SOLUÇÃO APROXIMADA DE EQUAÇÕES DIFERENCIAIS I

Partição da unidade splines

Praciano-Pereira, Tarcisio [*] Neves, Antônio Jorge[†]

VII Encontro de Pós-Graduação e Pesquisa - Outubro/2012

Resumo

Este é um artigo de uma série que mostra o uso de uma base de splines por convolução na construção de um projetor de interpolação. Mostramos também a potencialidade de uso desta ténica em duas áreas que devem ser nosso objetivo em trabalhos posteriores, expandir a definição do projetor de interpolação para obter a imagem num espaço de dimensão finita, de *Sobolev splines*, ou usar a base splines em séries wavelets.

palavras chave: base de splines por convolução, núcleos de convolução, séries wavelets

Classificação de assuntos da AMS-MSC: 47E05,41A05,42C40,46E35

This is a paper in a series which shows the use of a base of *splines by convolution* in the construction of an *interpolation projector*. We also show here the potentiality of the use this technique in two areas that should be our goal in later works, to expand the definition of the interpolation projector over a finite dimensional *Sobolev splines spaces*, or to use the base of convolution splines in a wavelet series.

keywords: convolution splines base, differential operator, convolution kernel, wavelet series.

AMS-MSC subject classification 47E05,41A05,42C40,46E35

[*]tarcisio@member.ams.org

[†]U Aveiro - Portugal jorgeneves@ua.pt

1 Introdução

Nesta primeira seção vamos apresentar parte do plano do trabalho, a notação que será usada neste artigo, que apesar de ser padronizada terá aqui alguns viezes particulares que vão facilitar a linguagem e mostrar alguns testes computacionais que guiaram o trabalho. Na última seção se encontram as perspectivas que temos para a continuação do trabalho.

A ferramenta básica é uma partição da unidade construída ao fazermos a *regularização por convolução* das funções características duma partição de um intervalo $[A, B]$, e para conseguir esta regularização, será usado um *núcleo com suporte compacto*, uma função positiva cuja integral é 1, e cuja classe de diferenciabilidade será herdada pela partição da unidade, como já foi feito, por exemplo em [6]. Ainda assim vamos entrar nos detalhes da construção porque vamos usar uma outra técnica.

Primeiro a análise da precisão do algoritmo.

1.1 Análise do algoritmo computacional

O símbolo f, irá representar a enésima potência por convolução de $\chi_{[0,1]}$, consistentemente, em todo este artigo. Em geral $n = 5$, mas algumas vezes faremos referência a uma potência genérica. Como não faremos uso da potência aritmética então usaremos a notação de "potência" para "*potência por convolução*".

Vamos designar por *núcleo* um elemento fixo de uma sequência que seja uma *unidade aproximada*, [9, ch 6, page 157], uma função positiva cuja integral é 1 e com alguma classe de diferenciabilidade cuja utilidade, neste trabalho, será sempre como fator na multiplicação por convolução para regularizar alguma função.

Um exemplo de tal núcleo é qualquer potência por convolução de $\chi_{[0,1]}$, f, e estaremos escolhendo a quinta potência por convolução apenas por comodidade no uso dos programas que serão usados como ferramentas neste trabalho, mas tudo que for feito aqui por ser facilmente obtido com qualquer outra potência e é fácil de verificar que os cálculos são independentes de $n = 5$. Além disto, em [5], foi estabelecido um algoritmo para construir f para qualquer que seja n. A única dependência de n é a classe de diferenciabilidade, a quinta potência de convolução da função característica do intervalo $[0, 1]$ é um 4-spline com suporte $[0, 5]$ sendo também um núcleo-4-spline, portanto de classe C^3. Este detalhe será importante, posteriormente, quando o método for aplicado no estudo de algum operador diferencial cuja ordem irá indicar quantas derivadas *do tipo função* devem ter os elementos da partição da unidade quando se deverá fazer a seleção adequada da potência.

Nesta secção inicial $[A, B] = [0, m]$ em que m é um inteiro e mais a frente vai representar o número de subintervalos da partição do intervalo $[A, B]$.

