

L'extension pour $\text{T}_{\text{E}}\text{X}$ et $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

systeme

v0.51

19 décembre 2025

Manuel de l'utilisateur

Christian TELLECHEA

unbonpetit@netc.fr

Résumé

Cette petite extension met en forme des systèmes d'équations ou d'inéquations où les termes et les signes sont alignés verticalement, tout en permettant une saisie quasi naturelle.

1	Versions 0.5x	2
2	La commande <code>\systeme</code>	2
3	La commande <code>\setssysteme</code>	3
4	Tri des inconnues, la clé <code>sort</code>	3
	4.1 Inconnues non indicées	3
	4.2 Inconnues indicées	5
5	Les autres clés et valeurs	5
	5.1 Séparateur entre équations, la clé <code>eq sep</code>	5
	5.2 Retour charriot entre équations, la clé <code>cr eq sep</code>	6
	5.3 Équations vides ignorées, la clé <code>ignore empty eq</code>	6
	5.4 Choix des délimiteurs, la clé <code>delim</code>	7
	5.5 Alignement des termes dans les colonnes, la clé <code>align</code>	7
	5.6 Alignement des signes des termes constants, la clé <code>align const sign</code>	7
	5.7 Liste des séparateurs de membres, la clé <code>member sep list</code>	7
	5.8 Ajout/suppression d'un séparateur de membres, les clés <code>add member sep</code> et <code>remove member sep</code>	8
	5.9 Substitution de séparateur de membres, la clé <code>member sep subst</code>	8
	5.10 Espacements : les clés <code>sign space</code> , <code>member sep space</code> , <code>line skip coeff</code> et <code>coeff space</code>	8
	5.11 Colonne supplémentaire, clés <code>extra col sign</code> , <code>extra col pre</code> et <code>extra col post</code>	9
	5.12 Numérotation automatique, les clés <code>autonom code</code> , <code>autonom continue</code>	9
	5.13 Augmentation en hauteur et profondeur, la clé <code>extra height</code>	11
	5.14 Non alignement des termes, la clé <code>align terms</code>	11
	5.15 Substitution post traitement, la clé <code>post subst</code>	11
	5.16 Sauvegarde du code produit dans une macro, la clé <code>store</code>	12
	5.17 Code exécuté avant l'alignement, la clé <code>code before</code>	12
6	Exemples	13

Pour l'utiliser l'extension « systeme », il faut exécuter :

- `\input systeme.tex` ou `\input{systeme.tex}` lorsqu'on utilise (pdf)(Xe)(lua)TeX ;
- `\usepackage{systeme}` dans le préambule lorsqu'on utilise (pdf)(Xe)(lua)LaTeX.

Les extensions `xstring` et `simplekv` sont requises et chargées si cela n'a pas été le cas.

1 Versions 0.5x

nouveau v0.5 Dans une consigne de tri (et seulement là), des inconnues peuvent comporter plusieurs tokens ; ainsi, les inconnues « `x^2` », « `x'` », « `x'^2` », ou encore « `a'_{*}` » sont valides dans une consigne de tri. On peut donc composer des systèmes non linéaires :

Inconnues comportant plusieurs tokens

<pre>\systeme[sort = {x^2 , x = x', x'^2 }]{ 2x-3x^2 = 1-x'^2+x', 1+2x^2+x = m+x'^2-2x } \medbreak \systeme[sort = {a'_{*} = {} }]{ a+a'+a'_1=2, 2a'-3a'_1+2a'_2=k=1 }</pre>	$\begin{cases} -3x^2 + 2x & = x' - x'^2 + 1 \\ 2x^2 + x + 1 & = x'^2 + m - 2x \end{cases}$ $\begin{cases} a' + a'_1 & + a = 2 \\ 2a' - 3a'_1 + 2a'_2 - k & = 1 \end{cases}$
--	--

2 La commande `\systeme`

La commande principale est `\systeme` dont l'argument obligatoire contient les équations séparées les unes des autres par une virgule :

La commande `\systeme`

<pre>un système : \systeme{ 2a-3b+4c=2, a+8b+5c=8, -a+2b+c=-5 }</pre>	$\text{un système : } \begin{cases} 2a - 3b + 4c = 2 \\ a + 8b + 5c = 8 \\ -a + 2b + c = -5 \end{cases}$
---	--

La commande `\systeme` fonctionnera indifféremment en mode math ou non. Les équations doivent être *développées*, c'est-à-dire que chaque terme est séparé de son voisin par un « + » ou un « - ». De plus, les inconnues doivent être des lettres minuscules non accentuées, c'est-à-dire tout caractère de « a » à « z ». Les espaces sont ignorés dans l'argument de la macro `\systeme`.

L'alignement construit sera précédé d'une accolade et aura les spécificités suivantes :

- les séparateurs de membres (signes d'égalité ou d'inégalité) sont alignés ;
- les signes « + » ou « - » entre chaque terme des membres sont alignés ;
- les terme du membre de gauche se trouvent dans une colonne au fer à droite et ceux du membre de droite dans une colonne au fer à gauche.

Les inconnues peuvent avoir un indice, sous réserve que celui-ci soit un nombre *entier positif ou nul* (la clé `sort` permet, depuis la version 0.41 de spécifier des indices non entiers) :

Inconnues indicées

<pre>\systeme{ 4x_1-x_2=3, -x_1+5x_2=-1 }</pre>	$\begin{cases} 4x_1 - x_2 = 3 \\ -x_1 + 5x_2 = -1 \end{cases}$
---	--

Si une inconnue est manquante dans une équation, la colonne de l'alignement reste vide :

Inconnues manquantes

<pre>\systeme{ a-2b=3, b-3c=4, -a+4c=-1 }</pre>	$\begin{cases} a - 2b & = 3 \\ & b - 3c = 4 \\ -a & + 4c = -1 \end{cases}$
---	--

Les inconnues et les termes constants peuvent être présents aussi bien dans le membre de droite que celui de gauche. Les termes constants sont toujours affichés en dernier.

