Numerical Methods
EN 530.766
HW 3

Consider the unsteady conduction problem

$$u_t = u_{xx} \text{ for } 0 \le x \le 2\pi \text{ given } u(0,t) = u(2\pi,t) = 0 \text{ and } u(x,0) = \sin(mx).$$

$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

$\delta = u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}$

(a) Derive the exact solution to this problem.

(b) Write computer programs to obtain the numerical solutions for the above equation using the:

(i) Forward Euler method,

(ii) Backward Euler method*, and

(iii) Crank-Nicolson method*

**Write your own* <u>*TDMA*</u> *solver and use it for the BE and CN method*s.

$t = 0.1, 0.5 \cdots \neq n\Delta t$

$r = \frac{\alpha \Delta t}{\Delta x^2}$

(c) For m=2,

(i) Obtain numerical solutions using a mesh with $\Delta x = 2\pi/20$ and r=1/3. Plot and compare the numerical and exact solutions for the three methods at time t=0.1, 0.5 and 1.0. Discuss your observations regarding this comparison.

choose closest pt. to 0.1/0.5

(ii) Based on your numerical solutions, show that the error and stability characteristics for each method behave as expected. Specifically,

a) does the spatial error scale as O(Δx^2) ?.

$e(\Delta x) = \sqrt{\sum_0^{2\pi} (u - u_0)^2}$

b) Is the stability predicted by VN stability analysis verified in your simulations?

$r < \frac{1}{2} \neq \epsilon$

c) What happens to the accuracy of the solution when you operate at r > 1/2 for the implicit schemes?

$r = 0.51$ a-b ...

You will have to conduct a grid refinement study and other "numerical experiments" to answer this question.

$\epsilon(\Delta x_i) = \cdots$

Δx	ϵ_{FE}	ϵ_{BE}	ϵ_{CN}

(d) Obtain a numerical solution of the above problem using a mesh with $\Delta x = 2\pi/20$ and r=0.5, and for m=3, 5, and 7.

(i) Make plots to compare the solution for the three methods with the exact solution and

(ii) interpret the results within the context of the amplification factor for these methods, i.e. show that the numerical results are consistent with what you expect from the amplification factor for these methods.

$G_k = \frac{|A_k^{n+1}|}{|A_k^n|}$ = Ex, FE, BE, CN

$u_t = u_{xx}$

$\epsilon_i^{n+1} - \epsilon_i^n = \alpha_2 (\epsilon_{i+1}^n - 2\epsilon_i^n + \epsilon_{i-1}^n)$

$\epsilon = A_k e^{j(kx - \alpha t)}$

NOTE:

1) Prepare your report using a word processor (MS Word, TeX, etc.). You can attach a hand-written derivation for the problem (a).

$\epsilon = A_k e^{j(\frac{2\pi k}{L} x - bt)}$

$\epsilon(x,t) = A_k e^{j(kx - bt)}$ $k = \frac{2\pi k}{L}$ $\beta = k\Delta x = m\Delta x$

2) Plots should only be as big as they need to be for clear viewing. Do not make full page plots because none of the plots you will make need to be that big. Plots should be numbered and have clear captions that describe the plot.

$\frac{\epsilon_i^{n+1}}{\epsilon_i^n} - 1 = \alpha_2 \left(\frac{\epsilon_{i+1}^n}{\epsilon_i^n} - 2 + \frac{\epsilon_{i-1}^n}{\epsilon_i^n} \right)$

3) <u>*Please*</u> *include the source code of your programs in the report.*

$u_t = u_{xx}$

$u = A e^{jkx - bt}$

$G - 1 = \alpha_2 \left(\frac{e^{j(kx_{i+1} - bt)}}{e^{j(kx_i - bt)}} - 2 + \frac{e^{jkx_{i-1}}}{e^{jkx_i}} \right)$

$G - 1 = \alpha_2^r \left(e^{jk\Delta x} - 2 + e^{-j(k\Delta x)} \right)$ β

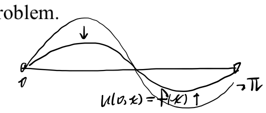
$|G| = |1 - 2r(1 - \cos(k\Delta x))| < 1$

Consider the unsteady conduction problem

$$u_t = u_{xx} \text{ for } 0 \le x \le 2\pi \text{ given } u(0,t) = u(2\pi,t) = 0 \text{ and } u(x,0) = \sin(mx).$$

(a) Derive the exact solution to this problem.

$\partial_t u = \partial_{xx} u$



Seperation variables:

$u = u(t,x) = A(t)B(x)$

$B(x)\frac{dA(t)}{dt} = A(t)\frac{d^2B(x)}{dx^2}$

$\frac{1}{A(t)}\frac{dA(t)}{dt} = \frac{1}{B(x)}\frac{d^2B(x)}{dx^2} = F(t) = G(x) = k$

$\begin{cases} \frac{dA(t)}{A(t)} = k\,dt \rightarrow A(t) = ce^{kt} \\ \frac{d^2B(x)}{B(x)dx^2} = k \end{cases}$

Try $B(x) = \sum_n C_n e^{nx}$

for each n: $(n)^2 = k$, $k = n^2$

$n = \sqrt{k}$ or $n = -\sqrt{k}$

$B(x) = C_k' e^{\sqrt{k}x} + C_{-k}' e^{-\sqrt{k}x}$

$u(t,x) = e^{kt}(C_k e^{\sqrt{k}x} + C_{-k}e^{-\sqrt{k}x}) = e^{kt}(\alpha\cos(\frac{\sqrt{k}}{i})x + \beta\sin(\frac{\sqrt{k}}{i})x)$

IC: $u(0,x) = \sin mx \rightarrow \alpha = 0,\ \frac{\sqrt{k}}{i} = m,\ \beta = 1$

$k = (im)^2 = -m^2$

$u(t,x) = e^{-m^2t}\sin mx$

BC: $u(t,0) = u(t,2\pi) = 0$ ✓

Exact Soln: $u(t,x) = u(x,t) = e^{-m^2t}\sin mx$

(b) Write computer programs to obtain the numerical solutions for the above equation using the:

(i) Forward Euler method,
(ii) Backward Euler method*, and
(iii) Crank-Nicolson method*

