

# Practica lección 2 — polinomios y series formales

Marcos Bujosa

## Polinomios como series formales finitas

El siguiente bloque de código es para que este cuaderno electrónico (notebook) cargue mi implementación de algunos objetos (no es una buena implementación, pero es suficiente para ilustrar algunas ideas). Puede consultar el código de mi [implementacion\\_series\\_formales.py](#) en python con la definición las clases `SerieConPrincipio` y `SerieConFinal`:

```
%run -i src/implementacion_series_formales.py
```

- Creación del directorio auxiliar para albergar las figuras de la lección Para publicar la lección como pdf o página web, necesito los gráficos como ficheros `.png` alojados algún directorio específico:

```
imagenes_leccion = "../img/P-L02-A-polinomios_y_series_formales" # directorio para las imágenes de la lección
import os
os.makedirs(imagenes_leccion, exist_ok=True) # crea el directorio si no existe
```

## Polinomios como series formales

Recuerde que un polinomio tiene cogrado no negativo; por tanto la siguiente serie con principio y cogrado 0 es un polinomio:

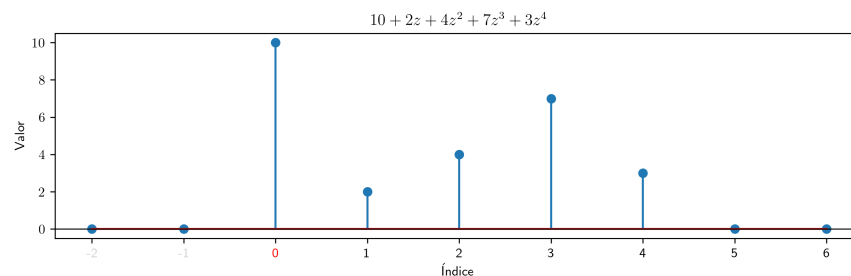
```
a=SerieConPrincipio([10,2,4,7,3,0,0],0)
pinta(a)
```

Para ilustrar el producto por convolución también usaremos esta representación en forma de escalera tumbada y donde se destaca el coeficiente situado en la posición 0.

```
a.RepEscalera=True
a
```

También usaremos representaciones gráfica de este tipo:

```
a.plot_serie(indices_previos_al_cogrado=2, title='$'+latex(a)+'$', savefig(imagenes_leccion + '/polinomioA.png', dpi=300, bbox_inches='tight'))
```



La condición para que una secuencia sea un polinomio es que sea una serie formal (que su *cogrado* no sea negativo) y que además tenga final (que tenga *grado* finito); consecuentemente, un polinomio solo tiene un número finito de coeficientes no nulos.

Por tanto, también es un polinomio la siguiente serie con principio, finita y con cogrado 4:

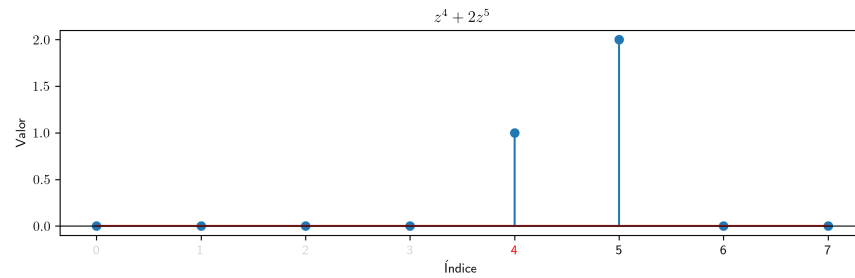
---

```

b = SerieConPrincipio([1,2,0,0],4)
pinta(b)
b.plot_serie(indices_previos_al_cogrado=4, title='$'+latex(b)+'$').savefig(imagenes_leccion + '/polinomioB.png', dpi=300, bbox_inches='tight')
b.RepEscalera=True
b

```

---



## No toda secuencia finita es un polinomio

Sin embargo, esta serie con principio y cogrado  $-1$  NO es un polinomio:

