

siunitx: passato, presente e futuro

Joseph Wright*

Sommario

Il pacchetto `siunitx` fornisce una potente serie di strumenti per la composizione di numeri e unità in \LaTeX . Incorporando i dettagli sulle regole comunemente accettate per la presentazione dei dati scientifici, `siunitx` permette agli autori di concentrarsi sul contenuto e il significato del proprio contributo, accollandosi *in toto* l'onere della formattazione. In questa sede getterò uno sguardo sul retroterra del pacchetto, sulle motivazioni che mi hanno portato dalla versione 1 alla versione 2 e sul perché ora è in fase di sviluppo la terza versione.

Abstract

The `siunitx` package provides a powerful toolkit for typesetting numbers and units in \LaTeX . By incorporating detail about the agreed rules for presenting scientific data, `siunitx` enables authors to concentrate on the meaning of their input and leave the package to deal with the formatting. Here, I look at the background to the package, what led me from version 1 to version 2, and why version 3 is now under development.

1 Introduzione

In \TeX , la composizione delle unità si presta naturalmente a essere supportata da macro. Le regole formali per le unità SI (BIPM, 2006) mettono in relazione i nomi delle unità con i relativi simboli e non sorprende che numerosi autori abbiano scritto pacchetti che affrontano alcune o tutte le sottigliezze coinvolte. Tempo fa, l'ultima volta che ho esaminato `siunitx` per *TUGboat* (WRIGHT, 2011), ho descritto alcuni esempi degli uni e delle altre. Qui riassumerò brevemente alcuni di questi punti chiave, poi analizzerò le motivazioni trainanti verso la terza versione del pacchetto.

2 I primi tempi

Prima di `siunitx` esisteva un certo assortimento di pacchetti dedicati alla composizione delle unità, tra i quali uno, chiamato `Slunits` (HELDOORN e WRIGHT, 2007), che definiva macro di tipo semantico. Il mio coinvolgimento nella faccenda è incominciato quando, nel novembre del 2007, ho

*Questo articolo è stato pubblicato originariamente su *TUGboat*, vol. 39, n. 2 (2018), p. 119–121. È stato tradotto da Tommaso Gordini con il permesso dell'autore. Eventuali errori o fraintendimenti del testo originale sono del traduttore. (N.d.R.)

risposto a un messaggio apparentemente innocuo inviato da Stephen Pinnow a `comp.text.tex`, nel quale si lamentava un bug inequivocabile:

```
I want to report that \reciprocal,
\rpsquare, \rpcubic, etc. output is
written as "-1" instead of a $-1$,
when the package option "textstyle"
is used. I tried to contact Mr.
Heldoorn, but he didn't answer until
now. Does anyone have an idea what to
do?
```

«Vorrei segnalare che quando l'opzione `textstyle` è attiva, il risultato di `\reciprocal`, `\rpsquare`, `\rpcubic`, eccetera è stampato come `-1` anziché come `--1`. Ho provato a contattare il signor Heldoorn, ma finora non mi ha risposto. Qualcuno ha idea di che cosa fare?»

Essendo giovane e incosciente, dopo aver esaminato il bug e aver scoperto che `Slunits` non era più mantenuto, mi sono offerto volontario per rilevare il pacchetto. Ancora più incautamente, sempre nello stesso mese ho replicato come segue:

```
As some of you may have noticed,
following a recent bug report
concerning the SIunits package, I
have taken over as the package
maintainer. I have uploaded a bug fix
for the specific issue to CTAN, and
so hopefully it will appear within a
day or two.
```

```
It has been suggested by the
maintainer of the SIstyle package
that integration of the two be would
worth considering. Other suggestions
have also been made in the newsgroup
and by private mail. I am therefore
planning to review the existing
situation and see what improvements
are needed/desirable. As well as
SIunits and SIstyle, I am going to
look at numprint, units, unitsdef and
hepunits for inspiration/points to
consider/etc. So far, I have some
outline ideas, for example: ...
```

