Tecnologia Chebyshev

Palavras-Chave: INTERPOLAÇÃO, MATLAB, CHEBYSHEV, CHEBFUN

Autores(as):

MARCELLO HENRIQUE FERREIRA DOS SANTOS, IMECC– UNICAMP

Prof(a). Dr(a). LUCIO TUNES DOS SANTOS (orientador), IMECC – UNICAMP

INTRODUÇÃO:

A interpolação polinomial é um método fundamental em análise numérica, utilizado para encontrar um polinômio que passe exatamente por um conjunto de pontos (x, y) dados. Em outras palavras, a interpolação polinomial procura um polinômio de grau adequado tal que o polinômio $P(x)$ satisfaça $P(x_i) = y_i$ para cada ponto (x_i, y_i) do conjunto de dados. Entre as várias técnicas de interpolação polinomial, destacam-se o método de Newton e a forma de Lagrange.

A unicidade desse polinômio interpolador é garantida por um teorema da álgebra, que afirma que existe um e apenas um polinômio de grau n ou menor que assume valores específicos para $n + 1$ valores de x.

No entanto, para algumas funções, como a função de *Runge* $f(t) = \frac{1}{1+(kt)^2}$, onde k é uma constante real, ocorre o fenômeno de *Runge*. Esse fenômeno descreve o aumento do erro entre o polinômio interpolador e o valor da função nos extremos do intervalo de interpolação, um problema que tende a se agravar com o aumento do grau do polinômio.

Para mitigar esse erro, uma alternativa é escolher cuidadosamente o conjunto de pontos a serem interpolados. Pontos igualmente espaçados podem não produzir os melhores resultados comparados a pontos acumulados nos extremos. Um conjunto eficiente desses pontos são os chamados pontos de Chebyshev, que formam a base deste trabalho.

Em 2002, *Zachary Battles* e *Nick Trefethen* desenvolveram para MATLAB um pacote chamado Chebfun, com o objetivo de lidar com problemas envolvendo funções matemáticas. O Chebfun representa funções utilizando a expansão em série dos polinômios de *Chebyshev*, buscando alcançar para funções o que a aritmética de ponto flutuante alcança para números: cálculos rápidos e precisos, com erros de arredondamento mínimos.

Este trabalho estudará a utilidade dos pontos de *Chebyshev*, explicando por que eles funcionam bem e comparando o pacote *Chebfun* com outros métodos para medir sua eficiência.

METODOLOGIA:

O projeto é baseado em estudar o *Chebfun*, um pacote *MATLAB* que expande as capacidades da linguagem para funções contínuas e operadores, logo, a linguagem usada no texto foi o *MATLAB* e assim os resultados provenientes do pacote puderam ser validados e posteriormente comparados com algum método análogo já presente no *MATLAB*.

A primeira parte do projeto consiste em entender como o pacote funciona e a parte matemática e a segunda em entender melhor os métodos e a sua usabilidade.

O pacote *Chebfun* é baseado em escrever as funções usando polinômios de *Chebyshev*. Para um conjunto de pontos (x_j, f_j), com x_j os pontos de *Chebyshev* (de segunda espécie) definidos por:

$$x_j = \cos\left(\frac{\pi k}{n}\right)$$

para $k = 0, 1, \dots, n$ e n é a quantidade de pontos e $f_j = (-1)^{n-j}$, o polinômio $p(x)$ que interpola esses pontos será exatamente o polinômio de *Chebyshev* $T_n(x)$, uma observação aqui é que a escolha da quantidade de pontos está relacionada com o grau do polinômio interpolador que será obtido. A “sacada” que o pacote faz é que, ao invés de representar polinômios, que geralmente são representados na base usual de monômios $\{1, x, x^2, \dots\}$, ele os representa na base $\{T_0(x), T_1(x), T_2(x), \dots\}$, por exemplo, podemos representar o polinômio $q(x) = 2x\text{^}2 + 3x + 1$ por $q(x) = T_2(x) + 3T_1(x) + 2T_0(x)$, uma vez que $T_0(x) = 1, T_1(x) = x$ e $T_2(x) = 2x^2 - 1$. Esse mesmo processo é feito para funções que não são exatamente polinômios, como por exemplo, $f(x) = e^x$, nesse caso, o pacote usa a série de Taylor, que é um polinômio, para transformar a base e fazer a aproximação. Além disso, uma das vantagens é que não é preciso saber o número de pontos para fazer a interpolação, uma vez que o próprio pacote se encarrega de escolher o melhor n que reduz o erro da interpolação.

