

TR-AC-0045

014

動的過程のQoS分析

滝沢 航一郎（一橋大） 新上 和正

2000. 4.19

ATR環境適応通信研究所

CONTENTS

1.	始めに	3
2.	適応写像の構成	3
2.1.	ニューラルネットを使った写像	3
2.2.	力の計算	4
2.3.	ニューラルネットでの計算プログラム	5
3.	入力データの変換 (Part 1)	5
4.	入力データの変換 (Part 2)	5
5.	終りに：感想	5
6.	プログラムのソース (ニューラルネット): 付録 1	7
7.	データの変換のプログラム: 付録 2	14
8.	データの変換のプログラム: 付録 3	15

1. 始めに

後間もなくで二一世紀を迎える日本では、様々な分野において自転車、車等の交通手段やテレビ、冷蔵庫、固定電話等の家電製品を始めとする物理的またはインフラ的なモノがある程度以上整いました。そのような状況下ではモノによって人々が左右されなくなってきました。

では彼らは何に左右されているのかというと実は何にも左右されないという傾向が強くなってきました。逆に彼らの「趣向」がモノの市場を左右する時代になった、と表現してもいいかもしれません。

このことは例えば日本において固定電話の台数がPHSや携帯電話等の移動電話のそれを上回った事実や、その移動電話の利用方法においても通話形態よりもEメール等の文字メッセージの送信形態のほうが金額換算で上回っている事実からも分ります。昔は仕事や日常生活における必要性に従い固定電話を購入したけれども、今は「他人とコミュニケーションがとれるから」とか「デザインが可愛いから」といった理由で携帯電話やその他のコミュニケーション機器を購入するようになりました。

インターネットという新たなメディアが人々の間に急激に普及してくにつれ、モノを作り出す(デザインする)過程において人々のモノに対する「趣向」を注意深く考察して把握していくことが重要になってきました。デザインする人つまりデザイナーは多種多様な分野において存在します。「モノに対するデザインコンセプトを明確にしてから、そのコンセプトを具体化する方法を考案して最終的にそのコンセプトが消費者に受け入れられたかを売り上げという手段等で検証する」ヒトは皆デザイナーと言えます。ある時は洋服のデザイナーであり情報システムのデザイナーであり商品企画のデザイナーでありイベント企画のデザイナーであったりします。

ここでは、モノの動きと人側の感じ方の対応関係、より広く言えば、一見異種と思える動き(人やモノ)の対応関係を構成することを目的とする。このような例をリストすると、

- (1) タクシーの運転の仕方と運転手の評価(例えば、営業成績、お客の乗り心地、事故に繋り易さなど)、
- (2) 照明器具(照度、デザイン、配置、機能など)とお客の印象(購入するかどうかなど)、
- (3) 男性や女性の特徴(振る舞い、見かけ、など)とストーカーに狙われる度合い: 被害届けを出さないように見える人や大人しそうな人を狙うなど、また、どのような点がそのような印象を与えるかなど、
- (4) 通信ネットワーク上での新規サービスとユーザーの印象、
など。

このレポートでは、特に(1)のタクシー運転手の運転データ(入力側)と実際にその運転手の運転をどう感じるかという、つまりは人間の「趣向評価」(出力側)を用いて、その二つの間に介在する写像を検討します。そのことにより最終的には一般的に通用するデザインコンセプトの明確化の助けになります。

