

Numero 19  
Aprile 2015

# Ars<sub>TeX</sub>nica

Rivista italiana di  $\text{TeX}$  e  $\text{L}^{\text{A}}\text{TeX}$

GU<sub>IT</sub>

<http://www.guitex.org/arstexnica>

# TeXnica Ars

G<sub>U</sub>T – Gruppo Utilizzatori Italiani di T<sub>E</sub>X

ArsT<sub>E</sub>Xnica è la pubblicazione ufficiale del G<sub>U</sub>T

## Comitato di Redazione

Claudio Beccari – *Direttore*

Roberto Giacomelli – *Comitato scientifico*

Enrico Gregorio – *Comitato scientifico*

Ivan Valbusa – *Comitato scientifico*

Lorena Rachele Badile, Renato Battistin,

Riccardo Campana, Massimo Caschili,

Gustavo Cevolani, Massimiliano Dominici,

Tommaso Gordini, Carlo Marmo,

Gianluca Pignalberi, Ottavio Rizzo,

Gianpaolo Ruocco, Enrico Spinielli,

Emiliano Vavassori

ArsT<sub>E</sub>Xnica è la prima rivista italiana dedicata a T<sub>E</sub>X, a L<sup>A</sup>T<sub>E</sub>X ed alla tipografia digitale. Lo scopo che la rivista si prefigge è quello di diventare uno dei principali canali italiani di diffusione di informazioni e conoscenze sul programma ideato quasi trent'anni fa da Donald Knuth.

Le uscite avranno, almeno inizialmente, cadenza semestrale e verranno pubblicate nei mesi di Aprile e Ottobre. In particolare, la seconda uscita dell'anno conterrà gli Atti del Convegno Annuale del G<sub>U</sub>T, che si tiene in quel periodo.

La rivista è aperta al contributo di tutti coloro che vogliono partecipare con un proprio articolo. Questo dovrà essere inviato alla redazione di ArsT<sub>E</sub>Xnica, per essere sottoposto alla valutazione di revisori. È necessario che gli autori utilizzino la classe di documento ufficiale della rivista; l'autore troverà raccomandazioni e istruzioni più dettagliate all'interno del file di esempio (.tex). Tutto il materiale è reperibile all'indirizzo web della rivista.

Gli articoli potranno trattare di qualsiasi argomento inerente al mondo di T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X e non dovranno necessariamente essere indirizzati ad un pubblico esperto. In particolare tutorials, rassegne e analisi comparate di pacchetti di uso comune, studi di applicazioni reali, saranno bene accetti, così come articoli riguardanti l'interazione con altre tecnologie correlate.

Di volta in volta verrà fissato, e reso pubblico sulla pagina web, un termine di scadenza per la presentazione degli articoli da pubblicare nel numero in preparazione della rivista. Tuttavia gli articoli potranno essere inviati in qualsiasi momento e troveranno collocazione, eventualmente, nei numeri seguenti.

Chiunque, poi, volesse collaborare con la rivista a qualsiasi titolo (recensore, revisore di bozze, grafico, etc.) può contattare la redazione all'indirizzo:

arstexnica@guitex.org.

## Nota sul Copyright

Il presente documento e il suo contenuto è distribuito con licenza  Creative Commons 2.0 di tipo "Non commerciale, non opere derivate". È possibile, riprodurre, distribuire, comunicare al pubblico, esporre al pubblico, rappresentare, eseguire o recitare il presente documento alle seguenti condizioni:

- ① **Attribuzione:** devi riconoscere il contributo dell'autore originario.
- Ⓢ **Non commerciale:** non puoi usare quest'opera per scopi commerciali.
- Ⓒ **Non opere derivate:** non puoi alterare, trasformare o sviluppare quest'opera.

In occasione di ogni atto di riutilizzazione o distribuzione, devi chiarire agli altri i termini della licenza di quest'opera; se ottieni il permesso dal titolare del diritto d'autore, è possibile rinunciare ad ognuna di queste condizioni.

Per maggiori informazioni:

<http://www.creativecommons.org>

## Associarsi a G<sub>U</sub>T

Fornire il tuo contributo a quest'iniziativa come membro, e non solo come semplice utente, è un presupposto fondamentale per aiutare la diffusione di T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X anche nel nostro paese. L'adesione al Gruppo prevede una quota di iscrizione annuale diversificata: 30,00 € soci ordinari, 20,00 (12,00) € studenti (junior), 75,00 € Enti e Istituzioni.

## Indirizzi

Gruppo Utilizzatori Italiani di T<sub>E</sub>X

c/o Università degli Studi di Napoli Federico II

Dipartimento di Ingegneria Industriale

Via Claudio 21

80125 Napoli – Italia

<http://www.guitex.org>

[guit@sssup.it](mailto:guit@sssup.it)

Redazione ArsT<sub>E</sub>Xnica:

<http://www.guitex.org/arstexnica/>

[arstexnica@guitex.org](mailto:arstexnica@guitex.org)

Codice ISSN 1828-2369

Stampata in Italia

Napoli: 15 Aprile 2015

# ArsT<sub>E</sub>Xnica

Rivista italiana di T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X

*Numero 19, Aprile 2015*

Claudio Beccari	
Editoriale . . . . .	5
Frank Mittelbach	
Come si può influenzare la posizione degli ambienti galleggianti come figure e table in L <sup>A</sup> T <sub>E</sub> X? . . . . .	6
Barbara Beeton	
Una figura larga sotto un testo a due colonne . . . . .	14
Claudio Beccari	
<code>\vadjust</code> , questo sconosciuto . . . . .	15
James R. Hunt	
Comporre liste: un viaggio nella grammatica sconosciuta . . . . .	19
Claudio Fiandrino	
Personalizzazione e riproducibilità nel disegno programmato: introduzione al pacchetto <code>pgfkeys</code> . . . . .	27
Gianluca Pignalberi	
L <sup>A</sup> T <sub>E</sub> X in un'editrice universitaria: come e perché <code>book</code> non è adatta e una possibile alternativa funzionante . . . . .	33

# Come si può influenzare la posizione degli ambienti galleggianti come `figure` e `table` in $\LaTeX$ ?\*

Frank Mittelbach

## Sommario

Nel 2012, su  $\TeX$ - $\LaTeX$  Stack Exchange venne posta la domanda «How to influence the position of float environments like `figure` and `table` in  $\LaTeX$ ?» (DANIEL, 2012), e siccome in precedenza erano comparse numerose altre domande su quest'argomento, ho deciso di trattarlo in profondità e di spiegare la maggior parte dei misteri che il meccanismo sottostante presenta a chi cerca di usarlo con successo.

