



## **Relatório do Lab Extra 1 de CCI-22**

### **Trabalho Extra 01**

**Aluno:**

Bruno Costa Alves Freire

**Turma:**

T 21.4

**Professor:**

Luiz Gustavo Bizarro Mirisola

**Data:**

24/04/2018

**Instituto Tecnológico de Aeronáutica – ITA**  
**Departamento de Computação**

# 1. Épsilon Generalizado

Foi implementada a função  $\text{eps}_x(x)$  no MATLAB, a qual fornece para um dado  $x$  o valor do menor número  $\varepsilon(x)$  tal que  $x + \varepsilon(x) > x$ . Em seguida, essa função foi utilizada para calcular o épsilon de todos os números da forma  $10^i$ , com  $i$  variando de -300 a 300. Por fim, esses dados foram plotados num gráfico  $\varepsilon(10^i) \times i$ , com escala logarítmica no eixo  $y$ , conforme a figura 1:

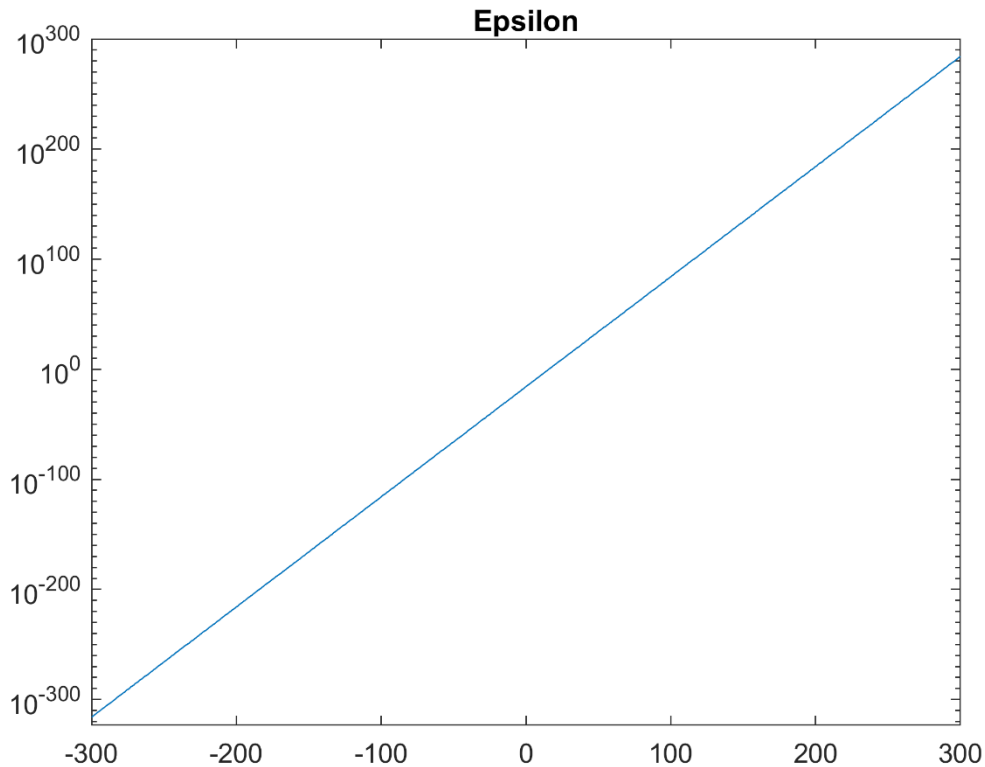


Figura 1: Gráfico  $\varepsilon(10^i) \times i$

Como se pode verificar, a curva obtida é uma reta. A explicação para esse formato reside no fato de que o épsilon de um número em representação em ponto flutuante, segundo o padrão IEEE-754, equivale ao espaçamento mínimo entre dois números “consecutivos”, isto é, a diferença entre um dado número, em módulo, e primeiro número representável imediatamente maior em módulo.

Acontece que por termos uma representação baseada em expoente e mantissa, essa diferença acaba sendo sempre uma potência de 2, diretamente associada ao expoente do número. Se um dado número  $x$  for representado com  $n$  bits em sua mantissa, e seu expoente for  $m$ , então o valor de épsilon para esse número será  $2^{m-n}$ . Dessa forma, se calculamos  $\varepsilon$  para uma progressão geométrica, isto é, onde o expoente dos números cresce em progressão aritmética, obteremos uma curva exponencial. Aplicando uma escala logarítmica ao eixo  $y$ , obteremos então uma reta.

## 2. Interseção de Cônicas

Para testar a função que encontra interseções entre cônicas (quádricas no plano), foram construídos dois casos de teste:

- Com a circunferência  $\Gamma: x^2 + y^2 - 4 = 0$ , e a parábola  $\Pi: x^2/4 + y = 2 + 10^{-10}$ , com  $(x_0, y_0) = (0.3, 1.8)$ ;
- Com a hipérbole  $\Omega: x^2 + xy - y^2 - 3 = 0$ , e a parábola  $\Psi: y^2 - x + y + 1 = 0$ , com  $(x_0, y_0) = (1, 0)$ ;

As soluções dos sistemas não-lineares envolvidos, através do método de Newton, estão plotadas nos gráficos abaixo. O script utilizado para definir as quádricas e gerar os gráficos está no arquivo `quadtest.m`.

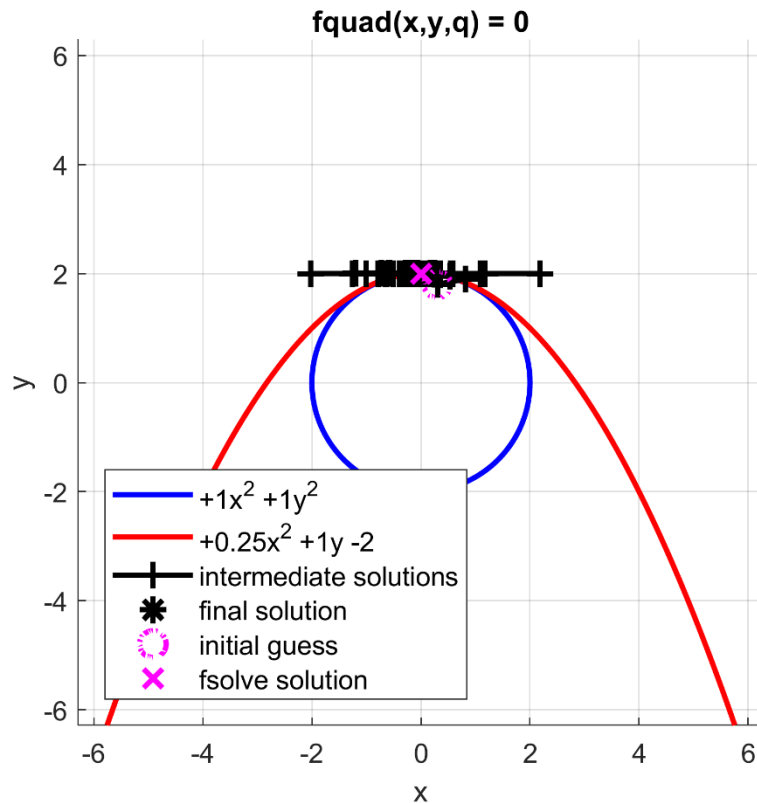


Figura 2: Caso a) quase interseção

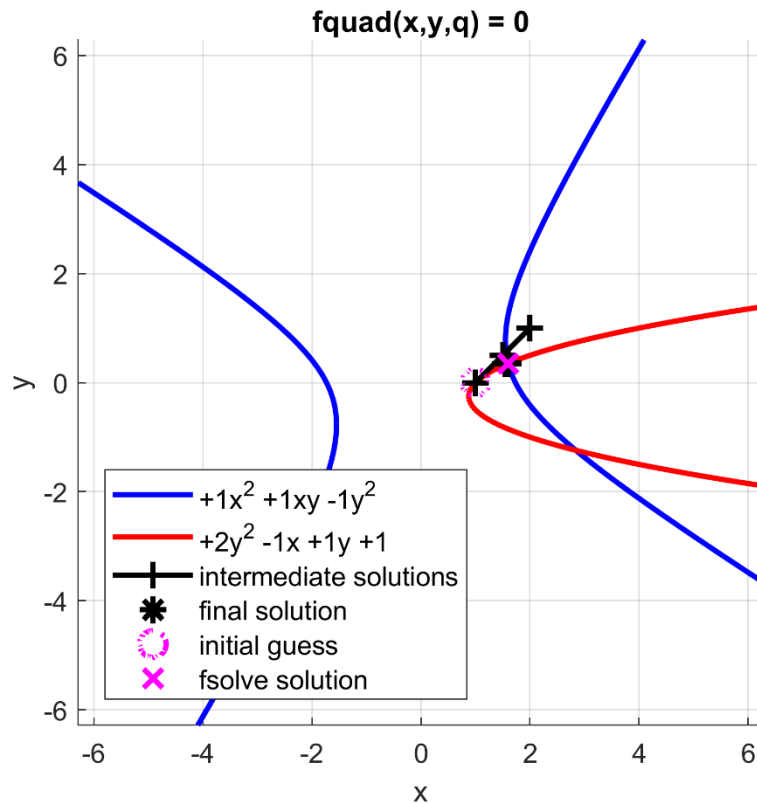


Figura 3: Caso b) dupla interseção

Note que no caso a), o sistema não possui solução, no entanto, existe um ponto suficientemente próximo das duas curvas para que seja encontrado como solução. Já para o caso b), as curvas se intersectam de fato em dois pontos.

