**Aproximação por Splines e Polinómios Interpoladores**

FCUP

Análise Numérica (M2018) 2018/2019

Trabalho de Grupo 3

Ângelo Gomes – 201703990 – MIERSI

Eduardo Morgado – 201706894 – MIERSI

Simão Cardoso – 201604595 – MIERSI

Sónia Rocha – 201704679 – MIERSI

1. **Considerações Iniciais:**

Na realização deste trabalho, utilizamos como linguagem de implementação Python(3.7.2), tendo dividido o trabalho em 4 grandes classes, *Main.py*, *Spline.py* (classe de cálculo de spline), *NewtonDiferences.py* (classe de cálculo de diferenças divididas e aplicação do método de Newton para criação do polinómio interpolador) e *Interpolation.py* (classe de cálculo do polinómio interpolador).

Todos estes programas podem ser encontrados no Github no repositório da equipa, <https://github.com/thejoblessducks/Trabalho3AN>, iremos mostrar as classes em geral bem como as funções dos exercícios 1 e 2, no entanto, recomendamos que consulte o repositório, uma vez que, será mais fácil analisar o código e torna possível o seu teste.

Para algumas das classes utilizamos bibliotecas específicas do Python para facilitar a resolução do problema, essas bibliotecas foram, *matplotlib*, *numpy* e *scipy*, caso queira testar o problema, estas bibliotecas devem estar instaladas, a instalação poderá ser feita em linha de comandos UNIX/Windows/Mac através dos comandos: *pip install matplotlib*, *pip install numpy* e *pip install scipy* respetivamente.

As *Figuras* *1*, *2¸ 3* e *4-6* apresentam os códigos de *Spline.py*, *NewtonDiferences.py*, *Interpolation.py*  e *Main.py* respetivamente.

É também importante referir que no Python os dados são tratados em formato *double*, ou seja, todos os resultados, das tabelas, do cálculo dos splines e dos erros estará em formato *double* sendo assim, por uma questão de facilidade de visualização, iremos apresentar os resultados, com 4 casas decimas (apesar de poder existir um certo erro quando comparado com os resultados do programa em si), isto se a tabela de dados permitir. Todos os resultados, desde a tabela de pontos, os gráficos, erros e os sistemas de equações de spline são fornecidos pelo programa.

