Introdução à manipulação de fórmulas

por F. Teixeira de Queiroz
(Instituto Gulbenkian de Ciência)

Introdução. Aparecidos com o objectivo de tratar problemas numéricos, os ordenadores, têm pouco a pouco evoluído no sentido de efectuarem o tratamento de informação não numérica. Tal aspecto não pode ser ignorado pelo matemático, tanto mais que algumas das aplicações que daí resultam são de primordial importância para ele. Apontarei apenas, entre elas, os métodos automáticos de demonstração, o tratamento de problemas ligados à teoria dos grupos, à teoria dos gráficos e, finalmente, a manipulação de fórmulas. Procurarei nesta nota referir-me a esta última aplicação.

Por manipulação de fórmulas entender-se-á o tratamento analítico de expressões matemáticas, envolvendo números ou variáveis, sem que seja tomado em consideração qualquer conjunto de valores particulares que estas últimas possam tomar. Assim, faz parte do objecto da manipulação de fórmulas, as operações sobre polinómios, sobre expressões trigonométricas, a derivação de funções, a integração formal, o desenvolvimento em série de funções, a obtenção de equações diferenciais, etc.

Um matemático habituado a tratar problemas numéricos por meio de ordenador, encontrará ao abordar este assunto, uma panorâmica completamente diferente. Enquanto no tratamento de problemas numéricos, existe o domínio quase exclusivo de três linguagens: ALGOL, FORTRAN, BASIC, existe uma grande diversidade de linguagens destinadas à manipulação de fórmulas: ALTRAN, FORMULA ALGOL, FORMAC, MATHLAB, SIMBAL, REDUCE, etc. Nenhuma delas se impõe verdadeiramente às demais.

Há várias razões para uma tão grande multiplicidade de linguagens. Primeiramente, a estrutura dos elementos sobre que opera um programa destinado a manipular fórmulas é muito mais complexa do que qualquer das estruturas que aparecem em problemas numéricos. Em segundo lugar, como os programas manipuladores são normalmente redigidos em linguagens evoluídas (ALGOL, FORTRAN, LISP, etc.), eles serão fortemente marcados pelas possibilidades latentes destas. Finalmente, importa salientar que a construção de programas destinados à manipulação de fórmulas está longe de estar estabilizada. Os objectivos dos primeiros programas são muito menos ambiciosos que os de programação mais recente. Além disso, como alguns desses programas tiveram origem na necessidade de tratar uma família de problemas concretos, resulta daí que a natureza desses problemas marcou grandemente a forma e as possibilidades dessas linguagens. Assim, enquanto programas tais como PM e Alpak se limitavam quase exclusivamente a operar com polinómios com qualquer número de variáveis, as linguagens de construção mais recente admitem a possibilidade de trabalhar com funções racionais e, mesmo, com algumas transcendentes.

Haverá ainda a salientar um outro elemento que contribui para a diversidade das linguagens: uma característica comum a todos os programas manipuladores de fórmulas é o grande dispêndio de memória provocado pelos cálculos intermédios. É esse um dos elementos a ter sempre presente na elaboração dum programa. Como consequência disso resulta que as possibilidades a introduzir numa linguagem dependem muito da capacidade de memória do ordenador a que se destinam. Ordenadores mais rápidos e com maior capacidade de memória terão certamente linguagens com maior versatilidade.

Elementos dum manipulador de fórmulas. De entre as possibilidades que é usual encontrar-se num moderno manipulador de fórmulas destacamos a existência duma aritmética exacta, duma aritmética complexa, e dum conjunto de rotinas destinadas a operar sobre uma grande variedade de funções algébricas e transcendentes, entre as quais se inclue uma destinada a calcular derivadas. Quanto à integração, que nós saibamos, só o sistema MATH LAB dispõe duma rotina que a permite efectuar com suficiente generalidade.