Da quando la mia risposta è apparsa sul web, si è chiesto di vederla trasformata in un articolo e io risposi stupidamente «solo se questa risposta finisce per diventare una 'grande' risposta» (cioè se ottiene 100 voti). Al momento di scrivere questo articolo la risposta ha collezionato 222 voti, perciò sarà meglio mantenere la promessa.

## Abstract

In 2012, a question “How to influence the float placement in  $\LaTeX$ ” was asked on  $\TeX$ .stackexchange and as there had been many earlier questions around this topic I decided to treat the topic in some depth and explain most of the mysteries that the underlying mechanism poses to people trying to use it successfully.

Once my answer appeared on the web, people asked to see this converted into an article and I foolishly replied “only if this answer ends up becoming a ‘great’ answer” (gets 100 votes). At the time of writing this article, the answer stands at 222 votes, so I had better make good on that promise.

## Nota dei traduttori

Nella traduzione si è deciso di lasciare il termine *float* in inglese per la mancanza di un corrispettivo traduttore italiano sintetico. In pratica, con *float* si intende uno dei due ambienti mobili standard di

\*Questo articolo è una traduzione autorizzata dall'autore della risposta a (DANIEL, 2012) (pubblicata originariamente su  $\TeX$ - $\LaTeX$  Stack Exchange il 21 dicembre 2011) eseguita da Claudio Beccari e Tommaso Gordini, unici responsabili di eventuali travisamenti del testo originale. Frank Mittelbach vive in Germania e ha studiato matematica e informatica. Si occupa di  $\LaTeX$  dalla metà degli anni '80 del Novecento ed è autore e coautore di numerosi pacchetti, manuali e pubblicazioni su questo programma. Attualmente è direttore tecnico del  $\LaTeX$ 3 Project e supervisore di  $\LaTeX$ 2 $\epsilon$ .

$\LaTeX$  (`figure` o `table`) o gli ambienti mobili definiti da certe classi di documento o certi pacchetti di estensione. Con *oggetto* intendiamo il contenuto di un float: un'immagine, una tabella, un disegno, un grafico, eccetera. Per *composizione a due colonne* si intende il risultato dell'opzione di classe `twocolumn` o del comando `\twocolumn` e non quello possibile grazie al pacchetto `multicol`. Con l'espressione *colonna di testo* si intende l'area della pagina corrispondente a `\columnwidth` (che nella composizione a una sola colonna coincide con `\textwidth`). Con *area di float* si intende lo spazio della pagina destinato a ospitare i float. Infine, con *float a giustezza piena* si intende un oggetto largo quanto due colonne nella composizione a due colonne.

## 1 Introduzione

Per rispondere a questa domanda, bisogna prima comprendere le regole fondamentali che governano la collocazione standard dei float sulla pagina in  $\LaTeX$ . Capite queste, si possono operare aggiustamenti, per esempio cambiando i parametri dei float o caricando certi pacchetti che modificano o estendono le funzionalità di base del programma.

## 2 Terminologia

### 2.1 Categorie di float

In  $\LaTeX$ , ogni float appartiene a una categoria. Per impostazione predefinita  $\LaTeX$  ne conosce due, corrispondenti agli ambienti mobili standard `figure` e `table`. Non sono le uniche: altre classi di documento e altri pacchetti possono aggiungerne di nuove. La categoria di appartenenza influenza alcuni aspetti della collocazione del float sulla pagina, come la sua specificazione di collocazione predefinita (a meno che non venga sovrascritta dal float stesso nei modi descritti nella sezione 2.3).

Un'importante proprietà dell'algoritmo di collocazione dei float<sup>1</sup> è che  $\LaTeX$  non viola mai l'ordine della disposizione all'interno di una categoria. Per esempio, se in un documento si ha la serie ‘Figura 1, Tabella 1, Figura 2’, la ‘Figura 1’ verrà posata sempre prima della ‘Figura 2’. Tuttavia, la ‘Tabella 1’ (appartenendo a un'altra categoria) sarà gestita indipendentemente e quindi nel documento finito potrà comparire prima, dopo o tra le due figure.

1. D'ora in poi *algoritmo*. (N.d.T.)

## 2.2 Aree di float

In una colonna di testo, L $\TeX$  conosce due aree in cui può mettere i float: l'area superiore e l'area inferiore. Nella composizione a due colonne, conosce anche un'area superiore che si estende su entrambe le colonne per float larghi l'intera pagina, mentre non esiste una corrispondente area inferiore.

Inoltre, L $\TeX$  è in grado di produrre colonne e pagine che contengono solo float.<sup>2</sup>

Infine, L $\TeX$  è capace di mettere i float a metà pagina intercalato nel testo (ma solo se di volta in volta si guida la posa di ogni singolo float in questo senso).

## 2.3 Codici di posizione

Per indirizzare la posa di un float in una di queste aree, gli si può fornire un *codice di posizione*<sup>3</sup> come argomento facoltativo. Se tale indicazione non viene espressa, L $\TeX$  userà un codice predefinito (che, come s'è detto sopra, dipende dalla categoria di appartenenza ma di solito permette all'oggetto di andare in tutte le aree a meno di altre restrizioni).