2. 適応写像の構成

2.1. ニュラルネットを使った写像. 入力を

$$(2.1) \quad d(p) = (d_1(p), d_1(p), d_1(p), d_1(p), d_1(p), d_{N_{in}}(p))$$

とする。 p は教師データの番号を意味する: $p = 1, \dots, N_p$

$$(2.2) \quad x_k^-(p) = \sum_{j=1}^{N_{in}} \gamma_{kj}^1(1) d_j(p) = x_k^{1-}(p).$$

$$(2.3) \quad x_k^+(p) = f(x_k^-(p)).$$

$$(2.4) \quad f(x) = \frac{1}{1 + e^{-x/T}}.$$

同様に

$$(2.5) \quad y_k^-(p) = \sum_{j=1}^{N_1} \gamma_{kj}^1(2) x_j^+(p) = y_k^{1-}(p).$$

$$(2.6) \quad y_k^+(p) = f(y_k^-(p)).$$

$$(2.7) \quad z_k^-(p) = \sum_{j=1}^{N_2} \gamma_{kj}^1(3) y_j^+(p) = z_k^{1-}(p).$$

$$(2.8) \quad z_k^+(p) = f(z_k^-(p)).$$

2.2. 力の計算. Cost function を

$$(2.9) \quad V = V_0 + V_1,$$

$$(2.10) \quad V_0 = \frac{1}{2} \sum_{p=1}^{N_p} \sum_{k=1}^{N_{out}} (z_k^+(p) - e_k(p))^2 +$$

$$(2.11) \quad V_1 = \frac{\alpha}{2} \sum_{p=1}^{N_p} \sum_{k=1}^{N_{in}} (|x_k^-(p)| - c_0)^2 + \frac{\alpha}{2} \sum_{p=1}^{N_p} \sum_{k=1}^{N_{in}} (|y_k^-(p)| - c_0)^2 + \frac{\alpha}{2} \sum_{p=1}^{N_p} \sum_{k=1}^{N_{ou}} (|z_k^-(p)| - c_0)^2.$$

とする。

$$(2.12) \quad \frac{\partial V_0}{\partial \gamma_{ij}^1(3)} = \sum_{p=1}^{N_p} \sum_{k=1}^{N_{out}} (z_k^+(p) - e_k(p)) \frac{\partial z_k^+(p)}{\partial \gamma_{ij}^1(3)}.$$

$$(2.13) \quad \frac{\partial z_k^+(p)}{\partial \gamma_{ij}^1(3)} = h(z_k^-(p)) \frac{\partial z_k^-(p)}{\partial \gamma_{ij}^1(3)} = h(z_k^-(p)) \delta_{ki} y_j^+(p).$$

$$(2.14) \quad \delta_{ij} = 1 \text{ for } i = j, \quad = 0 \text{ otherwise.}$$

$$(2.15) \quad \frac{\partial V_0}{\partial \gamma_{ij}^1(3)} = \sum_{p=1}^{N_p} (z_i^+(p) - e_i(p)) h(z_i^-(p)) y_j^+(p).$$

$$(2.16) \quad \frac{\partial V_0}{\partial \gamma_{ij}^1(2)} = \sum_{p=1}^{N_p} \sum_{k=1}^{N_{out}} (z_k^+(p) - e_k(p)) h(z_k^-(p)) h(y_i^-(p)) x_j^+(p) \gamma_{ki}^1(3).$$

$$(2.17) \quad \frac{\partial V_0}{\partial \gamma_{ij}^1(1)} = \sum_{p=1}^{N_p} \sum_{k=1}^{N_{out}} (z_k^+(p) - e_k(p)) h(z_k^-(p)) \sum_{m=1}^{N_2} \gamma_{km}^1(3) h(y_m^-(p)) h(x_i^-(p)) d_j(p) \gamma_{mi}^1(2).$$

V_1 の微分から

$$(2.18) \quad \frac{\partial V_1}{\partial \gamma_{ij}^1(1)} = \alpha \sum_{p=1}^{N_p} (|x_i^-(p)| - c_0) d_j(p) s_0 \quad (s_0 = 1 \text{ for } 0 < x_i^-(p), = -1 \text{ for } x_i^-(p) < 0)$$

$$(2.19) \quad \frac{\partial V_1}{\partial \gamma_{ij}^1(2)} = \alpha \sum_{p=1}^{N_p} (|y_i^-(p)| - c_0) x_j^+(p) s_0 \quad (s_0 = 1 \text{ for } 0 < y_i^-(p), = -1 \text{ for } y_i^-(p) < 0)$$

$$(2.20) \quad \frac{\partial V_1}{\partial \gamma_{ij}^1(3)} = \alpha \sum_{p=1}^{N_p} (|z_i^-(p)| - c_0) y_j^+(p) s_0 \quad (s_0 = 1 \text{ for } 0 < z_i^-(p), = -1 \text{ for } z_i^-(p) < 0)$$

