VIII Encontro Cajazeirense de Matemática

Uma década da licenciatura em matemática no IFPB-CZ: tecendo histórias e interligando culturas!

20 a 22 de outubro de 2021

Artimética com o *Sagemath*

Vinicius Martins Teodosio Rocha
Larissa Soares de Sousa
Jose Jorge de Souza Silva

Instituto Federal da Paraíba - Campus Cajazeiras

O *Sagemath*

- https://sagecell.sagemath.org/
- Quase um Python com **muitas** ferramentas extras.
- Missão

  Criar uma alternativa viável de código aberto para o Magma, Maple, Mathematica e Matlab
- Licença GPL (GNU General Public License): Livre!
- https://doc.sagemath.org/html/en/developer/
- E o que tanto ele faz?
  - doc.sagemath.org/html/pt/tutorial/
  - doc.sagemath.org/html/pt/a_tour_of_sage/
  - doc.sagemath.org/html/en/thematic_tutorials/ só em inglês :(

Gráficos

In [4]:
```
plot(sin(x),-pi,pi)
```
Out[4]:

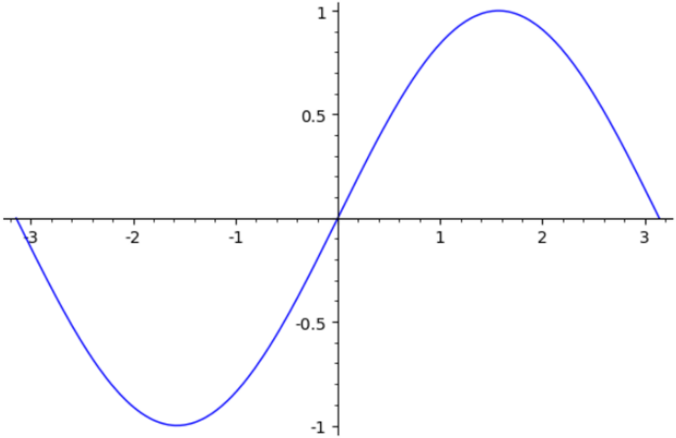


Figure: Gráfico de $\sin(x), x \in [-\pi, \pi]$

Gráficos Interativos

In [11]:

```
@interact
def _(n=slider([1..10])):
    show(plot(x^n, -1,1))
```

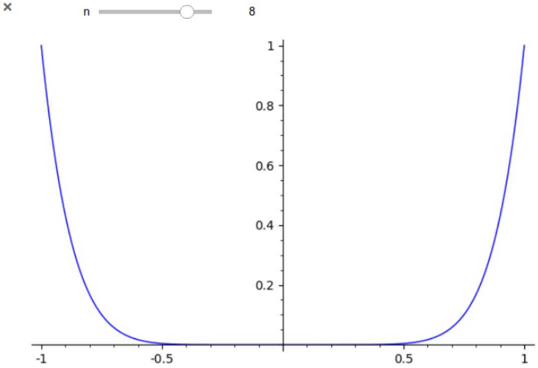


Figure: Gráfico de $x^n, x \in [-1, 1]$, $n = 1, 2, \ldots$.

Gráficos 3D

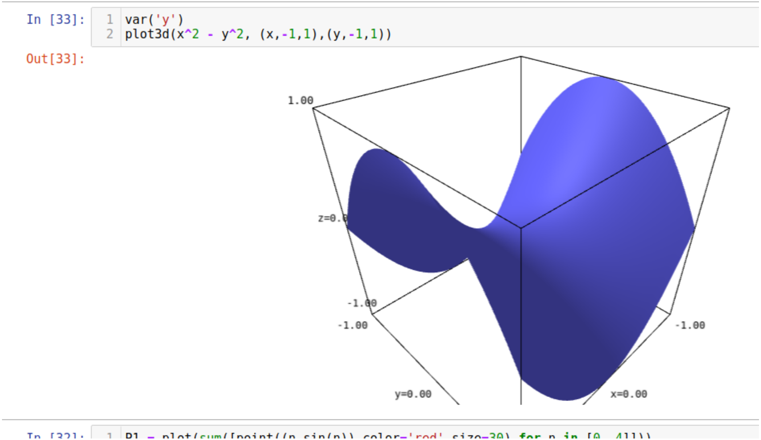
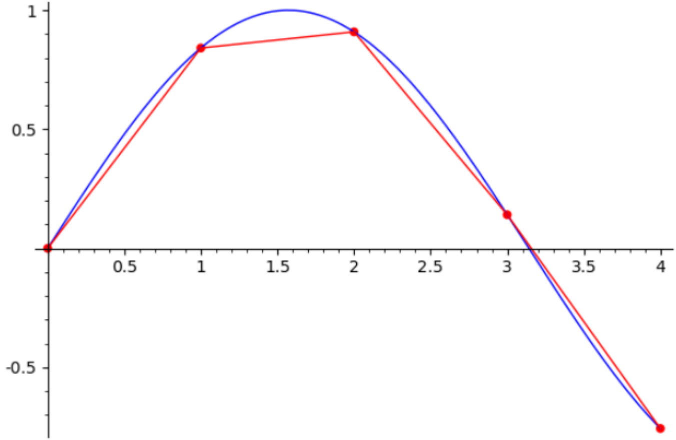


