

CBF program 161264 edda sveinsdottir

data: n, /text/, gas, pCO2, hgb, tstart, remact, backgr, fmax, scalfac,
dt, recirc, restime, infection, f(1), f(2), ..., f(n).

```
begin integer n; n:= in one;  
begin integer i, j, h, g, K, C; boolean boole;  
real pCO2, hct, tstart, remact, backgr, fmax, scalfac, dt, recirc, restime,  
lg, lw, S, fi, w, v, sum, t, d, tstar, astar, bstar;  
array k, kk, a, x, VA[1:4], A[1:4, 1:4], f[1:n];
```

```
procedure gjr(a, n, eps) exit: (singular);  
integer n; real eps; array a; label singular;  
begin integer i, j, k; real pivot, z;  
array b, c[1:n]; integer array p, q[1:n];  
for k:=1 step 1 until n do  
begin pivot:=0;  
for i:=k step 1 until n do  
for j:=k step 1 until n do  
if abs(a[i, j]) > abs(pivot) then  
begin pivot:= a[i, j]; p[k]:=i; q[k]:=j end;  
if abs(pivot) ≤ eps then go to singular;  
if p[k] ≠ k then for j:=1 step 1 until n do  
begin z:= a[p[k], j]; a[p[k], j]:= a[k, j]; a[k, j]:=z end j;  
if q[k] ≠ k then for i:=1 step 1 until n do  
begin z:= a[i, q[k]]; a[i, q[k]]:= a[i, k]; a[i, k]:=z end i;  
for j:=1 step 1 until n do begin  
if j=k then begin b[j]:= 1/pivot; c[j]:=1 end  
else begin b[j]:= -a[k, j]/pivot; c[j]:= a[j, k] end;  
a[k, j]:= a[j, k]:= 0 end j;  
for i:=1 step 1 until n do  
for j:=1 step 1 until n do a[i, j]:= a[i, j] + c[i]×b[j]  
end k;  
for k:=n step -1 until 1 do  
begin  
if p[k] ≠ k then for i:=1 step 1 until n do  
begin z:= a[i, p[k]]; a[i, p[k]]:= a[i, k]; a[i, k]:=z end i;  
if q[k] ≠ k then for j:=1 step 1 until n do  
begin z:= a[q[k], j]; a[q[k], j]:= a[k, j]; a[k, j]:=z end j;  
end k;  
end gjr;
```

```
outcr; outtext(⟨<CBF-program 161264 e.s.⟩);  
outcr; outcopy(⟨<///⟩); outcr;  
GAS: K:= in char; if K≠34∧K≠162∧K≠23∧K≠151 then goto GAS;  
input(pCO2, hct, tstart, remact, backgr, fmax, scalfac, dt, recirc, restime);  
INJ: C:= in char; if C≠51∧C≠179∧C≠57∧C≠185∧C≠21∧C≠149 then goto INJ;  
hct:= hct/34; if K=34∨K=162  
then begin outcr; outtext(⟨<krypton 85⟩);  
lg:= 0.92×0.843/(hct + 0.685×(1-hct));  
lw:= 1.26×0.843/(hct + 0.685×(1-hct)) end  
else begin outcr; outtext(⟨<xenon 133, ⟩);  
lg:= 0.77×0.744/(hct + 0.487×(1-hct));  
lw:= 1.44×0.744/(hct + 0.487×(1-hct)) end;  
S:= 1/(0.5×(1+exp((pCO2-40)/18.5)));  
if tstart=1 then outtext(⟨<,,,digital output,,,⟩) else begin  
outtext(⟨<,,,tstart:,,,⟩); output(⟨<-n.dd⟩, tstart) end;  
outtext(⟨<,,,st factor, S:⟩); output(⟨<-n.dd⟩, S);  
outtext(⟨<,,,remaining act:⟩); output(⟨<ndd.d⟩, remact); outcr;  
outtext(⟨<aPCO2:,,,⟩); output(⟨<dd⟩, pCO2); outtext(⟨<,,,fmax:,,,⟩);  
output(⟨<dddd.d⟩, fmax); outtext(⟨<,,,recirc corr:, ⟩);  
output(⟨<n.dd⟩, recirc); outtext(⟨<,,,background:,,,⟩);  
output(⟨<ndd.d⟩, backgr); outcr; outtext(⟨<hgb:,, ⟩);  
output(⟨<nd.d⟩, hct×34); outtext(⟨<,,,scal fact:⟩);
```

```

output({dddd.d}, scalfac); outtext({<,,,inj:});
outtext({if C=51vC=179 then {<car communis} else if C=57vC=185
  then {<,car interna} else {<,vertebralis}});
outtext({<,,,resolv time:}); output({d.d10-d}, restime); outcr;
outtext({<dt:,,,}); output({n.dd}, dt);

input(f); for i:=1 step 1 until n do begin fi:= f[i];
  f[i]:= fixscalfac*(1+fixscalfac*restime)-backgr*scalfac end;
fmax:= fmax*scalfac*(1+fmax*scalfac*restime) - backgr*scalfac;
remact:=remact*scalfac*(1+remact*scalfac*restime) - backgr*scalfac;
g:= 0; i:= 0;
for i:= i+1 while ln(f[i+1]) > 0.5*(ln(f[i])+ln(f[i+2])) do g:=i;
outtext({<,,,data excluded:}); output({nd}, g);
outtext({<,,,data analysed:}); output({ndd}, n-g); outcr; outtext({<