Vamos transformar f

$$\eta(x) = Rf(Rx); R = medida(supp(f)) = n = 5; supp(\eta) = [0,1]; \quad (1)$$

$$\rho(x) = \eta(x + \tfrac{1}{2}); supp(\rho) = [-\tfrac{1}{2}, \tfrac{1}{2}]; \quad (2)$$

$$\rho(x) = wf(w(x + \epsilon); supp(\rho) = [-\epsilon, \epsilon]; 2w\epsilon = R; \quad (3)$$

e será ρ o núcleo. A transformação, na equação (3), está condicionada à potência por convolução, $f = \chi^n_{[0,1]}$ portanto $R = n$ e $2w\epsilon = R$ para conseguirmos a precisão 2ϵ como medida do suporte do núcleo.

Na última seção vamos fazer referência a uso de outras *unidades aproximadas* quando discutirmos a possibilidade de evitar a condição de *uniformidade* nas partições dos intervalos.

Foi construida uma classe em `python` para representar os objetos matemáticos deste artigo, mas não vamos entrar em grandes detalhes porque a leitora interessada poderá consultar e analisar o código fonte, ou admitir a ferramenta "*como boa*", observando os dados aqui descritos. É preciso levar em consideração às limitações com que foram obtidos estes dados, por exemplo, um *processador* `Pentium(R) Dual-Core CPU T4400 @ 2.20GHz` de um laptop[1].

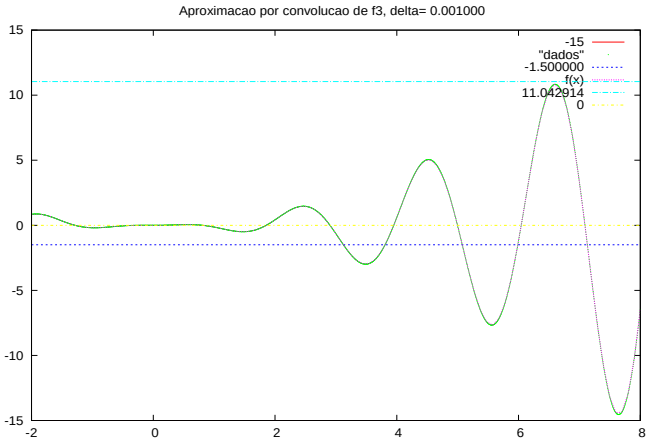

Figura 1: aprox. p/ convol. de $h_3(x) = sin(3x + 7)(\frac{x}{2})^2$

Os resultados 01,02,03 registrados na (Tabela 1.1), na página 3, são testes da "integral de Riemann", o primeiro é a integral de $h(x) = x^2$ no intervalo $[-1, 1]$ e os dois seguintes a verificação do núcleo e sua derivada. De 04 a 11 foi verificada a acuracidade da convolução, foi feita uma aproximação por convolução do coseno usando o núcleo ρ, isto é, foram impressos os valores do *coseno* e de $\rho * cos$, nos pontos $1, 3, 4, 7$, os erros observados são inferiores a 0.008 usando um passo 0.001 para calcular integrais com somas de Riemann que são muito pouco efetivas para cálculo aproximado, porem o objetivo neste momento ainda não é "precisão".

A conclusão deste quadro é que o algoritmo para o cálculo da convolução é confiável, entretanto, ele não é comutativo, no sentido de que para este algoritmo $f * g \neq g * f$, porque o algoritmo foi otimizado para ser usado com núcleos cujos suportes meçam no máximo 1. Obviamente, se

[1] tremendamente bem aproveitado pelo sistema `Debian/Gnu/Linux`

integral aproximada e convolução

01) integral de $h(x) = x^2$ em $[-1, 1]$	0.666667
02) integral do núcleo é	1.0
03) integral da derivada do núcleo é	$-1.06591124815e-15$
04) convolução no ponto 1 é	0.535817005817
05) o valor do coseno em 1 é	0.540302305868
06) convolução no ponto 3 é	−0.981774109695
07) o valor do coseno em 3 é	−0.9899924966
08) convolução no ponto 4 é	−0.648217421985
09) o valor do coseno em 4 é	−0.653643620864
10) convolução no ponto 7 é	0.747643762045
11) o valor do coseno em 7 é	0.753902254343

Tabela 1: Testes, integral aproximada e convolução

$supp(f)$, $supp(g)$ estiverem contidos em intervalos de medida 1 então o algoritmo será comutativo.