Inconnues dans les deux membres

$$\begin{array}{l} \backslash\text{systeme}\{ \\ \quad -a+c-3=-2b+3c-1, \\ \quad 2b-3c=a+2b-2, \\ \quad a-b+c+1=2a-c \\ \} \end{array} \quad \left\{ \begin{array}{l} -a + c - 3 = -2b + 3c - 1 \\ 2b - 3c = a + 2b - 2 \\ a - b + c + 1 = 2a - c \end{array} \right.$$

Quel que soit l'ordre dans lequel sont entrées les inconnues lors de la saisie, elles seront triées par ordre alphabétique à l'affichage, en tenant compte de leur éventuel indice. Un signe « + » ne sera pas affiché lorsqu'il précède le premier terme d'une équation :

Tri des inconnues

$$\begin{array}{l} \backslash\text{systeme}\{ \\ \quad 2y+x-3z=4, \\ \quad z-y+2x=-1, \\ \quad -2x+3z-4y=0 \\ \} \end{array} \quad \left\{ \begin{array}{l} x + 2y - 3z = 4 \\ 2x - y + z = -1 \\ -2x - 4y + 3z = 0 \end{array} \right.$$

Les macros `\+` et `\-` sont redéfinies pendant l'exécution de la macro `\systeme` pour échapper « + » et « - », au cas où les coefficients doivent contenir ces signes, afin qu'ils ne soient compris comme la fin d'un terme :

Coefficient contenant une addition ou soustraction

$$\begin{array}{l} \backslash\text{systeme}\{ \\ \quad (2\backslash+\sqrt{2})x-(1\backslash-\sqrt{2})y=1, \\ \quad x+(1\backslash+\sqrt{2})y=-1 \\ \} \end{array} \quad \left\{ \begin{array}{l} (2 + \sqrt{2})x - (1 - \sqrt{2})y = 1 \\ x + (1 + \sqrt{2})y = -1 \end{array} \right.$$

3 La commande `\setssysteme`

La macro `\setssysteme` reçoit dans son argument obligatoire une liste de « clés/valeurs »

```
\setssysteme{<clé1>=<valeur1> ,<clé2>=<valeur2>,etc}
```

Les réglages concernés seront modifiés pour tout le reste du document. Pour revenir aux réglages par défaut, il faut exécuter la macro `\resetsysteme`.

Pour modifier les réglages pour un seul système, les `<clés>=<valeurs>` doivent se trouver dans l'argument optionnel de la macro `\systeme` :

```
\systeme[<clé1>=<valeur1> ,<clé2>=<valeur2>,etc]{code du système}
```

Remarque : lorsqu'une `<valeur>` contient une virgule, un crochet fermant ou un signe égal, cette `<valeur>` doit se trouver entre accolades afin que ces caractères spéciaux n'interfèrent pas avec la syntaxe d'autres commandes et ne provoque une erreur de compilation.

4 Tri des inconnues, la clé `sort`

4.1 Inconnues non indicées

Par défaut, les inconnues sont triées par ordre alphabétique, aussi bien dans le membre de droite que dans celui de gauche. On peut spécifier un tri avec la syntaxe

```
sort = {(consigne pour le membre de gauche) = (consigne pour le membre de droite)}
```

Une `<consigne>` peut être :

- « * », ce qui est la valeur par défaut ; dans ce cas, seules les inconnues entre `a` et `z` (en minuscules) sont prises en compte et sont triées dans l'ordre alphabétique ;
- un ordre de tri forcé indiqué par une liste d'inconnues séparées par des virgules (les espaces sont ignorés dans les consignes de tri), sachant que « 0 » représente le terme constant et si « 0 » est omis, le terme constant est placé en dernière position ;
- vide « {} » auquel cas, aucune inconnue ne sera reconnue et tout le membre sera considéré comme le terme constant.

La valeur par défaut de la clé `sort` est

```
sort = {*=*}
```

ce qui signifie que le tri par ordre alphabétique est actif dans les deux membres.

Tri forcé des inconnues

```
Tri alphabétique : \systeme{
  x+2y-3z+t=0,
  2x-y-z+3t=4,
  2y+3z+4t=-1,
  3x-2z-2t=3
}
```

$$\text{Tri alphabétique : } \begin{cases} t + x + 2y - 3z = 0 \\ 3t + 2x - y - z = 4 \\ 4t + 2y + 3z = -1 \\ -2t + 3x - 2z = 3 \end{cases}$$

```
Tri forcé : \systeme[sort={x, y, z, t = {} }]{
  x+2y-3z+t=0,
  2x-y-z+3t=4,
  2y+3z+4t=-1,
  3x-2z-2t=3
}
```

$$\text{Tri forcé : } \begin{cases} x + 2y - 3z + t = 0 \\ 2x - y - z + 3t = 4 \\ 2y + 3z + 4t = -1 \\ 3x - 2z - 2t = 3 \end{cases}$$

Toute lettre ne se trouvant pas dans la consigne de tri n'est pas considérée comme inconnue. C'est ici le cas avec m dans le membre de gauche :

Inconnues ignorées

```
\systeme[sort={x, y = m, n}]{
  mx-y=1+m^2+n,
  x-m^2y=-3+2m-3n^2
}
```

$$\begin{cases} mx - y = m^2 + n + 1 \\ x - m^2y = 2m - 3n^2 - 3 \end{cases}$$

On peut ainsi utiliser des inconnues en lettres majuscules ou grecques, qui ne sont pas autorisées par défaut :

Inconnues majuscules ou grecques

```
\systeme[sort={A, \alpha, \beta = {} }]{
  A-\alpha+3\beta=4,
  2A+\alpha-\beta=0,
  -A+3\alpha-2\beta=5
}
```

$$\begin{cases} A - \alpha + 3\beta = 4 \\ 2A + \alpha - \beta = 0 \\ -A + 3\alpha - 2\beta = 5 \end{cases}$$

