

## Discussion and Results --- Circadian Rhythms Models programmed into DMSolver

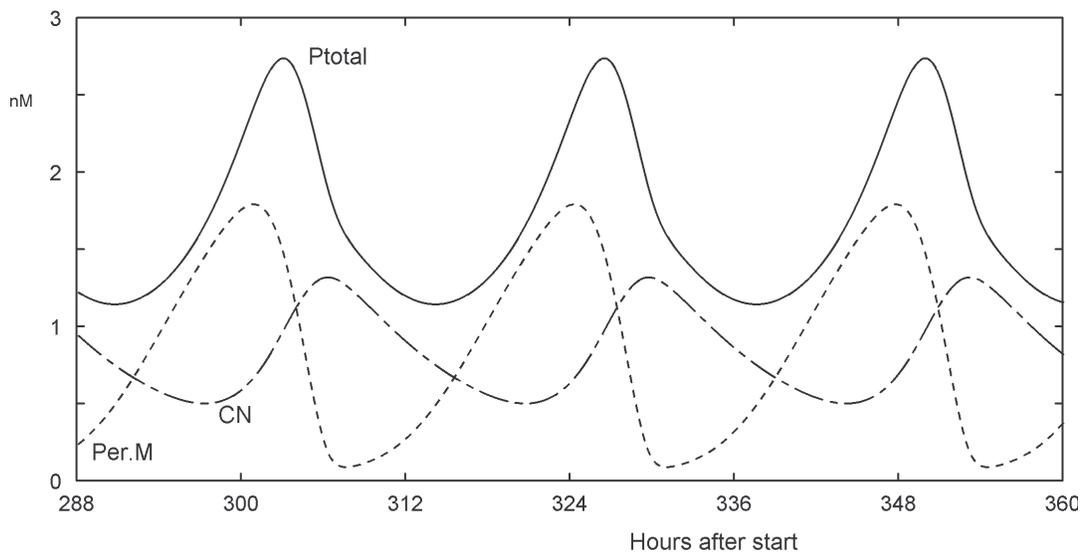
Copyright (c) 2010 F. D. Singleton -- all rights reserved

LeLoup et al<sup>1</sup> present a model for circadian rhythms in *Drosophila* (the fruit fly). There are two different types of protein involved, along with messenger RNA for these proteins, mass transfer within the cell, and other details which one needs to have biological training to appreciate fully. Our exercise simply involved taking the authors' equations and parameters as given (their equations 1a through 1j and 2) and expressing them as DMSolver code, and closely replicating the authors' graphs (figures 2A and 2B). The DMSolver program was run for two cases, and the results transferred to our EDTECH program for more finished-style plotting.

The results are shown in our figures 1 and 2.

If Figure 2 is compared with the original authors' Figure 2B the comparison is very close because in both cases the circadian rhythm is locked in the same way to the 24-hour light-dark cycle. On the other hand, our Figure 1 does not look exactly like the authors' 2A because the biological behavior is not locked exactly to a 24-hour period and the plotted results are shifted relative to the 24-hour time axis marks, the shift depending on how long the simulation was previously running.

Figure 1 --- Some model results for *Drosophila* in darkness replicating (except for a phase shift) Figure 2A of LeLoup et al<sup>1</sup>

