Partição da unidade diferenciável

Praciano-Pereira, Tarcisio *

8 de abril de 2021
preprints da Sobral Matemática
no. 2021.01
Editor Tarcisio Praciano-Pereira
tarcisio@sobralmatematica.org

Resumo

Suponha que você tenha uma partição da unidade cujos átomos pertençam a uma certa classe de diferenciabilidade, digamos, sejam de classe $\mathcal{C}^n$, então é possível estender o método descrito em [4] de uma forma similar a um polinômio de Taylor por pedaços de grau $n-1$. Eu consegui rodar um programa mas com erros que vou expor neste artigo na esperança de obter colaboração.

palavras chave: partição da unidade diferenciável, polinômio de Taylor por pedaços, projetor de interpolação.

Suppose you have a partition of the unity whose atoms belong to differentiability class, say $\mathcal{C}^n$, then it is possible to extend the method of [4] to have piece wise Taylor polynomials of degree $n-1$. I have run a program but it has some errors and I will make them public in the hope to have collaboration.

keywords: differentiable partition of the unity, piece wise Taylor polynomials, interpolation projector.

* tarcisio@sobralmatematica.org

1 Plano do trabalho

1.1 É um pouco da história do trabalho

Este artigo foi publicado quando estava tentando encontrar parâmetros adequados para construir a aproximação splines duma função usando uma simulação dum polinômio do segundo tipo *polinômio de Taylor*, quer dizer usando os coeficientes de Taylor para um polinômio do segundo grau. Os resultados que eu vinha obtendo se mostravam satisfatórios porque eu conseguia fazer transformações que descrevo mais abaixo corrigindo o resultado. Acontece que eu consegui *sintonizar* os coeficientes o que vou descrever num próximo artigo deixando este aqui como está apenas para mostrar a história do processo, sem deixar registro do número de horas que gastei tentando encontrar uma saída que, finalmente, apareceu no meio da noite como aconteceu já muitas vezes comigo e consigo!

Se você sentir que o artigo está incompleto, a sensação é correta, eu parei de escrevê-lo e apenas não posso *despublicar* o que foi publicado e assim fica a história das tentativas registradas aqui. Pelo menos para mim, vale o registro.

Quero estender o trabalho descrito no artigo [4] agora sob a suposição de que tenho uma partição da unidade cujos átomos sejam altamente diferenciáveis. Que isto é possível está demonstrado no artigo [4] que se baseia na possibilidade de calcular-se a potência por convolução de ordem n, para um número natural qualquer o que foi feito no artigo, [2], em que Neves e eu conseguimos desenvolver com sucesso a potência arbitrária de ordem n da função característica do intervalo $[0, 1]$ e eu produzi um programa para criar as equações desta potência. Depois do artigo [2] já muito trabalho foi desenvolvido que me levou a registrar no artigo [4] em que eu desenvolvi uma técnica para obter as expressões algébricas dos splines o que resulta em melhor aproximação do que as expressões aproximadas que se podem obter com o programa de 2011,[2], que eu considero com uma parte importante da história. Mas agora o meu ponto de partida será o artigo [4], usando as expressões algébricas dos splines.

1.2 Plano do trabalho

Qualquer potencia por convolução da função característica $\chi_{[0,1}$ funciona como um *gerador* duma *unidade aproximada* de acordo com a definição de Rudin,[5]. Eu *estou deturpando* aqui o conceito de *unidade aproximada*, para Rudin, uma *unidade aproximada* é uma sucessão que converge fracamente para a medida de Dirac, a unidade da convolução. Eu estou aqui *deturpando* a definição de Rudin e chamando de *unidade aproximada* um dos geradores duma tal família, por exemplo $\chi_{[0,1]}$, uma função positiva cuja integral vale 1 é um exemplo. Qualquer potência por convolução de $\chi_{[0,1]}$ é também um exemplo. Esta *deturpação* é razoável porque

$$\eta \mapsto (x \mapsto R\eta(Rx); R \in \mathbf{N}^* \tag{1}$$

cria uma *unidade aproximada* de acordo com Rudin a partir duma *unidade aproximada* como estou definindo aqui. Em suma, para mim, uma *unidade aproximada* é um gerador daquilo que Rudin define como *unidade aproximada*.

Eu vou usar a transformação da equação (eq.1) com muita frequência.

Qualquer potência por convolução, $f_n = \chi^n_{[0,1]}$, de $\chi_{[0,1]}$ pode ser usada como um gerador duma sequência que convirja fracamente para a medida de Dirac, porém o suporte da enésima potência é $[0, n]$ o que a torna contraprodutiva para usar como um *unidade aproximada* como que eu quero usar aqui, uma função é uma *unidade aproximada* se sua convolução com a função g produz uma função próxima de g, e o suporte $[0, n]$ torna esta aproximação grosseira. A solução para este problema é usar a transformação

$$\eta(x) = Rf_n(Rx); R > n; \tag{2}$$

em que f_n é a enésima potência de $\chi_{[0,1]}$.

