

AGOSTO/2007

ATRAC.TXT

1 Introdução

Num <campo de luta>, CLuta, com N dimensões existem instaladas várias entidades, ENTg pertencentes a um conjunto JENT dado.

As entidades ENTg criam, em CLuta, campos atractivos e reulsivos, AtRe, e são mais ou menos sensiveis aos campos produzidos pelas restantes entidades.

Os actos executados por qualquer ENTg resulta duma opeação construida com uma razoavel racionalidade sobre o comportamento passado e psente dos campos AtRe e dos decorrentes comportamentos passados .

2 Campo de Luta, CLuta.

Formalmente o campo de luta é descrito por um domínio num espaço linear com N dimensões .

Em geral N é ou 2 ou 3 . (Ver 5)

3 Campos AtRe

Os campos AtRe têm as seguintes características:

- (a) estão centrados no ente emissor , ENTg.
- (b) têm caracter vectorial e dependem da distância a ENTg .
- (c) o ENTg emissor é dotado da faculdade de poder variar esse campo em intensidade e orientação.
- (d) existe uma colecção de campos AtRe, ColAtRE, contendo tipos de campos com características diferentes a fim de descrever os campos mais frequentes mas é sempre possivel criar novos tipos e acrescentar à colecção.

4 Sensibilidade dos ENTg

Nem todos os campos influem igualmente os entes quando estes desempenham a função de receptores. Esta propriedade vai designar-se por "sensibilidade" , SeSi, e pode ser regulada pelo ente receptor .

5 Distâncias e Campos AtRe .

A distancia é uma funcional linear como a usada corrnemente em espaços lineares como o definido em (1).

Não é necessário prever outro tipo de funcional porque existe uma colecção de campos emissores para uma entidade emissora escolher e uma função "sensibilidade" para graduar a influência de um campo numa entidade receptora .

6 Tipos de Entidades

São entidades típicas as seguintes

MUR a localização e o campo são invariantes e sensibilidade nula. exemplos: muros e vedações.

MOV a localização pode variar, o campo é invariante mas fraco e a sensibilidade é nula .

Exemplos viaturas e outros moveis mas estacionados.

AGE entes dispondo de todas as facultades, podendo deslocar-se, praticar acções e com sensibilidade aguada.

Nota: muitos tipos podem ser construidos e dotados de propriedades diverssas e o desenvolvimento deste tema só depende da imaginação do criador do modelo.

7 Operações Correntes

program ATRAC.TXT;

< * Variaveis * >

VDIS[Coord] Distancia entre dois pontos
VDGG [Ga,Gb,Coord] Distancia entre agentes Ga e Gb
VLOC [G,Coord] Local onde está gente G
VPA [G,Coord] Parametro A na Fórmula de atracção de G
VPB [G,Coord] Parametro B na Fórmula de atracção de G
vMuro[G,Coord] Repulsão dos Muros em G
vRact[Ga,Gb,Coord] Forma_Tracção emitida por Ga recebida por Gb
rRec Record rLOC,rMuro,rPA,rPB:tLoc; rDGG,rRact:tDGG;
 rEE:integer; end;
vRec [Pont] de pREC ponteiros para Recordo Rec, pRec=^rREC

< * Funções * >

F1 MuroAT(G,E) MuroAT:=1/sqr(M99-sqr(vLOC[G,E]))
F2 MuroReact(G); Reacção_Muro a agente G
F3 DistAge(Ga,Gb); Distancia entre agentes Ga e Gb
F4 EntReact(D,A,B) Reacção $R:=(-1/D) + (A/(D+B))$ onde D=distância e
 A e B são parametros que dependem do agente e das
 coordenadas .
F5 DecidLoc Função que decide o deslocamento dum agente
 conhecendo a importância dos campos-forma .
F6 ExecLoc Função que desloca o agente segindo instruções
 de DecidLoc.

< * Operações * >

01 <Geração de Campos>
O agente, Ga, cria um <campo-forma> centrado em A, onde Ga está
localizado e cujo valor em B distante é calculado usando
F4, vRact[Ga,Gb,Coord]= EntReact(D,A,B) e onde a distancia, D, entre
A e B é calculado usando F3, D= DistAge(Ga,Gb).
O agente Gb tem acesso a essa informação, vRact[Ga,Gb,Coord] e
poderá decidir como entender.
02 <Decisor>
O agente Gb tem de tomar uma decisão que é um processo complexo que
aqui será substituida por uma decisão simplificada, DecidLoc, que
consta de :
(a) se 0< atracção <rAMin não toma qualquer decisão, com rAMin>0 .
(b) se atracção >rAMin decide aproximar-se .
(c) se atracção <0 decide afastar-se .
O deslocamento de Gb é proporcional ao valor absoluto da atracção .
Todos os campos gerados menos o gerado pelo próprio influem
cumulativamente
02 <Deslocação>
Acto de deslocar um agente por força de uma decisão tomada em 02 .
A função utilizada é F6 ExecLoc .

< * Léxico * >

1 <Campo-Forma> é um campo escalar onde a grandeza é uma <informação>
 que pode ou não levar o ente receptor a tomar uma
 decisão que pode ser a prática de uma acção .

type tst14=string[14]; tst80=string[80];

const G3=3; G4=4; G5=5; G9=9; M99=10000; cEE=3;
 cRec=50; cPec=5; NomR='ATTRACT.000';

type tCOO= array[1..G3] of real;
 tLOC= array[1..G5] of tCOO;

```

tSUC= array[1..G9] of tLOC;
tDGG= array[1..G5, 1..G5] of tCOO;
rRec= Record rLOC,rMuro,rPA,rPB:tLoc;   rDGG,rRact:tDGG;
      rEE:integer; end;
pRec:=^rREC;
apRec=array[0..cPec] of pRec;

var   vDIS: tCOO; vLOC,vMuro,vPA,vPB: tLOC;   vSUC: tSUC;
      vDGG, vRact: tDGG; vRec: apREC;
      filR: file of rREC;
      EE,Ymp,MON:integer; boo:boolean;

      (* DISCO *)
function LerREC(Nf,Np:integer):pRec;
begin boo:=ENTRE(Nf,0,cRec) and ENTRE(Np,0,cPec); if boo=True then begin
Assign(filR, NomR); reset(filR); seek(filR,Nf); read(filR,vRec[Np]^);
close(filR); LerREC:=vRec[Np]; write('LEU') end else write('NÃO LEU'); end;
procedure RegREC1(Nf,Np:integer);
begin boo:=ENTRE(Nf,0,cRec) and ENTRE(Np,0,cPec); if boo=True then begin
Assign(filR, NomR); reset(filR); seek(filR,Nf); write(filR,vRec[Np]^);
close(filR); write('Regista') end else write('NÃO Regista'); end;

      (* Funcoes *)
function MuroAT(G,E:integer):real;
var Mu:real; begin Mu:=1/sqr(M99-sqr(vLOC[G,E])); MuroAT:=Mu;
vMuro[G,E]:=Mu; MoMuro(G,E); end;

procedure MuroReact(G:integer);
var j:integer; begin for J:=1 to EE do begin
if vLOC[G,j]>0 then vLOC[G,j]:=vLOC[G,j]-SQRT(vMURO[G,j]) else
vLOC[G,j]:=vLOC[G,j]+SQRT(vMURO[G,j]) ;
writeln(vLOC[G,j]:12:6); end; end;

procedure DistAge(G1,G2:integer);
var j:integer; D:real; begin for j:=1 to EE do begin
vDIS[j]:=Sqrt(Sqr(vLOC[G1,j]-vLOC[G2,j]));
writeln('Dista=',vDIS[j]:12:9); vDGG[G1,G2,j]:=vDIS[j]; end; end;

function EntReact(D,A,B:real):real;
var R:real; begin R:=(-1/D) + (A/(D+B)); EntReact:=R; end;

```

```

program ExpE;
var H:char;
procedure GerDeCrs(H:char);
var x,y,x1,x2,x3, y1,y2,y3, U,W,Z, fx,fy, R,J,J0:real; Ji,Ri:integer;
begin x1:=1; x2:=-1; x3:=0.1;
      y1:=3; y2:=0; y3:=1; U:=-50; W:=1; Z:=0.1;
clrscr; case H of '1': J0:=20; '2': J0:=60; '3':j0:=20; end;
writeln('X= ',x1:7:3,' ',x2:7:3,' ',x3:7:3);
writeln('Y= ',y1:7:3,' ',y2:7:3,' ',y3:7:3,' U= ',U:7:3,' ',W:7:3); readln;
J:=0; While J<J0 do begin fx:=x2+x3*J; fy:=y2+y3*J;
      X:=x1/Exp(fx); Y:=y1/Exp(fy); R:=X+U*Y+W; case H of
'1': begin write(J:3,' ',X:12:3, Y:12:3, R:12:3); readln; end;
'2': begin Ri:=20-Round(10*R); Ji:=round(J); gotoXY(Ji,Ri); write(Ri); end;
'3': begin write(R:8:3); end;
end; J:=J+Z; end; end;
begin repeat writeln; write('Tabela=1 ou 3 Grafo=2 Sair=0 '); readln(H);
      if H in ['1','2','3'] then GerDeCrs(H); until H='0'; end.