Un codice può consistere dei seguenti caratteri in un ordine *qualsiasi*.

- ! Indica che alcune delle restrizioni normalmente attive dovrebbero essere ignorate (se ne discuterà più avanti).
- h Indica che il float ha il permesso di essere messo intercalato con il testo (*here*, 'qui').
- t Indica che il float ha il permesso di andare nell'area superiore (*top*, 'in alto').
- b Indica che il float ha il permesso di andare nell'area inferiore (*bottom*, 'in basso').
- p Indica che il float ha il permesso di andare in una pagina *p* o nell'area occupata da una colonna (*page*, 'pagina').

L'ordine in cui si scrivono questi caratteri nell'argomento facoltativo *non* influenza il modo in cui l'algoritmo cerca di posare il float! L'esatta sequenza osservata dall'algoritmo verrà discussa nella sezione 3.2. Questo è uno degli equivoci più comuni che si verifica, per esempio, quando gli utenti pensano che [bt] chieda a L $\TeX$  di provare per prima l'area inferiore.

Tuttavia, se una delle lettere non è presente, l'area corrispondente non verrà nemmeno provata.

## 2.4 Parametri dell'algoritmo

La posa di un float sulla pagina è influenzata da circa 20 parametri. Fondamentalmente, essi definiscono quanto segue.

1. Quanti float possono andare in una certa area.

2. D'ora in poi *pagina* o *colonna* *p*. (*N.d.T.*)

3. D'ora in poi *codice*. Nella sola sezione 5 *codice* è usato nell'accezione di *codice sorgente*. (*N.d.T.*)

2. Quanto grande un'area di float può divenire.

3. Quanto testo deve esserci in una pagina (cioè, quanta parte della pagina le aree di float superiore e inferiore possono occupare).

4. Quanto spazio verrà inserito:
  - tra float consecutivi in una stessa area e
  - tra l'area di float e il testo fra cui è intercalato.

## 2.5 Punto di riferimento di un float

Un luogo del documento che contiene il riferimento a un float (per esempio, «vedi la figura X») è detto *chiamata*. Il corpo del float dovrebbe essere messo vicino alla propria (prima) chiamata: infatti, la posizione dell'ambiente mobile nel documento sorgente influenza quella dell'oggetto nel documento finito, poiché determina il momento in cui L $\TeX$  lo vede per la prima volta.

È importante capire che se un float viene messo in mezzo a un capoverso, per l'algoritmo il punto di riferimento diventa la successiva interruzione di riga o di pagina nel capoverso che segue l'effettiva sede del float nel sorgente.

Per ragioni tecniche e pratiche, di solito è meglio mettere tutti i float tra capoversi (per esempio, dopo il capoverso che ne contiene la chiamata), anche se la cosa allontana leggermente chiamata e punto di riferimento.

## 3 Comportamento fondamentale del meccanismo flottante

Sulla scorta di queste conoscenze, ora si può scavare a fondo nel comportamento dell'algoritmo.

Per prima cosa, bisogna capire che tutti gli algoritmi di composizione tipografica di L $\TeX$  sono progettati per evitare qualsiasi tipo di ripensamento. Ciò significa che L $\TeX$  legge da cima a fondo il documento sorgente, formatta ciò che vi trova e lo compone (più o meno) immediatamente. I programmatori scelsero questa condotta per limitare la complessità (che rimane ancora piuttosto elevata) e mantenere ragionevole la velocità di composizione (si ricordi che il programma risale ai primi anni '80 del Novecento quando anche i main frame erano lentissimi a confronto con i calcolatori moderni).

Per quanto riguarda i float, in altre parole l'algoritmo è 'avido': nel momento in cui incontra un ambiente mobile, L $\TeX$  proverà immediatamente a posarlo e, se ci riesce, non ritornerà mai più su questa decisione. Dunque, questo significa anche che la soluzione adottata potrebbe rivelarsi infelice alla luce dei dati ricevuti successivamente.

Per esempio, se una figura ha il permesso di andare nell'area superiore o inferiore, L $\TeX$  può decidere di metterla nella prima. Se a questa figura seguono due tabelle che possono andare solo in

cima alla pagina, queste tabelle potrebbero non starci più. Una soluzione probabilmente efficace in questo caso (ma che l'algoritmo non ha esaminato) sarebbe stata quella di mettere la figura in basso e le due tabelle in alto.

### 3.1 Sequenza di base

Ecco la sequenza di base eseguita dall'algoritmo.

1. Quando incontra un float, L<sup>A</sup>T<sub>E</sub>X prova a posarlo immediatamente secondo le proprie regole (descritte nel dettaglio più avanti).
2. Se ci riesce, lo posa e non cambierà più questa decisione.
3. Se non ci riesce, lo mette in una *coda di attesa*<sup>4</sup> per riconsiderarlo quando comincerà a comporre la pagina successiva (ma non prima).
4. Una volta terminata una pagina, L<sup>A</sup>T<sub>E</sub>X esamina la coda di attesa e prova a svuotarla meglio che può, innanzitutto cercando di generare il maggior numero possibile di pagine *p* (nella speranza di svuotare la coda). Esaurita anche questa possibilità, tenterà di posare i float rimanenti in aree superiori e inferiori e, se non ci riesce, li differirà in una pagina successiva (cioè aggiungendoli ancora una volta alla coda di attesa).
5. Dopo di che, comincia a elaborare il materiale per questa nuova pagina. Durante il processo, può incontrare ulteriori float.
6. Se ha raggiunto la fine del documento o ha incontrato un comando `\clearpage`, L<sup>A</sup>T<sub>E</sub>X comincia una nuova pagina, allenta tutte le restrizioni gravanti sui float e svuota la coda sistemandone il contenuto in una o più pagine *p*.