A figura (1) página 2, mostra os gráficos simultâneos de

$$h_3(x) = sin(3x + 7)(\frac{x}{2})^2$$

e de

$$h_3 * \rho_{10}; \rho_{10} : x \mapsto 10\rho(10x)$$

tendo sido o mesmo núcleo usado na aproximação do cosseno. Visualmente os dois gráficos se superpõem. Calculando $\|h_3(x) - (h_3 * \rho_{10})(x)\|$ no intervalo $[-10, 14]$, o erro máximo foi 0.0360500108807 com passo $delta = 0.2$ em `10.603s`, (tempo de uso do sistema $0.012s$) incluindo o tempo para obter o gráfico com `gnuplot`. A denominação "h_3" apenas indica que foram usadas diversas funções para testar o algoritmo tendo ficado finalmente a de índice 3 com melhor adequação visual.

Considerando que h_3 é uma função com alta taxa de variação (polinomial), e o suporte do núcleo mede 0.1, concluimos também com esta experiência que obtivemos uma aproximação razoável. Também fixamos a função h_3, em todo o artigo, para os testes numéricos, com o objetivo de ter uniformidade nos cálculos.

1.2 Construção de uma partição da unidade

Uma forma simples de produzir uma partição da unidade cujos elementos sejam splines está esquematizada nos cálculos que seguem, e as *legendas*, logo depois, justificam os passos.

$$f(x) = \chi^5_{[0,1]}; \eta(x) = wf(wx); medida(supp(\eta)) = 1; w = R = 5; \quad (4)$$

$$h_j(x) = h(x-j) = (h * \delta_j)(x); \tag{5}$$

$$\sum_{j=0}^{m} \chi_{[0,1]}(x-j) = 1 \ \text{ q.s.} \tag{6}$$

$$a = \eta * \chi_{[0,1]}; a_j(x) = a(x-j) = \eta * \chi_{[0,1]}(x-j); \tag{7}$$

$$\eta * (\sum_{j=0}^{m} \chi_{[0,1]}(x-j)) = 1; x \in \mathcal{V}_{[1,m]} \tag{8}$$

$$(a_j)_{j\in\{0,\ldots,m\}} = \eta * (\chi_{[0,1]} * \delta_k)_{k\in\{0,\ldots,m\}} = (a * \delta_i)_{i\in\{0,\ldots,m\}}; \tag{9}$$

Legendas:

- Na equação (4) mostramos a transformação da quinta potência por convolução, f, num núcleo, η, com suporte medindo 1. Isto vai possibilitar o uso do *algoritmo otimizado do cálculo de convoluções* na fase de testes, e também vai otimizar a parte do algoritmo na construção da partição da unidade, voltaremos a este ponto posteriormente. A classe de continuidade de f, η é C^3, ou seja, no caso genérico da enésima potência por convolução, teremos f, η de classe C^{n-2}.
- A equação (5) registra que a translação j de h, uma função genérica, é uma convolução com a medida de *Dirac*;
- Na equação (6) temos as somas das translações de $\chi_{[0,1]}$ pelos sucessivos inteiros $j = 0, \ldots, m$, resultando numa quasi-partição da unidade uma vez que a soma falha em assumir o valor 1 em cima dos nós inteiros da partição, um conjunto de medida zero. Como a convolução é distributiva relativamente à soma, a multiplicação por convolução por um núcleo contínuo regulariza esta soma transformando-a numa partição da unidade com a classe de continuidade impartida pelo núcleo multiplicador. Faremos isto usando η.
- A equação (7) mostra um dos elementos da partição da unidade, a função a, chamada de "*átomo*" neste artigo, é um 5-spline cujo suporte mede 2, suas translações por inteiros sucessivos j, formam uma partição 5-spline da unidade com nós inteiros. Estes átomos são também vetores básicos que geram um espaço de splines onde iremos projetar funções com um projetor de interpolação, [5, seção final].
- A equação (8) mostra a soma que aparece na equação (7) multiplicada por convolução pelo núcleo η e que agora é uma função de classe C^4, identicamente 1 sobre um aberto contendo $[1, m]$. Como vamos precisar de uma partição da unidade subordinada a um aberto contendo $[0, m]$, a solução consistirá do acréscimo de mais um átomo, a_{-1}, na soma, voltaremos a esta questão posteriormente.