Figure: Um parabolóide hiperbólico.

Muitos gráficos juntos!

```
P1 = plot(sum([point((n,sin(n)),color='red',size=30) for n in [0..4]]))
P2 = plot(sin(x),0,4)
P3 = line([(n,sin(n)) for n in [0..4]],color='red')
P1+P2+P3
```



Cálculo Diferencial e Integral

Cálculo Diferencial e Integral

In [42]:
```
f(x) = (3*x^2 - x + 2)/(5*x - 4)
show(lim(f,x=1))
```

$$x \mapsto 4$$

In [49]:
```
show(f.derivative())
```

$$x \mapsto \frac{6\,x - 1}{5\,x - 4} - \frac{5\left(3\,x^2 - x + 2\right)}{(5\,x - 4)^2}$$

In [52]:
```
show(f.integral(x))
```

$$x \mapsto \frac{3}{10}\,x^2 + \frac{7}{25}\,x + \frac{78}{125}\,\log(5\,x - 4)$$

Figure: Derivadas e integrais

Cálculo Diferencial e Integral

edos, series,?

Matrizes

In [67]:

```
A = matrix([[1,2,4],
            [2,3,1],
            [3,1,5]])
show(A)
```

$$\begin{pmatrix} 1 & 2 & 4 \\ 2 & 3 & 1 \\ 3 & 1 & 5 \end{pmatrix}$$

In [57]:

```
show(det(A))
```

$$-28$$

In [88]:

```
show("A^(-1) = ", A.inverse(), " ; A na forma escalonada: ", A.echelon_form())
```

$$\text{A^(-1) } = \begin{pmatrix} -\frac{1}{2} & \frac{3}{14} & \frac{5}{14} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{5}{28} & \frac{1}{28} \end{pmatrix} \texttt{ ; A na forma escalonada: } \begin{pmatrix} 1 & 0 & 18 \\ 0 & 1 & 7 \\ 0 & 0 & 28 \end{pmatrix}$$

Figure: Manipulando matrizes

Álgebra Linear

In [120]:
```
show("Polinômio Car.: ", A.charpoly())
show("Polinômio Min.: ", A.minpoly())
```

Polinômio Car.:$x^3 - 9x^2 + 6x + 28$

Polinômio Min.:$x^3 - 9x^2 + 6x + 28$

In [121]:
```
A.eigenvalues()
```

Out[121]:
```
[-1.378695206755170?, 2.616358832559789?, 7.762336374195381?]
```

In [123]:
```
show(A.LU())
```

$$\left(\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & 1 & 0 \\ \frac{1}{3} & \frac{5}{7} & 1 \end{pmatrix}, \begin{pmatrix} 3 & 1 & 5 \\ 0 & \frac{7}{3} & -\frac{7}{3} \\ 0 & 0 & 4 \end{pmatrix}\right)$$

Figure: Polinômios, autovalores e decomposições

Álgebra - Cálculo Simbólico

Álgebra - Cálculo Simbólico

In [127]:
```
var('a,b')
show(expand((a+b)^5))
```

$$a^5 + 5\,a^4 b + 10\,a^3 b^2 + 10\,a^2 b^3 + 5\,a b^4 + b^5$$

In [134]:
```
show(factor(a^6 - b^6))
```

$$\left(a^2 + ab + b^2\right)\left(a^2 - ab + b^2\right)(a + b)(a - b)$$

In [139]:
```
show(solve([a^2 + a - 1 == 0],a))
```

$$\left[a = -\frac{1}{2}\sqrt{5} - \frac{1}{2}, a = \frac{1}{2}\sqrt{5} - \frac{1}{2}\right]$$