«Come alcuni di voi avranno notato, in seguito a una recente segnalazione di bug relativa al pacchetto `Slunits` mi sono assunto l'incarico di mantenere il pacchetto. Ho caricato su CTAN una correzione per il problema in questione, quindi spero che nel giro di uno o due giorni sarà visibile. Il manutentore del pacchetto `SIstyle` ha

consigliato di prendere in considerazione una possibile integrazione dei due pacchetti. Altri suggerimenti sono stati avanzati anche nel newsgroup e per posta privata. Pertanto, ho intenzione di rivedere la situazione esistente e valutare quali miglioramenti siano necessari o desiderabili. Oltre a `Slunits` e `Slstyle`, esaminerò `numprint`, `units`, `unitdef` e `hepunits` per ispirarmi, isolare i punti da considerare, eccetera. Per il momento ho solo abbozzato qualche idea, per esempio: ...»

In poco tempo ha preso forma un elenco piuttosto corposo di punti a cui mettere mano e il lavoro al nuovo pacchetto è incominciato sul serio; nel febbraio del 2008 ne è uscita la prima versione di test. Dopo averci lavorato ancora un po' e aver scelto un nuovo nome, il 16 aprile 2008 è approdata a CTAN la prima versione ufficiale di `siunitx` (WRIGHT, 2018).

3 Caratteristiche chiave

`siunitx` presenta le proprie caratteristiche fondamentali fin da quella prima versione e sono piuttosto conosciute. Le riassumerei come segue.

- Formattazione automatica e semantica delle quantità (numeri con unità).
- Analisi e gestione dei numeri.
- Controllo della stampa di numeri, unità e quantità.
- Allineamento dei numeri nelle tabelle.
- Interfaccia $\langle \text{chiave} \rangle$ - $\langle \text{valore} \rangle$ unificata per il controllo delle opzioni.

Per quanto mi riguarda, il pacchetto ha sempre riguardato le *unità* ed è rimasto fedele all'idea fondamentale per cui un input come

```
\joule\per\mole\per\kelvin
```

possa restituire $\text{J mol}^{-1} \text{K}^{-1}$, $\frac{\text{J}}{\text{mol K}}$ o $\text{J}/(\text{mol K})$ a seconda delle opzioni attive. Quest'idea c'è fin dal primo giorno e in questo senso il codice è rimasto più o meno immutato.

4 Dalla versione 1 alla versione 2

Fintanto che si trattava di garantire quelle caratteristiche chiave, la versione 1 di `siunitx` ha funzionato bene. I successivi rilasci si sono succeduti rapidamente culminando nella versione 1.4c del febbraio 2010. Tuttavia, aggiungere nuove funzionalità era un problema: internamente, infatti, il codice era un po' disastroso. Se si osserva il vecchio codice, per esempio, si vedranno:

- comandi interni *diversi da quelli dedicati all'analisi delle unità* presi da pacchetti esistenti e piuttosto a caso;

- scelte $\langle \text{chiave} \rangle$ - $\langle \text{valore} \rangle$ non ottimali;
- assenza quasi totale di API interne;
- cicli realizzati con codice personale di dubbia qualità;
- ...

Più o meno in questo periodo, Will Robertson mi contattò per chiedermi che cosa pensavo del linguaggio `expl3` di LATEX3 (LATEX3 PROJECT, 2019). Questo accadeva prima della mia entrata nel LATEX Team, e `expl3` sembrava un po' diverso da com'è oggi, anche se le idee centrali erano già tutte lì. Le idee mi piacevano, ma per il momento ero un po' cauto nel caricare una libreria esterna (e quindi una dipendenza). Così ho incominciato a selezionare le idee e a ricodificarle nella mia configurazione di sviluppo per la versione 2. Presto fu chiaro che avevo bisogno di *molte* idee e mi resi conto che sarei stato molto meglio semplicemente passando a `expl3`.

Lavorare alla seconda versione di `siunitx` mi ha portato a programmare in `expl3` e a chiedere al team molte funzionalità. In particolare, piuttosto che dover ricorrere a un pacchetto esterno, volevo avere un supporto integrato per le opzioni del tipo $\langle \text{chiave} \rangle$ - $\langle \text{valore} \rangle$. Così, per risolvere il problema ho scritto del codice che ho chiamato `keys3`. La cosa si sarebbe rivelata essere la mia parte per entrare a far parte del team: oggi, infatti, lo stesso `l3keys` è compreso in `expl3`!

