Editori di testo per scrivere in TeX

Word rispetto agli altri editori.

Spiace doverlo ammettere, ma MSWord nella release 5.1 (non nelle successive che sono piene di complicazioni e meno affidabili), rispetto agli altri editori **di testo** che conosco ha alcune caratteristiche che lo rendono insostituibile:

• Drag-and-drop

È possibile spostare o copiare testo semplicemente selezionandolo e trascinandolo. È molto più veloce del tradizionale taglia-incolla o copia-incolla, soprattutto per piccole quantità di testo.

• Selezionare

È possibile selezionare un'intera riga o un intero paragrafo con uno o due click senza dover trascinare il mouse cercandone inizio e fine o senza dover cliccare tre volte.

• Cerca e cambia

È possibile eseguire il "cambia tutto" solo nel testo selezionato e non in tutto il documento o mediante la verifica uno per uno.
Questa è una caratteristica di un'utilità incredibile e insostituibile.
Purtroppo nelle release successive al 5.1 è più complicato e meno affidabile.

• Memoria del punto di inserzione

Si deve copiare del testo che si trova nel documento lontano dal punto in cui si sta lavorando ? Come si fa ? Si cerca il testo da copiare e lo si copia. Come si fa a ritornare al punto di prima ? C'è un tasto che lo fa automaticamente perché ricorda gli ultimi tre punti di inserzione. Un incredibile risparmio di tempo e di stress.

• Formattazione del testo

È possibile di applicare formattazioni (bold, italic, colore, dimensione) con un click o con una combinazione di tasti. A che servono gli stili in un file di semplice testo ? Servono, servono eccome a muoversi velocemente in documento lungo e a riconoscere titoli, a marcare i punti notevoli etc.,

• Memoria dell'ultima azione svolta

Si deve scrivere per venti volte di seguito la stessa cosa in venti punti diversi ? C'è un tasto che memorizza l'ultima azione o combinazione di tasti. In moltissimi casi è più veloce del copia-incolla.

• Il word-wrap

Word effettua automaticamente il word-wrap e lo effettua senza inserire un simbolo di andata a capo come fanno certi editori (per esempio quello di Textures). I simboli di andata a capo inseriti dagli editori che non supportano il soft word-wrap sono invisibili e ostacolano l'editing del testo quando questo coinvolge due o più righe.

• Cose minori:

- È possibile cancellare in avanti oltre che indietro. Sembra incredibile, ma lo si usa come e più del back space.
- Si può cancellare un'intera parola in avanti o indietro con un solo tasto.
- Il tasto Home funziona (e porta il cursore all'inizio del testo).
- E anche il tasto End funziona (e porta il cursore alla fine del testo).

- Ci sono tasti per muoversi all'inizio o alla fine di una riga o di una parola.
- C'è un comando per cambiare il testo selezionato da minuscolo a maiuscolo o viceversa e anche per rendere maiuscola solo l'iniziale.
- C'è un comando per estendere la selezione del testo da cursore fino a un certo carattere. Spesso è molto meglio che trascinare il mouse millimetricamente.
- È possibile visualizzare i caratteri invisibili tipo i [TAB], i [RETURN] e gli spazi bianchi che spesso interferiscono col testo.

Il programma TexEdit nel Mac.

L'unico editore di testo per Mac in grado di competere col vecchio Word 5 è TexEdit di Tom Bender (Tex sta per Texas, lo stato dell'autore, non per Text o TeX).
Purtroppo dal 2006 il suo creatore ha abbandonato il progetto e non lo aggiorna più. Ma il programma funziona ottimamente anche col sistema 10.

• Drag-and-drop

Come nel Word 5, anzi meglio (consente di farlo anche esternamente al programma).

• Selezionare

Come nel Word

• Cerca e cambia

È possibile eseguire il "cambia tutto" solo nel testo selezionato, ma è un po' più complicato che nel Word 5.

• Memoria del punto di inserzione.

Questa purtroppo (per quanto mi risulta) manca. È un grosso difetto.

• Formattazione del testo

È possibile di applicare formattazioni (bold, italic, colore, dimensione) con un click o con una combinazione di tasti, ma occorre AppleScript, quindi richiede la conoscenza di Applescript.

