A LINGUAGEM C NA OBTENÇÃO DOS PRIMOS GÊMEOS: UMA ABORDAGEM DESCRITIVA NA TEORIA DOS NÚMEROS

THE LANGUAGE C TO OBTAINING TWIN PRIMES: DESCRIPTIVE APPROACH IN THE NUMBER THEORY

Alessandro Firmiano de Jesus*
Rafael Ayres Claudino**

RESUMO

A ocorrência e a distribuição dos números primos é assunto que ainda intriga o verdadeiro espírito matemático. Diversas denominações e características deste simples conceito numérico surgem nas mais variadas aplicações científicas, pois, a sua intrínseca versatilidade percorre diferentes épocas do seu contínuo desenvolvimento acadêmico. Neste trabalho, para melhor compreensão dos conceitos, dois códigos computacionais escritos em linguagem C são disponibilizados com o objetivo de permitir a exploração desta intrigante abstração. Numa particular descrição de resultados relacionados aos primos gêmeos, uma aplicação robusta do programa identificou importantes diferenças na densidade de ocorrência desses pares, quando os valores numéricos para o teste de primalidade tornaram-se arbitrariamente grandes.

Palavras-chave: Código numérico C. Teste de primalidade. Decomposição em fatores primos. Densidade dos primos gêmeos.

ABSTRACT

The occurrence and distribution of primes is subject still intrigues the true mathematical mind. Different names and characteristics of this simple numerical concept arise in various scientific applications, since its intrinsic versatility goes through different periods of its ongoing academic development. In this study, to better understand the concepts, two computational codes written in C language are provided with the purpose of enabling the exploration of this intriguing abstraction. A particular description of results related with the twin prime, a robust implementation of the program identified significant differences in the density of occurrence of these pairs, while the numerical values for the primality test become arbitrarily large.

Keywords: Numeric code C. Primality test. Prime factors decomposition. Density of twin primes.

---

* Docente na Academia da Força Aérea AFA-Pirassununga/SP. alessandroafj@afa.aer.mil.br
** Bolsista de Iniciação Científica da FATECE-Pirassununga/SP. rafael-claudino@hotmail.com

Introdução

Os números primos descrevem um mundo fascinante em relação aos vários resultados que foram descritos sob a simplicidade contida na sua definição: "Um número natural $p \neq 1$ é *primo* se os seus únicos divisores são 1 e p". Um número natural que não é primo é chamado de número *composto*. A unidade 1 (do grego *monad*) não é considerada nem primo e nem composto, e esta denominação peculiar não está associada com nenhuma relação de parentesco entre os números, mas sim com o conceito de números primários (do grego *protoi arithmói*). Nesse contexto, um número primário é aquele que não pode ser gerado, através da operação de multiplicação, por outro número primário.

Desde a remota época de **Pitágoras** (Samos, 570 a.C. - Metaponto, 497 a.C.), passando pelos mais renomados matemáticos da história, tais como **Euclides** (Alexandria, 325 a.C. - 265 a.C), **Eratóstenes** (Cirene, 276 a.C. - Alexandria, 194 a.C.), **Fermat** (Beaumont-de-Lomagne, 1601 - Castres, 1665), **Euler** (Basileia, 1707 - São Petersburgo, 1783), **Gauss** (Braunschweig, 1777 - Göttingen, 1855), **Ramanujan** (Erode, 1887 - Kumbakonam, 1920) e ainda nos dias atuais, com o desenvolvimento contínuo da computação algébrica e das técnicas de criptografia RSA, a pesquisa sobre propriedades intrínsecas e aplicações variadas dos primos sempre esteve em um patamar de destaque (BURTON, 1989).

Durante estes árduos estudos, outras terminologias associadas aos números primos foram estabelecidas em função dos vários contextos e das diferentes épocas que buscavam a demonstração de relevantes resultados numéricos. Segundo Wells (2005), os mais conhecidos e os mais curiosos são:

- **Primos gêmeos** – são os pares de números primos da forma $p,\ p+2$. A quantidade infinita de pares de primos gêmeos é ainda uma questão em aberto. Viggo Brun (1919) provou que a soma dos recíprocos de todos os primos gêmeos converge para a constante de Brun, ou seja,

$$\left(\frac{1}{3}+\frac{1}{5}\right)+\left(\frac{1}{5}+\frac{1}{7}\right)+\left(\frac{1}{11}+\frac{1}{13}\right)+\left(\frac{1}{17}+\frac{1}{19}\right)+\left(\frac{1}{29}+\frac{1}{31}\right)+\left(\frac{1}{41}+\frac{1}{43}\right)+\cdots$$
$$=1{,}9021605\ldots$$ [1]

[1] Euler provou a existência de infinitos números primos demonstrando que a soma dos inversos de todos os primos diverge.