Figure: Manipulando expressões, resolvendo equações

In [162]:

```
# interseção do círculo unitário com a parábola y = x^2
var('x,y')
sol = solve([
    x^2 + y^2 == 1,
    y == x^2,
],x,y)
show(sol)
```

$$\left[\left[x = -\sqrt{\frac{1}{2}\sqrt{5} - \frac{1}{2}}, y = \frac{1}{2}\sqrt{5} - \frac{1}{2}\right], \left[x = \sqrt{\frac{1}{2}\sqrt{5} - \frac{1}{2}}, y = \frac{1}{2}\sqrt{5} - \frac{1}{2}\right], \left[x = -\sqrt{-\right.\right.$$

$$\left.\left[x = \sqrt{-\frac{1}{2}\sqrt{5} - \frac{1}{2}}, y = -\frac{1}{2}\sqrt{5} - \frac{1}{2}\right]\right]$$

In [180]:

```
f(x) = a*x^2 + b*1/x^2
show(f(x))
show(f(f(f(x))))
show(f(x).subs(a=3,b=2))
```

$$ax^2 + \frac{b}{x^2}$$

$$\left(\left(ax^2 + \frac{b}{x^2}\right)^2 a + \frac{b}{\left(ax^2 + \frac{b}{x^2}\right)^2}\right)^2 a + \frac{b}{\left(\left(ax^2 + \frac{b}{x^2}\right)^2 a + \frac{b}{\left(ax^2+\frac{b}{x^2}\right)^2}\right)^2}$$

$$3\,x^2 + \frac{2}{x^2}$$

Álgebra abstrata

Álgebra Abstrata

In [187]:
```
G = SymmetricGroup(3)
G.is_abelian()
```
Out[187]:
```
False
```

In [193]:
```
G.multiplication_table(names='elements')
```
Out[193]:
```
      *        ()   (2,3)   (1,2) (1,2,3) (1,3,2)   (1,3)
       +------------------------------------------------
     ()|       ()   (2,3)   (1,2) (1,2,3) (1,3,2)   (1,3)
  (2,3)|    (2,3)      () (1,2,3)   (1,2)   (1,3) (1,3,2)
  (1,2)|    (1,2) (1,3,2)      ()   (1,3)   (2,3) (1,2,3)
(1,2,3)|  (1,2,3)   (1,3)   (2,3) (1,3,2)      ()   (1,2)
(1,3,2)|  (1,3,2)   (1,2)   (1,3)      () (1,2,3)   (2,3)
  (1,3)|    (1,3) (1,2,3) (1,3,2)   (2,3)   (1,2)      ()
```

In [194]:
```
g = G("(1,2,3)")
g.inverse()
```
Out[194]:
```
(1,3,2)
```

Figure: Trabalhando com grupos

E muito mais:

- Combinatória, análise numérica, polinômios, grafos, etc
- Tópicos avançados (geometria algébrica, curvas elíticas, formas modulares, etc)
- Interface com outras ferramentas (GP/PARI, GAP, Singular,...)
- LaTeX!
- Todo poder do *Python*!
- O principal...

Teoria dos Números

A rainha da matemática

Figure: Carl F. Gauss

Por quê?

"Associado ao pensamento computacional, ***cumpre salientar a importância dos algoritmos e de seus fluxogramas****, que podem ser objetos de estudo nas aulas de Matemática. Um algoritmo é uma sequência finita de procedimentos que permite resolver um determinado problema. [...] A linguagem algorítmica tem pontos em comum com a linguagem algébrica, sobretudo em relação ao conceito de variável.* ***Outra habilidade relativa à álgebra que mantém estreita relação com o pensamento computacional é a identificação de padrões para se estabelecer generalizações, propriedades e algoritmos.****""*

Brasil (2017). Base nacional comum curricular. Ministério da Educação e Cultura

Computador não é o suficiente!

$$\begin{array}{rl} 31 & \text{é primo} \\ 331 & \text{é primo} \\ 3331 & \text{é primo} \\ 33331 & \text{é primo} \\ 333331 & \text{é primo} \\ 3333331 & \text{é primo} \\ 33333331 & \text{é primo} \\ 333333331 & \text{é composto} \end{array}$$

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

31	é primo
331	é primo
3331	é primo
33331	é primo
333331	é primo
3333331	é primo
33333331	é primo
333333331	é composto

