```
SetDirectory["C:\\Users\\bbaiser\\Documents\\currenthfdesk\\thresholds\\model"];
```

Photosynthetic Square-Sine Wave

```
sine[A_, F_, PSI_, time_, D_] := Module[{recov, trunk},
   recov = A * N[Sin[(2 * Pi * F * time + PSI)]] + D];
```

sine wave over 16 days (23,040 min)##

```
SineWave = Map[sine[2500, .0006944444444444445, 0, #, 0] &, Table[i, {i, 1, 23 040}]];
```

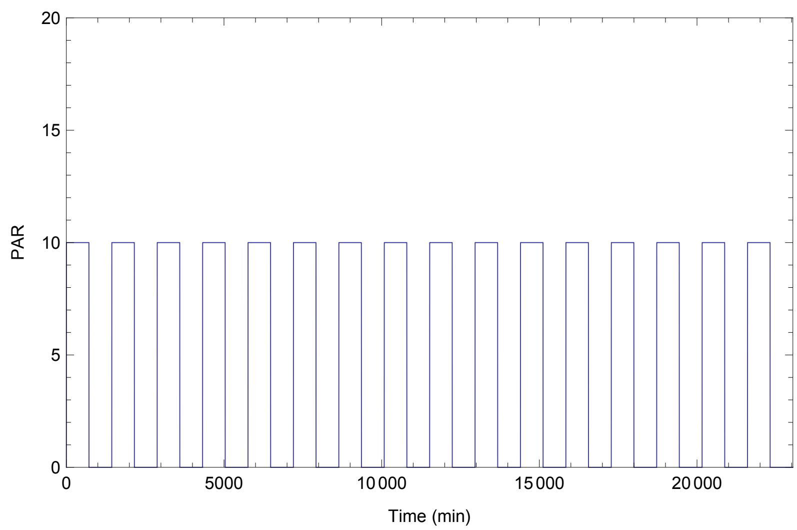
##change negative values to zero##

```
SineNoNeg = Map[If[# ≤ 0, 0, #] &, SineWave];
```

##modify to squre sine wave##

```
SqSine = Map[If[# ≥ 20, 10 * (1 - E^(-.3 * (# - 20))), 0] &, SineNoNeg];
```

```
ListLinePlot[SqSine, PlotRange → {{0, 23 040}, {0, 20}},
 Frame → True, FrameLabel -> {"Time (min)", "PAR"}]
```



Biological Oxygen Demand (BOD) curve

```
BOD[prey_, h_] := Module[{shit},
  shit = 1 + 10 * (prey / (prey + h));
  shit]
```

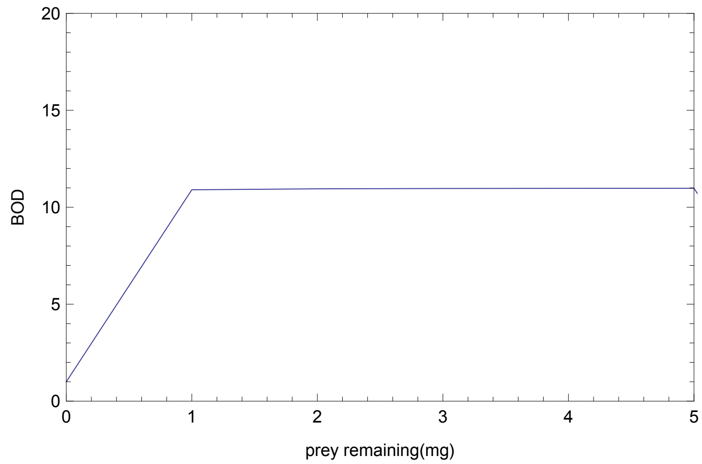
######################Test of BOD curve at h = 0.01######################

```
BODcurve = Map[BOD[#, .01] &, Table[i, {i, 0, 5}]]
```

```
{1., 10.901, 10.9502, 10.9668, 10.9751, 10.98}
```

```
coupled = Partition[Riffle[ Table[i, {i, 0, 10}], BODcurve], 2];
```

```
ListLinePlot[coupled, PlotRange → {{0, 5}, {0, 20}},
 Frame → True, FrameLabel -> {"prey remaining(mg)", "BOD"}]
```



Prey Consumption Curve

```
preyConsumptionCurve[a_, b_, t_] := Module[{curve},
  curve = a * E^(-b * t)]
```

####The soulution function solves pcurve for any prey (chew) value#####

```
solutionFunction[chew_, day_] := Round[N[t /. Solve[pcurve[20, 4, t] == chew]] * day]
```

```
Round[N[t /. Solve[preyConsumptionCurve[20, 4, t] == 5]][[1]] * 1440];
```