Nella modalità a due colonne, il programma usa lo stesso algoritmo, con la differenza che lavora a livello di colonne: quando ha terminato una colonna, cioè, L<sup>A</sup>T<sub>E</sub>X esaminerà la coda e genererà colonne *p*, eccetera.

### 3.2 Regole di collocazione dettagliate

Ogni volta che L<sup>A</sup>T<sub>E</sub>X incontra un ambiente galleggiante nel sorgente, prima di tutto esaminerà la coda di attesa per controllare se ne contiene già un altro della stessa categoria. Se lo trova, il float appena incontrato non ha il permesso di essere posato e va immediatamente in coda.

In caso contrario, L<sup>A</sup>T<sub>E</sub>X ne esamina il codice, sia quello esplicito eventualmente dichiarato nell'argomento facoltativo, sia quello predefinito per la categoria di float in questione. Quest'ultimo è impostato nel file di classe (per esempio, `article.cls`) e molto spesso, ma non sempre, si risolve in `tbp`.

4. D'ora in poi anche semplicemente *coda*. (*N.d.T.*)

1. Se il codice contiene un `!`, l'algoritmo ignorerà qualunque restrizione connessa al numero di float che può andare in un'area o allo spazio massimo occupabile da quest'ultima sulla pagina. In caso contrario, le restrizioni imposte dai parametri restano attive.
2. Il passo successivo è controllare se tra i codici c'è `h`.
3. Se c'è, l'algoritmo tenterà di posare il float esattamente dove l'ha incontrato. Se c'è abbastanza spazio da permetterlo, l'elaborazione di quest'oggetto termina qui.
4. Se non c'è, subito dopo verificherà la presenza di `t`. Se lo trova, l'algoritmo proverà a posare il float nell'area superiore. In assenza di altre restrizioni che lo impediscano, il float viene messo lì e la sua elaborazione termina.
5. Se nemmeno `t` è tra i codici, L<sup>A</sup>T<sub>E</sub>X va alla ricerca di `b e`, se lo trova, proverà a posare il float nell'area inferiore (sempre obbedendo a tutte le restrizioni attive in assenza di `!`).
6. Se la cosa non funziona o non è permessa perché il codice non è stato indicato, il float viene aggiunto alla coda di attesa.
7. Durante il processo appena descritto, un codice `p` (se presente) non viene considerato. Verrà esaminato solo quando la coda viene svuotata al successivo confine di pagina o di colonna.

Questo termina il procedimento per posare un float quando L<sup>A</sup>T<sub>E</sub>X ne trova uno nel documento.

### 3.3 Svuotamento della coda al confine di pagina o di colonna

Dopo aver terminato una pagina o una colonna, L<sup>A</sup>T<sub>E</sub>X esamina la coda di attesa e cerca di svuotarla nel miglior modo possibile. Per farlo, prima di tutto proverà a creare pagine *p*.<sup>5</sup>

Ogni float che partecipa a una pagina (o colonna) *p* deve avere *p* tra i propri codici. Se non ce l'ha, non solo non può finire in una pagina di questo tipo, ma impedisce di finirci anche a ogni ulteriore float della stessa categoria che lo segue nella coda di attesa!

Se il float ha il permesso di andarci, verrà contrassegnato per esservi incluso, ma se la pagina non verrà riempita 'abbastanza' (secondo le impostazioni dei parametri che governano le pagine *p*) il programma può ancora interrompere il tentativo. Queste restrizioni vengono abolite solo alla fine del documento o quando è stato emesso un `\clearpage`: allora, un float verrà messo in una

5. Nella modalità a due colonne, L<sup>A</sup>T<sub>E</sub>X creerà colonne *p* (quando ha terminato una colonna) e cercherà anche di generare pagine *p* quando ha terminato una pagina. Nel resto dell'articolo, *pagina p* indicherà l'una o l'altra a seconda del contesto.

pagina `p` anche se non ha `p` tra i codici e risultasse essere l'unico di quella pagina.

La creazione di pagine `p` continua finché l'algoritmo non ha più float da posare o finché non fallisce nel produrle a causa delle impostazioni dei parametri. In questo secondo caso, tutti i float ancora in attesa di sistemazione vengono considerati per l'inclusione nelle aree superiore o inferiore della successiva pagina (o colonna).

Il processo è lo stesso di quello appena descritto, tranne che per i due punti seguenti.

1. Il codice `h` non ha più alcun significato (dato che ora si è molto lontani dall'originario *here*, 'qui') e perciò viene ignorato.
2. Questa volta i float non provengono dal documento sorgente, ma sono prelevati uno dopo l'altro dalla coda di attesa.

Tutti i float che non possono essere sistemati vengono rimessi in coda, in modo che quando  $\LaTeX$  è pronto per esaminare l'ulteriore input testuale dal sorgente, la coda può già contenerne. Una conseguenza di questo comportamento è che un float incontrato nel sorgente può essere immediatamente accodato solo perché un precedente float della stessa categoria è ancora in attesa di sistemazione.

### 3.4 Parametri che influenzano la posa

Quattro contatori controllano quanti float possono andare nelle varie aree di float.

`totalnumber` (valore predefinito: 3) Specifica il numero massimo di float permesso in una colonna di testo indipendentemente dalla loro posizione rispetto al testo. Non viene usato per le pagine `p`.

`topnumber` (valore predefinito: 2) Specifica il numero massimo di float permesso nell'area superiore della colonna di testo.

`bottomnumber` (valore predefinito: 1) Specifica il numero massimo di float permesso nell'area inferiore della colonna di testo.

`dbltopnumber` (valore predefinito: 2) Specifica il numero massimo di float a piena pagina permesso sopra le colonne di testo nella composizione a due colonne.

Le dimensioni delle aree sono controllate tramite parametri (modificabili con `\renewcommand`) che definiscono la massima (o minima) dimensione dell'area, espressa come una frazione dell'altezza della pagina. Di seguito se ne dà l'elenco.

