HW3 Implicit Temporal Integrators

June Yang

October 2019

1 Implicit Timestepping for the Advection Equation in 1D

Consider numerically solving the advection equation with constant coefficients

$$u_t + au_x = 0,$$

on the periodic domain $0 < x < 1$, for $a = 1$ and initial condition

$$u(x, 0) = [\sin(\pi x)]^{100},$$

on a uniform grid with $m = 1000$ points (this is the same problem as in Homework 1), using a method-of-lines (MOL) approach.

Spatially discretize this equation using the *third-order upwind biased scheme* (as in HW1), and integrate it forward in time to a time $T = 1$ using a time step size corresponding to an advective CFL number of $\nu = a\Delta t/\Delta x = 1, 10, 100$ and the following implicit temporal integrators:

1. Backward Euler method.
2. Implicit trapezodial method (Crank-Nicolson).

Compare the three numerical solutions $\boldsymbol{w}(t = 1)$ to the exact solution $u(x, 1)$ and comment on your observations. Explain how you solved the linear system arising in the implicit temporal discretization. Integrate the spatially-discrete system in time "exactly" based on your solution to HW1 (you can improve your solution based on discussion in class, of course) and show the result for comparison.

See page 3 and 4.

Take $h \to 0$ to get a semi-discretizaion in time, $u' = Au$, where A is some linear differential operator. Consider the θ method,

$$u^{n+1} = u^n + k[(1-\theta)Au^n + \theta Au^{n+1}]$$

The local truncation error of this method is

$$\begin{aligned}
\tau^n &= \frac{1}{k}[u(t_{n+1}) - u(t_n)] - [(1-\theta)Au(t_n) + \theta Au(t_{n+1})] \\
&= \frac{1}{k}\left(ku(t_n) + \frac{k^2}{2}u''(t_n) + \frac{k^3}{6}u'''(t_n)\right) - (1-\theta)Au(t_n) - \theta A\left(u(t_n) + ku'(t_n) + \frac{k^2}{2}u''(t_n) + \frac{k^3}{6}u'''(t_n)\right) \\
&= Au + \frac{k}{2}A^2u + \frac{k^2}{6}A^3u - (1-\theta)Au - \theta A\left(u + kAu + \frac{k^2}{2}A^2u + \frac{k^3}{6}A^3u\right) \\
&= k\left(\frac{1}{2} - \theta\right)A^2u + k^2\left(\frac{1}{6} - \frac{\theta}{2}\right)A^3u
\end{aligned}$$

This means that our scheme is closer to solving the modified equation $\widetilde{u}' = \widetilde{A}\widetilde{u}$, where the modified operator is

$$\widetilde{A} = A - k\left(\frac{1}{2} - \theta\right)A^2 - k^2\left(\frac{1}{6} - \frac{\theta}{2}\right)A^3$$

For pure advection equation, $A = -a\partial_x$. Thus, $A^2 = a^2\partial_{xx}$ and $A^3 = -a^3\partial_{xxx}$. Therefore, the modified equations is

i. $\theta = \frac{1}{2}$ (Crank-Nicolson)

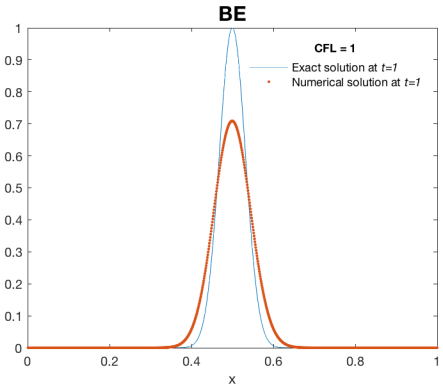
$$\widetilde{u}' = \left(A + \frac{k^2}{12}A^3\right)\widetilde{u}$$

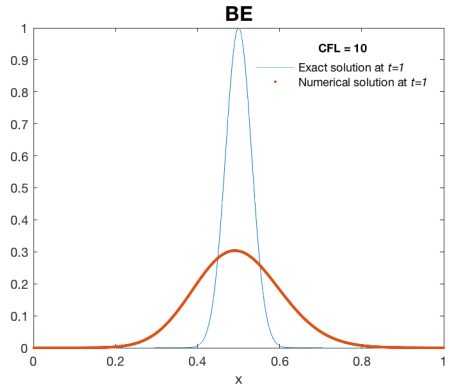
$$\widetilde{u}_t + a\widetilde{u}_x = -\frac{k^2}{12}a^3\widetilde{u}_{xxx}$$