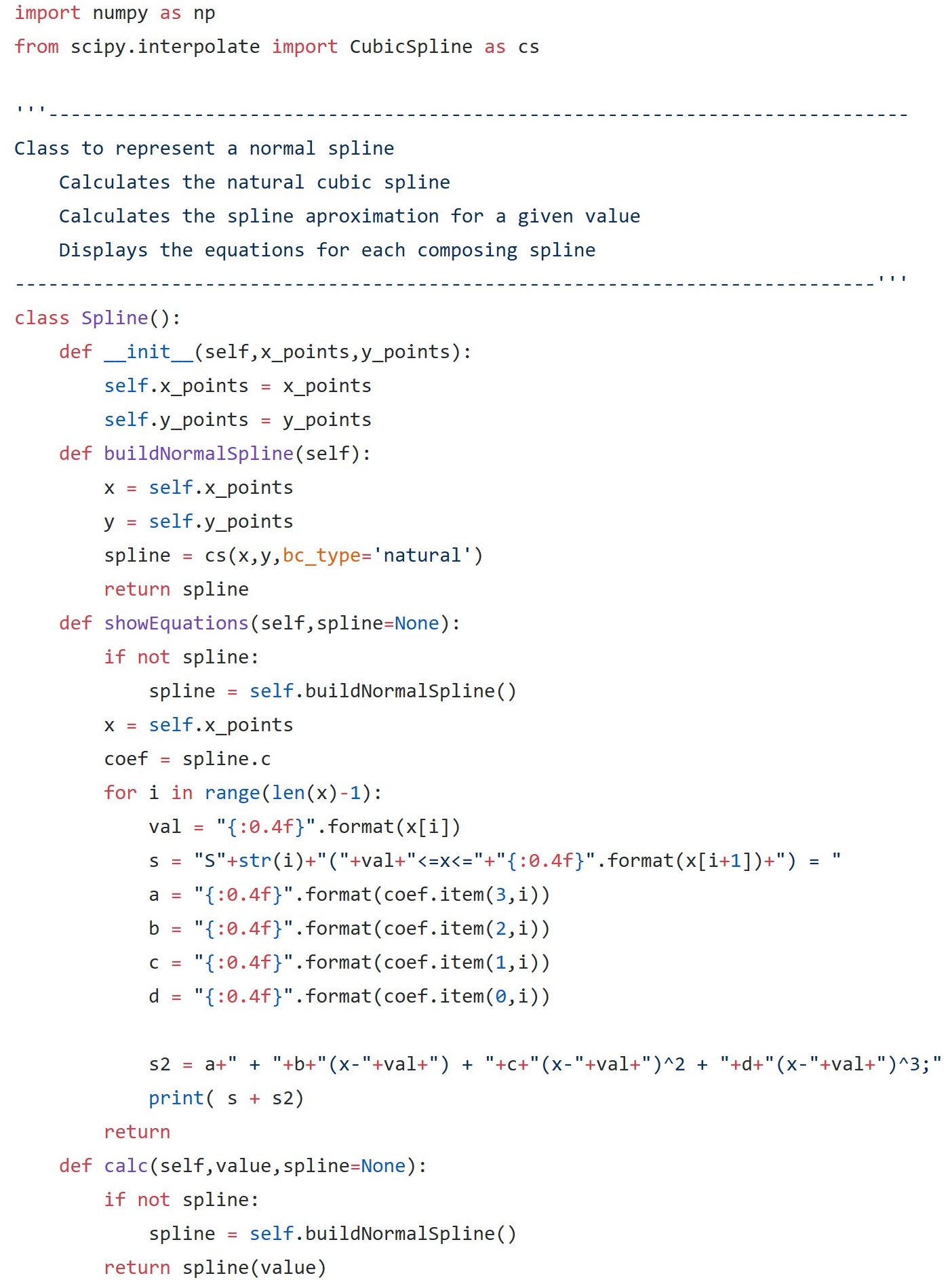


Figura 1-Spline.py

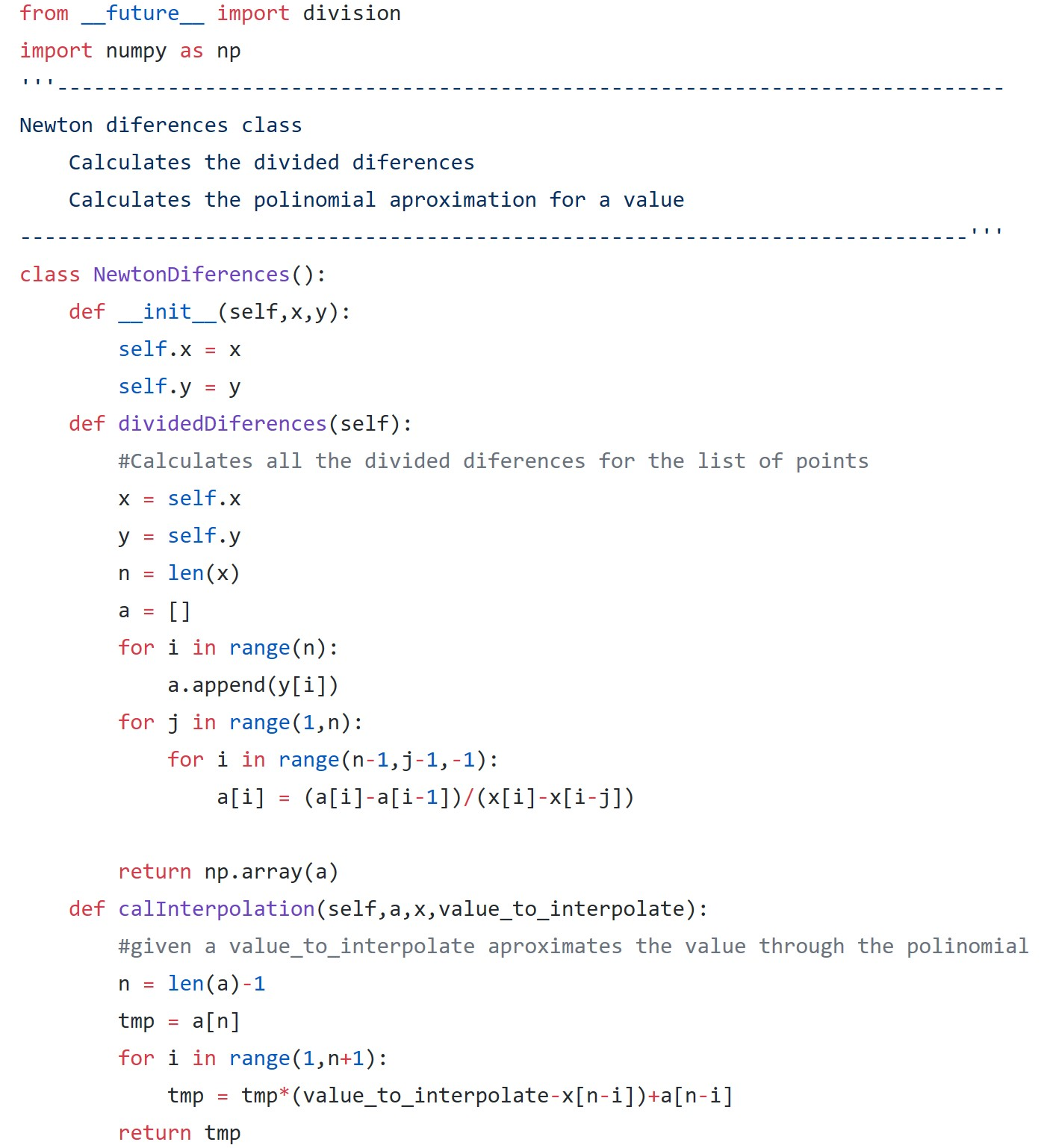


Figura 2- NewtonDiferences.py

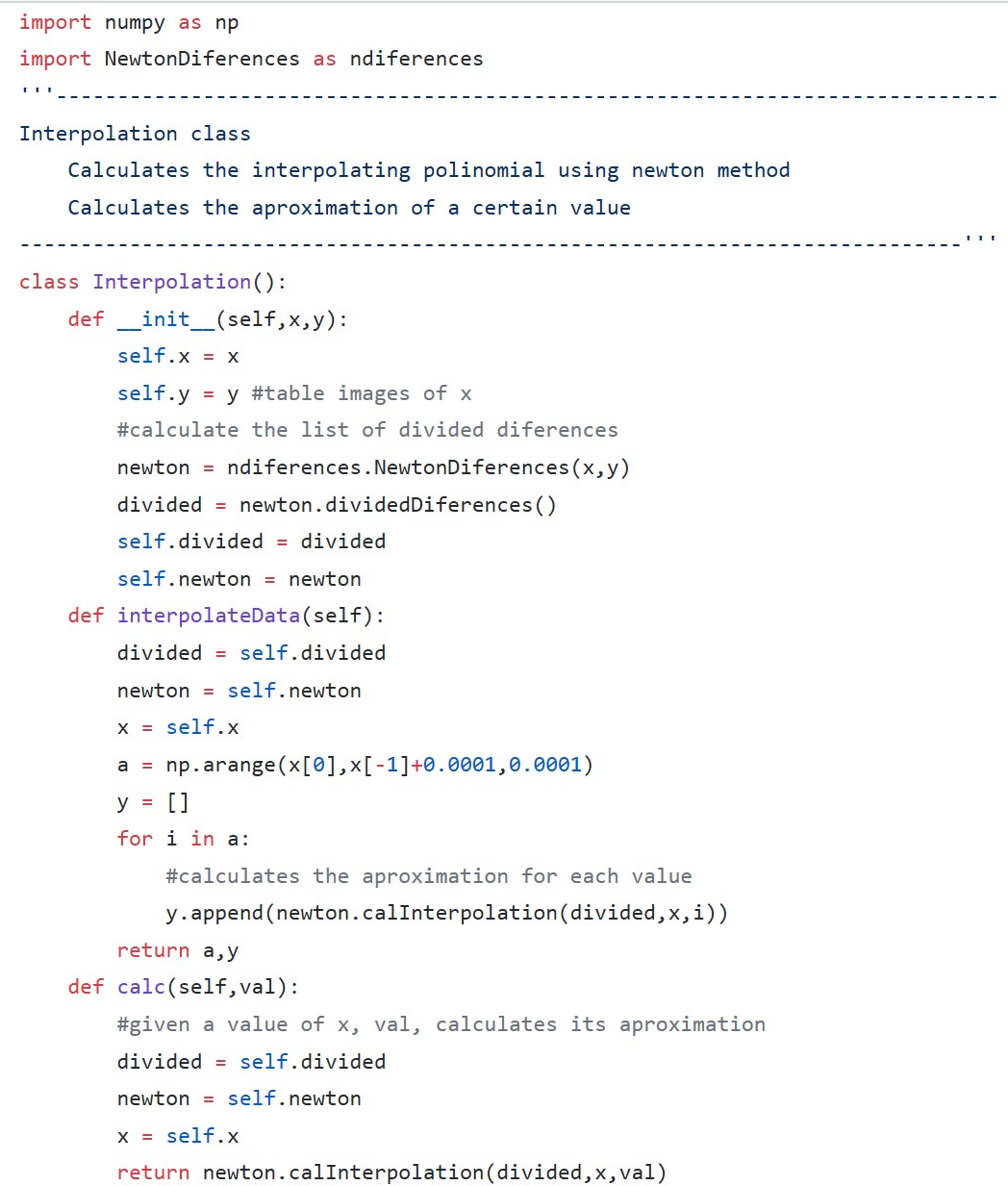


Figura 3- Interpolation.py



Figura 4- Main.py



Figura 5- Main.py



Figura 6- Main.py

1. **Exercício 2:**
   1. **Alínea a:**

