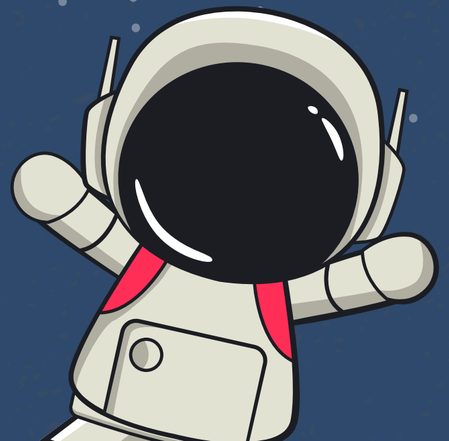
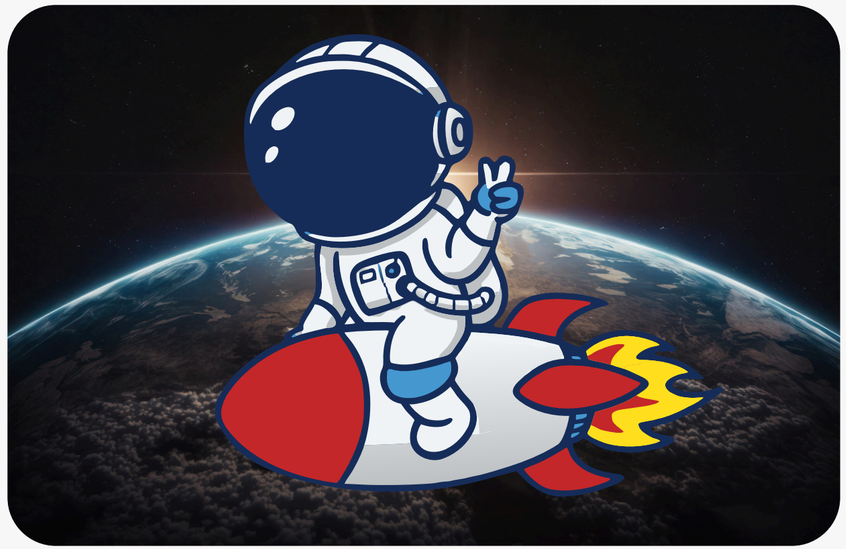
FLUID HONOR OPTION

Sami & Kaylie



BACKGROUND

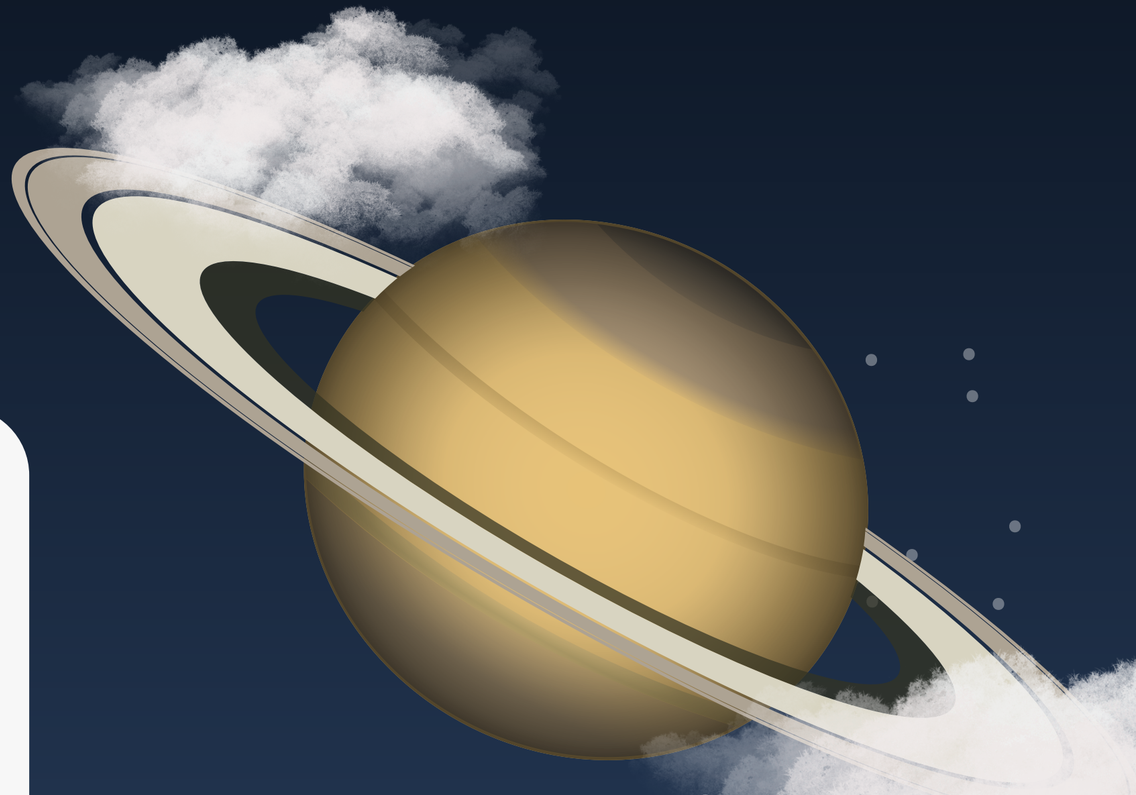


Bob is an astronaut coming back from his latest space mission. Due to his 6 months in the solar system, Bob is now facing muscle atrophy, affecting his overall mobility and way of life. Thanks to the innovative research done at CSU, there exists a pharmaceutical drug to reverse these effects.

