Aug 01, 2025 Version 2

Simulation and Flux Analysis of β-Oxidation Using MATLAB: A Stepwise ODE-Based Protocol V.2

DOI
https://dx.doi.org/10.17504/protocols.io.x54v95yepl3e/v2

Ethan Shaw[1], Henry Chuo[2]

[1]University of Houston, Baylor College of Medicine; [2]University of Houston

Ethan Shaw
Baylor College of Medicine

DOI: https://dx.doi.org/10.17504/protocols.io.x54v95yepl3e/v2

Protocol Citation: Ethan Shaw, Henry Chuo 2025. Simulation and Flux Analysis of β-Oxidation Using MATLAB: A Stepwise ODE-Based Protocol. **protocols.io** https://dx.doi.org/10.17504/protocols.io.x54v95yepl3e/v2Version created by **Ethan Shaw**

Protocol status: Working
We use this protocol and it's working

Created: July 31, 2025

Last Modified: August 01, 2025

Protocol Integer ID: 223844

Keywords: oxidation, using ode45, using matlab, reaction kinetic, matlab, defining reaction kinetic

Abstract

This protocol outlines the complete process of modeling mitochondrial β-oxidation as a system of first-order differential equations, beginning with stoichiometric derivation and matrix reduction, and culminating in a MATLAB-based simulation. The user is guided through each step of linearizing the pathway, defining reaction kinetics, and numerically solving the system using ode45.

Attachments

M

beta_oxidation_model...

22KB

Materials

MATLAB (R2020b or later) — Required for numerical integration of ODE systems, matrix computation, and time-series visualization. The protocol uses built-in functions such as ode45, linspace, and plot, and is compatible with both MATLAB desktop and MATLAB Online.

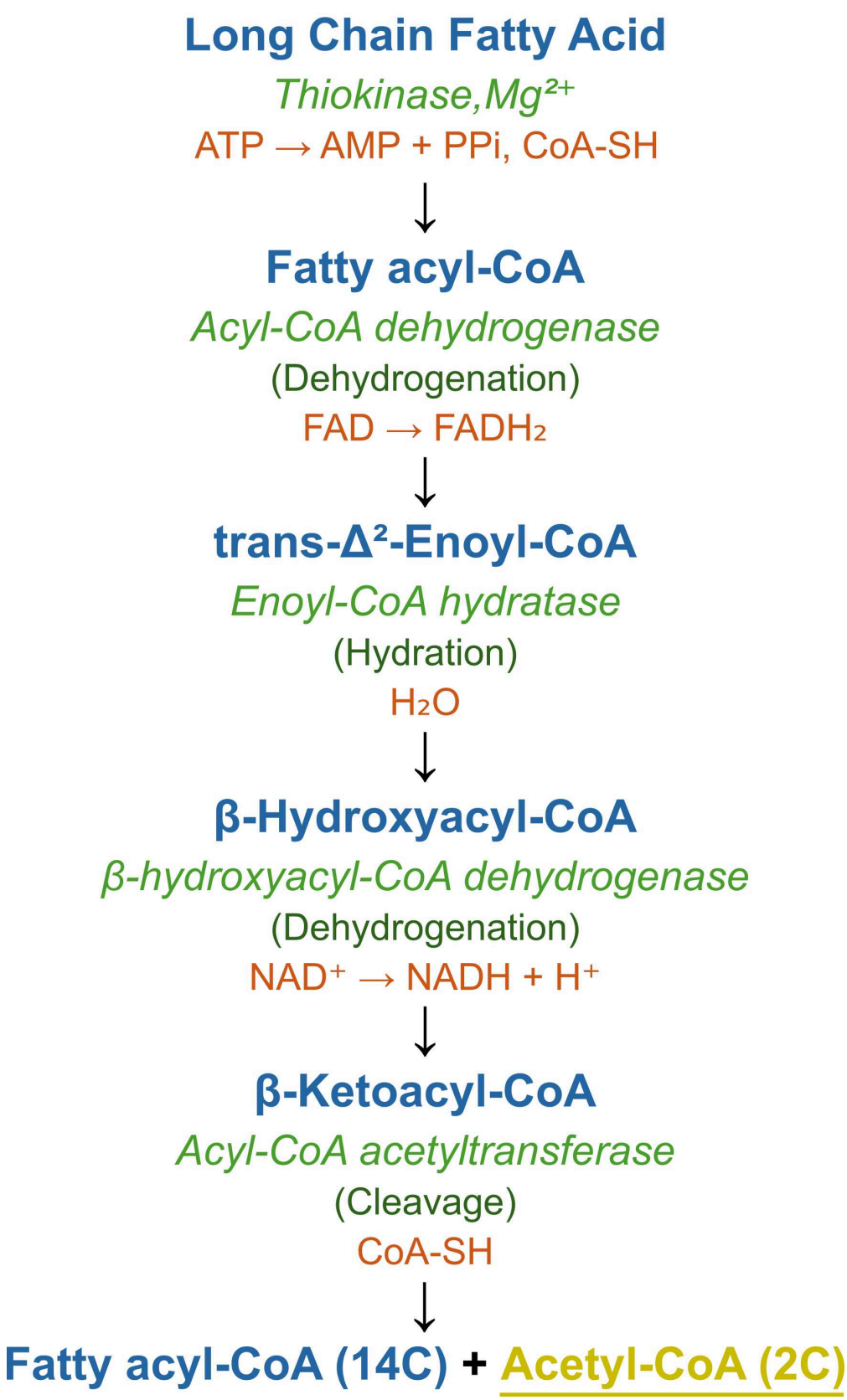
Pathway Linearization and Stoichiometric Modeling

