Espacios de Hilbert

April 9, 2020

1 Espacios de Hilbert y Expansiones ortogonales

Estudiando las soluciones de la ecuación de Laplace por el método de separación de variables encontramos que las soluciones se podían expresar según

$$\Phi(\vec{x}) = \sum_{n_1,n_2} A_{n_1,n_2}\varphi_{n_1,n_2}(\vec{x})$$

con $\varphi_{n_1,n_2}(\vec{x})$ un conjunto infinito de soluciones tipo producto. Para el problema del potencial dentro de una "caja" cuadrada, con todos menos uno de sus bordes sometidos a potencial $V = 0$,

$$\varphi_{n_1,n_2}(\vec{x}) = \sin(\frac{n_1\pi}{a}x)\sin(\frac{n_2\pi}{a}y)\sinh(\frac{\sqrt{n_1^2+n_2^2}\pi}{a}z)$$

Por otro lado, para el problema en el exterior de una esfera,

$$\varphi_{l,m}(\vec{x}) = \frac{Y_{lm}(\theta,\phi)}{r^{l+1}}$$

con $Y(\theta,\phi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos(\theta))e^{\mathbf{i}m\phi}$ los *esféricos armónicos*.

Estas soluciones "separables" cumplen dos condiciones muy importantes:

1. Son *completas* en el sentido de que superponiéndolas linealmente, nos permiten construir, eligiendo adecuadamente los coeficientes A_{n_1,n_2}, soluciones que sobre el borde que no fijamos ($z = a$ en el primer caso, $r = a$ en el segundo) sean iguales a cualquier función arbitraria (suave).

$$\phi(\vec{x}) = \sum_{m,n} A_{m,n}\varphi_{m,n}(\vec{x})$$

2. Son *ortogonales* en el sentido de que si definimos un producto escalar de la forma[1]

$$(f,g) = \int_\partial f^*(\vec{x})g(\vec{x})dA$$

donde la integral se define sobre el borde "libre" estas cumplen que

$$(\varphi_{n_1,n_2},\varphi_{n_1',n_2'}) = 0 \quad \text{si } n_1 \neq n_1' \text{ o } n_2 \neq n_2'$$

[1]Ejercicio: verificar que esta definición es compatible con la de un producto escalar complejo (1. positividad, 2. bilinealidad 2. simetria)

Podemos pensar entonces que las soluciones de la ecuación diferencial forman un *espacio vectorial* con *producto interno*, donde las funciones $\varphi_{m,n}(x)$ forman una *base ortogonal*.

Como las expansiones involucran sumas infinitas, necesitamos además establecer una noción de convergencia para estas. Para esto, usando que un *producto escalar* induce una noción de *distancia*

$$\|f(\vec{x}) - g(\vec{x})\| = \sqrt{\int |f(\vec{x}) - g(\vec{x})|^2 d^3x}$$

sobre cualquier par de funciones f, g *cuadrado integrables*:

$$\int |f(x)|^2 d^3x, \int |g(x)|^2 d^3x < \infty$$

Si ahora consideramos la sucesión de funciones

$$f_N(\vec{x}) = \sum_{m=0,n=0}^{m+n\leq N} A_{mn}\varphi_{mn}(\vec{x})$$

dadas por sumas parciales *finitas*, diremos que estas convergen *en el sentido de Cauchy* si para cualquier ε, existe un N_0 tal que

$$\|f_{N_1} - f_{N_2}(x)\| < \varepsilon$$

para cualqueier par $N_1, N_2 > N_0$.

Si asumimos que cualquier sucesión que cumple la condición (de Cauchy) se corresponde a un elemento del espacio de soluciones, decimos que el espacio de soluciones es un *Espacio de Hilbert*.

Veremos que los *Espacios de Hilbert* un una herramienta muy poderosa para describir soluciones aproximadas de ecuaciones diferenciales, pero para entender cómo trabajar con estos, empecemos por algo más simple.