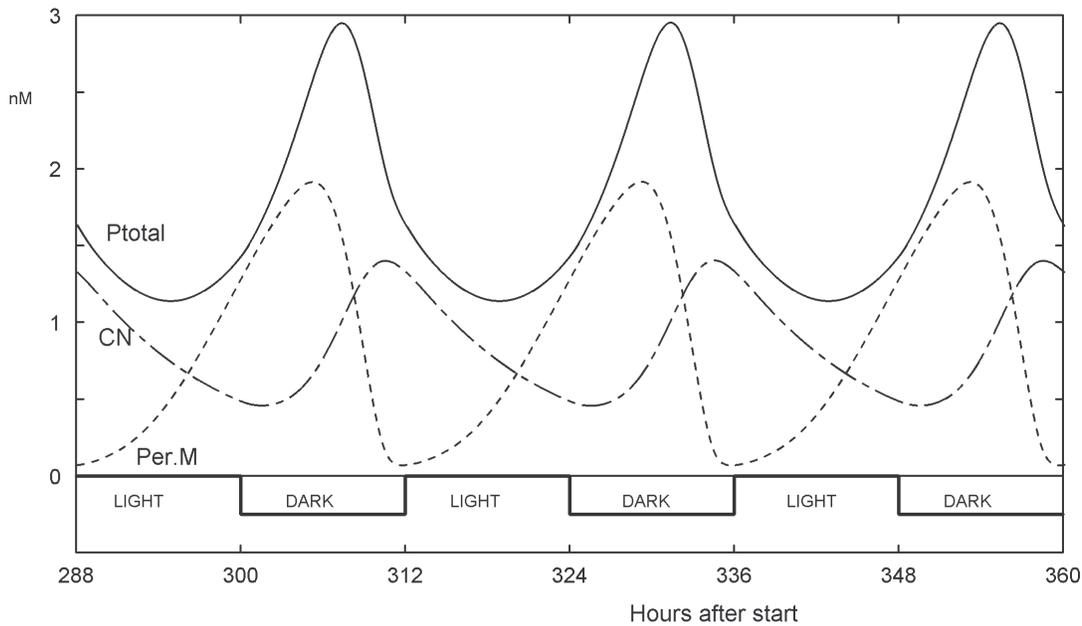


Ptotal = Total amount of PER protein

Per.M = per mRNA

CN = nuclear PER-TIM complex

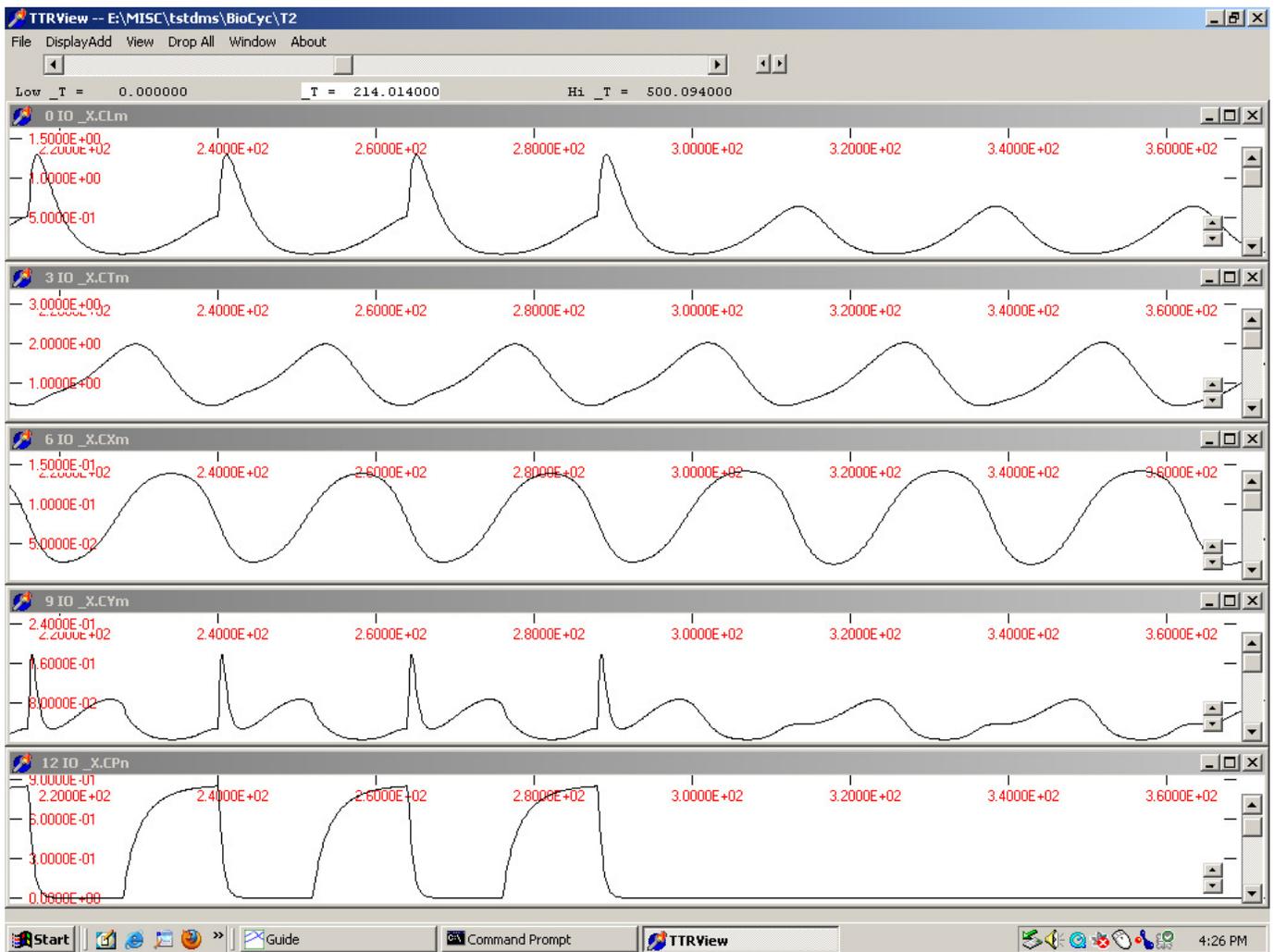
Figure 2 --- Some model results for *Drosophila* in periodic light/darkness replicating Figure 2B of LeLoup et al<sup>1</sup>



