinsert	inserisce spazio per figure in cima alla pagina
\topskip	determina lo spazio bianco in cima alla pagina (default)
\undebar	sottolineatura del testo (non in modo matematico)
\vbox	definisce una scatola verticale
\vcenter	centra verticalmente una scatola orizzontale
\vfil	completa la pagina con spazio bianco
\vfill	completa la pagina con spazio bianco (tolleranza maggiore)
\voffset	altera margine verticale
\vrule	scrive una riga verticale
\vsize	determina la dimensione verticale secondo valore e unità (default=8.9in)
\vskip	salto verticale secondo valore unità
\write	scrive su file di uscita
\xleaders	come \leaders ma espansi

26. Sequenze Matematiche

Le sequenze matematiche di uso più frequente sono illustrate di seguito. Per un elenco completo consultare il TEXbook.

^	esponente
_	indice, limite inferiore, etc.
\above	produce il segno di frazione come \over ma con lo spessore indicato (es. \above1pt aumenta lo spessore di 1pt), utile nelle frazioni composte
\abovewithdelims	come \above ma aggiunge i delimitatori indicati
\alpha	lettera greca alfa, e analogamente tutte le lettere minuscole comprese le lettere per le variabili e cioè \varphi, \varrho \vartheta, \varpi. Le maiuscole sono invocate con inziale maiuscola (es.\Gamma), ne esiste solo un set limitato all'uso matematico
\approx	circa
\arcos	arcocoseno
\arcsin	arcoseno
\arctan	arcotangente
\atop	analogo a \choose ma senza parentesi
\atopwithdelims	come \atop ma aggiunge i delimitatori indicati
\bar	barra sopra al simbolo
\big	designa un delimitatore (con l per left ed r per right)

\Big	come sopra (più grande)
\bigg	come sopra (più grande)
\Bigg	come sopra (più grande)
\bullett	cerchietto pieno
\cases	sequenza di assegnazione valori alternativi
\cdot	puntino di moltiplicazione
\cdots	3 puntini centrati sul segno della formula
\circ	cerchietto vuoto
\choose	binomio
\cos	coseno
\cosh	coseno iperbolico
\cot	cotangente
\coth	cotangente iperbolica
\ddots	3 puntini diagonali
\det	determinante
\displaylines	gestione formule complesse con o senza numerazione
\displaystyle	forza l'uso del fonte per formule isolate
\ell	l minuscola corsiva matematica
\eqalign	allinea equazioni
\eqalignno	allinea e numera le equazioni a destra
\eqno	numera le equazioni a destra
\equiv	equivalente
\Gamma	lettera greca gamma maiuscola
\infty	simbolo di infinito
\int	integrale
\ldots	3 puntini orizzontali
\le	minore o uguale
\left	designa delimitatore sinistro
\leqalignno	allinea e numera le equazioni a sinistra
\leqno	numera le equazioni a sinistra
\lim	limite
\liminf	limite inferiore
\limits	pone i limiti di inegrale, sommatoria, etc. sopra e sotto
\limsup	limite superiore
\ln	logaritmo naturale
\log	logaritmo
\matrix	gestisce le matrici, delimitatori espliciti
\max	massimo
\medmskip	spaziatura matematica analoga a \medskip
\min	minimo
\mit	fonte italico per lettere matematiche
\mp	meno o più
\mskip	skip matematico (analogo a \hskip e \vskip)
\muskip	designazione dei registri di skip in unità mu
\ne	non uguale