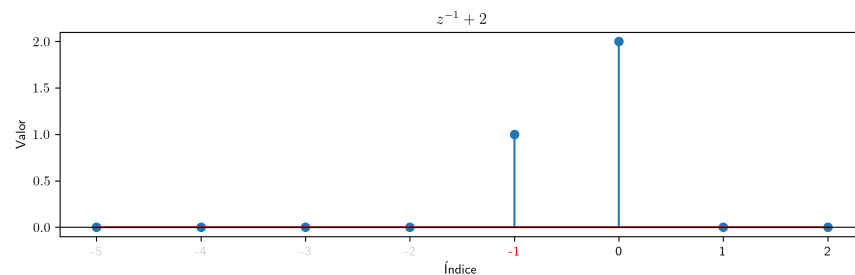
---

```

c=SerieConPrincipio([1,2,0,0],-1)
pinta(c)
c.plot_serie(indices_previos_al_cogrado=4, title='$'+latex(c)+'$').savefig(imagenes_leccion + '/serie_con_principioC.png', dpi=300, bbox_inches='tight')
c.RepEscalera=True
c

```

---



Fíjese que como el cogrado es negativo, esta secuencia NO es un polinomio.

## El producto entre polinomios es un producto convolución

**Ejercicio A** Ejecute los siguientes pasos:

- Multiplique el polinomio  $x + 2x^2$  con el polinomio  $1 - x$ .
  - Fíjese que al hacerlo, multiplica cada uno de los elementos del primer polinomio por cada uno de los elementos del segundo.
  - Fíjese que luego agrega (suma) los términos que corresponden a la misma potencia de  $x$ .
- Se pueden hacer los mismos cálculos usando la siguiente representación, donde el coeficiente de cada potencia  $k$  se calcula sumando los productos de los pares de números cuyos índices suman  $k$ .

El producto entre polinomios es un producto convolución.

$$(a * b)_t = \sum_{r+s=t} a_r b_s; \quad r, s, t \in \mathbb{Z}$$

Nótese que el *cogrado* de la secuencia resultante necesariamente es *suma* de los *cogrados*; y que el *grado* del resultado necesariamente es *suma* de los *grados*.

**Ejercicio B** Ejecute los siguientes pasos:

- Multiplique el polinomio  $x + 2x^2 - x^3$  con el polinomio  $x - x^2 + x^3$ .
  - Fíjese que al hacerlo, multiplica cada uno de los elementos del primer polinomio por cada uno de los elementos del segundo.
  - Fíjese que luego agrega (suma) los términos que corresponden a la misma potencia de  $x$ .
- Se pueden hacer los mismos cálculos usando la siguiente representación, donde el coeficiente de cada potencia  $k$  se calcula sumando los productos de los pares de números cuyos índices suman  $k$ .

El producto entre polinomios es un producto convolución.

$$(a * b)_t = \sum_{r+s=t} a_r b_s; \quad r, s, t \in \mathbb{Z}$$

Nótese que el *cogrado* de la secuencia resultante necesariamente es *suma* de los *cogrados*; y que el *grado* del resultado necesariamente es *suma* de los *grados*.

## Polinomio como SerieConPrincipio y su secuencia inversa con principio

Para mayor claridad, los siguientes ejemplos usan un parámetro simbólico. El bloque de código que puede ver a continuación define un parámetro simbólico  $a$ .

---

```
from sympy import symbols
a = symbols('a')
```

---

Definamos el polinomio  $1 - az$  como una **SerieConPrincipio** con cogrado 0:

---

```
p=SerieConPrincipio([1, -a],0)
pinta(p)
```

---

Su inversa  $p^{-▷}$  (en el conjunto de series *con principio*) es:

---

```
número_coeficientes = 12
pinta(p.inversa(número_coeficientes))
```

---

(donde hemos indicado el número de coeficientes que queremos mostrar, pero la secuencia tiene infinitos coeficientes no nulos si  $a \neq 0$ ; y por tanto NO es un polinomio si  $a \neq 0$ ).

**Nota.** Al haber creado este polinomio como un elemento de la clase **SerieConPrincipio**, el procedimiento `.inversa()` genera la inversa como una serie con principio (con cogrado). A esta inversa la hemos denotado con  $p^{-▷}$  en la lección correspondiente.

## Cogrado de la inversa con principio

Fíjese que su inversa (*con principio*) tiene cogrado 0 (el cogrado de la inversa es menos el cogrado de la serie. Veámoslo con un ejemplo:

---

```
q=SerieConPrincipio([1,-a],3)
pinta(q)
```

---

Su inversa con principio es

---

```
pinta(q.inversa())
```

---

Que posee infinitos términos no nulos si  $a \neq 0$  (por tanto no puede ser un polinomio cuando  $a \neq 0$ ). Pero además tiene cogrado negativo (no es un polinomio ni siquiera cuando  $a = 0$ ).