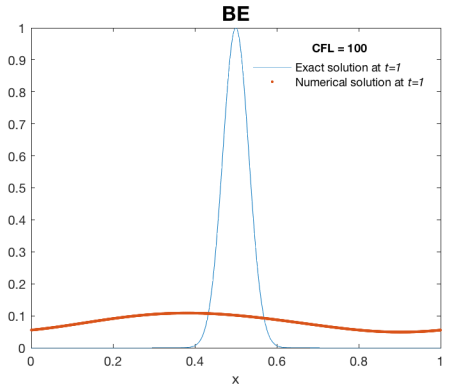
ii. $\theta = 1$ (Backward Euler)

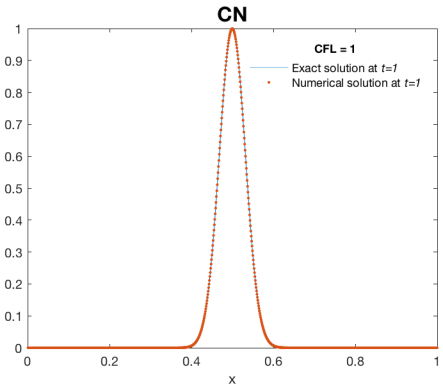
$$\widetilde{u}' = \left(A + \frac{k}{2}A^2\right)\widetilde{u}$$
$$\widetilde{u}_t + a\widetilde{u}_x = \frac{k}{2}a^2\widetilde{u}_{xx}$$

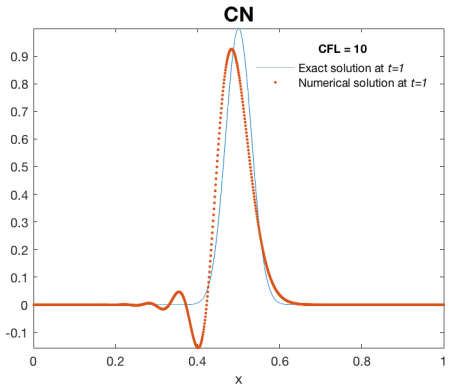
The term u_{xxx} in the modified equation of the Crank-Nicolson method leads to dispersive behavior, whereas the term u_{xx} in the modified equation of the Backward Euler method leads to dissipative behavior.