```

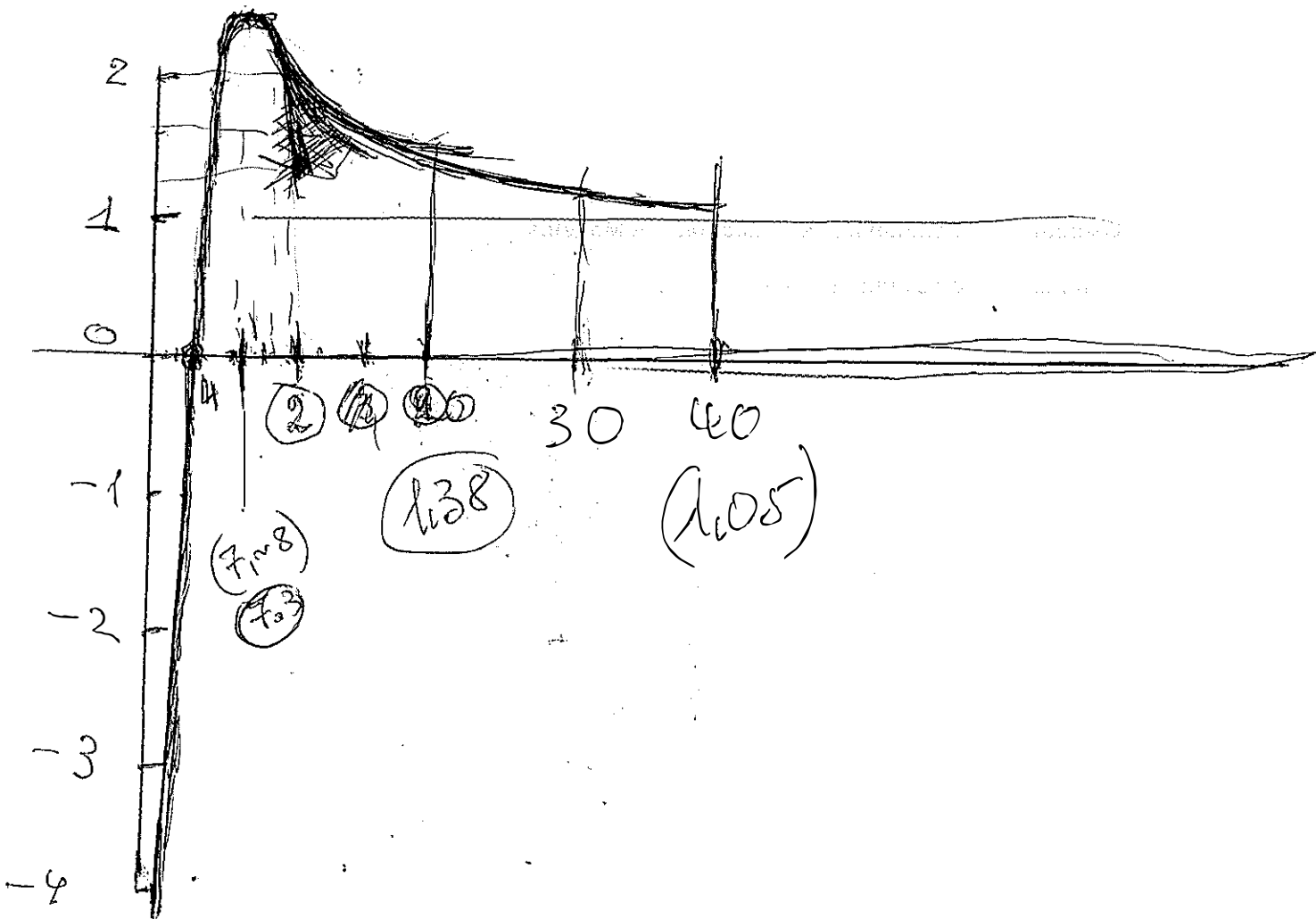
TEXTO

```

fx:= x2 + x3*j      fy:= y2 + y3*j
X := x1/Exp(fx)    Y := y1/Exp(fy)
R := X + U*Y+W

```

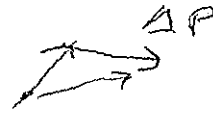
fx e fy operadores lineares da coordenada (j) , no exemplo a funcional fy identifica-se com j .
X=x1/Exp(fx) e Y=y1/Exp(fy) são inversos duma exponencial
U multiplicador de Y
W somador de Y que define o limite para que valor tende R quando j tende para infinito .
R a função é usado no domínio dos jj positivos .
o conjunto de parametros usados permite gerar uma função continua sem singularidades monotaneamente crescente no intervalo [0..7] com R(j=0)=-146, e R(j=7)=2.213, e monotaneamente decrescente para J>7 e tendo por assintota W=1.



Avanço

①

$$X + \Delta X + \Delta P$$



$$X_2 = X_1 + \Delta X_1 + \Delta P_1$$

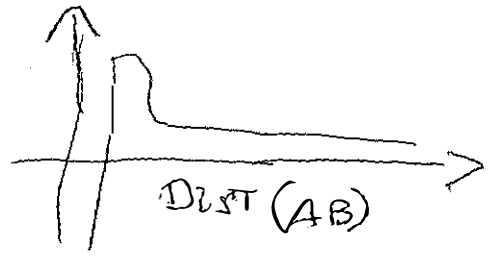
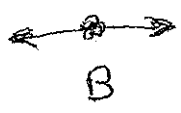
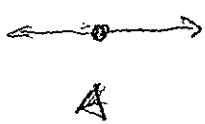
$$X_3 = X_2 + \Delta X_2 + \Delta P_2 + \dots + X_1 + \Delta X_1 + \Delta P_1 + \dots$$

$$X_{j+1} = X_1 + \sum_{k \in [1..j]} \Delta X_k + \sum_{k \in [1..j]} \Delta P_k + \sum M_{u_k}$$

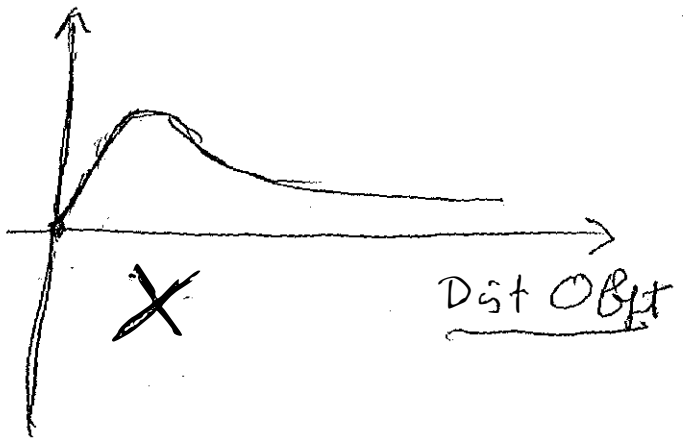
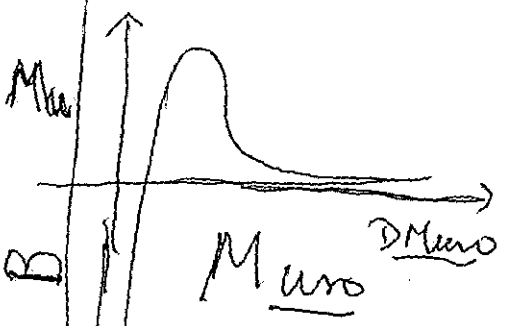
Dist(A-O)
Dist(A-B)

FBA (AB)

FAB (BA)



Problema (Aluno)



$$A_j = A_1 + \sum A_k + \sum O_k + \sum B_k + \sum M_{u_k}$$



$$A_k = A_1 + \sum A O_k + \sum A B_k + \sum M_{un}$$

O = objectivo de A que é fixo

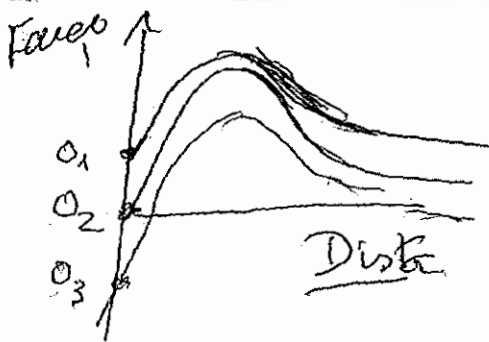
B = Atractor que é movível.

