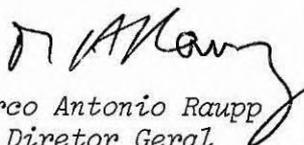


1. Publicação nº <i>INPE-3649-PRE/816</i>	2. Versão	3. Data <i>Set., 1985</i>	5. Distribuição <input type="checkbox"/> Interna <input checked="" type="checkbox"/> Externa <input type="checkbox"/> Restrita
4. Origem <i>DIN/DCS</i>	Programa <i>DENUME</i>		
6. Palavras chaves - selecionadas pelo(s) autor(es) <i>LISP</i> <i>MATEMÁTICA SIMBÓLICA</i> <i>TRANSFORMADA DE LAPLACE</i>			
7. C.D.U.: <i>681.3.019:517.442</i>			
8. Título <i>TRANSFORMADA DE LAPLACE</i> <i>NO REDUCE 2</i>		10. Páginas: <i>37</i>	
		11. Última página: <i>B.10</i>	
9. Autoria <i>Frederico L.G. de Freitas*</i> <i>Guilherme Bittencourt</i>		12. Revisada por  <i>Edson Luiz Franca Senne</i>	
Assinatura responsável 		13. Autorizada por  <i>Marco Antonio Raupp</i> <i>Diretor Geral</i>	
14. Resumo/Notas <p><i>Neste trabalho descreve-se um sistema para determinação analítica de transformadas de Laplace, sua estrutura e alguns resultados obtidos com o sistema. Desenvolveu-se o sistema contando com a capacidade de manipulação simbólica apresentada pelo programa "REDUCE 2", que é um sistema geral de manipulação simbólica desenvolvido na Universidade de Utah, E.U.A. Utilizou-se o computador B6800 do INPE/CNPq. Para realizar as transformadas de Laplace implementaram-se rotinas, na linguagem RLISP, que expandem a capacidade de manipulação simbólica do programa "REDUCE 2".</i></p>			
15. Observações <i>*Universidade Federal de Pernambuco (UFPE).</i> <i>Este trabalho foi apresentado no 2º Encontro Regional de Matemática Aplicada e Computacional, 16 e 17 de maio de 1985 - INPE.</i>			

ABSTRACT

In this report a system for Laplace transform analytic calculation is presented; its structure and some results obtained making use of the system are shown. The system was developed based on the symbolic manipulation capabilities of the REDUCE 2 program, a general symbolic manipulation system, developed at the Utah University, USA. The B-6800 INPE-CNPq computer was used. In order to perform the Laplace transformations, routines were implemented, in the RLISP Language, to expand the symbolic manipulation capabilities of the REDUCE 2 program.

SUMÁRIO

	<u>Pág.</u>
1 - <u>INTRODUÇÃO</u>	1
2 - <u>O SISTEMA REDUCE 2</u>	2
2.1 - Módulo Algébrico	4
2.2 - Módulo Simbólico	4
2.3 - Módulo Integrador	5
2.4 - Módulo "LISP"	5
3 - <u>O SISTEMA PARA TRANSFORMADA DE LAPLACE</u>	5
3.1 - A Transformada de Laplace	5
3.2 - Implementação do Sistema	7
4 - <u>CONCLUSÃO</u>	8
<u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	
<u>APÊNDICE A - EXEMPLOS DE UTILIZAÇÃO</u>	
<u>APÊNDICE B - LISTAGEM DOS PROCEDIMENTOS</u>	

1. INTRODUÇÃO

Originariamente os computadores foram criados com o objetivo explícito de realizar rapidamente cálculos numéricos. No entanto, a partir de meados da década de 50, logo após o surgimento das primeiras linguagens de alto nível, a capacidade de memória e processamento dos computadores passaram a ser utilizadas na manipulação de símbolos, codificados internamente como números.

Inicialmente foram implementados programas para tarefas simples como ordenação alfabética, edição e formatação de textos, armazenagem e manipulação de grande número de dados não-numéricos, mas aos poucos estas manipulações foram se complicando a ponto de hoje existirem programas que jogam damas (ao nível de campeão mundial), xadrez e outros jogos de tabuleiro, resolvem integrais, fazem diagnósticos médicos ou geológicos e muitos outros.

Destas aplicações onde os programas parecem exibir inteligência nasceu a "Inteligência Artificial". Dentre as aplicações de Inteligência Artificial, os programas cuja área de "conhecimento" fosse a Matemática foram reunidos na área chamada "Matemática Simbólica". As aplicações de Matemática Simbólica são programas que manipulam conceitos matemáticos solucionando problemas como integração e derivação de funções, problemas de álgebra polinomial entre outros.

Existem atualmente muitos sistemas de manipulação simbólica, alguns específicos para determinados problemas e outros de caráter geral. Estes últimos são escritos, em geral, em LISP e contam com capacidades gerais de manipulação como derivação e integração de funções, álgebra polinomial, tratamento de matrizes simbólicas, etc. Contando ainda, em maior ou menor grau, com capacidade de programação que permite sua expansão. Os sistemas gerais mais conhecidos são MACSYMA, desenvolvido pelo MIT, e REDUCE 2, desenvolvido na Universidade de Utah.

Este trabalho consiste em uma série de procedimentos escritos na linguagem RLISP que, incorporados ao sistema REDUCE 2, permitem a determinação analítica da transformada de Laplace de uma dada classe de funções. O sistema descrito neste trabalho fez parte do "Trabalho de Graduação" de Frederico L.G. de Freitas no "Curso Superior de Tecnologia de Computação" do Instituto de Tecnologia de Aeronáutica do Centro Técnico Aeroespacial em dezembro de 1984, tendo como orientadores Guilherme Bittencourt, do INPE, e Luiz Carlos Rossato, do ITA.

A seguir, na Seção 2, apresenta-se uma breve descrição do sistema REDUCE 2 utilizado e, principalmente, de sua estrutura interna. Na Seção 3 descreve-se o sistema para transformadas de Laplace que foi desenvolvido, comentando sua estrutura e sua capacidade atual. Nas conclusões apresentam-se alguns resultados e comentam-se algumas aplicações e possíveis expansões ao sistema. Finalmente, nos apêndices, apresentam-se alguns exemplos de seções de terminal com utilização do sistema e uma listagem com comentários dos procedimentos.