La riscrittura mi ha dato la possibilità di rivedere in modo significativo gli aspetti interni delle API e di migliorarne sensibilmente le prestazioni. Inoltre, sono state introdotte nuove funzionalità per gli utenti e nomi completamente nuovi per l'interfaccia $\langle \text{chiave} \rangle$ - $\langle \text{valore} \rangle$. Nella versione 2.0 non ho incluso la retrocompatibilità: ho imparato alla svelta, e questa caratteristica è stata presente fin da pochi giorni dopo il suo rilascio.

5 Dalla versione 2 alla versione 3

La versione 2 di `siunitx` conserva la maggior parte delle funzionalità presenti in quella precedente, ma, oltre a quelle buone, ne mantiene anche alcune di cattive! In particolare:

- ipotesi sui font: OpenType, eccetera;
- nessuna API a livello di codice `expl3`;
- comandi interni ancora troppo disordinati;
- test della sola documentazione in PDF;
- codice sorgente monolitico;
- lentezza ancora eccessiva.

5.1 Controllo dei font

Le ipotesi sui font, che ho modificato solo leggermente, sono state prese in blocco da `Slstyle` (ELS, 2008). L’approccio attualmente seguito è il seguente.

1. Rilevamento del carattere corrente mediante i dati interni di `LATEX`.
2. Inserimento del tutto all’interno di `\text`.
3. Applicazione di `\ensuremath` dentro la scatola.
4. Probabilmente, ulteriore applicazione di `\text` (per l’output in modo testo).
5. Forzatura del font con `\mathrm` o `\rmfamily`, per esempio.

Il tutto richiede molto lavoro e, cosa più importante, è inaffidabile: non è sempre facile ottenere il carattere giusto ‘dentro’ la sezione di output. Inoltre, il procedimento fallisce malamente con i font OpenType in modo matematico, dove le idee del `TEX` tradizionale sulle famiglie matematiche semplicemente non si applicano. Quindi, la versione 3 prevede un nuovo approccio.

1. Rilevamento del carattere corrente mediante i dati interni di `LATEX`.
2. Impostazione di tutti gli aspetti *necessari*.
3. Uso di `\mbox` solo se è necessario modificare la versione matematica.

Questo approccio ‘a cambiamento minimo’ è molto più veloce di quello attuale e molto più efficace nel rispettare i cambiamenti del font. Sto ancora ultimando la compatibilità per le attuali configurazioni dei casi limite, ma credo che, nel complesso, il nuovo codice sia di molto preferibile al precedente.

5.2 Il codice delle API e il test

Lo sviluppo della versione 2 di `siunitx` è molto simile a quello di `expl3` come linguaggio completamente utilizzabile: nel periodo in cui l’ho adoperato, `expl3` è passato dall’essere una serie di esperimenti intelligenti a costituire un approccio consolidato alla scrittura di codice `TEX`. Ma mantenere ogni cosa di `siunitx` al passo con le idee è stato complicato.

Il problema più grande è che quando ho scritto il codice dell’attuale versione, c’erano solo comandi per l’utente come `\num` e relativa implementazione interna. Tuttavia, ora è chiaro che ogni comando per l’utente dovrebbe avere un’interfaccia *documentata* a livello di codice. Inoltre, queste interfacce dovrebbero essere testate a dovere: il team ha creato `l3build` (`LATEX3 PROJECT`, 2018) proprio per questo. Combinando queste idee, il nuovo codice avrà come input

```
\siunitx_unit_format:nN
  { \joule \per \mole }
  \l_tmpa_tl
\l_show:N \l_tmpa_tl
e restituirà
> \l_tmpa_tl=
  \mathrm {J}\,\mathrm {mol}^{-1}
```

Si noti che questo secondo codice è facile da testare e mette in evidenza un’altra nuova idea: la fase di analisi dovrebbe produrre gli stessi risultati ottenuti da un utente che scriva la formattazione ‘a mano’.