`\neq`	non uguale
`\noalign`	elimina l'allineamento durante comandi `\eqalign` etc.
`\nolimits`	pone i limiti di integrale, sommatoria, etc. a lato
`\not`	negazione di altro simbolo (es. `\equiv`)
`\over`	segno di frazione
`\overbrace`	parentesi graffa orizzontale posta sopra alla formula
`\overline`	linea sopra alla variabile
`\overwithdelims`	come `\over` ma aggiunge i delimitatori indicati, ad esempio $\left(\frac{a}{b}\right)$ va scritto `a\overwithdelims() b`
`\pi`	pi–greco
`\pm`	più o meno
`\pmatrix`	gestisce matrici, delimitatori impliciti
`\prime`	derivata prima
`\root n \of`	radice n (variabile) di
`\scriptstyle`	fonte indici del primo ordine
`\scriptscriptstyle`	fonte indici del secondo ordine
`\sec`	secante
`\sim`	simile
`\simeq`	simile o uguale
`\sin`	seno
`\sinh`	seno iperbolico
`\sqrt`	radice quadrata
`\sum`	sommatoria
`\strut`	gestisce scatole verticali in modo matematico
`\tan`	tangente
`\tanh`	tangente iperbolica
`\textstyle`	fonti matematici nelle dimensioni di tipo text
`\thickmskip`	spaziatura matematica di 5mu
`\thinmskip`	spaziatura matematica di 3mu
`\times`	segno di moltiplicazione ($\times$)
`\underbrace`	parentesi graffa orizzontale posta sotto la formula
`\underline`	sottolineatura
`\varphi`	variabile phi
`\varpi`	variabile pi
`\varrho`	variabile ro
`\vartheta`	variabile theta
`\vec`	vettore
`\vdots`	3 puntini verticali

27. I tools

Come abbiamo già avuto modo di constatare nella precedente esposizione TEX è un sistema complesso per la cui costruzione, manutenzione e gestione necessita di "tools" preposti alle varie funzioni, quali manutenzione dei programmi e dei fonti, stampa e visualizzazione del testo prodotto, supporto di debug in caso di problemi.

Normalmente in una installazione funzionante oltre a TEX, gli unici tools di uso comune sono gli spoolers, ossia i programmi che producono l'uscita su stampanti o video grafici per la lettura e la visualizzazione. Occasionalmente può presentarsi la necessità di usare alcuni dei programmi di supporto per la gestione dei fonti. Infine è possibile che si presenti la necessità di utilizzare altri tools se si devono ricompilare e/o debuggare i programmi di uso comune in caso di problemi o di incompatibiltà con il sistema operativo. In questa sezione daremo una descrizione introduttiva dei tools a disposizione, rimandando ad altre documentazioni per una conoscenza più completa.

Per consentire una migliore comprensione, manutenzione e implementazione di TEX su macchine e con sistemi operativi diversi, l'autore (D. E. Knuth) ha sviluppato il sistema `WEB` [4] di programmazione documentata e strutturata che consente di scrivere un programma con istruzioni TEX e PASCAL combinate in modo da produrre un assieme strutturato in cui TEX consente di illustrare la struttura e i concetti che le stanno dietro per ogni parte di programma, mentre PASCAL specifica gli algoritmi in modo formale e non ambiguo. Ogni sezione così composta prende il nome di modulo.

Il sistema `WEB` è composto da 2 elementi: `WEAVE` che elabora la parte di documentazione del programma e produce il file `.TEX` che sostituisce la stampa tradizionale e `TANGLE` che produce il file `.PAS` con il codice PASCAL. Il programma così suddiviso in moduli viene manipolato e modificato tramite files opportuni detti change `files`. Inoltre `WEAVE` adotta un sistema particolare per la gestione delle stringhe `ASCII` che vengono codificate sotto forma di interi e vengono trascritte in uscita sul file ausiliario `.POO` denominato "string pool file". Dal punto di vista operativo dato un file `.WEB`, ad esempio `TEST.WEB`, per produrre il programma eseguibile e la relativa stampa si avrà:

```
$ WEAVE  TEST                                    !write .TEX
$ TEX    TEST                                    !write .DVI
$ DVIQMS TEST                                    !write .BIT
$ PQMS   TEST                                    !print TEST listing
$ TANGLE TEST/CHANGE=TEST.CH/PASCAL=TEST.PAS     !write .PAS
$ PAS    TEST                                    !compile TEST
$ LINK   TEST                                    !create .EXE
```