$\gamma_{ij}^1(3)$ の実際の coding は $\gamma_{ji}^1(3)$ である。 $j = 1, \dots, N_{in}, i = 1, \dots, N_{ou}$

$$(2.21) \quad \frac{\partial V_0}{\partial \gamma_{ji}^1(3)} = \sum_{p=1}^{N_p} (z_i^+(p) - e_i(p)) h(z_i^-(p)) y_j^+(p).$$

$$(2.22) \quad \frac{\partial V_1}{\partial \gamma_{ji}^1(3)} = \alpha \sum_{p=1}^{N_p} (|z_i^-(p)| - c_0) y_j^+(p) s_0 \quad (s_0 = 1 \text{ for } 0 < z_i^-(p), = -1 \text{ for } z_i^-(p) < 0)$$

2.3. ニュラルネットでの計算プログラム. 計算プログラムを Fortran 言語で書いた (付録 1 にプログラムソースのリスト)。その特徴はベクトル化の効率が非常に良い。ベクトル化の効率の良い計算機 (supercomputer) での計算速度は、そうでない計算機に比べて 15 倍程度速いことが確認されている。

3. 入力データの変換 (PART 1)

ニュラルネットに入力するタクシードライバーのデータは、行列の形式で表現されている。カラムには様々なデータ (左右の加速度、上下の加速度、前後の加速度、速度など) が入っている。行番号は、時間に対応していて、その間隔は一分である。測定データはエクセルに入っていて、コンピュータで計算できる形式にデータを変換した。

変換は、

- (1) 意味のないデータを排除して、カラムを詰める、
- (2) 時間を 1 時間を単位にして変更した。

このプログラムのソースを、付録 2 に載せてある。

4. 入力データの変換 (PART 2)

更に、Part 1 で変更した入力データを、計算しやすいように変更した。そのプログラムソースが付録 3 にある。

5. 終りに：感想

今回石野教授の勧めでここの研究所の学外実習生として学んだことは、僕の人生の中でも「未知の分野にコミットする」という意味でいい経験であったことと思います。文系大学に進学し経営学を専攻して、これといった専門的なスキルもない自分にとってプログラミング等のスキルやその他様々な理系的知識はかなり馴染みの薄いものでした。これまで実際に触れたプログラミング言語といえば大学一年目に単位数取得のためだけに特に明確な目的意識のないまま学習したパスカル言語だけという状況でした。

この先理系方面に進まないとしても「『ものごと』を『みる』尺度を増やす」という意味だけでも貴重な経験だったと思います。前に将来人間の命をあずかる職業に携わるであろう医学部生達が倫理観の欠けているとしか思えないような犯罪行動を短絡的に犯してしまったというニュースがありました。最近では現職知事が言語道断の犯罪を犯したにもかかわらずそれを全面否定していたが結局情状酌量を目論み事実を認めるという恥ずかしい事件もありました。

これら事件の内容を知ると既存の文系理系という二つの枠組みにはっきりと区別化された大学教育制度に疑問を感じるとともに、どちらかと言えば文系的である倫理とか哲学とかいったモラルに焦点を当てる学問の重要性が浮彫りになってきます。尊敬する哲学者を語る小児科の先生が身近かにいてもいいと思います。その逆も同様だと思います。実際に医者資格を持つ小説家や文豪であつた政治家だっています。もちろん専門的分野の学問も大事なのです。しかし今の世の中においては文系でも理系でもない、人間として生きていく中でその基礎を形成するための教育が足りないのではないのでしょうか。「もの」の「見方」をある程度広げた上で専門的な学問をもっと極める必要があると思います。