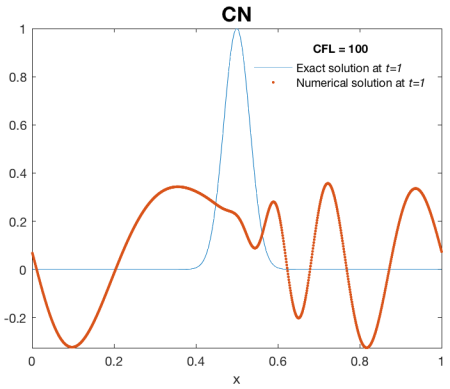

BE
CFL = 1
Exact solution at t=1
Numerical solution at t=1
x

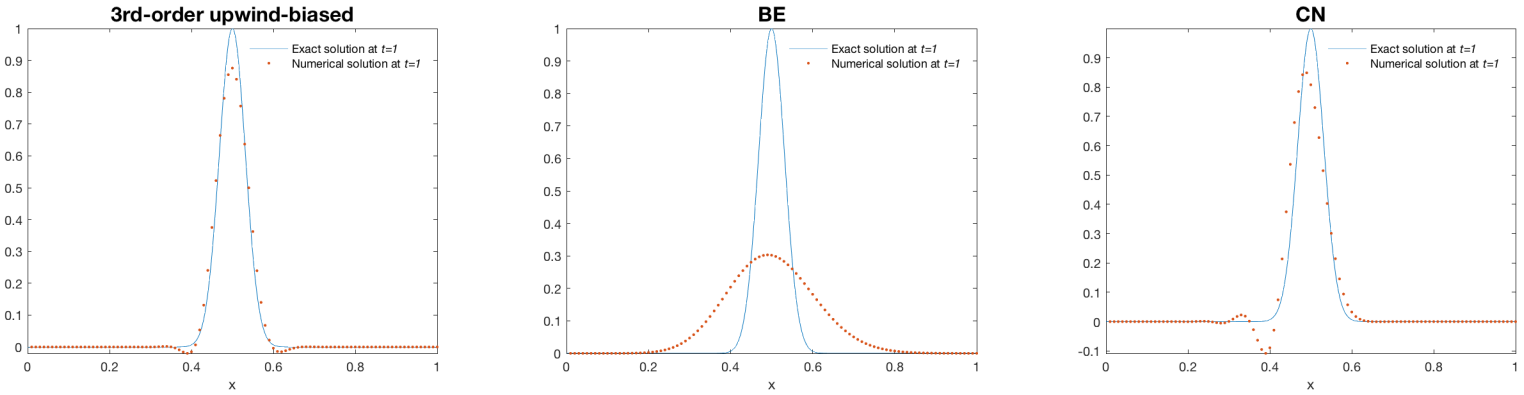

BE
CFL = 10
Exact solution at t=1
Numerical solution at t=1
x


BE
CFL = 100
Exact solution at t=1
Numerical solution at t=1
x


CN
CFL = 1
Exact solution at t=1
Numerical solution at t=1
x


CN
CFL = 10
Exact solution at t=1
Numerical solution at t=1
x


CN
CFL = 100
Exact solution at t=1
Numerical solution at t=1
x



Advection, $h = 1/100$, with the third order upwind biased scheme. i) Spatial discretization only. ii) and iii) Backward Euler method and Implicit trapezoidal method, respectively, with time step corresponding to CFL $= 1$.

2 Implicit Timestepping for the Diffusion Equation

Consider solving the diffusion equation in one dimension

$$u_t = u_{xx},$$

on the domain $0 < x < 1$ with Dirichlet BCs, $u(0,t) = 0$, $u(1,t) = 1$, and a discontinuous initial condition

$$u(x,0) = \begin{cases} 0 & \text{if } x \leq 1/2 \\ 1 & \text{otherwise.} \end{cases}$$

Choose the method of spatial discretization and use a grid spacing of $h = 1/100$. Write code to integrate the resulting semi-discrete system forward in time using the backward Euler (BE) and the implicit trapezoidal method (Crank-Nicolson, CN) for a specified time step size.

1. Using a time step size corresponding to a diffusive CFL number of $\mu = a\Delta t/\Delta x^2 = 1, 10, 100$, compare the numerical solutions $\boldsymbol{w}(t = 0.01)$ for the two methods (BE vs CN) and comment on your observations. Explain how you solved the linear system arising in the implicit temporal discretization. Try to construct the exact solution of the PDE and show that also for comparison.

$u(x,t) = x + \frac{1}{\pi}\sum_{k=1}^{\infty}(-1)^k \frac{1}{k}\sin(2k\pi x)e^{-(2k\pi)^2 t}$

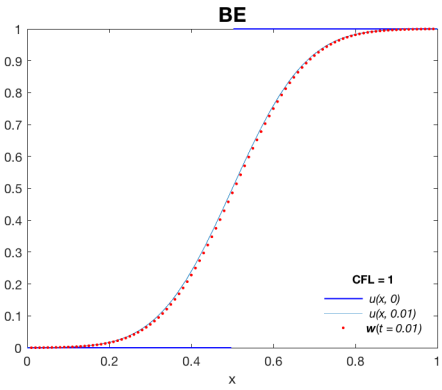
See page 6.

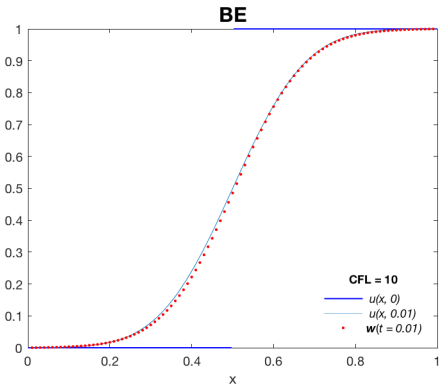
2. Compare the solutions obtained by the two temporal integrators at a much longer time $t = 10$ for a time step size corresponding to a diffusive CFL number of $\mu = 1000$ and comment on your observations. Compare the numerical solution to the steady state solution $u(x,\infty)$ of the PDE and discuss your observations of the differences between CN and BE.