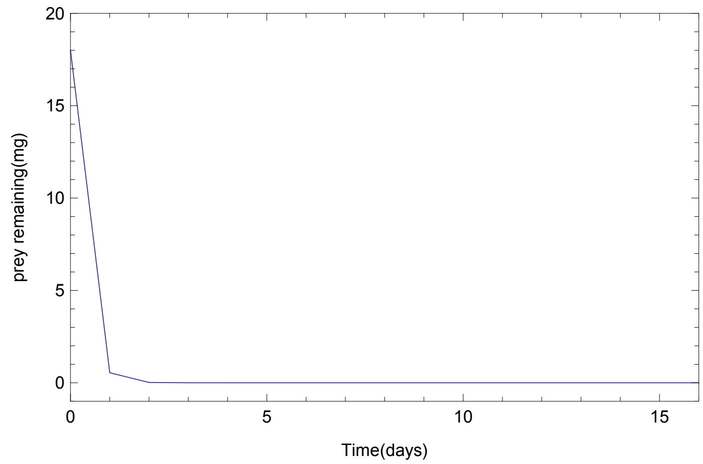
Solve::ifun :
 Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. >>

```
Testpcurve = N[Map[preyConsumptionCurve[18, 3.5, #] &,
   {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}]]
```

$\{18., 0.543553, 0.0164139, 0.000495656, 0.0000149675, 4.5198\times10^{-7}, 1.36486\times10^{-8}, 4.12152\times10^{-10}, 1.24459\times10^{-11}, 3.75834\times10^{-13}, 1.13492\times10^{-14}, 3.42716\times10^{-16}, 1.03491\times10^{-17}, 3.12517\times10^{-19}, 9.43719\times10^{-21}, 2.84979\times10^{-22}, 8.60561\times10^{-24}\}$

```
plot = Partition[
   Riffle[{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}, Testpcurve], 2];
```

```
ListLinePlot[plot, PlotRange → {{0, 16}, {-1, 20}},
 Frame → True, FrameLabel -> {"Time(days)", "prey remaining(mg)"}]
```



Nitrogen as a function of prey and oxygen

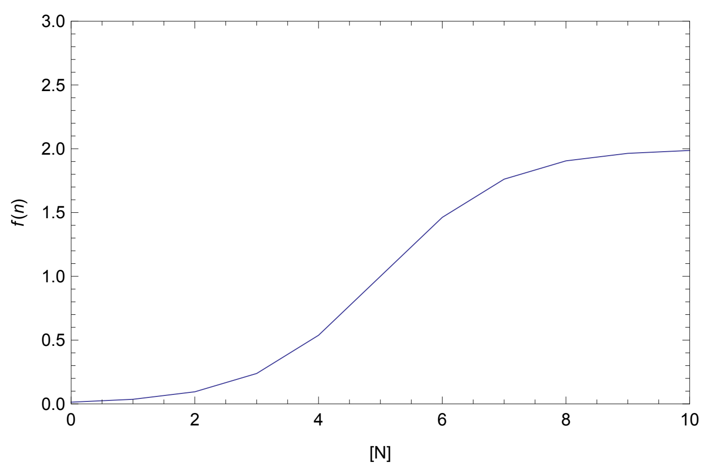
Oxygen augmentation to the sine curve as a function of mineralized N

```
sig[MO_, MX_, a_, x_, d_] := Module[{one, two, three},
  one = a (x - d);
  two = N[E^one];
  three = (MO) + ((MX - MO) / (1 + two));
  three]
```

```
feedback = Map[sig[0, 2, -1, #, 5] &, {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}];
```

```
coupled = Partition[Riffle[{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, feedback], 2];
```

```
ListLinePlot[coupled, PlotRange → {{0, 10}, {0, 3}},
 Frame → True, FrameLabel -> {"[N]", "f(n)"}]
```



#####Augmentation Function#####

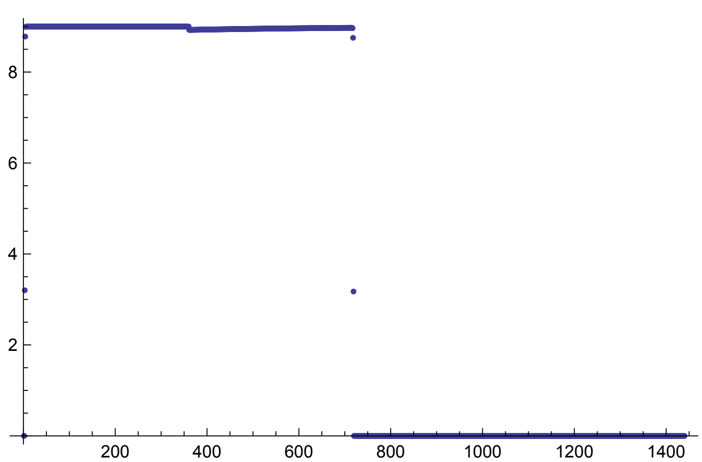
```
Aug[prey_, oxy_, slope_] := Module[{Aug, N},
  N = Nitrogen[prey, oxy];
  Aug = sig[0, 2, slope, N, 5];
  Aug]
```