2. O SISTEMA REDUCE 2

Nesta seção descreve-se o sistema de manipulação simbólica "REDUCE 2" (Hearn, 1968). Este sistema permite a execução de cálculos algébricos gerais de interesse nas áreas de Física, Matemática e Engenharia. Suas capacidades incluem:

- Expansão e ordenamento de polinômios e funções racionais;
- Diferenciação e integração simbólica;
- Substituições gerais de variáveis;
- Cálculo do máximo divisor comum de dois polinômios;
- Simplificação de expressões;

- Cálculos com matrizes simbólicas;
- Uma linguagem completa para cálculos simbólicos;
- Cálculos de interesse para Física de Alta Energia;
- Operação com tensores.

Pode-se apresentar o sistema como composto por quatro módulos: Algébrico, Simbólico, Integrador e "LISP".

Ao ser ativado o programa "REDUCE 2", o controle é entregue ao módulo algébrico. Para passar de um módulo para outro basta digitar comandos específicos, os quais são apresentados na Figura 2.1.

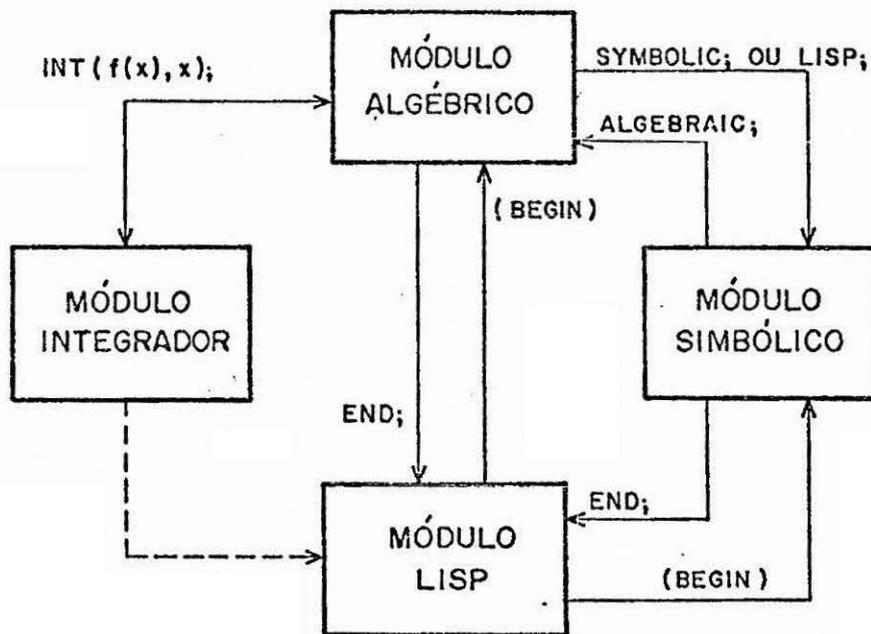


Fig. 2.1 - Módulos do REDUCE 2.

A seguir descrevem-se os quatro módulos:

2.1 - MÓDULO ALGÉBRICO

Este módulo contém o manipulador simbólico propriamente dito. Todas as capacidades citadas no início deste capítulo podem ser acessadas a partir deste módulo.

Esse consiste num interpretador para a linguagem RLISP, uma linguagem de alto nível muito semelhante ao "ALGOL", com a particularidade de que as variáveis podem assumir tanto valores numéricos quanto simbólicos. As operações matemáticas usuais estão presentes, bem como algumas funções básicas (SIN, COS, LOG, etc.).

Além disto estão presentes algumas funções específicas para manipulação simbólica como, por exemplo, a função "DF (Expressão, Variável, Ordem)" que calcula a derivada de qualquer ordem da expressão dada, ou a função "COEFF (Variável, Expressão, Vetor de coeficientes)" que permite obter o coeficiente de uma determinada variável, numa dada potência, dentro de uma expressão.

No entanto, a flexibilidade maior do sistema consiste na possibilidade de expansão destas capacidades de manipulação através da definição de operadores no modo simbólico, como será descrito abaixo na Seção 2.2.

2.2 - MÓDULO SIMBÓLICO

Este módulo consiste num tradutor da sintaxe "RLISP" para a sintaxe "LISP", isto é, quando se está operando em modo simbólico é possível escrever programas para o "LISP", do módulo descrito na Seção 2.4, utilizando a sintaxe "ALGOL-Like" do "RLISP".

A grande utilidade deste módulo reside no fato de que os procedimentos escritos em "RLISP" podem ser declarados "OPERATOR" e utilizados a partir do modo algébrico, oferecendo uma possibilidade de expansão da capacidade de manipulação simbólica do sistema.

2.3 - MÓDULO INTEGRADOR

Este módulo é um programa escrito em "LISP" com a finalidade de realizar a integração simbólica de funções. Estão presentes neste módulo procedimentos heurísticos de integração, além de procedimentos que implementam o enfoque algorítmico dado ao problema de integração por Risch (1969). A interação com o usuário dá-se através do operador "INT(Função, Variável)" a partir do modo algébrico.

2.4 - MÓDULO "LISP"

Este módulo consiste num interpretador da linguagem "LISP" (Martí et alii, 1979), e serve como base para os demais módulos escritos nesta linguagem. Neste módulo estão implementados, também, as interfaces para os vários periféricos específicos de cada instalação.

No sistema utilizado no presente trabalho o módulo "LISP" apresenta uma limitação que deve ser mencionada: não são admitidos números em ponto flutuante, apenas inteiros e números racionais. Esta limitação não é inerente ao sistema "REDUCE 2", mas apenas ao sistema específico que foi utilizado. Em outras instalações esta limitação, em geral, não existe.

3. O SISTEMA PARA TRANSFORMADA DE LAPLACE

Nesta seção define-se, sem entrar em detalhes matemáticos, a transformada de Laplace (Spiegel, 1979), descrevendo-se a seguir algumas de suas propriedades que foram implementadas no presente sistema. Logo após comentam-se alguns detalhes de implementação.