OUR MISSION

We will use MatLab to develop a pharmokinetics and finite volume model to simulate the effect of the drug entering Bob's blood stream. With this model, we will be able to study the rate of capillary action of his cells and give an accurate estimate of how his body may react quantitatively after exposure.

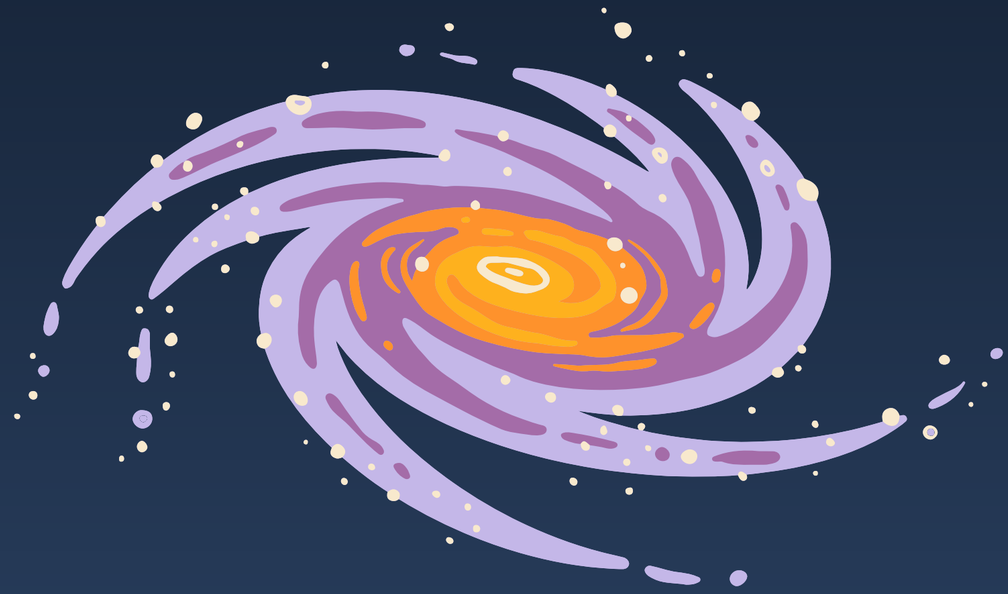
THE SCIENCE



Muscle Atrophy impacts people in all different situations. One that is less common is people who come back from space missions. People who travel in space can experience muscle atrophy similar to that of wheelchair patients. In both cases patients can experience greater than a 50% decrease in muscle protein synthesis. Astronauts who come back from short missions can see a 10 to 20% decrease in total muscle loss whereas longer missions can cause up to a 50% decrease in total muscle mass loss.

PROBLEM STATEMENT

We need to use a pharmokinetics model to show the effect of the drug on Bob’s cells. We also need to make a finite volume model to incorporate class concepts into our project.