Ptotal = Total amount of PER protein  
 Per.M = per mRNA  
 CN = nuclear PER-TIM complex

In more recent circadian rhythm work, Locke et al<sup>2</sup> have studied rhythms in a small plant, *Arabidopsis thaliana*. The equations, parameters, and some graphical results are published in on-line appendices. Our DMSolver results closely agree with Locke et al. We present some results in "raw" form without EDTECH in Figure 3.

Figure 3 -- Some DMSolver results from a model by Locke et al<sup>2</sup> for circadian rhythms in the plant *Arabidopsis thaliana*



In the DMSolver display in Figure 3 five variables have been selected for display out of 30 possible by clicking on the variable list. The variable windows have been resized by the Window | Tile command, the time axis has been stretched using its scaling button and the time position has been moved out to the point of interest using its horizontal scroll bar. For times on the left half of the figure there are alternating 12-hr periods of light/dark, whereas on the right half the DMSolver TimeSet procedure was programmed to switch to continuous light. At "dawn" (beginning of light period) certain variables (1st and 4th ones displayed) show sharp peaks, which no longer happen when there is continuous light. The first and second plots in our Figure 3 correspond to curves in the Supplementary Figure 4 of Locke et al.

As with our EUROKIN report we include an appendix showing the DMSolver source code for the calculations. In the LeLoup et al case there were two proteins with identical sets of equations governing them, so we programmed one set of equations into a DMSolver module and called two instances of the module (on the other hand we could have programmed without modularity). In the Locke et al case there were detailed differences for the various proteins, and we did not try to use DMSolver modularity. If the equation-definition part of the source code is compared line-by-line with the authors' equation list the only

difference to be noted is that we identified 4 mass-transfer fluxes each of which appears in 2 different equations, and we decided to make these DMSolver simultaneous variables with their own procedure, RateDiff, rather than embed their calculations in the main list of rate equations (which could have been done).

## References

1. Leloup, J-C., Gonze, D., and Goldbeter, A., "Limit Cycle Models for Circadian Rhythms based on Transcriptional Regulation in *Drosophila* and *Neurospora*, *Journal of Biological Rhythms*, Vol. 14 No.6, 433-448 (1999)
2. Locke, J. CW., Southern, M. M., Kozma-Bognar, L., Hibberd, V., Brown, P. E., Turner, M. S., and Millar, A. J., "Extension of a genetic network model by iterative experimentation and mathematical analysis", *Molecular Systems Biology* (2005) doi:10.1038/msb4100018