1 Construct a linearized kinetic representation of your metabolic pathway by referencing an established biochemical model. Then, decompose the pathway into a discrete, stepwise system of reactions in which each metabolite is uniquely identified, each enzymatic transformation is assigned a distinct reaction velocity (V_n), and each associated parameter (e.g., cofactors, substrates, rate constants) is explicitly mapped.

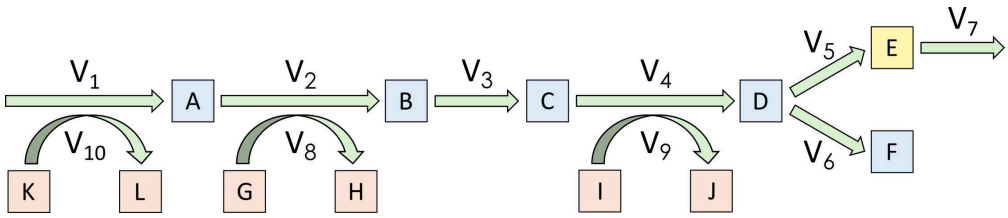
1.1 In Step 2, the biochemical β-oxidation pathway is outlined using conventional nomenclature, showing the conversion of long-chain fatty acids into acetyl-CoA through sequential enzymatic reactions, along with the involvement of key cofactors (ATP, FAD, NAD^+, CoA-SH). Each intermediate and catalytic step is named and associated with its corresponding transformation.

1.2 In Step 3, the pathway is converted into a modular kinetic model, where each metabolite is denoted by a letter (A through F), and each enzymatic reaction is represented by a velocity term (*V1 through V*10). Cofactors (K through L) are similarly identified. This form isolates the stoichiometric and kinetic relationships needed for constructing ordinary differential equations (ODEs) for dynamic simulation and parameter analysis.

2


Fatty Acid β-Oxidation Pathway
Long Chain Fatty Acid
Thiokinase,Mg²⁺
ATP → AMP + PPi, CoA-SH
Fatty acyl-CoA
Acyl-CoA dehydrogenase
(Dehydrogenation)
FAD → FADH₂
trans-Δ²-Enoyl-CoA
Enoyl-CoA hydratase
(Hydration)
H₂O
β-Hydroxyacyl-CoA
β-hydroxyacyl-CoA dehydrogenase
(Dehydrogenation)
NAD⁺ → NADH + H⁺
β-Ketoacyl-CoA
Acyl-CoA acetyltransferase
(Cleavage)
CoA-SH
Fatty acyl-CoA (14C) + Acetyl-CoA (2C)

3



4 Derive a set of first-order differential equations from the multi-step biochemical pathway in Step 3. Write out the reaction in the pathway, specifying reactants and products along with their stoichiometric coefficient. A negative coefficient indicates the substrate is a reactant, and a positive coefficient indicates the substrate is a product of the reaction.

5

$$\frac{dA}{dt} = V_1 - V_2 \qquad \frac{dB}{dt} = V_2 - V_3$$

$$\frac{dC}{dt} = V_3 - V_4 \qquad \frac{dD}{dt} = V_4 - V_6 - V_5$$

$$\frac{dE}{dt} = V_5 - V_7 \qquad \frac{dF}{dt} = V_6 - V_{10}$$

$$\frac{dG}{dt} = -V_8 \qquad \frac{dH}{dt} = V_8$$

$$\frac{dI}{dt} = V_9 \qquad \frac{dJ}{dt} = -V_9$$

$$\frac{dK}{dt} = -V_{10} \qquad \frac{dL}{dt} = V_{10}$$

6 Create a matrix corresponding to the first order differential equations, where each row represents a substrate, and each column represents a reaction in the pathway. Fill in entries of matrix with the corresponding stoichiometric coefficients determined in the previous step.