- A equação (9) mostra que a translação $\underline{j}$ do átomo a é igual a convolução do núcleo η com a translação j da função característica $\chi_{[0,1]}$. Foi aumentada, assim, de uma ordem a classe de continuidade da partição da unidade, se trata de uma partição da unidade de classe C^4. Como o suporte do átomo a é um intervalo de medida 2, as interseções dos suportes das translações coincidem exatamente em um dos intervalos da partição o que vai nos permitir otimizar o algoritmo da soma destas translações: para cada valor de $x \in [0, m]$ a soma se reduz à duas parcelas:

$$(\forall x \in [0, m])\left(\sum_{j=0}^{m} a_j(x) = a_k(x) + a_{k+1}(x)\right); \tag{10}$$

  sendo k a parte inteira de x.

O uso de η transformou as translações de $\chi_{[0,1]}$, descontínuas, multiplicadas por convolução com η que é de classe C^3, numa família de de classe C^4. Se usarmos $f = \chi^n_{[0,1]}$ teremos uma partição da unidade de classe C^{n-1}.

Uma das vantagens do método reside no facto de que, em vez de *calcular convoluções*, vamos trabalhar com *translações de uma convolução que já foi calculada*, ver [5], que são os elementos que aparecem na equação (10), ou transformações deste elemento para obter uma determinada precisão de acordo com o próximo item.

A partição que estamos construindo é relativa a um intervalo com nós inteiros, entretanto, uma *mudança de variável* adequada

$$medida([A, B]) = B - A; R = 5 = medida(supp(f)); w = \tfrac{mR}{B-A}; \tag{11}$$

$$\eta(x) = wf(wx); medida(supp(\eta)) = \tfrac{B-A}{mR}; \tag{12}$$

$$\text{precisão desejada } medida(supp(\eta)) = \epsilon = \tfrac{B-A}{mR}; \tag{13}$$

irá nos fornecer a precisão desejada, em que o inteiro m é o número de subintervalos da partição $\Pi_m([A, B])$ e $R = n$ é medida do suporte da potência f escolhida.

As equações (11)-(13) nos fornecem os elementos para obter uma determinada precisão.

Os cálculos ficam mais simples se trabalharmos com nós inteiros o que continuaremos a fazer, entretanto, no programa associado a este artigo, que estará disponível e será de publicado sob a GPL, estas transformações foram usadas permitindo que se obtenha uma partição da unidade com classe arbitrária de continuidade e num intervalo qualquer, a entrada de dados será: (A, B, n, m) identificando, respectivamente,

- o intervalo $[A, B]$;

- a potência de convolução desejada, com observação de que a constante R na equação (11) coincide com potência n;
- o número m de subintervalos da partição de $[A, B]$.

2 Projetor de interpolação

Considerando o exposto na seção anterior, vamos construir agora projetor de interpolação associado a uma partição com nós inteiros do intervalo $[0, m]$. Os passos são os seguintes e eles estão baseados no sistema de equações(4)-(9).

