

Polar circularity getting coordinates and primes related :

By Luis Felipe Massena Misiec

n = 2000;

(* Gerar a lista dos primeiros n números primos *)

primos = Prime[Range[n]];

(* Calcular a diferença entre cada número primo e sua posição *)

diferencas = primos - Range[n];

(* Calcular as diferenças entre os elementos consecutivos *)

a = Differences[diferencas];

(* Calcular a média das diferenças *)

mediaDiferencas = Mean[diferencas];

(* Exibir o resultado com precisão de 9 dígitos *)

N[mediaDiferencas, 9];

(* Calcular a média das diferenças consecutivas *)

mediaDifConsec = Mean[a];

N[mediaDifConsec, 9];

(* Somar as diferenças consecutivas com a média das diferenças *)

c = a + mediaDiferencas;

(* Exibir o resultado com precisão de 9 dígitos *)

cc = N[c, 9];

(* Plotar os dados em um gráfico polar *)

```
ListPolarPlot[c];
```

```
(* Selecionar os primeiros 100 dados *)
```

```
data = Take[c, 100];
```

```
(* Calcular as coordenadas x e y a partir do raio e do ângulo *)
```

```
polarPoints = Table[  
  {data[[i]] Cos[i], data[[i]] Sin[i]}, {i, 1, Length[data]}  
]
```

```
(* Exibir as coordenadas no gráfico *)
```

```
Graphics[  
  Table[  
    {Text[i, polarPoints[[i]]]}, {i, 1, Length[data]}  
  ],  
  Axes -> True  
]
```

```
N[polarPoints,9]
```

```
(* Selecionar os primeiros 100 dados *)
```

```
data = Take[c, 200];
```

```
(* Calcular as coordenadas polares com precisão numérica *)
```

```
polarPoints = Table[  
  N[{data[[i]] Cos[i], data[[i]] Sin[i]}, {i, 1, Length[data]}  
];
```

```
(* Função para encontrar o índice correspondente ao ponto mais próximo *)
```

```
findIndex[coord_] :=  
  Nearest[polarPoints -> Automatic, N[coord], 1][[1]];
```

```
(* Calcular o número primo correspondente dado x e y *)
```

```

getPrimeFromCoordinates[x_, y_] := Module[
  {index},
  index = findIndex[{x, y}];
  If[index <= Length[primos],
    primos[[index]],
    "Prime Not Found"]
];
(* Substitua x e y pelas coordenadas numéricas *)
(* Assuming polarPoints is already calculated *)
xCoords = polarPoints[[All, 1]];

(* Alternatively, using Transpose *)
xCoords = Transpose[polarPoints][[1]];

```

```

(* Display the x-coordinates *)

```

```

xCoords

```

```

(* Assuming polarPoints is already calculated *)
yCoords = polarPoints[[All, 2]];

```

```

(* Alternatively, using Transpose *)
yCoords = Transpose[polarPoints][[2]];

```

```

(* Display the x-coordinates *)

```

```

yCoords

```

```

queryPoints = Transpose[{xCoords, yCoords}];
(* Ensure query points are formatted as {x, y} pairs *)
queryPoints = Transpose[{xCoords, yCoords}];

```

```

(* Get the nearest prime for each query point *)
results = Table[

```

```
getPrimeFromCoordinates[queryPoints[[i, 1]], queryPoints[[i, 2]],  
{i, 1, Length[queryPoints]}  
];
```

results

```
Length[xCoords] === Length[yCoords]
```

```
{3856.04, -2970.38, -7066.38, -4666.89, 2024.73, 6855.44, 5381.21, \  
-1038.84, -6507.12, -5989.14, 31.6075, 6024.96, 6477.19, 976.278, \  
-5425.55, -6839.43, -1964.06, 4715.86, 7059.17, 2912.81, -3911.78, \  
-7139.53, -3805.39, 3030.26, 7077., 4617.59, -2085.82, -6870.9, \  
-5340.99, 1102.87, 6531.09, 5957.87, -94.7669, -6063.72, -6450.39, \  
-913.893, 5466.44, 6819.05, 1904.31, -4763.15, -7047.44, -2858.22, \  
3962.3, 7138.69, 3749.65, -3089.13, -7093.03, -4570.51, 2145.57, \  
6889.68, 5300.33, -1163.4, -6561.88, -5922.78, 158.025, 6093.54, \  
6423.08, 851.162, -5505.37, -6798.15, -1844.35, 4815.45, 7039.12, \  
2797., -4015.82, -7147.29, -3697.82, 3145.18, 7090.64, 4521.78, \  
-2206.98, -6909.86, -5257.75, 1226.37, 6581.13, 5887.22, -221.28, \  
-6124.55, -6400.65, -788.807, 5543.84, 6786.22, 1781.17, -4856.6, \  
-7028.27, -2740.3, 4070.19, 7135.34, 3641.55, -3199.16, -7107.55, \  
-4475.2, 2266.38, 6925.64, 5213.3, -1288.6, -6612.78, -5847.93, \  
284.871, 6158.52, 6374.1, 725.506, -5586.55, -6758.57, -1720.88, \  
4905.5, 7018.83, 2681.82, -4118.53, -7134.82, -3589.08, 3255.53, \  
7104.03, 4428.22, -2328.18, -6935.04, -5169.92, 1350.73, 6630.97, \  
5811.47, -347.838, -6188.56, -6341.71, -662.776, 5628.86, 6743.65, \  
1660.39, -4949.92, -7006.87, -2623.13, 4171.14, 7133.74, 3534.29, \  
-3311.65, -7115.87, -4373.48, 2389.22, 6957.51, 5131.87, -1411.96, \  
-6657.91, -5774.56, 410.557, 6218.09, 6315.92, 600.294, -5662.85, \  
-6716.87, -1597.47, 4999.51, 6992.41, 2562.79, -4222.27, -7142.08, \  
-3478.24, 3369.4, 7123.17, 4325.78, -2445.35, -6967.76, -5083.36, \  
1476.31, 6675.01, 5740.41, -473.766, -6254.12, -6282.57, -537.059, \  
5701.11, 6698.98, 1535.38, -5041.74, -6977.4, -2505.13, 4276.66, \  
}
```

