**Model 2 Code (current version as of June 29, 2022)**

**Contents: This file has Metabolites, Fluxes (p 3), Indices (p 5), a ModelRun script (p6), and a pasted shot of the loading values. This code was used to run simulations on June 28, 2022.**

**Metabolites**

% Defining Global Variables

global iFAT iCHO iGr iGh iGA iPCR

% Defining local variables

FAT = x(iFAT);

CHO = x(iCHO);

Gr = x(iGr);

Gh = x(iGh);

GA = x(iGA);

PCR = x(iPCR);

%Defining parameters (Vmax of DH & proton leak; conductances of DH, ETC, VA; velocity of ATPase)

BOX = param(1); %conductance of the Beta oxidation pathway into matrix redox tank

ETC = param(2); %conductance of the ETC (= J/(Gr-Gh))

VA = param(3); %conductance of Comp V + ANT (= J/(Gh-GA))

ATPase = param(4); %Rate of ATP breakdown (in amps and 1.0 amp ~ 10.4 umol ATP/sec)

pleak = param(5); %conductance of proton leak

eleak = param(6); %conductance of electron leak (= J/Gr)

CK = param(7); %high conductance pipe (CKase) connecting PCr and ATP tanks

Cpdh = param(13); %Vmax in m^3 H2O per sec

Pa = 9806; %converts tank height to Pa of driving pressure

% Computing flux into fuel source

J\_fuel = 0;

% Computing beta oxidation flux of lipid into matrix redox potential

J\_BOX = Pa\*(FAT-Gr)\*BOX;

% Computing PDH flux of CHO into matrix redox potential

Cpdh = param(13); %Conductance of glycolytic carbon entry into matrix redox

CHOSat = param(14); %Gives status of CHO availability (0 - 1.0)

Act = (8-GA)^3.5/(8-5)^3.5; %Delta Gatp related activation term of glycolysis

Inhib = (8-Gr)^0.25/(8-5)^0.25; %Delta G redox related feedback inhibition of PDH

J\_pdh = Pa\*(CHO-Gr)\* Cpdh \* Act \* CHOSat \* Inhib;

% Computing movement of electrons into proton motive force

J\_ETC = Pa\*(Gr-Gh)\*ETC;

% Computing flux of protons into ATP Synthesis & Export

KmADP = 0.5;

nH = 2;

VAkin = (8-GA)^nH/((8-GA)^nH+KmADP^nH); %M-M relation with Hill nH and "KmADP" = 0.5 meters below 8

J\_VA = Pa\*(Gh-GA)\*VA\*VAkin;

% Computing flux of PCr into ATP

J\_CK = Pa\*(PCR-GA)\*CK;

% Computing ATP utilization rate

nH = 0.1;

ki = (GA-5)^nH/(8-5)^nH;

J\_ATPase = Pa\*GA\*ATPase\*ki;

% Computing proton leak flux

J\_HL = pleak\*exp(Gh);

% Computing superoxide leak flux

J\_SO = Pa\*eleak\*Gr;

Af = param(8); %area of the fat fuel tank in m^2

Ar = param(9); %area of the redox tank in m^2

Ap = param(10); %area of Dp tank in m^2

Aa = param(11); %area of ATP tank in m^2

Apcr = param(12); %area of PCr tank in m^2

% Computing Time Derivatives of Global Variables

f(iFAT) = -J\_BOX/Af;

f(iGr) = (J\_BOX + J\_pdh - J\_ETC - J\_SO)/Ar; %(J\_DH - J\_ETC - J\_SO)/param(11);

f(iGh) = (J\_ETC - J\_VA - J\_HL)/Ap; %(J\_ETC - J\_VA - J\_HL)/param(12);

f(iGA) = (J\_VA + J\_CK - J\_ATPase)/Aa;

f(iPCR) = (-J\_CK)/Apcr;

f = f';

**Fluxes**

function [J] = Fluxes(x,param)

% This function outputs the discrete time flux values for the hydraulic model.

% Defining Global Variables

global iFAT iCHO iGr iGh iGA iPCR

% Defining local variables

FAT = x(iFAT);

CHO = x(iCHO);

Gr = x(iGr);

Gh = x(iGh);

GA = x(iGA);

PCR = x(iPCR);

%Defining conductances or Vmax (Vmax for PDH only, see below)

BOX = param(1); %Beta oxidation pathway conductance from FAT to matrix redox

Vpdh = param(13); %PDH Vmax expressed in m^3 H2O per sec

ETC = param(2);

VA = param(3);

ATPase = param(4);

pleak = param(5);

eleak = param(6);

CK = param(7);

Pa = 9806; %converts tank height to Pa of driving pressure

% Computing flux into fuel source

J\_fuel = 0;

% Computing beta oxidation flux of lipid into matrix redox potential

J\_BOX = Pa\*(FAT-Gr)\*BOX;

% Computing PDH flux of CHO into matrix redox potential

Cpdh = param(13); %Conductance of glycolytic carbon entry into matrix redox

CHOSat = param(14); %Gives status of CHO availability (0 - 1.0)

