Part 1: Mitotic Trigger Waves - Reproduced Results

PDE Model of Cdk1 Activation and Propagation (Trigger Waves)

The PDE Model and its interfaces can be seen in the script *cdk1_act_inact_pde.m.*

Model and Simulation parameters.

```
consts.scale = 10;

% diffusion constant in micro-m^2 min^-1.
consts.Dp = 600;

% EC50 concentrations in nM.
consts.EC50deg = 32;
consts.EC50cdc25 = 35;
consts.EC50wee1 = 30;

% time duration.
consts.tmax = 400;
consts.del_t = 1;

% x range is from -200 to 200.
consts.xrange = 2000;
```

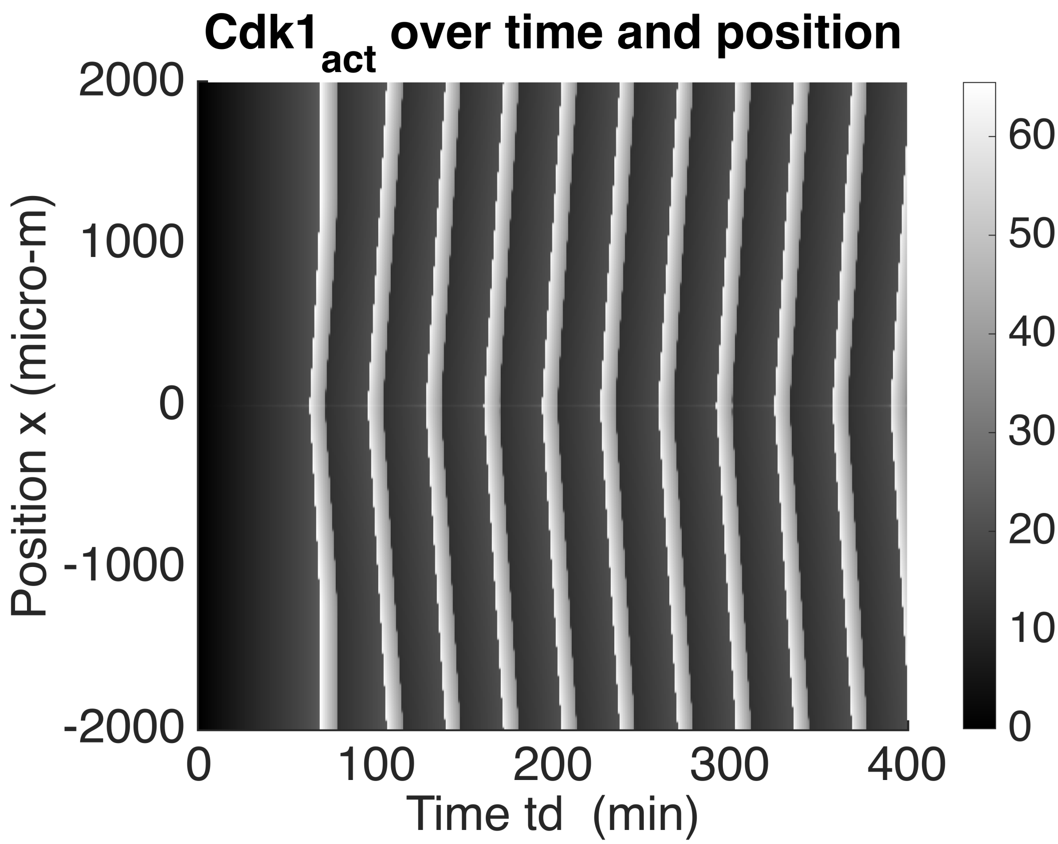
Solve the system of PDEs.

```
% define time interval and x-mesh.
tspan = linspace(0, consts.tmax, consts.tmax/consts.del_t);
xmesh = linspace(-1*consts.xrange, consts.xrange, 2*consts.xrange);

% get PDE functions.
pde_funcs = cdk1_act_inact_dt;

% solve PDE!
sol = pdepe(0, ...
    @pde_funcs.Cdk1_PDE, ...
    @pde_funcs.Cdk1_init, ...
    @pde_funcs.Cdk1_boundaries, ...
    xmesh, tspan, [], consts ...
);

% unpack solutions.
cdk1_act = sol(:, :, 1);      % active Cdk1 solution.
cdk1_inact = sol(:, :, 2);    % inactive Cdk1 solution.
```