## La inversa con principio puede ser sumable o no

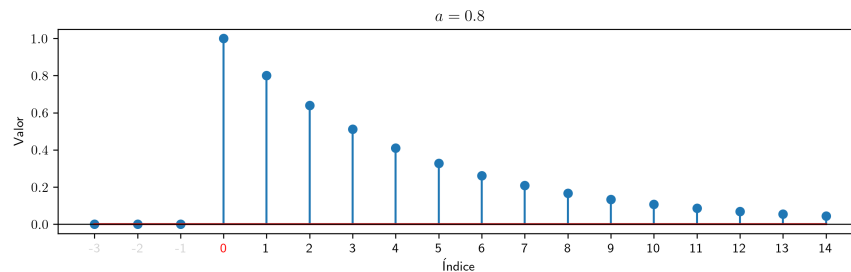
Consideremos el polinomio  $1 - ax$ . Mediante gráficos podemos ver el comportamiento de los coeficientes para distintos valores del parámetro  $a$ . Para estos dibujos calcularemos 15 coeficientes y sustituiremos el parámetro  $a$  por un valor concreto en cada caso.

- Para valores de  $a$  menores a 1 en valor absoluto (**raíz fuera del círculo unidad**)

---

```
valor = 0.8
p.inversa(15).subs((a,valor)).plot_series(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCP08.png', dpi=300, bbox_inches='tight')
```

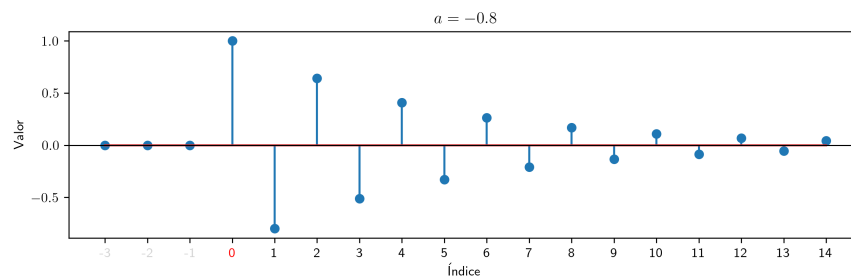
---




---

```
valor = -0.8
p.inversa(15).subs((a,valor)).plot_series(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCP-08.png', dpi=300, bbox_inches='tight')
```

---



Para polinomios con raíces fuera del círculo unidad los coeficientes de la inversa (con principio  $p^{-\triangleright}$ ) se aproximan a 0 cuando los índices crecen.

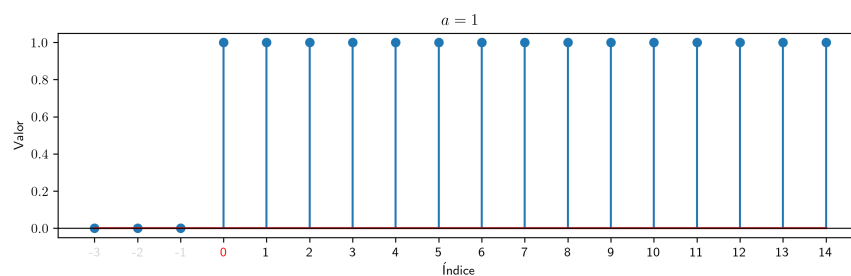
Éste es el caso en el que  $p^{-\triangleright} = p^{-1}$  (inversa absolutamente sumable).

- Para valores de  $a$  iguales a 1 en valor absoluto (**raíz sobre el círculo unidad**)

---

```
valor = 1
p.inversa(15).subs((a,valor)).plot_series(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCP1.png', dpi=300, bbox_inches='tight')
```

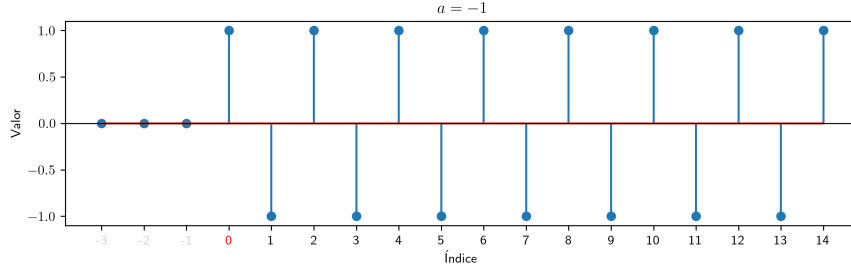
---




---

```
valor = -1
p.inversa(15).subs((a,valor)).plot_series(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCP-1.png', dpi=300, bbox_inches='tight')
```