• Memoria dell'ultima azione svolta

Anche questa purtroppo (per quanto mi risulta) manca.

• Il word-wrap

Come nel Word 5.

• Cose minori

- È possibile cancellare in avanti oltre che indietro.
- È possibile cancellare un'intera parola in avanti o indietro con un solo tasto.
- I tasti Home e End funzionano.
- Ci sono tasti per muoversi all'inizio o alla fine di una riga.
- Si può cambiare il testo selezionato da minuscolo a maiuscolo o viceversa.
- È possibile visualizzare i caratteri invisibili tipo i [TAB], i [RETURN] e gli spazi bianchi che spesso interferiscono col testo.

Altri editori (Alpha, BBedit, TextWrangler, TexShop).

Sono editori di puro testo. Ho provato a usarli, ma mancano molte delle cose sopradescritte (più che nel TexEdit). Hanno un sacco di funzioni esperte di ricerca

e cambiamento testo che però non sono poi tanto utili per il TeX.

La cosa peggiore è che sono **editori di puro testo** e non è possibile applicare manualmente formattazioni come Bold e colore per segnare i punti importanti.

Il fatto che per esempio colorino automaticamente le keyword del TeX può essere utile, ma non così tanto, come sto ora per spiegare.

Il programma TextEdit.

Viene fornito da Apple in ogni computer. È un pessimo editore di testo. Mancano quasi tutte le cose che rendono buono TexEdit e Word.

Mancano la visualizzazione dei caratteri invisibili, la formattazione con un tasto, il cambio da maiuscolo a minuscolo, il "trova e sostituisci solo nel testo selezionato".

Il drag-and-drop funziona malissimo etc. È quasi impossibile usarlo per comporre il sorgente di un file TeX, ma anche per altri usi è assai poco pratico. Eppure c'è gente che lo trova buono. Mah!

Il metodo Odetti per scrivere i testi in TeX

Scrivendo i testi in TeX la difficoltà maggiore è quella di ricordarsi le keywords e di scriverle correttamente: Per esempio:

```
\displaystyle     \emptyset        \rightarrow      \infty         \`{\i}
```

Non è facile ricordarle e comunque è facilissimo commettere errori di battitura per quelle più complicate e avere per esempio:

```
\displaystile     \emtpyset        \rigtharrow      \intfy         \`{\i{
```

L'ultimo errore può addiritura essere serio, dato che il lasciare aperta una parentesi graffa può causare errori a ripetizione e in alcuni casi interrompere il processo di typesetting.

Oltretutto spesso occorre lasciare uno spazio dopo una keyword per evitare che la lettera seguente sia interpretata come parte della keyword. Per esempio

`\intf(x)` per $\int f(x)dx$ è errato perché `\intf` non è una keyword.

`\lambdav` per λv è errato perché `\lambdav` non è una keyword.

Ma anche ricordando le keywords (parecchi editori hanno dei menu che le elencano con varie facilities, anche se il loro uso non è sempre agevole) e battendole correttamente (ma poi spesso nell'editing si può per errore cambiarle) si ha spesso uno scritto praticamente illegibile ed editabile con difficoltà:

Per esempio:

```
$\varphi:[0\:,1]\rightarrow{\bf R}\quad \varphi(x)=\mid\alpha x\mid-\sqrt x$
```

di difficile lettura, genera semplicemente la formula:

$$\varphi : [0 , 1] \rightarrow \mathbf{R} \quad \varphi(x) = \mid \alpha x \mid - \sqrt{x}$$

Anche nel seguente codice è difficile individuare le varie colonne della matrice.

```
$\left(\begin{array}{cccc}
1 & -2 & \alpha+2 &-\sqrt{3}\\
\cos(\pi/3) & 1-k^2 & 0 &2
\end{array}\right)$
```

$$\begin{pmatrix} 1 & -2 & \alpha+2 & -\sqrt{3} \\ \cos(\pi/3) & 1-k^2 & 0 & 2 \end{pmatrix}$$

Idea base

L'idea è quella di utilizzare i caratteri non ASCII della tastiera (quelli generati col tasto [ALT]) per scrivere su schermo mediante simboli le principali keyword TeX. Per fare questo occorre però lavorare con un font particolare. Ho creato il font "otx3" che contiene molti simboli.