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

31	é primo
331	é primo
3331	é primo
33331	é primo
333331	é primo
3333331	é primo
33333331	é primo
333333331	é composto

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

31	é primo
331	é primo
3331	é primo
33331	é primo
333331	é primo
3333331	é primo
33333331	é primo
333333331	é composto

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

31	é primo
331	é primo
3331	é primo
33331	é primo
333331	é primo
3333331	é primo
33333331	é primo
333333331	é composto

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

31	é primo
331	é primo
3331	é primo
33331	é primo
333331	é primo
3333331	é primo
33333331	é primo
333333331	é composto

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

$$\begin{array}{rl} 31 & \text{é primo} \\ 331 & \text{é primo} \\ 3331 & \text{é primo} \\ 33331 & \text{é primo} \\ 333331 & \text{é primo} \\ 3333331 & \text{é primo} \\ 33333331 & \text{é primo} \\ 333333331 & \text{é composto} \end{array}$$

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

31	é primo
331	é primo
3331	é primo
33331	é primo
333331	é primo
3333331	é primo
33333331	é primo
333333331	é composto

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

31	é primo
331	é primo
3331	é primo
33331	é primo
333331	é primo
3333331	é primo
33333331	é primo
333333331	é composto

$$333333331 = 17 \times 19607843$$

Computador não é o suficiente!

$$
\begin{array}{rl}
31 & \text{é primo} \\
331 & \text{é primo} \\
3331 & \text{é primo} \\
33331 & \text{é primo} \\
333331 & \text{é primo} \\
3333331 & \text{é primo} \\
33333331 & \text{é primo} \\
333333331 & \text{é composto}
\end{array}
$$

$$333333331 = 17 \times 19607843$$

Onde aprender?

- Comput. Math. with SageMath
  - http://sagebook.gforge.inria.fr/english.html
- Sage for undergraduates
  - people.vcu.edu/~clarson/bard-sage-for-undergraduates-2014.pdf
- Grátis (inglês/francês/alemão)

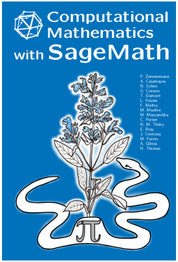


Figure: Comput. Math. with SageMath

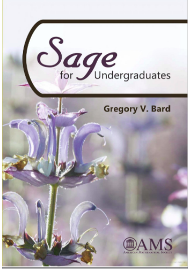


Figure: Sage for undergraduates

Em português

- Elementos de comput. matemática com o Sagemath
  - `https://loja.sbm.org.br/index.php/elementos-de-computac-html`
- Outros: `sagemath.org/library-publications.html#books`

Como usar?

- Baixando e instalando.
  - `www.sagemath.org/download.html`
  - Windows, Mac e Linux
  - Não recomendável (pelo menos inicialmente)
  - Terminal e Jupyter
- Cocalc
  - `cocalc.com/`
  - Colaborativo e interativo (mostro já!)
  - Nuvem
- SageMathCell
  - `sagecell.sagemath.org/`
  - Usaremos esse aqui!
  - Desvantagem: Não salva seu trabalho.

Tipos de dados †

- Números:
  - Inteiros `ZZ`
  - Racionais `QQ`
  - Reais `RR`
  - Complexos `CC`
- Booleanos: Verdadeiro/Falso (True/False)
- Strings (textos!)
- $\rightarrow$ Listas $\leftarrow$
  - Sequências indexadas por $0, 1, 2, \ldots$
  - Não são exatamente conjuntos... mas servem

$$\{x^2 \mid x \in \{1, \cdots, 5\}\}$$

```
[ x^2 for x in [1..5]]
```

- Operações e funções

Finalmente...

Teorema (Divisão euclidiana)

Sejam $a, b \in \mathbb{Z}$ com $b \neq 0$. Exitem $q, r \in \mathbb{Z}$ únicos tais que

$$a = bq + r, r \in \{0, 1, \dots, |b| - 1\}$$

- Método `.quo_rem()` (**Quo**tient e **Rem**ainder — Quociente e resto)
- Uso: `a.quo_rem(b)`
- Retorna o par `(q,r)`
- Cuidado! Para $b < 0$ o comportamento é diferente.

