11/16/2022

Home Work 6

BME 7410

Submitted by,

Rana Mozumder

DEPARTMENT OF BME, VANDERBILT UNIVERSITY

VANDERBILT UNIVERSITY

7.2 ODE model of convection-diffusion-reaction:

(a) Given that,

$$\frac{1}{Pe}\frac{d^2c}{dx^2} - \frac{dc}{dx} = Da \cdot c$$

$$\Rightarrow \frac{d^2c}{dx^2} = Pe\frac{dc}{dx} + Pe \cdot Da \cdot c$$

$$\Rightarrow \frac{d^2c}{dx^2} = Pe\left(\frac{dc}{dx} + Da \cdot c\right) \quad \text{—— (1)}$$

let, $c = y_1$

$\frac{dc}{dx} = y_2$

$$\therefore y_1' = y_2$$

from (1) $\Rightarrow$ $y_2' = Pe\,(y_2 + Da \cdot y_1)$

— set of coupled first-order ODEs.

(b) Jacobian is given by,

$$\underset{\approx}{J} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ Pe \cdot Da & Pe \end{bmatrix}$$

So, the eigen values are found from:

$$|\underset{\approx}{J} - \lambda \underset{\approx}{I}| = 0$$

$$\Rightarrow \begin{vmatrix} -\lambda & 1 \\ PeDa & Pe-\lambda \end{vmatrix} = 0$$

$$\Rightarrow -\lambda(Pe-\lambda) - Pe \cdot Da = 0$$

$$\Rightarrow \lambda^2 - Pe \cdot \lambda - Pe \cdot Da = 0$$

$$\Rightarrow \lambda = \frac{Pe \pm \sqrt{Pe^2 + 4Pe.Da}}{2}$$

$$\therefore \lambda_1 = \frac{Pe \pm \sqrt{Pe^2 + 4PeDa}}{2}, \quad \lambda_2 = \frac{Pe - \sqrt{Pe^2 + 4PeDa}}{2}$$

(c) Since Pe and Da are usually positive, the real part of λ_1 is always positive. Hence, the system becomes unstable.

Other possibilities:

\# If both Pe, Da are negative, the system will be unstable since real $(\lambda_1) > 0$.

\# If $Pe = 0$, both $\lambda_1, \lambda_2 = 0$. Hence, system is neutrally stable.

\# If $Pe < 0$ and $Da > 0$, λ_2 will always be negative.

Now, if,

$|Pe| > |\sqrt{Pe^2 + 4PeDa}|$, real $(\lambda_1) < 0$. So the system is stable.

$|Pe| < |\sqrt{Pe^2 + 4PeDa}|$, real $(\lambda_1) > 0$. So the system is unstable.

7.4 Population dynamics among hydrocarbon-consuming bacterial species

Matlab script for bacterial population dynamics:

```
% Evaluate slope f(t,y) of coupled ODEs for bacterial population dynamics

%Constants
mux1max = 0.185;   % /hr
mux2max = 0.185;   % /hr
Kx1 = 1e-05;       % g/L
Kx2 = 5e-06;       % g/L
Ki = 1e-04;        % g/L
invYx1O2 = 1/0.2;  % g/g
invYx1S = 1/5;     % g/g
invYx2s = 1/0.3;     % g/g
K1a = 42;          % /hr
CO2s = 0.008;      % g/L
FbyV = 0.08;       % /hr

rx1 = (mux1max*y(1)*y(3))/((Kx1+y(1))*(1+y(2)/Ki));
rx2 = (mux2max*y(2)*y(4))/(Kx2+y(2));

f = [K1a*(CO2s-y(1))-invYx1O2*rx1-FbyV*y(1); ...
    -FbyV*y(2)+invYx1S*rx1-invYx2s*rx2; ...
    -FbyV*y(3)+rx1; ...
    -FbyV*y(4)+rx2];
```