`\topfraction` (valore predefinito: 0,7) Specifica l'altezza massima della frazione di pagina destinata ai float in testa a una pagina, se questa contiene anche del testo.

`\bottomfraction` (valore predefinito: 0,3) Specifica l'altezza massima della frazione di pagina destinata ai float in calce a una pagina, se questa contiene anche del testo.

`\dbltopfraction` (valore predefinito: 0,7) Specifica l'altezza massima della frazione di pagina a giustezza piena da destinare ai float a giustezza piena in testa alla pagina nella composizione a due colonne.

`\textfraction` (valore predefinito: 0,2) Specifica l'altezza minima della frazione di pagina destinata al testo in una pagina che contenga anche del testo (cioè, l'altezza minima dell'area che non deve essere occupata da float).

La costruzione di pagine o colonne `p`, invece, è controllata tramite i seguenti due parametri.

`\floatpagefraction` (valore predefinito: 0,5) Specifica la minima frazione della pagina (o della colonna) che deve essere occupata da float per fare di quella pagina (o di quella colonna) una pagina (o una colonna) `p`.

`\dblfloatpagefraction` (valore predefinito: 0,5) Specifica la minima frazione di una pagina `p` da occupare con float a giustezza piena nella composizione a due colonne. (*Aggiunta d.T.*)

Lo spazio che separa i float all'interno di un'area, come pure quello che c'è tra aree di float e aree riservate al testo, è definito dai seguenti parametri (che sono tutti lunghezze elastiche, cioè possono contenere alcuni componenti di allungamento o di accorciamento). I loro valori predefiniti dipendono dal corpo del font del documento e variano al variare delle relative opzioni di classe come `11pt` o `12pt`. Di seguito mostriamo solo i valori predefiniti per l'opzione `10pt`.

`\floatsep` (valore predefinito: `12pt plus 2pt minus 2pt`) Specifica la distanza minima tra due float consecutivi nell'area di float superiore o inferiore.

`\dblfloatsep` (valore predefinito: `12pt plus 2pt minus 2pt`) Specifica la distanza minima tra due float larghi l'intera pagina consecutivi in testa a una pagina composta a due colonne.

`\textfloatsep` (valore predefinito: `20pt plus 2pt minus 2pt`) Specifica la distanza minima da interporre tra l'area di float superiore o inferiore e quella riservata al testo.

`\intertextsep` (valore predefinito: `12pt plus 2pt minus 2pt`) Specifica la distanza minima fra un float intercalato con il testo e il testo precedente e seguente. (*Aggiunta d.T.*)

`\dbltextfloatsep` (valore predefinito: `20pt plus 2pt minus 2pt`) È l'analogo di `\textfloatsep` per float a giustezza piena.

## 4 Conseguenze dell'algoritmo

### 4.1 Nel documento finito, un float può comparire prima di dove si trova nel sorgente

Nonostante che la posizione dell'ambiente mobile nel sorgente determini il punto a partire dal quale l'oggetto compare nel documento finale, il float può ancora indietreggiare di una certa misura, poiché gli è permesso andare nell'area superiore della pagina corrente (nella sezione 6.3 si spiega come cambiare questo comportamento). Tuttavia, non può finire in una pagina precedente il testo circostante, dato che L<sup>A</sup>T<sub>E</sub>X non riconsidera le pagine precedenti che sono già state composte.

È per questo motivo che di solito nel sorgente si mette un float vicino alla sua prima chiamata (cioè un testo come «si veda la figura 5»), poiché la cosa garantirà di vedere l'oggetto o nella stessa pagina della chiamata o in una pagina successiva. In alcune situazioni, tuttavia, si potrebbe voler mettere un float nella pagina precedente la chiamata (in modo che oggetto e chiamata si trovino in due pagine affiancate nel documento finale). Lo si può fare, ma solo spostando indietro a mano il float nel sorgente.

### 4.2 I float a giustezza piena vengono sempre differiti in prima istanza

Quando incontra un ambiente mobile largo tutta la pagina (indicato con un \* alla fine del nome dell'ambiente, come per esempio `figure*`) nella composizione a due colonne, L<sup>A</sup>T<sub>E</sub>X sposta immediatamente quel float nella coda di attesa. La ragione di questo comportamento risiede ancora una volta nell'avidità dell'algoritmo: se L<sup>A</sup>T<sub>E</sub>X al momento sta assemblando la seconda colonna di quella pagina, significa che la prima colonna è già stata composta ed è conservata lontano. Si ricordi che non c'è modo di far stare il float nella pagina in questione, dato che L<sup>A</sup>T<sub>E</sub>X non ritorna sui propri passi. Per mantenere l'algoritmo semplice, il programma si comporta nello stesso modo anche se sta lavorando alla prima colonna (dove in teoria potrebbe fare di meglio anche senza ripensamenti).

Perciò, per mettere un float di questo tipo nella pagina corrente, bisogna spostarlo a mano nel sorgente prima dell'inizio della pagina. Così facendo, naturalmente, ogni successivo cambiamento nel testo potrebbe vanificare questo accorgimento. Di conseguenza, tali aggiustamenti funzionano meglio (se non benissimo) solo nell'ultimissima fase della produzione del documento – quando, cioè, tutto il materiale è stato scritto e l'attenzione è concentrata sulla messa a punto fine dell'aspetto del proprio lavoro.

Si noti anche che l'algoritmo di base presenta un bug<sup>6</sup> in quest'area di float: esso mantiene due

6. Dato che questo è il comportamento documentato nel manuale di L<sup>A</sup>T<sub>E</sub>X, forse è più corretto chiamarlo *caratteristica indesiderata* anziché *bug*.

code di stampa indipendenti, una per i float larghi una sola colonna e una per quelli larghi due. Di conseguenza, l'ordine con cui i float si susseguono nel sorgente non viene necessariamente rispettato nel documento finale: lì si potrebbe vedere in una sequenza inaspettata. Se ciò dovesse accadere, bisogna spostare a mano l'oggetto a giustezza piena prima (o dopo) la sua sede originaria oppure caricare il pacchetto `fixltx2e`, che contiene una correzione per questo problema.