Examinemos algumas das características que referimos acima mais em pormenor:

A existência de aritmética exacta vem prolongar as possibilidades numéricas usualmente disponíveis em ordenadores. O leitor, habituado a utilizar ordenadores na resolução de problemas numéricos, lida usualmente com dois tipos de constantes: inteiras e reais. Estas últimas são representadas com notação de vírgula fluctuante. Isto é: os números são representados com a forma $m \times b^e$ em que m, a mantissa, é formada pelos algarismos mais significativos e onde e representa um expoente inteiro. Quanto às constantes inteiras, elas não podem exceder um determinado valor que depende do ordenador utilizado. Tais tipos de notação não se tornam convenientes para a manipulação de fórmulas, pois, não só aparecem com frequência números inteiros que excedem o valor limite, como também, em certos processos, torna-se necessário conhecer com exactidão os valores racionais. Em tais casos não será suficiente representar, por exemplo, $1/3$ com a forma 0.3333333333. A fim de evitar isso, os programas manipuladores têm dispositivos que permitem tratar os números fraccionários como se estes fossem fórmulas (quocientes de inteiros) e os números inteiros libertos de qualquer limite. Evidentemente, para que um tal programa seja eficiente torna-se necessário dotá-lo dum dispositivo que torne qualquer fracção irreductível.

A possibilidade de operar quer com polinómios quer com funções racionais constitue um dos elementos centrais dum programa destinado a manipular fórmulas. É relativamente a um tal elemento que os programas diferem mais uns dos outros. Se o programa actuar duma forma demasiado rígida, representará um polinómio com uma forma padrão o que acarretará, por vezes, sérios inconvenientes. Assim, se um programa representar um polinómio ordenando-o segundo as potências crescentes duma das suas variáveis, tomando como coeficientes polinómios das restantes variáveis (ordenados da mesma maneira), tratará duma forma altamente inconveniente o polinómio $(x+y+z+t)^{100}$. Da mesma forma, funções trigonométricas poderão ser substituídas por outras, equivalentes, mas menos simples. Assim, possìvelmente a função $(\cos x)^4$ será substituída por $1/8 \times \cos 4x + 1/2 \times \cos 2x + 3/8$.

Na realidade o conceito de fórmula simplificada difìcilmente é reductível a um conceito único. A fim de evitar os inconvenientes referidos, os modernos programas introduzem mecanismos que facultam quer operar sobre fórmulas segundo as conveniências do utilizador, quer criar estados diferentes no programa manipulador que o obriguem a actuar segundo essas conveniências. O utilizador poderá usar fórmulas expandidas ou não conforme a natureza das mesmas.

É certamente como consequência da não existência dum conceito único de fórmula simplificada que se deve não só o aparecimento. (por volta de 1968), dum certo número de linguagens conversacionais, como também a modificação de algumas das já existentes, por forma a dotá-las das mesmas possibilidades. O utilizador poderá, por meio duma consola e dum visor de raios catódicos, comandar o desenvolvimento das suas fórmulas, particularizar as soluções das suas equações, etc. De entre as linguagens deste tipo apontamos apenas a nova versão de MATH LAB 68, REDUCE, MAGIC PAPER e Simbolic Mathematical Laboratory.

Consideremos agora a utilização de funções algébricas e transcendentes. O matemático usa um certo número de funções que representa por meio de siglas ou cadeias alfanuméricas (sin, cos, sqrt, ...). A necessidade de utilizar tais siglas em fórmulas, implica a necessidade de dotar os programas manipuladores de rotinas que actuem sobre os seus argumentos e os simplifiquem.

Duma maneira geral, podemos considerar uma fórmula, como sendo constituída por uma função racional, podendo ter, em vez de algumas das suas variáveis, siglas representativas de funções, as quais, por sua vez, terão por argumento funções racionais (com ou sem novas siglas).

O conjunto de siglas seleccionadas, juntamente com a família de funções racionais, delimita o campo de actuação dum manipulador de fórmulas. A possibilidade de aí utilizar um operador de derivação não oferece dificuldade de maior. Foi certamente esse o primeiro elemento dum manipulador pois já em 1953 existiam algorítmos para efectuar a derivação.