Normalmente sia l'utente medio che l'installatore non sono interessati al sistema `WEB`, è però importante conoscerne i rudimenti perchè il nastro di distribuzione contiene tutti i sorgenti in `WEB` e il sistema `WEAVE/TANGLE` potrebbe essere necessario nel caso in cui si debba procedere ad una ricompilazione del programma. L'esempio che segue illustra l'aspetto della stampa di un modulo `WEB`.

```
20. (Read one string, but abort if there are problems)
for k 1 ← to l do
  begin if eoln(pool_file) then
    begin print_ln(new_line); abort;
    end;
  read(pool_file; print(xchr[xord[m]]);
  if xord[m] = title then print(xchr[title]);
  end;
print_ln(sub_title); incr(s);
end
```

Oltre al sistema WEB, che dovrebbe essere installato per primo su di una macchina e/o sistema operativo per cui non esista già un'implementazione di TEX, esistono svariati programmi di utilità generale che vanno sotto il nome generico di TEXware [5] . Uno di questi è POOLtype che produce una stampa interpretata del file .POO Prima di descrivere le altre utilità di TEXware, dobbiamo riprendere in esame il modo di lavoro di TEX.

Come abbiamo già visto, TEX elabora un testo di tipo .TEX e produce un file destinato alla stampa di tipo .DVI, cioè "device independent". Il file .DVI è in formato binario e per motivi di efficienza è scritto con records di lunghezza fissa. Il formato del .DVI è descritto nel TEXbook e contiene dei codici operativi e degli operandi in modo simile a quanto avviene per molti linguaggi "assembler". Il programma DVItype produce una stampa interpretata del file .DVI. Esiste un altro programma (TXLISDVI) per la stampa del .DVI che adotta criteri differenti di interpretazione e impaginazione e che fa parte dei tools di supporto di TXMAPPER[6] .

Per produrre una corretta spaziatura e impaginazione, TEX ha bisogno della descrizione geometrica dei fonti che vengono usati nel testo e precisamente ne deve conoscere la misura in punti tipografici, l'eventuale inclinazione, i gruppi che determinano kerning o legatura, lo spazio consentito per il massimo avvicinamento e allontanamento dal carattere precedente, etc. Queste informazioni sono contenute in files opportuni denominati "font metric files" di tipo .TFM. Anche questi files sono formati da records di lunghezza fissa e quindi non sono stampabili. Il programma TFtoPL crea i files .PL dai .TFM corrispondenti e contemporaneamente ne controlla la validità segnalando eventuali errori. I files .PL sono file ASCII quindi possono essere stampati e corretti con un editor. Nel caso in cui si voglia creare un nuovo fonte, formato ad esempio da caratteri semigrafici, occorre editare un file .PL con la descrizione dei caratteri generati, poi con il programma PLtoTF che esegue la conversione inversa si crea il file .TFM che consente di richiamare da TEX il nuovo fonte.

I fonti distribuiti con il nastro di TEX sono stati generati da METAFONT[7] , un programma di generazione caratteri sviluppato in funzione di TEX. METAFONT elabora il file che descrive il fonte (file .MF) e scrive il file .TFM con la descrizione geometrica dei caratteri destinato a TEX e il corrispondente file .GF analogo al .DVI con la descrizione fisica dei pixel che compongono il fonte e che deve essere interpretato e trasformato opportunamente in funzione del device di stampa. Per le stampanti laser viene generata una bit map dei fonti. Esistono due tipi di files di bitmap: in pixel (files .PXL) o in forma compatta (files .PK). Per la gestione e la creazione dei fonti sono disponibili i seguenti tools.