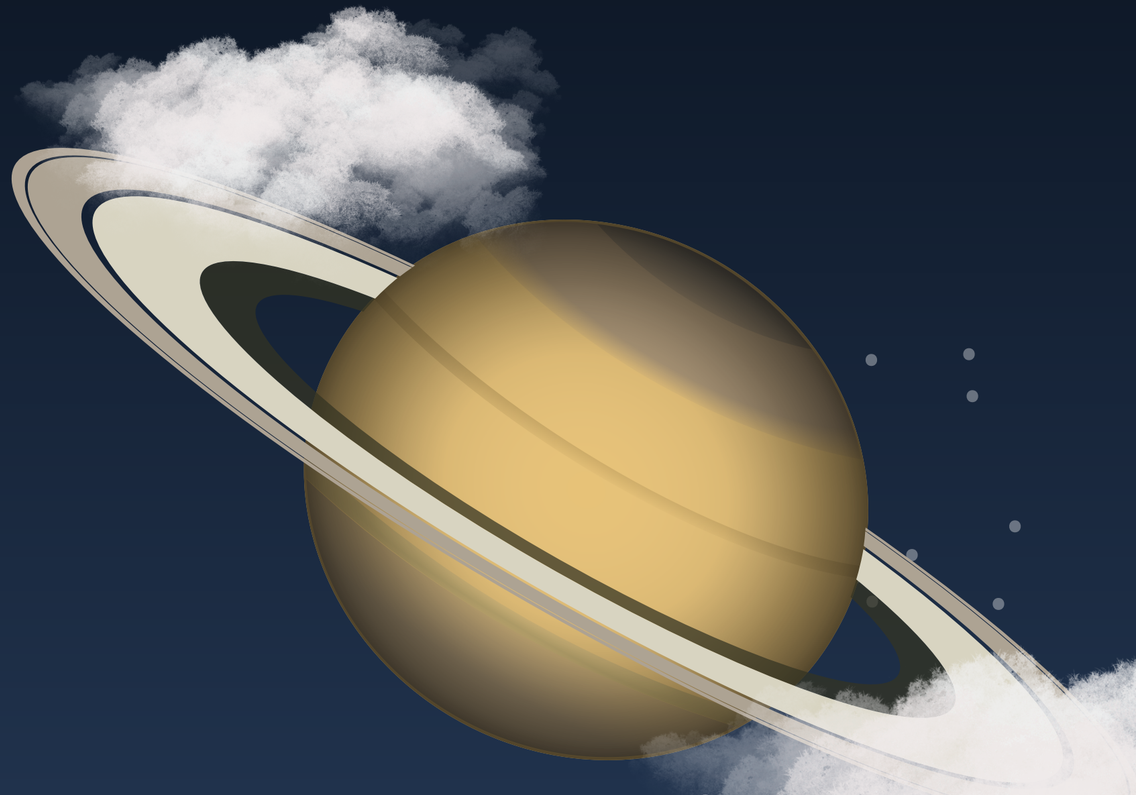
ASSUMPTIONS

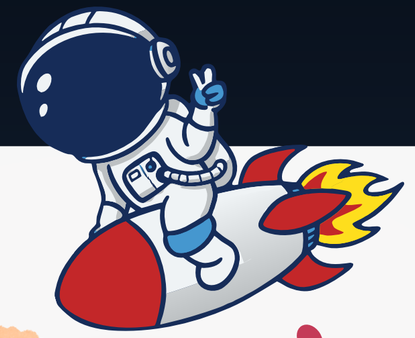
- ~1% Bone Density Loss per Month
- 10–20% loss on short missions, up to 50% on long missions

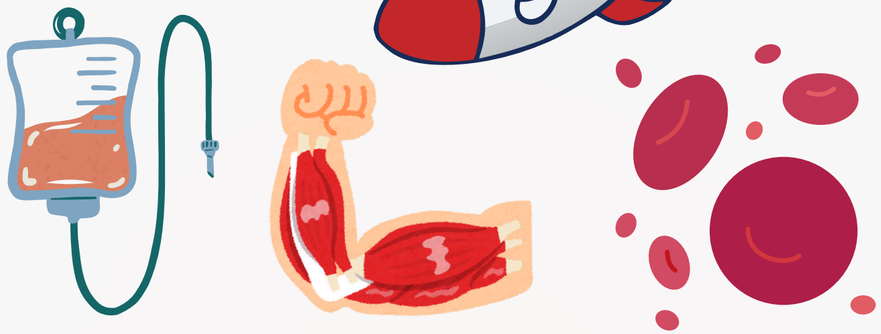
- **30% Muscle Atrophy Overall**

GOALS

- Use a pharmokinetics model to show how muscle atrophy and decreased protein synthesis impacts drug distribution
- Solve a Finite Volume problem to incorporate fluids principles into our project statement