**Write your own* <u>*TDMA*</u> *solver and use it for the BE and CN methods.*

(1) Forward Euler: $\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}$

Iteration formula: $T_i^{n+1} = \frac{\Delta t}{\Delta x^2}(T_{i+1}^n + T_{i-1}^n) + (1 - \frac{2\Delta t}{\Delta x^2})T_i^n$

Backward Euler: $\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{\Delta x^2}$

Scheme: $(1 + \frac{2\Delta t}{\Delta x^2})T_i^{n+1} - \frac{\Delta t}{\Delta x^2}(T_{i-1}^{n+1} + T_{i+1}^{n+1}) = T_i^{(n)}$

let $\frac{\Delta t}{\Delta x^2} = r$, then $(1+2r)T_i^{n+1} - r(T_{i-1}^{n+1} + T_{i+1}^{n+1}) = T_i^n$

Scheme BC:
$\begin{cases} \text{for } n=0:\ (1+2r)T_i^1 - r(T_{i-1}^1 + T_{i+1}^1) = T_i^0 \\ \text{for } i=0:\ T_0 = 0 = T(t,0) \\ \text{for } i=i_{max}:\ T_{imax} = T(t,2\pi) = 0 \\ \text{for } i=1:\ (1+2r)T_1^{n+1} - r(\cancel{T_0^{n+1}} + T_2^{n+1}) = T_1^n \\ \text{for } i=i_{max}-1=im-1:\ (1+2r)T_{im-1}^{n+1} - r(T_{im-2}^{n+1} + \cancel{T_{im}^{n+1}}) = T_{im-1}^n \end{cases}$

Scheme:
$$\begin{bmatrix} 1+2r & -r & & 0 \\ -r & \ddots & \ddots & \\ & \ddots & \ddots & -r \\ 0 & & -r & 1+2r \end{bmatrix} \begin{bmatrix} T_1 \\ \uparrow \\ T_i \\ \downarrow \\ T_{im-1} \end{bmatrix}^{n+1} = \begin{bmatrix} T_1 \\ \uparrow \\ T_i \\ \downarrow \\ T_{im-1} \end{bmatrix}^n$$

$[A]\{T^{n+1}\} = \{T^n\}$

$\{T^{n+1}\} = [A]^{-1}\{T^n\}$

C-N Method: $\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{1}{2}\left(\frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{\Delta x^2} + \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}\right)$

$r = \frac{\Delta t}{\Delta x^2}$ $\quad 2(1+r)T_i^{n+1} - r(T_{i+1}^{n+1} + T_{i-1}^{n+1}) = 2(1-r)T_i^n + r(T_{i+1}^n + T_{i-1}^n)$

Scheme BC:
$\begin{cases} i=0:\ T_0 = 0 \\ i=1:\ 2(1+r)T_1^{n+1} - r(T_2^{n+1} + \cancel{T_0^{n+1}}) = 2(1-r)T_1^n + r(T_2^n + \cancel{T_0^n}) \\ i=i_{max}=im:\ T_{im} = 0 \\ i=i_{max}-1=im-1:\ 2(1+r)T_{im-1}^{n+1} - r(\cancel{T_{im}^{n+1}} + T_{im-2}^{n+1}) = 2(1-r)T_{im-1}^n + r(\cancel{T_{im}^n} + T_{im-2}^n) \end{cases}$

Scheme:
$$\begin{bmatrix} 2(1+r) & -r & & 0 \\ -r & \ddots & \ddots & \\ & \ddots & \ddots & -r \\ 0 & & -r & 2(1+r) \end{bmatrix} \begin{bmatrix} T_1 \\ \uparrow \\ T_i \\ \downarrow \\ T_{im-1} \end{bmatrix}^{n+1} = \begin{bmatrix} 2(1-r) & r & & 0 \\ r & \ddots & \ddots & \\ & \ddots & \ddots & r \\ 0 & & r & 2(1-r) \end{bmatrix} \begin{bmatrix} T_1 \\ \uparrow \\ T_i \\ \downarrow \\ T_{im-1} \end{bmatrix}^n$$

$[B]\{T^{n+1}\} = [C]\{T^n\}$

$\{T^{n+1}\} = [B]^{-1}[C]\{T^n\}$

ANSWER BOX

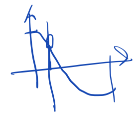
$-1 < 1 - 2r(1 - \cos k\Delta x) < 1$

$r < \frac{1}{1 - \cos k\Delta x}$

$-1 < 1 - 4r\sin^2\frac{\beta}{2} < 1$

$r > 0$

$r < \frac{1}{2}$



$r \le \frac{1}{2\sin^2\frac{\beta}{2}}$

$\beta \in [0, \pi]$

$r \le \frac{1}{2}$

Set T platform

input $\Delta x, \Delta t$

$\begin{cases} N_x = L_x \cdot \frac{1}{\Delta x} + 1 \\ N_t = L_t \cdot \frac{1}{\Delta t} + 1 \end{cases}$ $\quad x = (n_x - 1)\cdot\Delta x$, $t = (n_t - 1)\cdot\Delta t$

$T(t,x)$ create T[$N_t - 1$][$N_x - 1$]

BC:

A[$N_t - 2$][$N_x - 2$]

IC:

FE:

TDMA(a, b, c, d)

$$\begin{bmatrix} b & c & & \\ a & \ddots & \ddots & \\ & \ddots & \ddots & \end{bmatrix}_{n\times n} \begin{bmatrix} \\ \\ \end{bmatrix}_{n\times 1} = \begin{bmatrix} d \\ \\ \end{bmatrix}_{n\times 1}$$

a, b, c, d $\quad n\times 1$

c[$\quad$ x]

a[x $\quad$]

check TDMA: $x + y + z = 1$

$[\ \ ][x] = [b] \rightarrow x = A^{-1}b$

$\vec{u}^{n+1} = \text{TDMA}(\vec{u}^n)$

If (not of ... = ... of ...)