7119.93, 3425.79, -3422.14, -7123.93, -4281.27, 2504.63, 6977.45, \

5037.35, -1536.45, -6699.06, -5704.18, 536.823, 6282.67, 6266.27, \

473.286, -5743.76, -6678.68, -1475.27, 5083.51, 6967.72, 2446.52, \

-4328.37, -7117.17, -3368.26, 3479.4}

{6005.42, 6490.4, 1007.29, -5403.43, -6844.62, -1994.97, 4689.45, \

7063.83, 2943.27, -3883.12, -7141.74, -3831.03, 2999.07, 7072.75, \

4644.23, -2056.15, -6862.28, -5363.41, 1070.1, 6516.43, 5975.24, \

-63.1967, -6043.55, -6469.28, -944.967, 5443., 6828.35, 1933.67, \

-4738.22, -7064.24, -2884.75, 3938.19, 7137.18, 3780.73, -3056.29, \

-7083.1, -4596.03, 2116.02, 6883.25, 5321.46, -1132.22, -6549.29, \

-5937.05, 126.388, 6073.59, 6445.81, 883.278, -5485.2, -6807.71, \

-1873.31, 4786.65, 7042.36, 2829.21, -3990.77, -7140.06, -3724.82, \

3113.26, 7090.91, 4546.19, -2175.68, -6903.7, -5285., 1194.89, \

6566.97, 5903.4, -189.836, -6109.96, -6416.42, -819.313, 5525.44, \

6792.25, 1813.27, -4833.38, -7035.73, -2768.69, 4043.04, 7140.39, \

3669.71, -3172.66, -7102.14, -4496.02, 2238.27, 6912., 5236.31, \

-1257.15, -6595.17, -5870.91, 252.737, 6139.02, 6382.97, 757.579, \

-5568.36, -6770.56, -1752.03, 4878.36, 7024.6, 2713.37, -4092.69, \

-7148.14, -3616.37, 3230.09, 7104.87, 4449.27, -2295.68, -6931.38, \

-5196.02, 1319.68, 6619.16, 5829.76, -315.973, -6174.47, -6354.4, \

-693.666, 5610.89, 6755.9, 1689.24, -4924.31, -7012.92, -2652.5, \

4144.29, 7139.34, 3560.72, -3284.54, -7113., -4402.11, 2357.39, \

6950.23, 5150.96, -1381.75, -6642.64, -5794.7, 379.219, 6205.99, \

6325.34, 631.289, -5643.55, -6738.8, -1629.62, 4975.45, 6996.77, \

2595.15, -4195.57, -7128., -3504.82, 3342.42, 7124.57, 4348.46, \

-2414.96, -6958.81, -5111.23, 1443.31, 6661.87, 5757.55, -443.09, \

-6235.28, -6299.3, -568.439, 5685.22, 6708., 1567.09, -5016.47, \

-6995.74, -2533.99, 4248.9, 7126.08, 3452.54, -3396.29, -7127.61, \

-4298.17, 2475.71, 6970.72, 5063.95, -1504.73, -6688.04, -5724.76, \

505.016, 6271.09, 6263.97, 505.085, -5731.3, -6686.09, -1504.52, \

5059.85, 6974.58, 2475.5, -4300.75, -7121.61, -3396.1, 3459.5, \

7122.11, 4251.11, -2535.61, -6991.87, -5016.31, 1567.3, 6711.83, \

5688.27, -568.176, -6297.64, -6236.93}

{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, \

67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, \

139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, \

223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, \

293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, \

383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, \

463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, \

569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, \

647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, \

743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, \

839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, \

941, 947, 953, 967, 971, 977, 983, 991, 997, 1009, 1013, 1019, 1021, \

1031, 1033, 1039, 1049, 1051, 1061, 1063, 1069, 1087, 1091, 1093, \

1097, 1103, 1109, 1117, 1123, 1129, 1151, 1153, 1163, 1171, 1181, \

1187, 1193, 1201, 1213, 1217, 1223}