### 3. Matriz Inversa

Para calcular a matriz inversa, a função `myinv.m` foi implementada utilizando os scripts desenvolvidos no Lab 3, `DecomposicaoLU.m` e `SolucaoLU.m`, para resolver sucessivos sistemas lineares e encontrar a matriz inversa de uma matriz dada.

Após implementada a função `myinv.m`, esta foi testada com 6 matrizes e comparada com a função nativa do MATLAB, `inv`. As matrizes testadas foram criadas da seguinte maneira: A, B e C são bem condicionadas de tamanhos 4, 8 e 16, respectivamente; enquanto D, E e F são matrizes de Hilbert, de tamanhos 4, 8 e 16, respectivamente.

Para mensurar a exatidão dos cálculos, computou-se o valor do maior elemento da diferença em módulo entre a matriz identidade e o produto  $M \cdot M^{-1}$ , quando a inversa é calculada pelo script implementado e quando se utiliza a função pronta do MATLAB. Os resultados foram compilados na Tabela 1:

| <b>Matriz</b> | <b>Erro usando <code>inv</code></b> | <b>Erro usando <code>myinv</code></b> |
|---------------|-------------------------------------|---------------------------------------|
| <b>A</b>      | 0.444089209850063*e-15              | 0.444089209850063*e-15                |
| <b>B</b>      | 0.133226762955019*e-13              | 0.106581410364015*e-13                |
| <b>C</b>      | 0.383026943495679*e-14              | 0.199840144432528*e-14                |
| <b>D</b>      | 0.227373675443232*e-12              | 0.113686837721616*e-12                |
| <b>E</b>      | 0.894069671630859*e-07              | 0.596046447753906*e-07                |
| <b>F</b>      | 2.750000000000000                   | 1.750000000000000                     |

Tabela 1: Comparação entre `inv` e `myinv`

Pode-se observar que o erro obtido ao utilizar `myinv` é menor ou igual ao obtido com `inv` em todos os casos, indicando uma provável diferença de implementação que reflete na propagação de erros.

## 4. Base de Chebyshev

O método de interpolação por polinômios de Chebyshev é uma variação do método de Vandermonde, onde em vez de usar uma matriz formada por vetores da base canônica do espaço dos polinômios, utiliza-se a base de Chebyshev no lugar. Por conta dos polinômios de Chebyshev serem bastante distintos entre si, a matriz de Vandermonde formada com eles tende a ser mais bem condicionada que a matriz original, que emprega monômios da base canônica. Uma vez obtidos os coeficientes do polinômio interpolador na base de Chebyshev, esse polinômio é então reescrito na base canônica, no formato esperado pela função `polyval` do MATLAB.

Após implementado esse método, foram realizados testes com duas funções:

- A função de Runge,  $f(x) = \frac{1}{1+25x^2}$ ;
- A função  $g(x) = e^x \cdot \sin 5x$ ;

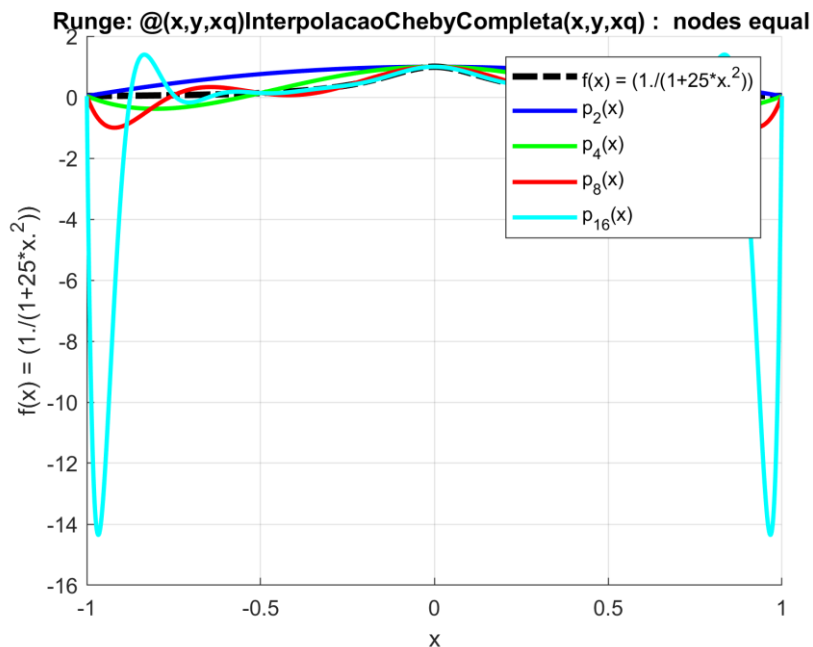
Sob duas circunstâncias: utilizando nós de interpolação linearmente espaçados e utilizando nós de Chebyshev.

Os mesmos testes foram feitos com outros métodos no Lab4, cujos resultados serão apenas anexados aqui em conjunto com os do método de Chebyshev.

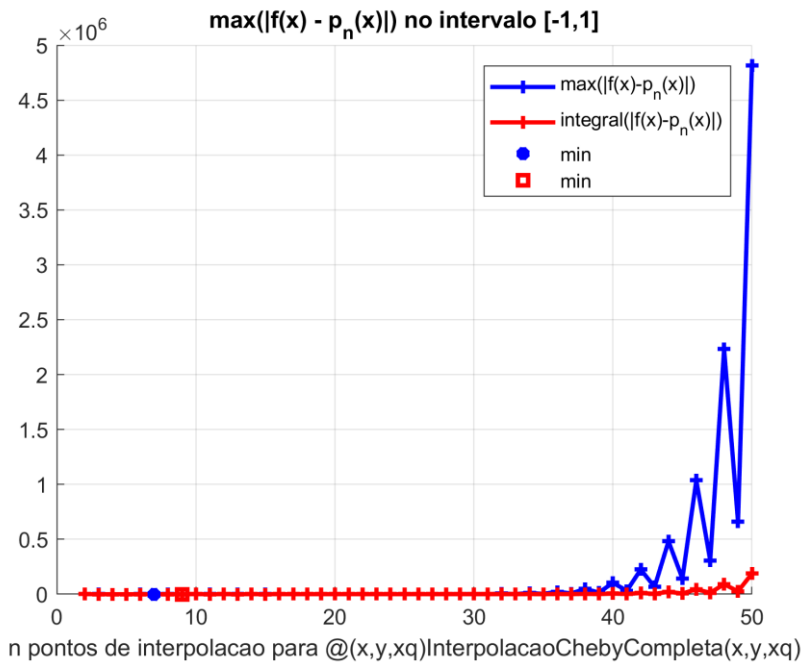
| Função de Runge              | Nós       | Min E(n)   | N para min E(n) | Converge para |
|------------------------------|-----------|------------|-----------------|---------------|
| <b>PolinomioInterpolador</b> | Regular   | 0,24736    | 7               | $\infty$      |
| <b>InterpolacaoNewton</b>    | Regular   | 0,24736    | 7               | $\infty$      |
| <b>SplineCubica</b>          | Regular   | 1,11883e-4 | 50              | 0             |
| <b>InterpolacaoChebyshev</b> | Regular   | 0,24736    | 7               | $\infty$      |
| <b>PolinomioInterpolador</b> | Chebyshev | 2,03589e-4 | 42              | ?             |
| <b>InterpolacaoNewton</b>    | Chebyshev | 1,46565e-3 | 32              | $\infty$      |
| <b>SplineCubica</b>          | Chebyshev | 0,74467    | 3               | $\infty$      |
| <b>InterpolacaoChebyshev</b> | Chebyshev | 1,93886e-4 | 42              | ?             |

| Função g(x)                  | Nós       | Min E(n)    | N para min E(n) | Converge para |
|------------------------------|-----------|-------------|-----------------|---------------|
| <b>PolinomioInterpolador</b> | Regular   | 4,21441e-11 | 23              | 0             |
| <b>InterpolacaoNewton</b>    | Regular   | 1,00053e-12 | 24              | 0             |
| <b>SplineCubica</b>          | Regular   | 5,54415e-3  | 50              | 0             |
| <b>InterpolacaoChebyshev</b> | Regular   | 3,33999e-12 | 24              | 0             |
| <b>PolinomioInterpolador</b> | Chebyshev | 9,08995e-15 | 45              | 0             |
| <b>InterpolacaoNewton</b>    | Chebyshev | 1,04805e-13 | 29              | 0             |
| <b>SplineCubica</b>          | Chebyshev | 2,32641     | 2               | $\infty$      |
| <b>InterpolacaoChebyshev</b> | Chebyshev | 6,27276e-15 | 28              | 0             |

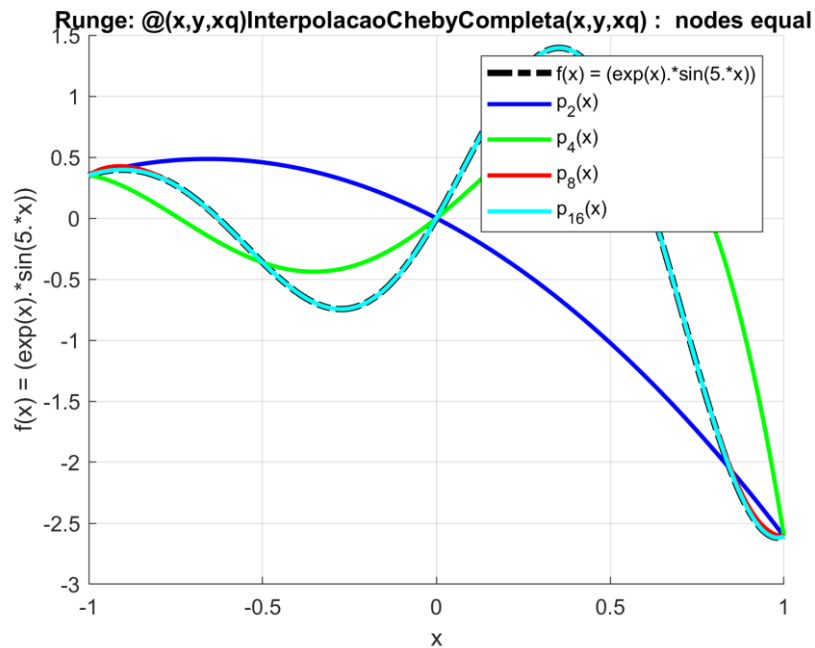
As informações e discussões referentes ao Lab4 serão adicionadas ao Apêndice.