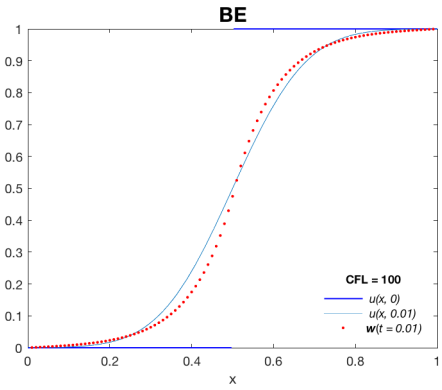
$u(x,\infty) = x$

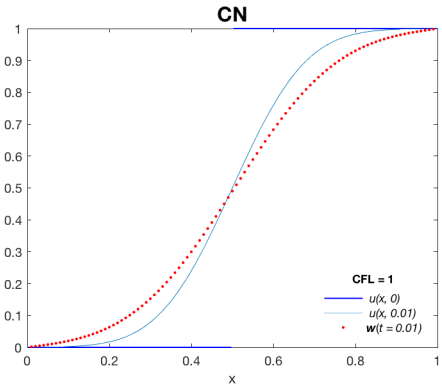
See page 7.

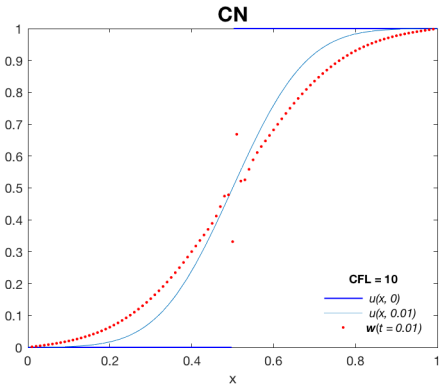
Animations: CN with CFL = 1 (0.01x) vs. CN with CFL = 100 (0.25x)