Plot Active Cdk1 concentration as a function of time and position.

```
figure; hold on; zoom on;
surf(tspan, xmesh, cdk1_act', cdk1_act', 'EdgeColor', 'none');
```

```
colormap gray; colorbar; view(0, 90);
title('Cdk1_{act} over time and position');
xlabel('Time td  (min)');
ylabel('Position x (micro-m)');
PrettyFig;
```



[This plot can be compared to Figure 1D of the Chang et al. "Mitotic trigger waves and the spatial coordination of the Xenopus cell cycle." *Nature* 500.7464 (2013): 603-607.]

Estimating the propagation speed based on the plot.

Trigger Wave Propagation Speed: $\frac{2000 - 0}{305.75 - 289.25} = \frac{2000}{16.5} \approx 121 \mu m\, min^{-1}$

Estimation Origin of Contraction Waves based on Time of Incidence at Surface

```
wave_funcs = surface_contraction_waves();
[surface_eqn, wave_eqn] = wave_funcs.surface_intersection_eqns()
```

surface_eqn = $x^2 + y^2 = \mathrm{rc}^2$

wave_eqn = $(x - x_0)^2 + (y - y_0)^2 = v^2\ (t - t_0)^2$

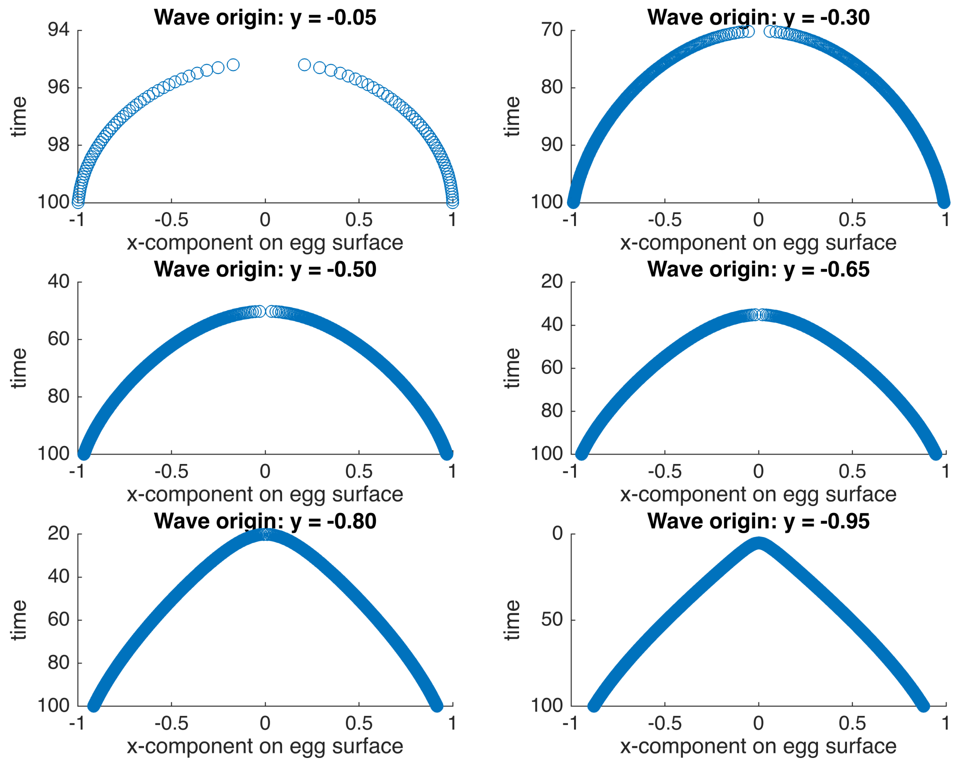
Solve the equation for (x, y) coordinate.

```
xy_soln = solve([surface_eqn wave_eqn], [sym("x") sym("y")]);
```

Plot the position of wave incidence against time.