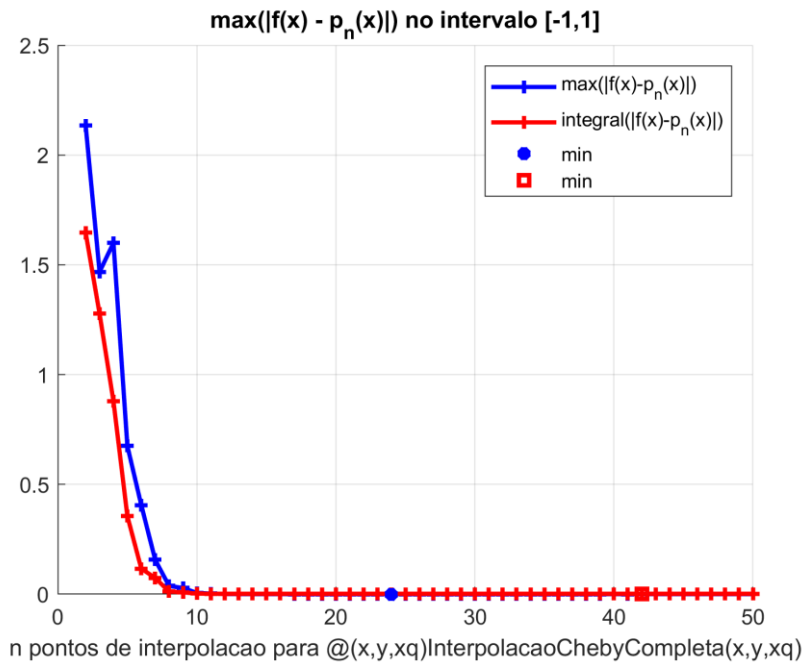
Aqui podemos notar que para a função de Runge com nós de interpolação igualmente espaçados existe o mesmo problema que todos os demais métodos polinomiais (exceto spline): o fenômeno de Runge.



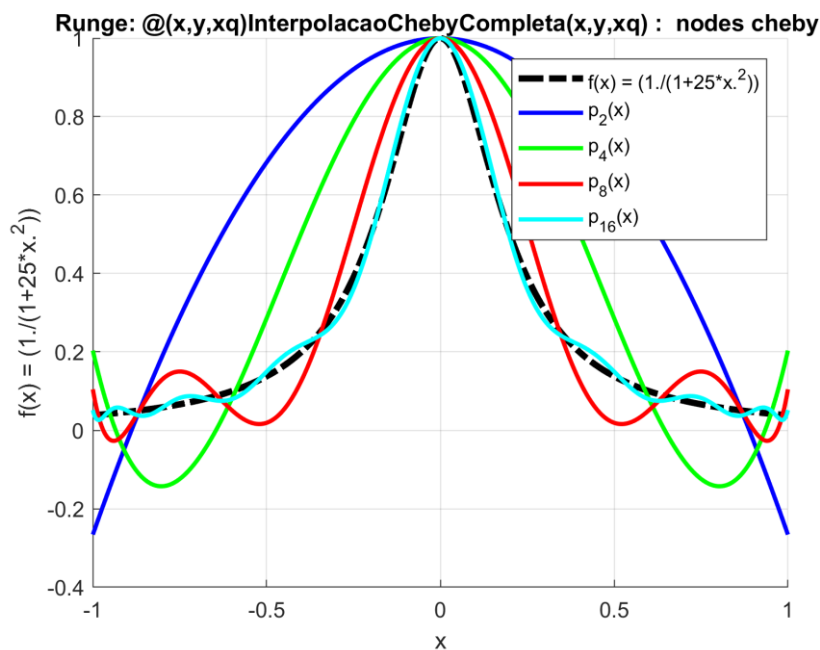
Avaliando o crescimento do erro, é bem evidente que a interpolação de Chebyshev não apresenta nenhuma atenuação visível do efeito de Runge, quando se utiliza nós linearmente espaçados.



Já para a função g(x), nenhuma surpresa, uma vez que essa função é bem comportada, o fenômeno de Runge não se pronuncia.

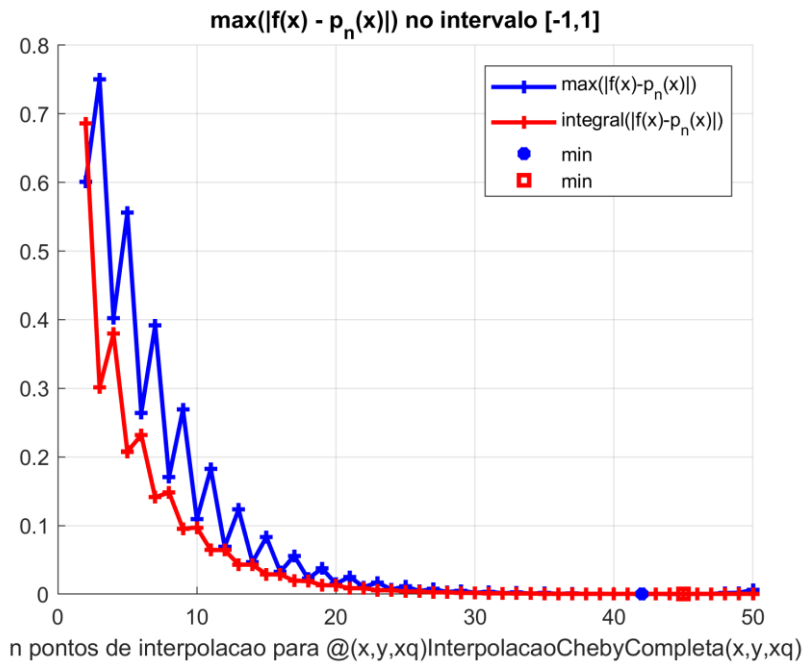


Podemos notar que o erro converge, não necessariamente para zero, mas para algum valor suficientemente próximo de zero para que seja considerado como tal.

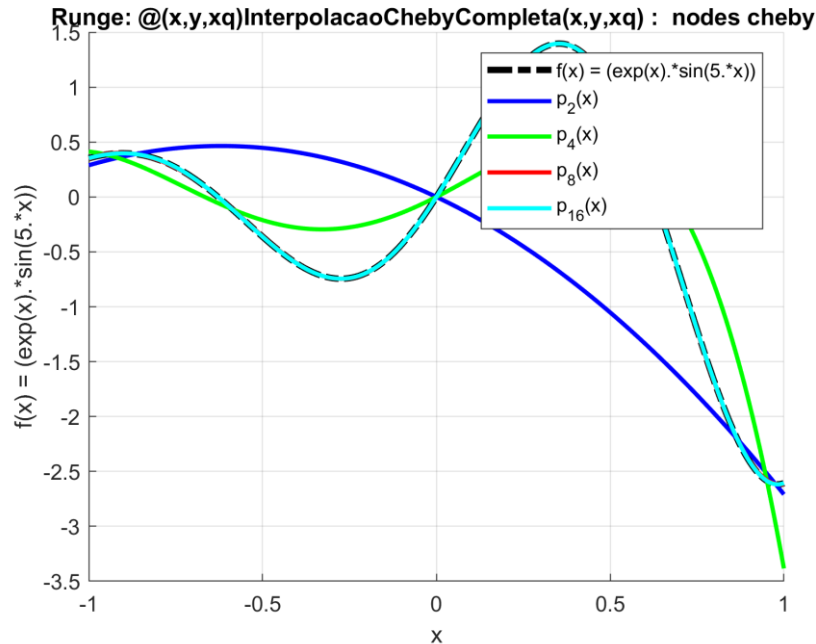


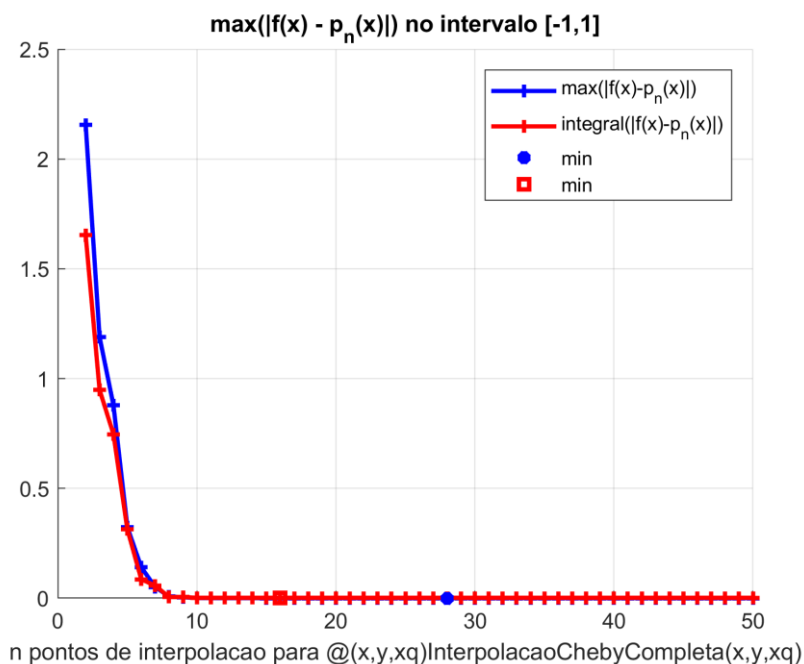
Unindo o método de Chebyshev ao uso dos nós de Chebyshev, observamos o mesmo que é observado para os métodos de Vandermonde e de Newton.





