

MOOC de Introducción a sage. Integración en LaTeX

Leandro Marín

- 1 Introduciendo Objetos en \LaTeX
- 2 Introduciendo Texto y Código \LaTeX
- 3 Gráficos

Introducción

- A diferencia de otros temas en los que no se consideraban muchos requerimientos previos, en este presupondremos un cierto conocimiento de \LaTeX .

Introducción

- A diferencia de otros temas en los que no se consideraban muchos requerimientos previos, en este presupondremos un cierto conocimiento de \LaTeX .
- En la redacción de documentos matemáticos suele ser habitual introducir resultados de cálculos o incluso texto generado a partir de algún algoritmo sencillo.

Introducción

- A diferencia de otros temas en los que no se consideraban muchos requerimientos previos, en este presupondremos un cierto conocimiento de \LaTeX .
- En la redacción de documentos matemáticos suele ser habitual introducir resultados de cálculos o incluso texto generado a partir de algún algoritmo sencillo.
- Toda esa información es mejor introducirla directamente en el código fuente de \LaTeX y recalcularla, lo cual hace más cómodo hacer cualquier tipo de cambio y nos mantiene los datos actualizados.

Las Cabeceras

- Supondremos que tenemos dentro de nuestro sistema una distribución de LaTeX y también una distribución de sage.

Las Cabeceras

- Supondremos que tenemos dentro de nuestro sistema una distribución de LaTeX y también una distribución de sage.
- Lo primero que necesitamos es que LaTeX y sage se puedan comunicar. Para ello nuestro archivo LaTeX debe usar el fichero de cabecera `sagetex.sty`.

Las Cabeceras

- Supondremos que tenemos dentro de nuestro sistema una distribución de LaTeX y también una distribución de sage.
- Lo primero que necesitamos es que LaTeX y sage se puedan comunicar. Para ello nuestro archivo LaTeX debe usar el fichero de cabecera `sagetex.sty`.
- Este fichero viene en la distribución de sage y debe ser accesible para LaTeX. Esto se puede hacer controlando las variables de entorno de LaTeX tal y como se puede ver en <http://doc.sagemath.org/html/en/tutorial/sagetex.html>.

Las Cabeceras

- Supondremos que tenemos dentro de nuestro sistema una distribución de LaTeX y también una distribución de sage.
- Lo primero que necesitamos es que LaTeX y sage se puedan comunicar. Para ello nuestro archivo LaTeX debe usar el fichero de cabecera `sagetex.sty`.
- Este fichero viene en la distribución de sage y debe ser accesible para LaTeX. Esto se puede hacer controlando las variables de entorno de LaTeX tal y como se puede ver en <http://doc.sagemath.org/html/en/tutorial/sagetex.html>.
- En cualquier caso, la forma más sencilla puede ser copiar el archivo `sagetex.sty` de nuestra distribución al directorio actual donde estamos escribiendo nuestro archivo LaTeX.

Las Cabeceras

- Supondremos que tenemos dentro de nuestro sistema una distribución de LaTeX y también una distribución de sage.
- Lo primero que necesitamos es que LaTeX y sage se puedan comunicar. Para ello nuestro archivo LaTeX debe usar el fichero de cabecera `sagetex.sty`.
- Este fichero viene en la distribución de sage y debe ser accesible para LaTeX. Esto se puede hacer controlando las variables de entorno de LaTeX tal y como se puede ver en <http://doc.sagemath.org/html/en/tutorial/sagetex.html>.
- En cualquier caso, la forma más sencilla puede ser copiar el archivo `sagetex.sty` de nuestra distribución al directorio actual donde estamos escribiendo nuestro archivo LaTeX.
- Finalmente, debemos poner `\usepackage{sagetex}` entre las cabeceras de nuestro archivo LaTeX.

Haciendo Cálculos

- Para poder introducir un cálculo hecho en sage dentro de LaTeX empezaremos viendo cómo asignar las variables y hacer los cálculos. Eso se hace con el entorno `sagesilent`.

Haciendo Cálculos

- Para poder introducir un cálculo hecho en sage dentro de LaTeX empezaremos viendo cómo asignar las variables y hacer los cálculos. Eso se hace con el entorno `sagesilent`.
- Entre el `\begin{document}` y el `\end{document}` de nuestro archivo LaTeX debemos incluir uno o varios entornos `sagesilent` en los que escribiremos código sage exactamente igual que en un programa sage normal.