---



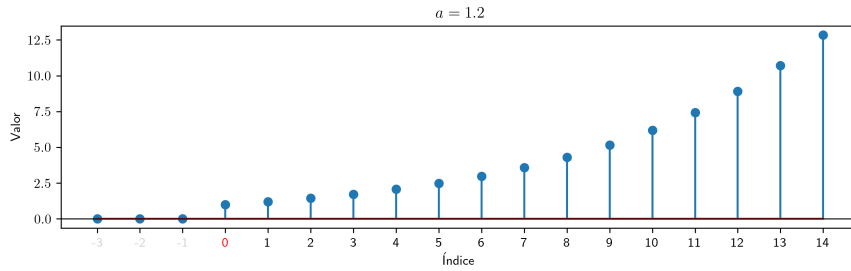
Para polinomios con raíces en el círculo unidad los coeficientes de la inversa mantienen patrones estables, y no son secuencias sumables (la suma de los coeficientes no converge).

- Para valores de  $a$  mayores a 1 en valor absoluto (**raíz en el interior del círculo unidad**)

---

```
valor = 1.2
p.inversa(15).subs((a,valor)).plot_series(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCP12.png', dpi=300, bbox_inches='tight')
```

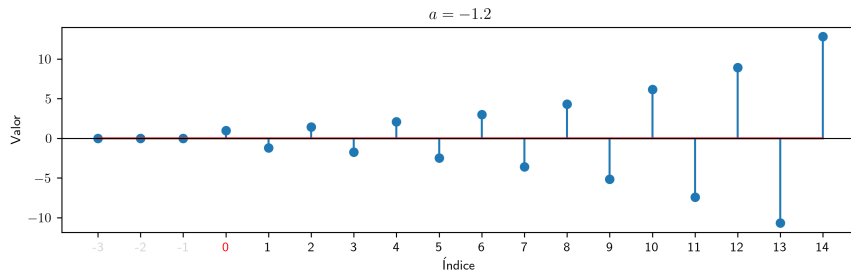
---




---

```
valor = -1.2
p.inversa(15).subs((a,valor)).plot_series(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCP-12.png', dpi=300, bbox_inches='tight')
```

---



Para polinomios con raíces dentro del círculo unidad los coeficientes de la inversa (con principio  $p^{-p}$ ) crecen exponencialmente cuando los índices crecen.

### Multipliquemos $1 - ax$ por su inversa con principio

Nótese que no estamos multiplicando dos polinomios.

$$\begin{aligned}
 (1 - ax) * (1 - ax)^{-p} &= (1 - ax) * (1 + ax + a^2x^2 + a^3x^3 + a^4x^4 + \dots) \\
 &= (1 + ax + a^2x^2 + a^3x^3 + a^4x^4 + \dots) - ax(1 + ax + a^2x^2 + a^3x^3 + a^4x^4 + \dots) \\
 &= (1 + ax + a^2x^2 + a^3x^3 + a^4x^4 + \dots) + (-ax - a^2x^2 - a^3x^3 - a^4x^4 - \dots) \\
 &= 1.
 \end{aligned}$$

Por ejemplo si el polinomio es  $1 - 2z$ :

Recuerde que el producto convolución entre secuencias es:

$$(a * b)_t = \sum_{r+s=t} a_r b_s; \quad r, s, t \in \mathbb{Z}$$

## La inversa de un polinomio con grado y cogrado distintos es una secuencia infinita

Recuerde que, en general, **la inversa de un polinomio es una serie con infinitos términos**. Como esta (incompleta) implementación en Python solo calcula una cantidad finita de términos, el producto del polinomio por la serie calculada no es exactamente 1, aunque nos podemos aproximar a la inversa tanto como queramos (dentro de las capacidades del ordenador y en casos sencillos).