Nota-se que o erro não converge para zero, assim como não convergia no caso do método de Vandermonde. Inclusive o comportamento do erro é extremamente parecido. A única diferença que se poderia esperar observar é a redução do condicionamento da matriz para  $n$  grande, contudo, essa diferença não é observável.





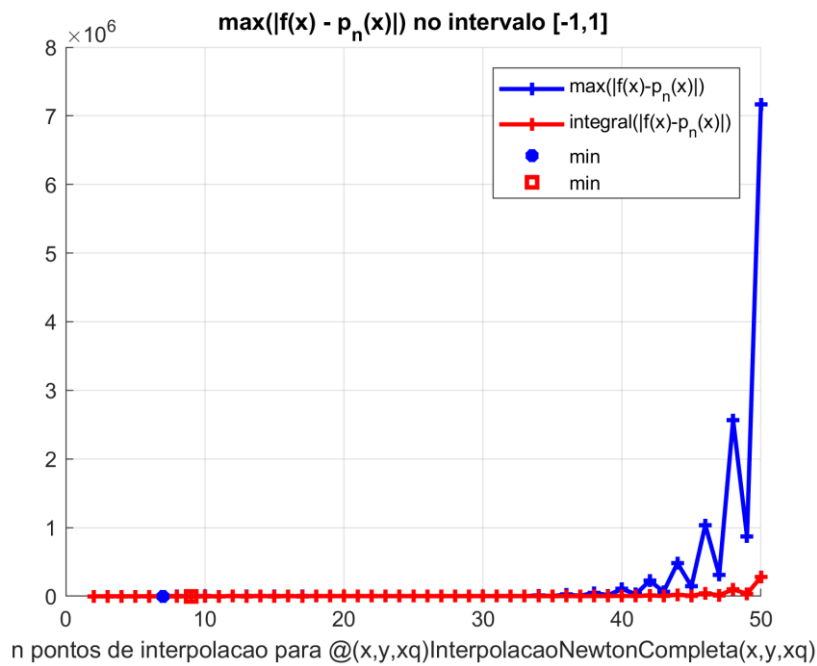
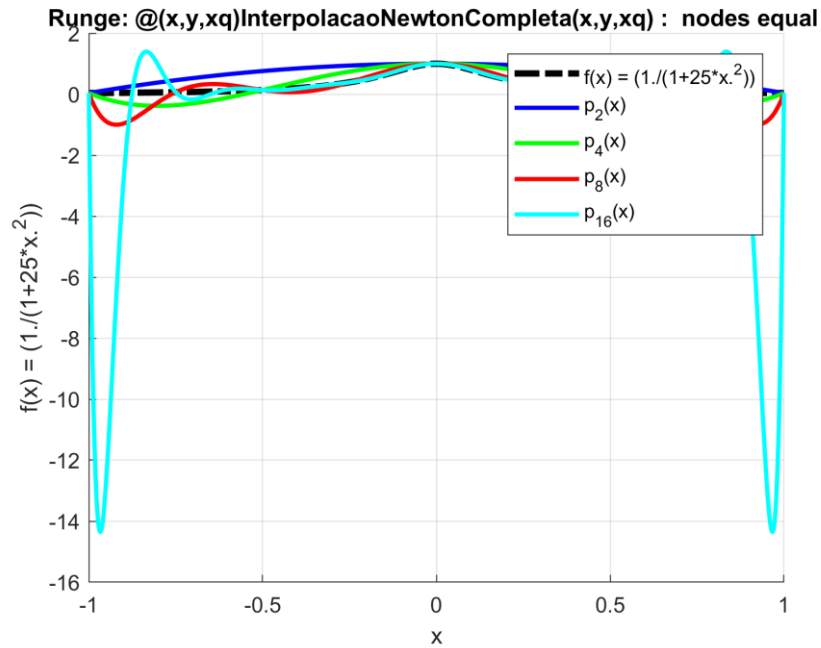
Para a função  $g(x)$  a utilização dos nós de Chebyshev apenas homogeneiza a distribuição do erro, sem nenhuma diferença sensível.

Em resumo, a interpolação de Chebyshev é praticamente equivalente à de Vandermonde, com a única diferença sendo que o condicionamento da matriz de coeficientes aumenta mais devagar, permitindo reduzir um pouco o erro para  $n$  grande em relação ao método de Vandermonde. No entanto, os gráficos para  $n$  até 50 não evidenciam essa redução.

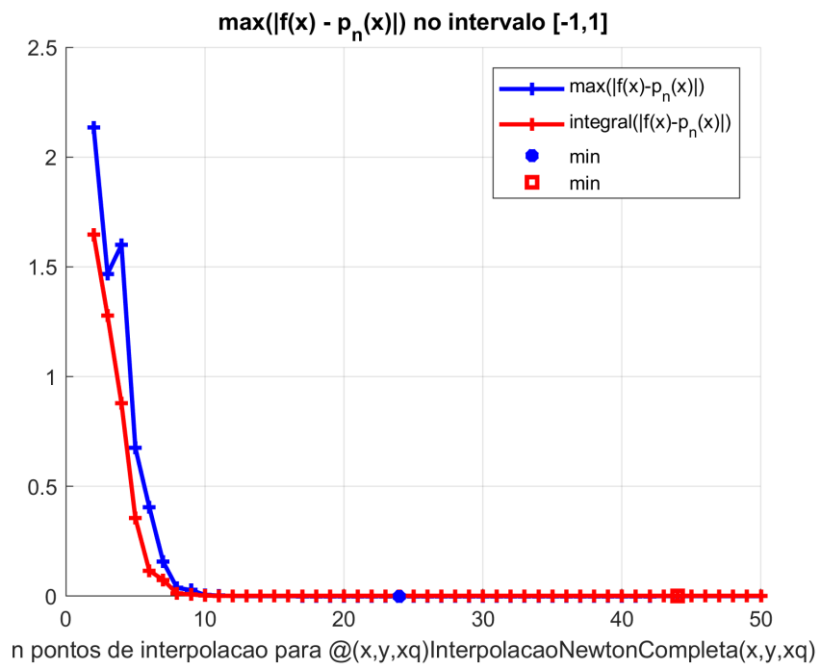
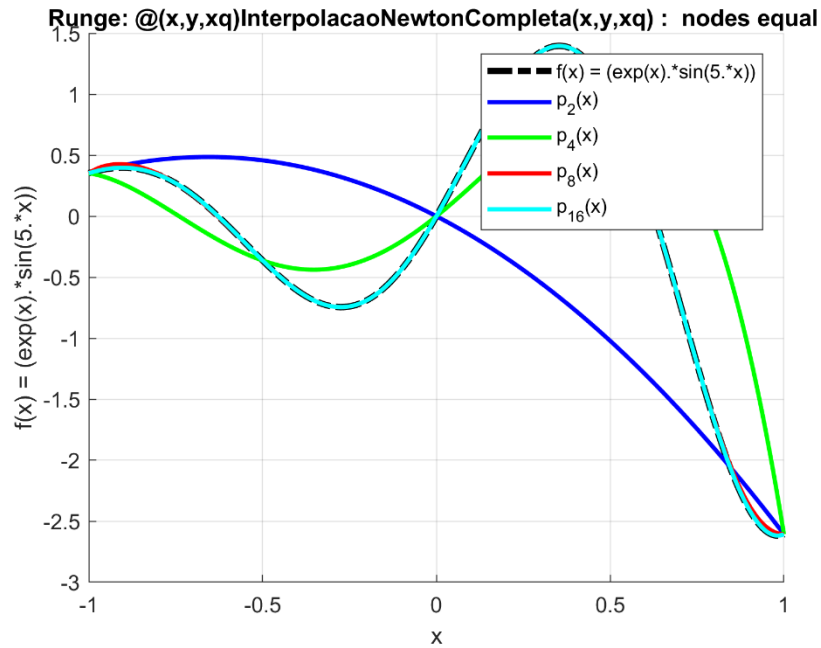
## 5. Apêndice

Aqui estão compilados os resultados e discussões do Lab4, em complemento às respostas do item 4.