3.1 - A TRANSFORMADA DE LAPLACE

A transformada de Laplace é uma transformada integral que apresenta entre outras vantagens a de transformar equações diferenciais

em equações algébricas. Define-se a transformada de Laplace de uma função $f(t)$ como a integral:

$$F(s) = L[f(x)] = \text{INT} (e^{*-s*t}) * f(t), t, 0, \text{INFINITO}),$$

onde s - é a variável na qual é obtida a transformada,

INT - é o operador integral,

INFINITO - é o limite superior da integral,

e - é a base dos logaritmos naturais, e finalmente,

F - é a transformada de Laplace de f .

Diz-se que a transformada existe quando a integral converge para algum valor de s .

A transformada de Laplace conta com uma série de propriedades, algumas das quais foram implementadas neste trabalho:

a) Linearidade em relação à soma e multiplicação. Esta propriedade permite que se escreva:

$$L[a_1*f_1(t) + a_2*f_2(t) + \dots] =$$

$$a_1*L[f_1(t)] + a_2*L[f_2(t)] + \dots,$$

onde os a_i são constantes.

b) Propriedade de translação. Esta propriedade aplica-se a funções que tenham um fator multiplicativo da forma $e^{*(a*t)}$. A propriedade afirma que:

$$L[e^{*(a*t)}*f(t)] = F(s-a),$$

onde $F(s) = L[f(t)]$ e a é uma constante,

c) Multiplicação por potência da variável. Esta propriedade afirma que:

$$L[f(t) * t^{**n}] = L[f(t)]^{(n)} * (-1)^{**n},$$

onde n é um número inteiro e o expoente (n) significa a derivada n ésima da função considerada em relação a s .

Além de implementar as propriedades descritas acima, o sistema conta com uma tabela básica de transformada que contém, atualmente, as transformadas de constantes, seno, co-seno, seno hiperbólico, e co-seno hiperbólico.

3.2 - IMPLEMENTAÇÃO DO SISTEMA

O sistema é invocado a partir do módulo algébrico do sistema REDUCE 2 (Hearn, 1973), recebendo quatro parâmetros na seguinte forma:

```
>TL (f (X), L, X, S);
```

onde X é a variável da função a ser transformada (este símbolo pode ser qualquer um com exceção de "T", pois em LISP o símbolo T é reservado para o valor lógico "TRUE"), S a variável da transformada, L o operador a ser aplicado (futuramente ao ser implementada a transformada inversa, esta será invocada com o operador LI) e $f(X)$ a função a ser transformada, escrita na sintaxe característica de linguagens de programação com ALGOL ou PASCAL.

O sistema REDUCE 2 faz a conversão da sintaxe de entrada para a notação polonesa utilizada na linguagem LISP, onde uma expressão aritmética aparece sob a forma de uma lista na qual o primeiro elemento

é o operador e os demais são parâmetros. Assim a função "E**X+X+3*Y" será traduzida para:

(PLUS X (EXPT E X) (TIMES 3 Y)),

por exemplo.

A seguir o sistema tenta aplicar as propriedades descritas acima e finalmente utiliza a tabela básica para determinar as transformadas. Caso o sistema não consiga calcular a transformada, a operação é deixada indicada utilizando o operador "TLA(f(t))", onde f(t) é a parte da função que não foi possível transformar. Caso o usuário inicialize a variável FALA com o valor SIM, o sistema gera mensagens de acompanhamento, informando cada propriedade aplicada ou qualquer problema que surja durante o cálculo.

4. CONCLUSÃO

O objetivo ao implementar o presente sistema foi construir uma ferramenta que permita a programação, no "REDUCE 2", de procedimentos matemáticos que envolvam transformadas de Laplace fornecendo a infra-estrutura necessária para este tipo de cálculo. Um exemplo de utilização para este sistema seria a solução de equações diferenciais utilizando transformadas de Laplace, o que poderia ser adicionado ao sistema de solução destas equações existente no INPE (Bittencourt, 1984).

Atualmente a classe de funções tratáveis pelo sistema pode ser descrita pela expressão:

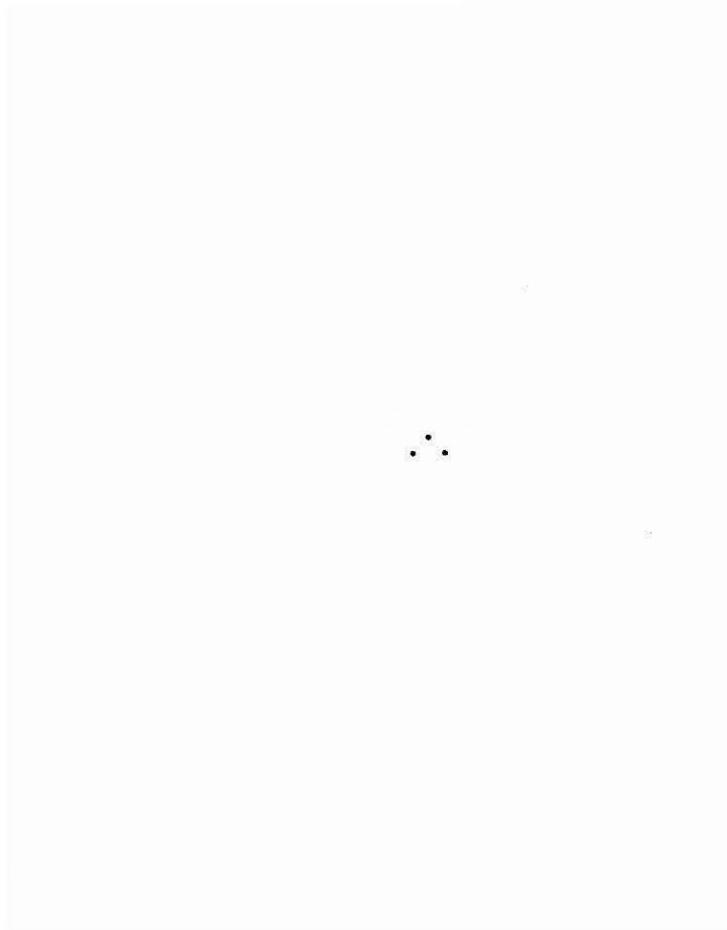
$$\text{SOMA } (a_i * t^{b_i} * e^{c_i * t} * f_i(d_i * t), i, 0, N),$$

onde a_i , b_i , c_i , d_i e N são constantes, e o operador SOMA significa o somatório de seu primeiro parâmetro com o segundo parâmetro variando entre os valores do terceiro e quarto parâmetros. As funções f_i presente

mente podem ser SIN, COS, COSH e SINH. Para estender a variedade destas funções, a tabela de busca deve ser acrescida das funções desejadas.