Já a existência de siglas cria problemas delicados na simplificação de fórmulas. Para vermos isso mais detidamente, consideremos as duas instruções

PI : = 3 14159265359 ;
F : = SIN(PI/2 – X) — COS(X) ;

que tanto podem pertencer a um programa destinado à manipulação de fórmulas, como a um programa destinado ao tratamento dum problema numérico. É evidente que, pertencendo a um programa destinado ao tratamento dum problema numérico, para todo o valor real que for atribuído a X, o valor de F será 0. Isso implica que a última instrução seja procedida, pelo menos, duma instrução que atribua um valor a X. Só assim ela poderá ser executável. No programa compilado haverá o cálculo dos dois argumentos e a chamada a duas subrotinas que calcularão o seno e o cosseno. Obter-se-ão assim dois valores sensìvelmente iguais que, por diferença, conduzirão a um resultado nulo.

O tratamento das duas instruções por um manipulador de fórmulas terá de ser completamente diferente, pois, o resultado zero deverá ser obtido independentemente de qualquer valor, real ou complexo, atribuído a X. Em particular, poderá mesmo não lhe ter sido atribuído nenhum valor. Isso cria o problema da representação interna da função que corresponde a uma determinada sigla por forma a permitir não só o cálculo da função para qualquer valor, real ou complexo do argumento, como também a simplificação de expressões contendo a sigla, como ainda a sua transformação no caso da função admitir alguma expressão algébrica de adição (como é o caso do exemplo acima).

Para concluir este parágrafo, vamos abordar o problema da integração formal. Ao contrário do que acontece com a derivação, a integração formal de fórmulas por meio dum manipulador cria problemas muito delicados. Isso não será de surpreender por quanto o cálculo duma primitiva depende muito da experiência pessoal do matemático. Realmente, o matemático recorre a métodos heurísticos, os quais são de difícil mecanização. Além disso, no seio duma família de fórmulas definidas com suficiente generalidade (Funções racionais + siglas), existem sempre elementos que não admitem primitiva nessa família. É o caso da função racional $1/x$ cuja primitiva não é racional, é o caso de EXP$(-x \uparrow 2)$, cuja primitiva não é uma exponencial, etc. Finalmente, relacionado com este problema, existe ainda o problema da factorização de polinómios e o da decomposição das funções racionais em elementos simples. Porém, apesar das dificuldades assinaladas, a primitivação mecânica encontra-se numa fase muito avançada de resolução, permitindo o cálculo da primitiva duma fórmula, caso ela exista na família admissível, ou o isolamento das partes não primitiváveis sempre que estas sejam detectadas [4].

Definição de fórmula. Demos, no parágrafo anterior, uma caracterização não formal de fórmula. Vamos seguidamente procurar representar o mesmo conceito duma maneira formal. Porém, a fim de aligeirarmos essa representação, suprimimos algumas definições menos relevantes. Algumas, o leitor completá-las-á fàcilmente reportando-se, por exemplo, ao relatório revisto da linguagem ALGOL 60, outras, que variam de manipulador para manipulador, encontrá-las-á nas descrições de qualquer linguagem particular que tencione utilizar.

Tomaremos para sintaxe do conceito de fórmula :