MF	METAFONT
GFtoPK	converte .GF in .PK
GFtoPXL	converte .GF in .PXL
PKtoPX	converte .PK in .PXL
PXtoPK	converte .PXL in .PK

Infine per la stampa e la visualizzazione del testo esistono spoolers per molti device grafici. I tools di cui disponiamo sono: DVIQMS che pilota le stampanti laser QMS modello 800 e 1200 con protocollo grafico QUIC e TXMAPPER che consente di visualizzare il testo sui seguenti VT grafici: Tektronix 40xx, 41xx, DEC con protocollo ReGiS (es. VT240, VT241, VT330, VT340) e DEC VAX Station GPX. Entrambi i programmi fanno uso dei files \.PXL e sono disponibili sia come eseguibili che come sorgenti. Esiste infine lo spooler DVIDIS * per la visualizzazione su DEC VAX Station

* Autore della versione di cui disponiamo è Jerry Leichter della Yale University

GPX che usa i files .PK e di cui è disponibile il solo .EXE. Sia DVIQMS che TXMAPPER sono stati scritti per VAX

VMS e fanno uso dei comandi tipici del VMS, DVIDIS invece è stato adattato da UNIX perciò usa la sintassi di UNIX per i comandi ed è meno flessibile nell'uso attraverso procedure DCL. Nel nostro archivio DECnet sono memorizzati altri spoolers per stampanti laser con e senza protocollo PostScript e visualizzatori per VT grafici di cui non disponiamo e per i quali non siamo in grado di fornire ulteriori notizie.

Per il controllo e la visualizzazione dei singoli caratteri di un fonte esiste infine il programma TXLISPXL che consente di selezionare in modo interattivo un singolo carattere o un gruppo di elementi del fonte per la visualizzazione su VT non grafici. Dato che il programma è molto veloce, funziona su qualsiasi video e stampa la tabella dei caratteri presenti nel fonte in uso, si rivela strumento molto utile per l'analisi e la scelta dei corpi disponibili in funzione di una particolare veste tipografica.

Dato che TEX è un vero e proprio linguaggio di programmazione a cui si può associare un sistema di macro definizioni anche molto complesso, sono stati sviluppati ulteriori prodotti che facilitano l'uso di TEX per l'utente meno esperto consentendo di gestire in modo piú semplice l'impaginazione, la creazione di tabelle e altre operazioni complesse. In questa sede daremo solo un rapido elenco di alcuni di questi tools.

Il sistema piú ampio e completo dedicato alla scrittura di articoli e libri in maniera semplice e automatica è LATEX[8] . Sia TEX che LATEX hanno riscosso enorme successo, di conseguenza sono stati sviluppati ulteriori pacchetti che fanno uso di uno dei due prodotti. Va notato che, sebbene LATEX sia stato scritto su TEX, la sintassi dei due linguaggi è differente e non si possono mescolare impunemente istruzioni di TEX e di LATEX. Dato che LATEX è in grado di scrivere files ausiliari, in appoggio a LATEX è stato sviluppato da Oren Pateshnik BIBTEX un sistema per la gestione automatica della bibliografia e da Leslie Lamport SLiTEX un sistema per la produzione di "slides" ossia di uscite adatte a essere copiate su supporto trasparente colorato da visionare per sovrapposizione.

Il sistema TEX sis [9] è stato scritto come libreria di macro TEX per la produzione di documenti di fisica dall'articolo all'atto del congresso, al libro. Alcune delle capacità più interessanti di TEX sis sono la numerazione automatica delle equazioni con riferimento simbolico, la gestione automatica dei riferimenti bibliografici, la suddivisione in capitoli, paragrafi e sezioni, la creazione dell'indice, la gestione semplificata di tabelle.

Sfruttando la possibilità di personalizzazione di TEX e LATEX si possono alterare le tabelle di sillabazione, sono perciò state sviluppate le due varianti ITEX e ILATEX ** che gestiscono le regole di divisione in sillabe valide per la lingua italiana.