## Appendix - The *Drosophila* model source code

```
(* LeLoup, Gonze, & Goldbeter, Limit Cycle Models for Circadian
Rhythms Based on Transcriptional Regulation in Drosophila
and Neurospora, J. of Biological Rhythms, 14, No.6 433-448 (1999) *)

TYPE
  PARSTY=RECORD
    vs,v1,K1,v2,K2,v3,K3,v4,K4,vm,Km,vd,Kd,kks,KL,KLtoN:SINGLE;
end;

VAR
  kk1,kk2,kk3,kk4,kkd,kkdC,kkdN,Nexp:SINGLE;
  Ppar,Tpar:PARSTY;

function AXB(A,B:DOUBLE):DOUBLE;
begin
  AXB:=AEXP(B*ALOG(A));
end;

PROCEDURE SETUP;
begin
  Ppar.KLtoN:=AXB(Ppar.KL,Nexp);
  Tpar.KLtoN:=AXB(Tpar.KL,Nexp);
end;

PROCEDURE TIMESET; { generates 24 hr light/dark cycles }
VAR NcyCs:INTEGER;
begin
  NcyCs:=Trunc(_T/24);
  IF (_T-24*NcyCs)<12 THEN Tpar.vd:=6 ELSE Tpar.vd:=3;
end;

PROCEDURE REPORT;
begin
  { no report }
end;

PROCEDURE PARSET; {optional-generates a GUI parameter editing window}
begin
  _PARS(3,'Per Rate const s','nM/h','Ppar.vs',Ppar.vs,1.1);
  _PARS(3,'Per Rate const 1','nM/h','Ppar.v1',Ppar.v1,8);
  _PARS(3,'Per Equil const 1','nM','Ppar.K1',Ppar.K1,2);
  _PARS(3,'Per Rate const 2','nM/h','Ppar.v2',Ppar.v2,1);
  _PARS(3,'Per Equil const 2','nM','Ppar.K2',Ppar.K2,2);
  _PARS(3,'Per Rate const 3','nM/h','Ppar.v3',Ppar.v3,8);
  _PARS(3,'Per Equil const 3','nM','Ppar.K3',Ppar.K3,2);
  _PARS(3,'Per Rate const 4','nM/h','Ppar.v4',Ppar.v4,1);
  _PARS(3,'Per Equil const 4','nM','Ppar.K4',Ppar.K4,2);
  _PARS(3,'Per Rate const m','nM/h','Ppar.vm',Ppar.vm,1.0);
  _PARS(3,'Per Equil const m','nM','Ppar.Km',Ppar.Km,0.2);
  _PARS(3,'Per Rate const d','nM/h','Ppar.vd',Ppar.vd,2.2);
  _PARS(3,'Per Equil const d','nM','Ppar.Kd',Ppar.Kd,0.2);
  _PARS(3,'Per Diff rate const s','/h','Ppar.kks',Ppar.kks,0.9);
  _PARS(3,'Per Equil const L','nm','Ppar.KL',Ppar.KL,1.0);

  _PARS(3,'Tim Rate const s','nM/h','Tpar.vs',Tpar.vs,1.0);
  _PARS(3,'Tim Rate const 1','nM/h','Tpar.v1',Tpar.v1,8);
  _PARS(3,'Tim Equil const 1','nM','Tpar.K1',Tpar.K1,2);
  _PARS(3,'Tim Rate const 2','nM/h','Tpar.v2',Tpar.v2,1);
  _PARS(3,'Tim Equil const 2','nM','Tpar.K2',Tpar.K2,2);
  _PARS(3,'Tim Rate const 3','nM/h','Tpar.v3',Tpar.v3,8);
  _PARS(3,'Tim Equil const 3','nM','Tpar.K3',Tpar.K3,2);
  _PARS(3,'Tim Rate const 4','nM/h','Tpar.v4',Tpar.v4,1);
  _PARS(3,'Tim Equil const 4','nM','Tpar.K4',Tpar.K4,2);
  _PARS(3,'Tim Rate const m','nM/h','Tpar.vm',Tpar.vm,0.7);
  _PARS(3,'Tim Equil const m','nM','Tpar.Km',Tpar.Km,0.2);
  _PARS(3,'Tim Rate const d','nM/h','Tpar.vd',Tpar.vd,3);
```

```

_PARS(3,'Tim Equil const d','nM','Tpar.Kd',Tpar.Kd,0.2);
_PARS(3,'Tim Diff rate const s','/h','Tpar.kks',Tpar.kks,0.9);
_PARS(3,'Tim Equil const L','nm','Tpar.KL',Tpar.KL,1.0);

_PARS(3,'Diff rate const 1','/h','kk1',kk1,0.8);
_PARS(3,'Diff rate const 2','/h','kk2',kk2,0.2);
_PARS(3,'Diff rate const 3','/h','kk3',kk3,1.2);
_PARS(3,'Diff rate const 2','/h','kk4',kk4,0.6);
_PARS(3,'Diff rate const','/h','kkd',kkd,0.01);
_PARS(3,'Diff rate const','/h','kkdC',kkdC,0.01);
_PARS(3,'Diff rate const','/h','kkdN',kkdN,0.01);
_PARS(1,'Exponent in M eqn','','Nexp',Nexp,4);

end;

function RateR(Vp,Kp:SINGLE; Q:DOUBLE):DOUBLE;
VAR Denom:DOUBLE;
begin
  Denom:=Kp+Q;
  IF Denom<1.0E-06 THEN begin RateR:=0; ERRINT end
  ELSE RateR:=Vp*Q/Denom;
end;

PROCEDURE ADDSET;
begin
  { not needed }
end;

PROCEDURE USERBPR;
begin
  { no user batch calls }
end;

XPROC MdotCalc(VAROT Md:XVR; VARIN Pn,M:XVR; vs,KLtoN,vm,Km:SINGLE);
begin
  Md:= vs*KLtoN/(KLtoN+AXB(Pn,Nexp)) - RateR(vm,Km,M) - kkd*M;
end;

XPROC TotalCalc(VAROT T:XVR; VARIN A,B,C,D:XVR);
begin
  T:=A+B+C+D;
end;

XPROC Add2(VAROT T:XVR; VARIN A,B:XVR);
begin
  T:=A+B;
end;

XPROC PrateCalc(VAROT Prate:XVR; VARIN PP0,PP1:XVR; Vp1,Kp1,Vp2,Kp2:SINGLE);
begin
  Prate:=RateR(Vp1,Kp1,PP0) - RateR(Vp2,Kp2,PP1);
end;

XPROC P0dCalc(VAROT P0deriv:XVR; VARIN MM,PR,PP0:XVR; kks:SINGLE);
begin
  P0deriv:=kks*MM - PR - kkd*PP0;
end;

XPROC P1dCalc(VAROT P1deriv:XVR; VARIN PR1,PR2,PP1:XVR);
begin
  P1deriv:=PR1 - PR2 - kkd*PP1;
end;

XPROC P2dCalc(VAROT P2deriv:XVR; VARIN PR2,PP2,PRN:XVR; vd,Kd:SINGLE);
begin
  P2deriv:=PR2 - PRN - RateR(vd,Kd,PP2) - kkd*PP2;
end;