- **Primos "primos"**[2] – são os pares de números primos da forma $p,\ p+4$. É facilmente verificada a existência de apenas 26 pares de primos "primos" menores do que 1000.
- **Primos sexy** – são os pares de números primos da forma $p,\ p+6$. O termo *sexy* vem do latim para a tradução tendenciosa do número "*six*" como sendo "sexo". A tripla $(47{,}53{,}59)$ é um exemplo de uma tripla sexy, a quádrupla $(61{,}67{,}73{,}79)$ é um exemplo de uma quádrupla sexy, a sequência $(5{,}11{,}17{,}23{,}29)$ é a única quíntupla sexy conhecida e não existe nenhuma sêxtupla sexy.
- **Primos siameses** – são os pares de números primos da forma $n^2-2,\ n^2+2$. Para $n=33$, segue que $(1087{,}1091)$ é um par de primos siameses e para $n=117$, $(13.687{,}13.691)$ é outro par.
- **Semiprimos** (ou 2-quase-primo) – são os números compostos cuja decomposição possui apenas dois fatores primos, repetidos ou não. Chen provou em 1975 que sempre existe um semiprimo entre n e $\sqrt{n}$ (Greaves, 2001). Segundo Bernedt (1998), todo número é a soma de, no máximo, seis primos, e que todo número suficientemente grande é a soma de um primo com um semiprimo.
- **Primos de Sophie Germain** – Se p e $2p+1$ são ambos primos, então p é um primo de Sophie Germain. Pois, em 1832, esta proeminente matemática provou que, para p primo, $x^p+y^p=z^p$ não tem solução se $2p+1$ também é primo. Ou seja, Sophie simplesmente demonstrou uma importante particularidade do 1º caso do Último Teorema de Fermat (DERBYSHIRE, 2003).
- **Primos seguros** – os primos de Sophie Germain também estão relacionados à criptografia. Se p e $q=2p+1$ são primos, então q é considerado um primo seguro, pois $q-1$ não tem muitos fatores e portanto não pode ser facilmente fatorado, tornando seguro o processo de encriptação de dados. Até o momento, o maior primo seguro conhecido é $18543637900515 \text{ x } 2^{666668}-1$.
- **Primo iccanobiF** – é todo primo representado por um número de Fibonacci reverso[3], por exemplo, 31, 773 e 52057. Substituindo cada dígito do primo 7-7-3 pelos seus quadrados 49-49-9 e também pelos seus cubos 343-343-27, dois outros primos são

[2] Do inglês *Cousin primes*.
[3] The Prime Glossary, disponível em http://primes.utm.edu/glossary/xpage/iccanobiF.html, acesso em dezembro 2013.

obtidos: 49499 e 34334327, e ainda, o segundo número é um **omirp**, isto é, seus dígitos reversos 72343343 também é um número primo.

- **Primo palíndromo** – é todo número palíndromo que também é primo. A obra de Honaker Jr. (2000) apresenta a lista ao lado de primos palíndromos compondo uma interessante pirâmide numérica.
  Um caso particular é quando os dígitos do primo vão aumentando e depois começam a diminuir, por exemplo, o primo palíndromo 12421. Neste caso, o número é conhecido por **primo montanha**.

2
30203
133020331
1713302033171
12171330203317121
151217133020331712151
1815121713302033171215181
16181512171330203317121518161
331618151217133020331712151816133
9333161815121713302033171215181613339
11933316181512171330203317121518161333911

- **Primo James Bond** – foi definido como sendo todo primo que termina em 007, tais como, 4007, 6007, 9007 e 171007. Uma sequência deste tipo de primo e, consequentemente, qualquer outra sequência de inteiros pode ser obtida na home page *Sloane's On-Line Encyclopedia of Integer Sequences* através do link: http://oeis.org/.

Com toda esta rica diversidade obtida a partir de um conceito simples que são os números primos, o objetivo deste trabalho é apresentar dois códigos computacionais escrito em linguagem C, um para obter a listagem de números primos dentro de um intervalo determinado, e outro para obter a decomposição em fatores primos de um número composto qualquer.

Evidentemente, esses programas possuem limitações computacionais que se referem ao tamanho do número inteiro a ser fornecido como dado de entrada (YAN, 2001). No entanto, de aplicação educacional e recreativa, a disponibilização dos códigos neste artigo permite a imediata investigação da primalidade de um número com as propriedades e as características previamente sugeridas.

Para ilustração dos códigos será discutida a obtenção de pares de primos gêmeos até o limite de representação de um número inteiro pelo computador comum. No entanto, para instigar novas motivações, são listados *a priori* os seguintes problemas em aberto e oriundos do ramo da Teoria dos Números que envolve esses fascinantes números primos (PANTOJA, 2012).

a.) A Conjectura de Goldbach: "Todo número par $n > 2$ é a soma de dois primos".
b.) O Problema Ímpar de Goldbach: "Todo número ímpar $n > 5$ é a soma de três primos".
c.) "Todo número par é a diferença de dois primos".
d.) "Para todo número par $2n$ existe uma quantidade infinita de pares de primos consecutivos que se diferem por $2n$".
e.) Conjectura dos Primos Gêmeos ($n = 1$ do caso **d.)**): "Existem infinitos Primos Gêmeos".
f.) "Existem infinitos números primos da forma $n^2 + 1$".
g.) "A quantidade de números primos de Fermat[4] é finita".
h.) "Sempre existe um número primo entre n^2 e $(n + 1)^2$".

Códigos e Método

Para verificação simples da primalidade de um número $n > 2$ foi desenvolvido um código em linguagem C que procura, caso exista, um divisor d tal que $2 \leq d \leq \sqrt{n}$. Enquanto o código não determina nenhum divisor neste intervalo, um contador é incrementado a cada passo e se superar $\sqrt{n}$ o número n é considerado primo, conforme descrito por Coutinho (2013) ao apresentar o Crivo de Eratóstenes. Caso o código encontra um divisor d o número é classificado como sendo composto.