2 Expansiones ortogonales en dimensión finita.

$\mathbb{R}^3$ es un ejemplo muy básico de Espacio de Hilbert. Aquí, el producto escalar entre dos vectores es el producto escalar usual: si

$$\vec{r} = r_x\check{e}_x + r_y\check{e}_y + r_z\check{e}_z \quad \text{y} \quad \vec{r}' = r'_x\check{e}_x + r'_y\check{e}_y + r'_z\check{e}_z$$

con $\{\check{e}_x, \check{e}_y, \check{e}_z\}$ la *base canónica*,

$$\vec{r}\cdot\vec{r}' = r_xr'_x + r_yr'_y + r_zr'_z$$

Como la base canónica se *define* ortonormal, $\check{e}_i \cdot \check{e}_j = \delta_{ij}$, podemos descomponer cualquier vector de manera única como indicamos arriba, y recuperar las proyecciones correspondientes tomando el producto escalar con los elementos de la base:

$$r_i = \check{e}_i \cdot \vec{r}$$

De esta manera, podemos descomponer cualquier vector en la base canónica. Lo mismo se aplica a cualquier otra base ortogonal: si $\{\vec{b}_i\}$ es otra base ortogonal ($\vec{b}_i \cdot \vec{b}_j = 0 \Leftrightarrow i \neq j$), podemos expresar cualquier vector como

$$\vec{r} = \sum_{i=1}^{3} \vec{b}_i \frac{\vec{b}_i \cdot \vec{r}}{\vec{b}_i \cdot \vec{b}_i}$$

Notemos que podemos obtener los coeficientes de esta forma debido a la ortogonalidad de la base.

Finalmente, como el conjunto de los números reales es *completo*, y la base de vectores del espacio es finita, cualquier sucesión de Cauchy de vectores converge a algún vector en $\mathbb{R}^3$: si $\vec{v}_n$ es una sucesión tipo Cauchy, para N suficientemente grande,

$$\|\check{e}_x \cdot \vec{v}_n - \check{e}_x \cdot \vec{v}_m\| \leq \|\vec{v}_n - \cdot \vec{v}_m\| < \varepsilon$$

luego, la "componente en x" es también de Cauchy. De esta manera,

$$\lim_{n\to\infty} \vec{v}_n = \lim_{n\to\infty} \sum_{i=x,y,z} ((\check{e}_y \cdot \vec{v}_n))\check{e}_i = \sum_{i=x,y,z} (\lim_{n\to\infty} (\check{e}_y \cdot \vec{v}_n))\check{e}_i = \vec{v}$$

Notemos que en esta prueba, es importante que la suma sobre las coordenadas sea finita para poder intercambiar los lugares de la suma y el límite.

3 Espacios de funciones en el intervalo $[-1, 1]$

Un ejemplo ligeramente más complicado lo encontramos al estudiar las funciones *suaves* $f : \mathbb{R} \to [-1, 1] \subset \mathbb{R}$.

Si definimos el producto escalar en ese intervalo como

$$(f, g) = \int_{-1}^{1} f(x)g(x)dx$$

está claro que cualquier par de funciones resulta en un producto escalar *finito*.

Un subconjunto de estas funciones viene dado por los polinomios de grado $\leq N$. Para este subconjunto, una base obvia es la de los *monomios* $\{1, x, x^2, \ldots, x^m \ldots x^N\}$. Obviamente, esta base no es ortonormal, ya que

$$(x^m, x^n) = \int_{-1}^{1} x^{m+n} dx = \frac{1 + (-1)^{m+n}}{m + n + 1}$$

que sólo se anula si m y n tienen diferente paridad. Por otro lado, si aplicamos el método de Gramm-Schmidt para ortogonalizar el conjunto

$$p_l(x) = x^l - \sum_{l'=0}^{l-1} \frac{(p_{l'}(x), x^l)}{(p_{l'}(x), p_{l'}(x))} x^{l'}$$

obtenemos para los primeros términos

$$\begin{aligned} p_0 &= 1 \\ p_1 &= x \\ p_2 &= x^2 - \frac{1}{3} \\ p_3 &= x^3 - \frac{3}{5}x \\ &\dots \end{aligned}$$

que resultan proporcionales a los Polinomios de Legendre.