Matlab script for RK4 method:

```
function [t, y] = RK4methodvectorized(odefunc, tf, y0, h)
% Classical RK-4 method is used to solve coupled first-order ODEs

% Input variables
% odefunc : name of the function that calculates f(t, y)
% tf : final time or size of interval
% y0 : vector of initial conditions y(0)
% h : step size

% Other variables
n = length(y0); % number of dependent time-varying variables

% Output variables
t = [0:h:tf]; % vector of time points
y = zeros(n, length(t)); % dependent variable vector
y(:,1) = y0; % initial condition at t = 0

% RK-4 method for solving coupled first-order ODEs
for k = 1:length(t)-1
    k1 = feval(odefunc, t(k), y(:,k));
    k2 = feval(odefunc, t(k)+ h/2, y(:,k) + h/2*k1);
    k3 = feval(odefunc, t(k)+ h/2, y(:,k) + h/2*k2);
    k4 = feval(odefunc, t(k)+ h, y(:,k) + h*k3);
    y(:,k+1) = y(:,k) + h/6*(k1 + 2*k2 + 2*k3 + k4);
    %so that concentration of methanol doesn't become negative
```

```
    if y(2, k+1)<0
        y(2, k+1) = 0;
    end
end
```

Matlab script for Euler forward method:

```
function [t, y]= eulerforwardmethodvectorized(odefunc, tf, y0, h)
% Euler's forward method is used to solve coupled first-order ODEs

% Input variables
% odefunc : name of the function that calculates f(t, y)
% tf : final time or size of interval
% y0 : vector of initial conditions y(0)
% h : step size

% Other variables
n = length(y0); % number of dependent time-varying variables

% Output variables
t = [0:h:tf]; % vector of time points
y = zeros(n, length(t)); % dependent variable vector
y(:,1) = y0; % initial condition at t = 0
             % indexing of matrix elements begins with 1 in MATLAB

%Euler's forward method for solving coupled first-order ODEs
for k = 1:length(t)-1
    y(:,k+1) = y(:,k)+ h*feval(odefunc, t(k), y(:,k));

    %so that concentration of methanol doesn't become negative
    if y(2, k+1)<0
        y(2, k+1) = 0;
    end
end
```

Matlab script for executing the solution methods:

```
% Variables
y0(1) = 1e-05;      % initial value of Co2 in g/L
y0(2) = 1.6;        % initial value of S in g/L
y0(3) = 0.672;      % initial value of x1 in g/L
y0(4) = 0.028;      % initial value of x2 in g/L
tf = 72;            % interval size (hour)
h = 0.01;           % step size (hour), 0.1, 0.01, 0.001 all were evaluated

% Call ODE Solver
% uncomment any one of these to execute that method
% [t, y] = RK4methodvectorized('bacterialpopulationdynamics', tf, y0, h);
% [t, y] = eulerforwardmethodvectorized('bacterialpopulationdynamics', tf, y0, h);
% [t, y] = ode113('bacterialpopulationdynamics', tf, y0);
% [t, y] = ode23tb('bacterialpopulationdynamics', tf, y0);



% Plot Results
figure;
```

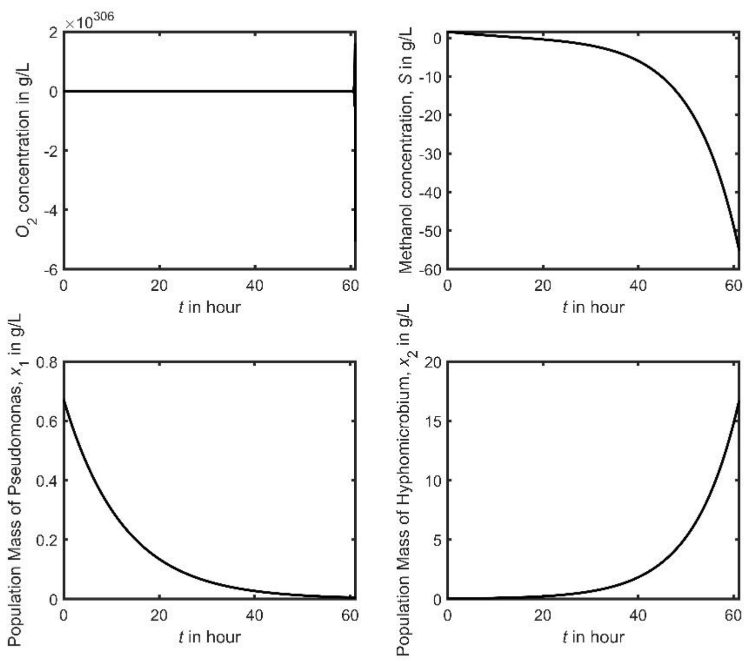
```
subplot(2,2,1)
plot(t,y(1,:),'k-','LineWidth',2)
set(gca,'FontSize',12,'LineWidth',2)
xlabel('{\itt} in hour')
ylabel('{\itO}_2 concentration in g/L')

subplot(2,2,2)
plot(t,y(2,:),'k-','LineWidth',2)
set(gca,'FontSize',12,'LineWidth',2)
xlabel('{\itt} in hour')
ylabel('Methanol concentration, {\itS} in g/L')

subplot(2,2,3)
plot(t,y(3,:),'k-','LineWidth',2)
set(gca,'FontSize',12,'LineWidth',2)
xlabel('{\itt} in hour')
ylabel('Population Mass of Pseudomonas, {\itx}_1 in g/L')

subplot(2,2,4)
plot(t,y(4,:),'k-','LineWidth',2)
set(gca,'FontSize',12,'LineWidth',2)
xlabel('{\itt} in hour')
ylabel('Population Mass of Hyphomicrobium, {\itx}_2 in g/L')
```