As linhas de comando abaixo descrevem o teste de primalidade feito por exaustão e que serve de base para a obtenção dos dois importantes códigos do artigo que foram citados anteriormente.

```
int main(int argc, char *argv[]) {
        unsigned long long int x,y,n,i;
        i=0;
        while(i==0){
        printf("digite o numero natural para o teste de primalidade\n");
        scanf("%I64d",&n);
        x=sqrt(n);
        for(y=2;y<=x;y++){
                    if(n%y==0){
                        printf("O NUMERO E COMPOSTO\n");
                        y=x+1;
                    }
                    if(y==x){
                        printf("O NUMERO E PRIMO\n");
                    }
        }
}
```

[4] São os números primos da forma $p = 2^{2^n} + 1$

A decomposição em fatores primos de um número natural fornecido é feita pelo código abaixo. Neste programa o valor de $n > 2$ é informado e os seus fatores primos com as respectivas potências são os dados de saída. Em Press et al. (1997) é encontrado várias aplicações para esta decomposição.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <limits.h>

int main(int argc, char *argv[])
{
    unsigned long long int x,n,i,d,t,u,f,z,a,b,e;
    int y,c;
    c=0;
    u=1;
    t=1;
    y=1;
   while(c==0){
         printf("DIGITE UM NUMERO PARA OBTER SEUS FATORES PRIMOS\n");
         scanf("%I64d",&n);
         b=sqrt(n);
         for(x=2;x<=b;x++){
                              if(n%x==0){
                                   if(x==2){
                                      d=x*x;
                                      for(y=1;n%d==0;y++){
                                             d=d*x;
                                                       }
                                        printf("o numero primo divisor e: %I64d elevado a: %d\n\n",x,y);
                                                                              t=pow(x,y);//colocando na var "t" a potencia de 2
                                                                              printf("ok1 %lld\n\n",t);
                                                                              f=n/t;
                                                                              e=sqrt(f);
                                                                              for(i=2;i<=e;i++){
                                                                                             if(f%i==0){
                                                                                                   i=e+1;
                                                                                             }
                                                                                             if(i==e){
                                                                                                     d=f*f;
                                                 for(y=1;n%d==0;y++){
                                                         d=d*f;
                                                                         }
                                                 printf("o numero primo divisor e:%I64d elevado a:%d\n\n",f,y);
                                                                 u=u*pow(f,y);//colocando a variavel u a potencia dos primos multiplos
                                                                                                                         printf("ok2
%lld\n\n",u);
                                                                                                       }
                                                                              }
                                     }
                              else{
                              //if(n%x==0){
                                     for(i=2;i<x;i++){
                                                 if(x%i==0){
                                                      i=x;
                                                          }
                                               if(i==x-1){
                                                    d=x*x;
                                                    for(y=1;n%d==0;y++){
                                                            d=d*x;
                                                                            }
                                                    printf("o numero primo divisor e:%I64d elevado a:%d\n\n",x,y);
                                                                u=u*pow(x,y);//colocando a variavel u a potencia dos primos multiplos
                                                                                                                         printf("ok2
%lld\n\n",u);
                                                                                                                         f=t*u;
```

```
                                                                a=n/f;
                                                                e=sqrt(a);
                                                                   }
                                                                         }
                                                 for(i=2;i<=e;i++){
                                                                     if(a%i==0){

     i=e+1;
                                                                      }
                                                                     if(i==e){

     d=a*a;

     for(y=1;n%d==0;y++){

     d=d*a;
                                                                       }
                                         printf("\no segundo primo divisor e:%I64d elevado a:%d\n\n",a,y);
                                                                     x=b+1;
                                                                     a=0;
                                                                  }
                                                                  }
                              }
                             }
                          if(f==n){
                                             printf("acabo a variavel f e:%I64d\n\n",f);
                                             x=b+1;
                                     }
                         if(x==b){
                            printf("O NUMERO E PRIMO\n\n");
                         }
                         }
u=1;   t=1;   f=0;
       }
c=0;
  system("PAUSE");
  return 0;
}
```

Na investigação dos pares de primos gêmeos, ao identificar que um número p é primo a estratégia do código aplica o teste de primalidade para o valor $p + 2$. Em caso afirmativo, os primos gêmeos são obtidos e ainda o teste continua a ser aplicado no valor $p + 4$ para a possível identificação de primos trigêmeos. As linhas de comando deste código dos Primos Gêmeos são:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
        unsigned long long int i,raiz1,raiz2,raiz3,primo,gemeo1,gemeo2,gemeo3;
        FILE *arquivo;
        arquivo=fopen("numerosGemeos3.txt","a");
        for(primo=3;primo<100000000000000000;primo=primo+2)
        {
                raiz1=sqrt(primo);
                for(i=2;i<=raiz1;i++){
                        if(primo%i==0)
                        {
                                i=raiz1+1;
                        }
```

```
                    if(i==raiz1)
                    {
                      gemeo1=primo;
                      gemeo2=primo+2;
                      raiz2=sqrt(gemeo2);
        for(i=2;i<=raiz2;i++)
        {
              if(gemeo2%i==0)
              {
                    i=raiz2+1;
              }
              if(i==raiz2)
              {
               fprintf(arquivo,"%I64d %I64d\n",gemeo1,gemeo2);
                      printf("%I64d %I64d\n",gemeo1,gemeo2);
                      gemeo3=primo+4;
                      raiz3=sqrt(gemeo3);
                      for(i=2;i<=raiz3;i++)
                      {
                              if(gemeo3%i==0)
                              {
                                      i=raiz3+1;
                              }
                              if(i==raiz3)
                              {
                                  fprintf(arquivo,"%I64d %I64d %I64d\n",gemeo1,gemeo2,gemeo3);
                            printf("%I64d %I64d %I64d\n",gemeo1,gemeo2,gemeo3);
                  system("PAUSE");
                    return 0;
                              }
                      }
              }
        }
                    }
              }
  }
  fclose(arquivo);
        system("PAUSE");
        return 0;
}
```