Embora o sistema solucione a quase totalidade das transformadas usadas na prática, sua arquitetura modular permite extensões sem maiores dificuldades. Com a implementação de limites no "REDUCE 2", outras propriedades das transformadas, que recaem sobre integrais impróprias, poderiam ser implantadas. Foram evitados métodos de raro uso operacional, além daqueles que não constituem um meio seguro em relação à direção da resposta, como o método de diferenciação em relação a um parâmetro e a uma propriedade das funções periódicas.

Adicionalmente, o sistema possui facilidades para a construção de aplicações semelhantes. A linearização, por exemplo, serve para qualquer operador que o usuário definir. Há também instruções LISP redefinidas (APPEND, por exemplo), que garantem uma resposta ao usuário.



REFERÊNCIAS BIBLIOGRÁFICAS

- BITTENCOURT, G. *Solução analítica de equações diferenciais ordinárias de primeira e segunda ordem por computador*, Dissertação de Mestrado, INPE, São José dos Campos, out. 1984. (INPE-3507-TDL/192)
- HEARN, A.C. REDUCE, a user-oriented interactive system for algebraic simplification. IN: ACM proceedings of the ACM SYMPOSIUM ON INTERACTIVE SYSTEMS FOR EXPERIMENTAL APPLIED MATHEMATICS. Washington, D.C., ACM, 1968.
- HEARN, A.C. *REDUCE 2 user's manual*. Salt Lake City, Utah, U.S.A., University of Utah, 1973.
- MARTI, J.B.; HEARN, A.C.; GRISS, C. Standart LISP report. Salt Lake City, UT., University of Utah. 1979.
- RISCH, R. The problem of integration in finite terms. *Transactions of the AMS*, 139:167-189, May, 1969.
- SPIEGEL, M.R. *Transformadas de Laplace e Aplicações*, Coleção Schawn, Editora McGraw-Hill, 1979.

•••

APÊNDICE A

EXEMPLOS DE UTILIZAÇÃO

Neste apêndice apresentam-se as listagens de três seções CANDE no computador B6800 do INPE/CNPq, nas quais foi utilizado o sistema de transformadas de Laplace.

Na primeira seção (arquivo SCHOUT/TABELA) foram submetidas ao sistema as funções básicas que aparecem na maioria das tabelas de transformadas de Laplace. A variável FALA não foi inicializada e assim o sistema apenas apresentou as soluções sem comentar os métodos de solução.

Na segunda seção (arquivo SCHOUT/S/OUTROS) foram solucionadas algumas transformadas um pouco mais complexas e, desta vez, a variável FALA recebeu o valor SIM, o que faz com que o sistema comente as propriedades utilizadas na solução.

Finalmente na terceira seção (arquivo SCHOUT/S/ERROS) apresentou o comportamento do sistema em algumas condições de erro como desconhecimento da função a ser transformada, função com denominador inválido, etc.

SCHOUT/S/TABELA (05/14/85)

#SCHEDULE #00029 ABORTED BY ERROR IN SESSION
SESSION(S): 2842.

#B6800:1374 CANDE 32.230; YOU ARE SCHED#000(73) ; 0 SESSIONS QUEUED
#BEM VINDO AO SISTEMA DE TIME-SHARING DO INPE *** NEWS DE 24/04/85
*URGENTE

0 PLANTAO DE ATENDIMENTO AOS USUARIOS
?? 8:15 ?? 11:45 ????? ?
?? 13:45 ?? 17:15 ?????.

A ?????????? ????? ?? ?????????? ??? ?????????? ??? ???? ??-
????????? ?? ?????????? ?? ?????????? ???? ?????????? 366.

#FAMILY DISK = PACK OTHERWISE DISK.

#SESSION 2842 17:29:32 05/14/85

R REDUCE

#WAITING FOR AVAILABLE TASK

#RUNNING 2873

#?

OPTIONS SET:

PAIR SPACE = 14 PAGES

REDUCE 2 (APR-15-79) ...

> OFF EXP;

> IN LAPLACES

> TL(A,L,X,S);

A/S

> TL(A*X,L,X,S);

2

A/S

> TL(A*X**2,L,X,S);

3

(2*A)/S

> TL(A*X**3,L,X,S);

4

(6*A)/S

> TL(A*X**4,L,X,S);

5

(24*A)/S

> TL(A*E**E*(E*X),L,X,S);

(- A)/(B - S)

> TL(A*X*E**E*(E*X),L,X,S);