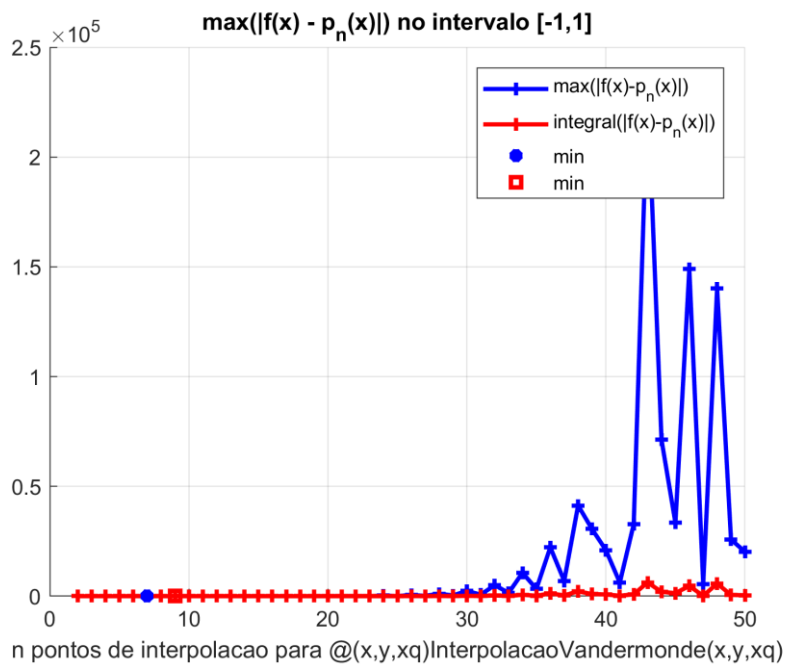
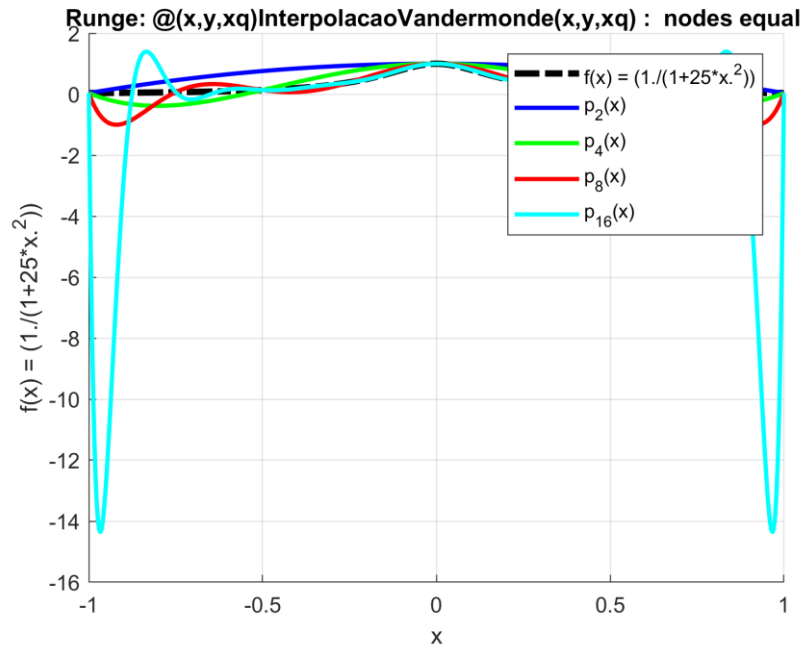
Utilizando os métodos de Newton, Spline Cúbica e Vandermonde para interpolação, com nós igualmente espaçados para as funções  $f(x)$  e  $g(x)$  descritas no roteiro do Lab4, foram gerados os seguintes gráficos:



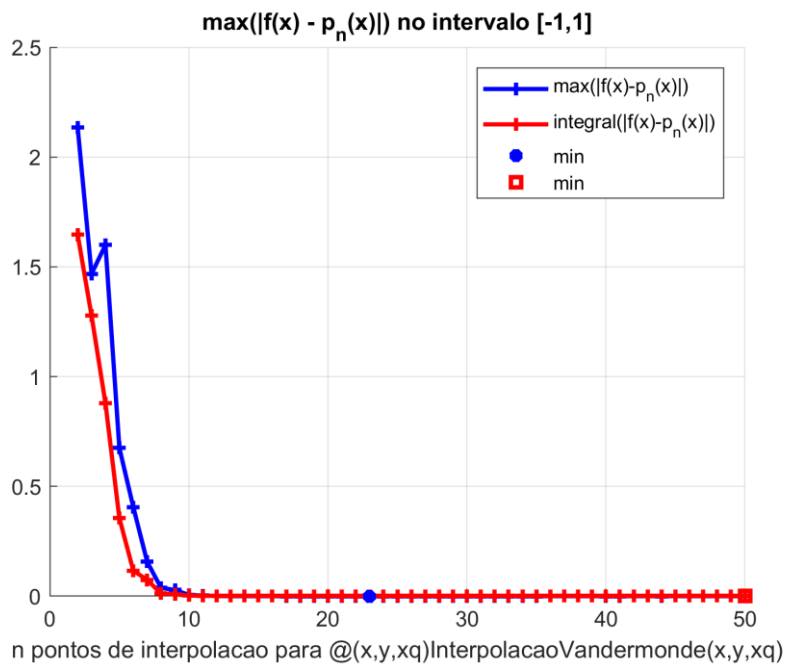
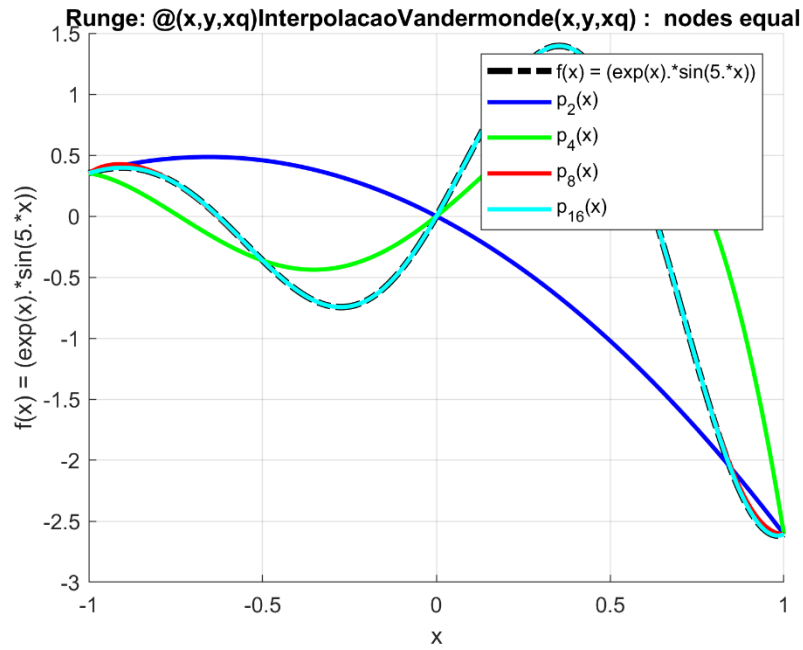
Aqui vemos que o aumento no número de nós melhora a interpolação apenas até certo ponto, e depois começa a divergir da função verdadeira.



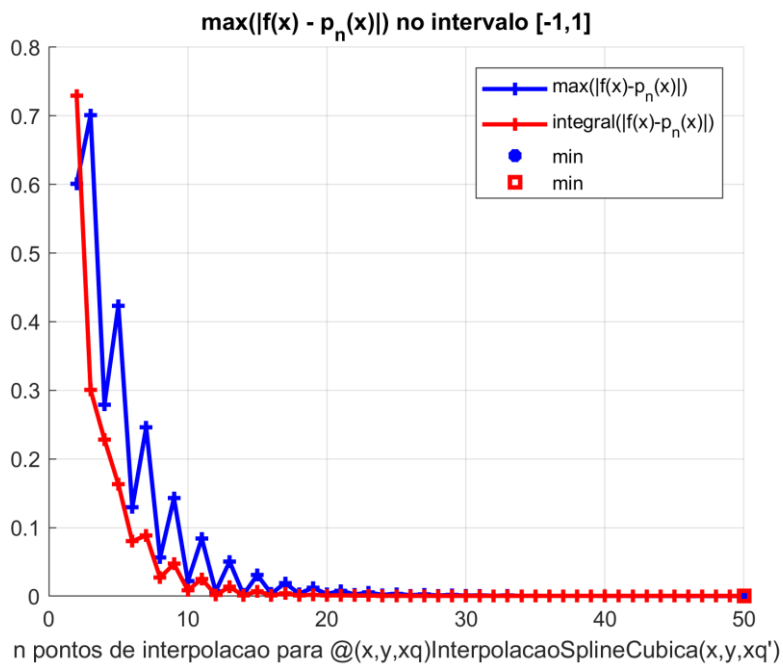
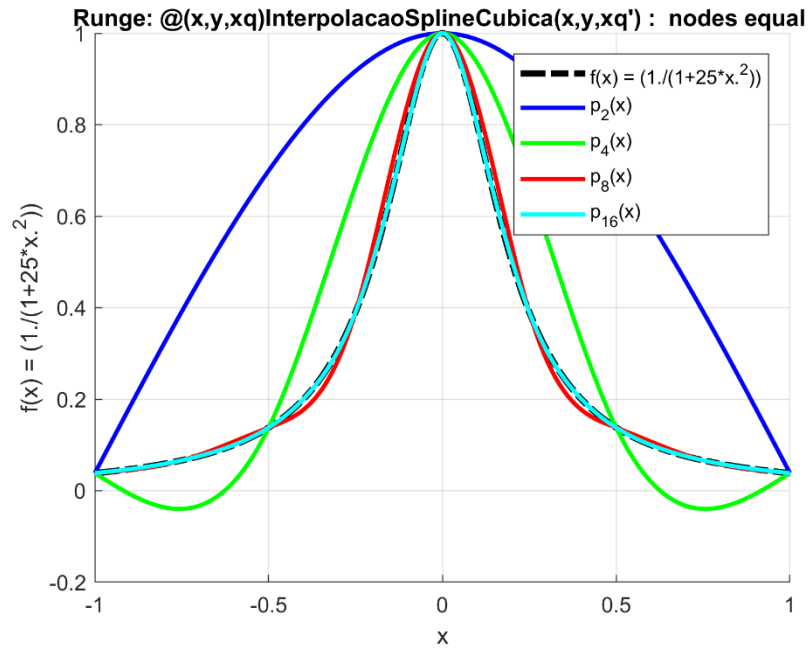
Já para a função  $g(x)$ , o erro parece não crescer descontroladamente com  $n$ , podendo tender a zero ou para alguma constante não nula. Isso mostra que a função  $g(x)$  é “bem comportada”, ou seja, não exibe o fenômeno de Runge.