São necessárias apenas pequenas modificações nestes códigos bases para implementar as outras denominações ou características dos demais números primos curiosos que foram descritos no início deste trabalho. Segundo Knuth (1981) e Shoup (2005) estas adaptações são técnicas iniciais de programação.

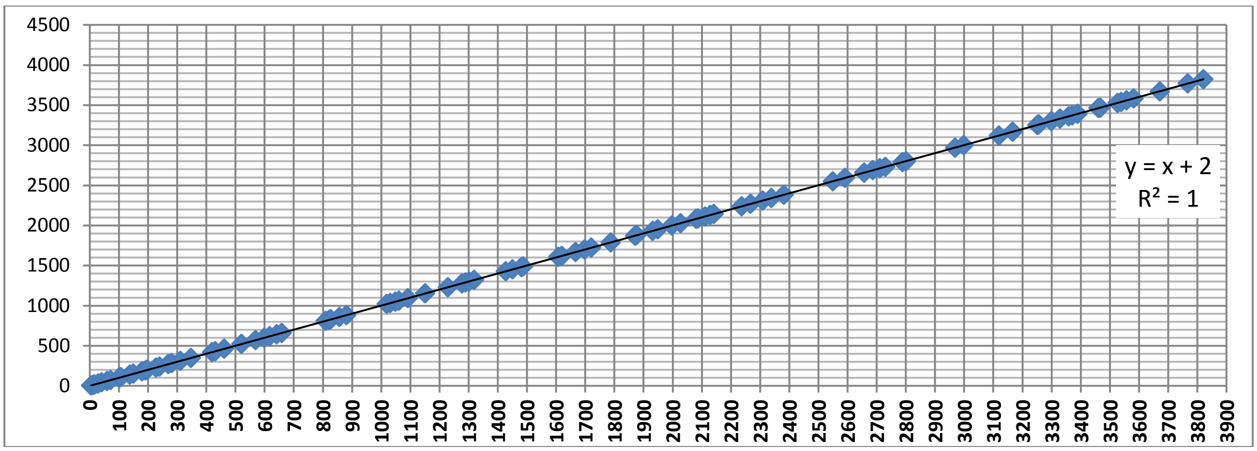
Resultados sobre os Primos Gêmeos

A saída do programa Primos Gêmeos descreveu, em um bloco de notas, um arquivo .txt[5] com mais de 20.400 páginas contendo os pares e a posição que os primos gêmeos ocupam, conforme a tabela 1, para os 100 primeiros pares e a tabela 2 para os 134 últimos

[5] Arquivo de 34,2 MB disponível na web através do link: https://www.dropbox.com/s/c5nsc1ujfc40x3s/numerosGemeos.txt

pares. Na realização de um cálculo médio da quantidade de números em que o programa aplica o teste de primalidade, constatou-se a impressão de cerca de 1.000.000 de primos gêmeos por hora sendo que o 1º é (3,5) e o 1.000.000º é (252.427.601, 252.427.603). E ainda, com a disponibilização dos primos gêmeos neste arquivo .txt, é possível plotar a ocorrência destes pares em um gráfico de dispersão, conforme visto no Gráfico 1.

Gráfico 1 – Alinhamento geométrico e densidade observada no gráfico de dispersão dos 100 primeiros pares de primos gêmeos



Neste gráfico, além do alinhamento esperado sobre a equação da reta $y = x + 2$, observa-se uma interessante densidade dos pares de primos gêmeos sobre a linha de tendência.

Tabela 1 – Identificação dos 100 primeiros pares de números primos gêmeos obtidos pelo código em linguagem C