Results:

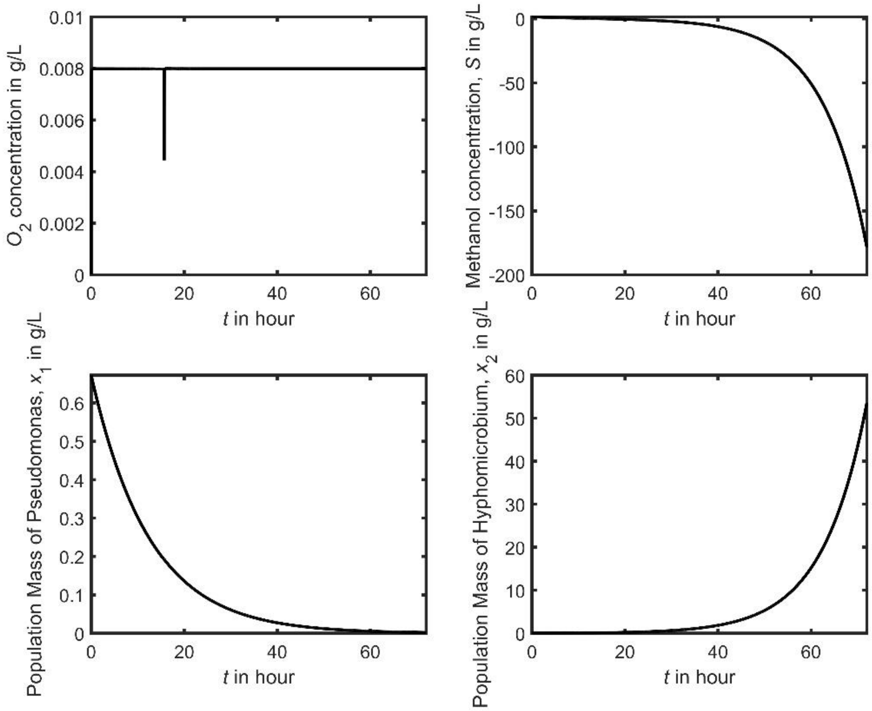
Now, both RK4 and Euler forward method was used with different step size to solve the ODEs.

Euler forward method:

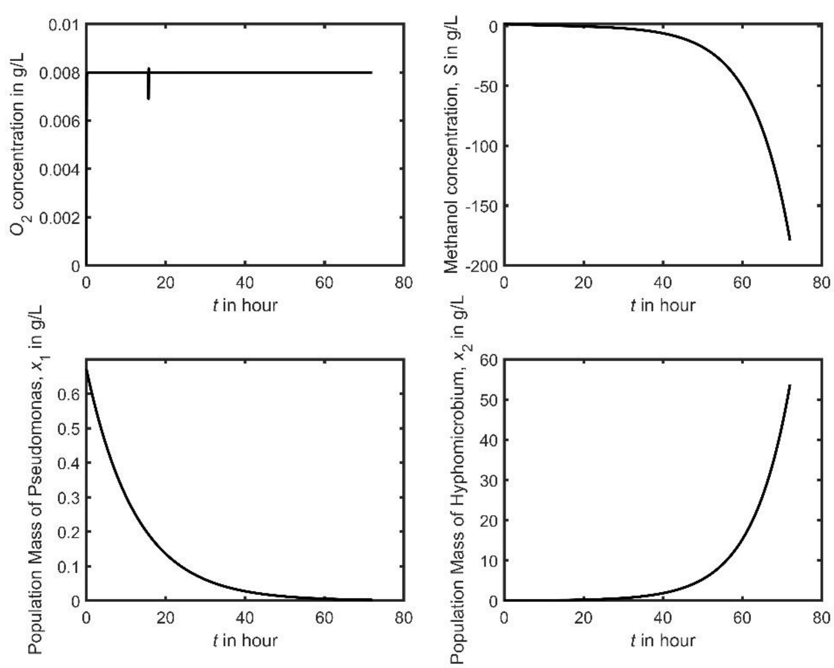
For stepsize = 0.1 hr;



For step size=0.01 hr;

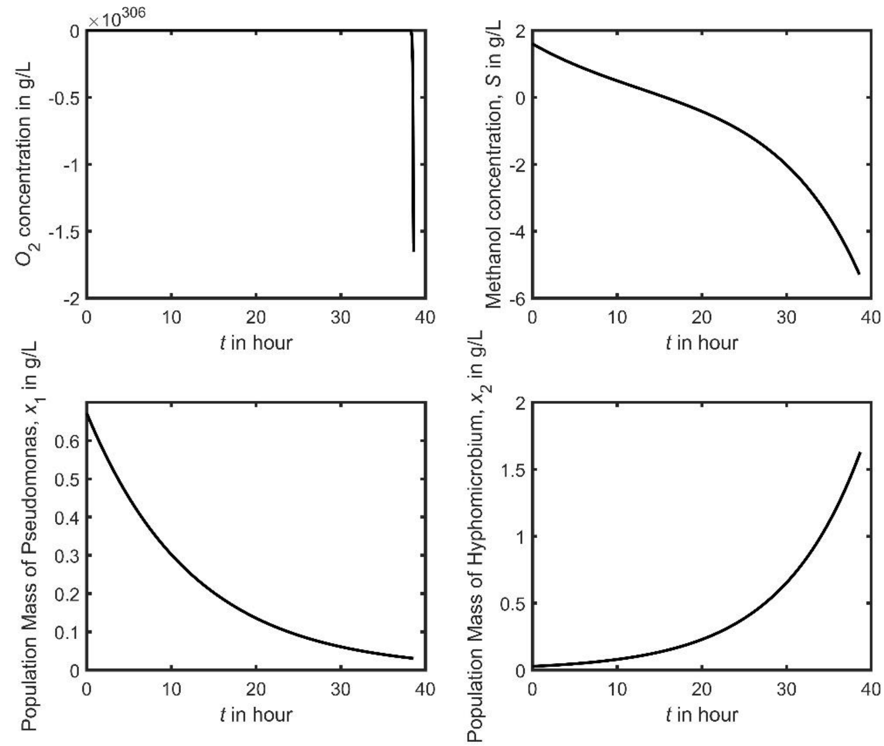


For step size =0.001 hr;