$\vec{u}^n = \vec{u}^{n+1}$

$$\begin{bmatrix} 1+2r & -r & & 0 \\ -r & \ddots & \ddots & \\ & \ddots & \ddots & -r \\ 0 & & -r & 1+2r \end{bmatrix}_{(m-1)\times(m-1)} \begin{bmatrix} T_1 \\ \uparrow \\ T_i \\ \downarrow \\ T_{m-1} \end{bmatrix}^{n+1} = \begin{bmatrix} T_1 \\ \uparrow \\ T_i \\ \downarrow \\ T_{m-1} \end{bmatrix}^{n}$$

TDMA: $\underbrace{\begin{bmatrix} a\ b\ c \end{bmatrix}}_{A} \begin{bmatrix} T_i^{n+1} \end{bmatrix} = \begin{bmatrix} T_i^n \end{bmatrix}$

Matrix Operation For $A:(m-1)\times(m-1)$

$$\begin{array}{c} ① \\ ② \\ ③ \\ \\ m-2 \\ m-1 \end{array} \begin{bmatrix} a_{11} & a_{12} & & & 0 \\ a_{21} & a_{22} & a_{23} & & \\ 0 & a_{32} & a_{33} & & \\ & & a & a_{(m-2)(m-2)} & a_{(m-2)(m-1)} \\ 0 & & & a_{(m-1)(m-2)} & a_{(m-1)(m-1)} \end{bmatrix} \xrightarrow{\text{operations}} \begin{array}{c} ① \\ ② \\ ③ \\ \\ m-2 \\ m-1 \end{array} \begin{bmatrix} 1 & a_{12} & & & 0 \\ 0 & 1 & a_{23} & & \\ & 0 & 1 & & \\ & & & 1 & a_{(m-2)(m-1)} \\ 0 & & & 0 & 1 \end{bmatrix} \xrightarrow{\text{operations}} 1$$

operations:

$① = \frac{①}{a_{11}}$

$② -= ① \cdot a_{21}$, $② = \frac{②}{a_{22}}$

$③ -= ② \cdot a_{32}$, $③ = \frac{③}{a_{33}}$

...

$k -= (k-1) \cdot a_{k(k-1)}$, $k = \frac{k}{a_{kk}}$

...

$(m-1) -= (m-2) \cdot a_{(m-1)(m-2)}$, $(m-1) = \frac{(m-1)}{a_{(m-1)(m-1)}}$

operations:

$(m-2) -= (m-1)\, a_{(m-2)(m-1)}$

$(m-3) -= (m-2)\, a_{(m-3)(m-2)}$

...

$k -= (k+1)\, a_{k(k+1)}$

...

$② -= ③\, a_{23}$

$① -= ②\, a_{12}$

$a_{kk} = b_k$

$a_{k(k-1)} = a_k$

$a_{k(k+1)} = C_k$ $\qquad T_k = T_k^n$

then for $H = [A, T_i^n]$ argumented matrix.

Algorithm: $① = \frac{①}{a_{11}}$ $\quad$ (n) means line n

$k = [2, m-1]$:

- $k = k - a_{k(k-1)}$
- $k = \frac{k}{a_{kk}}$

$k = [m-2, 1]$:

- $k = k - (k+1)\, a_{k(k+1)}$

$k = [1, m-1]$:

- $T_k^{n+1} = H_{km} T_k^n$

First line.

① $b_1 = \frac{b_1}{b_1}$ $\quad$ ① $C_1 = \frac{C_1}{b_1}$ $\quad$ $T_1 = \frac{T_1}{b_1}$

② $a_2 = a_2 - b_1 \cdot a_2$ $\quad$ ① $b_2 = b_2 - C_1 \cdot a_2$ $\quad$ $T_2 = T_2 - T_1 \cdot a_2$

② $b_2 = \frac{b_2}{b_2}$ $\quad$ ① $C_2 = \frac{C_2}{b_2}$ $\quad$ $T_2 = \frac{T_2^n}{b_2}$

$a_k = a_k - b_{k-1} a_k$, $b_k = b_k - C_{k-1} a_k$, $T_k = T_k - T_{k-1} \cdot a_k$

$b_k = \frac{b_k}{b_k}$, $C_k = \frac{C_k}{b_k}$, $T_k = \frac{T_k}{b_k}$

...

Last line. $a_{m-1} = a_{m-1} - b_{m-1} a_{m-1}$ $\quad$ $b_{m-1} = b_{m-1} - C_{m-2}\, a_{m-1}$

$b_{m-1} = \frac{b_{m-1}}{b_{m-1}}$, $\frac{T_{m-1}}{b_{m-1}}$

should get $a = (0)$

$b = (1)$

$k \in [m-2, 1]$ $\begin{cases} C_{m-2} = C_{m-2} - b_{m-1} \cdot C_{m-2} = 0 & T_{m-2} = T_{m-2} - \cancel{b_{m-1}}^{1} \cdot C_{m-2} \\ ② C_k = C_k - b_{k+1} \cdot C_k = 0 & ① T_k = T_k - C_k \end{cases}$

For BE:

$u_k^{n+1} = \text{TDMA\_BE}(a, b, c, D)$

$C_1 = \frac{C_1}{b_1}$ $\quad$ $\underset{D}{T_1} = \frac{T_1}{b_1}$ $\quad$ $\cancel{b_1 = 1}$

$k \in [2, m-1]$:

- $b_k = b_k - C_{k-1} a_k$ $\quad$ $T_k = T_k - T_{k-1} \cdot a_k$ $\quad$ $\cancel{a_k = 0}$
- $C_k = \frac{C_k}{b_k}$ $\quad$ $T_k = \frac{T_k}{b_k}$ $\quad$ $\cancel{b_k = 1}$

$k \in [m-2, 1]$:

- $T_k = T_k - C_k$ $\quad$ $\cancel{C_k = 0}$
- $T_k^{n+1} = T_k$

$D_k =$

For CN:

$u_k^{n+1} = \text{TDMA\_CN}(a, b, c, D, a', b', c')$

$D_1 = b_1' T_1 + C_1' T_2$

$k \in [2, m-2]$:

- $D_k = a_k' T_{k-1} + b_k' T_k + C_k' T_{k+1}$

$D_{m-1} = a_{m-1}' T_{m-2} + b_{m-1}' T_{m-1}$

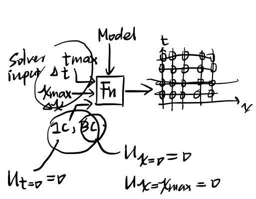
$C_1 = \frac{C_1}{b_1}$ $\quad$ $\underset{D}{T_1} = \frac{T_1}{b_1}$ $\quad$ $\cancel{b_1 = 1}$

$k \in [2, m-1]$:

- $b_k = b_k - C_{k-1} a_k$ $\quad$ $T_k = T_k - T_{k-1} \cdot a_k$ $\quad$ $\cancel{a_k = 0}$
- $C_k = \frac{C_k}{b_k}$ $\quad$ $T_k = \frac{T_k}{b_k}$ $\quad$ $\cancel{b_k = 1}$