私の座右の銘に「無分別」という言葉があります。この言葉の意味を世間では「分別が無い」、つまり「常識と非常識の違いを知らない」というマイナスな性格のものとして認知されています。元来この言葉は『「ものごと」というものを「分別無く」平等にみる』という意味を持ちます。例えば学歴社会の中ではどうしても第一志望の大学に合格した生徒と全大学に落ち浪人生活を余儀なくされた生徒が存在します。ここで世間一般では前者を「成功者」後者を「落伍者」と呼び前者を尊敬し、礼賛し後者を蔑み同情しますが元々人生自体に「成功」も「失敗」ありません。何年か前にOL達の間で「三高」という表現が流行りましたがこれも同じです。収入や学歴、身長でその人間を評価していました。「ものごと」をいちいち優劣をつけて「みる」こと自体よくないことなのです。全てのものごとを平等にみるという目を持って生きることを心掛けたい。この「平等にものごとをみる」ことは難しいものですが、そうするための助けとなることが「『ものごと』を『みる』尺度を増やす」ことだと私は思っています。

今回の研究テーマ、「時々刻々と変化するシステムデータ（入力側）を、システムと相互作用する人間側がどう受けとめるか（感じ方）のデータ（出力側）との対応関係（写像）を構築する。」についての知識はこちらに来た当初正直言ってゼロでした。この二ヶ月間、新上研究員の丁寧な御指導を受けながら最終的にテーマの一部は理解できたものと思います。もともと大学では「経営情報システム」という研究テーマのもと、データマイニングやインターネットマーケティングについて学んでいますが、今回のQOS分析とは「情報を扱う」という観点からみれば共通点もあると感じました。デジタルエコノミーが進行して既存の流通システムやそこを流れる「モノ」（消費者のニーズに応える「モノ」やコンテンツ）には大きな変化が訪れました。

情報を取り入れたSCMの重要度が増してきていることや、今までのようにスーパーやデパートで「モノ」が売れなくなっていることからそのことは窺えます。今回のテーマはこのうち後者の問題にたいして「モノ」のデザインコンセプトのヒントを与えてくれるものであると私は認識しました。この研究テーマの最終段階には立ち会うことはできませんでしたが様々な方面での「モノ」のデザインにおいて、QOS分析は有効な分析手段となるだろうという思いを強く感じました。今回実際に計測したタクシーの運転におけるデータを使用し分析を進めてきました。その他でもイベントという「モノ」の企画において「モノ」と消費者（観客）間に介在する関係についても使えるとのことでしたので「企画」にも興味がある自分として機会があればまた携わりたいと思います。

謝辞

最後になりましたが指導していただいた新上研究員をはじめ、基礎的な質問にも対応してくださった野口研究員、北川研究員、その他の方々に深く感謝します。

6. プログラムのソース (ニューラルネット): 付録 1

```

c sub.read_data でデータ (評価とタクシードライバー) を読み込む
c sub.cal_force_xyz の中で
c sub.init_force で力を初期化, 入力データをパターン毎に読み込んで力を計算する
c
  program qos2000
  include 'common_qos2000'
  call parameters
  call read_data1
c   call read_data1
  xmin_cost=10000000.0d0
  ixmin_cost=0
  do 1000 iii=1,iiimax
  call cal_force_xyzh
  call hokan_xyz
  if(iii.le.3)then
  cost0=cost
  vvcost0=vvcost
  h0=cost+vvcost
  endif
  if(cost.lt.xmin_cost)ixmin_cost=iii
  if(cost.lt.xmin_cost)xmin_cost=cost
  idiff=iii-ixmin_cost
  if(5000.lt.idiff)write(6,*)'stop: 5000<idiff'
  if(5000.lt.idiff)stop
  hh=cost+vvcost
  write(6,9000)iii,idiff,cost,vvcost,hh,
  &xmin_cost,cost/cost0,vvcost/vvcost0,hh/h0
9000 format(1h ,2i7,4f20.5,3f10.5)
  call flush(6)
1000 enddo
  stop
  end
  subroutine parameters
  include 'common_qos2000'
  iiimax=200000
  dt=0.00125d0
  dtt=d20*dt
  dt2=dt*dt
  temp=1.0d0
  alpha=10.0d0
  crit0=1.0d0-1.0d-14
  vrate=20.0d0
  xlambda=1.0d0
c-----
c   iiimax: 実行回数
c   dt: 時間刻み
c   temp: sigmoid 関数の温度
c   crit0: sigmoid 関数の boundary
c   alpha: boundary での跳ね返し係数
c   vrate: 初期での cost と速度エネルギーの比
c   xlambda: friction constat
c-----
  return
  end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

  subroutine cal_force_xyzh