Esto es un texto LaTeX que queremos escribir. Lo que escribamos dentro del entorno `{\tt sagesilent}` no será visible al compilar.

```
\begin{sagesilent}
a = factorial(30)
\end{sagesilent}
```

Este texto continuará al anterior como si no hubiéramos incluido ningún código `{\tt sage}`.

Compilando

- Cuando introducimos código sage dentro del fichero LaTeX tratamos de compilar con el comando `pdflatex` `fichero.tex` se nos generará un archivo llamado `fichero.sagetex.sage`.

Compilando

- Cuando introducimos código sage dentro del fichero \LaTeX tratamos de compilar con el comando `pdflatex fichero.tex` se nos generará un archivo llamado `fichero.sagetex.sage`.
- Ese fichero contiene todos los cálculos y debemos ejecutarlo en sage con el comando `sage fichero.sagetex.sage`

Compilando

- Cuando introducimos código sage dentro del fichero \LaTeX tratamos de compilar con el comando `pdflatex fichero.tex` se nos generará un archivo llamado `fichero.sagetex.sage`.
- Ese fichero contiene todos los cálculos y debemos ejecutarlo en sage con el comando `sage fichero.sagetex.sage`
- Una vez hecho esto, debemos compilar de nuevo con `pdflatex fichero.tex` y todos los cálculos entrarán dentro de nuestro pdf.

Funcionamiento

- La primera vez que ejecutamos `pdflatex fichero.tex` se extraerá del fichero fuente todo el código sage que hayamos introducido y se crearán referencias a todos los valores que deben ser introducidos de nuevo en el fichero LaTeX.

Funcionamiento

- La primera vez que ejecutamos `pdflatex fichero.tex` se extraerá del fichero fuente todo el código sage que hayamos introducido y se crearán referencias a todos los valores que deben ser introducidos de nuevo en el fichero LaTeX.
- La ejecución de `sage fichero.sagetex.sage` realizará los cálculos y asignará las referencias.

Funcionamiento

- La primera vez que ejecutamos `pdflatex fichero.tex` se extraerá del fichero fuente todo el código sage que hayamos introducido y se crearán referencias a todos los valores que deben ser introducidos de nuevo en el fichero LaTeX.
- La ejecución de `sage fichero.sagetex.sage` realizará los cálculos y asignará las referencias.
- Una vez calculadas todas esas referencias, la última ejecución de `pdflatex fichero.tex` nos las pondrá dentro del texto para que puedan ser visualizadas en el pdf.

El comando sage.

- La forma mas directa de introducir un valor calculado en sage dentro del código LaTeX es el comando sage.

El comando sage.

- La forma mas directa de introducir un valor calculado en sage dentro del código LaTeX es el comando `sage`.
- Por ejemplo, si en un entorno `sagesilent` hemos calculado la variable `a` a podemos introducirla dentro del código LaTeX como sigue:

```
\begin{sagesilent}
a = factorial(30)
\end{sagesilent}
```

La complejidad del algoritmo es alta puesto que este planteamiento nos llevaría a un número de ejecuciones igual al factorial de 100 que es $\text{sage}\{a\}$, claramente fuera de nuestro alcance.

El comando sage.

- La forma mas directa de introducir un valor calculado en sage dentro del código LaTeX es el comando `sage`.
- Por ejemplo, si en un entorno `sagesilent` hemos calculado la variable `a` a podemos introducirla dentro del código LaTeX como sigue:

```
\begin{sagesilent}
a = factorial(30)
\end{sagesilent}
```

La complejidad del algoritmo es alta puesto que este planteamiento nos llevaría a un número de ejecuciones igual al factorial de 30 que es $265252859812191058636308480000000$, claramente fuera de nuestro alcance.

- En el texto final nos aparecerá:
La complejidad del algoritmo es alta puesto que este planteamiento nos llevará a un número de ejecuciones igual al factorial de 30 que es $265252859812191058636308480000000$, claramente fuera de nuestro alcance.

Comentario sobre `sagesilent`

- Aunque anteriormente hemos comentado que debemos poner un entorno `sagesilent` para hacer los cálculos en sage y eso es lo normal no es del todo cierto.