3	5	7	posicao:1	569	571	posicao:26	1607	1609	posicao:51	2687	2689	posicao:76
	5	7	posicao:2	599	601	posicao:27	1619	1621	posicao:52	2711	2713	posicao:77
	11	13	posicao:3	617	619	posicao:28	1667	1669	posicao:53	2729	2731	posicao:78
	17	19	posicao:4	641	643	posicao:29	1697	1699	posicao:54	2789	2791	posicao:79
	29	31	posicao:5	659	661	posicao:30	1721	1723	posicao:55	2801	2803	posicao:80
	41	43	posicao:6	809	811	posicao:31	1787	1789	posicao:56	2969	2971	posicao:81
	59	61	posicao:7	821	823	posicao:32	1871	1873	posicao:57	2999	3001	posicao:82
	71	73	posicao:8	827	829	posicao:33	1877	1879	posicao:58	3119	3121	posicao:83
	101	103	posicao:9	857	859	posicao:34	1931	1933	posicao:59	3167	3169	posicao:84
	107	109	posicao:10	881	883	posicao:35	1949	1951	posicao:60	3251	3253	posicao:85
	137	139	posicao:11	1019	1021	posicao:36	1997	1999	posicao:61	3257	3259	posicao:86
	149	151	posicao:12	1031	1033	posicao:37	2027	2029	posicao:62	3299	3301	posicao:87
	179	181	posicao:13	1049	1051	posicao:38	2081	2083	posicao:63	3329	3331	posicao:88
	191	193	posicao:14	1061	1063	posicao:39	2087	2089	posicao:64	3359	3361	posicao:89
	197	199	posicao:15	1091	1093	posicao:40	2111	2113	posicao:65	3371	3373	posicao:90
	227	229	posicao:16	1151	1153	posicao:41	2129	2131	posicao:66	3389	3391	posicao:91
	239	241	posicao:17	1229	1231	posicao:42	2141	2143	posicao:67	3461	3463	posicao:92
	269	271	posicao:18	1277	1279	posicao:43	2237	2239	posicao:68	3467	3469	posicao:93
	281	283	posicao:19	1289	1291	posicao:44	2267	2269	posicao:69	3527	3529	posicao:94
	311	313	posicao:20	1301	1303	posicao:45	2309	2311	posicao:70	3539	3541	posicao:95
	347	349	posicao:21	1319	1321	posicao:46	2339	2341	posicao:71	3557	3559	posicao:96
	419	421	posicao:22	1427	1429	posicao:47	2381	2383	posicao:72	3581	3583	posicao:97
	431	433	posicao:23	1451	1453	posicao:48	2549	2551	posicao:73	3671	3673	posicao:98
	461	463	posicao:24	1481	1483	posicao:49	2591	2593	posicao:74	3767	3769	posicao:99
	521	523	posicao:25	1487	1489	posicao:50	2657	2659	posicao:75	3822	± 1	posicao:100

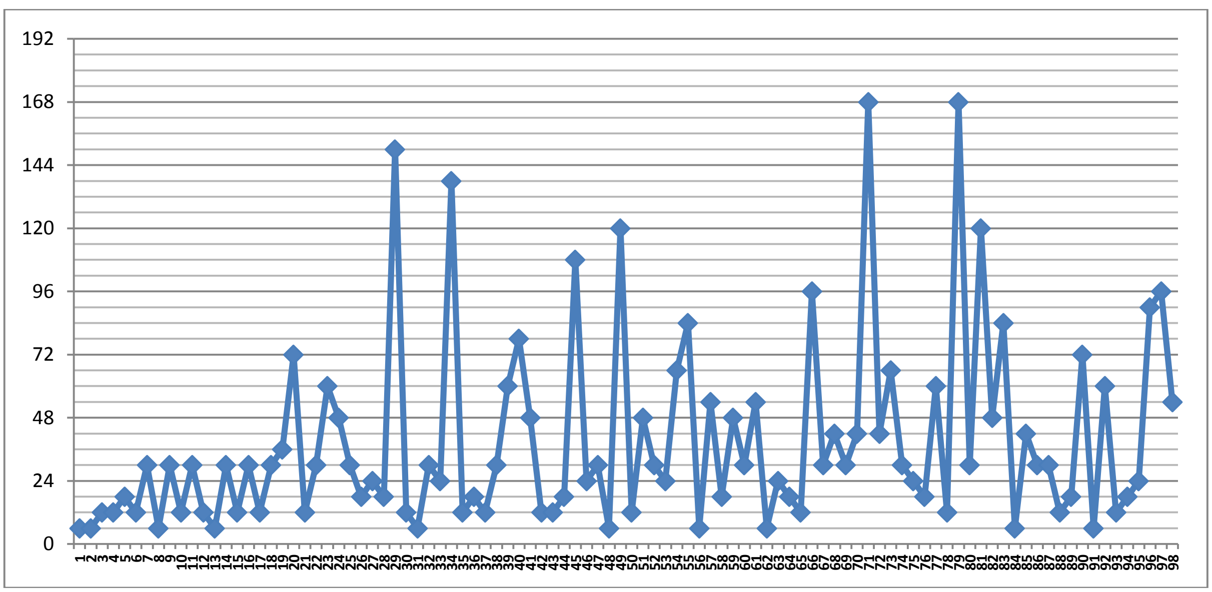
Através de inspeção conjunta nos valores da Tabela 1 com a respectiva disposição geométrica do Gráfico 1, observa-se a inexistência de primos gêmeos nos seguintes intervalos de amplitude $h = 100$:

$$[700{,}800],\ [900{,}1000],\ [2400{,}2500]\ [2800{,}2900] \text{ e } [3000{,}3100].$$

Para a verificação de como os primos gêmeos da Tabela 1 estão distribuídos ao longo da linha de tendência do Gráfico 1, um novo gráfico de linhas é obtido com os valores da diferença entre o maior primo gêmeo do par com o maior primo gêmeo do par anterior (ver Gráfico 2).

Como esperado, todos os valores das diferenças do gráfico de linhas do Gráfico 2 são múltiplos de 6, e ainda, o maior valor encontrado foi 168.