6 Prima versione alfa

Mentre sto scrivendo, lo sviluppo della versione 3 ha raggiunto il primo stadio alfa: il codice è adoperabile ma ci sono autentiche lacune. Attualmente funziona tutto quanto si elenca di seguito:

- funzioni di base:
 - analisi e formattazione delle unità;
 - formattazione dei numeri reali;
 - colonne nelle tabelle;
- API esistenti: `\num`, `\SI`, `\si`, colonna `S`;
- nuove API (sperimentali): `\unit`, `\qty`.

Ci sono alcune ampie sezioni ancora da coprire, come numeri complessi, intervalli, elenchi e, soprattutto, il livello di compatibilità per lavorare con i documenti già esistenti. Tuttavia, il tutto è relativamente gestibile e mi aspetto di terminare il lavoro verso la fine dell’anno.¹ Il 2019, dunque, dovrebbe vedere `siunitx` raggiungere la versione 3.0.0, sperando che mantenga il proprio posto come *il* pacchetto di `LATEX` per le unità.

A Esempi

Formattazione semplice dei numeri:

123	<code>\num{123}</code>
1234	<code>\num{1234}</code>
12 345	<code>\num{12345}</code>
0,123	<code>\num{0,123}</code>
0,1234	<code>\num{0,1234}</code>
0,123 45	<code>\num{,12345}</code>
3,45 × 10 ⁻⁴	<code>\num{3,45d-4}</code>
-10 ¹⁰	<code>\num{-e10}</code>

Angoli:

10°	<code>\ang{10}</code>
12,3°	<code>\ang{12,3}</code>
4,5°	<code>\ang{4,5}</code>
1°2’3’’	<code>\ang{1;2;3}</code>
1’’	<code>\ang{;;1}</code>
10°	<code>\ang{+10;;}</code>
-0°1’	<code>\ang{-0;1;}</code>

1. 2018. (*N.d.T.*)

Unità come macro:

kg m s^{-2}	<code>\si{\kilo\gram\metre\per\square\second}</code>
g cm^{-3}	<code>\si{\gram\per\cubic\centi\metre}</code>
$\text{V}^2 \text{lm}^3 \text{F}^{-1}$	<code>\si{\square\volt\cubic\lumen\per\farad}</code>
$\text{m}^2 \text{Gy}^{-1} \text{lx}^3$	<code>\si{\metre\squared\per\gray\cubic\lux}</code>
Hs	<code>\si{\henry\second}</code>

Quantità:

$1,23 \text{ J mol}^{-1} \text{ K}^{-1}$	<code>\SI[mode=text]{1,23}{J.mol^{-1}.K^{-1}}</code>
$0,23 \times 10^7 \text{ cd}$	<code>\SI{,23e7}{\candela}</code>
$1,99/\text{kg}$	<code>\SI[per-mode=symbol]{1,99}{\per\kilogram}</code>
$1,345 \frac{\text{C}}{\text{mol}}$	<code>\SI[per-mode=fraction]{1,345}{\coulomb\per\mole}</code>

Riferimenti bibliografici

LATEX3 PROJECT (2018). *The l3build package. Checking and building packages.* <https://ctan.org/pkg/l3build>.

— (2019). *The expl3 package and LATEX3 programming.* <https://ctan.org/pkg/exp3>.

BIPM (2006). «The International System of Units (SI)». Technical report, Bureau International des Poids et Mesures.

ELS, Daniel (2008). *The Slstyle package.* <https://ctan.org/pkg/sistyle>.

HELDOORN, Marcel e Joseph WRIGHT (2007). *The Slunits package. Consistent application of SI units.* <https://ctan.org/pkg/siunits>.

WRIGHT, Joseph (2011). «siunitx: A comprehensive (SI) units package». *TUGboat*, **32** (1), pp. 95–98. <http://tug.org/TUGboat/tb32-1/tb100wright-siunitx.pdf>.

— (2018). *siunitx — A comprehensive (SI) units package.* <https://ctan.org/pkg/siunitx>.

▷ Joseph Wright
Morning Star
2, Dowthorpe End
Earls Barton
Northampton NN6 0NH
United Kingdom
joseph dot wright (at)
morningstar2.co.uk