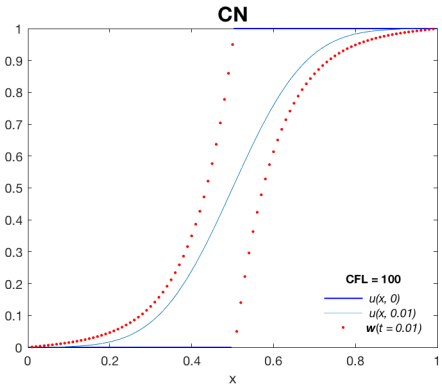

BE
CFL = 1
u(x, 0)
u(x, 0.01)
w(t = 0.01)
x

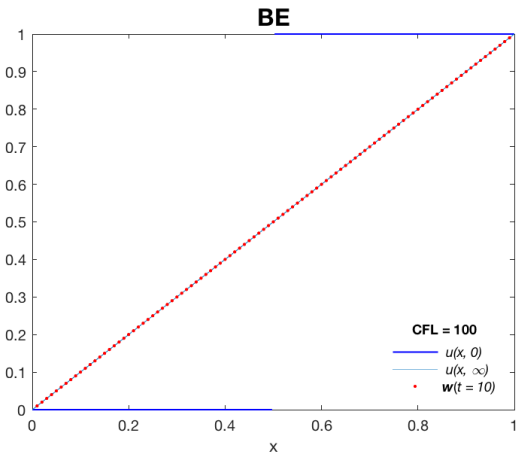

BE
CFL = 10
u(x, 0)
u(x, 0.01)
w(t = 0.01)
x

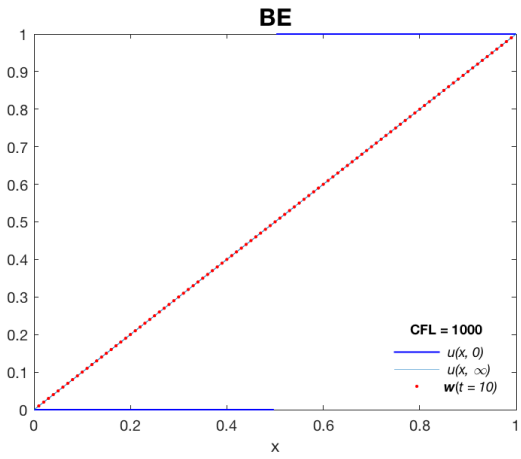

BE
CFL = 100
u(x, 0)
u(x, 0.01)
w(t = 0.01)
x

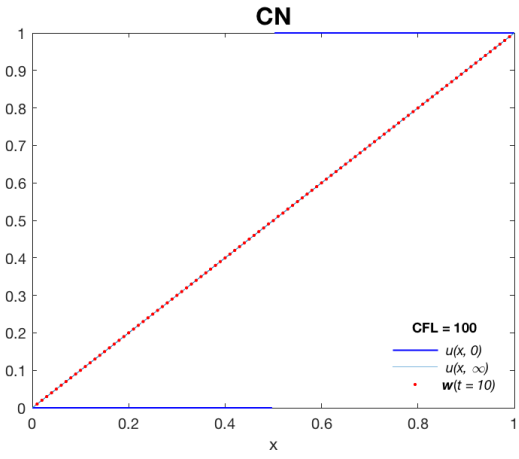

CN
CFL = 1
u(x, 0)
u(x, 0.01)
w(t = 0.01)
x

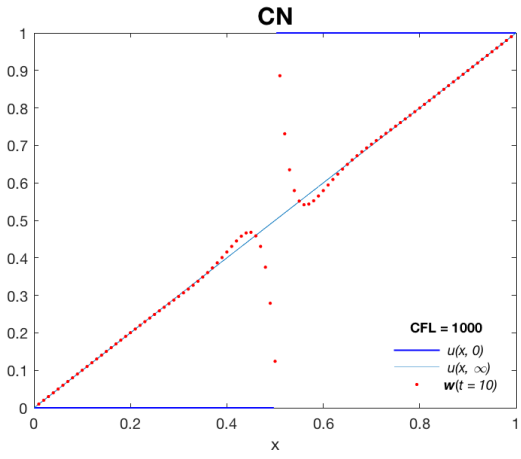

CN
CFL = 10
u(x, 0)
u(x, 0.01)
w(t = 0.01)
x


CN
CFL = 100
u(x, 0)
u(x, 0.01)
w(t = 0.01)
x


BE
CFL = 100
u(x, 0)
u(x, ∞)
w(t = 10)
x


BE
CFL = 1000
u(x, 0)
u(x, ∞)
w(t = 10)
x


CN
CFL = 100
u(x, 0)
u(x, ∞)
w(t = 10)
x


CN
CFL = 1000
u(x, 0)
u(x, ∞)
w(t = 10)
x

Code

```
% HW3 Implicit Temporal Integrators
% 1 Implicit Timestepping for the Advection Equation in 1D

a = 1;

m = 1000;
h = 1 / m; % h = dx

T = 1; % final time

f = @(x) (sin(pi * x)) .^ 100; % u(x, 0)
fplot(f, [0 1]);
hold on

A = - 1 / 2 * eye(m); % third-order upwind biased scheme
subdiag_A = ones(m - 1, 1);
A = diag(subdiag_A, -1) + diag(- 1 / 3 * subdiag_A, 1) + A;
subdiag_A = ones(m - 2, 1);
A = diag(- 1 / 6 * subdiag_A, -2) + A;
A(1, m) = 1;
A(1, m - 1) = - 1 / 6;
A(2, m) = - 1 / 6;
A(m, 1) = - 1 / 3;

c = 100; % advective CFL number c = a * dt / dx
k = c * h / a; % k = dt
final_step = T / k;

grid_x = h : h : 1;
grid_x = grid_x(:);
u_init = f(grid_x);

method = 'CN';

w = u_init; % numerical solution

for i = 1 : final_step

    switch method

    case 'FE'

        w = w + c * A * w;

    case 'BE'

        w = (eye(m) - c * A) \ w;

    case 'CN'

        w = (eye(m) - c/2 * A) \ ((eye(m) + c/2 * A) * w);

    end

end

plot(h : h : 1, w, '.')
```

```
xlabel('x')
legend('Exact solution at \itt=1', 'Numerical solution at \itt=1', 'Location',↙
'northeast');
legend('boxoff')
title(method, 'fontsize', 16)
```

```
% HW3 Implicit Temporal Integrators
% 2 Implicit Timestepping for the Diffusion Equation
% 2.1 One Dimension
% Backward Euler

h = 0.01; % h = dx
m = (1 / h) - 1;

T = 10; % 0.01 10; final time

c = 1000; % 1 10 100; diffusive CFL number c = dt / (dx)^2
k = c * (h ^ 2); % k = dt
final_step = T / k;

A = -2 * eye(m);
subdiag = ones(m - 1, 1);
A = diag(subdiag, 1) + A + diag(subdiag, -1);

grid_x = h : h : (1 - h);
grid_x = grid_x(:);
f = @(x) 1 .* (x > 0.5) + zeros(size(x)); % u(x, 0)
u_init = f(grid_x);

w = u_init; % numerical solution
bc = zeros(m, 1);
bc(m) = c; % u(1, t) = 1