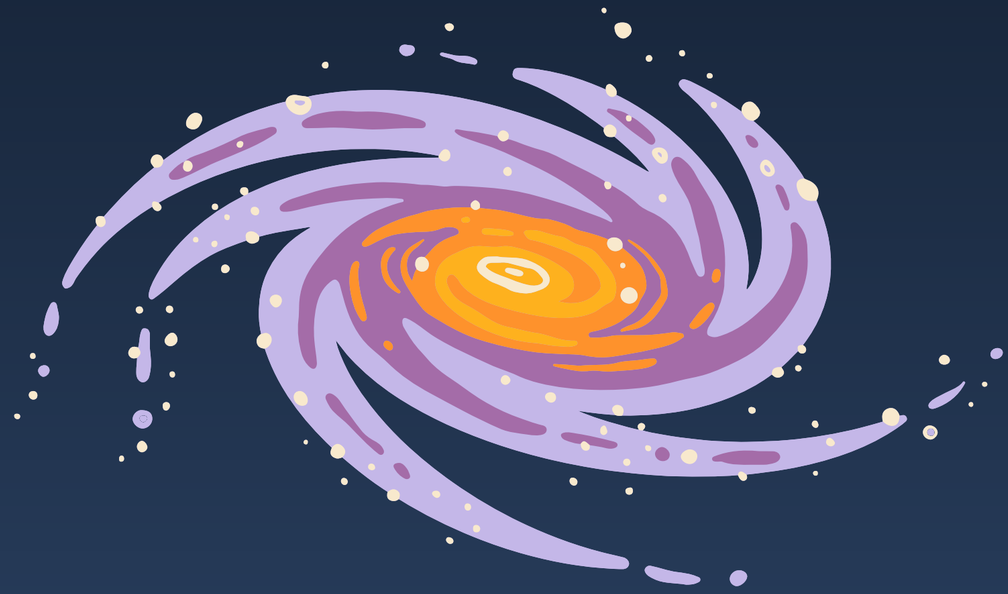




METHODS

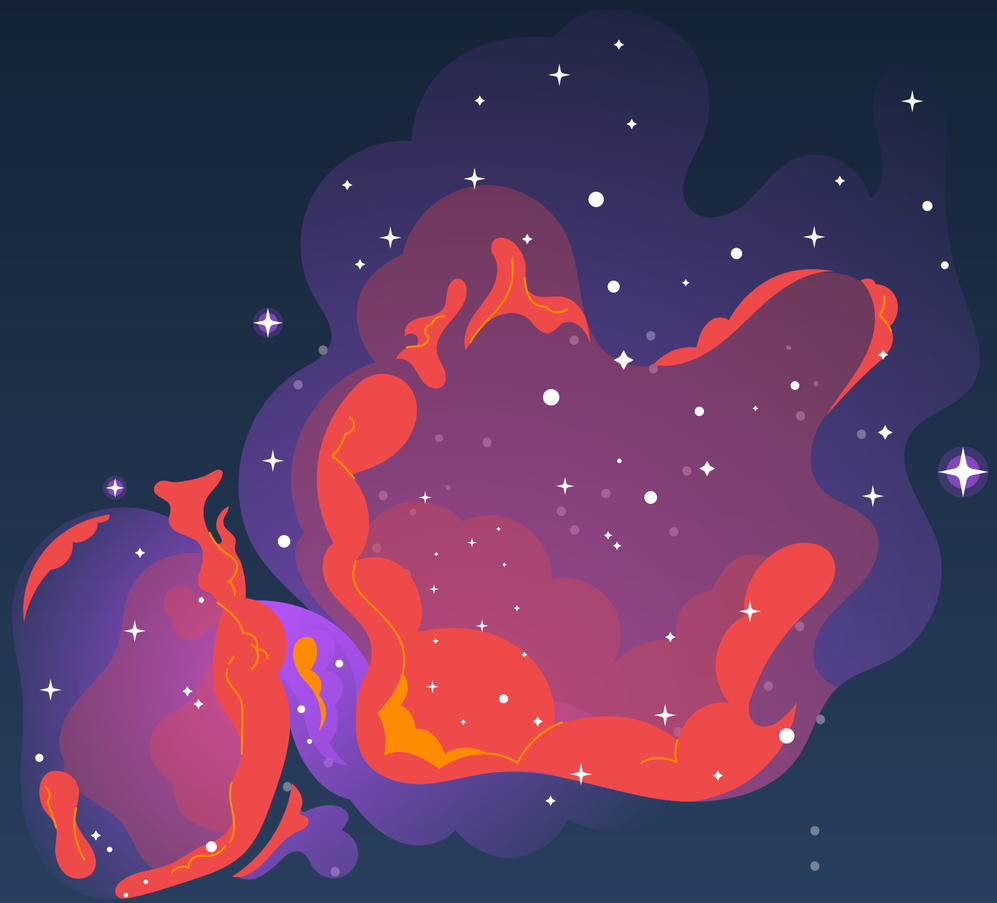
3 Compartment PK - Bob gets 1 IV Bolus shot directly into an extremity. We modeled how a decrease in plasma and muscle volume impacted transport and total distribution.

Finite Volume - After Bob is initially injected, looking at the concentration in the plasma, muscle, and rest of cells after hour increments,



```
properties
    % Volumes L
    Vp  = 3.0 * (1 - 0.12);   % Plasma vol ~12% lower
    Vm  = 25.0 * (1 - 0.30);  % Muscle vol ~30% lower (atrophy)
    Vr  = 30.0;               % Rest-of-body vol

    % Intercompartmental clearances L/hr
    Qpm = 20 * 0.85;          % Plasma<->Muscle exchange (~15% reduction)
    Qpr = 15;                 % Plasma<->Rest exchange

    % Systemic clearance L/h (acts on free Cp)
    CL  = 5;

    % Partition coefficients (unitless)
    Kpm = 2.0;                % Muscle:plasma
    Kpr = 1.2;                % Rest:plasma

    % Binding (fractions unbound)
    fu_p = 0.4;               % plasma fraction unbound (edit per drug/physiology)
    fu_m = 1.0;               % muscle free fraction (often ~1 in interstitium)
    fu_r = 1.0;               % rest free fraction

    % Single bolus dose & simulation grid
    Dose    = 100;                      % mg (entire dose at t=0 into plasma)
    t_array = linspace(0,12,1200);      % hours
    fignum  = 10;                       % figure number
end
```

THE CODE (PHARMOKINETICS)