Si calculamos solo cinco términos de la inversa, se anulan todos los coeficientes excepto el quinto (y el 1 de la posición 0):

---

```
pinta(p * p.inversa())
```

---

Con solo cincuenta términos se anulan todos excepto el quincuagésimo (y el 1 de la posición 0):

---

```
pinta(p * p.inversa(50))
```

---

Con quinientos términos se anulan todos excepto el parámetro en la posición 500 (y el 1 de la posición 0):

---

```
pinta(p * p.inversa(500))
```

---

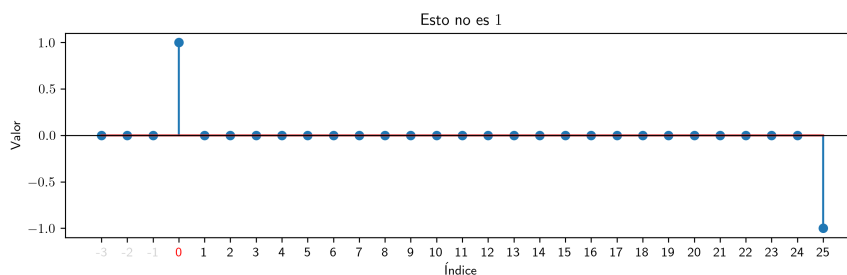
Dependiendo del valor de  $a$ , este producto puede estar arbitrariamente cerca, o arbitrariamente lejos de 1 (dada la existencia del  $n$ -ésimo término  $a^n$  donde  $n$  es el número de coeficientes calculados)

... aunque también podemos mantenernos a un palote (de norma 1) para todo  $n$ .

---

```
s = p.subs((a, 1))
(s * s.inversa(25)).plotSerie(title='Esto no es $1$').savefig(imagenes_leccion + '/noEs1.png', dpi=300, bbox_inches='tight')
```

---



En resumen, si  $a \neq 0$ , el producto  $p \cdot (p.inversa(500))$  no calcula exactamente la serie 1 (la que tienen un 1 en la posición 0 y es nula en todas las demás posiciones). Nótese que la componente correspondiente a la posición 500 es distinta de cero (su magnitud es  $a^{500}$ ). Por tanto,  $p.inversa(500)$  no es realmente la inversa de  $p$ . Fíjese que la serie  $p.inversa(500)$  solo tiene 500 términos no nulos; sin embargo, **la serie (con cogrado) que verdaderamente es inversa del polinomio  $p$  requiere de infinitos términos no nulos** (algo que esta implementación en Python no maneja).

## Otro ejemplo.

---

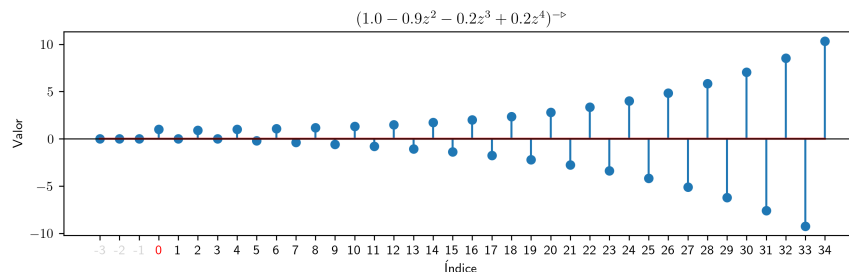
```
d = SerieConPrincipio([1, 0, -0.9, 0, -0.2, 0.2], 0)
pinta(d)
```

---

---

```
d.inversa(35).subs((a, valor)).plotSerie(title=r'$(1.0 - 0.9z^2 - 0.2z^3 + 0.2z^4)^{-\rightarrow}$').savefig(imagenes_leccion + '/Inversa')
```

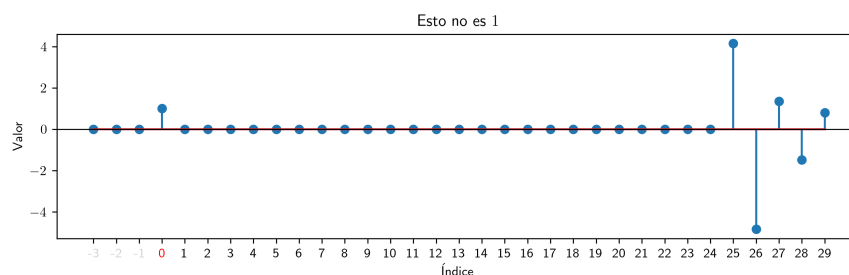
---




---

```
(d * d.inversa(25)).plot_serie(title='Esto no es $1$').savefig(imagenes_leccion + '/tampocoEs1.png', dpi=300, bbox_inches='tight')
```

---



Puede experimentar en el notebook todo lo que quiera, pero tenga en cuenta que si la inversa es una secuencia explosiva e intenta calcular muchos términos, excederá las capacidades de cálculo del ordenador.