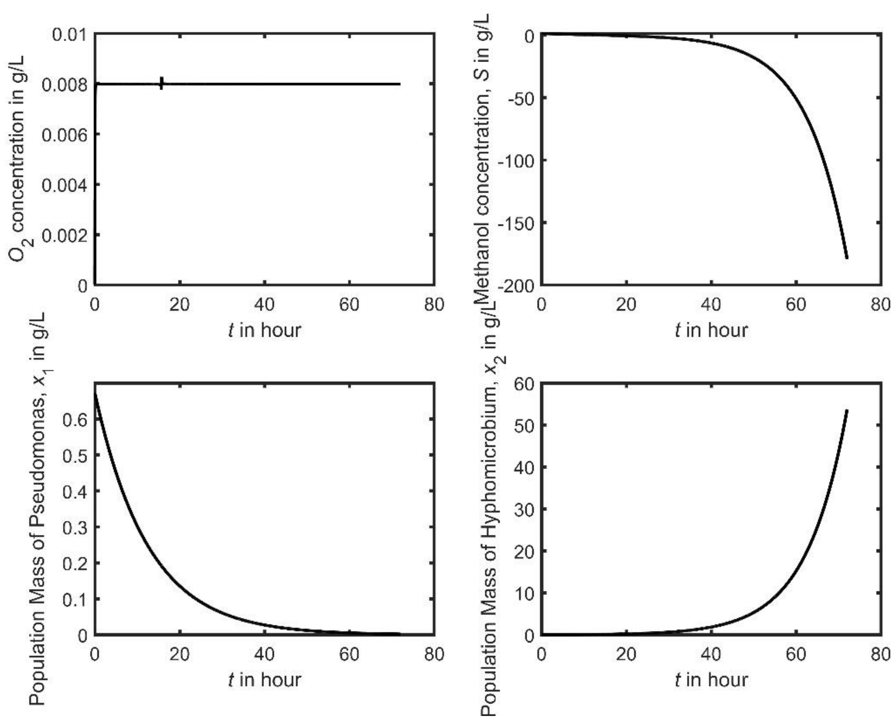


RK4 method:

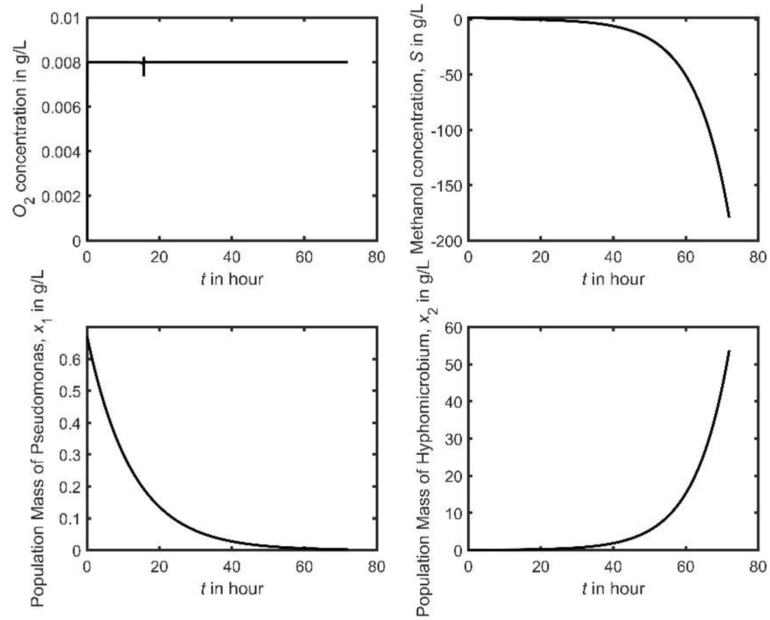
For step size =0.1hr;



For step size=0.01 hr;

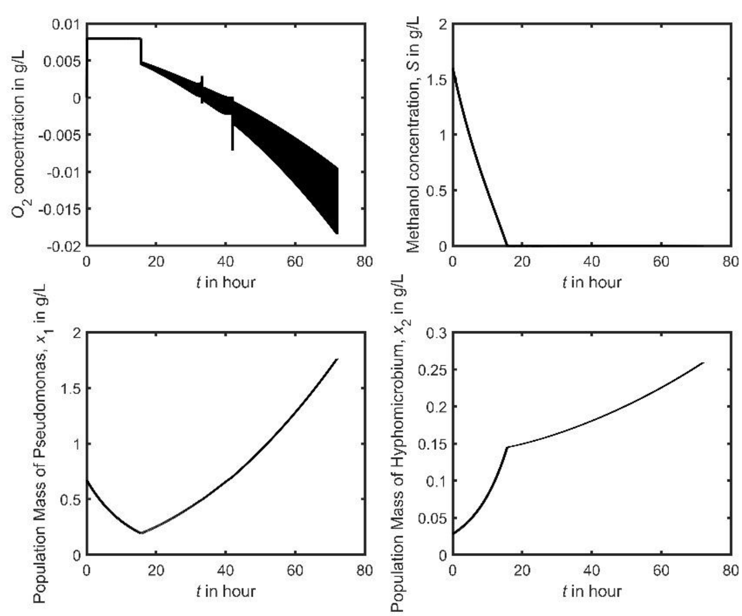


For step size=0.001 hr;