```

```

include 'common_qos2000'
call init_force
do 10 m=1,n_pa
call init_data(m)
call cal_xyzh
call force
10 enddo
return
end

subroutine init_force
include 'common_qos2000'
do 10 j=1,n_ou
do 10 i=1,n_in
force_g3(i,j)=0.0d0
10 enddo
do 20 j=1,n_in
do 20 i=1,n_in
force_g2(i,j)=0.0d0
force_g1(i,j)=0.0d0
20 enddo
cost=0.0d0
return
end

subroutine init_data(np)
include 'common_qos2000'
if(n_pa.lt.np)write(6,*)'init_data: stop due to n_pa<np'
if(n_pa.lt.np)stop
do 10 i=1,n_in
d(i)=d0(i,np)
10 enddo
do 20 i=1,n_ou
eval(i)=eval0(i,np)
20 enddo
return
end

subroutine force
include 'common_qos2000'
do 10 i=1,n_ou
ff=(z_r(i)-eval(i))*hz(i)
ff1=ccz(i)
do 15 j=1,n_in
force_g3(j,i)=force_g3(j,i)+ff*y_r(j)+y_r(j)*ff1
15 enddo
10 enddo
c write(6,*)'1 force:stop'
do 20 k=1,n_ou
ff=(z_r(k)-eval(k))*hz(k)
do 22 j=1,n_in
ffj=x_r(j)
force_g2(i,j)=force_g2(i,j)+x_r(j)*ccy(i)
do 24 i=1,n_in
force_g2(i,j)=force_g2(i,j)+ff*hy(i)*ffj*gamma3(i,k)
24 enddo
22 enddo
20 enddo
c write(6,*)'2 force:stop'

```



```

do 30 k=1,n_ou
ff=(z_r(k)-eval(k))*hz(k)
c write(6,*)'force: k, n_ou= ',k,'/',n_ou,'(n_in=',n_in,')'
do 32 j=1,n_in
gg=d(j)
do 34 i=1,n_in
ffxd=hx(i)*gg
force_g1(i,j)=force_g1(i,j)+d(j)*ccx(i)
do 36 m=1,n_in
force_g1(i,j)=force_g1(i,j)+ff*gamma3(m,k)*hy(m)*ffxd*gamma2(m,i)
36 enddo
34 enddo
32 enddo
30 enddo
return
end

```

```

subroutine read_data
include 'common_qos2000'
c データを読み込む。
c gamma3(i,j) の配列は gamma1,gamma2 とは逆にしてある。
do 10 kp=1,n_in
c read(10,*) m,(d0(kp,np),np=1,n_pa)
10 enddo
do 15 kp=1,n_ou
c read(10,*) m,(eval0(kp,np),np=1,n_pa)
15 enddo
do 20 i=1,n_in
do 20 j=1,n_in
gamma1(j,i)=0.1d0
gamma2(j,i)=0.1d0
mm=(i+j)-2*((i+j)/2)
xx=2.0d0*(float(mm)-0.5d0)
vgamma1(j,i)=0.1d0*xx
vgamma2(j,i)=-0.1d0*xx
gamma10(j,i)=gamma1(j,i)-dt*vgamma1(j,i)
gamma20(j,i)=gamma2(j,i)-dt*vgamma2(j,i)
20 enddo
do 30 i=1,n_ou
do 30 j=1,n_in
gamma3(j,i)=0.1d0
mm=(i+j)-2*((i+j)/2)
xx=2.0d0*(float(mm)-0.5d0)
vgamma3(j,i)=0.1d0*xx
gamma30(j,i)=gamma3(j,i)-dt*vgamma3(j,i)
30 enddo
call sym_gamma
degrees=float(n_in*n_in+n_in*n_ou)
return
end