```

```

MPROC protCalc(PID:MODID; VAR P2,Ptotal,PRN,CCN:XVR; VAR prp:PARSTY); {module for a protein}
VAR
  M,pr0,pr1,Mdot,pr0dot,pr1dot,pr2dot,prrate1,prrate2:XVR;
begin
  MdotCalc(Mdot,CCN,M, prp.vs,prp.KLtoN,prp.vm,prp.Km);
  PrateCalc(prrate1,pr0,pr1,prp.v1,prp.K1,prp.v2,prp.K2);
  PrateCalc(prrate2,pr1,P2,prp.v3,prp.K3,prp.v4,prp.K4);
  P0dCalc(pr0dot,M,prrate1,pr0,prp.kks);
  P1dCalc(pr1dot,prrate1,prrate2,pr1);
  P2dCalc(pr2dot,prrate2,P2,PRN,prp.vd,prp.Kd);
  PMSINTEG(M,Mdot);
  PMSINTEG(pr0,pr0dot);
  PMSINTEG(pr1,pr1dot);
  PMSINTEG(P2,pr2dot);
  TotalCalc(Ptotal,pr0,pr1,P2,CCN);
end;

XPROC CrossTermCalc(VAROT PRN:XVR; VARIN PP2,TT2,CC:XVR);
begin
  PRN:=kk3*PP2*TT2 - kk4*CC;
end;

XPROC CderivCalc(VAROT Cderiv:XVR; VARIN PRN,CC,CCN:XVR);
begin
  Cderiv:=PRN - kk1*CC + kk2*CCN - kkdC*CC;
end;

XPROC CNderivCalc(VAROT CNderiv:XVR; VARIN CC,CCN:XVR);
begin
  CNderiv:= kk1*CC - kk2*CCN - kkdN*CCN;
end;

MPROC SYSTEM;
VAR
  P2,T2,C,CN,PTcross,Cdot,CNdot,Psubt,Tsubt,Ptotal,Ttotal:XVR;
begin
  protCalc('Per',P2,Psubt,PTcross,CN,Ppar);
  protCalc('Tim',T2,Tsubt,PTcross,CN,Tpar);
  CrossTermCalc(PTcross,P2,T2,C);
  CderivCalc(Cdot,PTcross,C,CN);
  CNderivCalc(CNdot,C,CN);
  PMSINTEG(C,Cdot);
  PMSINTEG(CN,CNdot);
  Add2(Ptotal,Psubt,C);
  Add2(Ttotal,Tsubt,C);
end;

```

**NOTE:** Variables `Psubt`, `Tsubt`, `Ptotal`, and `Ttotal` could be left out of the model as well as procedures `Add2` and `Totalcalc` if the user wanted to compute the totals `Ptotal` and `Ttotal` later from the output data. It was convenient for output display purposes to compute them as shown, but small amounts of CPU time might be shaved from the calculation by leaving them out of the model e.g. if the user was doing a parameter estimation.