```

T W O - C O M P A R T M E N T A N A L Y S I S
(axexp(-alpha*x) + bxexp(-beta*x))});

```

begin array y[1:n]; real p,q;
procedure Lsq(starti, interval);
value starti, interval; integer starti, interval;
begin real X, XX, Y, XY, D; X:=XX:=Y:=XY:= 0;
for i:=starti step 1 until starti+interval do begin
  X:= X+i*dt; XX:= XX + (i*dt)2;
  Y:= Y + y[i]; XY:= XY + i*dt*y[i] end;
D:= (interval+1)*XX - X*X;
p:= ((interval+1)*XY - X*Y)/D; q:= (X*Y - X*XY)/D;
end Lsq;

```

```

for i:=entier(7/dt) step 1 until n do y[i]:= ln(f[i]);
Lsq(entier(7/dt), n-entier(7/dt)); k[4]:= -p; k[3]:= exp(q); i:= g;
if k[4]>0.23 then begin outcr; outcr;
outtext({<thalf of tail}); output({-nd.d}, 0.693/k[4]);
outtext({<,min too steep}); outcr;
outtext({<check log(tailpart-background) after 7 min});
goto END end;
for i:=i+1 while i<g+1+entier(5/dt) ^ f[i]-k[3]*exp(-k[4]*i*dt)>0.1 do
begin y[i]:= ln(f[i]-k[3]*exp(-k[4]*i*dt)); h:= i end;
Lsq(g+1, h-(g+1)); k[2]:= -p; k[1]:= exp(q); outcr; outcr;
outtext({<initial guess});
output({-ndddd.dd}, k[1]/fmax*100, k[2], k[3]/fmax*100);
output({-ndddd.ddd}, k[4]);
end of curvepeeling block;

```

```

for i:=1,2,3,4 do kk[i]:= k[i]; w:= 1;