Infine esistono dei tools che agiscono da post–processori del .DVI. Uno di questi è TEXTYL *** che consente di inserire dei comandi grafici nel testo come argomento di \special e opera sul .DVI prodotto generando un nuovo file .DVI in cui i comandi grafici specifici sono stati trasformati in comandi validi per TEX e comprensibili da qualsiasi spooler.

Come ulteriore tool possiamo citare il programma SPELL **** che opera in modo interattivo un controllo ortografico per la sola lingua inglese e consente di affiancare al suo dizionario interno di 90000 vocaboli un dizionario utente.

** I programmi ITEX eILATEX sono stati implementati da Massimo Calvani dell'Istituto di Astronomia di Padova in base alle regole di sillabazione fornite dal prof. Patergnani del Dipartimento di Fisica dell'Universotà di Padova

*** Versione originale di John Renner, Ohio State University, versione in nostro posseso di Jerry Leichter, Yale

**** Purtroppo non siamo in grado di citare l'autore

Per la maggior parte di questi tools esistono manuali e/o documentazioni accessibili elettronicamente e memorizzati sul disco che contiene l'archivio DECnet di TEX. Ulteriori fonti di informazioni sono il bolettino TUGboat edito dall'$\mathcal{AMS}$ e distibuito ai soci del TEXUsers Group e il bolettino eletronico TeXHaX edito da Malcolm Brown, Stanford Univesity e disponibile via rete sull'archivio DECnet.

28. L'installazione

TEX viene venduto per un valore nominale dall'$\mathcal{AMS}$. Dato che si tratta di un sistema di "dominio pubblico" anche se di proprietà dell'$\mathcal{AMS}$, il nastro di distribuzione contiene i sorgenti di tutti i programmi scritti da D. E. Knuth e di molti altri prodotti sviluppati come supporto da parte di istituti universitari e di ricerca. TEX è anche disponibile via rete e l'archivio DECnet per la comunità di Fisica delle Alte Energie si trova su di una macchina INFN presso il CNAF. Il software di cui disponiamo ci è stato fornito da Massimo Calvani dell'Istituto di Astronomia dell'Università di Padova e comprende oltre a TEX e ai tools descritti precedentemente molti altri pacchetti applicativi forniti da utenti TEX sia in Europa che in USA.

Data la vastità del materiale raccolto che comprende moltissimi tools, esempi, documentazioni, etc. sparsi su più di 40 directory con un'occupazione totale di oltre 120 Mbytes, forniamo un' indicazione dei files da copiare via rete essenziali all'installazione, all'uso e alla stampa di TEX.

Per prima cosa occorre creare una directory, generalmente ma non necessariamente `[TEX]`. Al CNAF la "master directory" si chiama `[TEXNET]`. Le subdirectory necessarie sono `[.EXE]`, `[.FONTS]`, `[.FORMATS]`, `[.PIXEL]` o in alternativa `[.PK]`, si possono poi creare `[.CLD]` e `[.INPUTS]`. Si devono copiare tutti i files di `[.FONTS]` e `[.FORMATS]`, i soli file da copiare da `[.EXE]` e da `[.CLD]` sono rispettivamente `TEX.EXE` e `TEX.CLD` che può essere copiato in un area temporanea fino a che non è completata l'installazione. Infine occorre copiare i fonti bit map per la stampa, in formato `.PK` e/o `.PXL` a seconda dei driver che interessano per i quali vanno copiati gli eseguibili su `[.EXE]`. Per maggiori informazioni si consiglia di consultare la guida alla distribuzione disponibile via rete e mantenuta aggiornata dagli autori di questo articolo. Per quanto riguarda i fonti in formato bit map, si possono copiare tutti o parte a seconda dello spazio disponibile secondo i suggerimenti forniti di seguito.