Il est également possible de spécifier des inconnues constituées de plusieurs tokens dans une consigne de tri :

Inconnues comportant plusieurs tokens

```
\systeme[sort = {x, x', x'' = {} } ]{
  2x'-x''+3x=0,
  x'+2x''-x=-1,
  -3x'+x''+x=2
}\bigbreak
\systeme[sort = {x, x^2, a = a, a^2} ]{%
  2x^2-x+a-1=a^2-a,
  x-a=-2a^2+1,
  -x^2+3x+1=2a+3
}
```

$$\begin{cases} 3x + 2x' - x'' = 0 \\ -x + x' + 2x'' = -1 \\ x - 3x' + x'' = 2 \end{cases}$$

$$\begin{cases} -x + 2x^2 + a - 1 = -a + a^2 \\ x - a = -2a^2 + 1 \\ 3x - x^2 + 1 = 2a + 3 \end{cases}$$

On peut observer dans cet exemple les différentes places qu'occupe le terme constant dans le membre de gauche selon la place de « 0 » dans la consigne de tri :

Consigne de tri du membre constant

```
\systeme[sort = {x, y = {} }]{% par défaut
  2x+y-1=0,
  x-3y+2-a=1
}
\bigbreak
\systeme[sort = {x, 0, y = {} }]{%
  2x+y-1=0,
  x-3y+2-a=1
}
\bigbreak
\systeme[sort = {0, x, y = {} }]{%
  2x+y-1=0,
  x-3y+2-a=1
}
```

$$\begin{cases} 2x + y - 1 = 0 \\ x - 3y + 2 - a = 1 \end{cases}$$

$$\begin{cases} 2x - 1 + y = 0 \\ x + 2 - a - 3y = 1 \end{cases}$$

$$\begin{cases} -1 + 2x + y = 0 \\ 2 - a + x - 3y = 1 \end{cases}$$

4.2 Inconnues indicées

On peut indiquer une liste d'inconnues indicées dans la consigne de tri. Dans cet exemple, seules a_1 et a_3 sont considérées comme inconnues et on voit que a_2 est considéré comme terme constant

Inconnues indicées

```
\systeme[ sort = {a_1, a_3={}} ]{
  2a_1-3a_2+a_3+4=-1,
  a_1-a_2-3a_3=5
}
```

$$\begin{cases} 2a_1 + a_3 - 3a_2 + 4 = -1 \\ a_1 - 3a_3 - a_2 = 5 \end{cases}$$

Il est également possible d'indiquer dans la consigne de tri des indices non numériques :

Indices non numériques

```
\systeme[ sort = {a_n, a_{n+1}={}} ]{
  2a_n-3a_{n+1}-a_0+1=5,
  a_n-a_{n+1}+4=2
}
```

$$\begin{cases} 2a_n - 3a_{n+1} - a_0 + 1 = 5 \\ a_n - a_{n+1} + 4 = 2 \end{cases}$$

On peut également demander la consigne de tri x_{*} pour indiquer que x seul ou muni d'un indice (numérique ou non) doit être considéré comme inconnue. L'astérisque *doit* être entre accolades. Dans le système final, l'ordre de tri des inconnues est celui dans lequel elles apparaissent dans le code.

Indices quelconques

```
\systeme[ sort = {x_{*}}={}} ]{
  x+x_1-x_n-c=3,
  x_0-3x_1+2x_n+1=0,
  x_n-x_0+2x_1-1=0
}
```

$$\begin{cases} x + x_1 - x_n - c = 3 \\ -3x_1 + 2x_n + x_0 + 1 = 0 \\ 2x_1 + x_n - x_0 - 1 = 0 \end{cases}$$

Pour indiquer une inconnue comportant plusieurs tokens, il est nécessaire de mettre tous les tokens constituant l'inconnue avant le token « $_$ ». On doit donc écrire x^2_1 et non pas x_1^2 qui serait compris comme l'inconnue x_1 suivi de 2 et donc indûment placé dans la colonne des x_1 :

Indices en dernier !

```
\systeme[ sort = {x_1,x^2_1 = {} } ]{
  x^2_1+x_1+1=0,
  x_1^2-2=0% x_1^2 placé dans la colonne des x_1 !
}
```

$$\begin{cases} x_1 + x_1^2 + 1 = 0 \\ x_1^2 - 2 = 0 \end{cases}$$

5 Les autres clés et valeurs

Toutes les clés et pour chacune, la valeur par défaut ainsi qu'une brève description de leur action se trouvent ci-après.

5.1 Séparateur entre équations, la clé `eq sep`

Cette clé permet de choisir le séparateur entre les équations. Sa valeur par défaut est

```
eq sep = { , }
```

Pour écrire des coefficients décimaux, il convient de redéfinir ce séparateur :

Séparateur entre équations

```
\systeme[eq sep = \\ ]{
  1,5x-0,45y=0,7\\
  x-0,8y=1,4
}
```

$$\begin{cases} 1,5x - 0,45y = 0,7 \\ x - 0,8y = 1,4 \end{cases}$$

5.2 Retour charriot entre équations, la clé `cr eq sep`

Afin que la saisie soit la plus naturelle possible, la clé booléenne `cr eq sep` définie par défaut à

```
cr eq sep = false
```

permet, si elle est `true` de faire du retour charriot le séparateur entre équations, quelle que soit la valeur de la clé `eq sep`.

Dans ce cas :

- si l'on veut placer la macro `\systeme` dans l'argument d'une macro, il est *impératif* de prendre des précautions avec les macros `\SYSmakecrother` et `\SYSrestorecr` qui délimitent un environnement spécial ou bien avec la macro `\SYSallowcr` ;
- la clé `ignore empty eq` (voir ci-dessous) est mise à `true` pour le système concerné.