Mu = Muros e Obstáculos (fixos)

D_{AO} = Distância de A a O

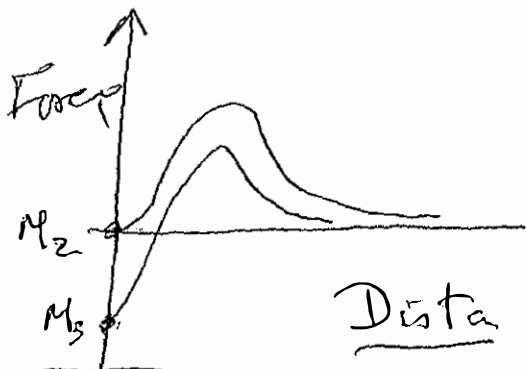
D_{AB} = " " A a B

D_{AM} = " " A aos Muros



Atracção de (A a O) e (A a B)

$\begin{cases} O_1 = \text{Contacto e Repulsão} \\ O_2 = \text{" " e Puntreat} \\ O_3 = \text{" "} \end{cases}$



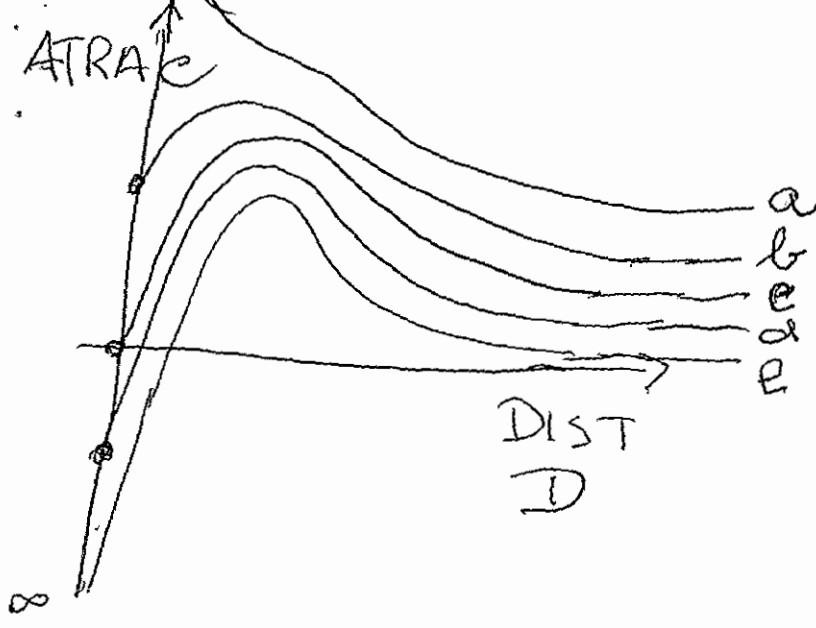
$M_2 = \text{Contacto.}$

$M_3 = \text{Contacto + Repulsão}$

$$P_k = \sum_{1..K} \Psi(D_{AO})_k \quad Y_k = \sum_{1..K} \Upsilon(D_{AB})_k$$

$$\Theta_k = \sum_{1..K} \Theta(D_{AM})_k$$

$$X_k = X_1 + P_k + Y_k + \Theta_k$$



Tipos de
Força

$t F_{ij} = F(P_i, P_j)$ = Força de atração do tipo t de dois entes (i, j) situados nos pontos P_i e P_j

$D_{ij} = D(P_i, P_j)$ = Distância dos pontos (P_i, P_j) .

Tipos de ENTES : E

- AGENTES $G_1 G_2 \dots$
- MUNDO M_u
- OBJECTIVOS $O_1 O_2 \dots$

CAMPOS e ATRAÇÃO

$(E_i, E_j) = F_{ij}(D_{ij})$ = Campo em E_j criado por E_i num ente E_j , cujo tipo é t $E_{\phi} = \text{Total}$

Dimensão DIM do espaço de duta

Coordenadas $(x_1, x_2, \dots, x_{Dim}) = \underline{\underline{C}}_k$



t Fisk (Disk) onde

$i, j \in [1..E\phi]$
 $k \in [1..Dim]$
 $t \in [a, \dots, e]$

e $[1..E\phi] = [1..AG, 1..Dulcor]$

Deslocamento DLOC

$$k X_{(i+1)} = k X_{(i)} + DLOC(t Fisk) \quad (I)$$

$([1..E\phi] - E_i)$

Seja $X = E_i$ um dos Agentes de referência

DLOC deslocamento de E_i resultando das alterações ou rupturas de todos os E_j relativos (exceto o de E_i) os que passam Agentes.

Ha que Aplicar (I) a todos os E_j que são

agentes $E.G. = [G_1, G_2, Mu, 0]$ neste caso aplica-se a G_1 e G_2 mas não a Mu e eventualmente a 0 se o alvo estiver em movimento e então 0 é tratado como um agente

t varia com o Enti para cada

Número de Agentes - N.A = 2 G_1, G_2 e G_2, G_1 } 2
 N.A = 4 $G_1, G_2, G_1, G_2, G_1, G_2$
 N.A = 3 G_1, G_2, G_1, G_2 } $6 = 3 \times 2$
 N.A = 3 $G_1, G_2, G_1, G_2, G_1, G_2$
 N.A = 3 $G_2, G_1, G_2, G_1, G_2, G_1$
 N.A = 3 G_3, G_1, G_3, G_1 }

EUROPE SOUTH AMERICA
 Pestana Sintra Golf
 Rua Mato do Mina, Nº 19 - 2710-692 Sintra - Portugal - Tel: +351 210 424 300 - Fax: +351 210 424 692
 pestana.sintra@pestana.com
 www.pestana.com
 Quinta da Beloura Golf SA - CONTRIBUINTE Nº 502 341 831 - MATRIC. CONS. REG. COM. SINTRA Nº 9370 - CAPITAL SOCIAL 4.307.420,00 EUROS

$$= \text{DIM} [P_a, P_b] [C_k] \quad [P_a, P_b] \quad \text{Pontos } P_a \text{ e } P_b \quad (2)$$

$$C_k \quad C_k \in [1..DIM]$$

D I S T

DIMS

```

For i := 1 to AG do
  For j := 1 to AG do
    For k := 1 to DIM do
      if (i <> j)
      then DIS[i, j, k] = ABS

```

$$\text{ABS}(P[i][k] - P[j][k])$$

ENTES
ambos
Movers

```

For i := 1 to AG do
  For j := 1 to AG do
    For k := 1 to DIM do
      if (i <> j) then

```

$$AT[i, j, k] = \begin{cases} \text{DIS}[i, j, k] \\ \text{FIL}[x \in \text{Linha}] \\ \text{Filtro} \end{cases}$$

ENTEFIXO [x]
ENTES Mover

o Agente é fixo então:

```

For j := 1 to AG do
  For k := 1 to DIM do

```

$$\text{AT}[x, j, k]$$

onde $[x]$ = endereço do Ponto fixo

Composições (Exemplos)

Há Muros, Mur

Há Fix, Fix

Há Abeula, $G_1 \dots G_n$
 ΦG

DYM

Diversos do
Carlo

Geradores de Campos:

Mur \Rightarrow

For $j := 1$ to ΦG
For $k := 1$ to DYM
do AT [MUR, j, k]

Fix \Rightarrow

For $j := 1$ to ΦG
For $k := 1$ to DYK
do AT [FIX, j, k]