```

```

subroutine read_data1
include 'common_qos2000'
c データを読み込む。
c gamma3(i,j) の配列は gamma1,gamma2 とは逆にしてある。
if(crit0.lt.0.999999d0)write(6,*)'stop because 0.9999<crit0'
if(crit0.lt.0.999999d0)stop

```

```

zz=1.0d0/crit0-1.0d0
crit=-temp*log(zz)
degrees=float(n_in*n_in+n_in*n_ou)
write(6,*) 'crit= ',crit
do 10 k1=1,n_in
do 10 k2=1,n_pa
xx=float(k1+k2)*0.4d0
d0(k1,k2)=0.5d0*(cos(xx)+1.0d0)
10  enddo
do 15 k1=1,n_ou
do 15 k2=1,n_pa
kxx=(k1+k2)*(k1+k2)
mm=kxx-2*(kxx/2)
eval0(k1,k2)=float(mm)
15  enddo
vv=0.0d0
do 20 i=1,n_in
do 20 j=1,n_in
mm=(i+j)**2
mm=mm-2*(mm/2)
xx=2.0d0*(float(mm)-0.5d0)
gamma1(j,i)=0.001d0*xx
gamma2(j,i)=-0.001d0*xx
vgamma1(j,i)=0.1d0*xx
vgamma2(j,i)=-0.1d0*xx
vv=vv+vgamma1(j,i)**2+vgamma2(j,i)**2
gamma10(j,i)=gamma1(j,i)-dt*vgamma1(j,i)
gamma20(j,i)=gamma2(j,i)-dt*vgamma2(j,i)
20  enddo
do 30 i=1,n_ou
do 30 j=1,n_in
mm=(i+j)**2
mm=mm-2*(mm/2)
xx=2.0d0*(float(mm)-0.5d0)
gamma3(j,i)=0.001d0*xx
vgamma3(j,i)=0.1d0*xx
vv=vv+vgamma3(j,i)**2
gamma30(j,i)=gamma3(j,i)-dt*vgamma3(j,i)
30  enddo
vvcost=0.5d0*vv
call sym_gamma
return
end
subroutine init_velocity(icon)
include 'common_qos2000'
rate=sqrt(vrate)*sqrt(cost)/sqrt(vvcost)
icon=100
vv=0.0d0
do 1 j=1,n_in
do 1 i=1,n_in
vgamma1(i,j)=rate*vgamma1(i,j)
vgamma2(i,j)=rate*vgamma2(i,j)
gamma10(i,j)=gamma1(i,j)-dt*vgamma1(i,j)
gamma20(i,j)=gamma2(i,j)-dt*vgamma2(i,j)
vv=vv+vgamma1(i,j)**2+vgamma2(i,j)**2
1  enddo
do 2 j=1,n_ou
do 2 i=1,n_in
vgamma3(i,j)=rate*vgamma3(i,j)

```

```

gamma30(i,j)=gamma3(i,j)-dt*vgamma3(i,j)
vv=vv+vgamma3(i,j)**2
2  enddo
write(6,*) cost,0.5d0*vv,cost/(0.5d0*vv)
return
end