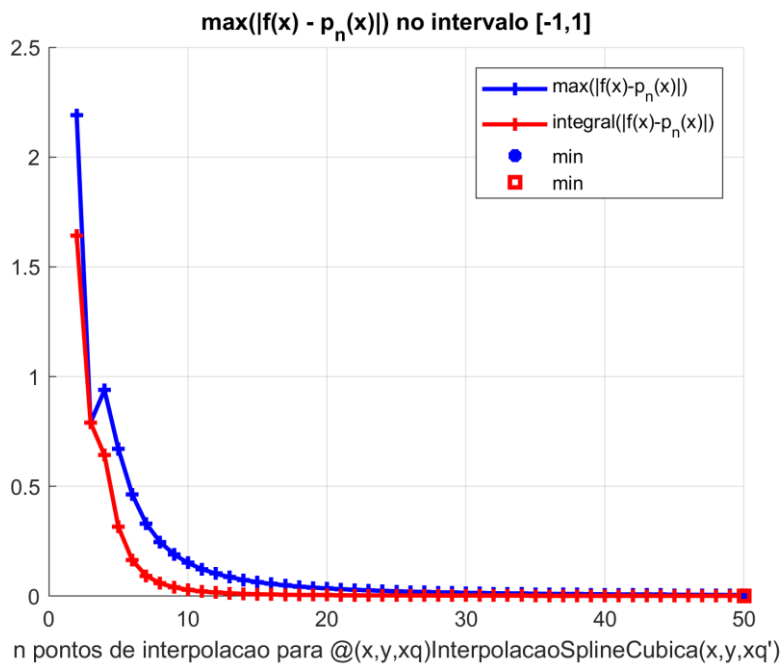
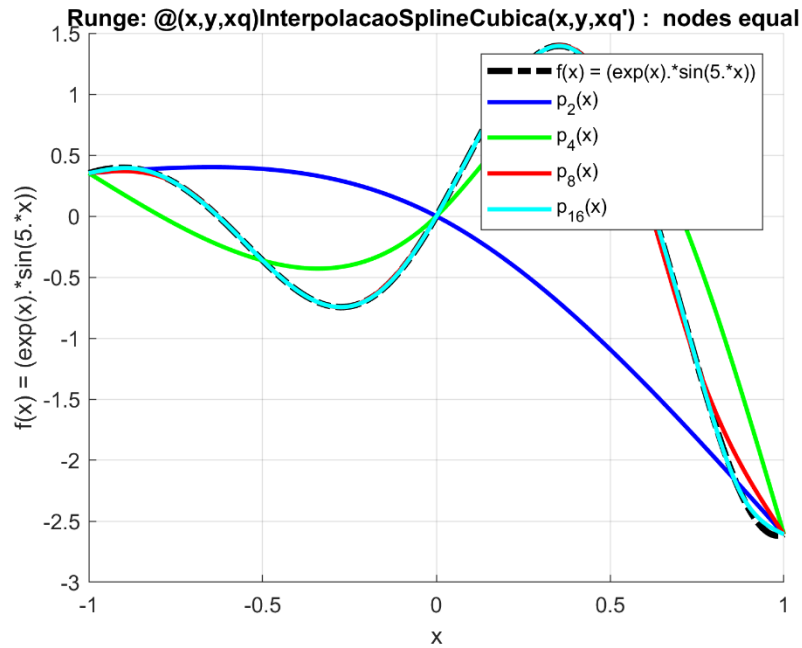
Novamente, podemos ver o efeito de Runge bem acentuado para o método de Vandermonde. O erro não diverge tão claramente quanto no método de Newton, mas evidentemente não tende a zero nem a alguma constante que possa ser identificada com o gráfico.



A função  $g(x)$  por sua vez não exibe o fenômeno de Runge, como esperado.



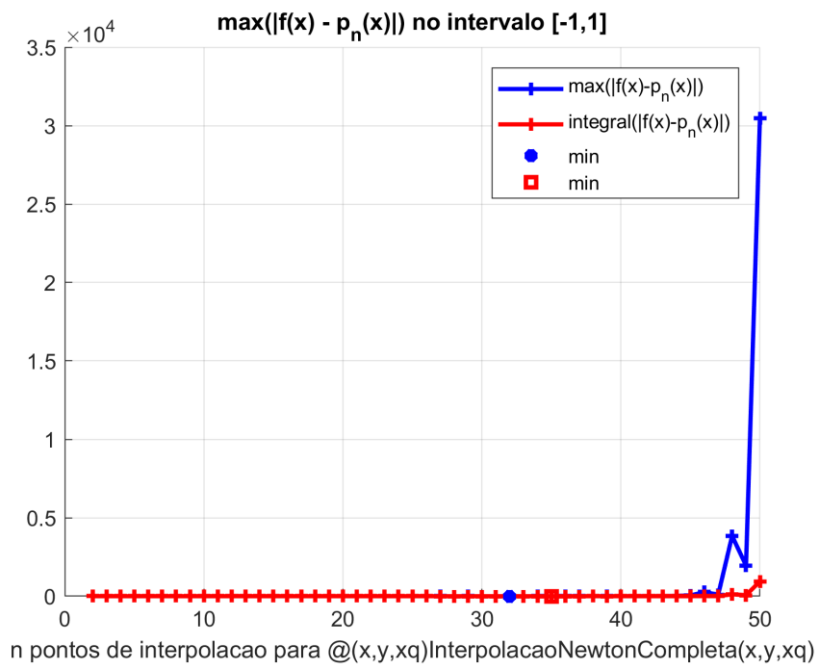
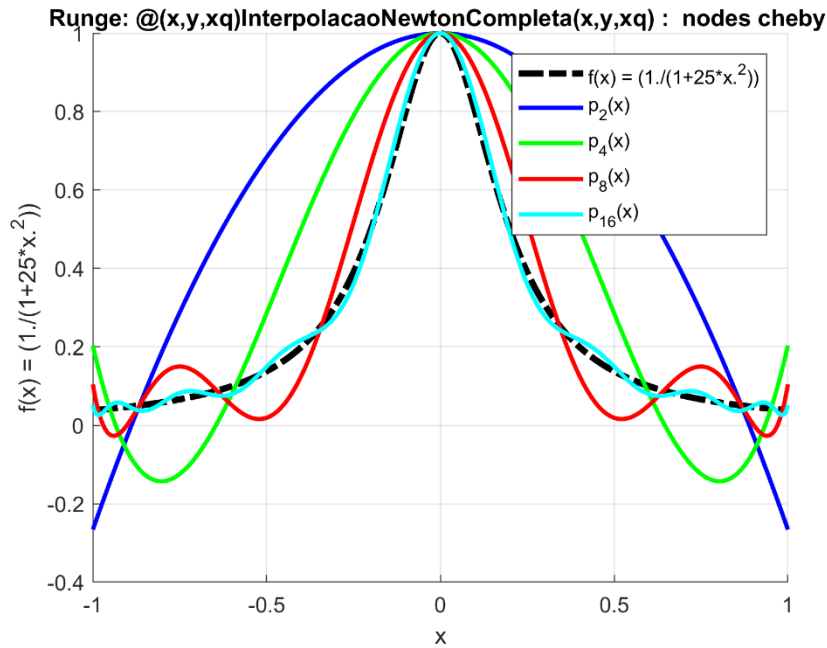
Com as splines, nota-se um ajuste muito fino à curva, com o erro diminuindo uniformemente conforme se aumenta o número de nós de interpolação. Convergência garantida, e neutralização do fenômeno de Runge!