Gráfico 2 – Gráfico de linhas das diferenças entre os maiores primos gêmeos de pares consecutivos. Neste intervalo o menor valor observado foi igual a 6 e maior valor observado foi igual a 168.



Para enriquecer essas discussões, foi selecionado do arquivo .txt, os maiores pares de primos gêmeos que estavam disponíveis e ocupavam posições superiores a 1.000.000, conforme a Tabela 2.

Tabela 2 – Identificação dos 134 pares de primos gêmeos ocupantes das posições superiores a 1.000.000 obtidos pelo código C

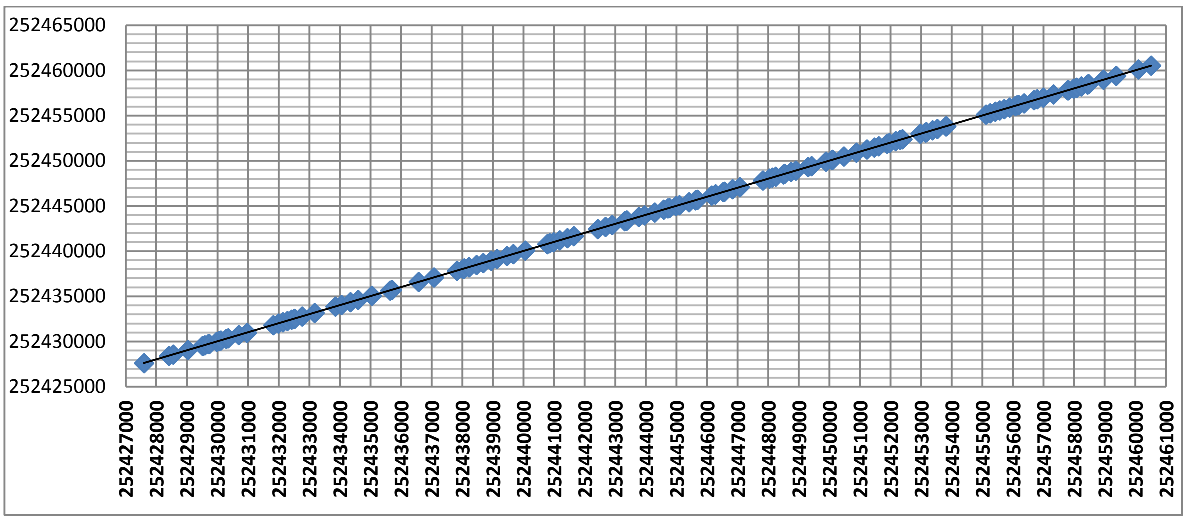
252427601 252427603	posicao:1000000	252439667 252439669	posicao:1000045	252450101 252450103	posicao:1000090
252428417 252428419	posicao:1000001	252440051 252440053	posicao:1000046	252450467 252450469	posicao:1000091
252428549 252428551	posicao:1000002	252440777 252440779	posicao:1000047	252450881 252450883	posicao:1000092
252429029 252429031	posicao:1000003	252440861 252440863	posicao:1000048	252451229 252451231	posicao:1000093
252429509 252429511	posicao:1000004	252440987 252440989	posicao:1000049	252451469 252451471	posicao:1000094
252429521 252429523	posicao:1000005	252441179 252441181	posicao:1000050	252451607 252451609	posicao:1000095
252429587 252429589	posicao:1000006	252441437 252441439	posicao:1000051	252451889 252451891	posicao:1000096
252429701 252429703	posicao:1000007	252441641 252441643	posicao:1000052	252451961 252451963	posicao:1000097
252429731 252429733	posicao:1000008	252442427 252442429	posicao:1000053	252452171 252452173	posicao:1000098
252429971 252429973	posicao:1000009	252442679 252442681	posicao:1000054	252452309 252452311	posicao:1000099
252430049 252430051	posicao:1000010	252442901 252442903	posicao:1000055	252452381 252452383	posicao:1000100
252430109 252430111	posicao:1000011	252443309 252443311	posicao:1000056	252452969 252452971	posicao:1000101
252430289 252430291	posicao:1000012	252443369 252443371	posicao:1000057	252453137 252453139	posicao:1000102
252430349 252430351	posicao:1000013	252443381 252443383	posicao:1000058	252453167 252453169	posicao:1000103
252430697 252430699	posicao:1000014	252443771 252443773	posicao:1000059	252453371 252453373	posicao:1000104
252430961 252430963	posicao:1000015	252443957 252443959	posicao:1000060	252453527 252453529	posicao:1000105
252431819 252431821	posicao:1000016	252444287 252444289	posicao:1000061	252453809 252453811	posicao:1000106
252431987 252431989	posicao:1000017	252444587 252444589	posicao:1000062	252453821 252453823	posicao:1000107
252432137 252432139	posicao:1000018	252444719 252444721	posicao:1000063	252455111 252455113	posicao:1000108
252432281 252432283	posicao:1000019	252444767 252444769	posicao:1000064	252455249 252455251	posicao:1000109
252432419 252432421	posicao:1000020	252444977 252444979	posicao:1000065	252455417 252455419	posicao:1000110
252432491 252432493	posicao:1000021	252445097 252445099	posicao:1000066	252455561 252455563	posicao:1000111
252432527 252432529	posicao:1000022	252445409 252445411	posicao:1000067	252455711 252455713	posicao:1000112
252432767 252432769	posicao:1000023	252445619 252445621	posicao:1000068	252455891 252455893	posicao:1000113
252433169 252433171	posicao:1000024	252445637 252445639	posicao:1000069	252456077 252456079	posicao:1000114
252433847 252433849	posicao:1000025	252445691 252445693	posicao:1000070	252456131 252456133	posicao:1000115
252434051 252434053	posicao:1000026	252446147 252446149	posicao:1000071	252456179 252456181	posicao:1000116
252434339 252434341	posicao:1000027	252446267 252446269	posicao:1000072	252456359 252456361	posicao:1000117
252434597 252434599	posicao:1000028	252446279 252446281	posicao:1000073	252456689 252456691	posicao:1000118
252435047 252435049	posicao:1000029	252446531 252446533	posicao:1000074	252456791 252456793	posicao:1000119
252435641 252435643	posicao:1000030	252446567 252446569	posicao:1000075	252456989 252456991	posicao:1000120
252435707 252435709	posicao:1000031	252446837 252446839	posicao:1000076	252457319 252457321	posicao:1000121
252436571 252436573	posicao:1000032	252447077 252447079	posicao:1000077	252457781 252457783	posicao:1000122
252437069 252437071	posicao:1000033	252447821 252447823	posicao:1000078	252457811 252457813	posicao:1000123
252437831 252437833	posicao:1000034	252448037 252448039	posicao:1000079	252457979 252457981	posicao:1000124
252438029 252438031	posicao:1000035	252448127 252448129	posicao:1000080	252458069 252458071	posicao:1000125
252438059 252438061	posicao:1000036	252448241 252448243	posicao:1000081	252458081 252458083	posicao:1000126
252438071 252438073	posicao:1000037	252448499 252448501	posicao:1000082	252458231 252458233	posicao:1000127
252438209 252438211	posicao:1000038	252448541 252448543	posicao:1000083	252458429 252458431	posicao:1000128
252438227 252438229	posicao:1000039	252448751 252448753	posicao:1000084	252458471 252458473	posicao:1000129
252438467 252438469	posicao:1000040	252448907 252448909	posicao:1000085	252458951 252458953	posicao:1000130
252438689 252438691	posicao:1000041	252449297 252449299	posicao:1000086	252459377 252459379	posicao:1000131
252438941 252438943	posicao:1000042	252449411 252449413	posicao:1000087	252460097 252460099	posicao:1000132
252439139 252439141	posicao:1000043	252449891 252449893	posicao:1000088	252460511 252460513	posicao:1000133
252439457 252439459	posicao:1000044	252450071 252450073	posicao:1000089		