Retour charriot entre équations

```
\systeme[cr eq sep = true]{
  2a-c=-1
  a-2b+c=4
  -3a-b+2c=-1
}\par
\SYSmakecrother% <- commentaire nécessaire
\inoval{%
  \systeme[cr eq sep = true]{
    a-2b=-1
    -3a+b=-1
  }%
}%
\SYSrestorecr\par
\SYSallowcr\inoval{%
  \systeme[cr eq sep = true]{
    a-2b=-1
    -3a+b=-1
  }%
}
```

$$\begin{cases} 2a - c = -1 \\ a - 2b + c = 4 \\ -3a - b + 2c = -1 \end{cases}$$

$$\begin{cases} a - 2b = -1 \\ -3a + b = -1 \end{cases}$$

$$\begin{cases} a - 2b = -1 \\ -3a + b = -1 \end{cases}$$

Il est important de noter qu'entre `\SYSmakecrother` et `\SYSrestorecr`, toutes les fins de lignes n'étant pas dans l'argument de la macro `\systeme` doivent être commentées (même celles se trouvant après une macro). L'utilisation de la macro `\SYSallowcr` est plus simple de ce point de vue :

```
\SYSallowcr\macro{\systeme[cr eq sep = true]{(code du système)}}
\SYSallowcr{\macro[(arg optionnels)]{\systeme[cr eq sep = true]{(code du système)}}
```

5.3 Équations vides ignorées, la clé `ignore empty eq`

Cette clé vaut par défaut

```
ignore empty eq = true
```

Elle peut être réglée à `false` pour tenir compte des équations vides qui alors, produisent une ligne blanche.

Équations vides

```
\systeme{
  a+b+c=1,
  a-b+c=-2,
  ,
  a-b-c=3
}\qqquad
\systeme[ignore empty eq = false]{
  a+b+c=1,
  a-b+c=-2,
  ,
  a-b-c=3
}
```

$$\begin{cases} a + b + c = 1 \\ a - b + c = -2 \\ a - b - c = 3 \end{cases}$$

$$\begin{cases} a + b + c = 1 \\ a - b + c = -2 \\ a - b - c = 3 \end{cases}$$

5.4 Choix des délimiteurs, la clé `delim`

Cette clé définit les séparateurs gauche et droit du système. Ces délimiteurs sont mis en argument des primitives `\left` et `\right`. La valeur par défaut est

```
delim = { \{ , . }
```

ce qui place une accolade à gauche du système et rien à droite.

Choix des délimiteurs

<code>\systeme[delim={[,]}]{2a+3b=5,3a-b=8}\quad</code>	$\left[\begin{array}{l} 2a + 3b = 5 \\ 3a - b = 8 \end{array} \right]$	$\left. \begin{array}{l} 2a + 3b = 5 \\ 3a - b = 8 \end{array} \right $	$\left(\begin{array}{l} 2a + 3b = 5 \\ 3a - b = 8 \end{array} \right)$
<code>\systeme[delim={.,}]{2a+3b=5,3a-b=8}\quad</code>			
<code>\systeme[delim={(,\rfloor)}]{2a+3b=5,3a-b=8}</code>			

5.5 Alignement des termes dans les colonnes, la clé `align`

Chaque terme est aligné au fer à droite dans le membre de gauche et au fer à gauche dans celui de droite. C'est ce que spécifie par défaut `align`. Pour chaque membre, on a le choix entre `r`, `c` ou `l`. La valeur par défaut est

```
align = { r , l }
```

Alignement des termes

<code>\systeme{3x-y=5a-b,x+4y=a+3b}</code> <code>\quad</code> (par défaut, droite/gauche)	$\left\{ \begin{array}{l} 3x - y = 5a - b \\ x + 4y = a + 3b \end{array} \right.$	(par défaut, droite/gauche)
<code>\systeme[align={l,l}]{3x-y=5a-b,x+4y=a+3b}</code> <code>\quad</code> (alignement gauche/gauche)	$\left\{ \begin{array}{l} 3x - y = 5a - b \\ x + 4y = a + 3b \end{array} \right.$	(alignement gauche/gauche)
<code>\systeme[align={r,r}]{3x-y=5a-b,x+4y=a+3b}</code> <code>\quad</code> (alignement droite/droite)	$\left\{ \begin{array}{l} 3x - y = 5a - b \\ x + 4y = a + 3b \end{array} \right.$	(alignement droite/droite)
<code>\systeme[align={l,r}]{3x-y=5a-b,x+4y=a+3b}</code> <code>\quad</code> (alignement gauche/droite)	$\left\{ \begin{array}{l} 3x - y = 5a - b \\ x + 4y = a + 3b \end{array} \right.$	(alignement gauche/droite)
<code>\systeme[align={c,c}]{3x-y=5a-b,x+4y=a+3b}</code> <code>\quad</code> (alignement centré/centré)	$\left\{ \begin{array}{l} 3x - y = 5a - b \\ x + 4y = a + 3b \end{array} \right.$	(alignement centré/centré)

5.6 Alignement des signes des termes constants, la clé `align const sign`

Par souci d'esthétique ou par goût personnel, on peut désirer que les signes des termes constants (lorsque la composition de fait au fer à droite) ne soient pas alignés entre-eux. Cette clé a pour valeur par défaut

```
align const sign = true
```

ce qui aligne toujours les signes des termes constants.