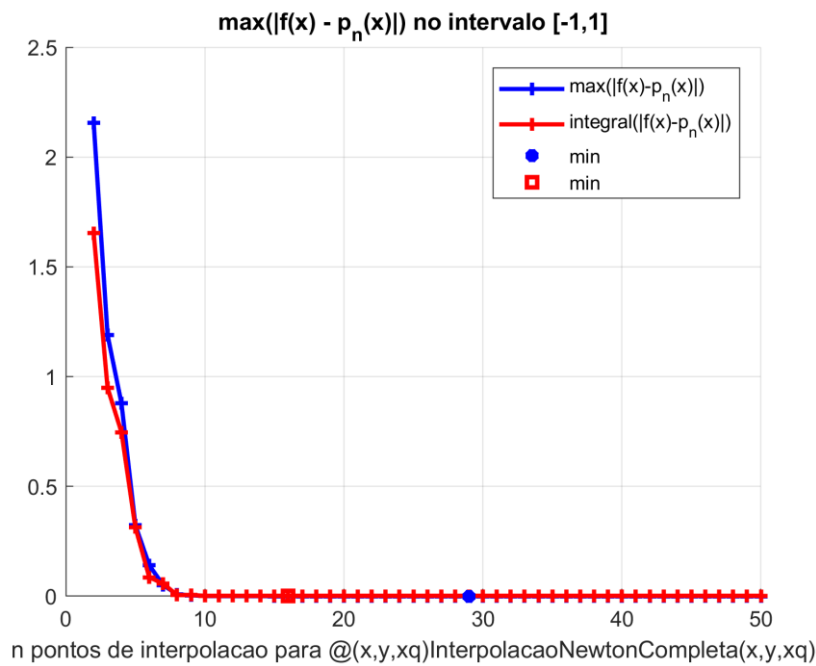
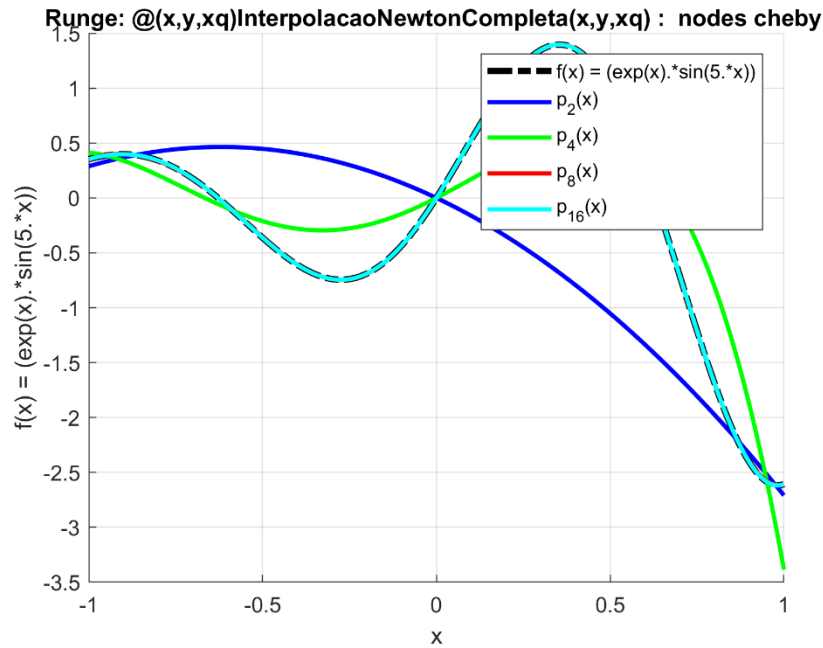
Para a função  $g(x)$ , como era de se esperar, as splines também convergem com eficácia crescente conforme  $n$  cresce. Enfim temos um caso onde aumentar o número de nós de fato implica numa melhor interpolação.

Nota-se que nem sempre aumentar o número de nós melhora a interpolação, como evidenciado para função de Runge para os métodos de Newton e Vandermonde.

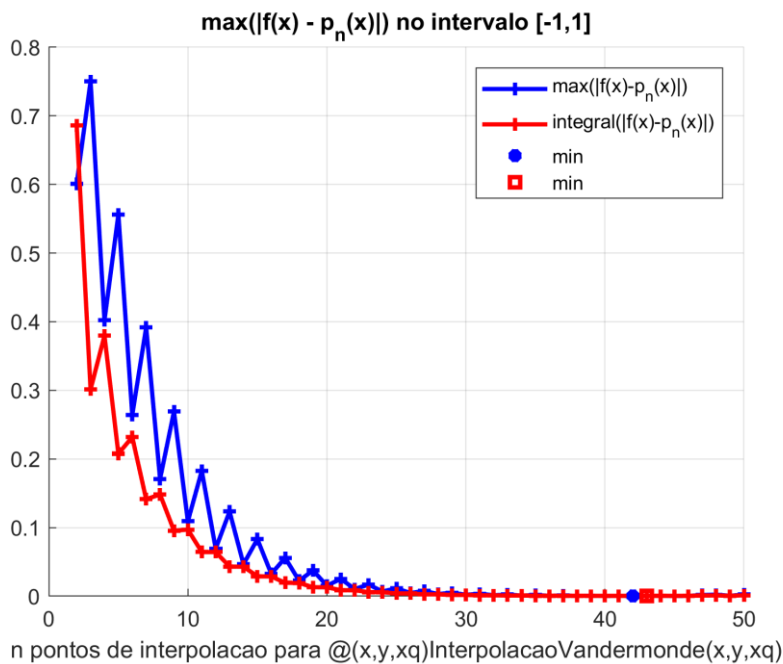
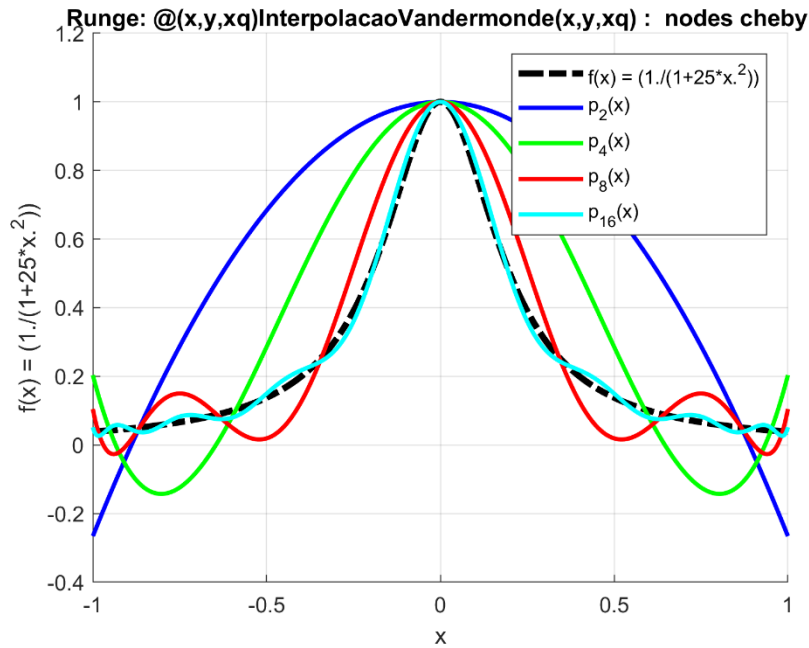




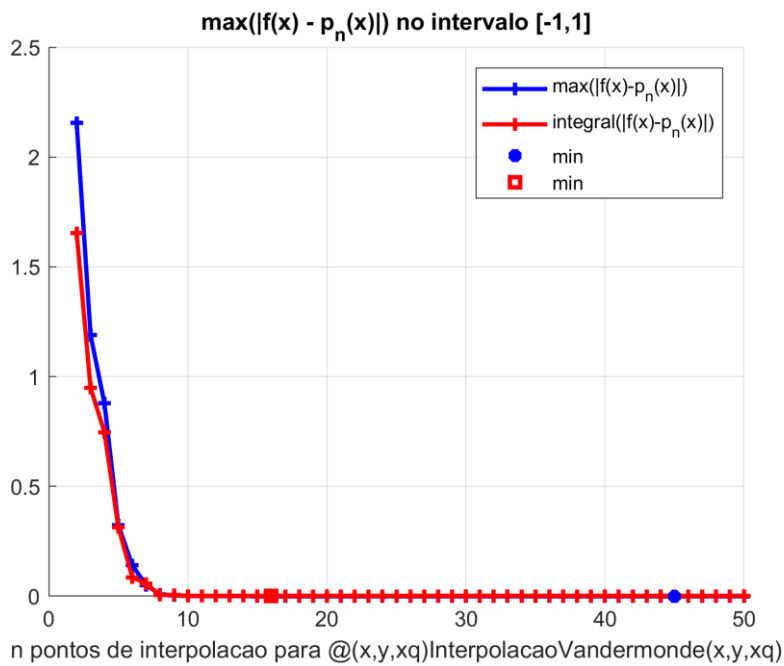
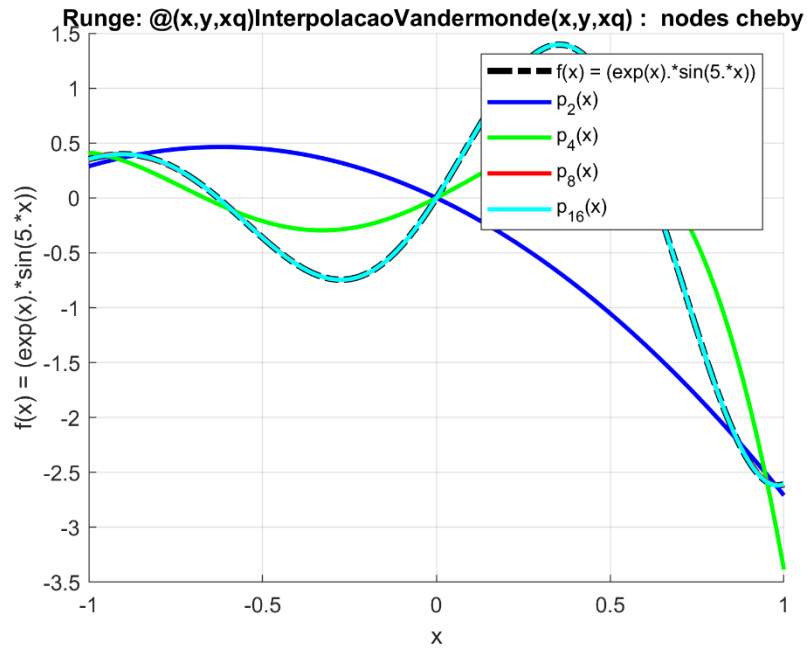
Nesse caso, vemos que o uso dos nós de Chebyshev permite, a princípio, uma melhor interpolação para pequenos valores de  $n$ , uma vez que o erro é distribuído de maneira mais homogênea. No entanto, isso não elimina o efeito Runge, que começa a se tornar evidente para  $n$  próximo de 50.