$$A/(E - S)$$

$$> TL(A*X**2*E**(E*X),L,X,S);$$

$$(- 2*A)/(E - S)^3$$

$$> TL(A*X**3*E**(E*X),L,X,S);$$

$$(6*A)/(E - S)^4$$

$$> TL(A*X**4*E**(E*X),L,X,S);$$

$$(- 24*A)/(E - S)^5$$

$$> TL(A*SIN(E*X),L,X,S);$$

$$(A*B)/(B^2 + S^2)$$

$$> TL(A*COS(E*X),L,X,S);$$

$$(A*S)/(B^2 + S^2)$$

$$> TL(A*X*SIN(E*X),L,X,S);$$

$$(2*A*B*S)/(B^2 + S^2)$$

$$> TL(A*X*COS(E*X),L,X,S);$$

$$(- A*(B^2 - S^2))/(B^2 + S^2)$$

$$> TL(A*X**2*SIN(E*X),L,X,S);$$

$$(- 2*A*B*(E^2 - 3*S^2))/(B^2 + S^2)^3$$

$$> TL(A*X**2*COS(E*X),L,X,S);$$

$$(- 2*A*S*(3*E^2 - S^2))/(B^2 + S^2)^3$$

$$> TL(A*X**3*SIN(E*X),L,X,S);$$

$$(- 24*A*B*S*(E^2 - S^2))/(B^2 + S^2)^4$$

$$> TL(A*X**3*COS(E*X),L,X,S);$$

$$(6*A*(B^4 - 6*E^2*S^2 + S^4))/(B^2 + S^2)^4$$

$$> TL(A*E**(B*X)*SIN(C*X),L,X,S);$$

```
(A*C)/(C + ( - B + S ) )
> TL(A*E**(B*X))*COS(C*X),L,X,S);

( - A*(B - S))/(C + ( - B + S ) )
> TL(A*SINH(B*X),L,X,S);

( - A*B)/(B - S )
> TL(A*COSH(B*X),L,X,S);

( - A*S)/(B - S )
> TL(A*X*SINH(B*X),L,X,S);

(2*A*B*S)/(B - S )
> TL(A*X*COSH(B*X),L,X,S);

(A*(B + S ))/(B - S )
> TL(A*X**2*SINH(B*X),L,X,S);

( - 2*A*B*(B + 3*S ))/(B - S )
> TL(A*X**2*COSH(B*X),L,X,S);

( - 2*A*S*(3*B + S ))/(B - S )
> TL(A*X**3*SINH(B*X),L,X,S);

(24*A*B*S*(B + S ))/(B - S )
> TL(A*X**3*COSH(B*X),L,X,S);

(6*A*(B + 6*B*S + S ))/(B - S )
> TL(A*E**(B*X))*SINH(C*X),L,X,S);

( - A*C)/(C - ( - B + S ) )
> TL(A*E**(B*X))*COSH(C*X),L,X,S);
#2873 ELAPSED TIME LIMIT EXCEEDED @ 03E:0023:3*
#R-DS @ 03E:0023:3, 03F:00AE:1, 04B:0007:2, 246:000D:4, 243:0053:0,
217:0(11:2, 216:0010:0, 210:0034:4, 20F:0006:1, 36C:02B9:0,
#ET=20:02.7 PT=1:14.8 IC=2.1
#END SESSION 2842 ET=25:42.1 PT=1:14.8 IC=2.1
#USER = GBDIN CHARGE = 212712091X 17:55:15 05/14/85
```

SCHOUT/S/OUTROS (05/14/85)

#SCHEDULE #00030 COMPLETED
SESSION(S): 2939.

#B6800:1374 CANDE 32.230; YOU ARE SCHED#000(73) ; 0 SESSIONS QUEUED
#BEM VINDO AO SISTEMA DE TIME-SHARING DO INFE *** NEWS DE 24/04/85
*URGENTE

0 PLANTAO DE ATENDIMENTO AOS USUARIOS

08:15 11:45

13:45 17:15

A 366

#FAMILY DISK = PACK OTHERWISE DISK.

#SESSION 2939 17:55:23 05/14/85

R REDUCE

#RUNNING 2940

#?

OPTIONS SET:

PAIR SPACE = 14 PAGES

REDUCE 2 (APR-15-79) ...

> FALA:=SIM;

FALA := SIM

> IN LAPLACE;

> TL(E**(B*X)-E**(A*X),L,X,S);

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DE TRANSLACAO APLICADA

*** PROPRIEDADE DE TRANSLACAO APLICADA

$$(- A + B) / (A * B - A * S - B * S + S^2)$$

> TL(B*E**(B*X)-A*E**(A*X),L,X,S);

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DE TRANSLACAO APLICADA

*** PROPRIEDADE DE TRANSLACAO APLICADA

$$(S * (- A + B)) / (A * B - A * S - B * S + S^2)$$

> TL(SIN(A*X)-A*X*COS(A*X),L,X,S)?

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$(2*A^3)/(A^4 + 2*A^2*S^2 + S^4)$$

> TL(SIN(A*X)+A*X*COS(A*X),L,X,S)?

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$(2*A^2*S^4)/(A^2 + 2*A*S^2 + S^4)$$

> TL(COS(A*X)-A*X*SIN(A*X)/2,L,X,S)?

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$S^3/(A^4 + 2*A^2*S^2 + S^4)$$

> TL(A*X*COSSH(A*X)-SINH(A*X),L,X,S)?

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$(2*A^3)/(A^4 - 2*A^2*S^2 + S^4)$$

> TL(SINH(A*X)+A*X*COSSH(A*X),L,X,S)?

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$(2*A*S^2)/(A^4 - 2*A^2*S^2 + S^4)$$

> TL(COSH(A*X)+A*X*SINH(A*X)/2,L,X,S);

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$S^3/(A^4 - 2*A^2*S^2 + S^4)$$

> TL((3-A**2*X**2)*SIN(A*X)-A*X**2*COS(A*X),L,X,S);

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$(A*(5*A^4 + 6*A^2*S^2 + 3*S^4 - 2*S^3))/(A^6 + 3*A^4*S^2 + 3*A^2*S^4 + S^6)$$

> TL(X*SIN(A*X)-A*X**2*COS(A*X),L,X,S);

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$(8*A^3*S*(A^2 + S^2))/(A^6 + 4*A^4*S^2 + 6*A^2*S^4 + 4*A^2*S^6 + S^6)$$

> TL((1+A**2*X**2)*SIN(A*X)-A*X*COS(A*X),L,X,S);

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$(6*A*S^3)/(A^3 + 3*A*S^2 + 3*A*S^2 + S^3)$$

> TL(3*X*SIN(A*X)+A*X**2*COS(A*X),L,X,S)?

*** LINEARIDADE APLICADA PARA SOMA

*** LINEARIDADE APLICADA PARA PRODUTOS

*** LINEARIDADE APLICADA PARA PRODUTOS

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

$$(8*A*S^3*(A^2 + S^2))/(A^6 + 4*A*S^2 + 6*A*S^4 + 4*A*S^2 + S^6)$$

> END)

ENTERING LISP...

S??2??2? LISP (4/1/79)

EVAL:

> (QUIT)

#?

END OF FILE = STANDARD INPUT DEVICE

TOTAL TIME = 56.07, GC TIME = 28.18, 50.26%

I = 1792, T = 256, E = 256, Q = 2573

#ET=3:54.5 PT=58.9 IO=2.0

#END SESSION 2939 ET=6:01.6 PT=58.9 IO=2.0

#USER = GEDIN CHARGE = 212712091X 18:01:25 05/14/85

SCHCUT/S/ERROS (05/14/85)

#SCHEDULE #00040 COMPLETED
SESSION(S): 3734.