3.1 Completitud

Por ser soluciones de una ecuación tipo Sturm-Liouville, el conjunto de los polinomios de Legendre es un *conjunto completo*, en el sentido de que permiten expresar a la distribución $\delta(x - x')$ con x' en cualquier punto del dominio como una combinación lineal de estos:

$$\delta(x - x') = \sum_l \frac{2l+1}{2} P_l(x) P_l(x') \tag{1}$$

Esto significa que toda función suave puede expandirse como una combinación lineal de estos polinomios:

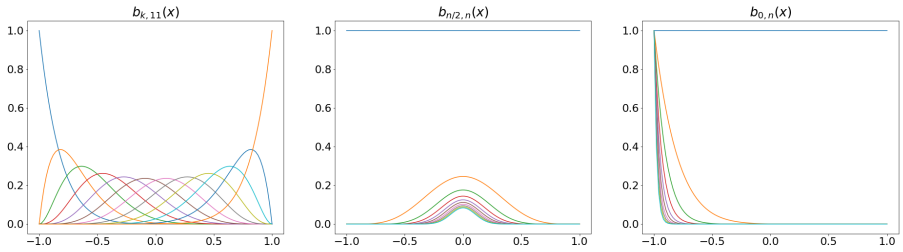
$$f(x) = \int f(x')\delta(x - x')dx' = \sum_l \left(\frac{2l+1}{2} \int f(x') P_l(x') dx' \right) P_l(x)$$

Para probar 1 (independientemente de la propiedad de la ecuación diferencial que dá origen a los $P_l(x)$, es más conveniente estudiar otra familia de polinomios: los Polinomios de Bernstein:

$$B_{k,N}(x) = \frac{1}{2^N} \binom{N}{k} (x+1)^k (1-x)^{N-k}$$

Como son polinomios de grado N, estos pueden expandirse como combinaciones lineales de los $P_l(x)$ con $l \leq N$. Vemos además que

- $B_{k,N}(x)$ son base de los polinomios de grado $\leq N$.
- $B_{k,N}(x) \geq 0$, y sólo se anulan si $x = 1$ o $x = -1$
- Tienen un único máximo en $x = \frac{2k}{N} - 1$
- Para $N \to \infty$, su soporte se encuentra concentrado en la región $|x - (\frac{2k}{N} - 1)| < \frac{4}{N}$
- $\sum_{k=0}^{N} B_{k,N}(x) = ((1-x)/2 + (1+x)/2)^N = 1$
- $\sum_{k=0}^{N} B_{k,N}(x) = 1$
- $\int_{-1}^{1} B_{k,N}(x)dx = \frac{2}{N+1}$



3.2 Aproximación polinómica

Ahora que sabemos que la base de polinomios es completa, podemos preguntarnos por cómo construir una aproximación polinómica de grado N a una función suave.

Una primera forma de obtener esta aproximación es minimizar la distancia

$$|f-g|^2 = \int_{-1}^{1} |f(x)-p(x)|^2 dx$$

respecto a los coeficientes de $p(x) = \sum_{k=0}^{d} a_k x^k$, esto es, hacer un ajuste de mínimos cuadrados.

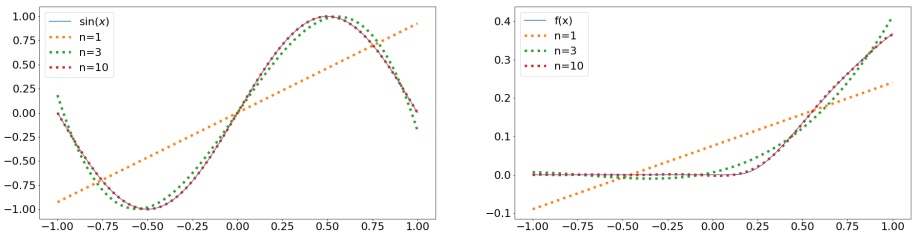
Sin embargo, aprovechando que tenemos una base ortonormal de este espacio, podemos construir este polinomio expandiendo en esa base:

$$p(x) = \sum_{l=0}^{d} \frac{2l+1}{2} P_l(x) \int_{-1}^{1} f(x') P_l(x') dx'$$

Nótese que dicha expansión puede o no parecerse a la expansión de $f(x)$ en serie de potencias: Si la función $f(x)$ es analítica en el intervalo (como la función $\sin(\pi x)$) ambas aproximaciones coinciden. Por otro lado, si $f(x)$ es una función suave (infinitamente derivable) pero no es analítica (en el sentido de que su serie de Taylor no converge a la función en el intervalo) esto no será así. Por ejemplo, la función

$$f(x) = \begin{cases} e^{-1/x} & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{2}$$

es una función infinitamente derivable en el intervalo $[-1, 1]$, pero no es analítica: Sus derivadas en $x = 0$ se anulan a todo orden, por lo que su serie de Taylor converge a la función nula. Sin embargo, su expansión polinómica converge uniformemente en el intervalo.



3.3 3.3 Funciones peso

En ocasiones, nos puede interesar que la aproximación polinómica se ajuste mejor a los datos en una zona del intervalo más que en otra. Podemos tener en cuensta esto modificando el "peso" que le damos a los diferentes puntos del intervalo en que definimos el producto escalar:

$$(f,g)_w = \int_{-1}^{1} f(x)g(x)w(x)dx$$

donde $w(x)$ es una función positiva en el intervalo. De esta manera, la distancia entre dos funciones toma la forma

$$|f-g|_w = \sqrt{\int_{-1}^{1}(f(x)-g(x))^2 w(x)dx}$$

Si $g(x)$ está restringido a ser un polinomio de grado $\leq N$, al optimizar esta distancia, se premiará a aquellos polinomios que ajusten mejor donde $w(x)$ tome valores más grandes.

Al cambiar $w(x)$, la noción de ortogonalidad cambia, y por lo tanto, el resultado de ortogonalizar a los monomios vía Gramm-Schmidt se ve modificado.

Por ejemplo, si $w(x) = \frac{1}{\sqrt{1-x^2}}$, los polinomios que obtenemos son de la forma

$$\tilde{T_0}(x) = 1 \quad (3)$$
$$\tilde{T_1}(x) = x \quad (4)$$
$$\tilde{T_n}(x) = xT_n(x) - T_{n-1}(x)/2 \quad (5)$$

Estos polinomios son (proporcionales a) los llamados Polinomios de Chebysheb, $T_n(x)$, que satisfacen

$$T_n(\cos(\theta)) = \cos(n\theta)$$

3.4 Series de Fourier

Los polinomios no son las únicas funciones que permiten construir bases ortonormales completas. En general, dado un producto escalar

$$(f,g) = \int_{-1}^{1} w(x)f^*(x)g(x)dx$$

donde generalizamos la definición para funciones a valores complejos, existen infinitas posibles bases $\{\varphi_i(x)\}$ que son ortonormales

$$\int \varphi_i^*(x)\varphi_j(x)w(x)dx = \delta_{ij}$$

y completas

$$\sum_i \varphi_i(x)\varphi_i^*(x') = \delta(x-x')/w(x)\,.$$

Un caso muy importante es el de las funciones de la forma $\varphi_n(x) = \frac{\exp(\mathbf{i}n\pi x)}{\sqrt{2}}$ con $n \in \mathbb{Z}$.

