

# Le module `paresse`\*

Le T<sub>E</sub>Xnicien de surface  
[le.texnicien.de.surface@wanadoo.fr](mailto:le.texnicien.de.surface@wanadoo.fr)

13 août 2008

## Résumé

Ce module, reprenant un exemple de T. LACHAND-ROBERT dans [1], fournit un moyen de taper des lettres grecques isolées à l'aide du caractère actif et redéfini. Au lieu de `\(\alpha\)` ou tape `§a` pour obtenir  $\alpha$ .

**Important :** Il doit être chargé **après** `inputenc` si ce dernier est utilisé. De plus, il faut que le signe § soit une lettre pour T<sub>E</sub>X.

## Abstract

This package implements an example from T. LACHAND-ROBERT in [1]. It provides a means of typing isolated greek letters with the character § activated and redefined. Instead of `\(\alpha\)` one types `§a` to obtain  $\alpha$ .

**Important:** You have to load it **after** the `inputenc` package if the latter is used. Moreover the sign § must be a letter for T<sub>E</sub>X.

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Utilisation</b>	<b>2</b>
2.1	Les options . . . . .	2
2.2	Commandes et environnement . . . . .	2
2.3	Tableau des macros . . . . .	3
<b>3</b>	<b>Le code</b>	<b>3</b>
	<b>Bibliographie</b>	<b>6</b>

## 1 Introduction

Cette extension ne fournit qu'un accès « rapide et économique » aux lettres grecques qui s'obtiennent à l'aide d'une macro comme `\alpha` ou `\Omega`. Elle fournit un environnement et une commande qui permettent d'utiliser § pour taper ces lettres. Un `\ensuremath` nous dispense de nous placer explicitement — c'est-à-dire en tapant `$ $` ou bien `\( \)` ou encore `\[ \]` ou tout autre chose ayant le même effet — en mode mathématique pour obtenir une lettre grecque.

---

\*Ce document correspond au fichier `paresse.sty` v2, du 2008/08/13, édition du cinquantième.

L'idée de la méthode est due à T. LACHAND-ROBERT et est exposée dans [1]. Je n'ai fait qu'ajouter le `\ensuremath` bien agréable pour l'écriture de macros.

Bien entendu, on **ne** dispose **pas** de macros pour la minuscule omicron ni pour les majuscules alpha, beta... qui s'obtiennent à l'aide des latines romaines de même apparence<sup>1</sup>. Je ne me suis pas senti le courage ni la force de fournir une solution qui permettent d'obtenir dans une formule baignant dans un texte en italique gras un alpha majuscule droit, romain, &c.

Pour finir cette introduction, glosons le nom de cette extension. `paresse` vient de ce que le signe § indique un « paragraphe » en ayant une forme lointainement apparentée au S et n'a donc aucun lien avec le *défaut* si fréquent, encore que, à bien y réfléchir...

## 2 Utilisation

On charge cette extension avec `\usepackage{paresse}` **après** l'extension `inputenc`. Il faut que la « caractère » § soit reconnu par `TeX` comme une lettre. On pourra, par exemple, utiliser l'option `latin1` de `inputenc` pour obtenir cela.

Par défaut, l'extension est chargée avec l'option `wild` ce qui signifie que les macros de la forme `§a` sont disponibles. Si l'on préfère, on peut passer à `paresse` l'option `tame` avec `\usepackage[tame]{paresse}`. Il faut alors utiliser la commande `\ActiveLaParesse` ou l'environnement `ParesseActive` pour utiliser les macros en §.

Lorsque la « paresse est active », il suffit de taper `§a` pour obtenir  $\alpha$ . On a de même accès à toutes les lettres grecques auxquelles sont consacrées une macro comme `\alpha`, voyez 3. On obtient, de même,  $\alpha^\beta$  avec `\(§a~{§b}\)` lorsque § est active. On notera que, si l'extension `amsmath` est utilisée, les accolades sont optionnelles et qu'on obtient un résultat identique avec `\(§a~§b\)`.