Different choices of surface wave origin for the same parameter values. *All values, except time, are expressed as a ratio to Rc, the radius of the egg cell.*

```
wave_params.y0 = [-0.05 -0.30 -0.5 -0.65 -0.8 -0.95];
wave_params.x0 = 0.001;
wave_params.t0 = 0.1;
wave_params.v = 0.01;
wave_params.rc = 1;
time_pts = linspace(0, 100, 1000);

figure; hold on;
for k=1:length(wave_params.y0)
    % get x over time and retain all real results.
    syms x0 y0 t0 v rc t;
    xt_reln = double(subs( ...
        xy_soln.x, ...
        {y0 x0 t0 v rc t}, ...
        {wave_params.y0(k) wave_params.x0 wave_params.t0 wave_params.v
wave_params.rc time_pts} ...
    ));

    % save real results.
    idx = 1;
    xt_plot_pts = [];
    for i=1:length(time_pts)
        if isreal(xt_reln(1, i))
            xt_plot_pts(idx, 1) = xt_reln(1, i);
            xt_plot_pts(idx, 2) = time_pts(i);
            idx = idx + 1;
        end
        if isreal(xt_reln(2, i))
            xt_plot_pts(idx, 1) = xt_reln(2, i);
            xt_plot_pts(idx, 2) = time_pts(i);
            idx = idx + 1;
        end
    end

    % plot real results.
    subplot(3, 2, k);
    scatter(xt_plot_pts(:, 1), xt_plot_pts(:, 2));
    title(sprintf("Wave origin: y = %.2f", wave_params.y0(k)));
    xlabel('x-component on egg surface');
    ylabel('time')
    set(gca, 'YDir','reverse');
end
```



[*This plot can be compared to concept presented in Figure 4 of* Chang et al. "Mitotic trigger waves and the spatial coordination of the Xenopus cell cycle." *Nature* 500.7464 (2013): 603-607.]

Part 2: ODE Model of Cdk1-APC/C Cell Cycle Oscillation - Extension

ODE Model of Cdk1-APC/C Cell Cycle Oscillator

Model Parameters.

```
% sysnthesis rate in nM/min.
ode_params.ksynth =    1;

% positive feedback strength.
ode_params.pos_fb_strength = 1;

% rate params in min^-1.
ode_params.adeg    = 0.01;
ode_params.bdeg    = 0.04;
ode_params.acdc25 = 0.16;
ode_params.bcdc25 = 0.80;
ode_params.awee1 = 0.08;
ode_params.bwee1 = 0.40;
```