### 4.3 Non esiste un'area di float inferiore per i float a giustezza piena

Questa non è tanto una conseguenza dell'algoritmo quanto una caratteristica della sua implementazione. Le uniche posizioni in cui i float a giustezza piena possono essere posati sono l'area superiore o una pagina p. Come caso speciale di una certa rilevanza, si noti che il float contenuto in un ambiente del tipo `{figure*}[b]` verrà posato solo dopo un `\clearpage` o alla fine del documento.

### 4.4 Tutti i parametri (normalmente) restringono le possibilità di posa

Potrebbe essere ovvio, ma vale la pena di ripeterlo: ciascun parametro relativo ai float diminuisce le possibilità di L<sup>A</sup>T<sub>E</sub>X di posare gli oggetti sulle pagine. L'entità di questa restrizione dipende da come il parametro è stato impostato: c'è sempre un modo per farlo che non influenza affatto la posa dell'oggetto. Sfortunatamente, nell'agire in questo modo si incoraggiano collocazioni piuttosto brutte.

Le impostazioni predefinite di L<sup>A</sup>T<sub>E</sub>X sono abbastanza permissive. Per esempio, perché una pagina p venga accettata, il float (o i float) deve occupare almeno la metà. Il che significa, in altre parole, che una pagina di questo tipo può rimanere mezza vuota (e questa, nella maggior parte dei casi, non è certo la migliore disposizione possibile).

Spesso gli utenti cercano di migliorare queste impostazioni e rimangono sorpresi quando all'improvviso tutti i float si accumulano alla fine del documento. Per restare su quest'esempio: se si porta il parametro `\floatpagefraction` a 0,8, diciamo, un float che occupi circa lo 0,75 della pagina non potrà formare da solo una pagina p. Così, se non c'è un altro float da affiancargli e che ci stia effettivamente nello spazio rimanente, verrà differito e con lui tutti gli altri float della stessa categoria. Ma, peggio ancora, questo float in particolare è anche troppo grande per andare nella successiva area di float superiore, che può occupare al massimo lo 0,7 della pagina (valore predefinito di `\topfraction`). Di conseguenza, tutti i float rimangono differiti fino al successivo `\clearpage`.

Per questa ragione, durante la stesura del documento è meglio non toccare i parametri, o almeno non farlo in modo da rendere più difficile all'algoritmo la posa di un float vicino alla sua chiamata. Per la correzione delle bozze è molto più importante

avere una figura vicino alla chiamata che evitare pagine mezze vuote. Le possibilità per mettere a punto finemente un documento il cui testo non necessita di modifiche vengono discusse più avanti.

Un'altra conclusione da tirare qui è che alcuni parametri dipendono da altri parametri: è importante tenere in considerazione queste dipendenze quando se ne modificano i valori.

#### 4.5 ‘Qui’ significa solo ‘qui se ci sta’

... e spesso non ci sta. La cosa è piuttosto sorprendente per molte persone, ma per il modo in cui l'algoritmo è stato progettato, il codice `h` (specificato da solo) non è un comando incondizionato. Se servisse un comando di questo tipo, si può ricorrere a pacchetti di estensione come `float`, che offre il codice alternativo `H`, che significa davvero ‘qui’ (e, se necessario, comincia una nuova pagina).

#### 4.6 I codici non definiscono un ordine di preferenza

Come s'è detto sopra, l'algoritmo prova a posare i float nelle aree di float disponibili in un ordine ben preciso nativo dell'algoritmo: ‘qui’, ‘in alto’, ‘in basso’ e – ai confini di pagina – prima ‘in una pagina `p`’ e, solo se la cosa non è più possibile, ‘in alto’ seguito da ‘in basso’ per la pagina successiva.

Così, specificare `[bt]` non significa ‘prova a posare questo float prima in basso e solo dopo in alto’, ma semplicemente ‘permetti che questo float finisca nell'area superiore o inferiore (ma non in una pagina `p`)’ esattamente come farebbe `[tb]`.

#### 4.7 Relazione tra float e note al piede

Questa non è esattamente una conseguenza dell'algoritmo ma una delle sue implementazioni: ogni volta che  $\LaTeX$  prova a decidere la posizione di un float (o di una nota al margine con `\marginpar!`) deve attivare la routine d'uscita per farlo. E come parte di questa routine, tutte le note al piede presenti sulla pagina vengono rimosse dalla loro posizione corrente nella bozza e riunite nella scatola `\footins`. Si tratta di un passaggio obbligato con cui  $\TeX$  si prepara a produrre la pagina.

Ma dopo aver posato il float (o dopo averlo differito),  $\LaTeX$  restituisce il materiale della pagina alla bozza, che ora è cambiata: le note, infatti, sono state rimosse dalle loro posizioni iniziali.  $\LaTeX$ , così, deve spostare le note, ma non può fare altro che metterle in uno stesso posto (non conoscendone più le sedi originali). Ciò che fa è reinserire le note (il loro testo, per la precisione) alla fine della bozza. Esistono alcune buone ragioni per fare così, una delle quali è che  $\LaTeX$  si aspetta che tutto il materiale restituitogli riesca a stare ancora nella pagina corrente.