Divisores

- Se o resto for zero dizemos que b *divide* a. A notação é $b \mid a$.
- No sage isso se verifica com o método `b.divides(a)`.
- Vamos criar uma lista com os divisores de a. †
- **Desafio:** Um natural n é *perfeito* se é a soma de seus divisores próprios, e.g.

  $$6 = 1 + 2 + 3 \text{ e } 28 = 1 + 2 + 4 + 7 + 14$$

  Encontre mais um número perfeito (Dica: Existe a função `sum`!)
- **Dever de casa 1**: Um teorema de Euclides/Euler classifica os perfeitos pares. Pesquise-o e encontre os 10 primeiros
- **Dever(?) de casa 2**: Encontre um número perfeito ímpar.

Divisores

- Se o resto for zero dizemos que b *divide* a. A notação é $b \mid a$.
- No sage isso se verifica com o método `b.divides(a)`.
- Vamos criar uma lista com os divisores de a. †
- **Desafio:** Um natural n é *perfeito* se é a soma de seus divisores próprios, e.g.

$$6 = 1 + 2 + 3 \text{ e } 28 = 1 + 2 + 4 + 7 + 14$$

  Encontre mais um número perfeito (Dica: Existe a função `sum`!)
- **Dever de casa 1**: Um teorema de Euclides/Euler classifica os perfeitos pares. Pesquise-o e encontre os 10 primeiros
- **Dever(?) de casa 2**: Encontre um número perfeito ímpar.

Divisores

- Se o resto for zero dizemos que b *divide* a. A notação é $b \mid a$.
- No sage isso se verifica com o método `b.divides(a)`.
- Vamos criar uma lista com os divisores de a. †
- **Desafio:** Um natural n é *perfeito* se é a soma de seus divisores próprios, e.g.

  $$6 = 1 + 2 + 3 \text{ e } 28 = 1 + 2 + 4 + 7 + 14$$

  Encontre mais um número perfeito (Dica: Existe a função `sum`!)
- **Dever de casa 1**: Um teorema de Euclides/Euler classifica os perfeitos pares. Pesquise-o e encontre os 10 primeiros
- **Dever(?) de casa 2**: Encontre um número perfeito ímpar.

Primos

- Um natural $p > 1$ é primo seus divisores positivos são 1 e p.
- Como decidir se um dado n é primo?
- Testar se n é divisível por algum inteiro > 1 menor que ele. †
- **Dever de casa**: Otimize o algoritmo acima.
- No sage: `is_prime()`
- Pseudoprimos.
- Outras funções envolvendo primos. †

Primos

- Um natural $p > 1$ é primo seus divisores positivos são 1 e p.
- Como decidir se um dado n é primo?
- Testar se n é divisível por algum inteiro > 1 menor que ele. †
- **Dever de casa**: Otimize o algoritmo acima.
- No sage: `is_prime()`
- Pseudoprimos.
- Outras funções envolvendo primos. †

Primos

- Um natural $p > 1$ é primo seus divisores positivos são 1 e p.
- Como decidir se um dado n é primo?
- Testar se n é divisível por algum inteiro > 1 menor que ele. †
- **Dever de casa**: Otimize o algoritmo acima.
- No sage: `is_prime()`
- Pseudoprimos.
- Outras funções envolvendo primos. †

MDC

- Maior divisor comum
- `max([k for k in divisors(a) if k in divisors(b)])`
- Algoritmo de Euclides: **Lema:** $\mathrm{mdc}(a, b) = \mathrm{mdc}(b, a - kb)$
- r resto da divisão de a por $b \Rightarrow \mathrm{mdc}(a, b) = \mathrm{mdc}(b, r)$
- Divisões euclidianas sucessivas: †

$$\begin{array}{rcrcl}
 & & a & & b \\
1001 & = & 9 \times 109 & + & 20 \\
109 & = & 5 \times 20 & + & 9 \\
20 & = & 2 \times 9 & + & 2 \\
9 & = & 4 \times 2 & + & 1 \\
2 & = & 2 \times 1 & + & 0.
\end{array}
\qquad
\begin{array}{l}
\\
\mathrm{mdc}(1001, 109) = \mathrm{mdc}(109, 20) \\
\mathrm{mdc}(109, 20) = \mathrm{mdc}(20, 9) \\
\mathrm{mdc}(20, 9) = \mathrm{mdc}(9, 2) \\
\mathrm{mdc}(9, 2) = \mathrm{mdc}(2, 1) \\
\mathrm{mdc}(2, 1) = \mathrm{mdc}(1, 0) = 1
\end{array}$$