```

```

subroutine sym_gamma
include 'common_qos2000'
do 10 i=1,n_in
do 10 j=1,n_in
gamma1p(i,j)=gamma1(j,i)
gamma2p(i,j)=gamma2(j,i)
10 enddo
do 20 i=1,n_ou
do 20 j=1,n_in
gamma3p(i,j)=gamma3(j,i)
20 enddo
return
end

```

```

subroutine cal_xyzh
include 'common_qos2000'
vv1=0.0d0
do 10 k1=1,n_in
xx=0.0d0
ccx(k1)=0.0d0
do 12 k2=1,n_in
xx=xx+gamma1p(k2,k1)*d(k2)
12 enddo
x_l(k1)=xx
gg=abs(xx)
if(crit.lt.gg) vv1=vv1+alpha*(gg-crit)
if(crit.lt.xx) ccx(k1)= alpha
if(xx.lt.-crit) ccx(k1)=-alpha
xx=xx/temp
if(40.0d0.lt.xx)xx=40.0d0
if(xx.lt.-40.0d0)xx=-40.0d0
xx=exp(-xx)
ff=1.0d0/(1.0d0+xx)
x_r(k1)=ff
hx(k1)=ff*(1.0d0-ff)/temp
10 enddo
do 20 k1=1,n_in
yy=0.0d0
ccy(k1)=0.0d0
do 22 k2=1,n_in
yy=yy+gamma2p(k2,k1)*x_r(k2)
22 enddo
y_l(k1)=yy
gg=abs(yy)
if(crit.lt.gg) vv1=vv1+alpha*(gg-crit)
if(crit.lt.xx) ccy(k1)= alpha
if(xx.lt.-crit) ccy(k1)=-alpha
xx=yy/temp
if(40.0d0.lt.xx)xx=40.0d0
if(xx.lt.-40.0d0)xx=-40.0d0
xx=exp(-xx)
ff=1.0d0/(1.0d0+xx)

```

```

y_r(k1)=ff
hy(k1)=ff*(1.0d0-ff)/temp
20  enddo
do 30 k1=1,n_ou
zz=0.0d0
ccz(k1)=0.0d0
do 32 k2=1,n_in
zz=zz+gamma3(k2,k1)*y_r(k2)
32  enddo
z_l(k1)=zz
gg=abs(zz)
if(crit.lt.gg) vv1=vv1+ alpha*(gg-crit)
if(crit.lt.zz) ccz(k1)= alpha
if(zz.lt.-crit) ccz(k1)=-alpha
xx=zz/temp
if(40.0d0.lt.xx)xx=40.0d0
if(xx.lt.-40.0d0)xx=-40.0d0
xx=exp(-xx)
ff=1.0d0/(1.0d0+xx)
z_r(k1)=ff
hz(k1)=ff*(1.0d0-ff)/temp
30  enddo
vv=0.0d0
do 40 k=1,n_ou
ff=z_r(k)-eval(k)
vv=vv+ff*ff
c   write(6,*) 'k,z_r(k),eval(k),d(k)= ',k,z_r(k),eval(k),d(k)
40  enddo
cost=cost+0.5d0*vv+vv1
c   if(0.00001d0.lt.vv1)write(6,*)'warnig vv1===',vv1
c   if(0.00001d0.lt.vv1)stop
c   write(6,*)vv,cost,vv1
return
end

```

```

subroutine hokan_xyz
include 'common_qos2000'
data iccn/0/
save icon
if(icon.eq.0)call init_velocity(icon)
vv=0.0d0
do 10 j=1,n_in
do 10 i=1,n_in
gamma12(i,j)=d20*gamma1(i,j)-gamma10(i,j)-dt2*force_g1(i,j)
gamma22(i,j)=d20*gamma2(i,j)-gamma20(i,j)-dt2*force_g2(i,j)
vgamma1(i,j)=(gamma12(i,j)-gamma10(i,j))/dt
vgamma2(i,j)=(gamma22(i,j)-gamma20(i,j))/dt
vv=vv+vgamma1(i,j)**2+vgamma2(i,j)**2
10  enddo
do 20 j=1,n_ou
do 20 i=1,n_in
gamma32(i,j)=d20*gamma3(i,j)-gamma30(i,j)-dt2*force_g3(i,j)
vgamma3(i,j)=(gamma32(i,j)-gamma30(i,j))/dt
vv=vv+vgamma3(i,j)**2
20  enddo
vvcost=0.5d0*vv
do 30 j=1,n_in
do 30 i=1,n_in
gamma12(i,j)=dt*vgamma1(i,j)+gamma10(i,j)