for i = 1 : final_step

    w = w + bc;
    w = (eye(m) - c * A) \ w; % solve the linear system to obtain w at t = i * k

end

x = linspace(0, 1, 401);
f = @(x) 1 * (x > 0.5);
y = [f(x(1 : 200)), NaN,  f(x(202 : 401))];
plot(x, y, 'b-')
hold on

% u_approx = @(x) x - exp(-(2 * pi) ^ 2 * T) / pi * sin(2 * pi * x) + exp(-(4 * pi) ^ 2 *↙
T) / (2 * pi) * sin(4 * pi * x) - exp(-(6 * pi) ^ 2 * T) / (3 * pi) * sin(6 * pi * x) +↙
exp(-(8 * pi) ^ 2 * T) / (4 * pi) * sin(8 * pi * x);
% fplot(u_approx, [0 1]);
% hold on

u_steady = @(x) x;
fplot(u_steady, [0 1]);
hold on

plot(h : h : (1 - h), w, '.', 'Color', 'r')


lgd = legend('\it u(x, 0)', '\it u(x, \infty)', '\bf\itw\rm(\itt = 10)', 'Location',↙
'southeast');
title(lgd, ['CFL = ', num2str(c)])
legend('boxoff')
title('BE', 'fontsize', 16)
```

```
% HW3 Implicit Temporal Integrators
% 2 Implicit Timestepping for the Diffusion Equation
% 2.1 One Dimension
% Crank-Nicolson

h = 0.01; % h = dx
m = (1 / h) - 1;

T = 10; % 0.01 10; final time

c = 1000; % 1 10 100; diffusive CFL number c = dt / (dx)^2
k = c * (h ^ 2); % k = dt
final_step = T / k;

L = (1 + 2 * c) * eye(m); % matrix on LHS
subdiag = -c * ones(m - 1, 1);
L = diag(subdiag, 1) + L + diag(subdiag, -1);

R = (1 - 2 * c) * eye(m); % matrix on RHS
subdiag = c * ones(m - 1, 1);
R = diag(subdiag, 1) + R + diag(subdiag, -1);

grid_x = h : h : (1 - h);
grid_x = grid_x(:);
f = @(x) 1 .* (x > 0.5) + zeros(size(x)); % u(x, 0)
u_init = f(grid_x);

w = u_init; % numerical solution

for i = 1 : final_step

    b = R * w;
    b(m) = b(m) + 2 * c; % result of the column matrix on RHS (u(1, t) = 1)

    w = L\b; % solve the linear system to obtain w at t = i * k

end

x = linspace(0, 1, 401);
f = @(x) 1 * (x > 0.5);
y = [f(x(1 : 200)), NaN,  f(x(202 : 401))];
plot(x, y, 'b-')
hold on

% u_approx = @(x) x - exp(-(2 * pi) ^ 2 * T) / pi * sin(2 * pi * x) + exp(-(4 * pi) ^ 2 *↙
T) / (2 * pi) * sin(4 * pi * x) - exp(-(6 * pi) ^ 2 * T) / (3 * pi) * sin(6 * pi * x) +↙
exp(-(8 * pi) ^ 2 * T) / (4 * pi) * sin(8 * pi * x);
% fplot(u_approx, [0 1]);
% hold on

u_steady = @(x) x;
fplot(u_steady, [0 1]);
hold on

plot(h : h : (1 - h), w, '.', 'Color', 'r')
xlabel('x')

lgd = legend('\it u(x, 0)', '\it u(x, \infty)', '\bf\itw\rm(\itt = 10)', 'Location',↙
```

```
'southeast');
title(lgd, ['CFL = ', num2str(c)])
legend('boxoff')
title('CN', 'fontsize', 16)
```