AGENTES \Rightarrow

For $i := 1$ to $\Phi \Phi$ do
For $j := 1$ to ΦG do
For $k := 1$ to DYK do
~~AT [i, j, k]~~ +
AT [i, j, k]

```
program ExpxE;
```

```
type   tst14 = string[14];   tst80 = string[80];
const  NomA = 'EXP.001';     NomP = 'EXP.P01';
       CRR=1000;   CPP=20;   CAA=50;
type   TRR = array[0..CRR] of real;   pRR = ^tRR;
var    vRA,vRP : pRR;   fila : Text;
       x,y,x1,x2,x3, y1,y2,y3, U,W,Z, fx,fy, R,J,J0,J1,J2:real;
       HH,NN,AA,PP:integer;   Rg,M0n:char;   Ymp:boolean;
```

```
procedure RegF(RR:pRR; Nome:tst14);
  var k:integer; begin Assign(fila,Nome); rewrite(fila);
  for k:=0 to CRR do writeln(fila,RR^[k]); close (fila); end;
```

```
function LerF(RR:pRR; Nome:tst14):pRR;
  var k:integer; begin Assign(fila,Nome); reset(fila);
  K:=0; while ((not EOF(fila)) and (k<CRR)) do begin readln(fila,RR^[k]);
  k:=k+1; end; close (fila); LerF:=vRA; end;
```

```
procedure Qreg(RR:pRR; Nome:tst14);
  begin write('Registrar=1 Não=0 '); readln(Rg);
  if Rg='1' then RegF(RR, Nome); end;
```

```
procedure Qymp;
  var b:char; begin write('Imprimir=1 Não=0 '); readln(b);
  Ymp:=False; if b='1' then Ymp:=True; end;
```

```
procedure QM0n;
  begin write('Monitor 0 a 9 '); readln(M0n); end;
```

```
function DaInt(frs:tst80):integer;
  var k:integer; begin write(frs,' (inteiro) '); readln(k); DaInt:=k; end;
```

```
procedure DaLinha;
  begin write('Numero Actual NN= ',NN,' ');
  NN:=DaInt('Confirme ou Emende (0..19) '); AA:=NN*CAA; PP:=NN*CPP;
  writeln('NN= ',NN:2,' AA= ',AA:2,' PP= ',PP:2); readln; end;
```

```
procedure LimpF(RR:pRR; Nome:tst14);
  var k,k1,k2:integer; begin k1:=DaInt('Linha Inicial ');
  k2:=DaInt('LinhaFinal'); for k:=k1 to k2 do RR^[k]:=0.0; Qreg(RR,Nome); end;
```

{ Monitor }

```
procedure MonF(RR:pRR; k1,k2:integer);
  var j:integer; begin for j:=k1 to k2 do write(RR^[j]:8:3); writeln; end;
```

```
procedure MonFX(RR:pRR; k1,k2:integer);
  var j:integer; begin
  for j:=k1 to k1+2 do write(RR^[j]:8:3); writeln;
  for j:=k1+3 to k1+5 do write(RR^[j]:8:3); writeln;
  for j:=k1+6 to k1+8 do write(RR^[j]:8:3); writeln;
  for j:=k1+9 to k1+11 do write(RR^[j]:8:3); writeln; readln; end;
```

```
procedure Mon3(J,X,Y,R:real);
  var Ji,Ri:integer; begin case HH of
  1: begin write(J:3,' ',X:12:3, Y:12:3, R:12:3); readln;
  if Ymp then writeln(LST, J:3,' ',X:12:3, Y:12:3, R:12:3); end;
  2: begin Ri:=20-Round(10*R); Ji:=round(J); gotoXY(Ji,Ri); write(Ri); end;
  3: begin write(R:8:3); end; end; end;
```

```
procedure MonPar; begin clrscr;
  writeln(' X= ', x1:7:3, ' ',x2:7:3,' ',x3:7:3);
```

```
writeln(' Y= ', y1:7:3, ' ', y2:7:3, ' ', y3:7:3);
writeln(' U= ', U:7:3, ' W=', W:7:3, ' Z=', Z:7:3);
writeln(' J0= ', J0:7:3, ' J1=', J1:7:3, ' J2=', J2:7:3); end;
```

{ Darametros }

```
procedure CopPP(N1,N2:integer);
var k,p1,p2:integer; begin p1:=N1*cPP; p2:=N2*cPP; vRP:=LerF(vRP,NomP);
for k:=1 to cPP do vRP^[p2+k]:=vRP^[p1+k]; RegF(vRP,NomP); end;
```

```
procedure CopAA(N1,N2:integer);
var k,a1,a2:integer; begin a1:=N1*cAA; a2:=N2*cAA; vRA:=LerF(vRA,NomA);
for k:=0 to cAA do vRA^[a2+k]:=vRA^[a1+k]; RegF(vRA,NomA); end;
```

```
procedure CopXRP(PP,q:integer); begin case q of
0: vRP^[PP+q] :=x1; 1: vRP^[PP+q] :=x2; 2: vRP^[PP+q] :=x3;
3: vRP^[PP+q] :=y1; 4: vRP^[PP+q] :=y2; 5: vRP^[PP+q] :=y3;
6: vRP^[PP+q] :=U; 7: vRP^[PP+q] :=W; 8: vRP^[PP+q] :=Z;
9: vRP^[PP+q] :=J0; 10: vRP^[PP+q] :=J1; 11: vRP^[PP+q] :=J2; end; end;
```

```
procedure CopTudXRP(PP:integer);
var q:integer; begin for q:=0 to 11 do CopXRP(PP,q); end;
```

```
procedure CopRPX(PP,q:integer); begin case q of
0: x1:=vRP^[PP+q]; 1: x2:=vRP^[PP+q]; 2: x3:=vRP^[PP+q];
3: y1:=vRP^[PP+q]; 4: y2:=vRP^[PP+q]; 5: y3:=vRP^[PP+q];
6: U :=vRP^[PP+q]; 7: W :=vRP^[PP+q]; 8: Z :=vRP^[PP+q];
9: J0:=vRP^[PP+q]; 10: J1:=vRP^[PP+q]; 11: J2:=vRP^[PP+q]; end; end;
```

```
procedure CopTudRPX(PP:integer);
var q:integer; begin for q:=0 to 11 do begin CopRPX(PP,q);
writeln(vRP^[PP+q]:6:1 ); end; end;
```

```
procedure DaParX;
begin x1:=1; x2:=-1; x3:=0.1; y1:=3; y2:=0; y3:=1;
U :=-50; W:=1; Z:=0.1; J0:=30; J1:=49; J2:=20;
NN:=0; PP:=0; CopTudXRP(PP); readln; end;
```

```
function DaPar(Frs:tst14; v:real):real;
begin write('Valor Actual de ',Frs,v:8:3,' Confirme ou Corrija ');
Readln(v); DaPar:=v; end;
```

```
procedure DaParPP;
var q:integer; begin DaLinha; CopTudRPX(PP); MonPar; readln; repeat
writeln('( X= x1 / Exp( x2 + x3 * J ) 0::x1 1::x2 2::x3 ');
writeln('( Y= y1 / Exp( y2 + y3 * J ) 3::y1 4::y2 5::y3 ');
writeln('( Y= X + U * Y + W ) ( Z ) 6::U 7::W 8::Z ');
writeln('( 9::J0 10::J1 11::J2 ');
q:=DaInt('Escolha 0 a 11 ou Sair=0 '); case q of
0: x1:=DaPar('x1',x1); 1: x2:=DaPar('x2',x2); 2: x3:=DaPar('x3',x3);
3: y1:=DaPar('y1',y1); 4: y2:=DaPar('y2',y2); 5: y3:=DaPar('y3',y3);
6: U:=DaPar('U',U); 7: W:=DaPar('W',W); 8: Z:=DaPar('Z',Z);
9: J0:=DaPar('J0',J0); 10: J1:=DaPar('J1',J1); 11: J2:=DaPar('J2',J2);
end; until q=0; CopTudXRP(PP); end;
```

```
procedure DaParam;
var g:integer; begin repeat
g:=DaInt('<Parametros> Exemplo=1 Fila_PP=2 Sair=0 ');
if g in [1,2] then begin vRP:=LerF(vRP, NomP);
case g of 1:DaParX; 2:DaParPP; end; MonPar; readln; Qreg(vRP, NomP); end;
until g=0; end;
```

{ Geradores }

```
procedure GerFunc(JJ:real);
var Ji,Ri,k:integer;
```

```

begin J:=0; k:=0; While J<JJ do begin fx:=x2+x3*J; fy:=y2+y3*J;
X:=x1/Exp(fx); Y:=y1/Exp(fy); R:=X+U*Y+W; vRA^[k+AA]:=R;
if J<50 then Mon3(J,X,Y,R); readln; J:=J+Z; k:=k+1; end; end;

```

```

procedure PrepFunc;

```

```

var JJ:real; k1,k2:integer; begin vRA:=LerF(vRA,NomA);
vRP:=LerF(vRP,NomP); DaLinha; CopTudRPX(PP); MonFX(vRP,PP,PP+cPP); readln;
HH:=DaInt('Apresenta: Lista 1 e 2 Grafo=3 ');
case HH of 1:JJ:=vRP^[9]; 2:JJ:=vRP^[10]; 3:JJ:=vRP^[11]; end;
writeln('JJ=',JJ:5:1); readln; GerFunc(JJ); Qreg(vRA,NomA); end;

```

```

procedure Instrui;