## Appendix --- the *Arabidopsis thaliana* model by Locke et al<sup>2</sup>

```
VAR
  a,b,c,d,e,f,g1,g2,g3,g4,g5,g6,k1,k10,k11,k12,k13,k2,k3,k4,k5,k6,k7,k8,k9
  ,m1,m10,m11,m12,m13,m14,m15,m2,m3,m4,m5,m6,m7,m8,m9,n1,n2,n3,n4,n5,p1,p2
  ,p3,p4,p5,q1,q2,q3,r1,r2,r3,r4,r5,r6,r7,r8
  ,Theta,UMTheta,g1a,g2b,g3c,g4d,g5e,g6f,CycleHr,DarkStart:SINGLE;

function AXB(A,B:DOUBLE):DOUBLE;
begin
  IF A<=1.0E-25 THEN AXB:=0.0
  ELSE AXB:=EXP(B*LN(A));
end;

PROCEDURE SETUP;
begin
  g1a:=axb(g1,a);
  g2b:=axb(g2,b);
  g3c:=axb(g3,c);
  g4d:=axb(g4,d);
  g5e:=axb(g5,e);
  g6f:=axb(g6,f);
end;

PROCEDURE TIMESET; {36 light/dark cycles, then continuous light }
VAR NCycles:INTEGER; ThisC:DOUBLE;
begin
  NCycles:=Trunc( T/CycleHr);           {Theta=1=light; Theta=0=dark}
  ThisC:=NCycles*CycleHr;
  IF (( T-ThisC)<DarkStart)OR(NCycles>=36) THEN Theta:=1 ELSE Theta:=0;
  UMTheta:=1.0-Theta;
end;

PROCEDURE REPORT;
begin
  { Optional reporting statements here }
end;

PROCEDURE PARSET; {optional-generates a GUI parameter editing window}
  {Parameter default values & descriptions were copied & pasted from Locke et al file}
begin
  _pars(2,'Cycle length','h','CycleHr',CycleHr,24);
  _pars(2,'Start of dark period from beginning of cycle','h','DarkStart',DarkStart,12);
  _pars(4,'Hill coefficient of activation by protein X',' ','a',a,3.3064);
  _pars(4,'Hill coefficient of activation by protein Y',' ','b',b,1.0258);
  _pars(4,'Hill coefficient of repression by LHY',' ','c',c,1.0258);
  _pars(4,'Hill coefficient of activation by TOC1',' ','d',d,1.4422);
  _pars(4,'Hill coefficient of repression by TOC1',' ','e',e,3.6064);
  _pars(4,'Hill coefficient of repression by LHY',' ','f',f,1.0237);
  _pars(4,'Constant of activation by protein X','nM','g1',g1,0.8767);
  _pars(4,'Constant of activation by protein Y','nM','g2',g2,0.0368);
  _pars(4,'Constant of repression by LHY','nM','g3',g3,0.2658);
  _pars(4,'Constant of activation by TOC1','nM','g4',g4,0.5388);
  _pars(4,'Constant of repression by TOC1','nM','g5',g5,1.178);
  _pars(4,'Constant of repression by LHY','nM','g6',g6,0.0645);
  _pars(4,'Michaelis const-LHY mRNA degradation','nM','k1',k1,1.817);
  _pars(4,'Michaelis const-protein Y mRNA degradation','nM','k10',k10,1.7303);
  _pars(4,'Michaelis const-cytoplasm protein Y degradation','nM','k11',k11,1.8258);
  _pars(4,'Michaelis const-nuclear protein Y degradation','nM','k12',k12,1.8066);
  _pars(4,'Michaelis const-protein P degradation','nM','k13',k13,1.2);
  _pars(4,'Michaelis const-cytoplasmic LHY degradation','nM','k2',k2,1.5644);
  _pars(4,'Michaelis const-nuclear LHY degradation','nM','k3',k3,1.2765);
  _pars(4,'Michaelis const-TOC mRNA degradation','nM','k4',k4,2.5734);
  _pars(4,'Michaelis const-cytoplasm TOC1 degradation','nM','k5',k5,2.7454);
  _pars(4,'Michaelis const-nuclear TOC1 degradation','nM','k6',k6,0.4033);
  _pars(4,'Michaelis const-protein X mRNA degradation','nM','k7',k7,6.5585);
  _pars(4,'Michaelis const-cytoplasm protein X degradation','nM','k8',k8,0.6632);
  _pars(4,'Michaelis const-nuclear protein X degradation','nM','k9',k9,17.1111);
```

```

_pars(4,'Max rate of LHY mRNA degradation','nM/h','m1',m1,1.5283);
_pars(4,'Max rate of degradation of cytoplasmic protein X','nM/h','m10',m10,0.2179);
_pars(4,'Max rate of degradation of nuclear protein X','nM/h','m11',m11,3.3442);
_pars(4,'Max rate of degradation of protein Y mRNA','nM/h','m12',m12,4.297);
_pars(4,'Max rate of degradation of cytoplasm protein Y','nM/h','m13',m13,0.1347);
_pars(4,'Max rate of degradation of nuclear protein Y','nM/h','m14',m14,0.6114);
_pars(4,'Max rate of protein P degradation','nM/h','m15',m15,1.2);
_pars(4,'Max rate of cytoplasmic LHY degradation','nM/h','m2',m2,20.44);
_pars(4,'Max rate of nuclear LHY degradation','nM/h','m3',m3,3.6888);
_pars(4,'Max rate of TOC mRNA degradation','nM/h','m4',m4,3.8231);
_pars(4,'Max rate -light dep cytoplasm TOC1 degradation','nM/h','m5',m5,0.0013);
_pars(4,'Max rate -light independent cytopl TOC1 degrad','nM/h','m6',m6,3.1741);
_pars(4,'Max rate -light dependent nuclear TOC1 degrad','nM/h','m7',m7,0.0492);
_pars(4,'Max rate -light independent nuclear TOC1 degrad','nM/h','m8',m8,4.0424);
_pars(4,'Max rate of degradation of protein X mRNA','nM/h','m9',m9,10.1132);
_pars(4,'Max light-independent LHY transcription rate','nM/h','n1',n1,5.1694);
_pars(4,'Max TOC1 transcription rate','nM/h','n2',n2,3.0087);
_pars(4,'Max transcription rate of protein X','nM/h','n3',n3,0.2431);
_pars(4,'Light dependent component of Y transcription','nM/h','n4',n4,0.0857);
_pars(4,'Light independent component of Y transcription','nM/h','n5',n5,0.1649);
_pars(4,'Rate constant of LHY mRNA translation','1/h','p1',p1,0.8295);
_pars(4,'Rate constant of TOC1 mRNA translation','1/h','p2',p2,4.324);
_pars(4,'Rate constant of X mRNA translation','1/h','p3',p3,2.147);
_pars(4,'Rate constant of Y mRNA translation','1/h','p4',p4,0.2485);
_pars(4,'Light dependent production of protein P','nM/h','p5',p5,0.5);
_pars(4,'Coupling const - light activation of LHY transcr','1/h','q1',q1,2.4514);
_pars(4,'Coupling const-light activation of Y mRNA transcr','1/h','q2',q2,2.4017);
_pars(4,'Coupling const-light activation - protein P degr','1/h','q3',q3,1);
_pars(4,'Rate constant - LHY transport into nucleus','1/h','r1',r1,16.8363);
_pars(4,'Rate constant - LHY transport out of nucleus','1/h','r2',r2,0.1687);
_pars(4,'Rate constant - TOC1 movement into nucleus','1/h','r3',r3,0.3166);
_pars(4,'Rate constant - TOC1 movement out of nucleus','1/h','r4',r4,2.1509);
_pars(4,'Rate constant - protein X movemnt into nucleus','1/h','r5',r5,1.0352);
_pars(4,'Rate constant - protein X movemnt out of nucleus','1/h','r6',r6,3.3017);
_pars(4,'Rate constant - protein Y movemnt into nucleus','1/h','r7',r7,2.2123);
_pars(4,'Rate constant - protein Y movemnt out of nucleus','1/h','r8',r8,0.2002);
end;