Il faut la mettre à `false` pour l'effet contraire

Alignement des signes des termes constants

<code>\systeme{</code> <code>2a-b+\sqrt{2}-1=0,</code> <code>a-2b+1=0</code> <code>}</code>	$\left\{ \begin{array}{l} 2a - b + \sqrt{2} - 1 = 0 \\ a - 2b + 1 = 0 \end{array} \right.$
<code>\systeme[align const sign=false]{</code> <code>2a-b+\sqrt{2}-1=0,</code> <code>a-2b+1=0</code> <code>}</code>	$\left\{ \begin{array}{l} 2a - b + \sqrt{2} - 1 = 0 \\ a - 2b + 1 = 0 \end{array} \right.$

5.7 Liste des séparateurs de membres, la clé `member sep list`

Cette clé définit tous les ensembles de tokens qui seront compris par `systeme` comme séparateur de membres. La valeur par défaut est

```
member sep list = { <= ,>= , = , < , > , \leq , \geq , \leqslant , \geqslant }
```

Il appartient à l'utilisateur, s'il en a le besoin, d'ajouter à cette liste d'autres possibilités (voir ci-dessous)

5.8 Ajout/suppression d'un séparateur de membres, les clés `add member sep` et `remove member sep`

La clé `add member sep` permet d'ajouter un signe séparateur de membres et `remove member sep` d'en supprimer un existant. Leurs valeurs par défaut sont

```
add member sep={}
remove member sep = {}
```

Ajout de signe d'égalité

```
\setssysteme{add member sep = \approx}
\systeme{
  x+y\approx-6,
  -2x+3y\approx1
}
```

$$\begin{cases} x + y \approx -6 \\ -2x + 3y \approx 1 \end{cases}$$

5.9 Substitution de séparateur de membres, la clé `member sep subst`

On spécifie avec cette clé une substitutions à faire entre les tokens « bruts » compris comme séparateur de membres (voir clé `member sep list`) et tokens servant à afficher le signe. Cette clé ajoute une substitution à chaque invocation selon la syntaxe

```
member sep subst = { <tokens bruts> ,<tokens pour affichage> }
```

Cette clé est exécutée 2 fois par défaut

```
member sep subst = { <= ,\leq },
member sep subst = { >= ,\geq }
```

5.10 Espacements : les clés `sign space`, `member sep space`, `line skip coeff` et `coeff space`

La clé `sign space` spécifie l'espace supplémentaire qui sera ajoutée de part et d'autre de chaque signe « + » ou « - » compris comme séparateur de termes.

La clé `member sep space` joue le même rôle pour les signes séparant les membres.

L'espacement vertical entre chaque équation est multiplié par le coefficient `line skip coeff`.

Les valeurs par défaut sont

```
sign space      = 0pt,
member sep space = 0pt,
line skip coeff = 1.25
```

Espacements

<p>Espacement entre lignes:\par <code>\systeme{x+y=200,x-2y=10}\qqquad</code> <code>\systeme[line skip coeff=1]{x+y=200,x-2y=10}\qqquad</code> <code>\systeme[line skip coeff=1.75]{x+y=200,x-2y=10}\bigbreak</code></p>	<p>Espacement entre lignes :</p> $\begin{cases} x + y = 200 \\ x - 2y = 10 \end{cases}$
<p>Espacement autour de «=» :\par <code>\systeme{x+y=200,x-2y=10}\qqquad</code> <code>\systeme[member sep space=2.5pt]{x+y=200,x-2y=10}\qqquad</code> <code>\systeme[member sep space=5pt]{x+y=200,x-2y=10}\bigbreak</code></p>	<p>Espacement autour de « = » :</p> $\begin{cases} x + y = 200 \\ x - 2y = 10 \end{cases}$
<p>Espacement autour de «\$+\$» et «\$-\$» :\par <code>\systeme{x+y=200,x-2y=10}\qqquad</code> <code>\systeme[sign space=2.5pt]{x+y=200,x-2y=10}\qqquad</code> <code>\systeme[sign space=5pt]{x+y=200,x-2y=10}</code></p>	<p>Espacement autour de « + » et « - » :</p> $\begin{cases} x + y = 200 \\ x - 2y = 10 \end{cases}$

Lorsqu'une inconnue est précédée d'un coefficient explicite (c'est-à-dire que cette inconnue ne suit pas un signe + ou -), la clé `coeff space` permet de définir une espace mathématique (donc impérativement exprimée en `mu`) *positive* qui sera inséré entre le coefficient et l'inconnue. Si la valeur spécifiée est vide, aucune espace n'est insérée. Sa valeur pas défaut est

```
coeff space = {}
```


Espace après le coefficient

```
\systeme{
  2x-3y=4a-1,
  -5x+y=3a+4
}
\systeme[coeff space = 2mu ]{
  2x-3y=4a-1,
  -5x+y=3a+4
}
```

$$\begin{cases} 2x - 3y = 4a - 1 \\ -5x + y = 3a + 4 \end{cases}$$

$$\begin{cases} 2x - 3y = 4a - 1 \\ -5x + y = 3a + 4 \end{cases}$$

5.11 Colonne supplémentaire, clés **extra col sign**, **extra col pre** et **extra col post**

Lorsqu'une équation comporte le signe « @ », tout ce qui se trouve à droite de ce caractère sera mis dans une colonne supplémentaire au fer à gauche qui se trouvera à droite du système, en dernière position.

Le signe « @ » peut être changé en un autre avec la clé **extra col sign** dont la valeur par défaut est

```
extra col sign = @
```

Colonne supplémentaire

```
\systeme{
  x+y=125@L_1,
  x-y=12@L_2
}
```

$$\begin{cases} x + y = 125 & L_1 \\ x - y = 12 & L_2 \end{cases}$$

Cette colonne supplémentaire n'est *pas* composée en mode mathématique, mais deux codes sont insérées au début et à la fin de cette colonne. Ils sont définis par les clés **extra col pre** et **extra col post** dont les valeurs par défaut sont

```
extra col pre = \kern1.5em$,
extra col post = $
```

ce qui signifie qu'une espace horizontale d'1.5em est insérée et le « \$ » entre en mode mathématique au début de la colonne. Enfin, pour le second argument, « \$ » termine le mode math à la fin de la colonne.