Comentario sobre sagesilent

- Aunque anteriormente hemos comentado que debemos poner un entorno `sagesilent` para hacer los cálculos en sage y eso es lo normal no es del todo cierto.
- No existe ningún problema en poner el comando `\sage{factorial(30)}` dentro del texto LaTeX y hacer el cálculo ahí mismo.

Comentario sobre sagesilent

- Aunque anteriormente hemos comentado que debemos poner un entorno `sagesilent` para hacer los cálculos en sage y eso es lo normal no es del todo cierto.
- No existe ningún problema en poner el comando `\sage{factorial(30)}` dentro del texto LaTeX y hacer el cálculo ahí mismo.
- Yo suelo poner todo el código fuera y poner el comando `sage` sólo con variables calculadas, pero no es estrictamente necesario.

Objetos Matemáticos Complejos

- El comando `sage` no solo nos permite introducir números. Podemos introducir valores más complejos.

Objetos Matemáticos Complejos

- El comando `sage` no solo nos permite introducir números. Podemos introducir valores más complejos.
- Por ejemplo:

```
\begin{sagesilent}  
M = matrix(ZZ, [[1,2,3],[4,5,6]])  
\end{sagesilent}
```

La matrix `\sage{M}` no es una matriz cuadrada.

Objetos Matemáticos Complejos

- El comando `sage` no solo nos permite introducir números. Podemos introducir valores más complejos.

- Por ejemplo:

```
\begin{sagesilent}
M = matrix(ZZ, [[1,2,3], [4,5,6]])
\end{sagesilent}
```

La matrix `\sage{M}` no es una matriz cuadrada.

- Nos producirá la salida:

La matrix $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ no es una matriz cuadrada.

Funciones y cálculos complejos

- Dentro del entorno `sagesilent` podemos introducir todo tipo de código sage que puede llegar a ser muy complejo.

Funciones y cálculos complejos

- Dentro del entorno `sagesilent` podemos introducir todo tipo de código sage que puede llegar a ser muy complejo.
- Podemos definir funciones, objetos o hacer cálculos complejos.

El comando `sagestr`

- Aunque el comando `sage` nos permite meter objetos en el código LaTeX no es la forma más versátil de hacerlo.

El comando `sagestr`

- Aunque el comando `sage` nos permite meter objetos en el código LaTeX no es la forma más versátil de hacerlo.
- Estamos ante un lenguaje de programación que nos permite hacer muchísimas cosas y en particular manipular cadenas de texto a nuestra conveniencia. Esas cadenas de texto pueden contener a su vez código LaTeX resultado de nuestro programa.

El comando `sagestr`

- Aunque el comando `sage` nos permite meter objetos en el código LaTeX no es la forma más versátil de hacerlo.
- Estamos ante un lenguaje de programación que nos permite hacer muchísimas cosas y en particular manipular cadenas de texto a nuestra conveniencia. Esas cadenas de texto pueden contener a su vez código LaTeX resultado de nuestro programa.
- Esas cadenas se pueden introducir junto con el resto de nuestro código LaTeX para ser procesado por el comando `pdflatex`.

Un Ejemplo

- Vamos a desarrollar un ejemplo en varios pasos.

Un Ejemplo

- Vamos a desarrollar un ejemplo en varios pasos.
- Supongamos que somos profesores y queremos generar una lista de ejercicios personalizados para nuestros alumnos.

Un Ejemplo

- Vamos a desarrollar un ejemplo en varios pasos.
- Supongamos que somos profesores y queremos generar una lista de ejercicios personalizados para nuestros alumnos.
- Estos ejercicios consistirán en resolver una ecuación de segundo grado, pero queremos que los valores sean números enteros.

Generando los Ejercicios

- Para generar las ecuaciones tomaremos dos valores enteros aleatorios a y b , calcularemos $p(x) = (x - a)(x - b)$ y expandiremos el polinomio.

Generando los Ejercicios

- Para generar las ecuaciones tomaremos dos valores enteros aleatorios a y b , calcularemos $p(x) = (x - a)(x - b)$ y expandiremos el polinomio.
- Para cada alumno de una lista generaremos una página con su nombre centrado y el enunciado del problema.