```

IGEN:

```

for j:=1 step 1 until 4 do
begin VA[j]:= 0; for i:=j step 1 until 4 do A[j,i]:= 0 end j;
for j:= g+1 step 1 until n do
begin if tstart=1 then w:= 1/f[j];
  a[1]:= exp(-k[2]*j*dt); a[2]:= -a[1]*k[1]*j*dt;
  a[3]:= exp(-k[4]*j*dt); a[4]:= -a[3]*k[3]*j*dt;
  v:= f[j] - a[1]*k[1] - a[3]*k[3];
  for i:=1 step 1 until 4 do begin
    VA[i]:= VA[i] + w*v*a[i];
    for h:=i step 1 until 4 do A[i,h]:= A[i,h]+w*a[i]*a[h] end i;
  end j;
for i:=2,3,4 do for j:=i-1 step -1 until 1 do A[i,j]:= A[j,i];
gjr(A,4,110-7,SINGULAR); for i:=1 step 1 until 4 do begin sum:= 0;
for j:=1 step 1 until 4 do sum:= sum+A[i,j]*VA[j]; x[i]:= sum end i;
for i:=1,2,3,4 do k[i]:= k[i]+x[i];
if k[1]>2*fmax v k[1]<-0.01*fmax v k[2]>5 v k[2]<-0.1
vk[3]>2*fmax v k[3]<-0.01*fmax v k[4]>1 v k[4]<-0.1

```

```

then begin outtext(⟨<,,,divergence⟩); boole:= true; goto FLOW end;
if abs(x[1]/kk[1])+abs(x[2])+abs(x[3]/kk[3])+abs(x[4]) > 10-4
then goto IGEN;
outcr; outcr; outtext(⟨<a,alpha,b,beta:..⟩);
output(⟨-ndddd.dd⟩, k[1]/fmax×100, k[2], k[3]/fmax×100);
output(⟨-ndddd.dd⟩, k[4]);
boole:= false; t:= 0; d:= 1;
TSTAR: if k[1]×exp(-k[2]×t) + k[3]×exp(-k[4]×t) > fmax then
begin t:= t+d; goto TSTAR end;
t:= t-d; d:= d/10; if d>0.001 then goto TSTAR; tstar:= t;
astar:=k[1]×exp(-k[2]×tstar); bstar:=(k[3]-remact)×exp(-k[4]×tstar);

FLOW: begin boolean digout; array r[1:n]; integer begi;
real fg,fw,wg,ww,lambda,delaytime,begarea,area,ten,CBF;
tstar:= (g+2)×dt - (fmax-f[g+2])×dt/(f[g+1]-f[g+2]);
if boole then begin k[4]:= kk[4]; goto RAW DATA end;
sum:= 0; for j:=g+1 step 1 until n do begin
r[j]:= k[1]×exp(-k[2]×j×dt) + k[3]×exp(-k[4]×j×dt) - f[j];
if tstart=1 then w:= 1/f[j]; sum:= sum+r[j]×2×w end j;
outcr; outtext(⟨<,,,,,,,,,,,,,S.D.:..⟩);
output(⟨-ndddd.dd⟩, sqrt(abs(A[1,1])×sum/(n-4-g))×100/fmax, sqrt(
abs(A[2,2])×sum/(n-4-g)), sqrt(abs(A[3,3])×sum/(n-4-g))×100/fmax);
output(⟨-ndddd.dd⟩, sqrt(abs(A[4,4])×sum/(n-4-g)));
outcr; outtext(⟨<
calculated minus observed values in percent of fmax:⟩); outcr;
for i:=g+1 step 1 until n do begin
output(⟨-ndd.d⟩, r[i]/fmax×100); if i-g = (i-g):10×10 then outcr end;
fg:= k[2]×lg×100; fw:= k[4]×lw×100; k[4]:= kk[4];
wg:= (astar/fg)/(astar/fg + bstar/fw); ww:= 1-wg; outcr; outcr;
outtext(⟨<,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,×S⟩);
outcr; outtext(⟨<rel weight grey:..⟩); output(⟨-n.dd⟩, wg);
outtext(⟨<,,,,,flow grey:⟩); output(⟨-ndddd.d⟩, fg, fg×S);
outtext(⟨<,,,ml/100g/min⟩);
outcr; outtext(⟨<rel weight white:..⟩); output(⟨-n.dd⟩, ww);
outtext(⟨<,,,,,flow white:⟩); output(⟨-ndddd.d⟩, fw, fw×S);
outtext(⟨<,,,ml/100g/min⟩);
outcr; outtext(⟨<weighted lambda:..⟩); output(⟨-n.dd⟩, wg×lg+ww×lw);
outtext(⟨<,,,,,flow mean:⟩); output(⟨-ndddd.d⟩,
wg×fg+ww×fw, (wg×fg+ww×fw)×S); outtext(⟨<,,,ml/100g/min⟩);
RAW DATA: outcr; outtext(⟨<