Algoritmo de Euclides Estendido

- **T. Bézout:** Existem $x, y \in \mathbb{Z}$ tais que $ax + by = \text{mdc}(a, b)$.
- As divisões sucessivas do Alg. de Euclides fornecem x e y.
- † Uma solução elegante: Seja $M = (m_{ij}) \in \text{M}_{3\times 2}(\mathbb{Z})$, dada por

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ a & b \end{pmatrix}.$$

Se $m_{32} \neq 0$, tome $q = \lfloor m_{31}/m_{32} \rfloor$ e substitua

$$M \longleftarrow M \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix}.$$

Enquanto $m_{32} \neq 0$ repita esse processo. Quando $m_{32} = 0$ teremos que $m_{31} = \text{mdc}(a, b)$ e $x = m_{11}$ e $y = m_{21}$ satisfazem $ax + by = \text{mdc}(a, b)$

Algoritmo de Euclides Estendido

- **T. Bézout:** Existem $x, y \in \mathbb{Z}$ tais que $ax + by = \mathrm{mdc}(a, b)$.
- As divisões sucessivas do Alg. de Euclides fornecem x e y.
- † Uma solução elegante: Seja $M = (m_{ij}) \in \mathrm{M}_{3\times 2}(\mathbb{Z})$, dada por

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ a & b \end{pmatrix}.$$

Se $m_{32} \neq 0$, tome $q = \lfloor m_{31}/m_{32} \rfloor$ e substitua

$$M \longleftarrow M \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix}.$$

Enquanto $m_{32} \neq 0$ repita esse processo. Quando $m_{32} = 0$ teremos que $m_{31} = \mathrm{mdc}(a, b)$ e $x = m_{11}$ e $y = m_{21}$ satisfazem $ax + by = \mathrm{mdc}(a, b)$

Algoritmo de Euclides Estendido

- **T. Bézout:** Existem $x, y \in \mathbb{Z}$ tais que $ax + by = \mathrm{mdc}(a, b)$.
- As divisões sucessivas do Alg. de Euclides fornecem x e y.
- † Uma solução elegante: Seja $M = (m_{ij}) \in \mathrm{M}_{3\times 2}(\mathbb{Z})$, dada por

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ a & b \end{pmatrix}.$$

Se $m_{32} \neq 0$, tome $q = \lfloor m_{31}/m_{32} \rfloor$ e substitua

$$M \longleftarrow M \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix}.$$

Enquanto $m_{32} \neq 0$ repita esse processo. Quando $m_{32} = 0$ teremos que $m_{31} = \mathrm{mdc}(a, b)$ e $x = m_{11}$ e $y = m_{21}$ satisfazem $ax + by = \mathrm{mdc}(a, b)$

Fatoração Única

Teorema (Fundamental da Aritmética)

Todo inteiro $n > 1$ pode ser escrito de forma única como produto de primos

$$n = p_1 p_2 \cdots p_k,$$

com $p_1 \leq p_2 \leq \cdots \leq p_k$.

- Forma mais natural: Testar todos os primos! †
- No sage: `factor`
- Aplicação Importante: Criptografia!

Mini-Projeto: 1º dia

$$\frac{\#\{n \in \mathbb{N} \mid n \leq N \text{ e } p \mid n\}}{\#\{n \in \mathbb{N} \mid n \leq N\}} \to \frac{1}{p}, \text{ quando } N \to \infty$$

- Prob. de um inteiro ser divisível por p é $1/p$.
- Prob. de dois inteiros serem divisíveis por p é $1/p^2$.
- Prob. de dois inteiros não serem simultaneamente divisíveis por p é $1 - \frac{1}{p^2}$
- Prob. de dois inteiros não serem simultaneamente divisíveis por qualquer primo p (i.e. serem coprimos!)

$$\prod_{p \in P} \left(1 - \frac{1}{p^2}\right) = \cdots = \frac{6}{\pi^2}$$

- Tarefa: Escreva um código que dê aproximações para π tomando pares de inteiros arbitrários e verificando se são coprimos.

Ideia:

- Escolha um limite K e uma quantidade de repetições N.
- Crie um contador C para guardar os casos coprimos.
- Para $n = 1, \ldots, N$
  - Escolha a, b aleatórios em $\{1, \ldots, K\}$
  - Se $\mathrm{mdc}(a, b) = 1$, aumente o valor no contador $(C \leftarrow C + 1)$
- Para N e K grandes devemos ter $\alpha := C/N \approx 6/\pi^2$.
- Isole o π na relação acima.
- Exiba uma aproximação para π usando α.
- Varie K e N e veja o efeito na aproximação.