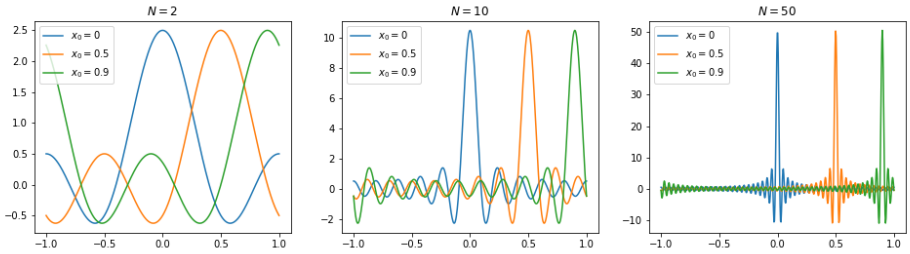
Verificamos que efectivamente son funciones ortogonales entre sí (con $w(x) = 1$):

$$\int_{-1}^{1} \varphi_n(x)^* \varphi_m(x) dx = \frac{1}{2} \int_{-1}^{1} \exp(\mathbf{i}(m-n)\pi x) dx = \frac{\exp(\mathbf{i}(m-n)\pi) - \exp(-\mathbf{i}(m-n)\pi)}{(m-n)\pi} = \delta_{mn}$$

por otro lado

$$\begin{aligned}
\sum_{n=-N}^{N} \varphi_n(x)^* \varphi_n(x') &= \frac{1}{2} \sum_{n=-N}^{N} \exp(\mathbf{i}n\pi(x'-x)) && (6)\\
&= \frac{e^{-\mathbf{i}N\pi(x'-x)}}{2} \sum_{n=0}^{2N} (\exp(\mathbf{i}\pi(x'-x)))^n && (7)\\
&= \frac{e^{-\mathbf{i}N\pi(x'-x)}}{2} \frac{1-\exp(\mathbf{i}\pi(N+1)(x'-x))}{1-\exp(\mathbf{i}\pi(x'-x))} && (8)\\
&= \frac{1}{2} \frac{\sin(\frac{\pi}{2}(2N+1)(x'-x))}{\sin(\frac{\pi}{2}(x'-x))} = \delta_N(x'-x) = \delta_N(x-x') && (9)
\end{aligned}$$

$\delta_N(x)$ es una función que conocimos al estudiar la difracción de la luz por un conjunto de N rendijas.



Ahora, notamos que

- La integral de $\delta_N(x - x')$ en el intervalo $[-1, 1]$ es igual a la unidad. Esto lo podemos comprobar integrando término a término la suma que la define:

$$\begin{aligned}
\int_{-1}^{1} \sum_{n=-N}^{N} \frac{e^{\mathbf{i}\pi n(x-x')}}{2} dx' &= \sum_{n=-N}^{N} \int_{-1}^{1} \frac{e^{\mathbf{i}\pi n(x-x')}}{2} dx' && (10)\\
&= \sum_{n=-N}^{N} \frac{e^{\mathbf{i}\pi nx}}{2} \int_{-1}^{1} e^{-\mathbf{i}\pi nx'} dx' && (11)\\
&= \sum_{n=-N}^{N} e^{\mathbf{i}\pi nx} \frac{e^{\mathbf{i}\pi n} - e^{-\mathbf{i}\pi n}}{\mathbf{i}\pi n} dx' = 1 && (12)
\end{aligned}$$

  donde sólo sobrevive el término con $n = 0$ por la periodicidad de la exponencial.
- La integral sobre un intervalo (a, b) con $-1 < x < a < b < 1$ se anula en el límite $N \to \infty$. Vemos eso observando que en el intervalo, $\sin(\frac{\pi}{2}(x'-x)) > \sin(\frac{\pi}{2}(a-x))$, luego,

$$\left|\int_a^b \delta_N(x-x')dx'\right| = \left|\int_a^b \frac{1}{2}\frac{\sin(\frac{\pi}{2}(2N+1)(x'-x))}{\sin(\frac{\pi}{2}(x'-x))}dx'\right| \quad (13)$$

$$< \frac{1}{2\sin(\frac{\pi}{2}(a-x))}\left|\int_a^b \sin(\frac{\pi}{2}(2N+1)(x'-x))dx'\right| \quad (14)$$

$$= \frac{1}{N+1/2}\left|\frac{|\cos(\frac{\pi}{2}(2N+1)(a-x)) - \cos(\frac{\pi}{2}(2N+1)(b-x))|}{2\pi\sin(\frac{\pi}{2}(a-x))}\right| \quad (15)$$

Cuando $N \to \infty$, esta integral se anula. De manera análoga, si $-1 < a < b < x < 1$, la integral también se anula.