Para esta secção iremos aplicar as classes representadas anteriormente, no ficheiro *Main.py* iremos chamar a função *ex1()*, apresentada na *Figura 5*, para esta função, a tabela considerada está representada na *Figura 7*, a *Figura* 8 apresenta o sistema de equações do spline cúbico natural gerado.

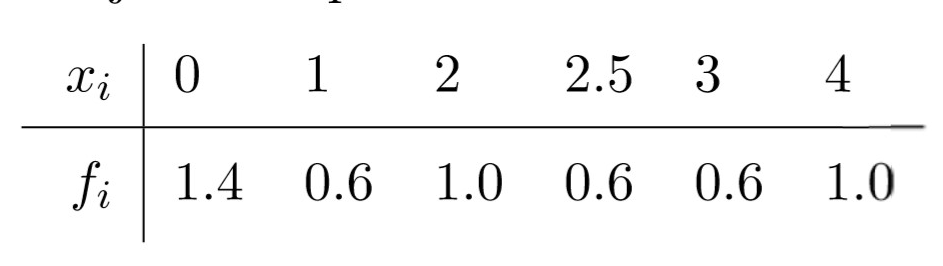


Figura 7- Tabela de valores para exercício 2.a

Figura 8- Sistema de equações para exercício 2.a)

A *Figura 9* apresenta os gráficos do spline cúbico natural e do polinómio interpolador para a tabela (azul estão os pontos da tabela, x, a linha a verde representa o polinómio interpolador e a linha a laranja, o spline).