2.1 O projetor $PU(h_3)$

A forma mais simples de justificar o método consiste de um exemplo trivial mas cuja generalização consiste de nossa presente construção. O gráfico na figura (2), página 6, é a interpolação linear de pontos sele-

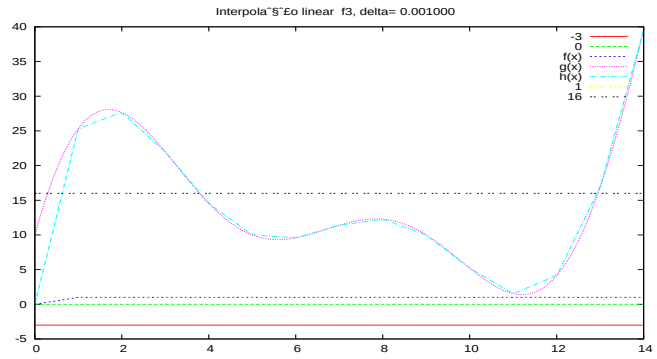

Figura 2: Projetor de interpolação: interpolação linear

cionados sobre o gráfico de

$$h_3(x) = sin(x/2)(x-5)(x-9) + 10) \tag{14}$$

que escolhemos fazer usando um exemplo concreto, mas adredemente escolhido para permitir a generalização posterior!

- Criamos um átomo $a(x)$

$$a(x) = \begin{cases} 0 & \Leftarrow \quad x < 0; \\ x & \Leftarrow \quad x < 1; \\ 2 - x & \Leftarrow \quad x < 2; \\ 0 & x > 2; \end{cases} \tag{15}$$

uma função com suporte $[0, 2]$. As translações de $a(x)$ de uma unidade vão fazer com que os piques de uma coincida com o ponto

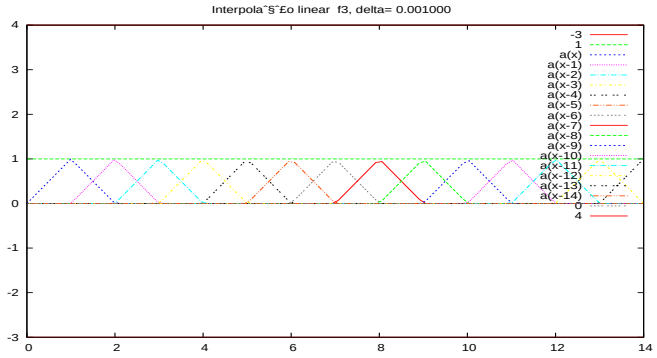

Figura 3: Translações de um atomo a suporte compacto

de mínimo da seguinte produzindo o gráfico na figura (3), página 7,

- A soma dos átomos a vai ser uma reta (soma de segmentos de reta é uma reta), a função constante 1. Logo temos uma partição da unidade formada pelo sistema de translações de a que pode ser vista também na figura (3).
- A interpolação linear é a soma das translações quando usamos os coeficientes $h_3(i)$ aplicado em cada uma das parcelas $a(x-i)$.
- Uma forma simples de explicar seria que a função

$$PU(h_3)(x) = \sum_{j=0}^{m} h_3(j)a_j(x) \tag{16}$$

foi obtida pela multiplicação por convolução da soma das funções características dos intervalos $([j, j+1])_{j=0}^{m-1}$ pelo núcleo $\chi_{[0,1]}$.

Neste ponto mostraremos como fazer a generalização do exemplo simples que escolhemos. Se em vez do núcleo $\chi_{[0,1]}$, usarmos o núcleo $\eta(x) = nf(nx)$ em que f é n-ésima potência por convolução de $\chi_{[0,1]}$, na equação (16), o resultado será uma partição da unidade por que regularizamos a quase partição da unidade formada pelas translações inteiras de $\chi_{[0,1]}$, mas resultando agora numa soma de funções de classe C^{n-1}. Os átomos a_j, independentemente de qual seja a potência de $\chi_{[0,1]}$, terão como suporte um intervalo medindo 2 o que torna a essencia do algorítmo aproveitável para qualquer potência n escolhida.

Mas o que nos interessa neste momento é a soma

$$h_3 \mapsto PU(h_3) = \sum_{j=0}^{m} h_3(j)a_j \tag{17}$$

que forneceu a interpolação linear de pontos escolhidos com coordenadas inteiras sobre o gráfico de h_3. Esta expressão define um projetor de interpolação de um certo espaço de funções a que h_3 pertence, no espaço vetorial das funçoes geradas pelas translações $(a_j)_{j=0}^m$ que é um espaço de 1-splines. Observando que este exemplo usa a segunda potência por convolução de $\chi_{[0,1]}$, os cálculos efetuados neste exemplo se generalizam *verbatim* para o caso geral em que η se deduz da enésima potência por convolução de $\chi_{[0,1]}$, conforme dissemos no ínicio deste parágrafo, construimos o exemplo cuja generalização é o caso geral descrito neste artigo.