**LO MÁS IMPORTANTE:** quédese con que, en general, *la inversa de un polinomio es una secuencia con infinitos términos no nulos*.

Digo en general, pues si el grado y cogrado del polinomio son el mismo índice, la inversa sí es finita:

$$(x^2) * (x^{-2}) = 1.$$

## Polinomio como SerieConFinal y su secuencia inversa con final

Ahora vamos a definir exactamente el mismo polinomio  $1 - ax$ ; pero en esta ocasión lo haremos como una `SerieConFinal` (con grado igual a 1).

---

```
p2=SerieConFinal([1,-a],1)
pinta(p2)
```

---

Su inversa  $p^{\leftarrow}$  (en el conjunto de series *con final*) es:

---

```
pinta(p2.inversa())
```

---

Fíjese que como el grado del polinomio `p2` es 1, el grado de  $p^{\leftarrow}$  es  $-1$ .

**Nota.** Al haber creado este polinomio como un elemento de la clase `SerieConFinal`, el procedimiento `.inversa()` genera la inversa como una serie con final (con grado). A esta inversa la hemos denotado con  $p^{\leftarrow}$  en la lección correspondiente.

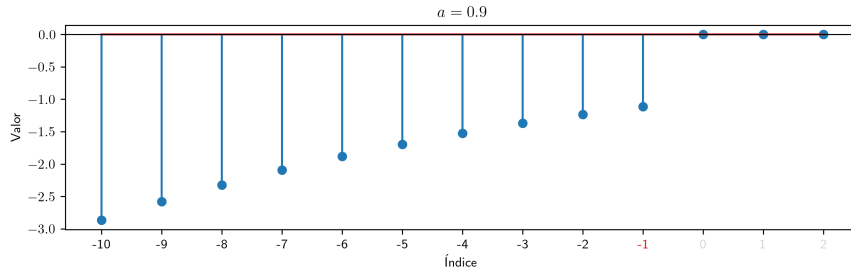
Veamos con unos gráficos el comportamiento de los coeficientes de esta nueva inversa  $p^{\leftarrow}$  para distintos valores del parámetro  $a$ .

- Para valores de  $a$  menores a 1 en valor absoluto (**raíz fuera del círculo unidad**)

---

```
valor = 0.9
p2.inversa(10).subs((a,valor)).plot_serie(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCF09.png', dpi=300, bbox_inches='tight')
```

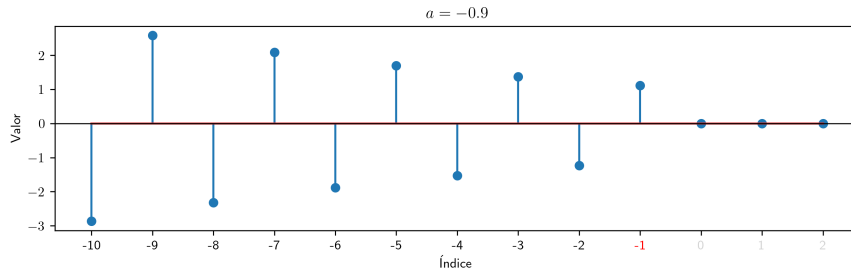
---




---

```
valor = -0.9
p2.inversa(10).subs((a,valor)).plot_serie(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCF-09.png', dpi=300, bbox_inches='tight')
```

---



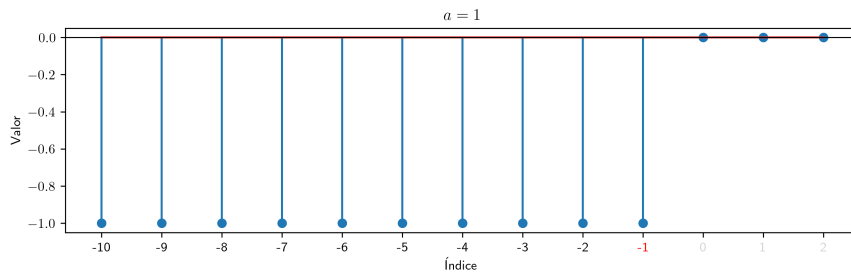
Para polinomios con raíces fuera del círculo unidad los coeficientes de la inversa (con final  $p^{\leftarrow}$ ) crecen exponencialmente cuando los índices decrecen (justo lo contrario que con las inversas con principio).