Se considerar os 200 números primos da Tabela 1, calcula-se que a maior diferença entre eles é $3823 - 3 = 3820$, o que indica uma densidade relativa de pares de primos gêmeos igual à razão $\frac{100}{3820} = 0{,}0262$. Esta mesma análise realizada nos últimos 134 pares de primos gêmeos da Tabela 2, verifica-se que a maior diferença entre estes primos é $252.460.513 - 252.427.601 =$ 32.912, o que indica uma densidade relativa de $\frac{134}{32.912} = 0{,}0041$, bem inferior à densidade calculada na primeira centena de pares de primos gêmeos.

E isto é uma forte evidência de que a distância entre os pares de primos gêmeos está aumentando quando a investigação de sua ocorrência avança para valores de ordem superior.

Outra característica segue da análise conjunta dos valores da Tabela 2 com a disposição geométrica do gráfico de dispersão (Gráfico 3), pois observa-se a inexistência de pares de primos gêmeos no intervalo de amplitude $h = 1000$ definido pelos extremos $[242.554.000, 242.555.000]$.

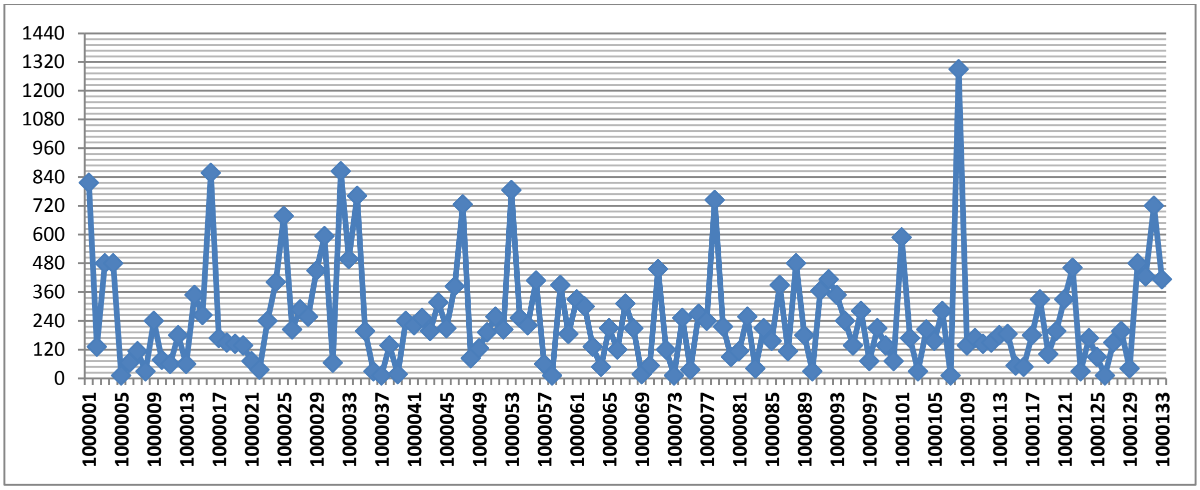
Gráfico 3 – Alinhamento geométrico e densidade observada no gráfico de dispersão dos pares de primos gêmeos com posições superiores a 1.000.000.