Per esempio per generare `\sqrt` si batte sulla tastiera **[shift] [alt] H** e si ottiene:

Nel font Times	Ó	(il Font Times è il più usato con MsWord)
Nel font Monaco	Ó	(il Font Monaco è il più usato con Textures e TeXShop)
Ma nel font otx3	√	

Per generare `\rightarrow` si batte sulla tastiera **[alt] 2** e si ottiene

Nel font Times	™
Nel font Monaco	™
Ma nel font otx3	→

- **È molto più facile ricordare queste combinazioni di tasti, che le keywords.**
- **Ci vuole molto meno tempo che a cercarle in un lungo menu.**
- **Non c'è pericolo di sbagliare la sintassi.**
- **Non ci si deve preoccupare di inserire spazi dopo le keyword quando occorre.**
- **Non c'è da inserire il carattere \ (backslash).**

Per esempio la formula precedente

$$\varphi : [0 , 1] \rightarrow \mathbf{R} \quad \varphi(x) = \mid \alpha x \mid - \sqrt{x}$$

si scrive, usando il font otx3 semplicemente:

$φ:[0␣␣,␣␣1]→ℝ␣φ(x)=|αx|−√x$

invece che

```
$\varphi:[0\:,1]\rightarrow{\bf R}\quad \varphi(x)=\mid\alpha x\mid-\sqrt x$
```

È evidente il risparmio di tempo e di caratteri, non c'è preoccupazione di inserire spazi dopo le keywords quando occorre e soprattutto c'è maggior affidabilità e leggibilità.

Nel Font otx3

Il segno	φ	(`\varphi` in TeX)	si ottiene con **[alt] f**
Il segno	␣␣	(`\:` in TeX)	si ottiene con **[alt] / [alt] /**
Il segno	→	(`\rightarrow` in TeX)	si ottiene con **[alt] 2**
Il segno	ℝ	(`{\bf R}` in TeX)	si ottiene con **[alt] R**
Il segno	␣	(`\quad` in TeX)	si ottiene con **[alt] "**
Il segno	√	(`\sqrt` in TeX)	si ottiene con **[alt] H**
Il segno	$	(`$` in TeX)	si ottiene con **$**

Nel Font otx3, il carattere $ è sostituito da , un carattere assai visibile che aiuta a non dimenticarlo e a individuare le formule matematiche a colpo d'occhio.

Ho anche modificato il layout di tastiera, in modo che il tasto [§] che genera un carattere di scarso uso dia invece l'usatissimo $.
Altri simboli usati sono per esempio:

α	(\alpha in TeX)	che si ottiene con **[alt] a**
λ	(\lambda in TeX)	che si ottiene con **[alt] l**
≠	(\neq in TeX)	che si ottiene con **[alt] =**
≥	(\geq in TeX)	che si ottiene con **[alt] >**
∞	(\infty in TeX)	che si ottiene con **[alt-shift] 8**
⊂	(\subset in TeX)	che si ottiene con **[alt] c**

Simboli speciali per keywords in uso comune soprattutto in LaTeX sono per esempio:

𝔹	(\begin in TeX)	che si ottiene con **[alt] B**
𝔼	(\end in TeX)	che si ottiene con **[alt] E**
𝔸	({array} in TeX)	che si ottiene con **[alt] A**
𝕋𝔸	({tabular} in TeX)	che si ottiene con **[alt] T [alt] A**
↵	(\\ in LaTeX)	che si ottiene con **[alt] '**
TAB	(& in TeX)	che si ottiene con **[TAB]**

In realtà non ci sarebbe bisogno di sostituire & con il TAB, ma questo facilita l'input e l'editing di arrays.

Per esempio si confronti come è possibile generare la matrice a lato direttamente in LaTeX e col mio font otx3 :

$$\begin{pmatrix} 3 & 2 & -\sqrt{\pi} \\ \alpha & 2 & \alpha+1 \\ 1 & 0 & 1-\alpha \end{pmatrix}$$

```
$\begin{array}{rrr}
3 & 2& -\sqrt{pi}\\
-\alpha &2 &\alpha+1\\
1 &0 &1-\alpha
\end{array}$
```

```
𝔹𝔸{rrr}
3       2       -√{π}↵
-α      2       α+1↵
1       0       1-α
𝔼𝔸
```

Leggibilità ed editabilità sono evidentemente superiori.