- Para valores de  $a$  iguales a 1 en valor absoluto (raíz sobre el círculo unidad)

---

```
valor = 1
p2.inversa(10).subs((a,valor)).plot_serie(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCF1.png', dpi=300, bbox_inches='tight')
```

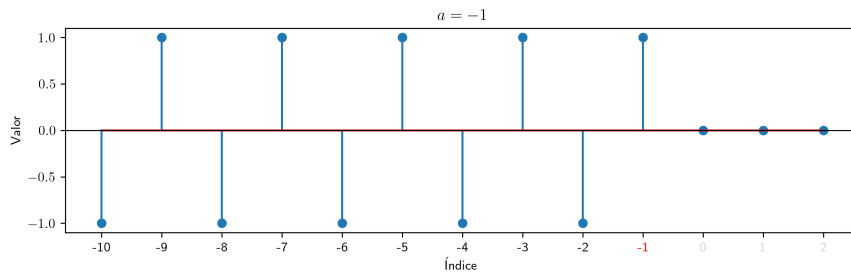
---




---

```
valor = -1
p2.inversa(10).subs((a,valor)).plot_serie(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCF-1.png', dpi=300, bbox_inches='tight')
```

---



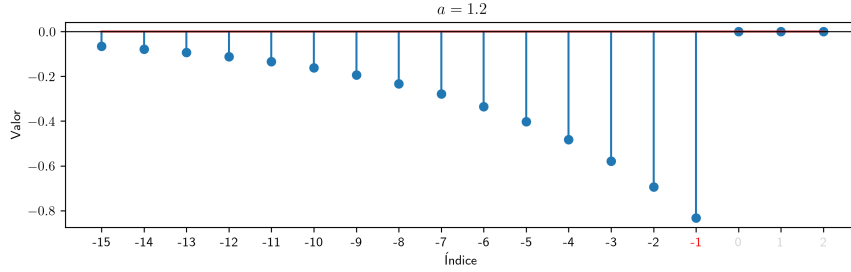
Para polinomios con raíces en el círculo unidad los coeficientes de la inversa mantienen patrones estables (de manera similar a lo que ocurre con las inversas con principio).

- Para valores de  $a$  mayores a 1 en valor absoluto (**raíz en el interior del círculo unidad**)

---

```
valor = 1.2
p2.inversa(15).subs((a,valor)).plotSerie(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCF12.png', dpi=300, bbox_inches='tight')
```

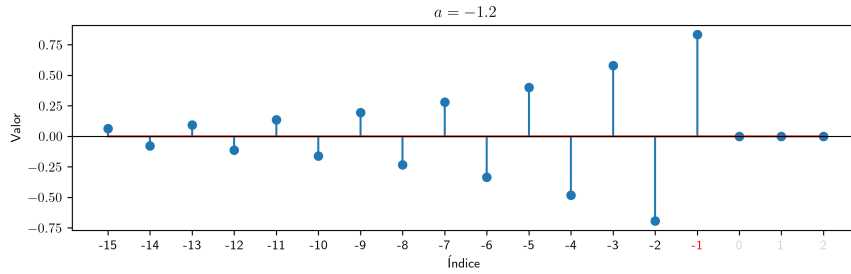
---




---

```
valor = -1.2
p2.inversa(15).subs((a,valor)).plotSerie(title='$a='+str(valor)+'$').savefig(imagenes_leccion + '/InvCF-12.png', dpi=300, bbox_inches='tight')
```

---



Para valores de  $a$  mayores a 1 en valor absoluto los coeficientes de la inversa (con final  $p^{\leftarrow}$ ) se aproximan a 0 cuando los índices decrecen (justo lo contrario que con las inversas con principio).

Este es el caso en el que  $p^{\leftarrow} = p^{-1}$  (inversa absolutamente sumable).