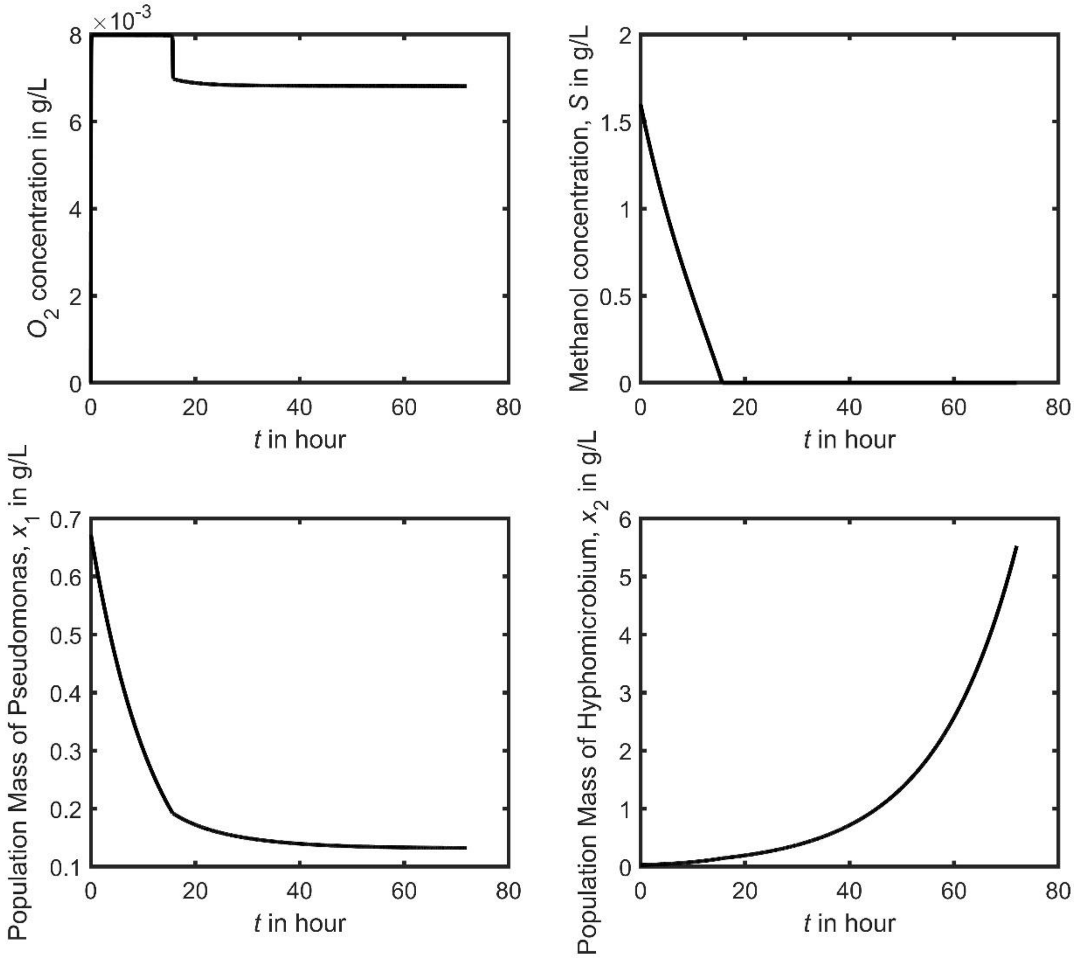
It seems like 0.01 is a good step size for the ODEs. However, in all these variations, the result doesn't make sense as concentration of methanol goes to negative. Between 16-17 hours, the concentration of methanol becomes zero. If S becomes zero, then in the next steps, it becomes more negative which in reality not possible. To prevent that, whenever the concentration of methanol is negative, it was replaced with zero. Now,

Euler forward method with step size=0.01hr:



Which is clearly not stable in transient response of C_{O2}, but results are not meaningless. Now for,

RK4 method with step size=0.01hr:



Except for population mass of Hyphomicrobium, x_2, the response become steady as time goes towards 72 hr/3 days. But x_2 doesn't reach steady state.

The ode113, ode23, ode45, ode15i, ode23t, odetb- all these methods were also evaluated. None of them could give meaningful results.