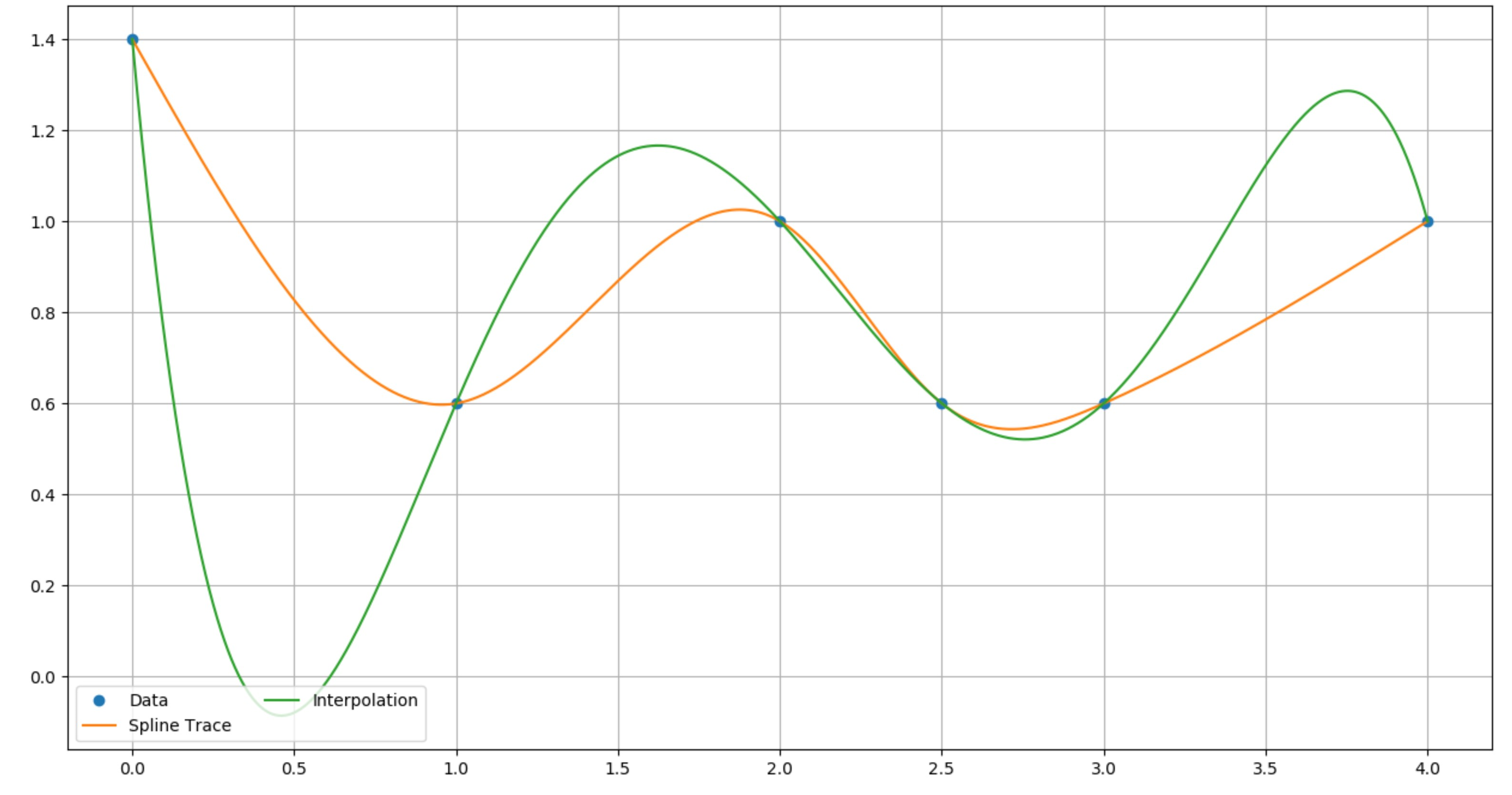


Figura 9- Gráficos de spline e polinómio interpolador para exercício 2.a)

Para este exercício não existe uma forma de comparar os gráficos quanto ao quão próximo estes aproximam a função , no entanto, a partir da observação dos gráficos, podemos concluir que o spline apresenta um gráfico mais suave para a representação de , observamos também que existe uma grande discrepância de valores entre os gráficos do spline e do polinómio, são poucos os pontos onde os gráficos se intersectam (quase todos os pontos, são pontos da tabela). Sem a função original torna-se difícil escolher qual dos métodos aproxima melhor a função, no entanto, a interpolação polinomial parece aproximar os dados de forma mais drástica e o spline uma aproximação mais estável.

* 1. **Alínea b:**

Para esta secção iremos aplicar as classes representadas anteriormente, no ficheiro *Main.py* iremos chamar a função *ex2()*, apresentada na *Figura 5*, para esta função, a tabela considerada está representada na *Figura 10* (tabela de pontos de abcissas igualmente espaçadas no intervalo [-1,1]), a *Figura* 11 apresenta o sistema de equações do spline cúbico natural gerado. É importante voltar a referir que os valores apresentados nas tabelas não são os valores fornecidos pelo programa, no programa os valores apresentam 16 casas decimais, no entanto, para uma facilidade de leitura, iremos apresentar todos os valores com 4 casas decimais apenas.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | -1 | -0.75 | -0.5 | -0.25 | 0 | 0.25 | 0.5 | 0.75 | 1 |
|  | 3.5879 | 1.8000 | 1.9775 | -0.5281 | 0 | 1.0281 | 0.0225 | 2.700 | 4.4121 |

Figura -Tabela de valores para exercício 2.b) i

Figura -Sistema de Equações de Spline Cúbico Natural para tabela da Figura 10

A *Figura 12* apresenta os gráficos do spline cúbico natural do polinómio interpolador para a tabela e da função (azul estão os pontos da tabela, x, a linha a laranja representa a função , a vermelho está representado o polinómio interpolador e a linha a verde, o spline cúbico natural).