$k \in [m-2, 1]$:

- $T_k = T_k - C_k$ $\quad$ $\cancel{C_k = 0}$
- $T_k^{n+1} = T_k$



For BE:

For CN:

Sys-setting

Solver input information
- t_{max} = 0.1/0.5/1.0
- x_{max} = 2π
- Δt =
- Δx = $\frac{2\pi}{\cancel{...}}$
- $L_x = x_{max}/\Delta x$
- $L_t = t_{max}/\Delta t$

Modeling information
- $m = 2$
- $r = \frac{1}{3}$
- $T[L_x+1][L_t+1] = 0$

BC: For $n_t \in [1, L_t]$:
- $T[n_t][0] = 0$ $T[n_t][L_x] = 0$

IC: For $n_x \in [0, L_x]$:
- $T[0][n_x] = \sin(m(n_x \cdot \Delta x))$

T = Solver(L_x, L_t, r, T)

$\Delta t = r\Delta x^2$

$\frac{1}{3} = \alpha \frac{\Delta t}{\Delta x^2}$ $\Delta t = \frac{\Delta x^2}{3} = \frac{1}{3}\left(\frac{2\pi}{20}\right)^2$

A[0] A[#A-1]

$A = [\emptyset \; --- \; \emptyset]$

Exact:

```
Te = solver_exact(Lx, Lt, m, Δt, Δx, T)
Te[Lt+1][Lx+1] = 0
For nt ∈ [0, Lt]:
    For nx ∈ [0, Lx]:
        Te[nt][nx] = exp(-m^2 nt Δt) · sin(m nx Δx)
return Te
```

Compare & ploting:

```
PlotTTe(T, Te,          ):
E = Te - T
x[Lx+1]
t[Lt+1]
x
```

Basic Solver: BE.

```
Solver_BE(Lx, Lt, r, T):
    a[Lx-1] = -r
    b[Lx-1] = 1+2r
    c[Lx-1] = -r
    For nt ∈ [1, Lt]
        D = T[nt-1]
        T[nt] = TDMA (a, b, c, D)
    return T
```

Basic Solver: CN:

```
Solver_CN (Lx, Lt, r, T):
    a[Lx-1] = -r
    b[Lx-1] = 2(1+r)
    c[Lx-1] = -r
    a*[Lx-1] = r
    b*[Lx-1] = 2(1-r)
    c*[Lx-1] = r
    T* = T
    For nt ∈ [1, Lt]:
        D = T[nt-1]
        D[1] = b*[1]·D[1] + c*[1]·D[2]
        For nx ∈ [2, Lx-2]:
            D[nx] = a*[nx]·D[nx-1] + b*[nx]·D[nx] + c*[nx]·D[nx+1]
        D[Lx-1] = a*[Lx-1]·D[Lx-2] + b*[Lx-1]·D[Lx-1]
        T[nt] = TDMA(a, b, c, D)
    return T
```

TDMA Algorithm

```
TDMA(a, b, c, D):
    c[1] = c[1]/b[1]
    D[1] = D[1]/b[1]
    For nx ∈ [2, Lx-1]:
        b[nx] -= c[nx-1] * a[nx]
        D[nx] -= D[nx-1] * a[nx]
        c[nx] = c[nx]/b[nx]
        D[nx] = D[nx]/b[nx]
    listnx[0] = 1
    For nx ∈ [2, Lx-2]
        Listnx + [nx]
    For nx ∈ reversed(Listnx):
        D[nx] -= c[nx] * D[nx+1]
    return D
```

nx

Solver_FE(

```
Iteration: For nt ∈ [0, Lt):
    For nx ∈ [0, Lx):
        T[n+1][i] = Δt/Δx² (T[n][i+1] + T[n][i-1]) + (1 - 2Δt/Δx²) T[n][i]
```

dx =

SEfe = ---

d. $SE[n_t] = E[n_t]^2$

$O(\Delta x^2, \Delta t)$.

(c) For m=2,

(i) Obtain numerical solutions using a mesh with $\Delta x = 2\pi/20$ and r=1/3. Plot and compare the numerical and exact solutions for the three methods at time t=0.1, 0.5 and 1.0. Discuss your observations regarding this comparison.

(ii) Based on your numerical solutions, show that the error and stability characteristics for each method behave as expected. Specifically,

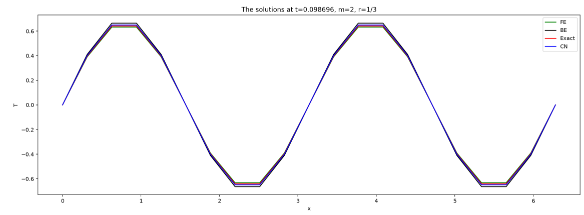
a) does the spatial error scale as O(Δx^2) ?.

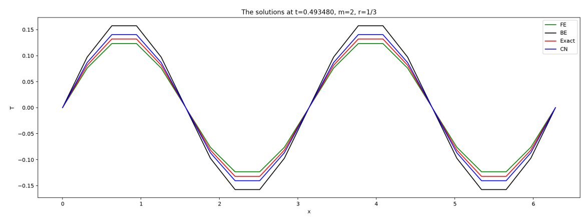
b) Is the stability predicted by VN stability analysis verified in your simulations?

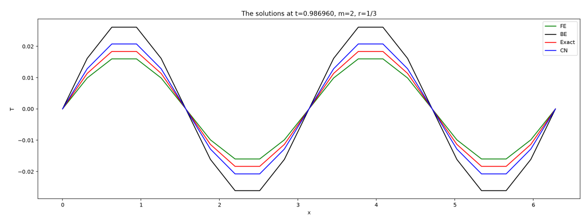
c) What happens to the accuracy of the solution when you operate at r > 1/2 for the implicit schemes?

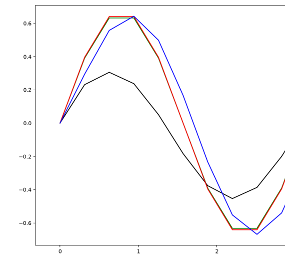
You will have to conduct a grid refinement study and other "numerical experiments" to answer this question.