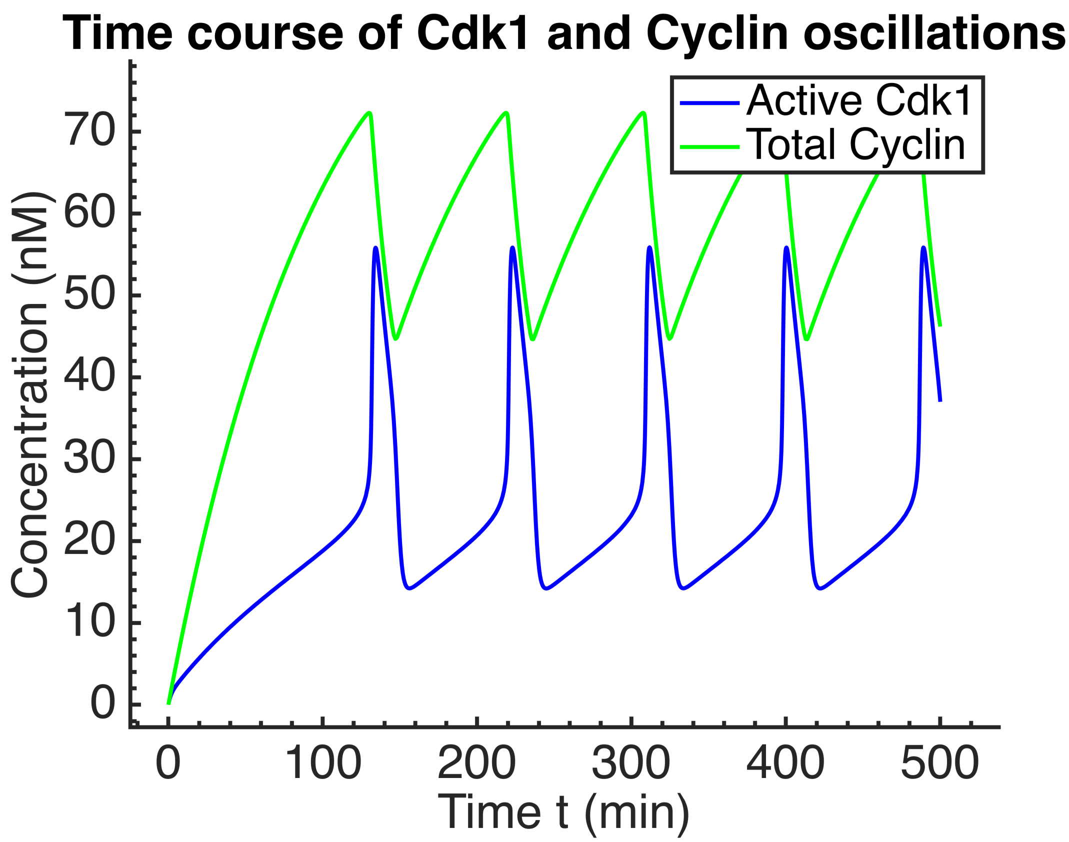
```
% Hill coefficients.
ode_params.ncdc25 = 11;
ode_params.nwee1 = 3.5;
ode_params.ndeg = 17;

% concentrations in nM.
ode_consts.EC50deg = 32;
ode_consts.EC50cdc25 = 35;
ode_consts.EC50wee1    = 30;
```

Solve the system of ODEs.

```
% time duration.
ode_consts.tmax = 500;

% initial Cdk1 and Cyclin concentrations are 0.
ode_consts.y_init = [0; 0];

% get ODE functions.
ode_funcs = cdk1_cyclin_ode;

% solve ODE.
[ode_tout, x] = ode45(@(t,y) ode_funcs.dydt(t, y, ode_params, ode_consts),
[0; ode_consts.tmax], ode_consts.y_init);
```

Unpack and Plot the ODE solutions.

```
% unpack.
ode_cdk1 = x(:, 1);
ode_cyclin = x(:, 2);
% plot.
figure; hold on; zoom on;
plot(ode_tout, ode_cdk1, 'Color', 'blue');
plot(ode_tout, ode_cyclin, 'Color', 'green');
title('Time course of Cdk1 and Cyclin oscillations');
xlabel('Time t (min)');
ylabel('Concentration (nM)');
legend('Active Cdk1', 'Total Cyclin');
PrettyFig;
```



Time periods of oscillations.

- *Active Cdk1*: (245 - 155) = 90 min.
- *Total Cyclin*: (235 - 145) = 90 min.

```
CyclePrettyFig;
```

Time periods of oscillations.

- *Active Cdk1*: (245 - 155) = 90 min.
- *Total Cyclin*: (235 - 145) = 90 min.

Try different strengths of Positive Feedback.