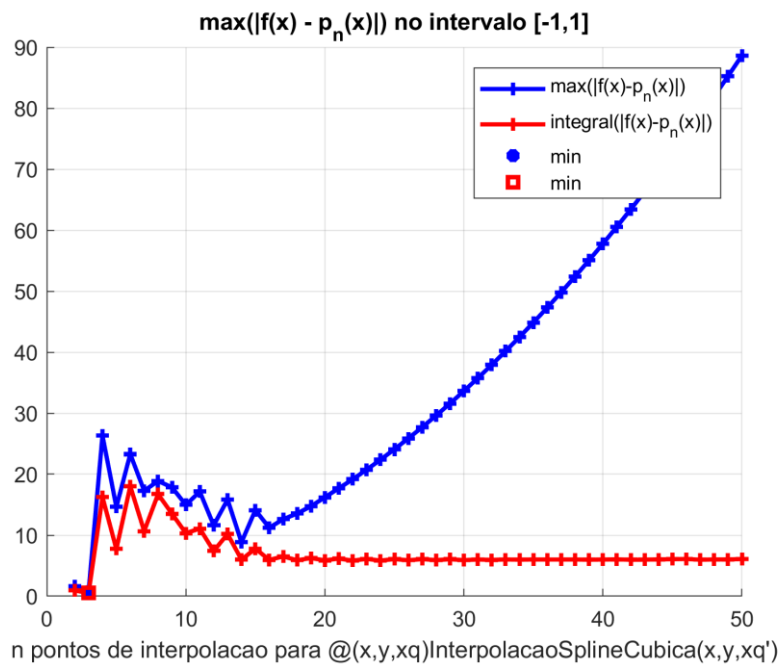
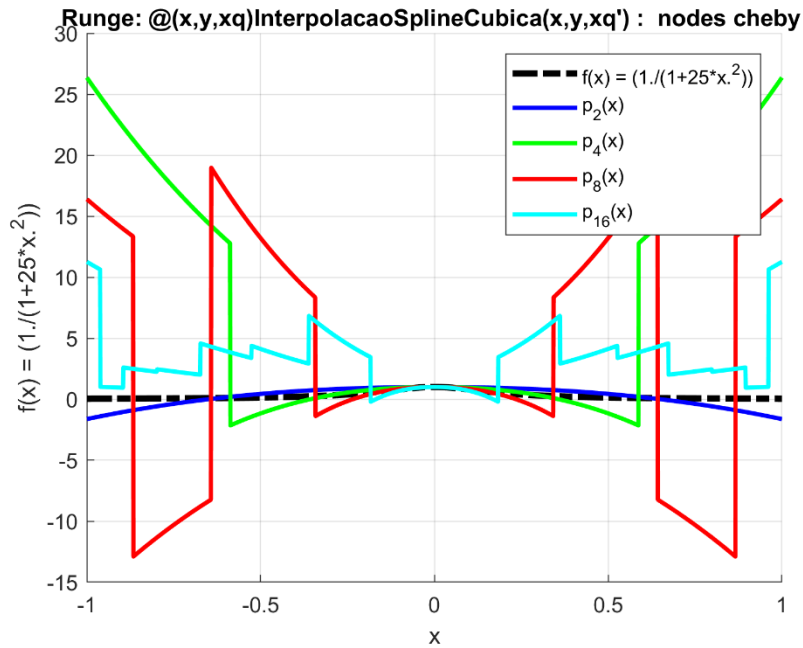
Já para a função  $g(x)$ , o efeito Runge não é evidenciado, e a interpolação funciona sutilmente melhor, conforme os dados tabelados ao final do relatório podem comprovar.



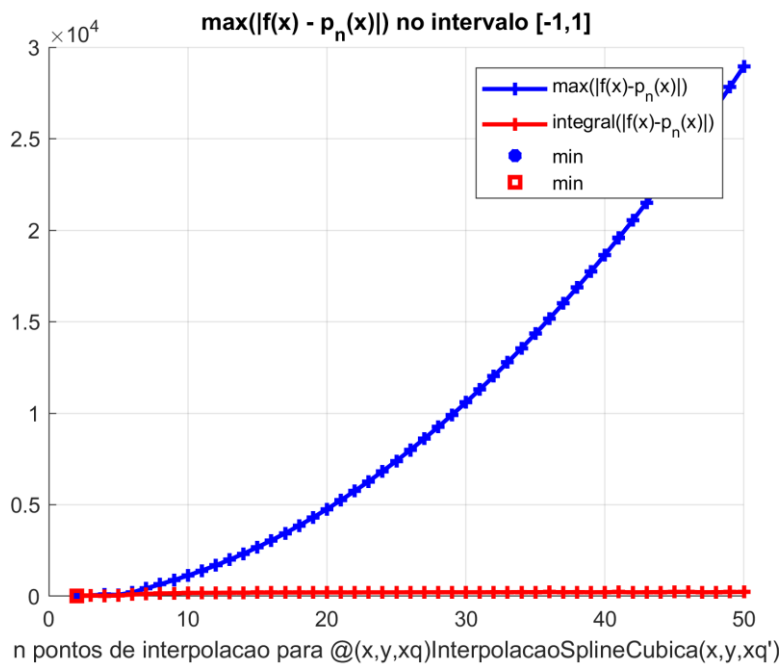
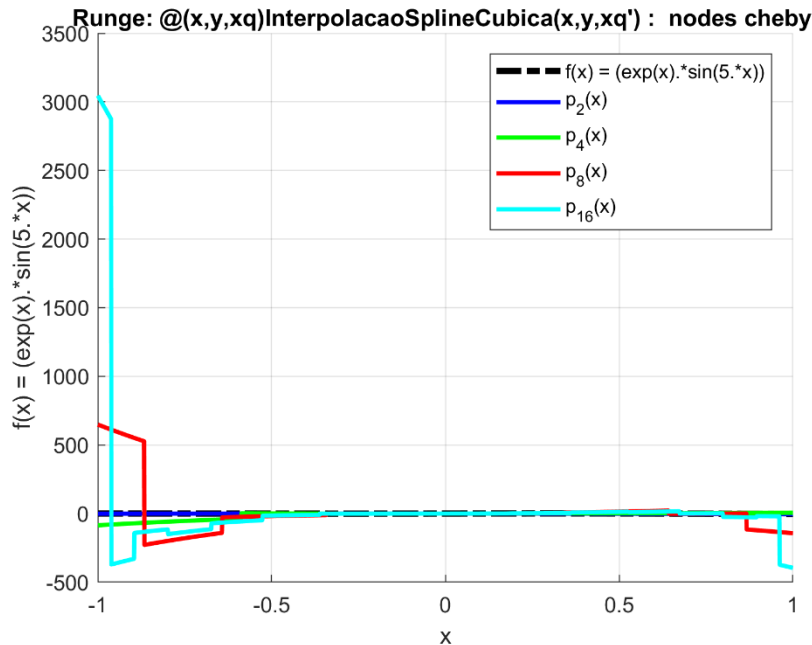
Aqui vale destacar o efeito da utilização dos nós de Chebyshev. O erro é acentuado para  $n$  pequeno, e decresce conforme  $n$  aumenta, porém não se pode dizer que converge para zero, ou que sequer converge. Contudo, é claro que o erro está distribuído muito mais homogeneamente do que no caso anterior.



Com a função  $g(x)$  a interpolação é quase sempre bem razoável, com o erro se mantendo estável e baixo para  $n$  na faixa observada. Pode-se dizer que, embora não se possa garantir convergência para o erro zero, o erro converge para alguma constante de valor bem próximo de zero.



Quando utilizamos nós de Chebyshev para splines, catástrofe. A junção de duas estratégias aparentemente tão eficazes resulta numa falha completa. O erro é crescente com  $n$  e a interpolação não é satisfatória em nenhum caso. Não se trata do fenômeno de Runge agindo, uma vez que não é um pequeno erro persistente mas uma discrepância descontrolada.



Analisando o resultado da união entre splines e nós de Chebyshev para a função  $g(x)$ , temos a certeza de sua incompatibilidade. Uma função que teoricamente não deveria apresentar o fenômeno de Runge, revela um erro de interpolação que cresce estrondosamente com  $n$ .

Utilizando agora nós de Chebyshev em vez de linearmente espaçados, os métodos de interpolação foram novamente testados para as duas funções. Pode-se observar que para todos os métodos, com exceção do Spline, o efeito Runge foi atenuado (para o caso da função de Runge) e o erro diminuiu significativamente (para o caso da função  $g(x)$ ).

Para os métodos de Vandermonde e Newton, percebe-se que o uso dos nós de Chebyshev fazem com que o erro se distribua mais homogeneamente sobre a função, aproximando de maneira visivelmente melhor a função.

Por outro lado, a spline tem sua eficácia completamente destruída pelo uso dos nós de Chebyshev. O que antes convergia muito bem mesmo para a função de Runge, agora falha até mesmo para a função  $g(x)$ . Isso ocorre pois as splines supõem um espaçamento uniforme entre os nós, o que não existe quando se utilizam os nós de Chebyshev.

Comparando a interpolação por polinômio único (Vandermonde e Newton) com as splines, pode-se destacar algumas vantagens das splines. As splines possuem convergência garantida mesmo para a função de Runge, e são menos custosas de serem calculadas ( $O(n)$ ). No entanto, não podem ser utilizadas com nós de Chebyshev. As interpolações por polinômio único em geral estão sujeitas ao fenômeno de Runge, do qual não podem escapar totalmente mesmo com a utilização dos nós de Chebyshev.

Quanto à utilização dos nós de Chebyshev, uma possível desvantagem é que ela não é compatível com o uso de splines. Outra desvantagem pode surgir caso se tenha uma função com muita informação concentrada na região central do intervalo, isto é, uma função que varie muito no meio do intervalo. Isso é problemático pois os nós de Chebyshev são mais “rarefeitos” na região central do intervalo, portanto, capturam menos informação nessa região.