Dynamics of entire model for 1 day (1440 min)

```
Day[{previousChow_, preyloss_, previousSpot_, prevO2_},
  addchow_, lastday_, h_, Aug_, b_, a_] :=
 Module[{index, o2, newChow, sin, nullsin, bod, remove, nullo2, tfake, curveTime2,
   curveTime, t, chew, left, BigcurveTime, BigcurveTime2, ppChow2, parO2},
  index = previousSpot + 1;
  sin = sine[2500 * Aug, .0006944444444444445, 0, index, 0];
  nullsin = If[sin ≤ 0, 0, sin];
  parO2 = If[nullsin ≥ 20, 10 * Aug * (1 - E^(-.3 * (nullsin - 20))), 0];
  ppChow2 = If[index == 360, addchow + previousChow, previousChow];
  BigcurveTime2 = If[index < 360,
    Round[N[t /. Solve[preyConsumptionCurve[a, b, t] == lastday]][[1]] * 1440], Round[
     N[t /. Solve[preyConsumptionCurve[a, b, t] == addchow + lastday]][[1]] * 1440]];
  BigcurveTime = If[BigcurveTime2 < 0, 0, BigcurveTime2];
  tfake = If[ppChow2 == 0, BigcurveTime, BigcurveTime + index];
  remove = N[preyConsumptionCurve[a, b, BigcurveTime * 0.0006944444444444445]] -
    N[preyConsumptionCurve[a, b, tfake * 0.0006944444444444445]];
  left = If[remove == 0, ppChow2, preyConsumptionCurve[
     a, b, tfake * 0.0006944444444444445]];
  bod = BOD[left, h];
  o2 = parO2 - bod;
  nullo2 = If[o2 < 0, 0, o2];
  {left, remove, index, nullo2}]

O2Dynamics = NestList[Day[#, .1, 0, 5, 1, 4, 20] &,
   {0, 0, 0, sine[2500 * 1, .0006944444444444445, 0, 0, 0]}, 1439];

or = ListPlot[Map[#[[4]] &, O2Dynamics]]
```



Looping theDynamics for all 16 days

```
dayPlus[{Augment_, old_, lastday_}, addchow_, b_, a_, h_, slope_] :=
 Module[{O2Dynamics, outs, aug, finalO2, nchow, leftoverChow},
  O2Dynamics = NestList[Day[#, addchow, lastday, h, Augment, b, a] &,
    {lastday, 0, 0, sine[2500 * Augment, .0006944444444444445, 0, 0, 0]}, 1439];
  outs = O2Dynamics[[All, {1, 2, 4}]];
  finalO2 = outs[[1440]][[3]];
  nchow = outs[[1440]][[2]];
  leftoverChow = outs[[1440]][[1]];
  aug = Aug[nchow, finalO2, slope] + Augment;
  {aug, outs, leftoverChow}]

run = dayPlus[{1, 1, 0}, 5, 4, 20, .1, -1];
```

Parameters

```
mParams =
  MapThread[Tuples[{{5, 1, 0}, {#1}, {#2}, {.1, .01, .001}, {-1, -2, -3, -4, -5}}] &,
   {{8, 4, 2, 1}, {40, 20, 10, 5}}];
Dimensions[mParams];
fParams = Partition[Flatten[mParams], 5];
Dimensions[fParams];
```

run model over parameter space

```
runz[{addchow_, b_, a_, h_, slope_}] := Module[{oxygen, fr, onlyO2},
  oxygen = NestList[dayPlus[#, addchow, b, a, h, slope] &, {1, 1, 0}, 16];
  fr = Partition[Flatten[Map[oxygen[[#]][[2]] &, Table[i, {i, 2, 17}]]], 3];
  onlyO2 = Map[#[[3]] &, fr];
  onlyO2]

Output = Map[runz[#] &, fParams[[1 ;; 2]]];

Export["n.csv", Transpose[Output]]
```

Moving Averages (ten = 10 minute window, hundred = 100 min window

```
ten = Map[MovingAverage[#, 10] &, Output];

hundred = Map[MovingAverage[#, 100] &, Output];
```