### 2.1 Les options

- `tame` s'oppose à `wild` qui est l'option par défaut. Lorsque `tame` règne, il **faut** un environnement `ParesseActive` ou une commande `\ActiveLaParesse` pour utiliser les macros à §.
- `ttau` s'oppose à `ttheta` qui est l'option par défaut. Lorsque `ttheta` est active `§t` donne  $\theta$  dans le cas contraire `§t` donne  $\tau$ . En tous cas,  $\theta$  est accessible par `§v` et  $\tau$  par `§y`.  
**Remarque :**  $\Theta$  est obtenu par `§V` de manière « régulière » et *également* par `§T` quelle que soit l'option choisie.
- `epsilon` s'oppose à `varepsilon` qui est l'option par défaut. Avec `epsilon`, `§e` donne  $\epsilon$  sinon `§e` donne  $\varepsilon$ .
- Se comportent comme le couple `epsilon`, `varepsilon` les couples suivants `theta` et `vartheta`; `pi` et `varpi`; `rho` et `varrho`; `sigma` et `varsigma`; `phi` et `varphi`.

Par défaut on a `varepsilon`, `theta`, `pi`, `rho`, `sigma`, `varphi` et `wild`.

### 2.2 Commandes et environnement

`\makeparesseletter`

Cette macro donne au « caractère » § le catcode d'une lettre. Après cela, on

---

1. Je ne ferai aucune remarque sur les problèmes de codage que cela pose.

peut se servir de § dans un nom de macro, par exemple. C'est le pendant de `\makeatletter`.

`\makeparesseother` Cette macro donne au caractère § le catcode *other*. C'est le « contraire » de la précédente. Cela équivaut au `\makeatother`.

`\ActiveLaParesse` Cette macro active le caractère § et permet ainsi d'accéder aux macros dont le nom commence par § comme §a. Pour une liste de ces macros et leurs significations, voyez le tableau 3.

`ParesseActive` Dans cet environnement le caractère § est actif ce qui permet d'utiliser les macros en §. On utilisera cet environnement si l'on veut utiliser les macros quand on a chargé l'extension *paresse* avec l'option *tame*.

## 2.3 Tableau des macros

§a	$\alpha$	§b	$\beta$	§g	$\gamma$	§d	$\delta$
§e	$\varepsilon$	§z	$\zeta$	§h	$\eta$	§v	$\theta$
§i	$\iota$	§k	$\kappa$	§l	$\lambda$	§m	$\mu$
§n	$\nu$	§x	$\xi$	§p	$\pi$	§r	$\rho$
§s	$\sigma$	§y	$\tau$	§u	$\upsilon$	§f	$\varphi$
§c	$\chi$	§q	$\psi$	§w	$\omega$		
§G	$\Gamma$	§D	$\Delta$	§V	$\Theta$	§L	$\Lambda$
§X	$\Xi$	§P	$\Pi$	§S	$\Sigma$	§U	$\Upsilon$
§F	$\Phi$	§Q	$\Psi$	§W	$\Omega$		

**Remarques :** à l'exception de  $\theta$ ,  $\tau$  et  $\psi$  les lettres utilisées dans les noms des macros sont chargées de vertu mnémotechniques :-)) et les majuscules grecques s'obtiennent à l'aide de la majuscule correspondante.

## 3 Le code

Le code est fourni presque sans commentaire. J'espère qu'il est assez clair. Cela dit, il ressemble beaucoup à celui de l'extension *engpron* du même *auteur*.

Ici j'utilise `\def` à outrance pour récupérer les redéfinitions des macros faites par des extensions comme *fourier*<sup>2</sup>.

On dit bonjour. On définit quelques variables booléennes et on fixe leur valeur par défaut. On traite les options et on crie si l'une d'elles n'est pas connue.

```
1 \newif\if@optiontame      \newif\if@optiontau
2 \newif\if@optionvareps    \newif\if@optionvartheta
3 \newif\if@optionvarrho    \newif\if@optionvarsigma
4 \newif\if@optionvarphi    \newif\if@optionvarpi
```

Déclarations des options. Suivant l'option on fixe la valeur des booléens avec les `\@option...true/false`.

```
5 \DeclareOption{tame}{\@optiontametrue}
```

---

2. Mes remerciements à Michel BOVANI pour ses conseils sur [fr.comp.text.tex](http://fr.comp.text.tex).

```

6 \DeclareOption{wild}{\@optiontamefalse}
7 \DeclareOption{ttau}{\@optiontauttrue}
8 \DeclareOption{ttheta}{\@optiontaufalse}
9 \DeclareOption{varepsilon}{\@optionvarepstrue}
10 \DeclareOption{epsilon}{\@optionvarepsfalse}
11 \DeclareOption{vartheta}{\@optionvarthetatrue}
12 \DeclareOption{theta}{\@optionvarthetafalse}
13 \DeclareOption{varrho}{\@optionvarrhotrue}
14 \DeclareOption{rho}{\@optionvarrhofalse}
15 \DeclareOption{varpi}{\@optionvarpittrue}
16 \DeclareOption{pi}{\@optionvarpifalse}
17 \DeclareOption{varsigma}{\@optionvarsigmatrue}
18 \DeclareOption{sigma}{\@optionvarsigmafalse}
19 \DeclareOption{varphi}{\@optionvarphitrue}
20 \DeclareOption{phi}{\@optionvarphifalse}