A segunda parte do projeto focou sobre entender as possibilidades e facilidades, assim como as limitações, trazidas pelo pacote, entre elas estão encontrar derivadas, integrais, resolver problemas que envolvem aritmética complexa, trabalhar com equações diferenciais e problemas de valor inicial e de contorno.

A estrutura básica do Chebfun são os chebfuns, que não são nada além da forma que o computador entende e discretiza as funções usando polinômios de *Chebyshev*. Há várias formas de construí-los, a mais usual é definir a variável independente como um *chebfun* e usá-la para definir as funções, para isso o usuário deve usar comando “`chebfun`” da seguinte forma:

```
>> x = chebfun('x',[-5,5]);
>> f = sin(5*x);
```

Figura 1 - Construção de um chebfun.

Na figura acima, foi construída a função $f(x) = \sin(5x)$, a partir da construção da variável x como um chebfun, essa função está localizada na variável f, que é um chebfun porque realizar operações em um chebfun ainda é um chebfun, usando esse modelo, tem-se uma forma de se escrever o comando para funções matemáticas.

RESULTADOS E DISCUSSÃO:

Com os métodos proporcionados pelo pacote, pode-se apresentar alguns dos resultados trazidos por ele e compará-los com alguns métodos análogos no *Chebfun*, para os casos que existem.

Uma das comparações que se pode fazer com o Chebfun é a integração, que é representada pelo comando $sum(f)$, onde f é a função construída através do *Chebfun*, considerando $f(x) = e^x \sin(x)$, e computando a sua integral no intervalo especificado, [0,1], quando este não for especificado na criação será usado o intervalo de definição do chebfun, no caso do exemplo abaixo é [-2,2], e comparando com o valor exato da integral, é possível perceber uma acurácia de 15 casas decimais.

O comando para realizar o cálculo dessa integração e do valor exato está ilustrado na imagem abaixo:

```
x = chebfun('x', [-2,2]);
f = exp(x)*sin(x);

int_f = sum(f,[0,1]);
int_exata = exp(1)*(sin(1)-cos(1))/2 - exp(0)*(sin(0)-cos(0))/2;

dif = int_exata - int_f;
```

```
Saída:
int_f = 0.9093306736314782, int_exata = 0.9093306736314786
e dif = 0.0000000000000003
```

Figura 2 - Utilização do método "sum" e comparação com seu valor exato.

Essa é apenas uma das possibilidades do pacote, outros métodos são descritos ao longo do texto.

CONCLUSÕES:

O pacote *Chebfun* é um pacote poderoso quando se trata de implementações computacionais pela sua versatilidade de tratar diferentes assuntos e também pela eficiência que traz consigo para resolver os problemas de forma simplificada, rápida e acurada.

Por ser um pacote relativamente novo, ele está constantemente em atualização, atualmente na versão 5, e os seus desenvolvedores continuam o aprimorando para resolver problemas cada vez mais complicados e complexos, como é o caso do *Chebfun2*, que foi montado com o propósito de lidar com funções de duas variáveis definidas num retângulo $[a, b] \times [c, d]$ e é capaz de computar manipulações algébricas, realizar otimizações, calcular integrações, encontrar zeros, além de possibilitar o usuário operar o cálculo vetorial.

BIBLIOGRAFIA

TREFETHEN, L.N. **Approximation Theory and Approximation Practice**. SIAM, 2018.

GREENBAUM, A. & CHARTIER, T.P. **Numerical Methods**. Princeton, 2012.

DRISCOLL, T. A., HALE, N., & TREFETHEN, **L. N. Chebfun Guide**. Pafnuty Publications, Oxford, 2014.

CAROTHERS, N. L. **A short course on approximation theory**. Bowling Green State University.

MUDDE, M.H. **Chebyshev approximation**. University of Groningen, 2017.

BATTLES, Z., & TREFETHEN, L. N. **An extension of MATLAB to continuous functions and operators**. SIAM Journal on Scientific Computing, 2004.