```

N O N - C O M P A R T M E N T A L A N A L Y S I S

```

(lambda for w-grey/w-white = 6/4 : ⟩);
lambda:= lg×0.6 + lw×0.4; output(⟨-n.dd⟩, lambda); outtext(⟨<⟩);
digout:= tstart=1; delaytime:= tstar - tstart;
for i:=1 step 1 until n do f[i]:= f[i] - remact×exp(-k[4]×i×dt);
fmax:= fmax - remact×exp(-k[4]×0.5×(tstar+tstart));
outcr; outcr; outtext(⟨<,,,,,,,,,,,,,,,,,,,,,,,,,,,,,×S
CBFr(10 min):⟩); recirc:= recirc×0.83;
begi:= if digout then 1 else if tstar<0 then 1 else entier(tstar/dt)+1;
ten:= if digout then 10/dt else (10+tstart)/dt;
if n<ten then begin outtext(⟨<,,,not enough data⟩); goto next end;
if digout then begin
area:= 0; for i:=1 step 1 until ten do area:= area + f[i];
area:= area×dt; goto CBFten end;
begarea:= area:= delaytime×fmax/dt+(begi-tstar/dt)×0.5×(fmax+f[begi]);
for i:=begi step 2 until ten-2 do
area:= area+1/3×(f[i]+4×f[i+1]+f[i+2]);
area:= dt×(area + (ten-begi-entier(ten-begi):2×2)×f[ten]);
CBFten: CBF:= (fmax-f[ten])×lambda×100/(area-recirc×fmax);
output(⟨-ndddd.d⟩, CBF, CBF×S); outtext(⟨<,,,ml/100g/min⟩);
next: outcr; outtext(⟨<CBFr(infinity):⟩); recirc:= recirc×2;
if digout then begin
area:= 0; for i:=1 step 1 until n-2 do area:= area+f[i];

```

```

area:= areaxdt + (f[n-3]+f[n-2]+f[n-1]+f[n])/4/k[4]; goto CBFinf end;
area:= begarea; for i:=begi step 2 until n-2 do
area:= area + 1/3*(f[i]+4*f[i+1]+f[i+2]);
area:= areaxdt + f[begi + (n-begi):2*2]/k[4];
CBFinf: CBF:= fmax*lambda*100/(area-recirc*fmax);
output(†-ndddd.d†, CBF, CBF*S); outtext(†<,,ml/100g/min†);
end of FLOW block; goto END;
SINGULAR: outtext(†<
normal matrix singular (the curve may be monoexponential)†);
boole:= true; goto FLOW;
END: outchar(11);
end end;

```

```

Ff
60
/Otto Lind f 050812 hø com ve temp inj 2 /
xenon
45.8
14.5
0.05
5.1
2.5
122
600
0.25
0.45
210-7
com

```

```

110,98,89,82,75,69,65,61,58,55,
52,49,47,45,44,43,41,39,38,37,
36,35,35,35,34,33,32,32,31,30,
30,29,29,29,28,28,28,26,26,26,
26,25,25,25,24,24,24,23,23,23,
22,22,22,22,21,21,21,21,21,21,

```

```

33
/Sonja Bjørkman f 310322 ve int ve temp, focal, inj 1 /
xenon
37.8
12.7
0.04
1.8
1.8
142
600
0.50
0.07
210-7
int

```

```

132,89,61,46,38,30,26,22,20,19,
18,16,16,15,14,12,10,10,9,9,
9,8,8,7,7,7,6,6,6,5,
5,4,3,

```