Breves Considerações Sobre a Utilidade dos Sistemas de Computação Algébrica no Ensino Médio

Lenimar Nunes de Andrade
UFPB – João Pessoa

27 de outubro de 2013

1 Introdução

A Computação, em suas várias formas, tem produzido significativas transformações sociais, econômicas e culturais no mundo contemporâneo. E, de fato, hoje em dia, é impossível imaginar nosso cotidiano sem uma efetiva utilização do computador.

Tradicionalmente, o vocábulo *computar* tem sido utilizado com o significado de *"efetuar cálculos com números"*. A Computação Numérica envolve não só as operações aritméticas básicas de adição, subtração, multiplicação, divisão e radiciação, mas também atividades mais sofisticadas, a exemplo do cálculo das raízes de um polinômio. Nesse tipo de computação, os dados iniciais e os resultados finais são números.

Apesar de atender às exigências de um usuário nas mais diversas situações, os cálculos puramente numéricos, em geral, não são exatos, porque eles acabam por embutir algum tipo de erro nos cálculos. Por exemplo, a simples divisão de 1 por 3 em notação decimal é aproximadamente igual a $0,33333333$. Não importam quantas casas decimais sejam utilizadas, esse valor será sempre aproximado: ao ser multiplicado por 3, o produto dará $0,99999999$. Se fosse um cálculo exato, a resposta deveria ser igual a 1. Além disso, esse pequeno erro inicial da ordem de $0,00000001$ se for agregado a outros cálculos também aproximados, tende a se propagar e aumentar.

Devido às inconveniências da Computação Numérica, como a falta de exatidão nos resultados apresentada acima, é que se tem estudado, há seis décadas, uma outra variedade de computação denominada Computação Simbólica ou Computação Algébrica. Nessa modalidade de computação, os objetos matemáticos podem ser representados por símbolos, não obrigatoriamente numéricos. Esses símbolos podem representar números inteiros, racionais, irracionais, imaginários, ou, ainda, polinômios, funções, matrizes, sequências, sistemas de equações e até mesmo estruturas mais abstratas, como grupos e anéis.

O principal objetivo da Computação Algébrica é obter resultados exatos baseados nas regras usuais da Álgebra, fórmulas consideradas "fechadas". Por exemplo, um programa de Computação Algébrica efetua cálculos com a raiz quadrada de dois sem a necessidade de representá-la em forma decimal aproximada. Em vez de usar aproximações numéricas, o programa conhece as regras e propriedades algébricas dos objetos envolvidos. Ele sabe que o objeto *"raiz quadrada de dois"* é positivo e que se for elevado ao quadrado, indicará como resultado dois. Isso basta para efetuar cálculos em inúmeras situações. Ao se calcular $\sqrt{2}\sqrt{3}$ usando uma calculadora ou um programa de Computação Numérica, o resultado será $1,4142135 \times 1,7320508 = 2,4494897$. No entanto, qualquer programa de computação algébrica mostrará resultado como sendo $\sqrt{2}\sqrt{3} = \sqrt{6}$, algo bastante próximo do que se faz em sala de aula. Outro exemplo: um programa de Computação Algébrica também percebe que $(a+b)^2 = a^2 + 2ab + b^2$ ou que $(a+b)(a-b) = a^2 - b^2$, sem ser necessário atribuir quaisquer valores numéricos para a e b.

2 Sistemas de Computação Algébrica

Nas últimas décadas, houve considerável progresso na teoria envolvendo algoritmos simbólicos ou algébricos. O conjunto de programas de computador relacionados com esse tipo de teoria passou a se chamar *Sistema de Computação Algébrica* ou *Sistema de Manipulação Simbólica,* dentre outras denominações.

Os Sistemas de Computação Algébrica (SCA) podem ser utilizados em inúmeras áreas da ciência e da tecnologia, e podem ser divididos em duas categorias: sistemas de uso específico e sistemas de uso geral. O que esses sistemas têm em comum é uma grande capacidade de efetuar rapidamente cálculos analíticos que podem ser bastante trabalhosos.

Os sistemas de uso específico foram desenvolvidos para resolver problemas em áreas específicas da Física ou da Matemática. São alguns exemplos desses sistemas: Sheep, Camal, GAP, Magma, Singular e Macaulay.