Ho definito per i miei usi più di 200 simboli, che richiedono, per essere memorizzati, meno sforzo di quello occorrente per memorizzare le corrispondenti keyword.

Altri esempi:
In LaTeX ci sono i comandi per gli spazi:
\, piccolo spazio \: spazio medio \; spazio pù largo
Vengono generati semplicemente con

[alt] / **[alt] / [alt] /** **[alt] /[alt] /[alt] /**

Se c'è da inserire una keyword di uso non frequente per cui non esiste simbolo, la si

inserisce semplicemente senza preoccupazione. Per esempio i comandi LaTeX.

```
\begin{document}
\setcounter{page}{1}
Se $\rho=\cos(\frac{\theta}{2})$ \`e una curva in $\real^2$.
```

si scrivono mediante il Font otx3:

```
𝔹𝔻
\setcounter{page}{1}
Se 𝔦ϱ=\cos(𝔽{θ}{2})𝔦 è una curva in 𝔦ℝ^2𝔦.
```

Il comando `\setcounter` non è tra quelli frequenti, quindi non ha simbolo nel font otx3 e viene inserito normalmente. Anche il comando `\cos` non è tra quelli previsti, ma in questo caso la leggibilità rimane buona

La conversione

Naturalmente per trasformare successivamente un file scritto in questo modo in qualcosa di leggibile dal TeX occorre un programma di traduzione.

La conversione va anche fatta in modo da lasciare uno spazio dopo ogni keyword che lo richieda.

Ho scritto nel 1991 un programma in Basic per fare questa conversione, con una buona interfaccia utente e l'ho utilizzato per 9 anni migliorandone via via le prestazioni.

Aveva alcuni pregi:

1. Una buona interfaccia utente che consentiva di operare velocemente con grande affidabilità.
2. Una conversione perfetta.

Aveva due difetti:

1. Conversione lenta: (era scritto con QuickBasic, un programma che risale al 1985) e poi occorreva effettuare un passaggio intermedio salvando il file in puro testo, anche se nelle ultime versioni era riuscito a evitarlo.
2. Non funzionava (o funzionava male) sui Mac PPC con molta RAM.

In seguito ho trovato un programma detto PowerReplace in grado di effettuare queste conversioni in modo assai rapido.

Aveva alcuni pregi:

1. Conversione velocissima (3 o 4 secondi per i documenti più lunghi).
2. Funzionava su ogni Mac (anche col sistema X).

Aveva qualche difetto:

1. L'interfaccia utente non era quella che avrei voluto ed era difficilmente adattabile, anche se ho fatto diverse macro per migliorarla.
2. L'affidabilità era inferiore; qualche volta la conversione non avveniva e occorreva ripeterla.

Nel 2004 ho scritto un programma in FutureBasic che effettua la conversione in un tempo assai breve (su iMac meno di 1/4 di secondo per un testo di 15 pagine fitte).

Pregi:

1. Molto affidabile, i crac sono rari.
2. E' veloce: su un iMac si possono tenere aperte contemporaneamente la finestra di