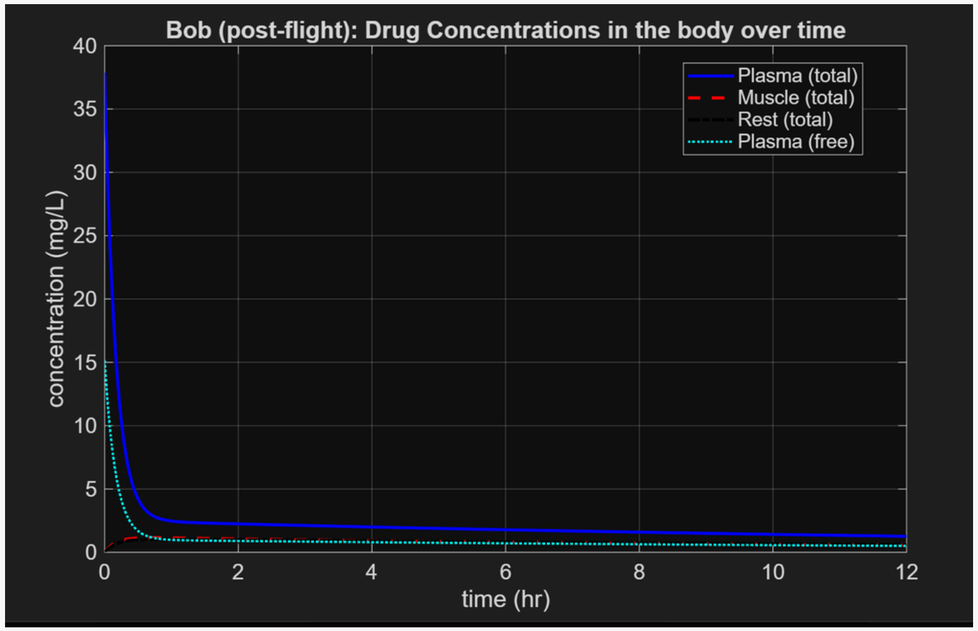
MORE CODE

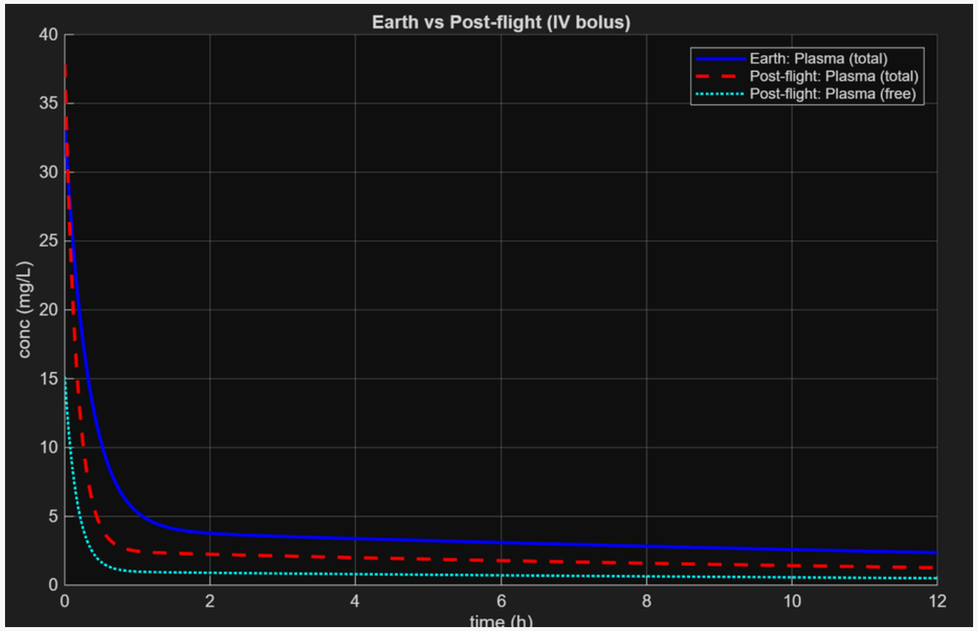
$$\frac{dA_p}{dt} = -CL \cdot C_{p,\text{free}} - J_{pm} - J_{pr}$$
$$\frac{dA_m}{dt} = J_{pm}$$
$$\frac{dA_r}{dt} = J_{pr}$$

```
methods
    function Traj = get.Numerical(obj)
        % Amount-based ODE RHS
        fdot = @(t, A) obj.rhsAmounts(t, A);

        % Initial amounts: IV bolus in plasma at t=0
        A0 = [obj.Dose; 0; 0];   % [Ap(0), Am(0), Ar(0)] in mg

        % Let MATLAB estimate derivatives internally (no Jacobian provided)
        opts = odeset('RelTol', 1e-8, 'AbsTol', 1e-9);
        [~, A] = ode45(fdot, obj.t_array, A0, opts);

        % Convert amounts to TOTAL concentrations [mg/L]
        Cp = A(:,1) / obj.Vp;
        Cm = A(:,2) / (obj.Vm * obj.Kpm);
        Cr = A(:,3) / (obj.Vr * obj.Kpr);

        Traj = [Cp, Cm, Cr];
    end

    function val = get.AUC(obj)
        C  = obj.Numerical(:,1);              % plasma total
        dt = obj.t_array(2) - obj.t_array(1);
        val = sum(C(1:end-1) + C(2:end)) * 0.5 * dt;
    end
```

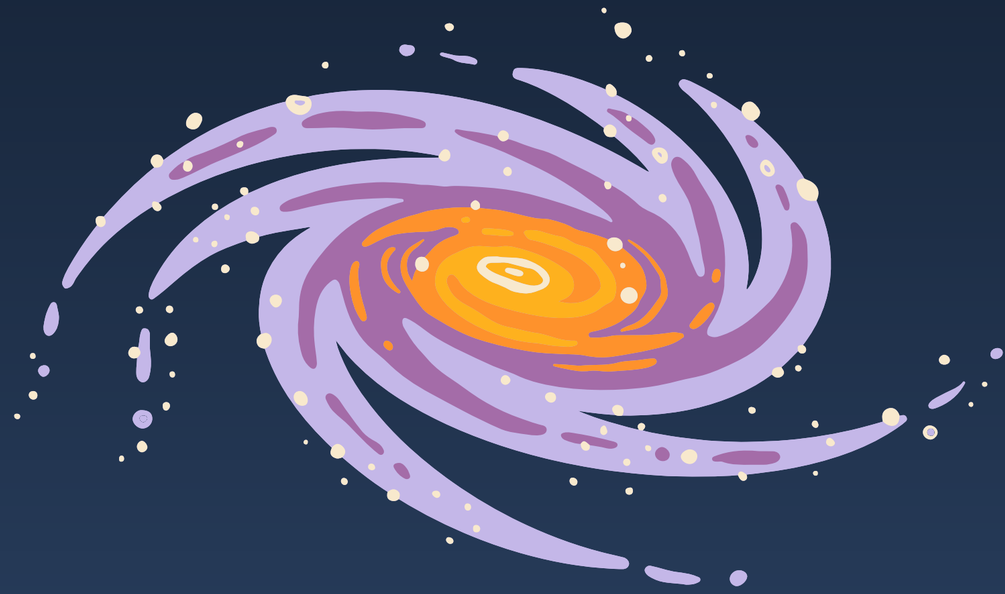
$$J_{pm} = Q_{pm}\left(C_{p,\text{free}} - C_{m,\text{free}}\right), \quad J_{pr} = Q_{pr}\left(C_{p,\text{free}} - C_{r,\text{free}}\right)$$