```

```

var n1,n2:integer; begin repeat clrscr; writeln('INSTRUÇÕES':25);
writeln('          Função          Parametros');
writeln('<Dar Param>          P=20 ');
writeln('<Ler Disc>          A=30          P=31 ');
writeln('<Reg Disc>          A=35          P=36 ');
writeln('<Limpa Mem>          A=40          P=41 ');
writeln('<Copiar>          A=45          P=46 ');
writeln('<Traduz>          X>R=55          R>X=56 ');
writeln('<Gerar> Listas=1,2          Graf=3 ');
writeln('<Sair do Programa> =0 '); readln(HH); Qymp; case HH of
20: DaParam;
30: begin vRA:=LerF(vRA,NomA); Dalinha; MonFX(vRA,AA,AA+CAA); readln; end;
31: begin vRP:=LerF(vRP,NomP); Dalinha; MonF(vRP, PP,PP+cPP); readln;
MonFX(vRP,PP,PP+cPP); readln; end;
35: RegF(vRA,NomA); 36: RegF(vRP,NomP);
40: LimpF(vRA,NomA); 41: LimpF(vRP,NomP);
45: begin n1:=DaInt('NN Dador 0..19 '); n2:=DaInt('NN Recebe 0..19 ');
CopAA(n1,n2); end;
46: begin n1:=DaInt('NN Dador'); n2:=DaInt('NN Recebe'); CopPP(n1,n2); end;
55: begin end;
56: begin vRP:=LerF(vRP,NomP); DaLinha; CopTudRPX(PP); end;
1..3: begin PrepFunc; end; end;
until HH=0; end;

```

```

procedure Inic;

```

```

begin vRA:=Nil; New(vRA); vRP:=Nil; New(vRP);
vRA:=LerF(vRA,NomA); MonF(vRA,1,20); writeln;
vRP:=LerF(vRP,NomP); MonF(vRP,1,10); readln; end;

```

```

begin Inic; Instrui; end.

```

```

{I EXP.TXT}

```

```

H→

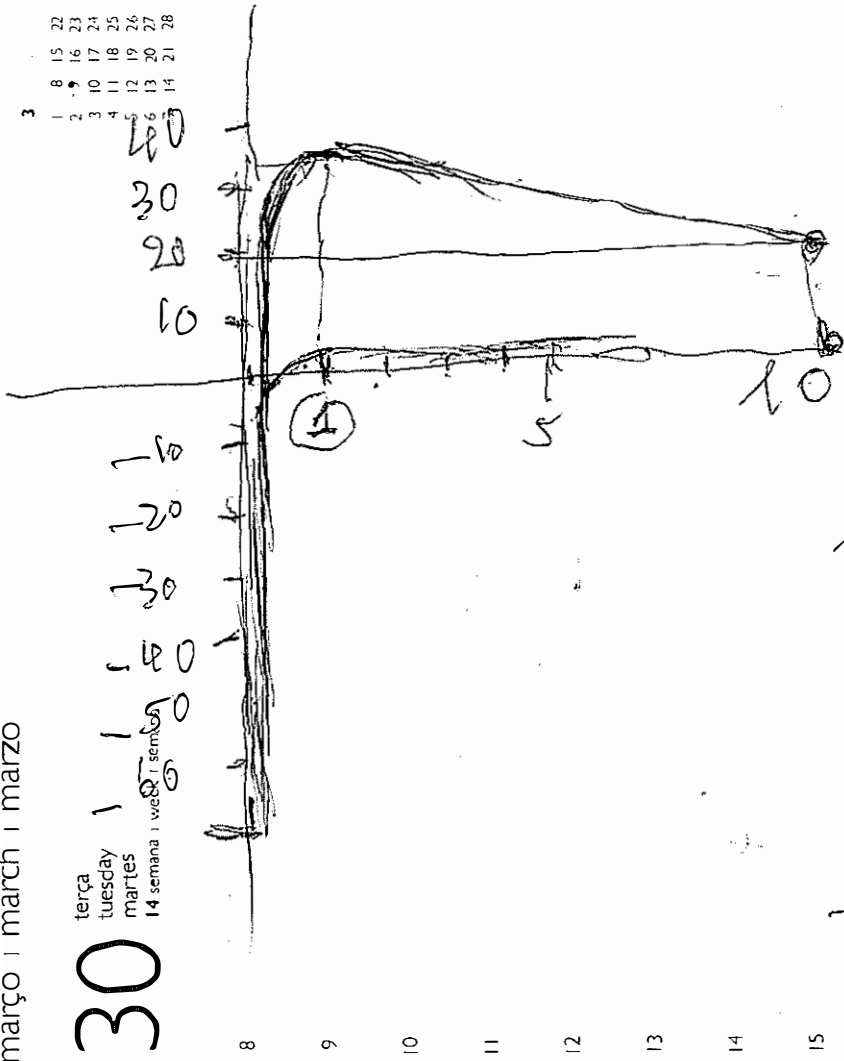
```


março | march | marzo

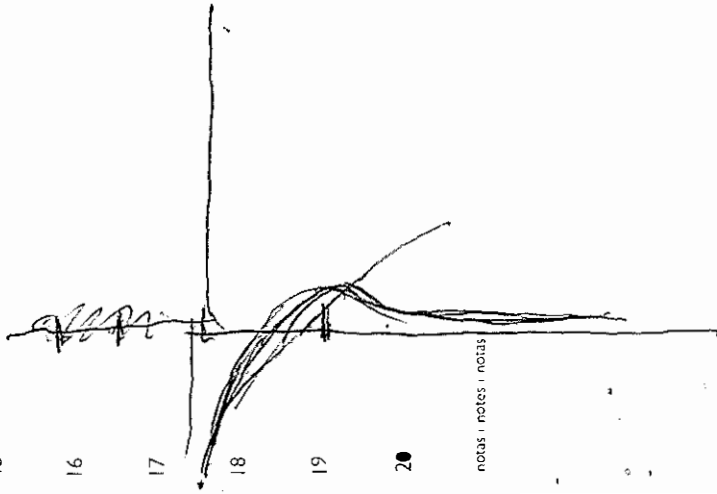
30

terça
tuesday
martes
14 semana | week | semana

1 8 15 22 29
2 9 16 23 30
3 10 17 24 31
4 11 18 25 1
5 12 19 26 2
6 13 20 27 3
7 14 21 28 4



- 03 → 99 (+)
 - 02 → 96 (+)
 - 01 → 6 (+)
 Δ → 2,6 (+)
 10 → 19 (+)
 100 → 0,3 (+)
 1000 → 0,03 (+)
 10.000 → 0,003 (+)



150A
11A

~~WHILE~~

while $U[N] \leq 9$ CAB do

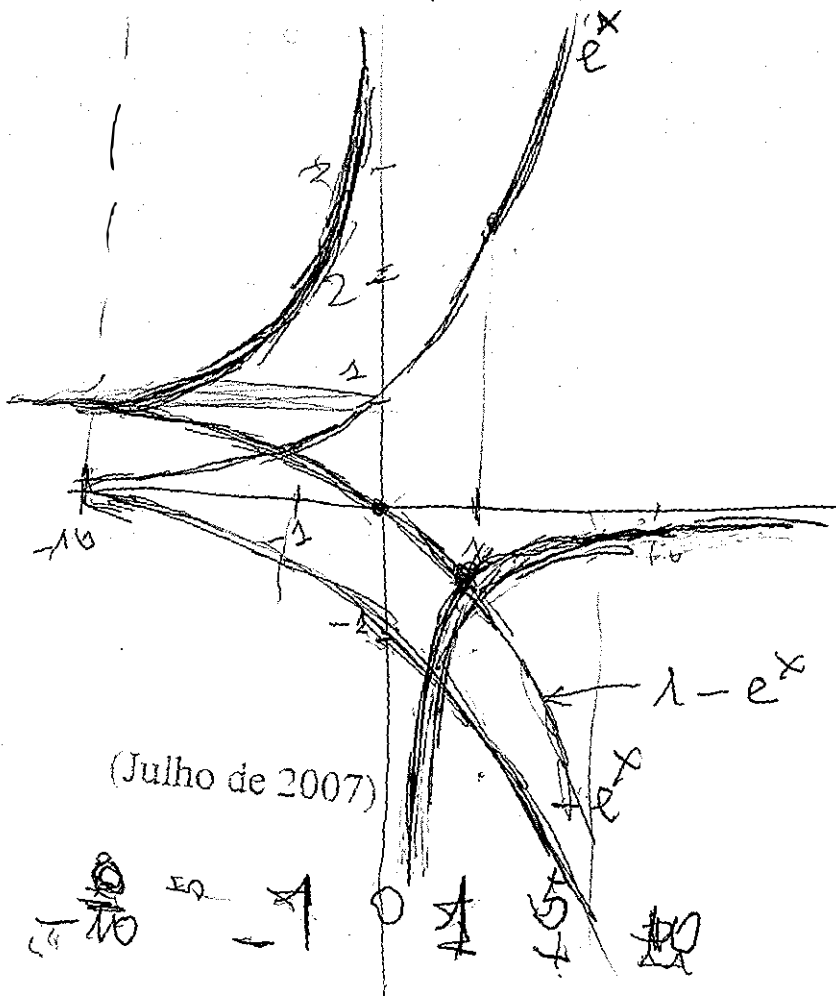
BEGIN
 $\left\{ \begin{array}{l} rU[N, U1, 1] = wC[U1] \\ rU[N, U1, 2] = wC[U2] \\ rU[N, U1, 3] = wC[U3] \\ rU[N, U1, 4] = wC[U4] \end{array} \right.$

on
for $k := 1$ to $\frac{N}{4}$ do
 $rU[N, U1, k] := wC[k]$

$VA = \frac{\text{CalcUPLo}}{\text{IPR, *}} \left(\begin{array}{l} rU[N] \\ UN, TIP \\ VA, W \end{array} \right)$
 $rU[N][U1] := VA$
 $k := k + 1; U_k := U_{k+1};$
end;