Chegamos assim à expressão do projetor de interpolação que nos interessa:

Definição 1 (Projetor) *de interpolação*

Considere o espaço vetorial $\mathcal{SP}_m([A.B])$ gerado pelas translações dos átomos a definidos na equação (7).

$$h \mapsto PU(h) = \sum_{j=0}^{m} h(j)a_j \tag{18}$$

define um projetor de interpolação do espaço das funções a que pertença a função h no espaço $\mathcal{SP}_m([A.B])$. Como $a_j(j) = 1$, então $PU(h)(j) = h(j)$, ou seja, h e $PU(h)$ coincidem nos nós inteiros do intervalo $[0, m]$ que são os pontos de precisão do projetor PU.

Em geral vamos querer que h seja um elemento de um espaço de funções diferenciáveis soluções de uma equação diferencial, mas o método também se aplica para obter interpolações de dados amostrais, neste caso $h(j)$ seriam os elementos do levantamento em questão.

A nossa construção privilegia os pontos médios dos suportes das translatadas dos átomos a que são chamados na literatura de *pontos de precisão*.

3 Trabalho em andamento

Com a ressalva de que estamos descrevendo um trabalho em andamento e que ainda é cedo para uma apresentação muito detalhada do formato final que venha a ter de algum dos problemas que nos interessam, é possível, entretanto, indicar algumas questões interessantes. O grande objetivo será o estudo de operadores diferenciais.

Todo o trabalho feito até agora tem sido acompanhado por processos computacionais, programas que realizam, com sucesso, as etapas alcançadas, traduzindo em termos algoritmicos o formalismo matemático descrito em nossos trabalhos. Temos o *código fonte* para apresentar, e parte dele já se encontra publicado, [8].

Em [5], construímos a potência por convolução de ordem arbitrária n para a função característica do intervalo $[0, 1]$ que mostramos sendo utilizada na construção de um projetor de interpolação com uma evidência computacional da eficiência do método. Neste mesmo trabalho apresentamos um algoritmo que nos oferece todas as derivadas do tipo *função* desta potência arbitrária da função característica.

Um único defeito nesta construção é a dependência de uma partição uniforme do intervalo em que se esteja trabalhando, é um defeito é considerável porque os problemas interessantes ocorrem quando hajam singularidades o que nos obriga a criar malhas com partições não uniformes no processo de aproximação, ver [3], por exemplo.

Em [3], os autores estão trabalhando com uma técnica ligeiramente diferente desta que estamos empregando aqui, por outro lado, nós temos a nosso favor a potencialidade de construir uma base para um espaço de dimensão finita arbitráriamente grande, m, dum espaço de Sobolev $H(\Omega)^n$, onde se encontre definido um operador diferencial, assim como um projetor de interpolação de $H(\Omega)^n$ em $\mathcal{SP}_m(\Omega)$, em que n é uma potência de convolução e m é a dimensão do espaço de splines. Nós já temos os elementos da base deste espaço, na forma da partição da unidade descrita nas seções anteriores, com a observação de que a passagem do caso univariado para o caso multivariado pode ser feita de pelo menos duas maneiras:

- No caso bivariado, considerando

$$\gamma(x, y) = a(x)a(y); \tag{19}$$

  em que a é o átomo na construção univariada, e γ seria o átomo no caso bivariado. o elemento a ser translatado, na construção da base do espaço vetorial de splines. O caso multivariado seria seria semelhante.

- possivelmente a construção direta das potências de convolução de funções características de retângulos, entretanto as tentativas feitas se revelaram infrutíferas, mas isto também aconteceu durante quase três meses de trabalho que precederam a publicação de [5].