De estos dos resultados se sigue que

$$\lim_N \delta_N(x - x0) = \delta(x - x') \text{ para } (|x|, |x'| < 1)$$

Con lo que probamos la completitud. Notemos que fuera del intervalo $[-1, 1]$, $\delta_N(x - x0)$ es una función periódica de período 2, por lo que su límite también lo es.

4 Bases producto

Supongamos que ahora buscamos una base ortogonal y completa para las funciones en $\mathcal{R} = [-1, 1] \times [-1, 1]$. Si el producto escalar se define de manera que

$$(f, g) = \iint f^*(x, y)g(x, y)dxdy$$

Si $\{\varphi_i(x)\}$ es una base ortonormal y completa de funciones para el intervalo $[-1, 1]$, entonces $\{\varphi_i(x)\varphi_j(y)\}$ será una base ortonormal y completa para $\mathbb{R}$. Para verlo, observamos que

$$(\varphi_i(x)\varphi_j(y), \varphi_{i'}(x)\varphi_{j'}(y)) = \iint \varphi_i^*(x)\varphi_j^*(y)\varphi_{i'}(x)\varphi_{j'}(y)dxdy \quad (16)$$

$$= (\int \varphi_i^*(x)\varphi_{i'}(x)dx)(\int \varphi_j^*(y)\varphi_{j'}(y)dy) \quad (17)$$

$$= \delta_{ii'}\delta_{jj'} \quad (18)$$

Por otro lado, para ver la completitud,

$$\sum_{ij} \varphi_i(x)\varphi_j(y)\varphi_i^*(x')\varphi_j^*(y') = (\sum_i \varphi_i(x)\varphi_i^*(x'))(\sum_j \varphi_j(y)\varphi_j^*(y')) \quad (19)$$

$$= \delta(x - x')\delta(y - y') \quad (20)$$

En esto se basa la estrategia de separación de variables para resolver el problema de Laplace: Supongamos que

$$\nabla^2\Phi_\lambda(\vec{r}) = \lambda\Phi_\lambda(\vec{r})$$

con $\Phi_\lambda(\vec{r})$ dependiendo de todas menos una variable (digamos, s) Llamemos Ω a esas variables que no son s. Luego, habrá alguna función $S_\lambda(s)$ tal que

$$\nabla^2(S_\lambda(s)\Phi_\lambda(\Omega)) = 0$$

pero esto significa que

$$\nabla^2(S_\lambda(s)\Phi_\lambda(\vec{r})) = \Phi_\lambda(\vec{r})\nabla^2 S_\lambda(s) + 2(\nabla\Phi)\cdot(\nabla S_\lambda) + S_\lambda(s)\nabla^2\Phi_\lambda(\Omega)$$

Si el sistema de variables (s, Ω) son coordenadas ortogonales, el término del producto de gradientes se anula y por lo tanto,

$$\nabla^2(S_\lambda(s)\Phi_\lambda(\vec{r})) = \Phi_\lambda(\vec{r})\nabla^2 S_\lambda(s) + \lambda S_\lambda(s)\Phi_\lambda(\Omega) = \nabla^2 S_\lambda(s) + \lambda S(s) = 0$$

Luego, $\{S_\lambda(s)\Phi_\lambda(\Omega)\}$ será una base completa de *soluciones* de la ecuación diferencial. Nos queda entonces la libertad para elegir los coeficientes de la expansión para fijar el valor de la solución para un cierto $s = s^*$, que define el borde de la región a resolver.