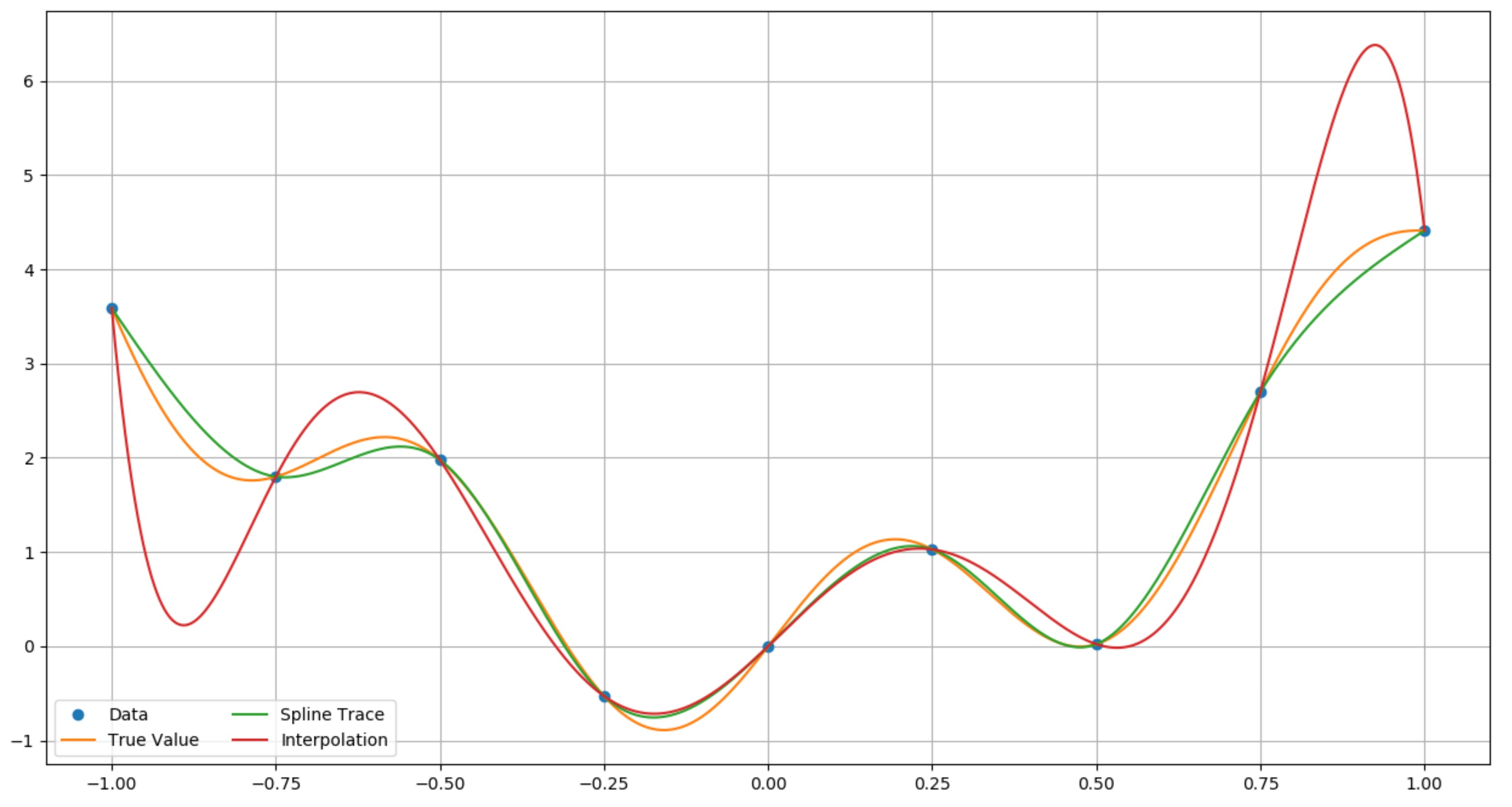


Figura -Gráfico para o exercício 2.b) ii intervalo [-1,1]

Numa primeira observação podemos concluir que o spline segue mais proximamente o traço de , já o polinómio interpolador apresenta maiores discrepâncias, especialmente para valores próximos dos extremos do intervalo [-1,1], esta diferença é normal, uma vez que, o polinómio interpolador é um polinómio que aproxima toda a função num intervalo, enquanto que o spline é composto por vários polinómios interpoladores num intervalo, ou seja, o spline vai adaptar-se melhor aos dados, daí seguir melhor o traço da função que está a ser aproximada.