```

```
PROCEDURE ADDSET;
```

```
begin
  {nothing}
end;
```

```
PROCEDURE USERBPR;
```

```
begin
  { nothing}
end;
```

```
{misc functions}
```

```
function KRatio(X:DOUBLE; K:SINGLE):DOUBLE;
```

```
VAR Denom:DOUBLE;
```

```
begin
  Denom:=X+K;
  IF Denom<1.0E-08 THEN
  begin
    ERRINT;
    KRatio:=0;
  end
  ELSE KRatio:=X/Denom;
end;
```

```

function KRatioE(X:DOUBLE; KE,E:SINGLE):DOUBLE;
VAR Denom,XE:DOUBLE;
begin
  XE:=AXB(X,E);
  Denom:=XE+KE;
  IF Denom<1.0E-08 THEN
    begin
      ERRINT;
      KRatioE:=0;
    end
  ELSE KRatioE:=XE/Denom;
end;

XPROC RateDiff(VAROT NetRt:XVR; VARIN Ccyt,Cnuc:XVR; Rcyt,Rnuc:SINGLE);
begin
  NetRt := Rcyt*Ccyt - Rnuc*Cnuc;
end;

XPROC RCLM(VAROT Rate:XVR; VARIN Cpn,Cxn,Clm:XVR);
begin
  Rate := Theta*q1*Cpn + n1*KRatioE(Cxn,g1a,a)-m1*KRatio(Clm,k1);
end;

XPROC RCLC(VAROT Rate:XVR; VARIN CNT,Clm,Clc:XVR);
begin
  Rate := p1*Clm - CNT -m2*KRatio(Clc,k2);
end;

XPROC RCLN(VAROT Rate:XVR; VARIN CNT,Cln:XVR);
begin
  Rate := CNT - m3*KRatio(Cln,k3);
end;

XPROC RCTM(VAROT Rate:XVR; VARIN Cyn,Cln,Ctm:XVR);
begin
  Rate := n2*KRatioE(Cyn,g2b,b)*g3c/(g3c+AXB(Cln,c)) - m4*KRatio(Ctm,k4);
end;

XPROC RCTC(VAROT Rate:XVR; VARIN CNT,Ctm,Ctc:XVR);
begin
  Rate := p2*Ctm - CNT - (UMTheta*m5 + m6)*KRatio(Ctc,k5);
end;

XPROC RCTN(VAROT Rate:XVR; VARIN CNT,Ctn:XVR);
begin
  Rate := CNT - (UMTheta*m7 + m8)*KRatio(Ctn,k6);
end;

XPROC RCXM(VAROT Rate:XVR; VARIN Ctn,Cxm:XVR);
begin
  Rate := n3*KRatioE(Ctn,g4d,d) - m9*KRatio(Cxm,k7);
end;

XPROC RCXC(VAROT Rate:XVR; VARIN CNT,Cxm,Cxc:XVR);
begin
  Rate := p3*Cxm - CNT - m10*KRatio(Cxc,k8);
end;

XPROC RCXN(VAROT Rate:XVR; VARIN CNT,Cxn:XVR);
begin
  Rate := CNT - m11*KRatio(Cxn,k9);
end;

XPROC RCYM(VAROT Rate:XVR; VARIN Cpn,Ctn,Cln,Cym:XVR);
begin
  Rate := (Theta*q2*Cpn +(Theta*n4 + n5)*g5e/(g5e+AXB(Ctn,e)))*g6f/(g6f+AXB(Cln,f))
    - m12*KRatio(Cym,k10);
end;