```

```

gamma22(i,j)=dtt*vgamma2(i,j)+gamma20(i,j)
gamma10(i,j)=gamma1(i,j)
gamma20(i,j)=gamma2(i,j)
gamma1(i,j)=gamma12(i,j)
gamma2(i,j)=gamma22(i,j)
30  enddo
do 40 j=1,n_ou
do 40 i=1,n_in
gamma32(i,j)=dtt*vgamma3(i,j)+gamma30(i,j)
gamma30(i,j)=gamma3(i,j)
gamma3(i,j)=gamma32(i,j)
40  enddo
call sym_gamma
return
end

cc  common_qos2000 file *****
implicit real*8 (a-h,o-z)
parameter (n_in=50,n_ou=10)
parameter (n_pa=2)
common /aa1/gamma1(n_in,n_in),force_g1(n_in,n_in)
common /aa2/gamma2(n_in,n_in),force_g2(n_in,n_in)
common /aa3/gamma3(n_in,n_ou),force_g3(n_ou,n_in)
common /aa4/d(n_in),d0(n_in,n_pa)
common /aa5/eval(n_ou),eval0(n_ou,n_pa)
common /aa6/gamma1p(n_in,n_in)
common /aa7/gamma2p(n_in,n_in)
common /aa8/gamma3p(n_ou,n_in)
common /bb1/x_l(n_in),x_r(n_in),hx(n_in)
common /bb2/y_l(n_in),y_r(n_in),hy(n_in)
common /bb3/z_l(n_ou),z_r(n_ou),hz(n_ou)
common /bb4/ccx(n_in),ccy(n_in),ccz(n_ou)
common /xyz1/gamma12(n_in,n_in),gamma10(n_in,n_in)
common /xyz2/gamma22(n_in,n_in),gamma20(n_in,n_in)
common /xyz3/gamma32(n_in,n_ou),gamma30(n_in,n_ou)
common /para0/dt,dt2,dtt,temp,crit0,crit,alpha,vvcost,cost
common /para1/iiimax,degrees,vrate,xlambda
common /xyz4/vgamma1(n_in,n_in),vgamma2(n_in,n_in)
common /xyz5/vgamma3(n_in,n_ou)
data d20/2.0d0/

```

7. データの変換のプログラム: 付録2

```
dimension a(40),b(40)
open(10,file='ftn05',status='old')
read(10,*) nmax
do 10 i=1,nmax
read(10,*)(a(k),k=1,35)
if(i.eq.1)aa=a(3)
b(1)=24.0*(a(3)-aa)+a(4)+a(5)/60.0
m=1
do 20 k1=7,34
m=m+1
b(m)=a(k1)
20  enddo
write(6,*) m
stop
write(6,1000)i,(b(k),k=1,m)
1000 format(1h ,i4,24f15.4)
10  enddo
stop
end
```

8. データの変換のプログラム: 付録3

```

dimension a(40),b(40)
open(10,file='ftn05',status='old')
read(10,*) nmax
do 10 i=1,nmax
read(10,*)(a(k),k=1,35)
if(i.eq.1)aa=a(3)
b(1)=24.0*(a(3)-aa)+a(4)+a(5)/60.0
m=1
do 20 k1=7,34
m=m+1
b(m)=a(k1)
20  enddo
write(6,*) m
stop
write(6,1000)i,(b(k),k=1,m)
1000 format(1h ,i4,24f15.4)
10  enddo
stop
end

```