Que a leitora observe a diferença de R na equação (eq.2), com o seu significado na equação (eq.1), onde $R \in \mathbf{N}$ varia no conjunto dos números naturais maiores que zero. Na equação (eq.2), R é um número fixo e nem sempre é um número natural.

Aqui estou indicando $R > n$ em que n é a potência por convolução escolhida, mas quanto maior for R melhor será o resultado do ponto de vista de aproximação.

Vou introduzir alguma notação que já foi usada em [4] para tornar este artigo mais independente e não obrigar a leitora a recorrer à fonte externa para entender o conteúdo do presente trabalho.

Acho mais fácil apresentar um sistema de equações com alguns comentários e fazer em seguida uma lista de comentários mais longos que a leitora pode simplesmente pular se os julgar desnecessários, o que provavelmente será verdadeiro, mas pode ajudar a quem não tenha maior experiência nesta área, coisa que também me interessa, ampliar a audiência.

$$f_n = \chi^n_{[0,1]}; w = \tfrac{n}{2}; \eta(x) = f_n(wx); supp(\eta(x)) = [0,2]; \tag{3}$$

$$\eta \mapsto \rho; \rho(x) = \eta(w(x-a); a = -1; supp(\rho) = [-1,1]; \tag{4}$$

$$x_0 = a; \Delta x = \tfrac{b-a}{N}; x_k = x_0 + k\Delta x; \tag{5}$$

$$pu(f) = \sum_{k=0}^{N} f(x_k)\eta(w(x-x_k) \tag{6}$$

1. Diferente do que escrevi na equação (eq.2), eu cheguei à conclusão de que é mais prático trabalhar com a função η que é uma transformada de f_n apenas contraindo o suporte para $[0,2]$, equação (eq.3);

2. A equação (eq.4) é uma transformação típica de *wavelets* em que w funciona para contrair (ou expandir) o suporte e a é um coeficiente de translação que no jargão de *wavelets* é uma *dilação*. Este método se mostrou muito eficiente nas minhas experiências para localizar o *transformação do polinômio de Taylor*. Um valor alto para w localiza as ondas que são os átomos da partição da unidade o que me permitiu conseguir obter o efeito da aproximação pontual do *polinômio de Taylor* usando a expressão dum *projetor de interpolação* expresso na equação (eq.6).

3. Na equação (eq.5) eu escolhi $N+1$ pontos do intervalo $[a,b]$ criando uma partição deste intervalo.

4. Na equação (eq.6) eu defini um *projetor de interpolação*

A figura (fig. 2), página 3,

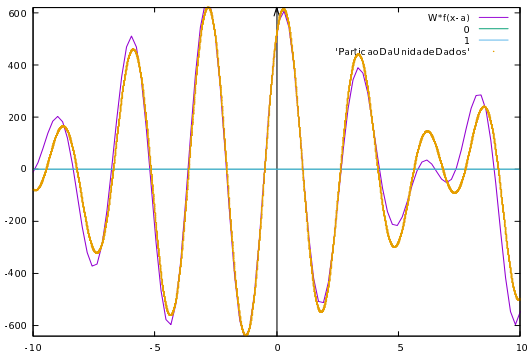

Figura 1:

foi editada usando os parâmetros

$$a = 0; W = 0.25; f(x) \mapsto W * f(x-a) \tag{7}$$

no comando do `plot` do `gnuplot` usando o arquivo produzido por uma programa escrito em `calc` cuja função principal dentro do programa fazia uma soma de cinco átomos sucessivos, em que 5 é a potência por convolução que usei para a função característica, confira [4].

Esta edição significa que não consegui obter uma imagem aceitável para a função que usei como exemplo no cálculo, mas que fazendo esta pequena edição transformei a imagem obtida numa imagem aceitável o que mostra que o método pode funcionar se eu conseguir corrigir o programa.

Vou descrever na próxima seção qual é a novidade entre [4] e a experiência aqui descrita.

A ideia não está ainda totalmente desenvolvida mas já consegui rodar um programa com um resultado promissor. Fazendo aproximação da função constante 1 obtive a imagem 1.001 usando $N = 20$ no intervalo $[-10, 10]$ com $\Delta x = 1$. O programa `ParticaoDaUnidade_2.calc` pode ser baixado de [3, programas]. O programa contém quatro funções usadas como teste em que três delas ficam sempre protegidas por comentários.

A função $f(x) = -(x-7)(x+10)\sin(2*x)$ é a minha preferida porque tem derivadas não nulas de qualquer ordem, é bastante oscilante para testar o algoritmo.