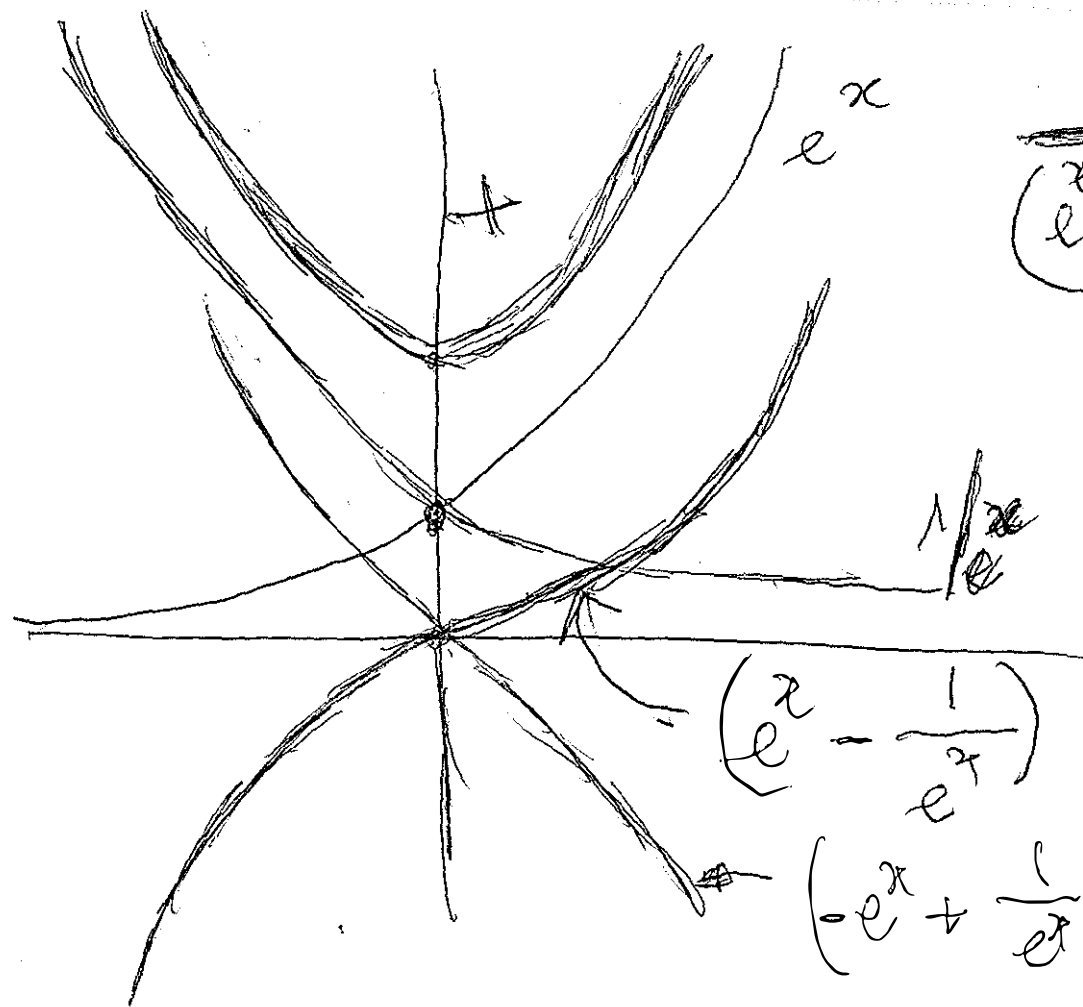
/AZTUR/PROG/Here

$$\frac{1}{1 - e^x}$$



(Julho de 2007)

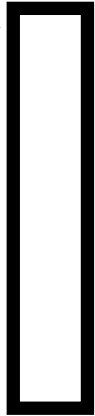
EQ - 1 0 1 5 10

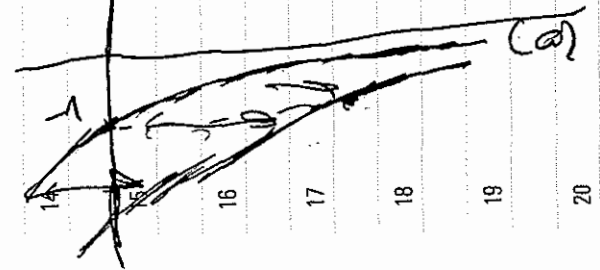
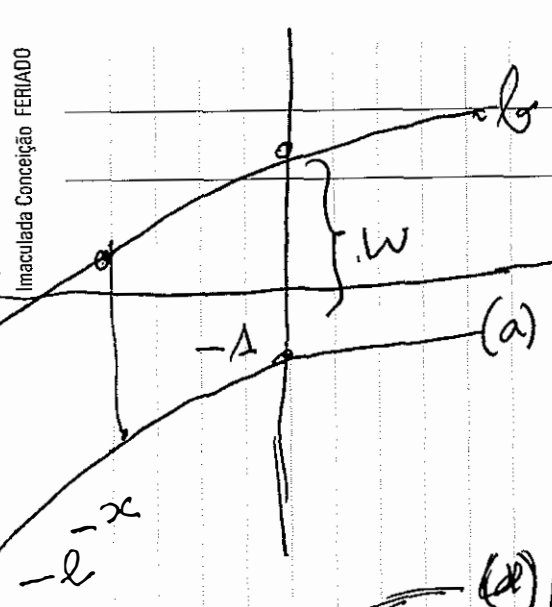
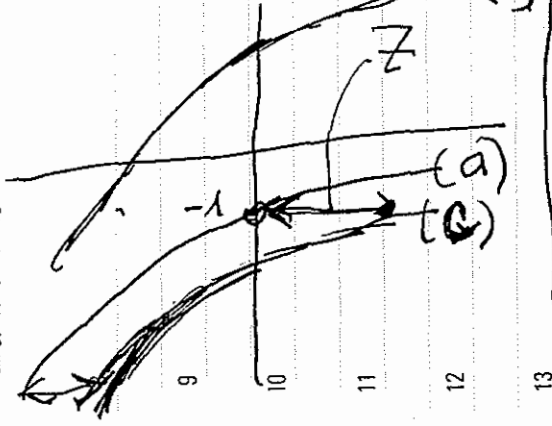