$$
\begin{aligned}
<\text{fórmula}> &::= <\text{termo}> \mid + <\text{termo}> \mid \\
&\quad - <\text{termo}> \mid \\
&\quad <\text{fórmula}> + <\text{termo}> \mid \\
&\quad <\text{fórmula}> - <\text{termo}> \\
<\text{termo}> &::= <\text{factor}> \mid \\
&\quad <\text{termo}> * <\text{factor}> \mid \\
&\quad <\text{termo}> / <\text{factor}> \mid \\
&\quad <\text{termo}> : <\text{fórmula}> \\
<\text{factor}> &::= <\text{primário}> \mid \\
&\quad <\text{factor}> \uparrow <\text{primário}> \\
\text{primário} &::= <\text{número}> \mid <\text{variável}> \mid \\
&\quad (<\text{fórmula}>) \mid \\
&\quad <\text{identificador}> \mid \\
&\quad \text{SIGLA } (<\text{fórmula}>) \mid \\
&\quad <\text{substituição}> \mid \\
&\quad <\text{derivada}> \\
\text{SIGLA} &::= \text{EXP} \mid \text{LN} \mid \text{SIN} \mid \text{COS} \mid \\
&\quad \text{ARCTAN} \mid \text{SQRT} \mid \text{SINH} \mid \\
&\quad \text{COSH} \mid \text{CC} \mid \text{SIMPL} \\
<\text{substituição}> &::= \text{SUBST } (<\text{fórmula}>, \\
&\quad <\text{lista de pares}>) \\
<\text{Derivada}> &::= \text{DER } (<\text{fórmula}>, \\
&\quad <\text{variável}>) \\
<\text{lista de pares}> &::= <\text{variável}>, \\
&\quad <\text{fórmula}> \mid <\text{lista de} \\
&\quad \text{pares}>, <\text{variável}>, \\
&\quad <\text{fórmula}>
\end{aligned}
$$

Nesta descrição CC corresponde à passagem à conjugada, SIMPL à simplificação de fórmulas. Quanto à operação :, que introduzimos na descrição de termo, não aparece usualmente em nenhuma formalização de manipuladores que conhecemos. Introduzimo-la porque nos tem permitido manipular duma maneira muito cómoda fracções contínuas e calcular os seus aproximantes. O seu significado será equivalente a / (...) em que o parêntesis é estendido, quer até ao fim da fórmula, quer até ao primeiro sinal) que não tenha correspondente à direita do sinal :.

Quanto à sigla SUBST destina-se a atribuir valores às diferentes variáveis que figuram na lista. Cada variável será substituída pela fórmula que a segue. Esta pode ser um valor numérico.

A descrição dada acima permitir-nos-á fazer atribuições numa família de fórmulas muito vasta. No decurso desta nota, sempre que apareça qualquer sigla que não figure nessa descrição será definida na ocasião.

Todo o manipulador de fórmulas representa estas com uma representação padrão. Porém, essa representação varia de manipulador para manipulador. Uma tal diversidade é devida não tanto à natureza das fórmulas manipuladas ou às operações definidas sobre elas, como à maior ou menor facilidade com que a linguagem em que o próprio manipulador foi redigido opera sobre as estruturas de informação. Assim, não será de estranhar que um manipulador, redigido em Fortran, represente as fórmulas como simples cadeias, um outro, redigido em Algol, as represente como arborescências binárias, ou um terceiro, redigido em Lisp, utilize listas para essa representação.

Domínios de aplicação. Os domínios de aplicação dos manipuladores de fórmulas são variados podendo desde já prever-se, para breve, o aparecimento de vasta literatura sobre o assunto. Dum modo geral, um manipulador de fórmulas tem um papel a desempenhar onde seja necessário efectuar cálculos analíticos longos ou onde seja de desejar um acréscimo na fiabilidade dos resultados finais.

Damos seguidamente alguns exemplos de trabalhos realizados com auxílio de manipuladores:

Cálculo das Efemérides Lunares. Trata-se da revisão dos cálculos executados sob a orientação de DELAUNAY e aos quais este consagrou grande parte da sua existência. Levou-se agora a aproximação mais longe do que então. O cálculo pôde ser efectuado em seis sessões de três horas cada.

Cálculo dos símbolos de CHRISTOFELL. Tem sido um programa construído em várias das linguagens dada a sua importância em relatividade geral.

Transformações canónicas. Formulado um problema de mecânica quântica num conjunto de variáveis, efectuar uma mudança de variáveis canónicas.

Temos ainda encontrado literatura com diversas aplicações a problemas de acústica, filas de espera, mecânica quântica, mecânica celeste, etc.