Baixando o programa, o comando para rodá-lo com os dados que estiverem já lançados é
`read 'ParticaoDaUnidade_2.calc'` dentro dum terminal do `calc`. As variáveis
`inicio,fim, N, potencia`
estão definidas no início do programa e o programa roda automaticamente a função
`main(inicio,fim,N,potencia)`
em que `potencia = 5` porque o programa está usando a quinta potência por convolução da função $\chi_{[0,1]}$. Então um novo comando
`main(-50, 50,50,potencia)`
é um comando legal e vai criar uma partição do intervalo $[-50, 50]$ com 100 subintervalos. As variáveis `inicio, fim,N,potencia` se encontram definidas no início do programa e o comando
`main(a,b,k,potencia)`
é legal para quaisquer números reais a, b com $a < b$ e um número inteiro positivo k que irá definir o número de nós da partição do intervalo $[a, b]$. Mas observe que não há interesse em que $\Delta x = \frac{b-a}{N}$ seja pequeno e eu estou trabalhando com o valor próximo de $\Delta x = 1$ que é o objetivo desta pesquisa, conseguir uma boa aproximação com poucos *pontos de precisão*.

1.3 O que consegui obter

Os próximos três gráficos mostram o que eu já consegui obter junto com os comentários explicando o que eles exibem. A figura (fig. 2), página 3, mostra o resultado do programa em que estou fazendo experiências para simular um polinômio de Taylor do segundo grau usando uma função que é um produto dum polinômio do segundo grau com uma senoide. A figura sugere que eu não consegui nenhum resultado. Mas como eu já sabia do erro, analisando a figura, observei que uma translação tornaria parte do gráfico de f um múltiplo do gráfico produzido pelo programa sugerindo a transformação

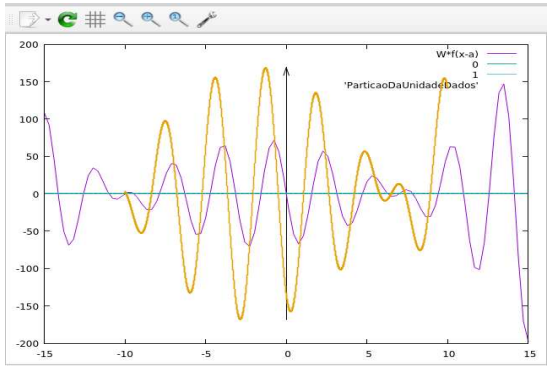

Figura 2:

$$f(x) \mapsto W * f(x-a) \tag{8}$$

e eu obtive os parâmetros a, W fazendo um dilatação no gráfico do `gnuplot`, em uma das ondas, resultando a medição em

$$a = -0.530171, W = 2.340197676; \tag{9}$$

Como o programa escreve uma lista de comandos para `gnuplot`, eu editei este arquivo de comandos usando os dados da equação (eq.9). O resultado se encontra na figura (fig. 3), página 4, ao lado qual se encontra um detalhe obtido fazendo um dilatação na figura do programa confirmando que o resultado é bom mas que eu preciso descobrir como associar "*coeficientes de Taylor*" junto com translações ade-

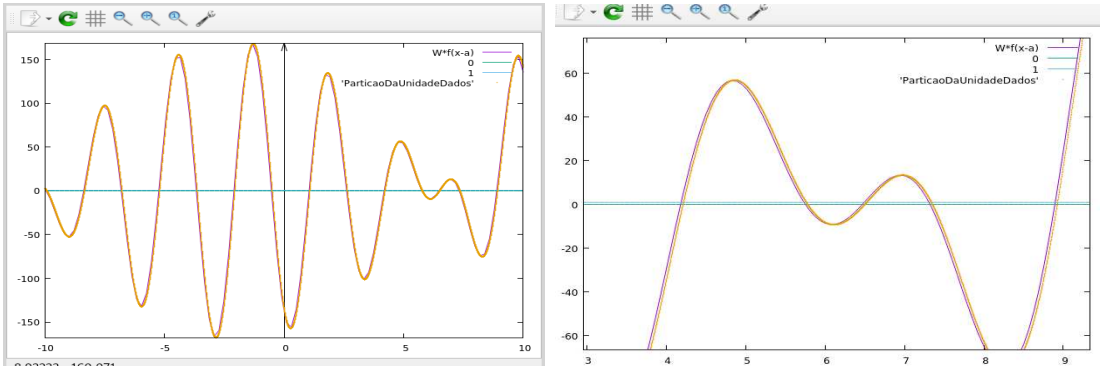

Figura 3:

quadas dentro do programa. Em outras palavras, tenho que corrigir o programa para que não seja preciso descobrir a, W como mostra a equação (eq.9).