O correspondente gráfico de linhas para as diferenças entre os maiores primos gêmeos de pares consecutivos, avaliadas na última centena dos valores disponíveis, está representado no Gráfico 4. Verifica-se que o menor valor encontrado para a diferença é igual a 12. O maior valor, igual a 1290 é, evidentemente, encontrado na mesma região de lacuna dos primos gêmeos visualizado no Gráfico 3.

Verifica-se, finalmente, nas sequências dos pares de primos gêmeos $(p, p + 2)$ que nenhum primo p termina em 3. O que de fato é um resultado esperado, pois $p + 2$ não seria primo por ser um múltiplo de 5. Esta discussão é útil no sentido de deixar o código mais rápido, ou seja, uma vez identificado que um primo p termina em 3, basta reiniciar o processo de teste de primalidade para o valor $p + 4$.

Gráfico 4 – Gráfico de linhas das diferenças entre os maiores primos gêmeos de pares consecutivos. Neste intervalo o menor valor observado foi igual a 12 e o maior valor observado foi igual a 1290.



Restrito apenas à limitação computacional da linguagem de programação C para trabalhar com números grandes, o maior par de primos gêmeos encontrado, que é menor que 100.000.000.000.000.001, foi o par (99.999.999.999.998.807, 99.999.999.999.998.809). E ainda, neste grande intervalo indicado, a única identificação de primos trigêmeos foi a sequência $(3,5,7)$. Um resultado que, mesmo sendo comprovadamente esperado na literatura, foi útil para certificar a validação e confiabilidade do código numérico desenvolvido.

Considerações Finais

Foram disponibilizados dois códigos computacionais, em linguagem C. Um para realizar o teste de primalidade de um número natural e outro que decompõe um número composto em seus fatores primos. Evidentemente, esses códigos abertos são passíveis de atualizações no sentido de torna-los mais rápidos e eficientes, como por exemplo, aplicar uma estratégia de incremento duplo nos candidatos de divisores ímpares no número a ser investigado, uma vez que não faz sentido a busca de divisores pares.

As limitações computacionais mencionadas neste artigo podem ser contornadas com técnicas avançadas de programação que empregam as bibliotecas em C para números inteiros de precisão aritmética arbitrária, tais como, o código livre GNU *Multiple Precision Arithmetic Library*.

Os números primos ainda guardam muito dos seus segredos e dos seus mistérios. É bem possível que certas situações reais, representadas por fenômenos naturais, sociais ou econômicos, possuam o mesmo comportamento arbitrário que foram observados nos gráficos das figuras 2 e 4. Neste sentido, o estudo dos números primos poderia contribuir para uma compreensão sistemática de eventos ao oferecer robustos mecanismos científicos de previsibilidade.

Referências

BERNEDT, B. C. **Ramanujan's Notebooks.** New York. Springer, 1998. Part V.

BRUN, V. **La série 1/5+1/7+1/11+1/13+1/17+1/19+1/29+1/31+1/41+1/43+1/59+1/61+..., ou les dênominateurs sont nombres premiers jumeaux est convergente ou finie**. Bulletin des Sciences Mathématiques, n. 43, p. 100-104,124-128, 1919.

BURTON, D. **Elementary Number Theory**, University of New Hampshire, Wm C. Brown, Durham, 1989.

COUTINHO, S. C. **Números Inteiros e Criptografia RSA**. 2. ed. Rio de Janeiro: Instituto Nacional de matemática Pura e Aplicada – IMPA, 2013. (Coleção Matemática e Aplicações)

DERBYSHIRE, J. **Prime Obsession:** Bernhard Riemann and the Greatest Unsolved Problem in Mathematics. Washington: Joseph Henry Press, 2003.

GREAVES, G. **Sieves in Number Theory**. New York: Springer. 2001.

HONAKER Jr., G. L.; CALDWELL, C. Palindromic Prime Pyramids. **J. Recreational Math.**, v. 3, n. 3, p. 169-176, 2000.

KNUTH, D. E. **The art of computer programming.** Reading, MA: Addison-Wesley. 1981. V. 2. (Semi Numerical Algorithms)

PANTOJA, P. Primos Gêmeos e outras Conjecturas. **Revista Escolar de la Olimpíada Iberoamericana de Matemática**, n. 45, p. 1-11, abr./jun. 2012.

PRESS, W. H. et al. **Numerical Recipes in C:** the art of scientific computing. 2 th. Cambridge: Cambridge University Press, 1997.
SHOUP, V. **A Computational Introduction to Number Theory and Algebra**. Cambridge: Cambridge University Press, 2005.

WELLS, D. **Prime Numbers:** the most mysterious figures in math. New Jersey: John Wiley & Sons. 2005.

YAN, S. **Number Theory for Computing**. 2 th. New York: Springer, 2001.