$$\overline{\left(e^x + \frac{1}{e^x} \right)}$$

$$\left(e^x - \frac{1}{e^x} \right)$$

$$\left(-e^x + \frac{1}{e^x} \right)$$





$$Y = -e^{(-x + 2z)} - w \quad (e)$$

$$Y = -e^{(-x + z)} + w \quad (d)$$

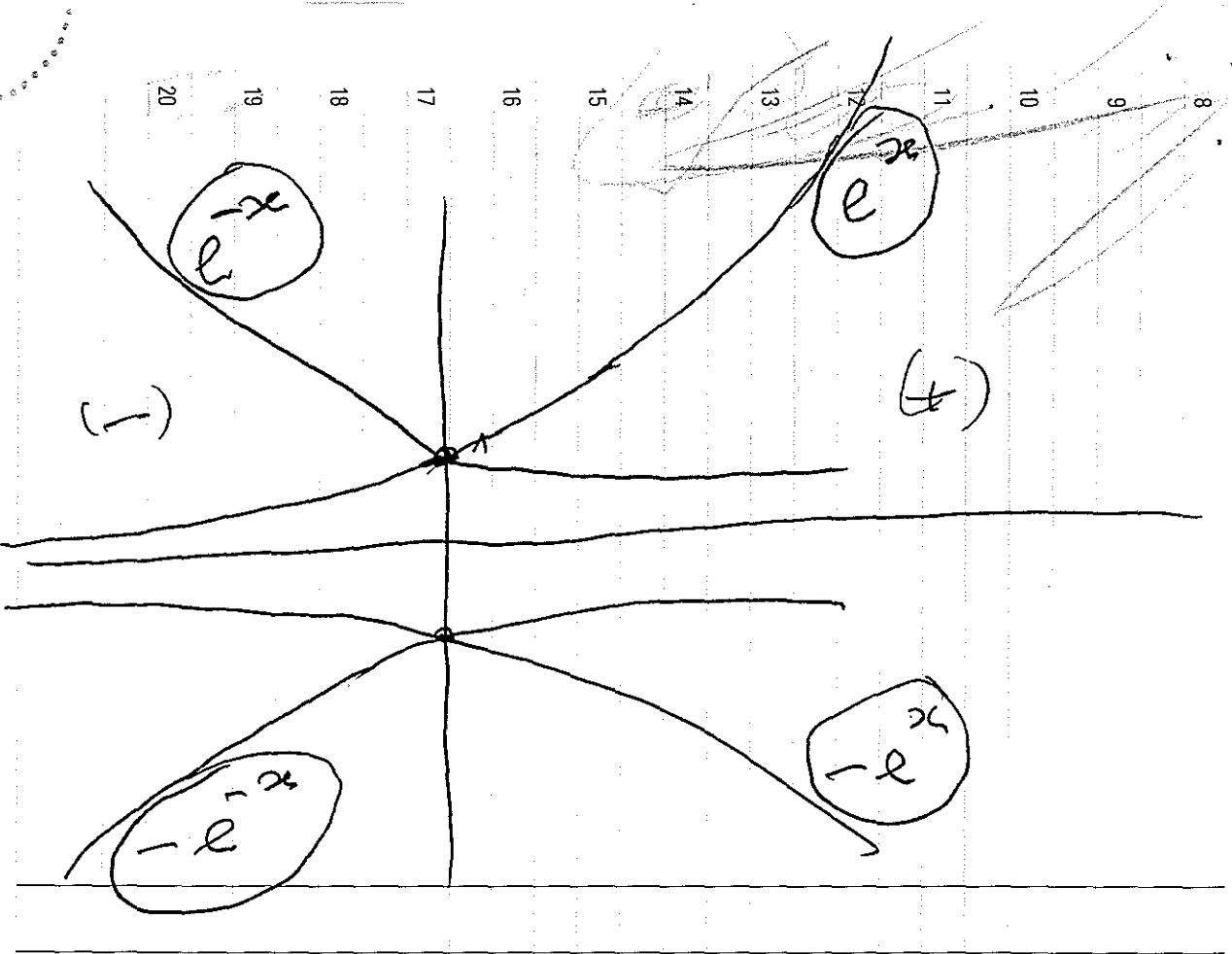
$$Y = -e^{(-x + z)} \rightarrow (c)$$

$$Y = -e^{-x} + w \rightarrow (b)$$

$$Y = -e^{-2x} \rightarrow (a)$$

SETEMBRO
15. SAB

SETEMBRO	
1	24
2	25
3	26
4	27
5	28
6	29
7	30
8	31
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	



Faep $W = 5$

$a_0 + b_0 = -895$	$(j = \phi)$
$a_0 e^a + b_0 e^b = -5$	$(j = 1)$
$a_0 e^{a.5} + b_0 e^{b.5} = 95$	$(j = 5)$
$a_0 e^{a.90} + b_0 e^{b.90} = 4$	$(j = 90)$

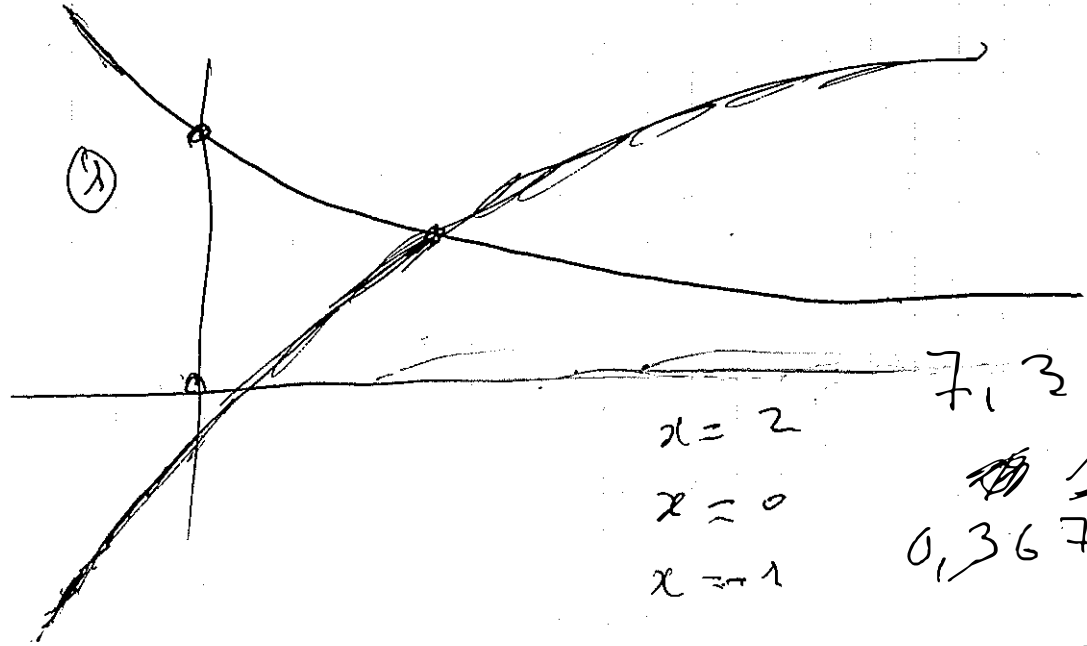
(5)

$a_0 = -b_0 - 895$



SETEMBRO	
31	0
30	1
29	2
28	3
27	4
26	5
25	6
24	7
23	8
22	9
21	10
20	11
19	12
18	13
17	14
16	15
15	16
14	17
13	18
12	19
11	20
10	21
9	22
8	23
7	24
6	25
5	26
4	27
3	28
2	29
1	30
31	31

8
9
10
11
12
13
14
15
16
17
18
19
20



$x = 2$ 7,3
 $x = 0$ ~~1~~ 1
 $x = -1$ 0,3679
 $x = 5$ 0,0067
 $x = 10$ 0,00004

→ x
y

ARBITRADO

$$\begin{cases}
 a_0 + b_0 + w = R_0 \\
 a_0 e^{a_{j1}} + b_0 e^{b_{j1}} + w = \phi \\
 a_0 e^{a_{j2}} + b_0 e^{b_{j2}} + w = R_2 \\
 a_0 e^{a_{j3}} + b_0 e^{b_{j3}} + w = R_3
 \end{cases}
 \begin{cases}
 R(\phi) < \phi \\
 R(j_1) = \phi \\
 R(j_2) > \phi \\
 R(j_2) > R(j_3) > \phi
 \end{cases}$$

Ha 5 parâmetros (a, a_0, b, b_0, w) e 4 condições

Escolhendo :



$$\begin{aligned}
 R_0 &= 900 \\
 R_1 &= 0 \\
 R_2 &= 100 \\
 R_3 &= 2
 \end{aligned}$$

$$\begin{aligned}
 j_0 &= \phi \\
 j_1 &= 5 \\
 j_2 &= 90
 \end{aligned}$$

$$\begin{cases}
 a_0 + b_0 + w = -900 \\
 a_0 e^{b_{j1}} + b_0 e^{b_{j1}} + w = \phi \\
 a_0 e^{b_{j2}} + b_0 e^{b_{j2}} + w = 100 \\
 a_0 e^{b_{j3}} + b_0 e^{b_{j3}} + w = 90
 \end{cases}$$

$$\begin{aligned}
 (j_0 = \phi) \\
 (j_1 = 5) \\
 (j_2 = 90) \\
 (j_3 = 90)
 \end{aligned}$$

Definicao de $a_0, a_1, b_0, b_1 \in W$

①

~~para~~ $R(\phi) = R_0$ $R_0 = a_0 \cdot e^{\phi} + b_0 \cdot e^{\phi} + W$ ③

$R_0 < \phi$ donde ~~$a_0 + b_0 < \phi$~~

$a_0 + b_0 + W = R_0$

Seja $W > \phi, R_0 < \phi \Rightarrow a_0 + b_0 < \phi$

②

$R(j_1) = \phi \Rightarrow a_0 \cdot e^{a_1 j_1} + b_0 \cdot e^{b_1 j_1} + W = \phi$