$$C_p = \frac{A_p}{V_p}, \quad C_m = \frac{A_m}{V_m \cdot K_{pm}}, \quad C_r = \frac{A_r}{V_r \cdot K_{pr}}$$





RESULTS (PHARMOKINETICS)



```
%% MuscleFVM_Bob
%
% Finite-Volume (1D) diffusion + uptake in muscle interstitium,
% driven by plasma concentration Cp(t) at x=0 via a Robin boundary.

% Purpose: Simulate 1D drug/solute transport in muscle interstitim using
% diffusion and firts-order uptake, driven by plasma concentration Cp(t) at
% x=0 through a Robin boundary condition (mass-transfer/exchange)
%
% PDE being solved: dC/dt = D d2C/dx2 - k_uptake * C
% BCs:
% Left: (x=0): -D dC/dx = k_exchange (Cp(t) - C(0,t)) % Robin BC
% Right (x=L): dC/dx = 0 % zero-flux (Neumann)

% Usage (coupled with ThreeCompPKModel):
%   PK = ThreeCompPKModel;           % Bob post-flight PK
%   Cpl = PK.Numerical(:,1);         % plasma concentration vs time [mg/L]
%   FVM = MuscleFVM_Bob('t_array', PK.t_array, 'L', 2.0, 'N', 100);
%   FVM.k_exchange = 0.20;           % reduced exchange after atrophy (illustrative)
%   FVM.k_uptake   = 0.12;           % slightly higher uptake (illustrative)
%   FVM.D          = 3.6e-3;         % cm^2/h ~ 1e-6 cm^2/s
%   Ct = FVM.Profiles(Cpl);          % [Nt x Nx] interstitial concentration
%   FVM.plotProfiles(Cpl, [2 6 12]); % snapshots at 2h, 6h, 12h
```

```
classdef MuscleFVM_Bob
    %% ===== Changeable properties =====
    properties
        L (1,1) {mustBePositive} = 2.0             % cm, slab length (domain size)
        N (1,1) {mustBeInteger,mustBePositive} = 100 % number of FV cells
        D (1,1) {mustBePositive} = 3.6e-3          % cm^2/h, diffusion coefficient
        k_uptake (1,1) {mustBeNonnegative} = 0.10  % 1/h, first-order tissue uptake
        k_exchange (1,1) {mustBeNonnegative} = 0.30% cm/h, boundary exchange at x=0
        t_array (1,:) double = linspace(0,12,1200) % time points [h]
        useImplicit (1,1) logical = false          % false=explicit, true=implicit (Backward Euler)
    end

    %% ===== Derived (private helpers) =====
    properties (Access=private)
        Dx          % cell size [cm] = L/N
        dt          % time step [h] from t_array
        Nt          % number of time steps
        A_mat       % system matrix for implicit step for Backward Euler
    end

    %% ===== Public API =====
```

CODE (FINITE VOLUME)

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - k_{\text{uptake}}\,C$$

$$-D\frac{\partial C}{\partial x}\bigg|_{x=0} = k_{\text{exchange}}\,(C_p(t) - C(0,t))$$

$$\frac{\partial C}{\partial x}\bigg|_{x=L} = 0$$

$$\frac{dC_i}{dt} = \frac{F_{i-1/2} - F_{i+1/2}}{\Delta x} - k_{\text{uptake}}C_i$$