Retornando à comparação da nossa metodologia com a dos autores em [3], uma falha importante no nosso trabalho é a dependência de partição uniforme. Entretanto em [7] um dos autores conseguiu, de uma forma artesanal, a construção de uma base de splines sem a restrição da partição ser uniforme, e no caso univariado, mostrou como seria possível eliminar a hipótese de uniformidade das partições. Temos evidências computacionais que nos sugerem que é possível eliminar o aspecto artesanal de [7] que estão relacionadas com a seleção de fator ρ, que neste trabalho é um núcleo equilibrado (mais exatamente, simétrico) em volta

da origem, por um fator cujo suporte seja $[0, 1]$ ou $[-1, 0]$ que é o ingrediente "artesanal" usado em [7] embora o autor não o tenha, convenientemente, formalizado.

Foi chamada nossa atenção para um trabalho dos autores *J.M. Melenk and I. Babuska*, [4], mas não tivemos ainda a oportunidade de analisá-lo.

Uma outra linha de trabalho está associada com o em que os elementos construídos em [5] podem ser usados com "*wavelets*". Este linha de trabalho se encontra plenamente em nossos objetivos uma vez que já temos os algoritmos computacionais para iniciar as experiências.

Finalmente também temos alguma evidência computacional de que os elementos aqui anunciados podem ser úteis dentro da linha do trabalho [10]. Os programas que rodamos ainda apresentavam erros signficativos, o que nos afastou momentaneamente desta linha de trabalho, entretanto, conseguimos, recentemente, otimizar um dos algoritmos que é essencial para o nosso trabalho, o da convolução, que tem um uso preliminar antes de efetuar as contas formais com uma convolução e isto nos tem ajudado a encontrar a direção certa.

4 Agradecimentos

Este trabalho é a expansão da conferência apresentada pelo primeiro autor no *Seminário de Grupo de Functional Analysis and Applications* do CIDMA, dep de Matemática - Universidade de Aveiro. O primeiro autor quer aqui agradecer a hospitalidade do Dep. de Matemática da U.A. que lhe tornou possivel obter os resultados aqui mencionados, em particular com referência a [5]. Também é preciso incluir neste agradecimentos o apoio financeiro aportado pela Universidade Estadual Vale do Acaraú na forma do salário sob a licença para este mesmo autor pudesse participar dos referidos seminários na U.A. onde todos os resultados aqui mencionados foram obtidos, assim como ao Colegiado do Curso de Ciências da Computação da Universidade Estadual Vale do Acaraú que autorizou a licença e o afastamento.

Referências

[1] Latour V. Dahlke S., Dahmen W. Smooth refinable functions and wavelets obtained by convolution. *Applied and Computational Harmonic Analysis*, 2 (1):68–84, 1995.

[2] Delvos F-J and Schempp Walter. *Boolean methods in interpolation a approximation*. Pitman research notes in Mathematics, 1989.

[3] Awanou G., Lai Ming-Jun, and P. Wenston. The multivariate spline method for numerical solution of partial differential equations and scattered data interpolation. *in Wavelets and Splines: Athens 2005, edited by G. Chen and M. J. Lai, Nashboro Press*, in Wavelets and Splines: Athens 2005, edited by G. Chen and M. J. Lai, Nashboro Press, 2006:24–74, 2006.

[4] J.M. Melenk and I. Babuska. The partition of unity finite element method: Basic theory and applications. *Seminar fur Angewandte Mathematik Eidgenossische Technische Hochschule CH-8092 Zurich Switzerland*, Research Report No. 96-01:23, 1996.

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

[6] T Praciano-Pereira. Splines por convolução. *Préprints da Sobral Matemática*
http://www.sobralmatematica.org/preprints, 2007.09:10, 2007.

[7] T. Praciano-Pereira. An interpolation projector associated to a non uniform partition. *Préprints da Sobral Matemática*
http://www.sobralmatematica.org/preprints, 2008.08:10, 2008.

[8] T. Praciano-Pereira. Convolução de funções características em r^2. *Préprints da Sobral Matematica*
http://www.sobralmatematica.org/preprints, 2011.05:10, 2011.

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

[10] et al. S. Pooseh. Approximation of fractional integrals by means of derivatives. *Computers and Mathematics with Applications*, 2012.