2 Polinômio de Taylor e potências por convolução

A derivada da quinta potência por convolução é diferença entre duas transladas da quarta potência, mais exatamente, se f_5 for a quinta potência por convolução da função característica $\chi_{[0.1]}$ e se f_4 for a quarta potência por convolução da mesma função característica, então

$$\frac{df_5(x)}{dx} = f_4(x) - f_4(x-1); \tag{10}$$

$$\frac{df_5^2(x)}{dx^2} = f_3(x) - 2f_3(x-1) + f_3(x-2); \tag{11}$$

$$\frac{df_5^3(x)}{dx^3} = f_2(x) - 3f_2(x-1) + 3f_2(x-3) - f_2(x-4); \tag{12}$$

$$\frac{df_5^4(x)}{dx^4} = f_1(x) - 4f_1(x-1) + 6f_1(x-3) - 4f_1(x-4) + f_1(x-5); \tag{13}$$

em que os *índices inferiores* indicam potência por convolução. O modelo de derivação é $(a-1)^n$ em que n é ordem de derivação em que a potência é substituída por translação. A quarta derivada de f_5 é uma combinação linear de translações de $f_1 = \chi_{[0.1]}$ portanto não é contínua sendo f_5 de classe $\mathcal{C}^3$ ou mais genericamente, se f_m representa a $m-$ésima potência por convolução de $\chi_{[0.1]}$, será $m-1$-splines.

No programa eu considero a soma

$$f(x_0)\eta(w(x-x_0) + f(x_0)\eta'(w(x-x_0) + \frac{1}{2}f(x_0)\eta''(w(x-x_0) + \frac{1}{6}f(x_0)\eta'''(w(x-x_0) \tag{14}$$

em que x_0 varre cinco nós sucessivos, em que "*cinco*" é a potência por convolução usada, como verificado em [4] este é o número de átomos *úteis* quando a potência for a quinta. Então, para cada valor de x no intervalo em que eu estiver fazendo a simulação, eu estou calculando uma soma de cinco nós sucessivos, como eu já havia feito em [4] usando apenas a quinta potência por convolução. O meu objetivo é obter uma formulação do polinômio de Taylor do terceiro grau usando as derivadas dos átomos até a terceira derivada como está descrito na equação (eq.14).

Neste momento da experiência eu obtive um resultado animador porque uma edição do resultado sugere uma aproximação. Estou fazendo experiencias com os os parâmetro w, x_0 na expressão em que x_0 está representando a na transformada *wavelet*

$$w\eta(x-a) \tag{15}$$

em que η é um átomo obtido da quinta potência por convolução transformada para seu suporte seja $[-1, 1]$, equilibrado em torno de zero, e portanto equilibrado em torno de x_0 que percorre os nós da partição do intervalo $[a, b]$.

3 Resumindo e concluindo

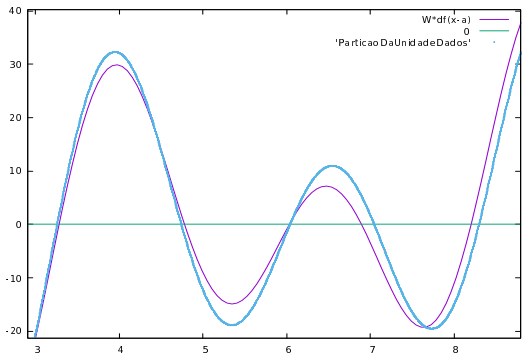

Figura 4:

O gráfico final mostra que consegui finalmente descobrir como interpretar os coeficientes de Taylor junto com as derivadas duma partição da unidade. Ele fecha este artigo que fica registrando uma sucessão de tentativas que mostravam que havia uma possibilidade resolver o problema porque aplicando as transformações de translação e ampliação eu consegui ajustar a imagem, entretanto ainda permanece uma distorção significativa nos extremos do intervalo visível na figura (fig. 4), página 5.

O trabalho prossegue para corrigir o programa e assim evitar a edição da imagem assim como na tentativa de reduzir ou eliminar a distorção no fim do intervalo que é típica da fórmula de Taylor.

Índice Remissivo

Referências

[1] David I. Bell Landon Curt Noll and other. Calc - arbitrary precision calculator. Technical report, http://www.isthe.com/chongo/, 2011.

[2] A.J. Neves and T. Praciano-Pereira. Convolutions power of a characteristic function. *arxiv.org*, 2012, April, 22:16, 2012.

[3] Tarcisio Praciano-Pereira. *Cálculo Numérico Computacional*. Sobral Matematica, 2007.

[4] Tarcisio Praciano-Pereira. Partição da unidade e projetores de interpolação. Technical report, Sobral Matemática, 2020.

[5] W. Rudin. *Functional Analysis*. McGraw-Hill Book Company, 1972.

[6] Thomas Williams, Colin Kelley, and many others. gnuplot, software to make graphics. Technical report, http://www.gnuplot.info, 2010.