```

On traite le cas des options inconnues avec un message d'erreur.

```

21 \DeclareOption*{\PackageError{paresse}
22   {The option \CurrentOption\space is not known!\MessageBreak
23     The default options are:\MessageBreak
24     varepsilon, theta, pi, rho, sigma, and varphi.}%
25   {Choose between 'wild' (for an active Paresse in all the document)
26     \MessageBreak and 'tame' (so that $ prints the paragraph symbol).
27     \MessageBreak See the documentation for other options.}}

```

On active les options par défaut : wild, varepsilon, theta, pi, rho, sigma et varphi.

```

28 \ExecuteOptions{wild,varepsilon,theta,pi,rho,sigma,varphi}
29 \ProcessOptions\relax

```

On sauvegarde le catcode et la définition courante du caractère \$.

```

30 \count@=\catcode'\$
31 \let\GA@LaVieilleParesse= $

```

On définit une macro pour pouvoir obtenir le caractère lui-même. Ça peut servir, parfois...

```

32 \def\${\$}

```

On définit une série de macros plus ou moins « cachées ». On doit recourir à des acrobaties pour obtenir ce que l'on veut et pour éviter que T<sub>E</sub>X ne hurle des méchancetés du genre \inaccessible.

```

33 \newcommand{\makeparesseletter}{\catcode '\$=11\relax}
34 \newcommand{\makeparesseother}{\catcode '\$=12\relax}
35 \newcommand{\GA@MakeParesseActive}{\catcode '\$=\active}
36 \def\GA@ActiveLaParessexii{\catcode '\$=\active}
37 \def\GA@ActiveLaParessexiii{\catcode '\$=\active
38   \def$##1{\ensuremath{\csname \string$##1\endcsname}}}
39 \ifnum\count@=12
40 \let\GA@ActiveLaParesse \GA@ActiveLaParessexii
41 \else
42 \let\GA@ActiveLaParesse \GA@ActiveLaParessexiii
43 \fi

```

La macro utile puis l'environnement itou. Ils font tous les deux appels à la macro *cachée* qui suit.

```

\ActiveLaParesse Ce n'est qu'un alias pour \GA@ActiveLaParesse.
44 \newcommand{\ActiveLaParesse}{\GA@ActiveLaParesse}

ParesseActive Environnement minimaliste qui se contente d'appeler \GA@ActiveLaParesse en
bornant son effet par le groupe créé par l'environnement.
45 \newenvironment{ParesseActive}{\GA@ActiveLaParesse}{

    Une série de macros pour les lettres à variantes.
46 \if@optionvareps \def\@epsilon {\varepsilon}
47 \else \def\@epsilon {\epsilon} \fi
48 \if@optionvartheta \def\@theta {\vartheta}
49 \else \def\@theta {\theta} \fi
50 \if@optionvarpi \def\@pi {\varpi}
51 \else \def\@pi {\pi} \fi
52 \if@optionvarrho \def\@rho {\varrho}
53 \else \def\@rho {\rho} \fi
54 \if@optionvarsigma \def\@sigma {\varsigma}
55 \else \def\@sigma {\sigma} \fi
56 \if@optionvarphi \def\@phi {\varphi}
57 \else \def\@phi {\phi} \fi

    On fait de § une lettre et on définit les macros à coup de \def\S.... Avant de
    passer aux majuscules, on s'occupe de la lettre t (option ttau et ttheta).
58 \makeparesseletter

59 \def\Sa {\alpha} \def\Sb {\beta}
60 \def\Sg {\gamma} \def\Sd {\delta}
61 \def\Se {\@epsilon} \def\Sz {\zeta}
62 \def\Sh {\eta} \def\Sv {\@theta}
63 \def\Si {\iota} \def\Sk {\kappa}
64 \def\Sl {\lambda} \def\Sm {\mu}
65 \def\Sn {\nu} \def\Sx {\xi}
66 \def\Sp {\@pi} \def\Ss {\@sigma}
67 \def\Sr {\@rho} \def\Su {\@upsilon}
68 \def\Sy {\tau} \def\Sc {\@chi}
69 \def\Sf {\@phi} \def\Sw {\omega}
70 \def\Sq {\psi}

71 \if@optiontau \def\St {\tau}\else \def\St {\@theta}\fi

72 \def\SG {\Gamma} \def\SD {\Delta}
73 \def\SV {\Theta} \def\ST {\Theta}
74 \def\SL {\Lambda} \def\SX {\Xi}
75 \def\SP {\Pi} \def\SS {\Sigma}
76 \def\SU {\Upsilon} \def\SF {\Phi}
77 \def\SQ {\Psi} \def\SW {\Omega}

    On rend actif le caractère § et on définit les macros à noms commençant par §.
78 \catcode '\S=\active
79 \def\S#1{\ensuremath{\csname \stringS#1\endcsname}}

    Si l'option tame est en vigueur, on remet § dans son état initial et on sort.
80 \if@optiontame
81 \catcode'\S=\count@
82 \ifnum\count@=13 \let S=\GA@LaVieilleParesse \fi
83 \fi
84 \endinput