On va redéfinir ces deux codes pour ne pas se mettre en mode math et insérer une espace de 2.5em :

Personnalisation de la colonne supplémentaire

```
\systeme[
  extra col pre = \kern2.5em ,
  extra col post = {} ,
  extra col sign = | ]{
  x+y=125|somme des deux nombres,
  x-y=12 |différence des deux nombres
}
```

$$\begin{cases} x + y = 125 & \text{somme des deux nombres} \\ x - y = 12 & \text{différence des deux nombres} \end{cases}$$

5.12 Numérotation automatique, les clés **autonom code**, **autonom continue**

La colonne supplémentaire peut être utilisée pour y mettre une numérotation automatique. La première façon de procéder est d'indiquer le schéma que doit prendre cette numérotation automatique à la première ligne, sachant que le caractère « * » déclenche la mise en place de la numérotation automatique et sera remplacé par le numéro de la ligne dans le système. Les autres lignes, même si elles n'ont pas de colonne supplémentaire, porteront un numéro :

Numérotation automatique

```
\systeme{x+y-z=3@L_{*},
  2x+y+z=4,
  x-y+2z=0
}
```

$$\begin{cases} x + y - z = 3 & L_1 \\ 2x + y + z = 4 & L_2 \\ x - y + 2z = 0 & L_3 \end{cases}$$

Si on souhaite spécifier explicitement des colonnes supplémentaires aux lignes qui n'en ont pas, la numérotation automatique vient *avant* le contenu explicite des colonnes supplémentaires :

Numérotation automatique

<pre>\systeme{ x+y-z=3@L_{*}\quad, 2x+y+z=4, x-y+2z=0 } \systeme{ x+y-z=3@L'_{*}, 3x+2y=7@=L_1+L_2, 3x+y=6@=2L_1+L_3 }</pre>	$\begin{cases} x + y - z = 3 & L_1 \\ 2x + y + z = 4 & L_2 \\ x - y + 2z = 0 & L_3 \end{cases}$ $\begin{cases} x + y - z = 3 & L'_1 \\ 3x + 2y = 7 & L'_2 = L_1 + L_2 \\ 3x + y = 6 & L'_3 = 2L_1 + L_3 \end{cases}$
---	--

Il est ennuyeux que l'on ne puisse mettre « $L'_* = L_1$ » à la première ligne car la totalité de ce schéma et notamment la partie indésirable « L_1 » serait reproduite aux lignes suivantes. On peut donc spécifier le schéma de numérotation d'une deuxième façon, *avant* d'écrire le système. On utilise la clé `autonum code` pour définir ce schéma. Sa valeur par défaut est

`autonum code = {}`

Numérotation automatique

<pre>\systeme[autonum code = L_{*}] { x+y-z=3, 2x+y+z=4, x-y+2z=0 } \systeme[autonum code = L'_{*}]\longleftarrow { x+y-z=3@L_1, 3x+2y=7@L_1+L_2, 3x+y=6@2L_1+L_3 }</pre>	$\begin{cases} x + y - z = 3 & L_1 \\ 2x + y + z = 4 & L_2 \\ x - y + 2z = 0 & L_3 \end{cases}$ $\begin{cases} x + y - z = 3 & L'_1 \leftarrow L_1 \\ 3x + 2y = 7 & L'_2 \leftarrow L_1 + L_2 \\ 3x + y = 6 & L'_3 \leftarrow 2L_1 + L_3 \end{cases}$
--	---

Le schéma spécifié avec la clé `autonum code` n'est valable *que pour le prochain système* et sera effacé ensuite, sauf si on met à `true` la valeur de la clé `autonum continue`, auquel cas le schéma de numérotation se poursuit pour tous les systèmes à venir. La valeur par défaut est

`autonum continue = false`

Numérotation automatique persistante

<pre>\setssysteme{ autonum code = L_{*}, autonum continue = true} \systeme{a+b=4,2a-b=5} \quad \systeme{x-3y=0,2x+y=1}</pre>	$\begin{cases} a + b = 4 & L_1 \\ 2a - b = 5 & L_2 \end{cases} \quad \begin{cases} x - 3y = 0 & L_1 \\ 2x + y = 1 & L_2 \end{cases}$
--	--

On peut effacer (et donc annuler) le schéma de numérotation automatique en mettant à vide la valeur de la clé `autonum code` :

`autonum code = {}`

Il existe un compteur global¹ d'équations accessible de la même façon que le compteur local mais avec « `**` ». Voici un exemple de numérotation persistante :

Numérotation globale

<pre>On met en évidence que \number\SYSeqnum{} équations ont été écrites jusqu'ici~: \setssysteme{autonum code = \hbox{eq (**)}, autonum continue = true} \systeme{x-y+z=3, 2x+y+z=1, x-z=8} \systeme{u+w=9, v+2w=0, u-v=1}</pre>	<p>On met en évidence que 137 équations ont été écrites jusqu'ici :</p> $\begin{cases} x - y + z = 3 & \text{eq (138)} \\ 2x + y + z = 1 & \text{eq (139)} \\ x - z = 8 & \text{eq (140)} \\ u + w = 9 & \text{eq (141)} \\ v + 2w = 0 & \text{eq (142)} \\ u - v = 1 & \text{eq (143)} \end{cases}$
---	--

¹ Il s'agit d'un compteur \TeX portant le doux nom de `\SYSeqnum`, accessible via les commandes \TeX habituelles, c'est-à-dire qu'on peut le faire précéder de `\number`, `\romannumeral` pour l'afficher.

Il est enfin possible d'utiliser la colonne supplémentaire pour retenir le numéro d'une équation et le rappeler plus tard :

Numérotation globale

```
\setssysteme{
  extra col pre = \kern1em ,
  extra col post = {}
}
\systeme{
  x-y+z=3 @\gdef\foo{**}({\bf**}),
  2x+y+z=1@\gdef\bar{**}({\bf**}),
  x-z=8 @\gdef\baz{**}({\bf**})
}
```

$$\begin{cases} x - y + z = 3 & \text{(144)} \\ 2x + y + z = 1 & \text{(145)} \\ x - z = 8 & \text{(146)} \end{cases}$$

L'équation numéro 144 est la première du système, 145 et 146 sont les deux suivantes.