Tuttavia, se per qualche ragione alla fine il programma rileva un'interruzione di pagina in un punto precedente, le note al piede compariranno nella pagina o nella colonna sbagliata. Si tratta di uno

scenario abbastanza improbabile e  $\LaTeX$  lavora duramente per renderlo quasi impossibile, ma se succedesse, si controlli se c'è un float vicino all'interruzione di pagina scelta e o lo si sposti o si guidi l'algoritmo usando interruzioni di pagina esplicite. Si può trovare un esempio di questo comportamento in un'altra domanda posta su  $\TeX$ - $\LaTeX$  Stack Exchange (HERMANN, 2012). Vale la pena di sottolineare il caso particolare discusso nella domanda appena segnalata: *non* si metta un float immediatamente dopo un titolo, a meno che non sia un titolo che comincia sempre una pagina. La ragione è che di solito i titoli formano oggetti molto grandi (dato che impediscono un'interruzione di pagina immediatamente dopo di sé). Tuttavia, mettere un float in questa posizione comporta l'attivazione della routine di uscita prima che  $\LaTeX$  abbia deciso dove interrompere la pagina, con il conseguente spostamento di eventuali note al piede nel posto sbagliato.

## 5 Documentazione dell'algoritmo

Come richiesto, ecco alcune informazioni sulla documentazione esistente. Algoritmo e relativa implementazione sono documentati nel file `loutput.dtx` come parte del sorgente del kernel di  $\LaTeX$ . Questo file può essere composto da solo o come parte dell'intero kernel (cioè componendo il file `source2e.tex` – si ignori il checksum error se è ancora lì, mi dispiace<sup>7</sup>).

Questa documentazione è un interessante manufatto storico. Parte di essa mostra pseudocodice semiformattato risalente a  $\LaTeX$  2.09; in altre parole, proviene dalla documentazione originale di Leslie Lamport. Il codice vero e proprio è documentato usando lo stile `doc` e in parte è più o meno documentato (da zero) e risale al 1994 o giù di lì, quando Chris Rowley e io abbiamo adeguato ed esteso l'algoritmo originale per  $\LaTeX$  2 $\epsilon$  (la versione corrente). Inoltre, documenta abbastanza apertamente i vari problemi con l'algoritmo e/o la sua implementazione – in molti casi non abbiamo avuto il coraggio di cambiarla per via delle numerose dipendenze e, naturalmente, per il pericolo di danneggiare troppi documenti esistenti che nel bene e nel male si affidavano implicitamente al comportamento corrente.<sup>8</sup> Verso la fine del codice si troverà un elenco di commenti sull'algoritmo compilati allora, ma ci sono anche commenti, domande e cose rimaste da fare (?-) disseminati lungo tutta la documentazione del codice.

Un aspetto interessante di questo file (che ho dimenticato del tutto) è che esso contiene tutto il codice necessario per tracciare il comportamen-

7. Ma ciò significa anche che si sta facendo girare una precedente versione di  $\LaTeX$ .

8. Questa è la ragione, per esempio, per cui la correzione del problema discusso nella sezione 4.2 è stata inserita nel pacchetto `fixltx2e` e non fa parte dell'algoritmo del kernel.

to dell'algoritmo nella vita reale. È un prodotto piuttosto grezzo e dettagliato e probabilmente per questa ragione non l'ho reso pubblicamente disponibile allora. Ma anche nella sua forma attuale permette di gettare qualche sguardo sul comportamento dell'algoritmo e su come avvengono certe decisioni.

Così, mentre scrivevo questo articolo ho avuto dei ripensamenti e ora la più recente distribuzione di L<sup>A</sup>T<sub>E</sub>X (maggio 2014)<sup>9</sup> offre il pacchetto `fltrace` che si può caricare per tracciare alcune strane decisioni sulla posa dei float o semplicemente per capire un po' meglio l'algoritmo. Esso offre i comandi `\tracefloats` e `\tracefloatsoff` per avviare o interrompere la tracciatura dell'algoritmo e `\tracefloatvals` per mostrare i valori correnti dei vari parametri discussi in questo articolo.

Poiché il pacchetto è identico al codice del kernel tranne che per l'aggiunta della tracciatura, potrebbe non funzionare correttamente caricando qualunque altro pacchetto che manipola quella stessa parte del codice del kernel. Se ciò dovesse accadere, la soluzione migliore è caricare `fltrace` per primo.

## 6 Affrontare problemi specifici

In questa sezione finale discuteremo alcune strategie per aggirare o risolvere problemi comuni. Non è in nessun modo completa e ulteriori informazioni si possono trovare in altre pubblicazioni, per esempio (GOOSSENS *et al.*, 2004) che dedica ai float un intero capitolo.

### 6.1 Assicurarsi che i float compaiano 'qui'

Talvolta bisogna garantire che i float compaiano in linea in determinati punti del documento, anche se ciò si traduce in pagine parzialmente vuote. Come s'è discusso sopra, il codice `h` non fornisce questa funzionalità, presente, però, in certi pacchetti di estensione: `float`, per esempio, definisce a questo scopo un codice `H`.

Una possibile alternativa è costituita dal comando `\captionof` definito dal pacchetto `caption`, che genera una normale didascalia (comprensiva della relativa voce nella lista delle figure o delle tabelle, eccetera) ma senza includere per forza l'oggetto in un float.

### 6.2 Area di float inferiore per float a giustezza piena

Come s'è visto nelle sezioni precedenti, l'algoritmo standard non supporta float a giustezza piena nel-

9. Se si dispone di una precedente versione di L<sup>A</sup>T<sub>E</sub>X, si può ancora estrarre il codice da sé, scrivendo un breve file di installazione `fltrace.ins` con il seguente contenuto:

```
\input docstrip
\generateFile{fltrace.sty}{t}{%
  \from{ltoutput.dtx}{fltrace,trace}}
\endbatchfile
```

ed elaborandolo con L<sup>A</sup>T<sub>E</sub>X.

la parte inferiore di una pagina composta a due colonne. Caricando il pacchetto `stfloats`, questa funzionalità mancante viene aggiunta, tranne che per la prima pagina.<sup>10</sup>

### 6.3 Assicurarsi che i float siano posati sempre dopo la loro chiamata

Per impostazione predefinita, l'algoritmo permette a un float di retrocedere prima della sua chiamata fin tanto che entrambi si trovano sulla stessa pagina. Più precisamente, gli permette di comparire nell'area superiore della colonna in cui è stato incontrato.