```

## Références

- [1] T. LACHAND-ROBERT. *La maîtrise de T<sub>E</sub>X et L<sup>A</sup>T<sub>E</sub>X*. Masson, Paris, Milan, Barcelone, 1995.  
ISBN : 2-225-84832-7.

## Changements

v0.0		dans Infofile qui empêchait le chargement du module. . . . . 1
	Général : Nouveau nom, quelques redéfinitions, première version publique. . . . . 1	v2
v1		Général : Réemballage pour fournir un tds.zip. . . . . 1
	Général : Mise à jour de la documentation. . . . . 1	Réorganisation de la documentation. . . . . 1
v1a		Suppression d'une macro inutile. 1
	Général : Correction d'une erreur	

## Index

Les nombres en italique renvoient à la page où l'entrée est décrite; les nombres soulignés renvoient à la ligne de code de la définition; les nombres en caractères romains renvoient à la ligne de code où l'entrée est utilisée.

<b>Symbols</b>	<code>\@sigma</code> . . . . . 54, 55, 67	<code>\if@optiontau</code> . . . . 1, 71
<code>\@epsilon</code> . . . 46, 47, 61	<code>\@theta</code> . . 48, 49, 62, 71	<code>\if@optionvareps</code> . 2, 46
<code>\@optiontamefalse</code> . . 6	<b>A</b>	<code>\if@optionvarphi</code> . 4, 56
<code>\@optiontametrue</code> . . . 5	<code>\ActiveLaParesse</code> . . <u>44</u>	<code>\if@optionvarpi</code> . . 4, 50
<code>\@optiontaufalse</code> . . . 8	<b>E</b>	<code>\if@optionvarrho</code> . 3, 52
<code>\@optiontautru</code> . . . . 7	environnements :	<code>\if@optionvarsigma</code> 3, 54
<code>\@optionvarepsfalse</code> 10	<code>ParesseActive</code> . . <u>45</u>	<code>\if@optionvartheta</code> 2, 48
<code>\@optionvarepstrue</code> . . 9	<b>G</b>	<b>M</b>
<code>\@optionvarphifalse</code> 20	<code>\GA@ActiveLaParesse</code>	<code>\makeparesseletter</code> .
<code>\@optionvarphitrue</code> . 19	. . . . 40, 42, 44, 45	. . . . . 33, 58
<code>\@optionvarpifalse</code> . 16	<code>\GA@ActiveLaParessexii</code>	<code>\makeparesseother</code> . 34
<code>\@optionvarpitrue</code> . 15	. . . . . 36, 40	<b>P</b>
<code>\@optionvarrhofalse</code> 14	<code>\GA@ActiveLaParessexiii</code>	<code>ParesseActive</code> (envi-
<code>\@optionvarrhotrue</code> . 13	. . . . . 37, 42	ronnement) . . . <u>45</u>
<code>\@optionvarsigmafalse</code>	<code>\GA@LaVieilleParesse</code>	<b>S</b>
. . . . . 18	. . . . . 31, 82	<code>\S</code> . . . . . 32
<code>\@optionvarsigmatrue</code> 17	<code>\GA@MakeParesseActive</code>	<b>§</b>
<code>\@optionvarthetafalse</code>	. . . . . 35	<code>\S</code> . 30, 32–37, 59–78, 81
. . . . . 12	<b>I</b>	
<code>\@optionvarthetatrue</code> 11	<code>\if@optiontame</code> . . . 1, 80	
<code>\@phi</code> . . . . . 56, 57, 69		
<code>\@pi</code> . . . . . 50, 51, 66		
<code>\@rho</code> . . . . . 52, 53, 67		