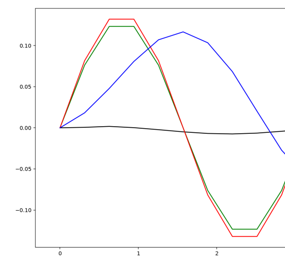
D

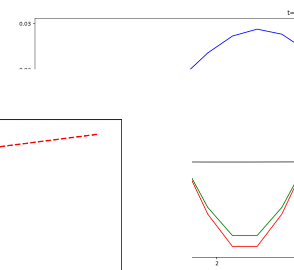












$\Delta x = \frac{C}{20}, \frac{C}{30}, \frac{C}{50}$

$N = [20, 30, 50 \ldots]$

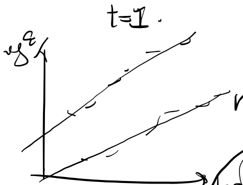
1 2 3

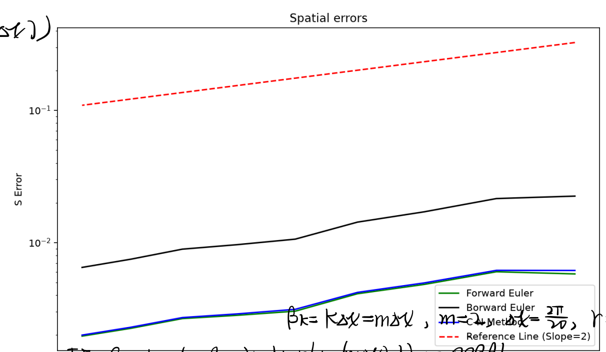
$N(i)$

$\epsilon = O(\Delta x^2)$

$\log(O(\Delta x^2)) = 2\log(O(\Delta x))$

t=T.





2) (a) $SE_{(\Delta x)} = \sqrt{\frac{\sum_{1}^{N_x}(u - u_{exact})^2}{N_x}}$

$SE(t)$

(b)

Von-Neuman Stability analysis: $r < \frac{1}{2}$

while $r = \frac{1}{3} < \frac{1}{2}$. within the line, so FE is stable.

All scheme work

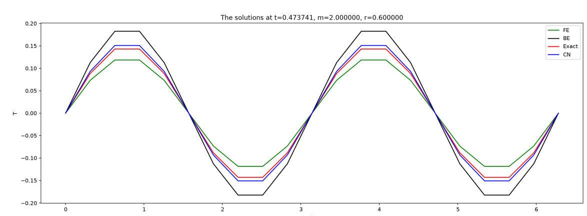
(c) If $r > \frac{1}{2}$, Ex. r = 0.6 FE NOT work

$\beta_k = k\Delta x = m\Delta x$, $m=2$, $\Delta x = \frac{2\pi}{20}$, $r = \frac{1}{2}$

FE: $G_k = 1 + 2r(\cos\beta_k - 1) = 1 + 2r(\cos(m\Delta x) - 1) = 0.99996$

BE: $G_k = \frac{1}{1 + 2r(1 - \cos\beta_k)} = 0.99993$

CN: $G_k = \frac{1 - r(1 - \cos\beta_k)}{1 + r(1 + \cos\beta_k)} = 0.499992$

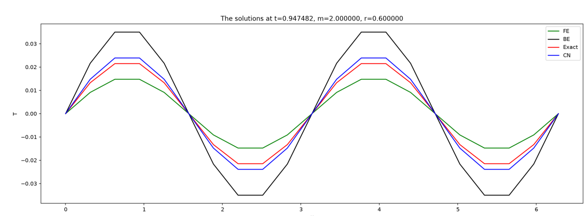


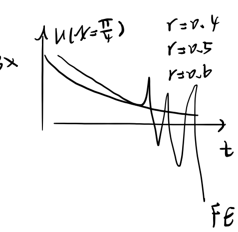
$\beta_k = k\Delta x = m\Delta x$, $m=2$, $\Delta x = \frac{2\pi}{20}$, $r = \frac{1}{3}$

FE: $G_k = 1 + 2r(\cos\beta_k - 1) = 1 + 2r(\cos(m\Delta x) - 1) = 0.99996$

BE: $G_k = \frac{1}{1 + 2r(1 - \cos\beta_k)} = 0.99996$

CN: $G_k = \frac{1 - r(1 - \cos\beta_k)}{1 + r(1 + \cos\beta_k)} = 0.599995$





(d) Obtain a numerical solution of the above problem using a mesh with $\Delta x = 2\pi/20$ and r=0.5, and for m=3, 5, and 7.

(i) Make plots to compare the solution for the three methods with the exact solution and

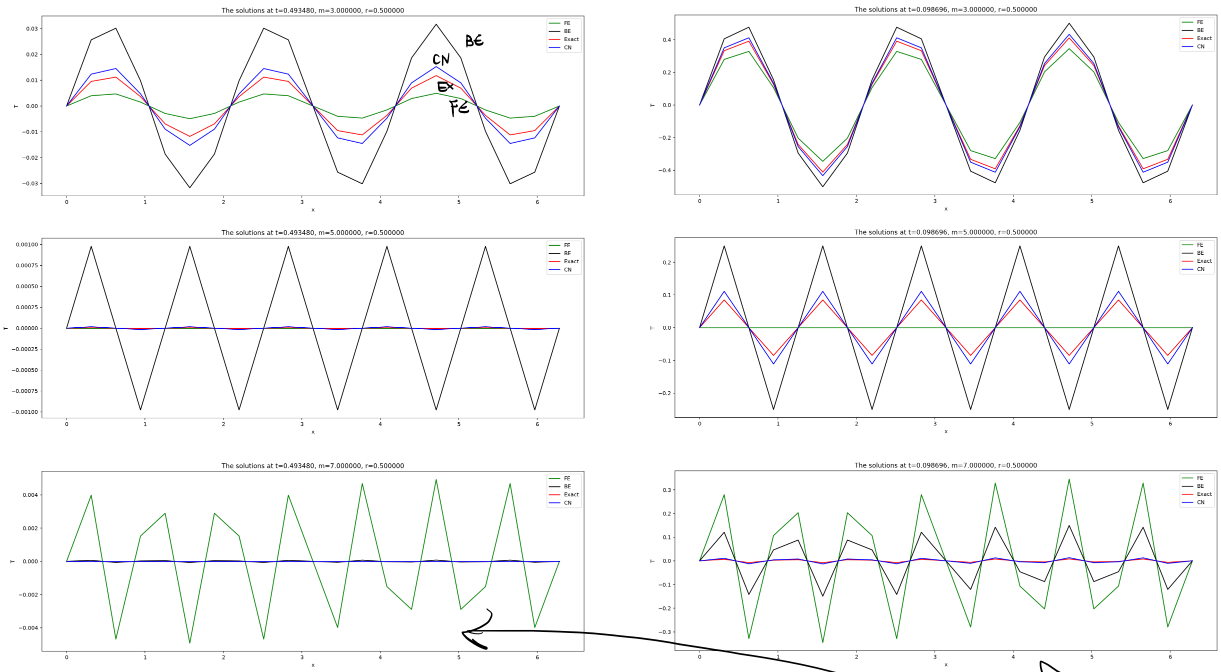
(ii) interpret the results within the context of the amplification factor for these methods, i.e. show that the numerical results are consistent with what you expect from the amplification factor for these methods.