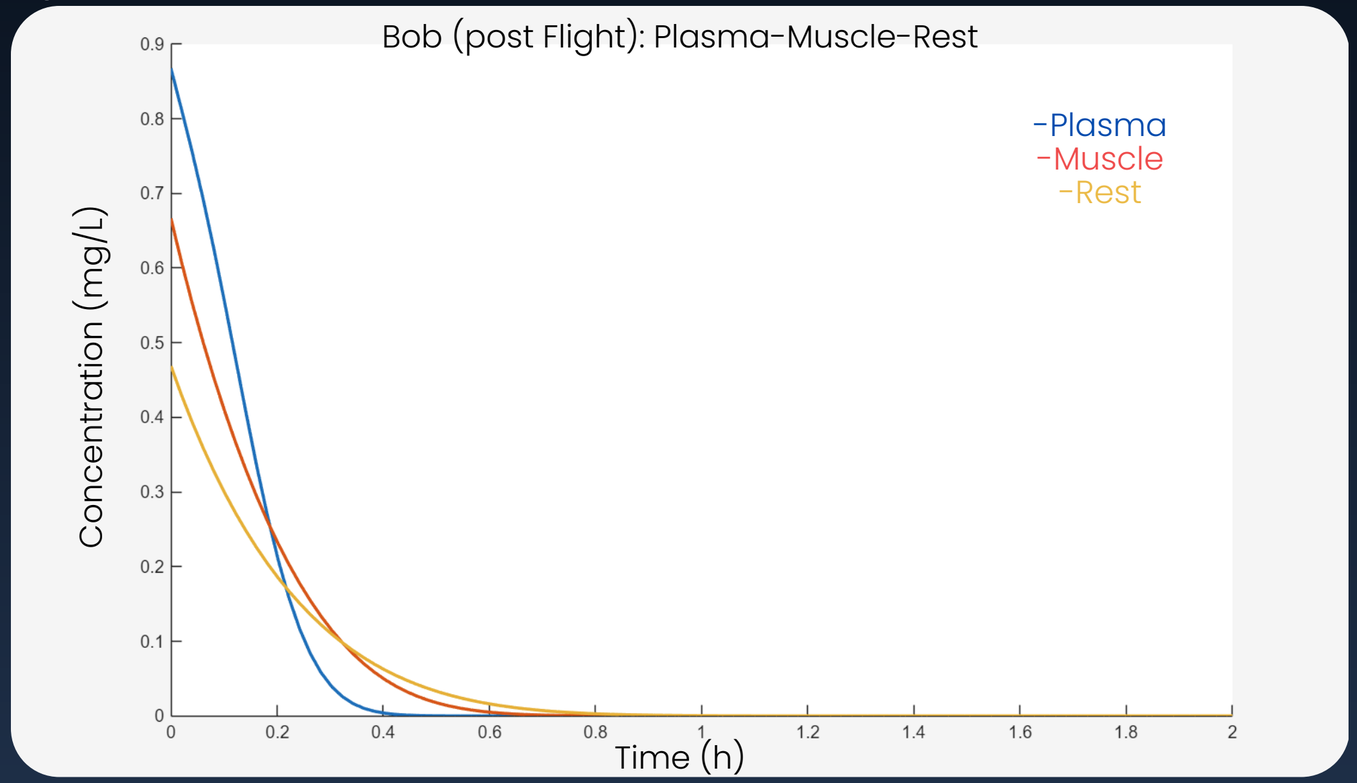
MORE CODE

```
methods
    function obj = MuscleFVM_Bob(varargin)
        if ~isempty(varargin)
            p = inputParser;
            addParameter(p,'L',obj.L);
            addParameter(p,'N',obj.N);
            addParameter(p,'D',obj.D);
            addParameter(p,'k_uptake',obj.k_uptake);
            addParameter(p,'k_exchange',obj.k_exchange);
            addParameter(p,'t_array',obj.t_array);
            addParameter(p,'useImplicit',obj.useImplicit);
            parse(p,varargin{:});
            f = fieldnames(p.Results);
            for k=1:numel(f), obj.(f{k}) = p.Results.(f{k}); end
        end
        obj.Dx = obj.L / obj.N;                          % FV cell width
        obj.Nt = numel(obj.t_array);                     % number of time samples
        obj.dt = obj.t_array(2) - obj.t_array(1);        % assumes uniform time spacing
    end

    function Ct = Profiles(obj, Cp_time)
        % Return [Nt x N] interstitial concentration profile [mg/L].
        % INPUT: Cp_time must be same length as obj.t_array (plasma concentration vs time).
        validateattributes(Cp_time, {'double'},{'vector'});
        if numel(Cp_time) ~= obj.Nt
            error('Cp_time length (%d) must equal t_array length (%d).', numel(Cp_time), obj.Nt);
        end

        % ---- Quick stability (CFL) check for explicit ----
     if ~obj.useImplicit
         % WARNING: Explicit diffusion can blow up if dt is too
         % large. For 1D diffusion explicits, typical condition:
         %  dt <= Dx^2/(2D)
         cfl = obj.Dx^2/(2*obj.D);
         if obj.dt > cfl
             warning('Explicit step may be unstable: dt=%.4g h > Dx^2/(2D)=%.4g h', obj.dt, cfl);
         end
     else
         % Prebuild implicit matrix (Backward Euler)
         obj = obj.buildImplicitSystem();
     end

     % Initialize tissue concentrations: Storage for concentration
     % vs time and space
     Ct = zeros(obj.Nt, obj.N);  % start at 0 mg/L

     % Time stepping loop (main simulation)
     for n = 1:obj.Nt-1
         C  = Ct(n,:).';  % current tissue profile [N x 1]
         Cp = Cp_time(n); % plasma concentration at current time
```