7

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}
A	1	−1	0	0	0	0	0	0	0	0
B	0	1	−1	0	0	0	0	0	0	0
C	0	0	1	−1	0	0	0	0	0	0
D	0	0	0	1	−1	−1	0	0	0	0
E	0	0	0	0	1	0	−1	0	0	0
F	0	0	0	0	0	1	0	0	0	0
G	0	0	0	0	0	0	0	−1	0	0
H	0	0	0	0	0	0	0	1	0	0
I	0	0	0	0	0	0	0	0	−1	0
J	0	0	0	0	0	0	0	0	1	0
K	0	0	0	0	0	0	0	0	0	−1
L	0	0	0	0	0	0	0	0	0	1

8 Transform the stoichiometric matrix to its row-reduced echelon form by performing Gaussian elimination with back substitution. Using this, we can define dimensions, rank, and nullity of the matrix. We can also define the number of free variables in the matrix.

9

$$\begin{pmatrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 & V_8 & V_9 & V_{10} \\ A & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ B & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ C & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ H & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ J & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ K & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \\ V_{10} \end{pmatrix} = 0$$

Row Reduced Echelon Form of the stoichiometric matrix for the β-oxidation pathway.

10

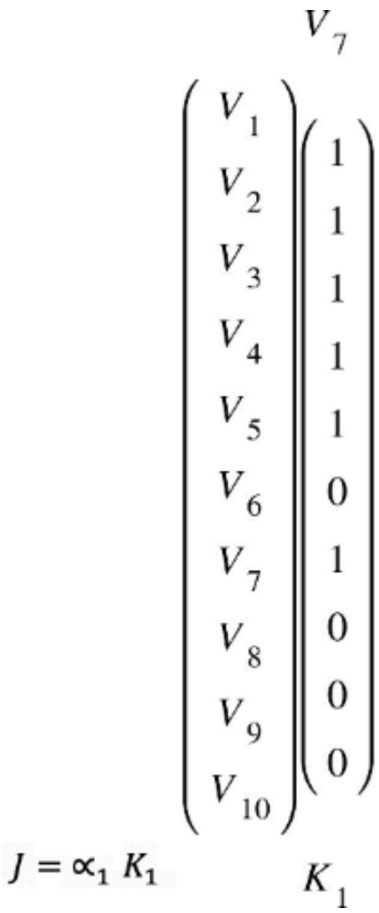
Dimensions = 10, Rank = 9, Nullity = 1, 1 Free Variable at V_7

11 Convert the Row Reduced Echelon Form into a system of equations with each equation corresponding to a substrate row. Make each of the equations equal to 0, then use algebraic manipulation to make an equation for each reaction (V_1, V_2, V_3, etc.).

12

$$
\begin{aligned}
V_1 - V_7 &= 0 \\
V_2 - V_7 &= 0 \\
V_3 - V_7 &= 0 \\
V_4 - V_7 &= 0 \\
V_5 - V_7 &= 0 \\
V_6 &= 0 \\
V_7 & \\
V_8 &= 0 \\
V_9 &= 0 \\
V_{10} &= 0
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
V_1 &= V_7 \\
V_2 &= V_7 \\
V_3 &= V_7 \\
V_4 &= V_7 \\
V_5 &= V_7 \\
V_6 & \\
V_7 &= V_7 \\
V_8 & \\
V_9 & \\
V_{10} &
\end{aligned}
$$

13



MATLAB Implementation and Dynamic Simulation

14 Write a new MATLAB function titled ode_system.m to define the ODE system governing β-oxidation dynamics. See Step 15 for confirmation.

14.1 At the top of the file, unpack the state vector y into individual metabolite variables (a through l) representing concentrations of each species.

14.2 Below, define all kinetic rate constants (k1 through k5) with realistic values derived from literature or database sources, keeping units in 1/s or 1/mM·s.

14.3 Define the full set of reaction velocities (V1–V10), ensuring correct substrate and cofactor dependencies for each enzymatic step. Note that V1 and V10 are equivalent.

14.4 Express each ODE as a difference in velocities, such that the change in each metabolite is written in the form dx_dt = Vi - Vj, avoiding invalid terms such as +V10 or +V7 in the final equations.