#00800:1374 CARDE 32.230; YOU ARE SCHED#000(73) ; 0 SESSIONS QUEUED
#BEP VINDO AO SISTEMA DE TIME-SHARING DO INFE *** NEWS DE 24/04/85
*URGENTE

???? ? ? ?????????????? ? ? ???? 11 ? ? ?????? ? ? ???????????
O PLANTAO DE ATENDIMENTO AOS USUARIOS ??? ?????????? ??????????
?? 8:15 ?? 11:45 ????? ?
?? 13:45 ?? 17:15 ?????.
A ?????????????? ?????? ?? ?????????? ??? ??????????? ????? ??? ??-
?????????? ? ? ?????????? ?? ??????? ???? ?????????? 366.

#FAMILY DISK = PACK OTHERWISE DISK.
#SESSION 3734 21:32:42 05/14/85
R REDUCE
#RUNNING 3735
#?
OPTIONS SET:
PAIR SPACE = 14 PAGES
REDUCE 2 (APR-15-79) ...

> OFF EXP;
> FALA:=SIM;

FALA := SIM

> OPERATOR F;
> IN LAPLACE;

> TL(E**((A+B)*X)*SIN(J*X),L,X,S);

*** LINEARIDADE APLICADA PARA PRODUTOS
*** PROPRIEDADE DE TRANSLACAO APLICADA

$$J/(J^2 + (-A - E + S)^2)$$

> TL(X**2*F(X),L,X,S);

*** LINEARIDADE APLICADA PARA PRODUTOS
*** PROPRIEDADE DO PRODUTO POR POTENCIAS DA VARIÁVEL APLICADA

DIF(TLA(F(X)),S,2)

> TL(3*F(X)+C*F(X)*E**((3*X)-F(X)/2),L,X,S);

*** LINEARIDADE APLICADA PARA SOMA
*** LINEARIDADE APLICADA PARA PRODUTOS
*** LINEARIDADE APLICADA PARA PRODUTOS
*** PROPRIEDADE DE TRANSLACAO APLICADA

```
(2*C*TLA(F(X - 3)) + 5*TLA(F(X)))/2
> TL(3+X+SIN(X),F,X,S);
*** LINEARIDADE APLICADA PARA SOMA
*** LINEARIDADE APLICADA PARA PRODUTOS
*** LINEARIDADE APLICADA PARA PRODUTOS
*** LINEARIDADE APLICADA PARA PRODUTOS
3*F(1) + F(SIN(X)) + F(X)
> TL(E**((3*X)*F(X)),L,X,S);
*** LINEARIDADE APLICADA PARA PRODUTOS
*** PROPRIEDADE DE TRANSLACAO APLICADA
TLA(F(X - 3))
> TL(1/(X-1)**2,L,X,S);
*** IMPOSSIVEL PARA DIVISOES POR FUNCOES DA VARIAVEL
      2
TLA(1/(X - 1) )
> TL(S,L1,X,S);
DEPENDENTE
> END;
ENTERING LISP...
S??????? LISP (4/1/79)
EVAL:
> (QUIT)
#?
END OF FILE - STANDARD INPUT DEVICE
TOTAL TIME = 23.17, GC TIME = 10.14, 43.75%
I = 1792, T = 256, E = 256, Q = 2573
#ET=56.5 PT=26.1 IO=2.0
#END SESSION 3734 ET=1:03.1 PT=26.1 IO=2.0
#USER = GBDIN CHARGE = 212712091X 21:33:45 05/14/85
```

APÊNDICE B

LISTAGEM DOS PROCEDIMENTOS

Neste apêndice apresenta-se a listagem com comentários dos procedimentos que compõem o sistema.

Todos os procedimentos do sistema iniciam com as letras "LP" e têm seis letras, isto evita que o usuário utilize inadvertidamente um nome de variável conflitante com os nomes de procedimentos do sistema.

LP/SIST (05/14/85)

COMMENT

```
*****
*
*
*   ARQUIVO      : LP/SIST
*
*   RESUMC      : PROCEDURES PARA DETERMINACAO DE TRANSFORMADAS
*                 DE LAPLACE.
*
*   PROGRAMADOR : FFEDERICO LUIZ GONCALVES DE FREITAS
*
*   DATA       : 3 MAIO 1985
*
*   LINGUAGEM   : MODULO SIMBOLICO
*
*****
;
```

COMMENT

```
*
*   PROCEDURE : LPFALA
*
*   PARAMETRO : S - MENSAGEM A SER IMPRESSA
*
*   RESUMC    : IMPRIMIR MENSAGENS DE ACOMPANHAMENTO.
*
;
```

PROCEDURE LPFALA(S);

```
IF FALA = 'SIM' THEN
WRITE '*** 'S;
```

SYMBOLIC;

COMMENT

```
*
*   PROCEDURE : LPLIS1
*
*   PARAMETROS : X - QUALQUER
*
*   RESUMC     : COLOCA UM PARENTESE SE FOR ATOMO
*
;
```

```
LISP PROCEDURE LPLIS1(X);
IF ATOM(X) THEN LIST(X)
ELSE X;
```

COMMENT

```
*
*   PROCEDURE : LPLIS2
*
*   PARAMETROS : X - QUALQUER
*
*   RESUMC     : COLOCA MAIS UM PARENTESE SE A LISTA FOR MAIOR
*                 QUE UM ELEMENTO.
*
```

```
;  
LISP PROCEDURE LPLIS2(X);  
IF GREATERP(LENGTH(X),1) THEN LIST(X)  
ELSE X;
```

COMMENT

```
*  
* PROCEDURE : LPDIFE  
*  
* PARAMETROS : U - FUNCAO A SER AVALIADA  
*  
* RESUMO : APLICA PROPRIEDADE DA DERIVADA SE A TRANSFORMADA ESTIVER  
* NA TABELA - CASO CONTRARIO DEVOLVE DIFEXTLA(U)).  
*  
;
```

```
LISP PROCEDURE LPDIFE(U);  
IF LPFUNC(U,'TLA) THEN  
LIST('TIMES,EXPT(-1,NUMER),LIST('DIF,U,S,NUMER))  
ELSE LIST('TIMES,EXPT(-1,NUMER),LIST('DF,U,S,NUMER));
```

COMMENT

```
*  
* PROCEDURE : LPTIRA  
*  
* PARAMETROS : X - FATOR A SER AVALIADO  
*  
* RESUMO : AUXILIAR NO TRATAMENTO DE PRODUTOS.  
*  
;
```

```
LISP PROCEDURE LPTIRAX(X);  
IF AND(NDT(ATOM(X)),EQUAL(CAR X,'TIMES)) THEN  
IF NULL(CDR X) THEN NIL  
ELSE IF EQUAL(LENGTH(CDR X),1) THEN CADR X  
ELSE X  
ELSE X;
```