5 Rutinas que generaron los gráficos

5.1 Ajuste de la función $\sin(x)$ y $f(x)$

[26]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as scpint

plt.rcParams.update({'font.size': 22})

fig, axs = plt.subplots(1,2,figsize=(32,8))




ax = axs[0]


xs = np.linspace(-1,1,100)
ys = np.sin(np.pi*xs)

ax.plot(xs, ys,label="$\\sin(x)$")


degree = 1
coeffs = np.polyfit(xs,ys,degree)


ysapp=sum([coeffs[n] * xs**(degree-n) for n in range(degree+1)])
```

```
ax.plot(xs, ysapp, label="n=1",ls=":",lw=5)


degree = 3
coeffs = np.polyfit(xs,ys,degree)

ysapp=sum([coeffs[n] * xs**(degree-n) for n in range(degree+1)])
ax.plot(xs, ysapp, label="n=3",ls=":",lw=5)


degree = 10
coeffs = np.polyfit(xs,ys,degree)

ysapp=sum([coeffs[n] * xs**(degree-n) for n in range(degree+1)])
ax.plot(xs, ysapp, label="n=10",ls=":",lw=5)
coeffs_sin = coeffs
ax.legend()




################################################################
ax = axs[1]


def func(x):
    if type(x) is np.ndarray:
        xs = x
        return np.array([np.exp(-1/x) if x>0 else 0 for x in xs])
    else:
        if x<0:
            return 0
        else:
            np.exp(-1/x)


xs = np.linspace(-1,1,100)
ys = func(xs)
ax.plot(xs, ys,label="f(x)")


degree = 1
coeffs = np.polyfit(xs,ys,degree)

ysapp=sum([coeffs[n] * xs**(degree-n) for n in range(degree+1)])
ax.plot(xs, ysapp, label="n=1",ls=":",lw=5)
```

```
degree = 3
coeffs = np.polyfit(xs,ys,degree)

ysapp=sum([coeffs[n] * xs**(degree-n) for n in range(degree+1)])
ax.plot(xs, ysapp, label="n=3",ls=":",lw=5)

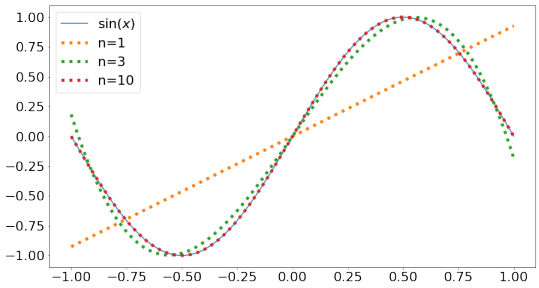

degree = 10
coeffs = np.polyfit(xs,ys,degree)

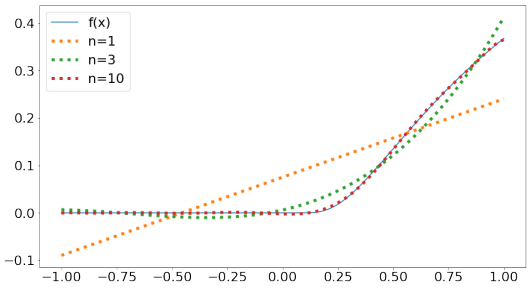
ysapp=sum([coeffs[n] * xs**(degree-n) for n in range(degree+1)])
ax.plot(xs, ysapp, label="n=10",ls=":",lw=5)

ax.legend()