L'équation numéro \foo{} est la première du système, \bar{} et \baz{} sont les deux suivantes.

On peut à tout moment réinitialiser le compteur global d'équations à l'aide de la clé `main eq count` :

```
main eq count = <valeur entière>
```

L'assignation ainsi effectuée est globale.

5.13 Augmentation en hauteur et profondeur, la clé `extra height`

L'alignement constitué par le système peut avoir sa hauteur et sa profondeur augmentées de la valeur *positive* de la clé `extra height` dont la valeur par défaut est

```
extra height = 1.5pt
```

Cet ajustement permet dans certains cas que la taille des délimiteurs soit mieux adaptée à celle du système.

5.14 Non alignement des termes, la clé `align terms`

Cette clé, qui vaut `true` par défaut, permet, si elle est assignée à `false` de ne pas aligner les termes dans un même membre. Dans ce cas, seul les signes séparateurs de membres sont alignés.

Non alignement des termes

```
\systeme{
  x-3y=4a-1,
  -5x+y=a+4
}
\systeme[align terms=false]{
  x-3y=4a-1,
  -5x+y=a+4
}
```

$$\begin{cases} x - 3y = 4a - 1 \\ -5x + y = a + 4 \end{cases}$$

$$\begin{cases} x - 3y = 4a - 1 \\ -5x + y = a + 4 \end{cases}$$

5.15 Substitution post traitement, la clé `post subst`

Juste avant l'affichage du système, il est encore possible de substituer dans le code du système toute suite de token par une autre. On utilise pour cela la clé `post subst` :

```
post subst = {<x1> <y1> <x2> <y2>...<xn> <yn>}
```

ce qui a pour effet de remplacer toutes les occurrences de `<x1>` par `<y1>`, celles de `<x2>` par `<y2>`, etc. Un ou plusieurs espaces (ignorés) peuvent se trouver entre les motifs de substitution.

La valeur de cette clé est

```
post subst = {}
```

Dans cet exemple, on entre des coefficients décimaux où le point est le séparateur décimal. On va demander à ce que tous les « . » soient remplacés par des « , ». On remplacera aussi l'inconnue a_1 par x_n , et a_2 par x_{n+1} :

```
\systeme[post subst = {.}{.}, {a_1}{x_n} {a_2}{x_{n+1}} ]{
  1.5a_1-0.5a_2=2,
  1.6a_1-2a_2=0.4
}
```

$$\begin{cases} 1,5x_n - 0,5x_{n+1} = 2 \\ 1,6x_n - 2x_{n+1} = 0,4 \end{cases}$$

5.16 Sauvegarde du code produit dans une macro, la clé `store`

Pour sauvegarder le code résultant de la macro `systeme` dans une `macro` en vue de l'utiliser à sa guise plus tard, on utilise la clé `store`. Son contenu doit être soit vide (aucune sauvegarde n'est effectuée), soit une macro *seule* qui recevra le code. Toute autre valeur conduit à une erreur.

La valeur par défaut est

```
store = {}
```

Si la clé contient une `macro`, la commande `systeme` ne produit aucun affichage et sauvegarde la totalité du code qu'elle produit dans la `macro`. L'assignation est globale. Après avoir été utilisée par la commande `\systeme`, la clé `store` est réinitialisée à sa valeur par défaut.

Si l'on écrit

```
\systeme[store=\mysys]{3x-2y+z=-4,
                      x+5z=2,
                      -x+y-2z=0}
```

alors la macro `\mysys` contient le code suivant

```
\csname SYS_\ifmmode id\else enter_math\fi\endcsname{%
  \left\{%
    \vcenter{\lineskiplimit=0pt \lineskip=0pt \baselineskip1.25\normalbaselineskip
      \halign{%
        \hfil###$&%
        \hbox to12.22214pt{\hss###$\hss}&%
        \hfil###$&%
        \hbox to12.22214pt{\hss###$\hss}&%
        \hfil###$&%
        \hfil###$\hfil&%
        $$$\hfil
      \cr
      \vrule depth0pt width0pt height9.0pt%
      3x&-&2y&+&z&{ }={ }&-4%
      \cr
      x&&&+&5z&{ }={ }&2%
      \cr
      -x&+&y&-&2z&{ }={ }&0%
      \vrule height0pt width0pt depth4.0pt%
      \cr
    }%
  }\right.%
}
```

5.17 Code exécuté avant l'alignement, la clé `code before`

Juste avant que le `\halign` n'entre en action, la valeur de la clé `code before` est exécutée. Elle peut être utile pour exécuter un code arbitraire en vue, par exemple, de faire certains réglages mathématiques effectués par `\mathcode`, `\delcode` ou autre. Par défaut, la valeur² de cette clé est

```
code before = \mathcode'\,'013B
```

ce qui donne à la virgule des espacements corrects lorsqu'elle est utilisée comme délimiteur décimal comme c'est l'usage en France.

² Dans ce manuel, composé avec OpTeX, la valeur du `\mathcode` de la virgule a été réglée à `"012C` avec le code

```
\setssysteme{code before = {\mathcode'\,'012C}} ».
```

6 Exemples

La macro `\systeme` peut être un peu détournée de ses fins pour écrire sans contrainte des calculs en ligne, où le signe `=` est répété au début de chaque ligne de calcul.

Certains réglages sont cependant nécessaires (délimiteurs désactivés, pas d'alignement des termes, alignement des colonnes, interdiction du tri automatique) :

Calculs en ligne

<pre>\systeme[delim={.,.}, align terms=false, align={r,l}, sort={{}}={}, cr eq sep = true]{ A=(2x-1)^2-(x-2)^2 =4x^2-4x+1-(x^2-4x+4) =4x^2-4x+1-x^2+4x-4 =3x^2-3 =3(x^2-1) =3(x-1)(x+1) }</pre>	$ \begin{aligned} A &= (2x - 1)^2 - (x - 2)^2 \\ &= 4x^2 - 4x + 1 - (x^2 - 4x + 4) \\ &= 4x^2 - 4x + 1 - x^2 + 4x - 4 \\ &= 3x^2 - 3 \\ &= 3(x^2 - 1) \\ &= 3(x - 1)(x + 1) \end{aligned} $
---	--

La macro `\aligncalc`, intégrée dans le package `systeme`, est programmée de la façon suivante

```
\def\aligncalc{%
  \systeme[
    delim={.,.},
    align terms=false,
    align={r,l},
    sort={{}}={},
    extra col pre=\kern1.5em,
    extra col post={}
  ]}
}
```

Ce qui permet d'alléger la syntaxe.