Os sistemas de uso geral possuem não só recursos algébricos, podendo também incorporar recursos numéricos ou gráficos, além de serem verdadeiras linguagens de programação para cálculos analíticos. Alguns são comercializados como o Maple e o Mathematica, outros são livres e gratuitos.

Alguns exemplos desses sistemas livres são:

- **Maxima** – antigamente, na década de 70, era um programa comercial e chamava-se Macsyma. Hoje em dia, é um SCA gratuito, com muitos recursos, muito bem documentado, e com página própria na Internet: a **maxima.sourceforge.net**.

- **Reduce** – surgido no final da década de 60, é um dos mais antigos SCA, mas está em pleno funcionamento hoje em dia. Pode ser copiado a partir do URL **reduce-algebra.com**.

- **MuPAD** – um SCA gratuito na sua versão Light 2.5.2 de 2006. A partir de 2008, foi incorporado ao programa comercial Matlab.

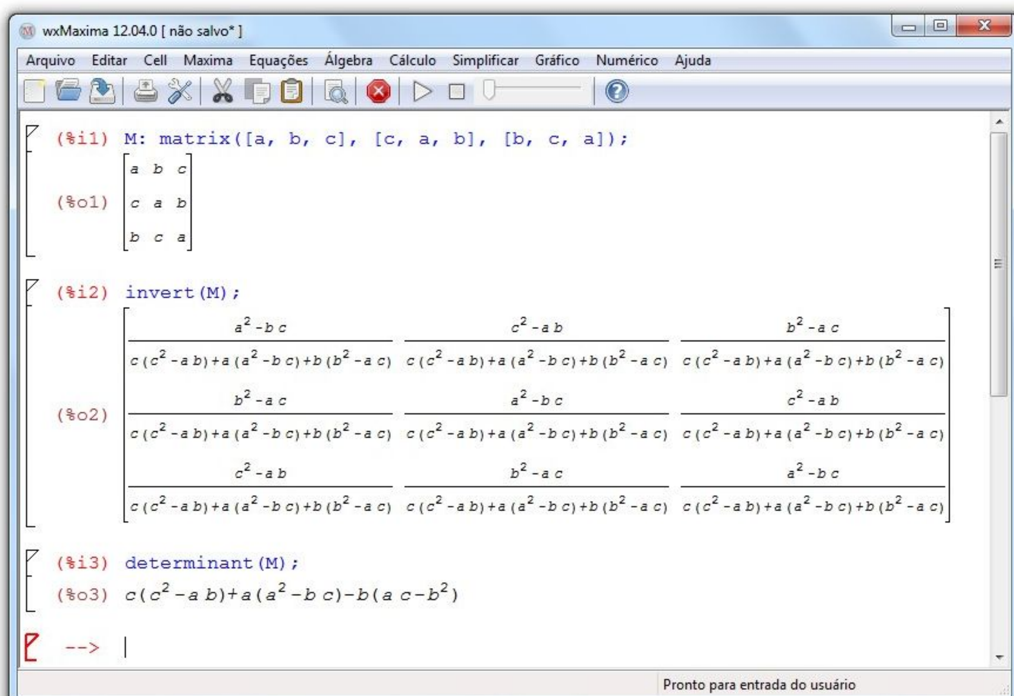
- **Yacas** – "Yet Another Computer Algebra System", um SCA gratuito e bastante completo, que pode ser copiado de **yacas.sourceforge.net**.

- **Axiom** – surgiu em 1971 e, antigamente, chamava-se Scratchpad. É um SCA gratuito de uso geral e foi originalmente desenvolvido pela IBM. Disponível para diversos sistemas operacionais a partir da página **axiom-developer.org**

- **GeoGebra** – É um conhecido programa de Geometria Dinâmica que já foi apresentado várias vezes na RPM. Em suas versões mais recentes, como a 4.2.55.0, incorporou uma *"Janela CAS"* na qual podem ser executados diversos comandos básicos em português tais como `Simplificar(128/200)`, `Fatorar(2013)` ou `Resolver(x^3-8 = 0)`. Pode ser copiado gratuitamente de **www.geogebra.org**