COMMENT

```
*  
* PROCEDURE : LPNIL1  
*  
* PARAMETROS : X - EXPRESSAO A SER AVALIADA  
*  
* RESUMO : CASO NADA SOBRE DAS DELECCES RESULTANTES DE LPCOME E  
* LPTNIL , TROCA NIL POR 1 PARA QUE SEJA ACHADA L(1).  
*  
;
```

```
LISP PROCEDURE LPNIL1(X);  
IF NULL(X) THEN '1  
ELSE X;
```

COMMENT

```
*  
* PROCEDURE : LPTIRR  
*  
* PARAMETROS : X - QUALQUER  
*  
* RESUMO : TIRA O "TIMES" DE UM PRODUTO.  
*  
;
```

```
LISP PROCEDURE LPTIRR(X);
```

```
IF EQUAL(CAR X, 'TIMES) THEN X
ELSE IF CDR X THEN LIST(X)
ELSE X;
```

COMMENT

```
*
* PROCEDURE : LPFUNC
*
* PARAMETROS : L - FUNCAO QUALQUER
*             A - VARIAVEL QUALQUER
*
* RESUMO     : VERIFICA SE A FUNCAO "L" DEPENDE DA VARIAVEL "A".
*
* INVOCA     : LPFUNC
*
*
*
LISP PROCEDURE LPFUNC(L,A);
IF EQUAL(L,A) THEN T
ELSE IF ATOM(L) THEN NIL
ELSE OR(LPFUNC(CAR L,A),LPFUNC(CDR L,A));
```

COMMENT

```
*
* PROCEDURE : LPAPEN
*
* PARAMETROS : X,Y - QUAISQUER
*
* RESUMO     : APPEND ESPECIAL , QUE SEMPRE DEVOLVE UMA RESPOSTA.
*
* INVOCA     : LPLIS1
*             LPLIS2
*
*
*
LISP PROCEDURE LPAPEN(X,Y);
IF X THEN APPEND(LPLIS1(X),LPLIS2(LPLIS1(Y)))
ELSE LPLIS1(Y);
```

COMMENT

```
*
* PROCEDURE : LPCOME
*
* PARAMETROS : X - FATOR A SER AVALIADO
*
* RESUMO     : CHAMA LPVEJA ATEH SER ENCONTRADA A ESTRUTURA E**(A*T)
*             OU CONSTATADA SUA INEXISTENCIA.
*
* INVOCA     : LPAQUI
*             LPVEJA
*
*
*
LISP PROCEDURE LPCOME(X);
IF ATOM(X) THEN NIL
ELSE IF EQUAL(CAR X, 'TIMES) THEN LPAQUI(CDR X)
ELSE LPVEJA(X);
```

COMMENT

```
*
* PROCEDURE : LPAQUI
*
* PARAMETROS : X - FATOR A SER AVALIADO
```

```
*
* RESUMC      : USADA PELA ROTINA LPVEJA NO TRATAMENTO DE PRODUTOS.
*
* INVCCA      : LPAGUI
*              LPVEJA
*
;
```

```
LISP PROCEDURE LPAQUI(X);
IF AND(NULL(LPVEJA(CAR X)),CDR X) THEN LPAQUI(CDR X);
```

COMMENT

```
*
* PROCEDURE   : LPVEJA
*
* PARAMETROS  : X - FATOR A SER AVALIADO
*
* RESUMO      : RECONHECE E DELETA A ESTRUTURA E**(A*T) SE EXISTIR.
*
* INVOCA      : LPTIRA
*              LPTIRR
*              LPFALA
*
;
```

```
LISP PROCEDURE LPVEJA(X);
BEGIN
IF AND(NOT(ATOM(X)),EQUAL(CAR X,'EXPT'),EQUAL(CADR X,'E')) THEN
  IF EQUAL(K,CADDR X) THEN A:='1
  ELSE IF AND(NOT(ATOM(CDDR X)),NOT(ATOM(CADDR X)),
    EQUAL(CAADDR X,'TIMES'),MEMBER(K,CADDR X)) THEN
    A:=LPTIRAC(DELETE(K,CADDR X));
IF A THEN
BEGIN
  YY:=DELETE(X,LPTIRR(YY));
  LPFALAC("PROPRIIDADE DE TRANSLACAO APLICADA");
END;
RETURN A;
END;
```

COMMENT

```
*
* PROCEDURE   : LPTNIL
*
* PARAMETROS  : X - FATOR A SER AVALIADO
*
* RESUMO      : O MESMO QUE LPCOME PARA A ESTRUTURA T**N.
*
* INVOCA      : LPVERF
*              LPSAIA
*
;
```

```
LISP PROCEDURE LPTNIL(X);
IF AND(NOT(ATOM(X)),EQUAL(CAR X,'TIMES')) THEN LPSAIA(CDR X)
ELSE LPVERF(X);
```