Seguidamente, para que o leitor possa averiguar melhor os domínios de aplicação dos manipuladores, damos alguns exemplos de aplicação obtidos com um pequeno manipulador com que temos trabalhado. Destinado a ordenadores de dimensões médias, tal manipulador não permitirá, talvez, grandes voos. Tem, porém, um valor didático evidente. A sua concepção original é de VAN DE RIET. Usamos uma versão devida a STELLA ATKINS da Universidade de Warwick, na qual introduzimos algumas modificações.

O primeiro problema que apresentaremos (a que demos o nome de problema n.º 5) consistirá no estudo do polinómio

$$p(x)=x^{12}+2\cdot x^{11}-4\cdot x^9-7\cdot x^8-4\cdot x^7+$$
$$+2\cdot x^5+5\cdot x^4+8\cdot x^3+12\cdot x^2+8\cdot x+4$$

sabendo-se que o mesmo tem raízes múltiplas.

Se o polinómio tem raízes múltiplas, essas raízes serão comuns a $p(x)$ e a $q(x)=p'(x)$. Então $h(x)$, máximo divisor comum dos dois polinómios e $g(x)=h(x)/h(0)$ terão como raízes as raízes múltiplas de $p(x)$, todas com o grau de multiplicidade diminuído duma unidade. Desta forma, $f(x)=$ $=p(x)/g(x)$ terá as mesmas raízes que $p(x)$, sendo todas elas simples. As raízes múltiplas serão as raízes de $r(x)$ máximo divisor comum de $g(x)$ e de $f(x)$ e as raízes simples serão as raízes de $b(x)=$ $=g(x)/r(x)$.

O programa que damos a seguir segue a par e passo este raciocínio. A sigla OUTR implica uma edição de fórmula, QUOT um quociente de fórmulas e COMMDIV o máximo divisor comum.

PROGRAMA

```
«PROBLEMA 5»
FIX;
P:=X↑12+2*X↑11—4*X↑9—
    7*X↑8—4*X↑7+2*X↑5+
    5*X↑4+8*X↑3+12*X↑2+
    8*X+4;
Q:=DER(P,X);
H:=COMMDIV(P,Q);
G:=H/SUBST(H,X,0);
OUTR(G:=G);
F:=QUOT(P,G,A);
OUTR(F:=F);
R:=COMMDIV(G,F);
OUTR(R:=R);
S:=R/SUBST(R,X,0);
OUTR(S:=S);
B:=QUOT(F,S,C);
OUTR(B:=B);
ERASE;
END;
```

RESULTADOS

```
PROBLEMA 5
G:=—1/2*X↑6—X↑5—1/2*X↑4+
    X↑3+5/2*X↑2+2*X+1;
F:=—2*X↑6+2*X↑4+2*X↑2+4;
R:=—2*X↑4—2*X↑3+2*X↑2+
    4*X+4;
S:=—1/2*X↑4—1/2*X↑3+1/2*X↑2
    +X+1;
B:=4*X↑2—4*X+4;
```

Poder-se-ia seguidamente tratar duma forma análoga $g(x)$.

O problema que seguidamente apresentamos (a que demos o nome de problema n.º 11) consiste na determinação do desenvolvimento, pela fórmula de Maclaurin, de $1/\sqrt{1+2\cdot m\cdot X+n\cdot X^2}$.

No programa, a sigla TPS designa série truncada, NOTEXP cria um estado que impede a simplificação de fórmulas e EXPAND repõe essa simplificação. Fazemos a edição da função, do seu valor e do das suas derivadas calculadas para $x=0$. Editamos ainda a função expandida com 4 termos.