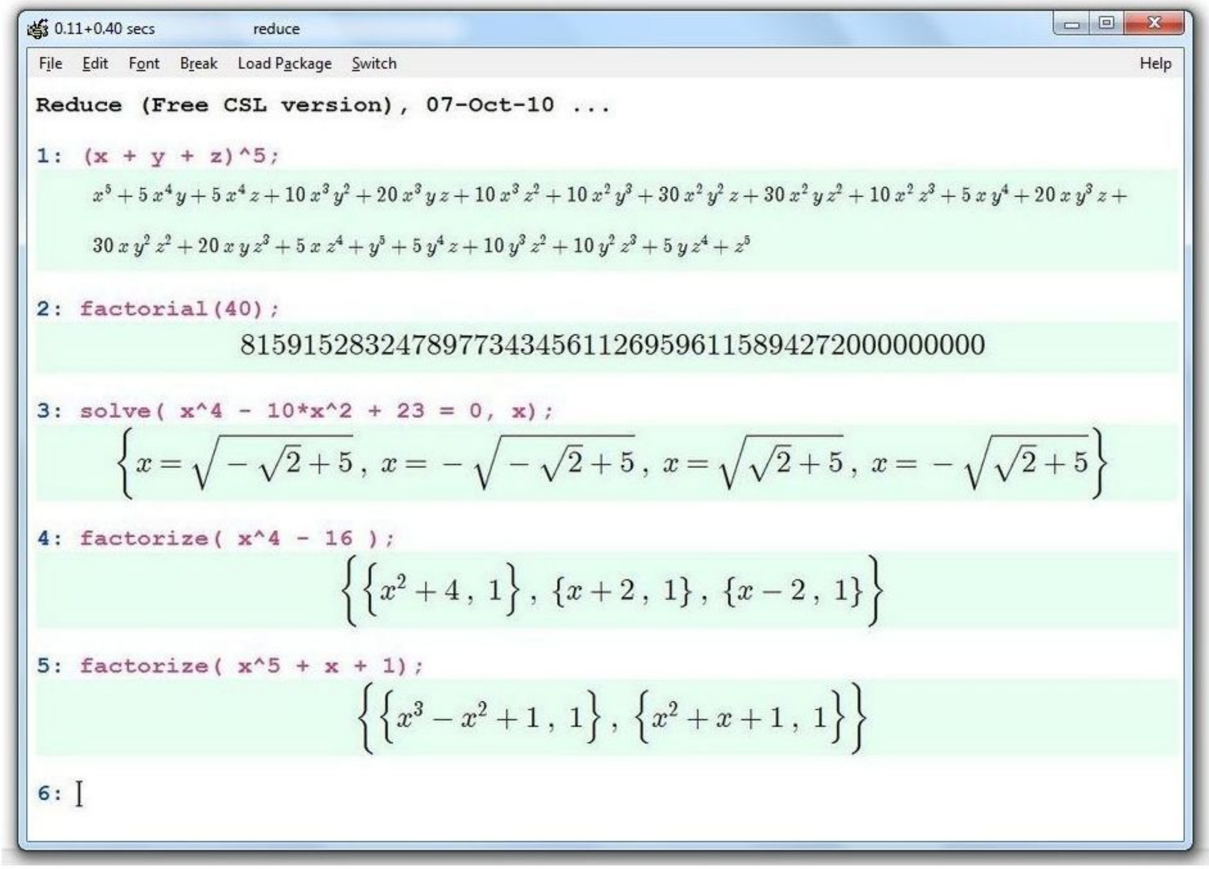
Muitos desses SCA possuem documentação disponibilizada na Internet em vários idiomas, e alguns deles têm até manuais em português. Em geral, possuem versões para vários sistemas operacionais tais como Windows, Linux e Mac-OS.