t=0.5 t=0.1

m=3

m=5

m=7



$\beta = k\Delta x = \frac{k 2\pi}{L}\Delta x$ $k \neq K$
if $x \notin (0, \pi)$ $m \neq k$

Ex: $G_k^e = e^{-r\beta_k^2}$

$r = 0.5$

$\beta_k = m\Delta x = \frac{2\pi}{20} \times 3/5/7$.

FE. $\dfrac{u_k(t_n + \Delta t)}{u_k(t_n)} = G_k = 1 + 2r(\cos\beta_k - 1)$

$|G_k| < 1$

$-1 < 1 + 2r(\cos\beta - 1) < 1$

$r < \frac{1}{1 - \cos\beta}$

$r > \frac{1}{2(\cos\beta - 1)}$

BE. $G_k = [1 + 2r(1 - \cos\beta_k)]^{-1}$

CN. $G_k = \dfrac{1 - r(1 - \cos\beta_k)}{1 + r(1 - \cos\beta_k)}$

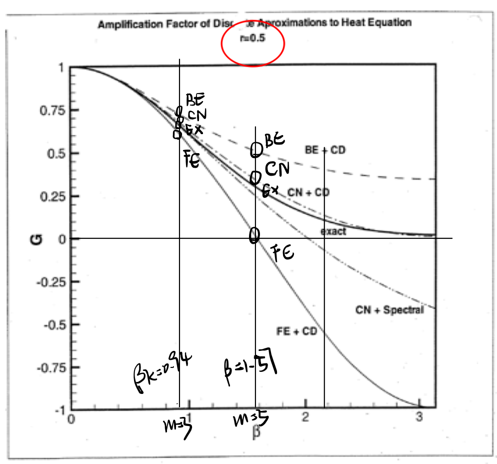
Method \ β_k, m	m=3	m=5	m=7
	0.94	1.57	2.19
Ex.			
FE			
BE			
CN.			

0.6413806259551538
0.5877852522924731
0.7081076030563466
0.658227155688125

0.29121293321402086
1.1102230246251565e-16
0.5
0.3333333333333333

0.08909490952726896
-0.5877852522924729
0.38643082887736446
0.11489393002107243

FE show un smooth

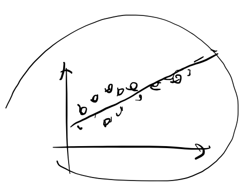


Could be seen at $m=3$ $\beta_k = \ldots$ $BE > CN > Ex > FE$, while CN closest to highest Amp.

Could be seen at m=3, Bk=0.94, showing in the table and the first line of the plot, Amplification Factor from Max to Min is BE>CN>EX>FE, where CN is most close to the Exact solution

Could be seen at m=5, Bk=1.57, showing in the table and the second line of the plot, Amplification Factor from Max to Min is still BE>CN>EX>FE, where CN is most close to the Exact solution, though the difference between numerical solutions and Exact solutions of Amplification Number is much larger than m=3. Also need to notice that in this case, Amplification Number of Forward Euler is pretty close to 0, where is also shown in the solution pictures before

Could be seen at m=7, Bk=2.19, showing in the table and the third line of the plot, Amplification Factor from Max to Min is still BE>CN>EX>FE, where CN is most close to the Exact solution, also need to notice at this case, FE Forward Euler method's Amp Number is smaller than 0