Seja $W > \phi$ para $a_0 + e^{a_1 j_1} + b_0 \cdot e^{b_1 j_1} < \phi$

③

$R(j_2) = R_2 = a_0 \cdot e^{a_1 j_2} + b_0 \cdot e^{b_1 j_2} + W$

④

$R(j_3) = R_3 = a_0 \cdot e^{a_1 j_3} + b_0 \cdot e^{b_1 j_3} + W$

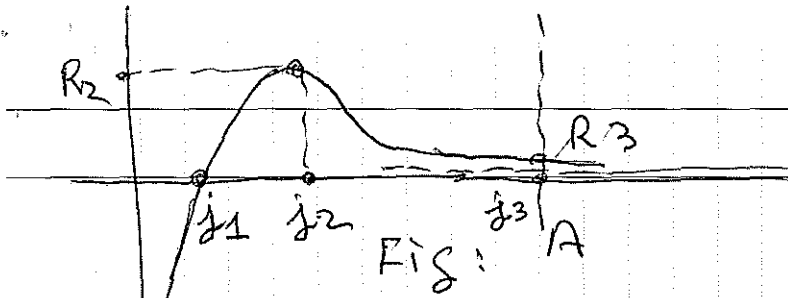
da Derivada

⑤

$R'(j) > \phi, j(\phi, j_2)$

$R'(j) < \phi, j \in [j_2, \rightarrow)$

$R'(j) = \phi, j = j_2$



(2)

10) $R(j_1) = \phi$ $\left\{ \begin{array}{l} \text{Max } R \text{ é em } j_2 \\ R(j_2) = R_2 \end{array} \right.$ $\left\{ \begin{array}{l} \text{Lim } R = R_3 \\ s \rightarrow \infty \end{array} \right.$

$R^1 = a_0 a e^{aj} + b_0 b e^{bj}$ Derivado de $R \frac{dR}{dj}$

$R^1(j_2) = \phi$ $\left\{ \begin{array}{l} R^1(j) > \phi \\ j \in (0, j_2) \end{array} \right.$ $\left\{ \begin{array}{l} R^1(j) < \phi \\ j \in (j_2, \infty) \end{array} \right.$

Par $j_3 \gg j_2$ $R(j_3) = R_3$

$R(\phi) = R_0 < \phi$

$$R = \frac{a_1}{\text{Exp}(a_2 + a_3 j)} + \frac{U \times b_1}{\text{Exp}(b_2 + b_3 j)} + W$$

(1)

$$\begin{aligned}
 R &= a_1 e^{-(a_2 + a_3 j)} + U b_1 e^{-(b_2 + b_3 j)} + W \\
 &= a_1 \cdot e^{-a_2} \cdot e^{-a_3 j} + U \cdot b_1 \cdot e^{-b_2} \cdot e^{-b_3 j} + W \\
 &= a_0 e^{a_1 j} + b_0 e^{b_1 j} + W
 \end{aligned}$$

and $a_0 = a_1 e^{-a_2}$
 $a = -a_3$

$b_0 = U \cdot b_1 \cdot e^{-b_2}$
 $b = -b_3$

$$R = a_0 e^{a_1 j} + b_0 e^{b_1 j} + W$$



Avanço

①

$$X + \Delta X + \Delta P$$



$$X_2 = X_1 + \Delta X_1 + \Delta P_1$$

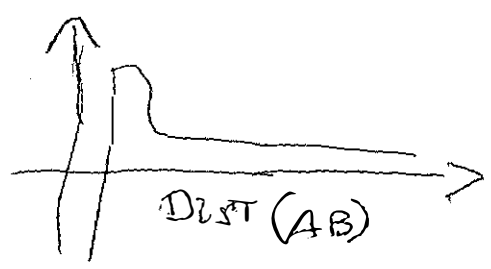
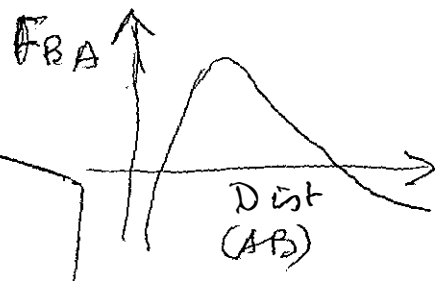
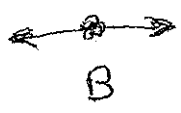
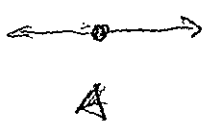
$$X_3 = X_2 + \Delta X_2 + \Delta P_2 + \dots + X_1 + \Delta X_1 + \Delta P_1 + \dots$$

$$X_{j+1} = X_1 + \sum_{k \in [1..j]} \Delta X_k + \sum_{k \in [1..j]} \Delta P_k + \sum M_{k,j}$$

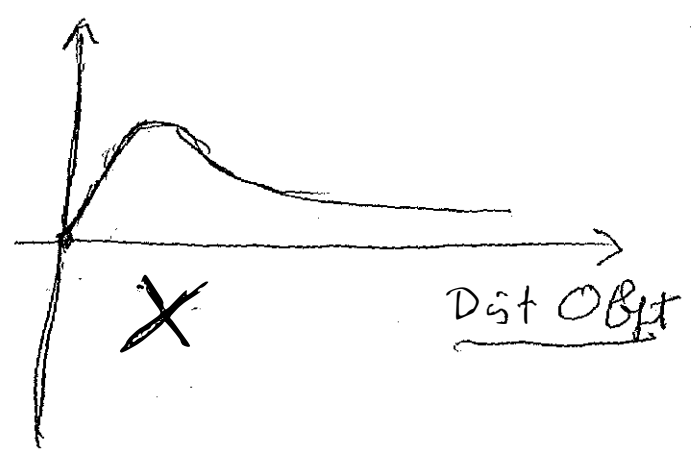
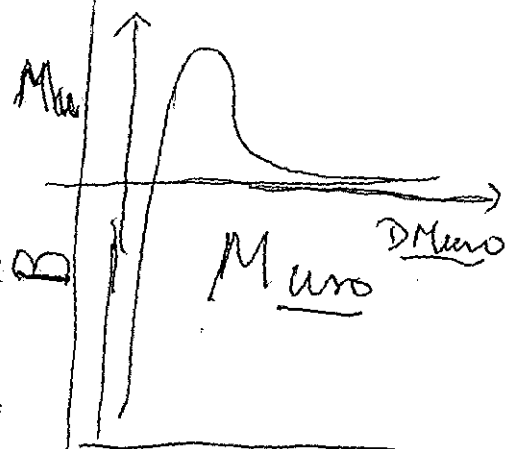
Dist (A-O)
Dist (A-B)

FBA (AB)

FAB (BA)



Problema (Aluno)



$$A_g = A_1 + \sum A_k + \sum O_k + \dots$$



$$\sum B_k + \sum M_{k,j}$$

Coordenadas $(E_1, E_2, \dots, E_{Dim}) = \underline{\underline{C^k}}$

Pestana Sintra Golf

 RESORT & SPA HOTEL
 SINTRA - PORTUGAL

t Fisk (Disk) onde

$i, j \in [1..E\phi]$
 $k \in [1..DIM]$
 $t \in [a, \dots, e]$

e $[1..E\phi] = [1..AG, 1..Dulcor]$

Deslocamento DLOC

$$k X_{(i+1)} = k X_{(i)} + DLOC(t Fisk) \quad (I)$$

([1..E\phi] - E_i)

sendo $X = E_i$ um dos Agentes de referência

DLOC deslocamento de E_i resultando das alterações ou rupturas de todos os E_j relativos (exceto o de E_i) os que passam Agentes.

Ha que Aplicar (I) a todos os E_j que são

agentes $E.G. = [G_1, G_2, Mu, 0]$ neste caso aplica-se a G_1 e G_2 mas não a Mu e eventualmente a 0 se o alvo estiver em movimento e estas 0 é tratado como um agente

t varia com o Enti para a qual

Queroses ou Agentes - N.A = 2 G_1, G_2 e G_2, G_1 } 2

NA = 4 $G_1, G_2, G_1, G_2, G_1, G_2$ } NA = 3 $G_1^A, G_2^A, G_1^A, G_2^A$ } $G = 3 \times 2$

G_2 } G_2^A, G_1, G_2, G_3 } G_2, G_1, G_3, G }

G_3 Told 4x3 } G_3, G_1, G_3, G }

G_4 } G_3, G_1, G_3, G }

EUROPE SOUTH AMERICA
 Pestana Sintra Golf
 Rua Mato do Mina, Nº 19 - 2710-692 Sintra - Portugal - Tel: +351 210 424 300 - Fax: +351 210 424 692
 pestana.sintra@pestana.com
 www.pestana.com
 Quinta da Beloura Golf SA - CONTRIBUINTE Nº 502 341 831 - MATRIC. CONS. REG. COM. SINTRA Nº 9370 - CAPITAL SOCIAL 4.307.420,00 EUROS