3 Exemplos

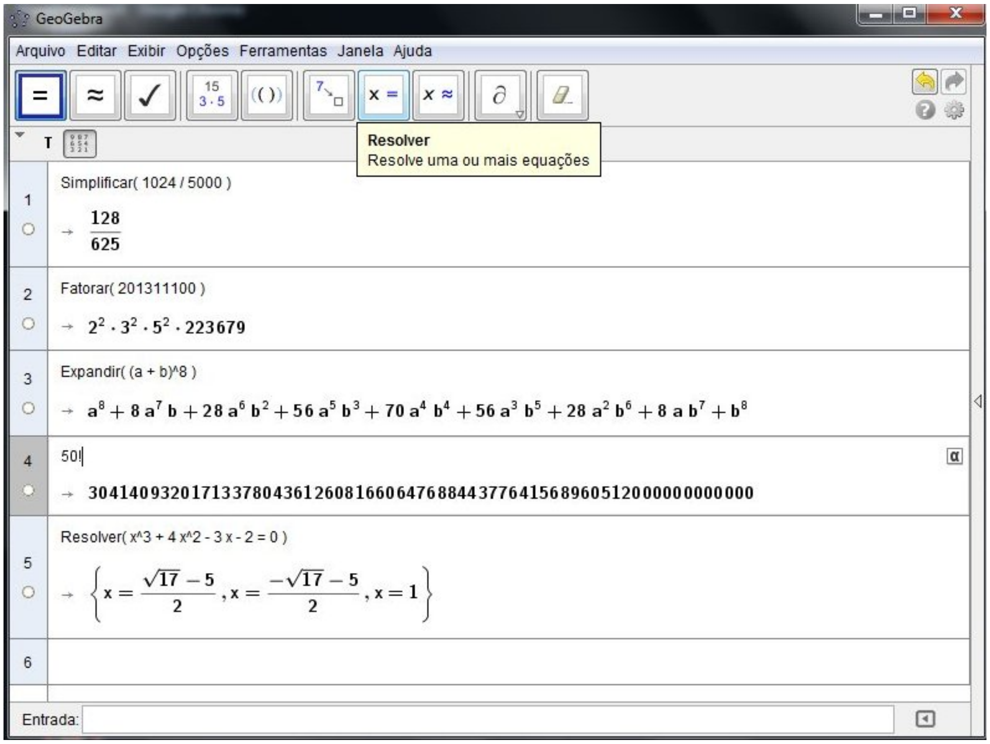


Na figura acima, é mostrada uma tela com um exemplo de utilização do Maxima, um SCA gratuito com inúmeras funções. Nessa tela são mostradas uma matriz $M = \begin{bmatrix} a & b & c \\ c & a & b \\ b & c & a \end{bmatrix}$, além do cálculo da sua inversa M^{-1} e do seu determinante $\det(M) = c(c^2 - ab) + a(a^2 - bc) - b(ac - b^2)$. Note que os valores envolvidos nos cálculos não são numéricos, mas o programa não se intimida com isso e realiza os cálculos na maior rapidez.

Na segunda figura, é mostrado um exemplo de utilização do Reduce, outro SCA gratuito e com bastantes recursos. É mostrado todo o desenvolvimento da potência $(x+y+z)^5$, o cálculo do fatorial de 40, a solução da equação biquadrada $x^4 - 10x^2 + 23 = 0$ e a fatoração dos polinômios $x^4 - 16$ e $x^5 + x + 1$. A apresentação dos resultados é toda em formato usual.



A seguir, exemplificamos alguns comandos de Computação Algébrica do GeoGebra.



Praticamente todos esses programas podem ser usados como uma linguagem de programação. Apesar desse tipo de recurso exigir um pouco mais de estudo para sua utilização, alguns exemplos simples podem ser executados. Por exemplo, no Maxima, um comando como

```
for n from 1 thru 100 do print(n, float(sqrt(n)));
```

mostra uma listagem com n e sua respectiva raiz quadrada para todo n de 1 a 100, enquanto que

```
for k from 100 thru 200 do if primep(k) then print(k, "é primo");
```

lista os inteiros primos de 100 a 200.

4 Um assistente matemático

Um programa de Computação Algébrica pode ser usado como um assistente matemático, simplificando expressões algébricas, encontrando raízes de equações, fatorando polinômios, efetuando operações com matrizes, construindo gráficos de funções, além de ser útil na resolução de problemas de Geometria Analítica e de outros assuntos.

Esses programas são capazes de obter respostas que podem ser comparadas com aquelas de que o usuário já dispõe. Podem, também, efetuarem a parte mais trabalhosa de cálculos, deixando para o usuário apenas a análise e conclusão a respeito dos resultados. Por tudo isso, esse tipo de programa pode ser considerado como uma ferramenta de ajuda valiosa, tornando-se um aliado em todo o processo de ensino-aprendizagem.