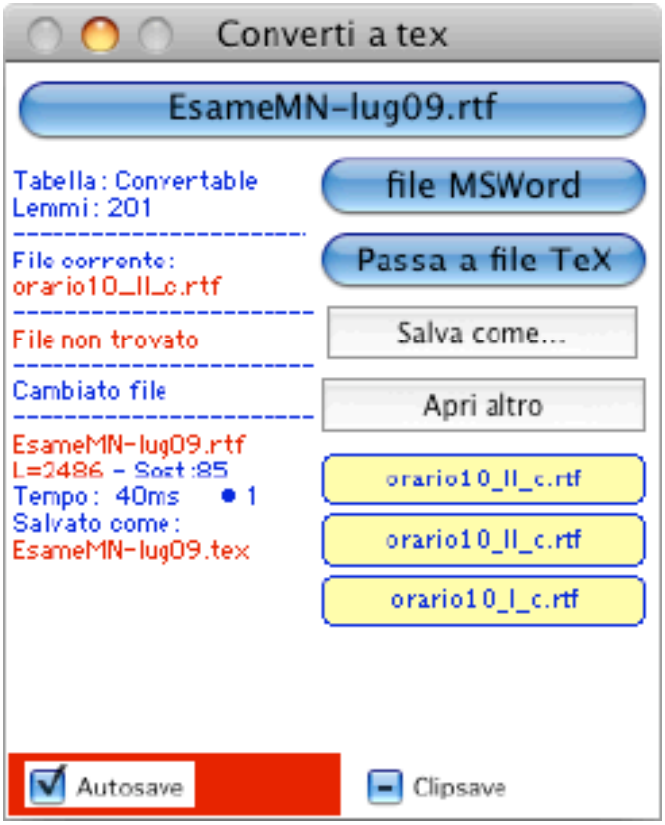
Word e quella del typeset e vedere le correzioni quasi in tempo reale.

Difetti:

1. La lunghezza del file è limitata a 64000 caratteri, comunque sufficiente per uno scritto di 15 pagine fitte.

Nel 2007 il programma è stato riscritto per il sistema MacOsX e ogni tanto viene migliorato

Questa è la finestra del programma.



Un tasto grande per convertire il file di testo corrente.

Un tasto per selezionare e convertire il file attualmente in primo piano in Word.

Un tasto per aprire il file convertito mediante TeXShop.

Un tasto per salvare.

Un tasto per aprire un altro file.

Tre tasti per accedere velocemente agli ultimi tre testi convertiti.

Un log con le operazioni effettuate.

Una checkbox che consente di attivare l'opzione di salvare automaticamente il file convertito.

Una checkbox che consente di attivare l'opzione di salvare una copia del file convertito nel Clipboard per eventuale trasferimento manuale.

Ho anche scritto un paio di macro per attivare il programma automaticamente da Word passando subito al Typeset.

Esiste anche la possibilità di fare la conversione opposta, da un file scritto in TeX a un file scritto col mio metodo. Questa conversione è un po' meno affidabile e più lenta, soprattutto perché usandola di rado non ho mai avuto occasione di migliorarla, ma in genere è buona e in poco tempo si possono correggere i pochi errori.

Il problema è comunque ancora quello di avere un editore di testo buono.

Per ora tutti gli editori di testo per sistema Mac OsX che ho provato non hanno (incredibile!) l'affidabilità e le caratteristiche del vecchio Word 5 che gira su System 9.2. Ultimamente uso MsWord 2007 che, pur molto meno pratico e affidabile del vecchio Word 5 è, tutto sommato il migliore.

Un esempio tratto liberamente dal mio libro:

Riporto, con alcuni adattamenti, un piccolissimo stralcio del mio libro di Algebra lineare in tre diverse visioni.

1. Il testo come lo si ha nel File sorgente TeX, colorato come fanno certi editori (Alpha, BBedit o TexShop o altri editori per Windows, ma non Textures).
2. Lo stesso testo scritto col mio metodo. Alcuni simboli sono colorati opportunamente (cosa non obbligatoria) per facilitare la navigazione.
3. L'output finale: il Typeset.

Confrontate i due testi e l'output.

Anche il testo sorgente TeX che è stato ottenuto con la conversione, è comunque molto ordinato per via del metodo di conversione che genera automaticamente gli "a capo" ed elimina spazi non necessari.

Col mio metodo e un clic si passa da uno all'altro.