### Multipliquemos $1 - ax$ por su inversa con final

Nótese que no estamos multiplicando dos polinomios.

$$\begin{aligned}
 (1 - ax) * (1 - ax)^{\leftarrow} &= (1 - ax) * \left( \cdots - \frac{1}{a^4}x^{-4} - \frac{1}{a^3}x^{-3} - \frac{1}{a^2}x^{-2} - \frac{1}{a}x^{-1} \right) \\
 &= \left( \cdots - \frac{1}{a^4}x^4 - \frac{1}{a^3}x^3 - \frac{1}{a^2}x^2 - \frac{1}{ax} \right) - ax \left( \cdots - \frac{1}{a^4}x^4 - \frac{1}{a^3}x^3 - \frac{1}{a^2}x^2 - \frac{1}{ax} \right) \\
 &= \left( \cdots - \frac{1}{a^4}x^4 - \frac{1}{a^3}x^3 - \frac{1}{a^2}x^2 - \frac{1}{ax} \right) + \left( \cdots + \frac{1}{a^3}x^3 + \frac{1}{a^2}x^2 + \frac{1}{ax} + 1 \right) \\
 &= 1.
 \end{aligned}$$

Por ejemplo si el polinomio es  $1 - \frac{1}{2}z$ :

Recuerde que el producto convolución entre secuencias es:

$$(a * b)_t = \sum_{r+s=t} a_r b_s; \quad r, s, t \in \mathbb{Z}$$

Cabe señalar que un polinomio es cero para índices menores que su co-grado y cero para índices mayores que su grado, por lo tanto, en el cálculo del producto de convolución, para cada  $t$  solo interviene la suma de una cantidad finita de productos de coeficientes no nulos.

Lo mismo ocurre con las secuencias con final, que son cero para índices mayores que su grado.

Sin embargo, el problema surge cuando intentamos multiplicar dos secuencias infinitas en ambos sentidos, es decir, que tienen un número infinito de coeficientes no nulos tanto en los índices negativos como en los positivos. Para que el producto de convolución esté definido entre ellas, es necesario que cada una de las sumas  $\sum_{r+s=t} a_r b_s$  sea convergente, y esto está garantizado si las secuencias son absolutamente sumables.

Además, se asume que los modelos deben explicar el presente en función del pasado (lo que ocurre hoy afecta a lo que ocurrirá mañana pero lo que ocurre hoy no cambia lo que pasó ayer). Consecuentemente se usa el operador de retardo combinado con secuencias con principio. En particular se emplean expresiones en las que intervienen inversas de polinomios que son secuencias con principio. Estas inversas solo pueden ser convergentes si las raíces del polinomio se encuentran fuera del círculo unidad (si tienen un módulo mayor que uno).

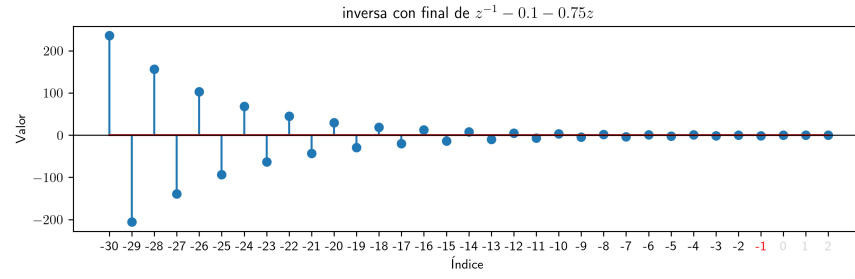
## Inversas de series con principio y de series con final

```
lista = [1, -.1, -.75]
s_cp = SerieConPrincipio(lista, -1)
pinta(s_cp)
```

inversa con principio de  $z^{-1} - 0.1 - 0.75z$

Índice	Valor
-2	0.00
-1	0.00
0	0.00
1	1.00
2	0.10
3	0.75
4	0.15
5	0.60
6	0.18
7	0.45
8	0.18
9	0.35
10	0.17
11	0.28
12	0.15
13	0.25
14	0.14
15	0.19
16	0.13
17	0.16
18	0.14
19	0.13
20	0.11
21	0.11
22	0.09
23	0.09
24	0.07
25	0.08
26	0.06
27	0.07
28	0.05
29	0.06
30	0.04

```
s_cf.inversa(30).plot_serie(title='inversa con final de $' + s_cf.latex() + '$').savefig(imagenes_leccion + '/InversaSerieConf')
```



Ambas son secuencias inversas de la expresión  $z^{-1} - 0,1 - 0,75z$ . Fíjese que una inversa es convergente y la otra es explosiva.