Per operare TEX ha bisogno di due informazioni: i dati sui fonti e le definizioni primitive e macro di base, inoltre è possibile definire un'area in cui andare a cercare eventuali altre definizioni locali. Queste informazioni vengono cercate in aree disco definite dai simboli logici `TEX_FONTS`, `TEX_FORMATS` e `TEX_INPUTS`. TEX e la maggior parte dei tools sono stati scritti facendo uso del "VAX Command Language" con il quale si crea un nuovo "verbo" che può essere aggiunto alle tabelle di sistema dal system manager o alle tabelle private dell'utente tramite il comando `Set Command`. Perciò dopo aver copiato i files suddetti sarà necessario creare un file di comando del tipo seguente da eseguire per installare TEX.

```
$  define TEX_DISK        $disk2
$  define TEX_FORMATS     tex$disk:[tex.formats]
$  define TEX_FONTS       tex$disk:[texnet.fonts], -
tex$disk:[texnet.amstex.amsfonts.tfm]
$!
$  set command/replace tex$cld:tex.cld
```

Si possono suddividere i dati relativi a fonti, macro, formati, etc. su più aree disco per semplificare le gestione e anche per consentire a ciascun utente l'aggiunta di definizioni personali facendo uso dell'opzione "search list" per i simboli logici. Da notare che i nomi logici del tipo `TEX_` si riferiscono a TEX 2.00 o seguenti, mentre per le versioni precedenti i nomi logici erano del tipo `TEX$`, forma che è stata abbandonata per evitare possibili conflitti con le definizioni del sistema operativo. Se si vuole eliminare un verbo definito con `set command`, si può usare il comando `set command/delete=(tex)`.

Se non si inseriscono i nomi dei tools nelle tabelle `DCL` di sistema e se ne installano molti a livello utente il tempo richiesto dall'operazione è decisamente sensibile, perciò si consiglia di utilizzare una procedura pilotata da un menu per un'installazione selettiva eventualmente dotata di messaggi che consentano di seguire l'operazione in corso.

Per quanto riguarda i fonti, sono disponibili le famiglie descritte in tabella nei corpi indicati e agli ingrandimenti `magstep` 0, 1/2, 1, 2, 3, 4, e 5.

CMR	17, 12, 10, 9, 8, 7, 6, 5	roman text
CMMI	12, 10, 9, 8, 7, 6, 5	math italic
CMSY	10, 9, 8, 7, 6, 5	math symbols
CMEX	10	math extension
CMBX	12, 10, 9, 8, 7, 6, 5	boldface extended
CMSL	12, 10, 9, 8	slanted roman
CMTT	12, 10, 9, 8	typewriter
CMTI	12, 10, 9, 8, 7	text italic
CMSS	17, 12, 10, 9, 8	sans serif
CMSSI	17, 12, 10, 9, 8	sans serif italic
CMSSQ	8	sans serif quotation
CMSSQI	8	sans serif quotation italic
CMB	10	boldface
CMBSY	10	boldface math symbols
CMBXSL	10	boldface slanted
CMBXTI	10	boldface italic
CMITT	10	italic typewriter
CMCSC	10	small caps (uppercase)
CMTCSC	10	typewriter small caps
CMDUNH	10	dunhill
CMFF	10	funny font
CMFI	10	Fibonacci
CMFIB	8	bold Fibonacci
CMINCH		sans serif
CMMIB	10	math italic bold
CMSLTT	10	slanted typewriter
CMSSBX	10	sans serif bold extend
CMSSDC	10	sans serif demi cond
CMTEX	10, 9, 8	tex typewriter
CMU	10	unslanted italic
CMVTT	10	var-width typewriter
CIRCLE	10	LaTeX circles

CIRCLEW	10	LaTeX wide circles
LINE	10	LaTeX lines
LINEW	10	LaTeX wide lines
LASY	10, 9, 8, 7, 6, 5	specials
LASYB	10	bold specials
LOGO	10, 9, 8	letters A E F M N O T
LOGOBF	10	bold letters A E F M N O T
LOGOSL	10	slanted letters A E F M N O T

I vari ingrandimenti espressi rispettivamete in valori di `\magnification` e in dots/inch a partire da 1000 e da 1500 sono indicati nella seguente tabella.