15

```
%% Establishing K-Vals, Reaction-V., and ODEs
function dydt = ode_system(~, y)
    % Unpack variables
    A = y(1); B = y(2); C = y(3); D = y(4); E = y(5);
    F = y(6); G = y(7); H = y(8); I = y(9); J = y(10);
    K = y(11); L = y(12);

    %% Define reaction rate constants (realistic estimates in 1/s or 1/mM·s)
    % [REPLACE IF/WHEN POSSIBLE]
    k1 = 0.05;  %Thiokinase [CORRECT]
    k2 = 0.9;   % Acyl-CoA dehydrogenase reaction rate [CORRECT]
    %k2 = 0.01;  % Modified k2 value to simulate MCADD -- ONLY UNCOMMENT IF SIMULATING MCADD
    k3 = 1.2;   % Enoyl-CoA hydratase reaction rate [CORRECT]
    k4 = 1.0;   % β-Hydroxy acyl-CoA dehydrogenase reaction rate [CORRECT]
    k5 = 1.1;   % Thiolase reaction rate [CORRECT] -- Thiolase is the shorthand for the Acyl-CoA Acetyltransferase Cleavage

    %% Define reaction velocities [CORRECT]
    %V1 and V10 are actually the exact same
    V1 = k1 * K;
    V2 = k2 * A * G;
    V3 = k3 * B;
    V4 = k4 * C * I;
    V5 = k5 * D;
    V6 = k5 * D;
    V7 = k2 * E;
    V8 = k2 * A * G;
    V9 = k4 * C * I;
    V10 = k1 * K;
    %% Define ODEs [CORRECT]
    % NO +V10 (dF_dt), or +V7 (dE_dt)
    dA_dt = V1 - V2;
    dB_dt = V2 - V3;
    dC_dt = V3 - V4;
    dD_dt = V4 - V6 - V5;
    dE_dt = V5 - V7;
    dF_dt = V6;
    dG_dt = -V8;
    dH_dt = V8;
    dI_dt = V9;
    dJ_dt = -V9;
    dK_dt = -V10;
    dL_dt = V10;
    %Sample Corrective ODE
    %dZ_dt = V10 - V1 + V7;

    %% Return derivative values
    dydt = [dA_dt; dB_dt; dC_dt; dD_dt; dE_dt; dF_dt;
            dG_dt; dH_dt; dI_dt; dJ_dt; dK_dt; dL_dt];

end
```

16 In your main MATLAB script, define initial conditions for all 12 metabolites as scalar values in millimolar (mM), referencing physiologically relevant ranges for fatty acid oxidation. See Step 17 for confirmation.

16.1 Specify a time vector using linspace(0,10,100) to define a uniform time domain from 0 to 10 seconds, with 100 stored time points.

16.2 Call the MATLAB solver using [t, y] = ode45(@ode_system, tspan, [A0 B0 C0 D0 E0 F0 G0 H0 I0 J0 K0 L0]); to simulate the ODE system and record the time-course evolution of each metabolite.

17

```
%% Define realistic initial conditions (in mM)
A0 = 0.5;   % Fatty Acyl-CoA ~ 0.5 mM (depends on lipid metabolism rate)
B0 = 0.05;  % Trans Δ2 Enoyl-CoA ~ 0.05 mM (intermediate product)
C0 = 0.03;  % β-Hydroxy acyl-CoA ~ 0.03 mM (intermediate product)
D0 = 0.02;  % β-Keto acyl-CoA ~ 0.02 mM (intermediate product)
E0 = 1.5;   % Acetyl-CoA ~ 1.5 mM (enters citric acid cycle)
F0 = 0.2;   % Shortened Acyl-CoA ~ 0.2 mM (for next cycle)
G0 = 0.1;   % FAD ~ 0.1 mM (electron carrier)
H0 = 0.05;  % FADH2 ~ 0.05 mM (reduced electron carrier)
I0 = 1.0;   % NAD+ ~ 1.0 mM (electron acceptor for oxidation)
J0 = 0.5;   % NADH ~ 0.5 mM (reduced electron carrier)
K0 = 2.0;   % ATP ~ 2.0 mM (energy donor for fatty acid activation)
L0 = 0.1;   % AMP ~ 0.1 mM (byproduct of ATP hydrolysis in activation step)
```

```
%% Figure Specification
% % Define time span with stored time values
tspan = linspace(0, 10, 100); % Store 100 evenly spaced time points from 0 to 10

% % Solve ODEs using ode45
[t, y] = ode45(@ode_system, tspan, [A0 B0 C0 D0 E0 F0 G0 H0 I0 J0 K0 L0]);
```

18

beta_oxidation_model.m 22.7KB

For further features such as custom figure generation, reaction logging, or MCADD simulation toggles, download the complete model file: beta*oxidation*model.m – which contains embedded comments and configurable options for simulation refinement.