PROGRAMA

```
«PROBLEMA 11»
FIX;
NOTEXP;
F:=1/SQRT(1+2*M*X+N*X↑2);
EXPAND;
OUTR(F:=F);
C0:=SUBST(F,X,0);
OUTR(C0:=C0);
G:=SIMPL(DER(F,X));
C1:=SUBST(G,X,0);
OUTR(C1:=C1);
H:=SIMPL(DER(G,X));
C2:=SUBST(H,X,0);
OUTR(C2:=C2);
P:=SIMPL(DER(H,X));
C3:=SUBST(P,X,0);
OUTR(C3:=C3);
Q:=SIMPL(ER(P,X));
C4:=SUBST(Q,X,0);
OUTR(C4:=C4);
Y:=TPS(X,C0,C1,C2/2,C3/6,C4/24);
OUTR(Y:=Y);
ERASE;
END;
```

RESULTADOS

```
PROBLEMA 11
F:=1/SQRT(X↑2*N+2*M*X+1);
C0:=1;
C1:=—M;
C2:=3*M↑2—N;
C3:=—15*M↑3+9*M*N;
C4:=105*M↑4—90*M↑2*N+9*N↑2;
```

```
Y:=1+(−M)*X+(3/2*M↑2−1/2*N)*X↑2+(−5/2*M↑3+3/2*M*N)*X↑3+
    (35/8*M↑4−15/4*M↑2*N+3/8*N↑2)*X↑4;
```

Finalmente, o terceiro problema (a que demos o nome de cálculo das variações n.º 1) consiste na determinação das equações de EULER, e integral primeiro correspondente ao funcional

$$I=\int_a^b \frac{\sqrt{1+y'^2}}{x}\,dx.$$

No programa, SPECDER é uma instrução qne permite dar nome às derivadas de funções. No exemplo chamamos DY à derivada de Y em ordem a X, e DDY à derivada de DY em ordem a X.

PROGRAMA

```
«CÁLCULO DAS VARIAÇÕES 1»
    FIX;
    L:=SQRT(1+DY↑2)/X;
    OUTR(FUNÇÃO:=L);
    SPECDER (X,Y,DY,DY,DDY);
    E:=DER(DER(L,DY),X)−
        DER(L,Y);
    OUTR(EQ. DE EULER 0:=E);
    C:=L−DY*DER(L,DY);
    OUTR(INTEGRAL PRIM. C:=C);
ERASE;
END;
```

RESULTADOS

```
CÁLCULO DAS VARIAÇÕES
FUNÇÃO:=EXP(1/2*LN(DY↑2+1))/X;
EQ. DE EULER 0:=(−DY↑3*EXP(1/2*LN(DY↑2+1))+X*DDY*EXP(1/2*
    LN(DY↑2+1))−DY*EXP(1/2*LN(DY↑2+1))/(DY↑4*X↑2+2*DY↑2*
    X↑2+X↑2);
INTEGRAL PRIM. C:=EXP(1/2*LN(DY↑2+1))/(DY↑2*X+X).
```

Chamamos a atenção do leitor para a forma como o processador tratou neste problema SQRT(DY↑2+1). A forma adoptada tornou-se necessária para permitir a simplificação das fórmulas. No problema 11 essa notação foi travada.

BIBLIOGRAFIA

A bibliografia sobre este tema é muito extensa e dispersa. Recomendamos ao leitor que deseje aprofnndar o assunto os artigos:

[1] JEAN SAMMET, *Formula manipulation by computer*, Anvances in computers n.º 8.

[2] ———, *Survey of Formula manipulation*, Comm of ACM n.º 8 Vol. 9.

[3] JOEL MOSES, *Algebric Simplification : A guide for the perplexed*, Comm. of the ACM n.º 8 Vol. 14.

[4] ———, *Simbolic integration: The stormy decade*, idem.

[5] ———, *Towrds a general theory of special functions*, Comm. of the ACM n.º 7 Vol· 15.

[6] M. E. ENGELI, *Formula manipulation — The user's points of view*, Advances in information systems science Vol. 1.

[7] VAN DE RIET, *Formula manipulation in Algol 60.*

[8] STELA ATKINS and H. S. JONES, *Formula manipulation using Algol 60.*