Apresentamos aqui duas atividades que podem ser realizadas em sala de aula. São atividades conceitualmente simples, mas trabalhosas sob o ponto de vista da execução dos cálculos de modo que justificam a utilização de um recurso computacional para sua realização. Escolhemos um tema que enfatiza a capacidade desse tipo de programa em manipular expressões algébricas: polinômios e equações polinomiais. Utilizamos o Maxima nos exemplos, mas poderia ter sido usado outro SCA com as devidas adaptações nos comandos.

Atividade 1

Como primeira atividade, vamos calcular o máximo divisor comum de dois polinômios usando o conhecido algoritmo da divisão euclidiana.

- Inicialmente, fornecemos dois polinômios p e q na variável x:
  `p:  x^7 + 5*x^6 - 9*x^5 + 27*x^4 - 56*x^3 - 16*x - 112`
  `q:  x^6 + x^5 - 4*x^4 - 2*x^3 - 3*x^2 - 15*x - 10`
- Usamos o programa para dividir p por q, digitando `r: divide(p, q);` cuja resposta é $[x+4,\ -9x^5+45x^4-45x^3+27x^2+54x-72]$. Esse resultado é da forma $[r_1, r_2]$, onde r_1 é o quociente da divisão e r_2 é o resto.
- A seguir, dividimos q pelo resto $r[2]$: `s: divide(q, r[2]);`
- Depois, dividimos $r[2]$ pelo resto da divisão anterior $s[2]$:
  `t: divide(r[2], s[2]);` cuja resposta é $[-\frac{21x-76}{49},\ \frac{440x^3+440x+880}{49}]$
- Dividimos $s[2]$ pelo resto da divisão anterior $t[2]$: `w: divide(s[2], t[2]);` que o programa responde $[\frac{1029x-1421}{440},\ 0]$.

Como o resto dessa divisão é igual a zero, encerramos o processo de sucessivas divisões e concluímos que $\text{mdc}(p, q)$ é o resto $t[2]$ da divisão de $r[2]$ por $s[2]$, dividido pelo coeficiente do termo de maior grau, ou seja, que $\text{mdc}(p, q) = \frac{t[2]}{\frac{440}{49}} = x^3 + x + 2$.

Podemos também aproveitar a capacidade do programa em resolver equações polinomiais para comprovar que as raízes do máximo divisor comum de p e q é a interseção dos conjuntos das raízes de p com as raízes de q. Para verificar isso, basta digitar os comandos a seguir:

- `mdc:  gcd(p, q);`
- `solve(mdc = 0, x);`
- `solve(p = 0, x);`
- `solve(q = 0, x);`

Atividade 2

Como segunda atividade, vamos determinar todas as possíveis raízes racionais de uma equação polinomial de coeficientes inteiros $f(x) = 0$ e, depois, testar uma por uma para verificar qual delas é raiz da equação.

- Definimos $f(x)$ com um comando `f(x) := 21*x^3-482*x^2-1089*x+350;`
- Calculamos uma lista q dos divisores do coeficiente do termo de maior grau `q: listify(divisors(21))` e o programa responde $[1, 3, 7, 21]$.
- Calculamos os divisores do termo constante `p: listify(divisors(350))` que o programa responde $[1, 2, 5, 7, 10, 14, 25, 35, 50, 70, 175, 350]$.
- ```
  for j from 1 thru length(p) do
  for k from 1 thru length(q) do
  (r:  p[j]/q[k],
  if (f(r)=0) then print(r,"é raiz") else print(r,"não é raiz"),
  if (f(-r)=0) then print(-r,"é raiz") else print(-r,"não é raiz"));
  ```

Nesse caso, para cada elemento da lista de divisores p e cada elemento de q, é testado se $r = \frac{p_j}{q_k}$ ou $r = -\frac{p_j}{q_k}$ é uma raiz da equação. Depois que a verificação é executada, são mostradas mensagens como "`-1 não é raiz`" ou "`2/7 é raiz`".

É possível a resolução direta da equação com um comando `solve(f(x)=0)` para comparar com as respostas mostradas anteriormente.

Referências

[1] SymbolicNet, *Symbolic Mathematical Computation Information Center*, disponível na Internet em *www.symbolicnet.org*.

[2] Galván, J. R. R., *Maxima con wxMaxima: software libre en el aula de matemáticas*, Oficina de Software Libre de la Universidad de Cádiz.
  ```