$\lg(?) = 0.5$

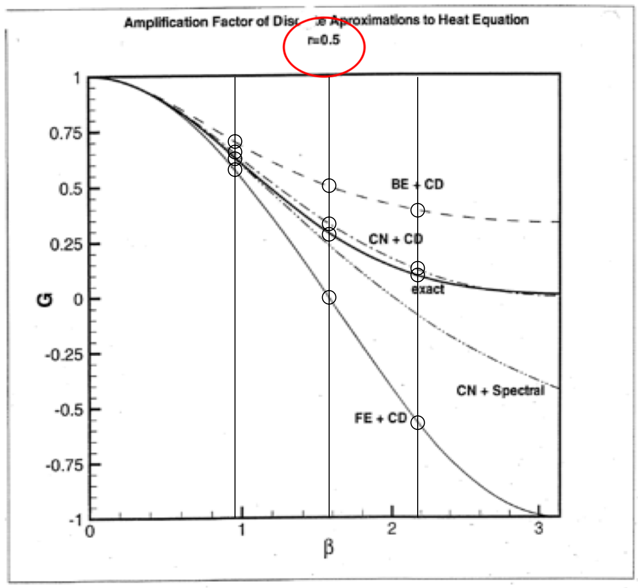
$(?) = \sqrt{e} \approx 7.18$

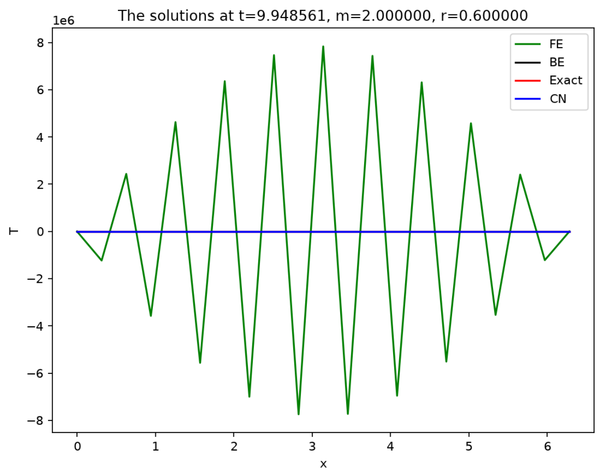
1.4

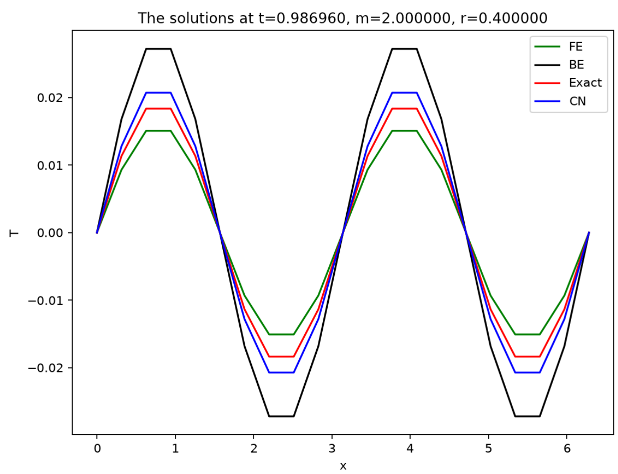
m=3
Bk=0.94

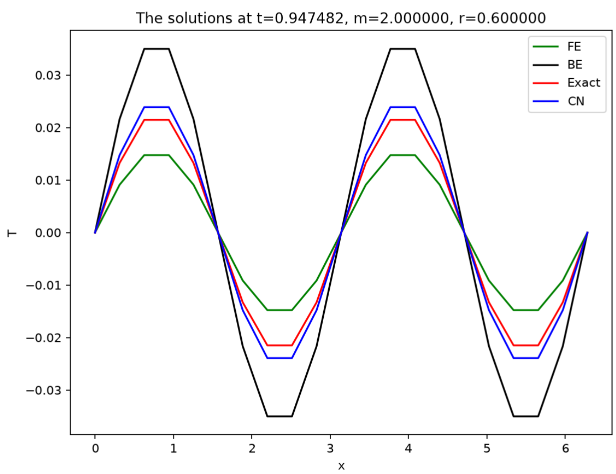
m=5
Bk=1.57

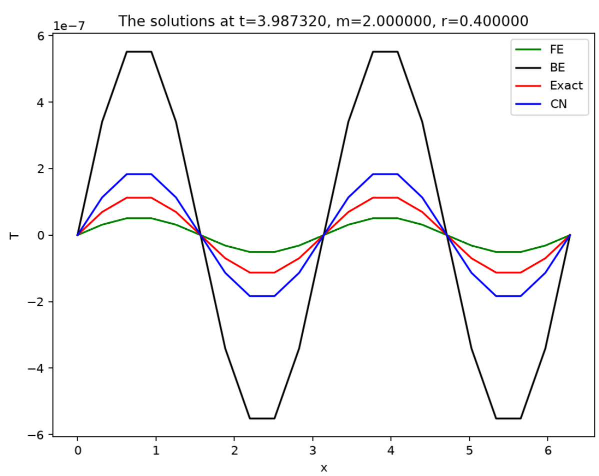
m=7
Bk=2.19

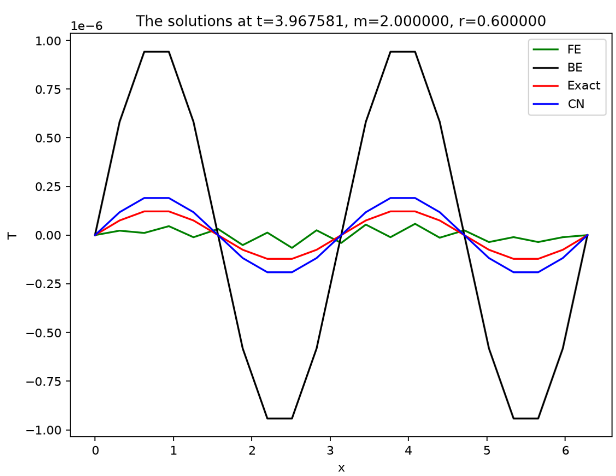



The solutions at t=9.948561, m=2.000000, r=0.600000
1e6
FE
BE
Exact
CN
T
x


The solutions at t=0.986960, m=2.000000, r=0.400000
FE
BE
Exact
CN
T
x


The solutions at t=0.947482, m=2.000000, r=0.600000
FE
BE
Exact
CN
T
x


The solutions at t=3.987320, m=2.000000, r=0.400000
1e−7
FE
BE
Exact
CN
T
x


The solutions at t=3.967581, m=2.000000, r=0.600000
1e−6
FE
BE
Exact
CN
T
x

For BE:

Sys-setting

Solver input information

```
tmax = 0.1/0.5/1.0 · 2π
xmax =
Δt =
Δx = 2π/...
Lx = xmax/Δx
Lt = tmax/Δt
```

Modeling information

```
m = 2
r = 1/3
T[Lx+1][Lt+1] = 0
```

```
BC: For nt ∈ [1, Lt]:
      └ { T[nt][0] = 0    T[nt][Lx] = 0
IC: For nx ∈ [0, Lx]:
      └ { T[0][nx] = sin(m(nx·Δx))
T = Solver(Lx, Lt, r, T)
```

$\frac{1}{3} = \alpha \frac{\Delta t}{\Delta x^2}$ $\quad \Delta t = \frac{\Delta x^2}{3} = \frac{1}{3}\left(\frac{2\pi}{20}\right)^2$

A[0] $\quad$ A[#$_A$-1]

A = [...]

For CN:

Exact:

```
Te = solver_exact(Lx, Lt, m, Δt, Δx, T)
Te[Lt+1][Lx+1] = 0
For nt ∈ [0, Lt]:
    For nx ∈ [0, Lx]:
        Te[nt][nx] = exp(-m²·nt·Δt) · sin(m·nx·Δx)
return Te
```

Compare & plotting:

```
PlotTTe(T, Te,        ):
E = Te - T
x[Lx+1]
t[Lt+1]
x[
```

Basic Solver: BE.

```
Solver_BE(Lx, Lt, r, T):
    a[Lx-1] = -r
    b[Lx-1] = 1+2r
    c[Lx-1] = -r
    For nt ∈ [1, Lt]
      └ T[nt] = TDMA(a, b, c, T)
    return T
```