Congruências

- Pela divisão euclidiana, na divisão por $m > 0$ há m restos possíveis: $0, 1, \dots, m-1$,
- **Def.:** $a, b \in \mathbb{Z}$ são congruentes módulo m se deixam o mesmo resto na divisão por m.
- Notação: $a \equiv b \pmod{m}$.
- Além do `.quo_rem()` há o operador `%`. †
- Melhor ainda: `mod`.

- Resumindo algumas aulas em poucos itens:
  - A congruência respeita a aritmética.
  - Na verdade, $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$ com a relação de congruência é um anel $\rightarrow$ Resultado do `mod`
- É preferível trabalhar em $\mathbb{Z}_m$.
  - Manipulações mais complexas.
  - Contas mais rápidas †
- Conexão com álgebra abstrata!
- Invertíveis.
- Função φ de Euler.

Mini-Projeto 2º dia: Criptografia

A principal ferramenta matemática utilizada será o Teorema de Euler.

Theorem (Teorema de Euler)

Sejam $m \in \mathbb{Z}$ e $n > 1$ natural com m e n coprimos. Então

$$m^{\varphi(n)} \equiv 1 \pmod{n},$$

onde φ é a função de Euler, definida por

$$\varphi(n) = \#\{1 \leq a \leq n \mid \mathrm{mdc}(a, n) = 1\}$$

- Recorde que em primos p, $\varphi(p) = p - 1$.
- Além disso, φ é multiplicativa, isto é, portanto se p e q são primos distintos então

$$\varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$$

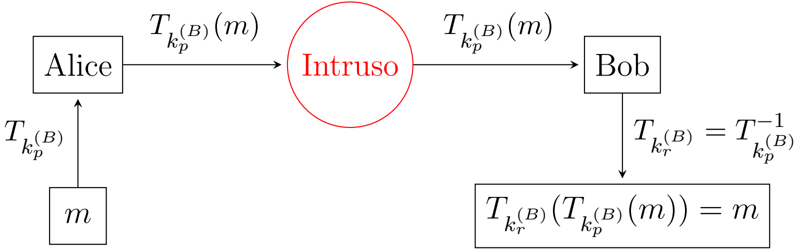
Criptografia assimétrica

Ideia:

- Sistema simétrico:
  - Criptografar uma mensagem seria como colocar a mensagem numa caixa e fechar uma fechadura.
  - Quem tem a chave da fechadura consegue abrir a caixa.
- Sistema assimétrico:
  - Fechadura especial com duas chaves: k_p e k_r.
  - Se a caixa é fechada com k_p, apenas é aberta com k_r.
  - Se a caixa é fechada com k_r, apenas é aberta com k_p.
  - **Cada pessoa** tem um par (k_p, k_r).
    - k_p é chamada de chave de pública.
    - k_r é chamada de chave de privada.
  - Formalmente $T_{k_p}^{-1} = T_{k_r}$, ou seja

$$T_{k_r}(T_{k_p}(m)) = m \text{ e } T_{k_p}(T_{k_r}(m)) = m$$

- Todo mundo deixa a sua chave pública disponível.
- Por exemplo, se Alice deseja enviar uma mensagem para Bob, ela usa a chave publica de Bob $k_p^{(B)}$ (disponível para todos).



- A mensagem cifrada $T_{k_p^{(B)}}(m)$ só pode ser decifrada com a chave privada $T_{k_r^{(B)}}$ de Bob (que só Bob tem)
- Vantagens:
  - Não é necessário trocar chaves.
  - Assinatura digital.

O Algorítmo RSA (Encontrando um par de chaves)

-
  - Escolha primos p e q distintos
  - Calcule $n = pq$
  - Calcule $\varphi(n) = (p-1)(q-1)$
  - Escolha $1 < e < \varphi(n)$ tal que $\text{mdc}(e, \varphi(n)) = 1$.
  - Calcule $d \equiv e^{-1} \pmod{\varphi(n)}$.
- A chave publica consiste é o par (n, e).
- A chave privada k_r é o número d.
  - Exceto pela chave publica, i.e., os números n e e, todos os outros são guardados em segredo.
  - Para encontrar d a partir de e, é necessário saber $\varphi(n)$
  - Saber $\varphi(n)$ se resume a saber a fatoração de n.
  - Se p e q forem primos muito grandes, encontrá-los a partir de n é uma tarefa muito dífícil.