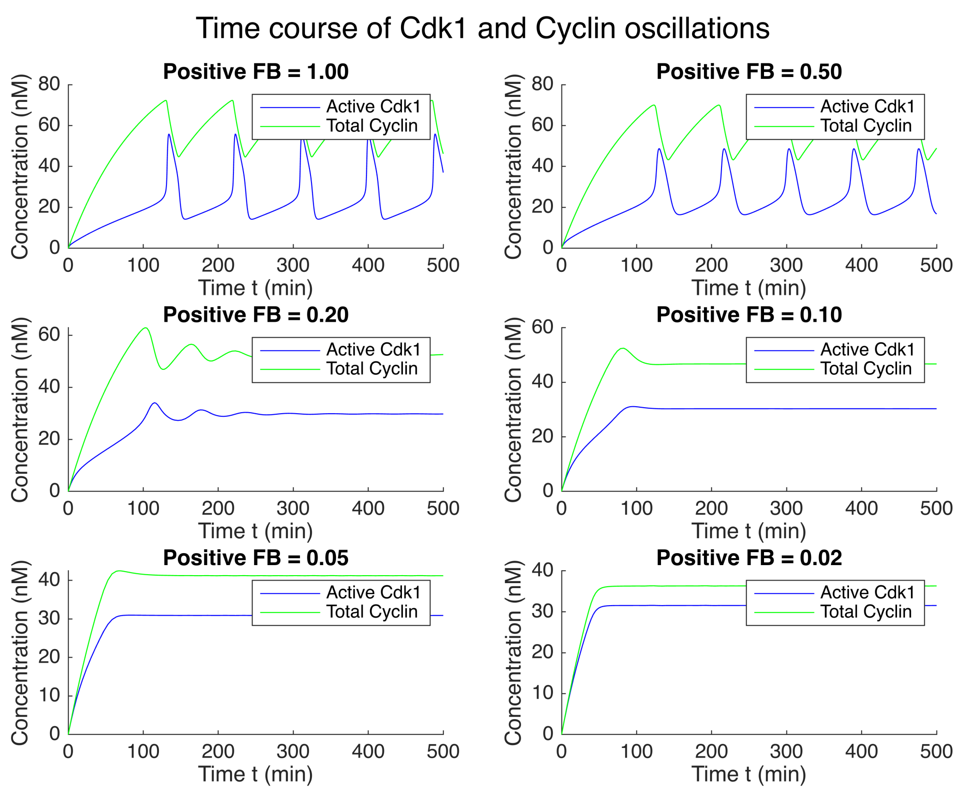
```
% positive feedback strength.
pos_fb_strength_v = [1 1/2 1/5 1/10 1/20 1/50];

figure;
for k=1:length(pos_fb_strength_v)
    % set strength.
    fprintf("Running %d ...", k);
    ode_params.pos_fb_strength = pos_fb_strength_v(k);
    ode_consts.tmax = 500;
    % initial Cdk1 and Cyclin concentrations are 0.
    ode_consts.y_init = [0; 0];
    ode_funcs = cdk1_cyclin_ode;
    % solve ODE.
    [ode_tout, x] = ode45(@(t,y) ode_funcs.dydt(t, y, ode_params,
ode_consts), [0; ode_consts.tmax], ode_consts.y_init);
    % unpack.
    ode_cdk1 = x(:, 1);
    ode_cyclin = x(:, 2);
```

```
    % plot.
    subplot(3, 2, k); hold on;
    plot(ode_tout, ode_cdk1, 'Color', 'blue');
    plot(ode_tout, ode_cyclin, 'Color', 'green');
    title(sprintf("Positive FB = %.2f", pos_fb_strength_v(k)));
    xlabel('Time t (min)');
    ylabel('Concentration (nM)');
    legend('Active Cdk1', 'Total Cyclin');
end
```

```
Running 1 ...Running 2 ...Running 3 ...Running 4 ...Running 5 ...Running 6 ...
```

```
sgtitle("Time course of Cdk1 and Cyclin oscillations");
```

```
hold off;
```



Analytically solve the ODEs to obtain Nullcline Equations.

Note:

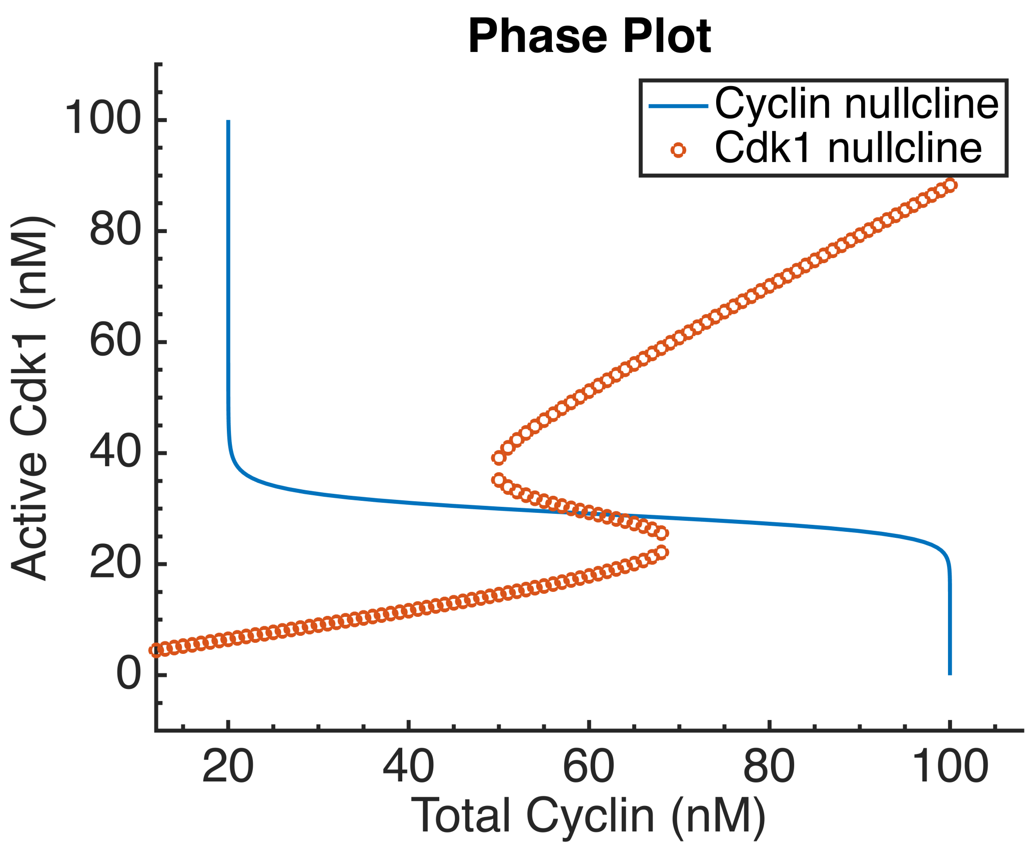
1. *The analytical solution of the Cdk1 nullcline is hard to solve; hence, this is solved numerically, taking regularly-spaced values of Cyclin concentration.*
2. *At the hysteretic region, there are multiple solutions for Cdk1 for a given Cyclin concentration; these are obtained by repeated find of numerical solution with restricted solution intervals (see cdk1_cyclin_ode.m).*

So the last two argument to the following function call specify the Cyclin concentrations for solving and the expected hyteretic region where multiple solutions are sought, respectively.

```
[ode_cdk1_nullcline_pts, ode_cyclin_nullcline] =
ode_funcs.nullclines(ode_params, ode_consts, linspace(0, 100, 101), [50
74]);
```

Plot the nullclines.

```
figure; hold on;
plot(double(subs(ode_cyclin_nullcline, sym('Cdk1'), linspace(0, 100,
201))), 0:0.5:100);
scatter(ode_cdk1_nullcline_pts(:, 1), ode_cdk1_nullcline_pts(:, 2));
title('Phase Plot');
xlabel('Total Cyclin (nM)');
ylabel('Active Cdk1 (nM)');
legend('Cyclin nullcline', 'Cdk1 nullcline');
PrettyFig;
```



The steady state occurs at about (61, 29) nM.

Plot the vector field against the nullclines to see the oscillation region.

```
[ode_cdk1_samps, ode_cyc_samps] = meshgrid([0:10:100],[0:10:100]);
```

```
% get ODE functions.
ode_funcs = cdk1_cyclin_ode;

% get rate values at all points in the mesh.
dydt = ode_funcs.dydt_array(0, cat(3, ode_cdk1_samps, ode_cyc_samps),
ode_params, ode_consts);
figure; zoom on; hold on;
quiver(ode_cdk1_samps, ode_cyc_samps, dydt(:, :, 1)', dydt(:, :, 2)',
'color', 'k');
plot(double(subs(ode_cyclin_nullcline, sym('Cdk1'), linspace(0, 100,
201))), 0:0.5:100);
scatter(ode_cdk1_nullcline_pts(:, 1), ode_cdk1_nullcline_pts(:, 2));
title('Vector Field on Phase Plot');
xlabel('Total Cyclin (nM)');
ylabel('Active Cdk1 (nM)');
legend('Vector field', 'Cyclin nullcline', 'Cdk1 nullcline');
PrettyFig;
```