Act = (8-GA)^3.5/(8-5)^3.5; %Delta Gatp related activation term of glycolysis

Inhib = (8-Gr)^0.25/(8-5)^0.25; %Delta G redox related feedback inhibition of PDH

J\_pdh = Pa\*(CHO-Gr)\* Cpdh \* Act \* CHOSat \* Inhib;

% Computing movement of electrons into proton motive force

J\_ETC = Pa\*(Gr-Gh)\*ETC;

% Computing flux of protons into ATP Synthesis & Export

KmADP = 0.5;

nH = 2;

VAkin = (8-GA)^nH/((8-GA)^nH+KmADP^nH);

J\_VA = Pa\*(Gh-GA)\*VA\*VAkin;

% Computing flux of PCr into ATP

J\_CK = Pa\*(PCR-GA)\*CK;

% Computing ATP utilization rate

nH = 0.1;

ki = (GA-5)^nH/(8-5)^nH;

J\_ATPase = Pa\*GA\*ATPase\*ki;

% Computing proton leak flux

J\_HL = pleak\*exp(Gh);

% Computing superoxide leak flux

J\_SO = Pa\*eleak\*Gr;

% Outputting Fluxes

J(1) = J\_fuel\*1000000; %\*1000000 converts m^3 H2O to mL H2O

J(2) = J\_BOX\*1000000;

J(3) = J\_pdh\*1000000;

J(4) = J\_ETC\*1000000;

J(5) = J\_VA\*1000000;

J(6) = J\_HL\*1000000;

J(7) = J\_SO\*1000000;

J(8) = J\_ATPase\*1000000;

J(9) = J\_CK\*1000000;

**Indices**

function Indices()

% This function creates an index for the state variables of the hydraulic

% model.

global iFAT iCHO iGr iGh iGA iPCR

iFAT = 1;

iCHO = 2;

iGr = 3;

iGh = 4;

iGA = 5;

iPCR = 6;

Model Run Script Example (was run 06-28-2022)

%This simple hydraulic model simulator includes a PCr tank that sets the

%outflow backpressure (hydraulic model of the CK clamp)

clear; close all;

global iFAT iCHO iGr iGh iGA iPCR

Indices();

% Load Initial Conditions

pathname = 'C:\Users\Ben\Desktop\ANT Model Files\Summer 2017 ANT Model Files\Hydraulic Model\';

filename = 'HydraulicLoading.xlsx';

[d] = xlsread([filename],'C3:C8');

xo = d;

% Load Parameters

[p] = xlsread([filename],'H3:H16');

param = p;

indexC = [1 3:6];

options = odeset('MaxStep',5e-2);

% Below are the intial water levels

xo(iFAT) = 8;

xo(iCHO) = 11.5;

xo(iGr) = 8;

xo(iGh) = 8;

xo(iGA) = 8;

xo(iPCR) = 8;

CHOSat = param(14); %CHO status (values 0 - 1.0) gives glucose & glycogen availability

param(14) = 1\*CHOSat;

Vpdh = param(13); % this is the pdh Vmax in m^3/sec

param(13) = 1\*Vpdh;

% Below are the conductances of each pipe

Cbox = param(1); % this is the conductance of the beta oxidation pathway

param(1) = 1\*Cbox;

Cetc = param(2); % this is the conductance of ETC

param(2) = 1\*Cetc;

Cva = param(3); % this is the conductance of Complex V + ANT

param(3) = 1\*Cva;

CCK = param(7); % this is the conductance of CK equilibration of ATP with PCr

param(7) = 1\*CCK;

CHL = param(5); % this is conductance factor of proton leak

param(5) = 1\*CHL;

CSO = param(6); % this is conductance factor of superoxide electron leak

param(6) = 1\*CSO;

XATCR = param(12); %cross-sectional area of PCr tank. Each x-fold adjusts [TCr] by that factor

param(12) = .5\*XATCR;

vatp = param(4); %Vmax = 1 this is the ATPase outflow rate

param(4) = 0.00045\*vatp;

tic

t\_ib1 = 300; % incubation period of 100 seconds

[t,x] = ode15s(@Metabolites,[0 t\_ib1],xo,options,param);

toc

J1 = Fluxes(x(end,:),param);

xo = x(end,:);

x1 = x;

t1 = t;

param(4) = 0.05\*vatp; % onset of first level of ATP breakdown

% xo(iPCR) = 8;

% param(1) = 4.64\*Cdh;

% param(2) = 1\*Cetc;

% param(3) = 1.8\*Cva;

tic

t\_ib2 = 300; % exercise time

[t,x] = ode15s(@Metabolites,[0 t\_ib2],xo,options,param);

toc

J2 = Fluxes(x(end,:),param);

xo = x(end,:);

x2 = x;

t2 = t;

param(4) = .1\*vatp; % onset of 2nd level of ATP breakdown

% xo(iPCR) = 8;