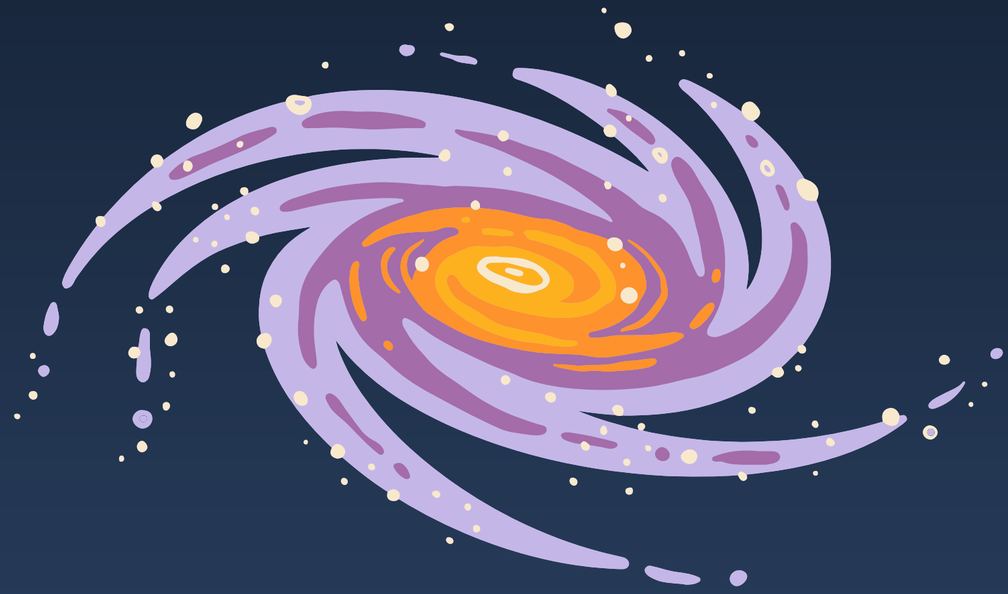
EVEN MORE CODE

```
        if ~obj.useImplicit
            % ---------- Explicit conservative FVM ----------
            %  Compute internal face fluxes F = -D*(C(i+1)-C(i))/Dx
            F = -obj.D * diff(C) / obj.Dx;

            % Mass balance in each control volume: dC_i/dt = (F_in
            % - F_out)/Dx - k_uptake * C_i
            dC = zeros(obj.N,1);

            % Internal cells have left and right internal fluxes
            dC(2:obj.N-1) = (F(1:obj.N-2) - F(2:obj.N-1))/obj.Dx - obj.k_uptake * C(2:obj.N-1);

            % Left boundary (Robin): flux into cell 1 is k_ex*(Cp - C(1))
            F_left = obj.k_exchange * (Cp - C(1));
            dC(1) = (F_left - F(1))/obj.Dx - obj.k_uptake * C(1);

            % Right boundary (Neumann, zero flux): last face flux = F(end), next=0
            dC(obj.N) = (F(obj.N-1) - 0.0)/obj.Dx - obj.k_uptake * C(obj.N);

            C_next = C + obj.dt * dC;
            C_next = max(C_next, 0);  % non-negativity safeguard
            Ct(n+1,:) = C_next.';
        else
            % ---------- Implicit Backward Euler ----------
            % (I - dt*L) * C^{n+1} = C^n + dt*(k_ex/Dx)*Cp^{n+1} in first row
            rhs = C;                          % base BE RHS
            Cp_next = Cp_time(n+1);
            rhs(1) = rhs(1) + obj.dt * (obj.k_exchange/obj.Dx) * Cp_next;

            C_next = obj.A_mat \ rhs;     % tridiagonal solve
            C_next = max(C_next, 0);
            Ct(n+1,:) = C_next.';
        end
    end
end
```

```
        function plotProfiles(obj, Cp_time, snap_hours)
            % Plot profiles at requested times (in hours)
            Ct = obj.Profiles(Cp_time);
            x  = linspace(0, obj.L, obj.N);

            figure; hold on;
            for k = 1:numel(snap_hours)
                [~, idx] = min(abs(obj.t_array - snap_hours(k)));
                plot(x, Ct(idx,:), 'LineWidth', 1.8, ...
                    'DisplayName', sprintf('t = %.1f h', obj.t_array(idx)));
            end
        end
    end
end
```



RESULTS (FINITE VOLUME)



Concentration is highest in plasma at initial injection. Muscle concentration starts lower than plasma and decays more slowly (enters through permeability-limited exchange). The rest retains drug at lower levels compared to plasma and muscle.

CONCLUSION

We determined the simulated rate of both the decrease in plasma and muscle volume impacted transport of Bolus shot along with its concentration in the muscle, plasma, and rest of the cells in the extremity using pharmokinetics and finite volume methods.

WORKS CITED

Musculo-skeletal system: Bone and Muscle loss. (n.d.). Retrieved December 16, 2025, from https://www.esa.int/Enabling_Support/Preparing_for_the_Future/Space_for_Earth/Space_for_health/Musculo-skeletal_system_Bone_and_Muscle_loss

Vandenburgh, H., Chromiak, J., Shansky, J., Del Tatto, M., & Lemaire, J. (1999a). Space travel directly induces skeletal muscle atrophy. FASEB Journal: Official Publication of the Federation of American Societies for Experimental Biology, 13(9), 1031–1038. https://doi.org/10.1096/fasebj.13.9.1031

Vandenburgh, H., Chromiak, J., Shansky, J., Del Tatto, M., & Lemaire, J. (1999b). Space travel directly induces skeletal muscle atrophy. The FASEB Journal, 13(9), 1031–1038. https://doi.org/10.1096/fasebj.13.9.1031