Generando los Ejercicios

- Para generar las ecuaciones tomaremos dos valores enteros aleatorios a y b , calcularemos $p(x) = (x - a)(x - b)$ y expandiremos el polinomio.
- Para cada alumno de una lista generaremos una página con su nombre centrado y el enunciado del problema.
- Una vez generado el texto, lo introduciremos dentro del archivo LaTeX.

```

\documentclass{amsart}
\usepackage{sagetex}
\begin{document}
\begin{sagesilent}
L = []
var('x')
for nombre in ["Antonio","Luisa","Pedro"]:
    L.append(r"\begin{center} " + nombre + r"\end{center}")
    a = ZZ.random_element()
    b = ZZ.random_element()
    p(x) = expand((x-a)*(x-b))
    L.append(r"Resuelve:  $\{0\} = 0$ ".format(latex(p(x))))
    L.append(r"\newpage")
texto = "\n".join(L)
\end{sagesilent}

\sagestr{texto}

\end{document}

```

Complicando el Código

- Este ejemplo sencillo podemos complicarlo añadiendo por ejemplo un control de que los valores a y b estén dentro de unos límites o que no sean 0.

Complicando el Código

- Este ejemplo sencillo podemos complicarlo añadiendo por ejemplo un control de que los valores a y b estén dentro de unos límites o que no sean 0.
- También podemos manejar ficheros y que el código introducido lea los nombres de los alumnos de un fichero externo.

Complicando el Código

- Este ejemplo sencillo podemos complicarlo añadiendo por ejemplo un control de que los valores a y b estén dentro de unos límites o que no sean 0.
- También podemos manejar ficheros y que el código introducido lea los nombres de los alumnos de un fichero externo.
- Todo lo que nos permita hacer sage puede estar incluido en nuestra generación de texto, el límite es nuestra imaginación.

El comando `sageplot`

- Aparte de valores numéricos o cadenas de texto generadas, un recurso que suele ser muy interesante es el de los gráficos.

El comando `sageplot`

- Aparte de valores numéricos o cadenas de texto generadas, un recurso que suele ser muy interesante es el de los gráficos.
- Para introducir un gráfico en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ generado en sage utilizaremos el comando `sageplot`.

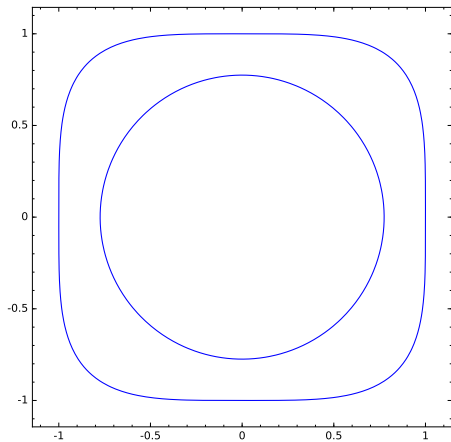
El comando `sageplot`

- Aparte de valores numéricos o cadenas de texto generadas, un recurso que suele ser muy interesante es el de los gráficos.
- Para introducir un gráfico en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ generado en sage utilizaremos el comando `sageplot`.
- Veamos un ejemplo.

```
\begin{sagesilent}
var('x,y')
A=implicit_plot(x^4+y^4-1,[x,-1.1,1.1],[y,-1.1,1.1])
B=implicit_plot(x^2+y^2-0.6,[x,-1.1,1.1],[y,-1.1,1.1])
\end{sagesilent}

\sageplot[scale=0.4]{A+B}
```

TituloFrame



Conclusión

- La combinación de sage con \LaTeX nos proporciona un entorno de trabajo de una potencia enorme y gran flexibilidad.

Conclusión

- La combinación de sage con \LaTeX nos proporciona un entorno de trabajo de una potencia enorme y gran flexibilidad.
- Podemos generar código o calcular valores y gráficos e introducirlos en nuestro fichero \LaTeX fácilmente.

Conclusión

- La combinación de sage con \LaTeX nos proporciona un entorno de trabajo de una potencia enorme y gran flexibilidad.
- Podemos generar código o calcular valores y gráficos e introducirlos en nuestro fichero \LaTeX fácilmente.
- De esta forma se puede ahorrar gran cantidad de tiempo en la generación de documentos complejos que se basan en una reglas siempre que estas se puedan programar.