##################################
# fig.savefig("fig_fit.png")
```

[26]: <matplotlib.legend.Legend at 0x7f064f738e10>





5.2 Grafica de los polinomios de Bernstein

[52]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as scpint
import scipy.special as scpsp

plt.rcParams.update({'font.size': 22})

n = 11

fig, ax = plt.subplots(1,3,figsize=(32,8))
```

```
xs = np.linspace(-1,1,100)
for k in range(n+1):
    ax[0].plot(xs, 2**(-n)*scpsp.
 ↪binom(n,k)*(xs+1)**(k)*(1-xs)**(n-k),label=str(k))

ax[0].set_title("$b_{k,11}(x)$")

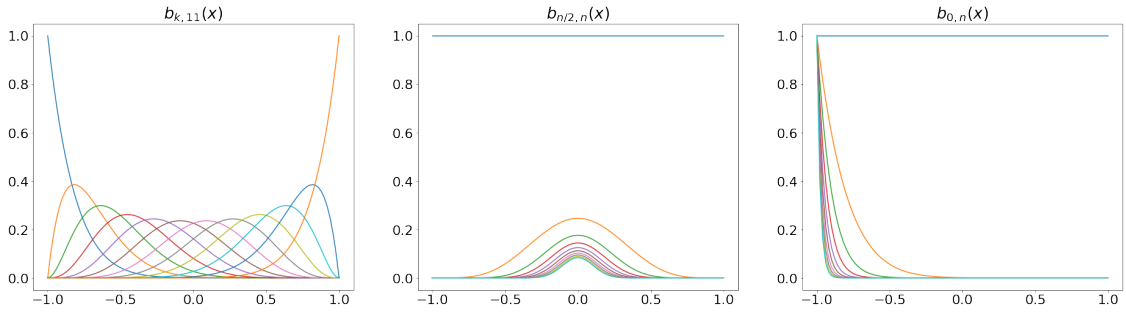

xs = np.linspace(-1,1,100)
for nn in range(10):
    n = nn*10
    ax[1].plot(xs, 2**(-n)*scpsp.binom(n,n/2)*(xs+1)**(n/2)*(1-xs)**(n/
 ↪2),label=str(n))

ax[1].set_title("$b_{n/2,n}(x)$")


for nn in range(10):
    n = nn*10
    ax[2].plot(xs, 2**(-n)*(1-xs)**(n),label=str(n))

ax[2].set_title("$b_{0,n}(x)$")


fig.savefig("fig_bkn1.png")
```



5.3 Grafica de $\delta_N(x' - x)$

[24]:
```
import matplotlib.pyplot as plt
import numpy as np

n=10
xs = np.linspace(-.99999999,1,500)
```

```
fig,axs = plt.subplots(1,3,figsize=(16,4))

ax = axs[0]
n = 2
ax.set_title("$N=2$")
ax.plot(xs,np.sin(pih*(xs)*(2*n+1))/(2.*np.sin(pih*(xs))),label="$x_0=0.$")
ax.plot(xs,np.sin(pih*(xs-.5)*(2*n+1))/(2.*np.sin(pih*(xs-.5))),label="$x_0=0.
 ↪5$")
ax.plot(xs,np.sin(pih*(xs-.9)*(2*n+1))/(2.*np.sin(pih*(xs-.9))),label="$x_0=0.
 ↪9$")
ax.legend()


ax = axs[1]
n = 10
ax.set_title("$N=10$")
ax.plot(xs,np.sin(pih*(xs)*(2*n+1))/(2.*np.sin(pih*(xs))),label="$x_0=0.$")
ax.plot(xs,np.sin(pih*(xs-.5)*(2*n+1))/(2.*np.sin(pih*(xs-.5))),label="$x_0=0.
 ↪5$")
ax.plot(xs,np.sin(pih*(xs-.9)*(2*n+1))/(2.*np.sin(pih*(xs-.9))),label="$x_0=0.
 ↪9$")
ax.legend()

ax = axs[2]
n = 50
ax.set_title("$N=50$")
ax.plot(xs,np.sin(pih*(xs)*(2*n+1))/(2.*np.sin(pih*(xs))),label="$x_0=0.$")
ax.plot(xs,np.sin(pih*(xs-.5)*(2*n+1))/(2.*np.sin(pih*(xs-.5))),label="$x_0=0.
 ↪5$")
ax.plot(xs,np.sin(pih*(xs-.9)*(2*n+1))/(2.*np.sin(pih*(xs-.9))),label="$x_0=0.
 ↪9$")
ax.legend()


# fig.savefig("difraccion.png")
```