1000	200	1500	300
1095	219	1643	329
1200	240	1800	360
1444	288	2160	432
1728	345	2595	518
2074	415	3110	622
2488	497	3732	896

Normalmente i fonti più usati sono: CMR, CMMI, CMSY, CMEX, CMBX, CMSL, CMTT, CMTI, CMSSQ nei corpi 10, 8, 7, 5 dove disponibili e agli ingrandimenti 1500, 1643 e 1800, perció se non si dispone di abbastanza spazio disco ci si può limitare a questi fonti ed eventualmente copiarne altri in caso di necessità. Per i file che contengono i fonti sia di tipo `.PK` che `.PXL` esitono diverse convenzioni di nomenclarura per specificare l'ingrandimento del fonte. Il nome del file è dato dalla famiglia (ad es. CMR, CMI, etc.) seguito dal corpo (ad es. 10, 8, 7, etc.). Il tipo del file indica il formato, cioè `.PK` o `.PXL`. Per l'ingrandimento sono possibili due diverse convenzioni: i file hanno tutti lo stesso nome e sono memorizzati in diversi sottodirettori, ad esempio `[TEX.PIXEL.1500]`, i file sono memorizzati in un'unica area disco (es. `[TEX.PIXEL]` e l'ingradimento fa parte del nome, ad esempio `CMR10.1500PXL`. Infine l'ingrandimento può essere indicato in termini di magnificazione o in dots/inch.

Il programma `TEX.EXE` che viene distribuito è stato creato per usare il formato descritto da `PLAIN.TEX` e le regole di sillabazione descritte da `HYPHEN.TEX`. Per generare l'eseguibile dal sorgente `WEB` si crea il programma `INITEX.EXE` applicando a `TEX.WEB` il change file `INITEX.CH` con il quale si produce il formato desiderato con un comando del tipo:

```
$ initex/nobatch "\input plain \dump"
```

con il quale si genera il file `PLAIN.FMT`. Variando il file di formato si possono creare altri `.FMT`, ad esempio per generare LaTeX si usa il file `LPLAIN.TEX`.

Per generare TeX si esegue `WEAVE` applicando a `TEX.WEB` il change file `TEX.CH` che crea un eseguibile convenzionalmente chiamato `VIRTEX.EXE` che carica il formato `PLAIN.FMT` dinamicamente a "run time". È possibile generare una versione con il fomato "preloaded" con il comado:

```
$ virtex/nobatch "\relax\end" save/traceback tex
```

Per maggiori informazioni sulla generazione di versioni alternative di TeXquali LaTeX, TeX sis, etc. si consiglia di consultare il relativo file `BUILD.COM` memorizzato assieme al pacchetto di installazione.

Bibliografia

[1] The TEXbook, Donald E. Knuth, Addison Wesley Publishing Company

[2] B. Beeton, Mathematical Symbols and Cyrillic Fonts, TUGboat, Volume 6 (1985), No. 3

[3] TUGboat, Vol. 6, No. 2, pag. 83

[4] The WEB System of Structured Documentation, Donald E. Knuth, Stanford University

[5] TEXware, Donald E. Knuth, Stanford University

[6] A TEX82 spooler for VT and dot matrix printers, M. L. Luvisetto – E. Ugolini, TUGboat, Vol. 6, No. 1

[7] The METAFONTbook, Donald E. Knuth, Addison Wesley Publishing Company

[8] LATEX, A Document Preparation System, Leslie Lamport, Addison Wesley Publishing Company

[9] A TEXFormat for Physics, Eric Myers – Boston University & Frank E. Paige, Brookhaven National Laboratory