Basic Solver: CN:

```
Solver_CN(Lx, Lt, r, T):
    a[Lx-1] = -r
    b[Lx-1] = 2(1+r)
    c[Lx-1] = -r
    a*[Lx-1] = r
    b*[Lx-1] = 2(1-r)
    c*[Lx-1] = r
    T* = T
    For nt ∈ [1, Lt]
      | T*[nt][1] = b*·T[nt][1] + c*·T[nt][2]
      | For nx ∈ [2, Lx-2]:
      |   └ T*[nt][nx] = a*[nx]·T*[nt][nx-1] + b*[nx]·T*[nt][nx] + c*[nx]·T*[nt][nx+1]
      └ T*[nt][Lx-1] = a*[Lx-1]·T*[Lx-2] + b*[Lx-1]·T*[Lx-1]
    For nt ∈ [1, Lt]
      └ T[nt] = TDMA_BE(a, b, c, T)
    return T
```

TDMA Algorithm

```
TDMA(a, b, c, T):
    D[Lx+1] = 0
    For nx ∈ [0, Lx]
      └ D[nx] = T[nt-1][nx]
    c[1] = c[1]/b[1]
    D[1] = D[1]/b[1]
    For nx ∈ [2, Lx-1]:
      | b[nx] -= c[nx-1] * a[nx]
      | T[nt-1][nx] -= c[nx-1] * a[nx]
      | c[nx] = c[nx]/b[nx]
      └ D[nx] = D[nx]/b[nx]
    listnx[0] = 1
    For nx ∈ [2, Lx-2]
      └ Listnx + [nx]
    For nx ∈ reversed(Listnx):
      | D[nx] -= c[nx]
      └ T[nt][nx] = D[nx]
    return T[nt]
```

nx

TDMA Algorithm

```
TDMA(a, b, c, T*):
    D[Lx+1] = 0
    For nx ∈ [0, Lx]
      └ D[nx] = T*[nt-1][nx]
    c[1] = c[1]/b[1]
    D[1] = D[1]/b[1]
    For nx ∈ [2, Lx-1]:
      | b[nx] -= c[nx-1] * a[nx]
      | T[nt-1][nx] -= c[nx-1] * a[nx]
      | c[nx] = c[nx]/b[nx]
      └ D[nx] = D[nx]/b[nx]
    listnx[0] = 1
    For nx ∈ [2, Lx-2]
      └ Listnx + [nx]
    For nx ∈ reversed(Listnx):
      | D[nx] -= c[nx]
      └ T[nt][nx] = D[nx]
    return T[nt]
```

nx

(c) For m=2,

(i) Obtain numerical solutions using a mesh with $\Delta x = 2\pi/20$ and r=1/3. Plot and compare the numerical and exact solutions for the three methods at time t=0.1, 0.5 and 1.0. Discuss your observations regarding this comparison.

(ii) Based on your numerical solutions, show that the error and stability characteristics for each method behave as expected. Specifically,

a) does the spatial error scale as $O(\Delta x^2)$?.

b) Is the stability predicted by VN stability analysis verified in your simulations?

c) What happens to the accuracy of the solution when you operate at r > 1/2 for the implicit schemes?

You will have to conduct a grid refinement study and other "numerical experiments" to answer this question.

$u_j^{n+1}(1+2r) - ru_{j+1}^{n+1} - ru_{j-1}^{n+1} = u_j^n$ → Backward Euler

$u_j^{n+1}(1+2r) - r'u_{j+1}^{n+1} - r'u_{j-1}^{n+1} = u_j^n + r'(-2u_j^n + u_{j-1}^n + u_{j+1}^n)$

T [timestep n+1] = CN (T_{n-1})

k = 2 : length(T_{n-1}) − 1 $T_{n-1} \to T_1$

$d = T_1(j) + r(T_{1,j+1} + T_{1,j-1} - 2T_1(j))$

[T2] = TDMA (a, b, c, d)

$[T_2]$ = TDMA (a, b, c, T_1)

TDMA (a, b, c, ~~TDMA(T, j)~~)

$(1+2r)T_i^{n+1} - r(T_{i-1}^{n+1} + T_{i+1}^{n+1}) = T_i^n$

$aT_{i-1}^{n+1} + bT_i^{n+1} + cT_{i+1}^{n+1} = T_i^n$

$$\begin{bmatrix} 1+2r & -r & \\ -r & 1+2r & -r \end{bmatrix}$$

$$\begin{bmatrix} 1+2r & -r & 0 \\ -r & 1+2r & -r \\ 0 & -r & 1+2r \end{bmatrix}$$

② − = ①/a_{11} a_{21} ...

③ − = ②/a_{22} a_{32} ...

④ − = ③/a_{33} a_{43} ...

$[\quad]_{n\times n}$

$$\begin{bmatrix} b & c & 0 \\ a & b & c \\ 0 & a & b \end{bmatrix} \sim \begin{bmatrix} b & c & 0 \\ 0 & b-\frac{ac}{b} & c \\ 0 & a & b \end{bmatrix} \sim \begin{bmatrix} b & c & 0 \\ 0 & b-\frac{ac}{b} \end{bmatrix}$$

$(1+2r)T_i^{n+1} - r(T_{i-1}^{n+1} + T_{i+1}^{n+1}) = T_i^n$

(d) Obtain a numerical solution of the above problem using a mesh with $\Delta x = 2\pi/20$ and r=0.5, and for m=3, 5, and 7.

(i) Make plots to compare the solution for the three methods with the exact solution and

(ii) interpret the results within the context of the amplification factor for these methods, i.e. show that the numerical results are consistent with what you expect from the amplification factor for these methods.

$-ru_{j-1} + (1+2r)u_j + -ru_{j-1} = u_j^n$

j = 1

$-ru_2 + (1+2r)u_1 - ru_0 = u_1^n$

$-ru_2 + (1+2r)u_1 = u_1^n + ru_0$

$a_1u_2 + b_1u_1 = d_1 + c_1u_0 \Rightarrow u_1 = \frac{(d_1 + c_1u_0) - a_1u_2}{b_1}$

j = 2

$a_2u_3 + b_2u_2 + c_2u_1 = d_2$

$a_2u_3 + b_2u_2 + c_2\left(\frac{d_1 + c_1u_0 - a_1u_2}{b_1}\right) = d_2$

$u_2 = d_2 - \frac{c_2d_1}{b_1} - c_1c_2u_0$

$T_j^{n+1} \to c_f T_{j+1}^{n+1} + c_g$

T → [0 ... k]

j = k−1

$T_{k-1}^{n+1} = c_f T_k^{n+1} + c_g$