% param(1) = 4.17\*Cdh;

% param(2) = 1\*Cetc;

% param(3) = 2.01\*Cva;

tic

t\_ib3 = 300; % exercise time

[t,x] = ode15s(@Metabolites,[0 t\_ib3],xo,options,param);

toc

J3 = Fluxes(x(end,:),param);

xo = x(end,:);

x3 = x;

t3 = t;

% param(1) = 4.64\*Cdh;

% param(2) = .9\*Cetc;

% param(3) = 2.01\*Cva;

param(4) = 0.2\*vatp; % onset of third level of ATP breakdown

% xo(iPCR) = 8;

tic

t\_ib4 = 300; % exercise time

[t,x] = ode15s(@Metabolites,[0 t\_ib3],xo,options,param);

toc

J4 = Fluxes(x(end,:),param);

xo = x(end,:);

x4 = x;

t4 = t;

x = [x1' x2' x3' x4']';

t = [t1' (t2+t\_ib1)' (t3+t\_ib2+t\_ib1)' (t4+t\_ib3+t\_ib2+t\_ib1)'];

%Po = zeros(length(x));

%P%a = 9806;

%f%or i = 1:length(x)-1

% Po(i) = Pa\*param(4)\*x(iGA,i+1)-Pa\*param(4)\*x(iGA,i);

%end

%Defining array for download into Excel

tt1=t';

y=tt1(1:30:end,1);

xlswrite('kindataNEW.xlsx',y,1,'A3:A811');

variables=x(1:30:end,4); %variables=x(1:200:end,[25 35 36 62 92]); %this is the full list of variables if wanted and expand Column letters accordingly e.g., 'B3:C410' for one additional variable

xlswrite('kindataNEW.xlsx',variables,1,'B3:B1600');

%Plotting

figure %Water levels in each tank

subplot(1,5,1),plot(t,x(:,1));

ylabel('Fat')

set(gca,'xlim',[0 (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],'ylim',[5 8]);

subplot(1,5,2),plot(t,x(:,3));

ylabel('Redox')

set(gca,'xlim',[0 (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],'ylim',[5 8]);

subplot(1,5,3),plot(t,x(:,4));

ylabel('Delta p')

set(gca,'xlim',[0 (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],'ylim',[5 8]);

subplot(1,5,4),plot(t,x(:,5));

ylabel('ATP')

set(gca,'xlim',[0 (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],'ylim',[5 8]);

subplot(1,5,5),plot(t,x(:,6));

ylabel('PCr')

set(gca,'xlim',[0 (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],'ylim',[5 8]);

figure %Fluxes

subplot(2,4,1),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(2) J2(2) J3(2) J4(2)],'ko')

ylabel('BetaOx (mL H2O/sec)'),xlabel('Time (Seconds)')

subplot(2,4,2),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(3) J2(3) J3(3) J4(3)],'ko')

ylabel('PDH (mL H2O/sec)'),xlabel('Time (Seconds)')

subplot(2,4,3),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(4) J2(4) J3(4) J4(4)],'ko')

ylabel('ETC (mL H2O/sec)'),xlabel('Time (Seconds)')

subplot(2,4,4),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(5) J2(5) J3(5) J4(5)],'ko')

ylabel('Comp V + ANT (mL H2O/sec)'),xlabel('Time (Seconds)')

subplot(2,4,5),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(9) J2(9) J3(9) J4(9)],'ko')

ylabel('CKase(mL H2O/sec)'),xlabel('Time (Seconds)')

subplot(2,4,6),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(6) J2(6) J3(6) J4(6)],'ko')

ylabel('H Leak (mL H2O/sec)'),xlabel('Time (Seconds)')

subplot(2,4,7),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(7) J2(7) J3(7) J4(7)],'ko')

ylabel('Superoxide (mL H2O/sec)'),xlabel('Time (Seconds)')

subplot(2,4,8),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(8) J2(8) J3(8) J4(8)],'ko')

ylabel('ATP(mL H2O/sec)'),xlabel('Time (Seconds)');

% subplot(2,4,8),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(4)/1000\*x(t\_ib1, iGh-iGA)\*9806 J2(4)/1000\*x((t\_ib2+t\_ib1), iGh-iGA)\*9806 J3(4)/1000\*x((t\_ib3 + t\_ib2+t\_ib1), iGh-iGA)\*9806 J4(4)/1000\*x((t\_ib4 + t\_ib3 + t\_ib2+t\_ib1), iGh-iGA)\*9806],'ko')

% ylabel('Po (watts)'),xlabel('Time (Seconds)');

% subplot(2,4,8),plot([t\_ib1 (t\_ib2+t\_ib1) (t\_ib3+t\_ib2+t\_ib1) (t\_ib4+t\_ib3+t\_ib2+t\_ib1)],[J1(Po) J2(Po) J3(Po) J4(Po)],'ko')

% ylabel('Po (watts)'),xlabel('Time (Seconds)');

Loading used 06-28-2022