COMMENT

```
*
* PROCEDURE   : LPSAIA
*
* PARAMETROS  : X - FATOR A SER AVALIADO
*
* RESUMO      : USADA POR LPTNIL PARA O TRATAMENTO DE PRODUTOS.
```

```
*
*   INVOCA      : LPVERF
*               LPSAIA
*
LISP PROCEDURE LPSAIA(X);
IF AND(NULL(LPVERF(CAR X)),CDR X) THEN LPSAIA(CDR X);

COMMENT
*
*   PROCEDURE  : LPVERF
*
*   PARAMETROS : X - FATOR A SER AVALIADO
*
*   RESUMO     : RECONHECE E DELETA A ESTRUTURA T**N SE EXISTIR.
*
*
LISP PROCEDURE LPVERF(X);
BEGIN
IF EQUAL(X,K) THEN NUMER:=*1
ELSE IF AND(NOT(ATOM(X)),EQUAL(CAR X,'EXPT'),EQUAL(CADR X,K),
NUMBERP(CADDR X),NOT(MINUSP(CADDR X))) THEN NUMER:=CADDR X;
IF NUMER THEN
BEGIN
  YY:=LPNIL1(LPTIRA(DELETE(X,LPTIRR(LPLIS1(YY)))));
  LPFALAC("PROPRIIDADE DO PRODUTO POR POTENCIAS DA VARIABEL APLICADA");
END;
RETURN NUMER;
END;

COMMENT
*
*   PROCEDURE  : LPSEPR
*
*   PARAMETROS : F - LISTA DE FATORES
*
*   RESUMO     : SEPARA OS FATORES DEPENDENTES DE "K".
*
*
LISP PROCEDURE LPSEPR(F);
BEGIN
IF LPFUNC(CAR F,K) THEN
LISTA1:=LPAPEN(LISTA1,LPLIS2(CAR F))
ELSE
LISTA2:=LPAPEN(LISTA2,LPLIS2(CAR F));
IF CDR F THEN
LPSEPR(CDR F);
END;

COMMENT
*
*   PROCEDURE  : LPINSE
*
*   PARAMETROS : X - FUNCAO A SER TRATADA
*
*   RESUMO     : SUBSTITUI F(X) POR L(F(X)) .
*
*   INVOCA     : LPSEPR
*               LPTIRA
*               LPAPEN
*               LPLIS1
```



```
SCALAR Y;
IF ATOM(X) THEN Y:=LPAVAL(X)
ELSE
IF EQUAL(CAR X,'MINUS) THEN
BEGIN
LPFALAC("SINAL NEGATIVO");
Y:=APPEND(LIST('MINUS),LIST(LPLINE(CADR X)));
END ELSE
IF EQUAL(CAR X,'PLUS) THEN
BEGIN
LPFALAC("LINEARIDADE APLICADA PARA SOMA");
Y:=APPEND('PLUS),MAPCAR(CDR X,'LPAVAL));
END
ELSE Y:=LPAVAL(X);
RETURN Y;
END;
```

COMMENT

```
*
* PROCEDURE : L
*
* PARAMETROS : F - FUNCAO A SER TRANSFORMADA
*
* RESUMO : APLICA AS PROPRIEDADES DE TRANSLACAO E POTENCIA DA
* VARIAVEL, PROCURANDO NA TABELA O QUE RESTOU DAS DELECOES
* RECORRENTES DA APLICACAO DESTAS PROPRIEDADES
* E REALIZANDO AS SUBSTITUICOES NECESSARIAS.
*
* INVOCA : LPCOME
* LPTNIL
* LPNIL1
* LPTIRA
* LPFUNC
* LPDIFE
*
*
;
```

```
LISP PROCEDURE L(F);
BEGIN
SCALAR A,NUMER,V,YY;
V:=1;
YY:=F;
LPCOME(YY);
LPTNIL(YY);
YY:=LPNIL1(LPTIRA(YY));
IF AND(NOT(ATOM(YY)),NOT(ATOM(CDR YY)),
NOT(ATOM(CADR YY)),EQUAL(CADR YY,'TIMES)) THEN
V:=LPTIRA(DELETE(K,CADR YY));
IF OR(LPFUNC(V,K),AND(NOT(ATOM(YY)),EQUAL(CAR YY,'TIMES))) THEN
YY:=LIST('TLA,YY)
ELSE
BEGIN
IF NOT(EQUAL(YY,'1)) THEN YY:=LIST(CAR YY,LIST('TIMES,'V,'K));
IF ASSOC(YY,PROCLA) THEN
YY:=SUBST(S,'S,SUBST(K,'K,SUBST(V,'V,CADR(ASSOC(YY,PROCLA)))));
ELSE YY:=LIST('TLA,SUBST(K,'K,SUBST(V,'V,YY)));
IF A THEN YY:=SUBST(LIST('DIFFERENCE,S,A),S,YY);
IF NUMER THEN YY:=LPDIFE(YY);
END;
RETURN YY;
END;
```

```
COMMENT
*
* PROCEDURE : LI
*
* PARAMETROS : F - FUNCAO A SER TRANSFORMADA
*
* RESUMO : REALIZA A TRANSFORMADA INVERSA DE UMA FUNCAO SIMPLES.
*
*
;
LISP PROCEDURE LI(F);
WRITE "*** FUNCAO AINDA NAO IMPLEMENTADA...";
```

```
COMMENT
*
* PROCEDURE : TL
*
* PARAMETROS : F - FUNCAO A SER TRANSFORMADA
*             H - DIRECAO DA TRANSFORMAÇÃO
*             (L - DIRETA, LI - INVERSA)
*             K - VARIÁVEL DE "F"
*             S - VARIÁVEL DA TRANSFORMADA
*
* RESUMO : REALIZA A TRANSFORMADA DE LAPLACE DE "F".
*
* INVOCA : LPLINE
*         LPAPEN
*         LPFUNC
*         LPLIS2
*         LPFALA
*
*
;
LISP PROCEDURE TL(F,H,K,S);
BEGIN
SCALAR Y,PROCURA;
PROCURA:="(1 (QUOTIENT 1 S))
          ((SIN(TIMES V K)) (QUOTIENT V (PLUS (EXPT S 2) (EXPT V 2))))
          ((COS(TIMES V K)) (QUOTIENT S (PLUS (EXPT S 2) (EXPT V 2))))
          ((SINH(TIMES V K)) (QUOTIENT V (DIFFERENCE (EXPT S 2) (EXPT V 2))))
          ((COSH(TIMES V K)) (QUOTIENT S (DIFFERENCE (EXPT S 2) (EXPT V 2)))));
IF CR(LPFLAC(K,S),LPFUNC(F,S)) THEN Y:="DEPENDENTE
ELSE
IF ATOM(F) THEN Y:=LPLINE(F)
ELSE
IF EQUAL(CAR F,'QUOTIENT) THEN
  IF LPFUNC(CDDR F,K) THEN
    BEGIN
      Y:=CONS('TLA',LIST(F));
      LPFALA("IMPOSSIVEL PARA DIVISORES POR FUNCOES DA VARIÁVEL ");
    END ELSE
      Y:=LIST('QUOTIENT',LPLINE(CADR F),CADDR F) ELSE
      Y:=LPLINE(F);
IF NOT(LPFUNC(Y,'TLA)) THEN Y:=AEVAL(Y);
RETURN Y;
END;

OPERATOR TL,L,LI;

ALGEBRAIC;
```

```
OPERATOR TLA, SINH, COSH, DIF;  
END;
```