Testo in LaTeX

```
\subsection{La decomposizione $A=MN^T$}
La seguente proposizione fornisce una decomposizione per una
matrice che \`e in qualche modo un analogo della decomposizione $QR$.
\begin {proposition}
Sia $A$ una matrice quadrata $n\times n$ di rango $r$. Allora
esistono due matrici $M,N\in M_{n,r}$ con $\varrho (M)=\varrho (N)=r$
tali che
\[ A=M\by N^T\]
\end {proposition}
\begin {proof}
Sia $\varphi :\real ^n\rightarrow \real ^n$ la trasformazione
lineare associata ad $A$ rispetto alla base canonica e poniamo
$V=\Im(\varphi )$. Possiamo fattorizzare $\varphi $ come $\varphi
=i\comp \psi $ dove $\psi $ \`e la trasformazione surgettiva
$\psi :\real ^n\longrightarrow V$ e $i$ \`e l'immersione
$i:V\longrightarrow
\real ^n$. Fissiamo una base di $V$ e scriviamo le matrici $M$
e $Q$ associate rispettivamente a $i$ e $\psi $ mediante la
base canonica di $\real ^n$ e la base scelta di $V$. Allora
$M$ e $Q$ hanno entrambe rango $r$ e vale la decomposizione
$A=M\by Q$. Ponendo $N=Q^T$ si ha la tesi.
\end {proof}
\begin {example}
\item Sia $A=\left (\begin {array}{rrr}
1& 0& 3\\
1& 1& 1\\
0& -1& 2
\end {array}\right )$
di rango 2. \\[2mm]
Scegliamo una base per lo spazio generato dalle colonne (cio\`e
$\Im(\varphi ))$.
\end {example}
```

Testo usando il font otx3

\subsection{La decomposizione $A=MN^T$}

La seguente proposizione fornisce una decomposizione per una matrice che è in qualche modo un analogo della decomposizione QR.

\begin{proposizione}

Sia A una matrice quadrata $n\times n$ di rango r. Allora esistono due matrici $M,N\in M_{n,r}$ con $\varrho(M)=\varrho(N)=r$ tali che

$$A=M\cdot N^T$$

\end{proposizione}

\begin{proof}

Sia $\varphi:\mathbb{R}^n\to\mathbb{R}^n$ la trasformazione lineare associata ad A rispetto alla base canonica e poniamo $V=\Im(\varphi)$. Possiamo fattorizzare φ come $\varphi=i\circ\psi$ dove ψ è la trasformazione surgettiva $\psi:\mathbb{R}^n\longrightarrow V$ e i è l'immersione $i:V\longrightarrow\mathbb{R}^n$. Fissiamo una base di V e scriviamo le matrici M e Q associate rispettivamente a i e ψ mediante la base canonica di $\mathbb{R}^n$ e la base scelta di V. Allora M e Q hanno entrambe rango r e vale la decomposizione $A=M\cdot Q$. Ponendo $N=Q^T$ si ha la tesi.

\end{proof}

\begin{esempio}

•Sia $A=\mathbb{L}(\mathbb{BA}\{rrr\}$

1 0 3↵

1 1 1↵

0 –1 2

$\mathbb{EA}\mathbb{J})$

di rango 2. ↵□□

Scegliamo una base per lo spazio generato dalle colonne (cioè $\Im(\varphi)$).

\end{esempio}

3.1 La decomposizione $A = MN^T$

La seguente proposizione fornisce una decomposizione per una matrice che è in qualche modo un analogo della decomposizione QR.

Proposizione 1 *Sia A una matrice quadrata $n \times n$ di rango r. Allora esistono due matrici $M, N \in M_{n,r}$ con $\varrho(M) = \varrho(N) = r$ tali che*

$$A = M \cdot N^T$$

Dimostrazione: Sia $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ la trasformazione lineare associata ad A rispetto alla base canonica e poniamo $V = \mathrm{Im}\,(\varphi)$. Possiamo fattorizzare φ come $\varphi = i \circ \psi$ dove ψ è la trasformazione surgettiva $\psi : \mathbb{R}^n \longrightarrow V$ e i è l'immersione $i : V \longrightarrow \mathbb{R}^n$. Fissiamo una base di V e scriviamo le matrici M e Q associate rispettivamente a i e ψ mediante la base canonica di $\mathbb{R}^n$ e la base scelta di V. Allora M e Q hanno entrambe rango r e vale la decomposizione $A = M \cdot Q$. Ponendo $N = Q^T$ si ha la tesi. □

Esempio 1.2 Sia $A = \begin{pmatrix} 1 & 0 & 3 \\ 1 & 1 & 1 \\ 0 & -1 & 2 \end{pmatrix}$ di rango 2.

Scegliamo una base per lo spazio generato dalle colonne (cioè $\mathrm{Im}\,(\varphi)$).