A cifragem

Nesse algorítmo, os textos são transformados em números inteiros cod, com $0 \leq \text{cod} < n$.

- Suponha que, com o processo descrito acima, Bob tenha gerado sua chave pública (n, e_B) e sua chave privada d_B.
- Alice, conhecendo e_B, cifra a mensagem cod via

$$\text{cod}_{\text{cifrado}} = T_{k_p^{(B)}}(\text{cod}) := \text{cod}^{e_B} \pmod{n}$$

- Alice envia c para Bob
- Bob decifra a mensagem c via

$$T_{k_r^{(B)}}(\text{cod}_{\text{cifrado}}) := \text{cod}_{\text{cifrado}}^{d_B} \equiv \text{cod}^{e_B d_B} \equiv \text{cod} \pmod{n}$$

Pois $e_B d_B \equiv 1 \pmod{\varphi(n)}$, logo $e_B d_B = q\varphi(n) + 1$
Assim, pelo Teorema de Euler, $\text{cod}^{\varphi(n)} \equiv 1 \pmod{n}$.

$$\text{cod}^{e_B d_B} = \text{cod}^{q\varphi(n)+1} = (\text{cod}^{\varphi(n)})^q \text{cod} \equiv \text{cod} \pmod{n}$$

Observações:

- Os cálculos $cod^{e_B} \pmod{n}$ e $c^{d_B} \pmod{n}$ são eficientes.
- Calcular $\phi(n)$ sem saber a fatoração de n, isto é p e q, é difícil!
- Fatorar $n = pq$ é muito difícil.

$$n = RSA - 240 = 124620366781718784065835044608106590434820374651678805754818788883289666801188210855036039570272508747509864768438458621054865537970253930571891217684318286362846948405301614416430468066875699415246993185704183030512549594371372159029236099$$

$$p = 509435952285839914555051023580843714132648382024111473186660296521821206469746700620316443478873837606252372049619334517$$

$$q = 244624208838318150567813139024002896653802092578931401452041221336558477095178155258218897735030590669041302045908071447$$

  - Fatorado em Novembro de 2019, usando supercomputadores.
  - $\sim$ 900 cores-anos

Observações:

- Os cálculos $\mathrm{cod}^{e_B} \pmod{n}$ e $c^{d_B} \pmod{n}$ são eficientes.
- Calcular $\phi(n)$ sem saber a fatoração de n, isto é p e q, é difícil!
- Fatorar $n = pq$ é muito difícil.

$$n = RSA - 240 = 124620366781718784065835044608106590434820374651678805754818788883289666801188210855036039570272508747509864768438458621054865537970253930571891217684318286362846948405301614416430468066875699415246993185704183030512549594371372159029236099$$

$$p = 509435952285839914555051023580843714132648382024111473186660296521821206469746700620316443478873837606252372049619334517$$

$$q = 244624208838318150567813139024002896653802092578931401452041221336558477095178155258218897735030590669041302045908071447$$

  - Fatorado em Novembro de 2019, usando supercomputadores.
  - $\sim$ 900 cores-anos

Sistemas de congruências

Teorema

Se m_1 e m_2 são coprimos

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \end{cases}$$

tem solução única módulo $m_1 m_2$.

- Solução: $x = a_1 m_2 m_1' + a_2 m_1 m_2'$ onde m_1' é o inverso de m_2 módulo m_1 e m_2' é o inverso de m_1 módulo m_2.
- Atenção: Não podemos misturar os módulos
  - Usar `inverse_mod`
- No sage: `crt`

Ideias de projetos

- Equações diofantinas
  - Força bruta / otimizações
  - Soluções parametrizáveis (ternos pitagóricos, Pell, etc.)
  - Inexistência de soluções via redução módulo m.
- Criptografia: RSA, Diffie-Hellman, $M_n(\mathbb{Z}_m)$, etc
- Códigos

Outras Referências / Links

- Livros online
  - Teoria dos Números `http://math.gordon.edu/ntic/`
  - Álgebra abstrata
    `http://abstract.ups.edu/aata/aata.html`
- `https://www.sagemath.org/library.html`
- `https://www.sagemath.org/library-publications.html`
- Euler project (`https://projecteuler.net/`)