Questa prassi offre al float migliori possibilità di essere visibile dal punto in cui si trova la chiamata e di non finire in una pagina successiva. Tuttavia, le redazioni di alcune riviste considerano questo comportamento troppo permissivo e richiedono che gli oggetti siano messi tassativamente dopo la loro chiamata. Dal punto di vista dell'algoritmo, in altre parole, solo l'area inferiore della colonna in cui c'è la chiamata costituisce un'opzione di posa valida. Soddisfa tale requisito il pacchetto `flafter`, che implementa questa strategia.

La cosa può funzionare bene se il documento contiene solo pochi float. Per documenti che ne contengono moltissimi, ovviamente, la questione diventa molto più difficile, tanto che si potrebbero trovare gli oggetti tutti insieme alla fine del documento o del capitolo, o si potrebbe ricevere l'errore `Too many unprocessed floats`.

### 6.4 Evitare float in certe pagine

Talvolta è utile impedire che i float compaiano in una determinata pagina, come per esempio impedire che un float presente in una nuova sezione finisca nell'area superiore della pagina corrente ma consentendo un piazzamento nell'area superiore di una pagina successiva. Per questo tipo di messa a punto fine, L<sup>A</sup>T<sub>E</sub>X offre il comando `\suppressfloats[⟨collocazione⟩]`, dove `⟨collocazione⟩` può essere `t` o `b` e impedisce ogni ulteriore posa nelle rispettive aree della pagina corrente. Se non lo si dà, tutti i float rimanenti sulla pagina corrente verranno differiti.

### 6.5 Implementare barriere per i float

L<sup>A</sup>T<sub>E</sub>X standard implementa già una barriera per i float chiamata `\clearpage`. Se nel testo sorgente un float si trova prima della barriera, nel documento finito non potrà mai comparire dopo, e viceversa. La cosa funziona così: il programma emette tutti i float differiti estraendoli dalla coda di attesa, genera pagine `p` se necessario e infine comincia una nuova pagina. Mentre questo meccanismo è adatto a mantenere gli oggetti all'interno di un capitolo (dato che i capitoli cominciano di solito su una pagina nuova) esistono situazioni in cui si

10. Si veda (BEETON, 2014) per eliminare a mano anche questa restrizione.

potrebbe desiderare una barriera meno rigida, cioè che funzioni senza forzare una nuova pagina o che sia parzialmente permeabile.

Questa funzionalità è offerta dal pacchetto `placens`, il cui comando `\FloatBarrier` non introduce un'interruzione di pagina. Tramite le opzioni del pacchetto si può modificare il comportamento della barriera per permettere ai float di migrare prima o dopo di essa, ma fin tanto che compaiono ancora sulla stessa pagina.

### 6.6 Sovrascrivere restrizioni di posa

Se un determinato float è (leggermente) troppo grande per stare in una certa area di float o se un'area contiene già il massimo numero di float permesso ma si vuole forzare comunque un ulteriore oggetto in questa posizione, allora è una buona scelta aggiungere `!` all'argomento facoltativo. Questo carattere ordina all'algoritmo di ignorare tutte le eventuali restrizioni gravanti sul float in questione attuate mediante parametri, in modo che l'oggetto verrà posato sempre, purché non ci siano già altri float differiti della stessa categoria o le aree permesse non diventino più ampie dello spazio disponibile quando si aggiunge l'oggetto.

Siccome l'ordine dei tentativi è sempre lo stesso (prima in alto, poi in basso), si può usare `[\!b]` per forzare un float nell'area inferiore poiché `[\!tb]` riuscirebbe già normalmente a posarlo in quella superiore. Ovviamente, il rovescio della medaglia è che se il float non ci sta, comparirà solo nell'area inferiore di una pagina successiva. Perciò, qualsiasi modifica del testo operata in seguito potrebbe creare il caos nelle decisioni di posa dell'utente.

### 6.7 Un consiglio finale di messa a punto

Esistono molti modi per regolare il comportamento dell'algoritmo e la maggior parte di essi è stata discussa in questo articolo. Tuttavia, c'è ancora

una 'messa a punto' possibile, la più potente di tutte: le modifiche al testo del documento.

Pertanto, come ultimo consiglio: non cominciate a manipolare i parametri o a cambiare i codici o a spostare i float nel documento sorgente finché il testo non è definitivamente concluso e il documento è prossimo al termine. Queste manovre sono uno spreco di energie e possono produrre facilmente pose scadenti poiché, le restrizioni imposte all'inizio potrebbero non risultare più adeguate dopo una (anche minima) modifica del testo.

## Riferimenti bibliografici

BEETON, B. (2014). «Placing a full-width insert at the bottom of two columns». *TUGboat*, **35** (3), p. 255. URL <https://www.tug.org/TUGboat/tb35-3/tb111beet-banner.pdf>.

DANIEL, M. (2012). *How to influence the position of float environments like figure and table in LATEX?* URL <http://tex.stackexchange.com/questions/39017/>.

GOOSSENS, M., MITTELBACH, F., BRAAMS, J., CARLISLE, D. e ROWLEY, C. (2004). *The LATEX Companion. Tools and Techniques for Computer Typesetting*. Addison-Wesley, Reading, MA, USA, 2<sup>a</sup> edizione.

HERMANN, M. (2012). *“Thanks” note (footnote) placed below right column even though there is enough space on the left*. URL <http://tex.stackexchange.com/questions/43294/>.

LAMPORT, L. (1994). *LATEX. A Document Preparation System. User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 2<sup>a</sup> edizione. Ristampato con correzioni nel 1996.

▷ Frank Mittelbach  
LATEX3 Project  
<http://www.latex-project.org/>