```

```

XPROC RCYC (VAROT Rate:XVR; VARIN Cym,CNT,Cyc:XVR);
begin
  Rate := p4*Cym - CNT - m13*KRatio(Cyc,k11);
end;

XPROC RCYN (VAROT Rate:XVR; VARIN CNT,Cyn:XVR);
begin
  Rate := CNT - m14*KRatio(Cyn,k12);
end;

XPROC RCPN (VAROT Rate:XVR; VARIN Cpn:XVR);
begin
  Rate := UMTheta*p5 - m15*KRatio(Cpn,k13) - q3*Theta*Cpn;
end;

MPROC SYSTEM;
VAR
  CLm,CLc,CLn,CTm,CTc,CTn,CXm,CXc,CXn,CYm,CYc,CYn,CPn
  ,CLmdot,CLcdot,CLndot,CTmdot,CTcdot,CTndot,CXmdot
  ,CXcdot,CXndot,CYmdot,CYcdot,CYndot,CPndot
  ,CNTCL,CNTCT,CNTCX,CNTCY:XVR;
begin
  RateDiff (CNTCL,CLc,CLn,r1,r2);
  RateDiff (CNTCT,CTc,CTn,r3,r4);
  RateDiff (CNTCX,CXc,CXn,r5,r6);
  RateDiff (CNTCY,CYc,CYn,r7,r8);
  RCLM (CLmdot,CPn,CXn,CLm);
  RCLC (CLcdot,CNTCL,CLm,CLc);
  RCLN (CLndot,CNTCL,CLn);
  RCTM (CTmdot,CYn,CLn,CTm);
  RCTC (CTcdot,CNTCT,CTm,CTc);
  RCTN (CTndot,CNTCT,CTn);
  RCXM (CXmdot,CTn,CXm);
  RCXC (CXcdot,CNTCX,CXm,CXc);
  RCXN (CXndot,CNTCX,CXn);
  RCYM (CYmdot,CPn,CTn,CLn,CYm);
  RCYC (CYcdot,CYm,CNTCY,CYc);
  RCYN (CYndot,CNTCY,CYn);
  RCPN (CPndot,CPn);
  pmsInteg (CLm,CLmdot);
  pmsInteg (CLc,CLcdot);
  pmsInteg (CLn,CLndot);
  pmsInteg (CTm,CTmdot);
  pmsInteg (CTc,CTcdot);
  pmsInteg (CTn,CTndot);
  pmsInteg (CXm,CXmdot);
  pmsInteg (CXc,CXcdot);
  pmsInteg (CXn,CXndot);
  pmsInteg (CYm,CYmdot);
  pmsInteg (CYc,CYcdot);
  pmsInteg (CYn,CYndot);
  pmsInteg (CPn,CPndot);
end;

```