Calculs en ligne

<pre>\aligncalc{ D=x^2+4x-21, =x^2+4x+4-4-21 @faire apparaitre \$x^2+4x+4\$, =(x+2)^2-25 @qui se factorise en \$(x+2)^2\$, =(x+2)^2-5^2 @différence de 2 carrés, =(x+2+5)(x+2-5) @factorisation, =(x+7)(x-3) @résultat final }</pre>	$ \begin{aligned} D &= x^2 + 4x - 21 \\ &= x^2 + 4x + 4 - 4 - 21 && \text{faire apparaitre } x^2 + 4x + 4 \\ &= (x + 2)^2 - 25 && \text{qui se factorise en } (x + 2)^2 \\ &= (x + 2)^2 - 5^2 && \text{différence de 2 carrés} \\ &= (x + 2 + 5)(x + 2 - 5) && \text{factorisation} \\ &= (x + 7)(x - 3) && \text{résultat final} \end{aligned} $
--	--

Les équations sans signe `=` et sans second membre sont également acceptées.

Lignes sans second membre

<pre>\systeme{ -2a+b-c, a-5c, 3a-2b+3c }</pre>	$ \left\{ \begin{array}{l} -2a + b - c \\ a - 5c \\ 3a - 2b + 3c \end{array} \right. $
--	--

On peut ainsi, moyennant quelques réglages, mimer l'environnement `cases` de \LaTeX .

```
% https://tex.stackexchange.com/questions/366500
$ns=\systeme[
  align terms=false,
  sort={{}}={},
  align={l,l},
  extra col pre=\kern2em\hfill,
  extra col post={}
]{
s+s+\ldots+s      @($n$ fois) si $n>0$,
0                  @si $n=0$,
(-s)+(-s)+\ldots+(-s) @($|n|$ fois) si $n<0$%
}$
```

$$ns = \begin{cases} s + s + \dots + s & (n \text{ fois}) \text{ si } n > 0 \\ 0 & \text{si } n = 0 \\ (-s) + (-s) + \dots + (-s) & (|n| \text{ fois}) \text{ si } n < 0 \end{cases}$$

Charge à l'utilisateur de programmer une macro qui lui est spécifique pour alléger la syntaxe.

Voici des réponses valides à quelques questions posées sur le désormais moribond site StackExchange.

```
% https://tex.stackexchange.com/questions/679847
\systeme[sort={x,y={}},align const sign=false]{
  x + y + 2z = a,
  2x + 2y + {( 4+b )} z = b,
  -x -2z = ab,
  y + bz = a + b
}
```

$$\begin{cases} x + y + 2z = a \\ 2x + 2y + (4 + b)z = b \\ -x - 2z = ab \\ y + bz = a + b \end{cases}$$

```
% https://tex.stackexchange.com/questions/685384
\systeme[sort={{x_1,x_2,x_3}}]{
  x_1'(t) = x_1(t)-x_2(t)+2x_3(t),
  x_2'(t) = -2x_1(t)+5x_2(t)+2x_3(t),
  x_3'(t) = 8x_1(t)-5x_2(t)-2x_3(t)
}
```

$$\begin{cases} x_1'(t) = x_1(t) - x_2(t) + 2x_3(t) \\ x_2'(t) = -2x_1(t) + 5x_2(t) + 2x_3(t) \\ x_3'(t) = 8x_1(t) - 5x_2(t) - 2x_3(t) \end{cases}$$

```
% https://tex.stackexchange.com/questions/696681
\systeme[sort={x,y,z={}} ]{
  a_1 x+b_1 y+c_1 z=d_1,
  a_2 x+b_2 y+c_2 z=d_2,
  a_3 x+b_3 y+c_3 z=d_3
}
```

$$\begin{cases} a_1x + b_1y + c_1z = d_1 \\ a_2x + b_2y + c_2z = d_2 \\ a_3x + b_3y + c_3z = d_3 \end{cases}$$

```
% https://tex.stackexchange.com/questions/728852
\systeme[sort = {\alpha,\beta,\gamma=} ]{
  2 \sin \alpha - \cos \beta + 3\tan \gamma = 3,
  4 \sin \alpha + 2\cos \beta - 2\tan \gamma = 2,
  6 \sin \alpha - 3\cos \beta + \tan \gamma = 9
}
```

$$\begin{cases} 2 \sin \alpha - \cos \beta + 3 \tan \gamma = 3 \\ 4 \sin \alpha + 2 \cos \beta - 2 \tan \gamma = 2 \\ 6 \sin \alpha - 3 \cos \beta + \tan \gamma = 9 \end{cases}$$

```
% https://tex.stackexchange.com/questions/692368
\systeme[ sort = {x_i , x_2 = {} } ]{
  x_{i}=x_0,
  x_{2}=x_1
}
```

$$\begin{cases} x_i = x_0 \\ x_2 = x_1 \end{cases}$$

```
% https://tex.stackexchange.com/questions/475993/
\systeme[sort={ 0, s, t = {} } ]{
  3+2t+s=x,
  s+2=y,
  1+t+2s=z
}
```

$$\begin{cases} 3 + s + 2t = x \\ 2 + s = y \\ 1 + 2s + t = z \end{cases}$$

★
★ ★

J'espère que ce package vous sera utile !

Merci de me signaler par [email](#) tout bug ou toute proposition d'amélioration.