No entanto, tal como a *Figura 13* mostra, existem várias instâncias onde o polinómio interpolador aproxima melhor os dados que o spline

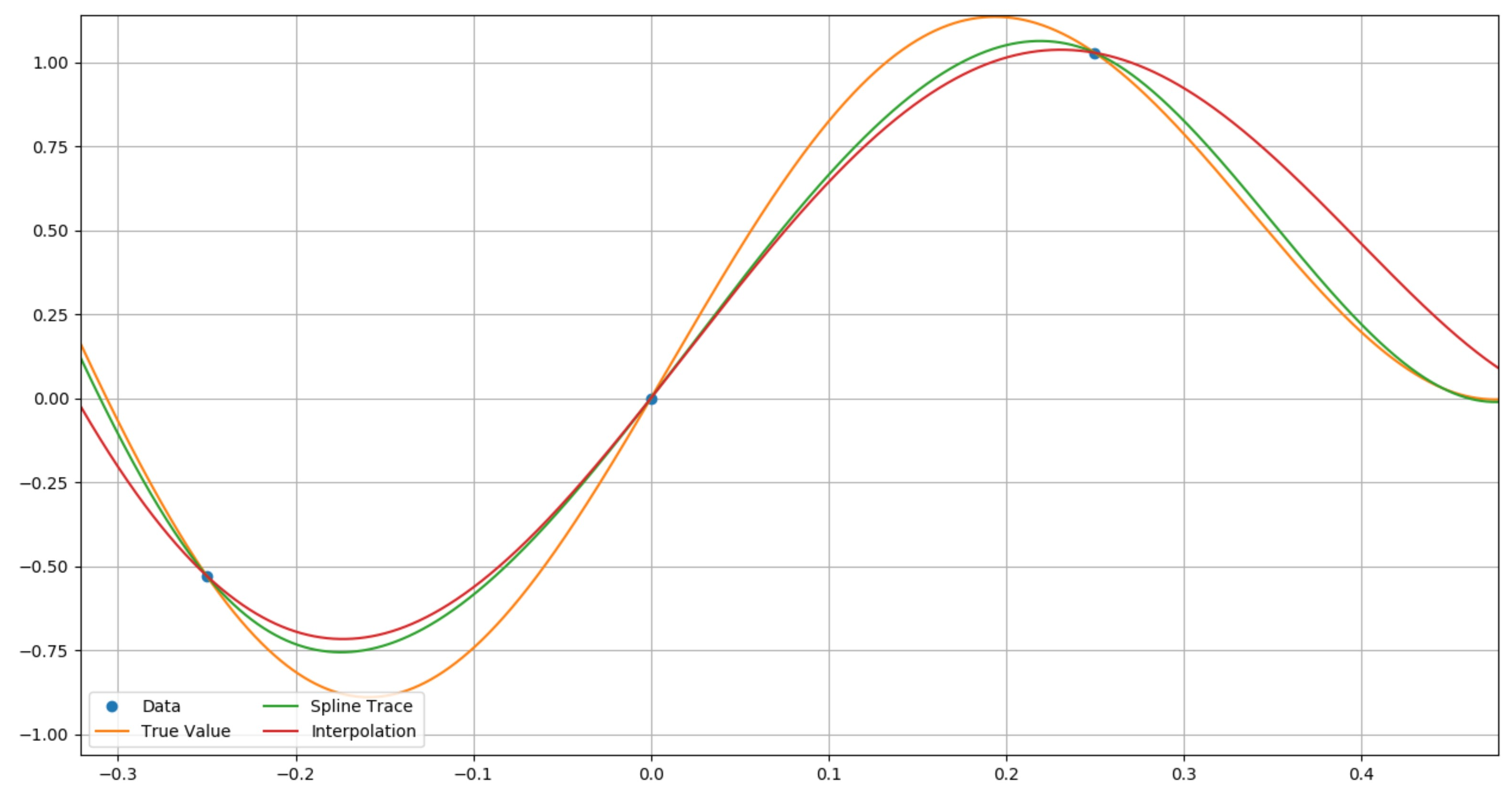


Figura - Gráfico para exercício 2.b) ii intervalo [-0.3,0.4]