Crank Nicolson Solution to the Heat Equation

ME 448/548 Notes

Gerald Recktenwald
Portland State University
Department of Mechanical Engineering
gerry@pdx.edu

Overview

1. Use finite approximations to $\partial u/\partial t$ and $\partial^2 u/\partial x^2$: same components used in FTCS and BTCS.
2. Average contribution from t_k and t_{k+1} in the approximation of $\partial^2 u/\partial x^2$.
3. Computational formula is (still) *implicit*: all u_i^{k+1} must be solved simultaneously. Information from u_{i-1}^k, u_i^k, and u_{i+1}^k are used at each time step in the computation of u_i^{k+1}.
4. Solution is not significantly more complex than BTCS.
5. Like BTCS, the Crank-Nicolson methods is unconditionally stable for the heat equation.
6. Truncation error is $\mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2)$. Time accuracy is better than BTCS or FTCS.

The θ Method

Evaluate the diffusion operator $\partial^2 u/\partial x^2$ at both time steps t_{k+1} and time step t_k, and use a weighted average

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \alpha\theta \left[\frac{u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}}{\Delta x^2}\right] + \alpha(1-\theta)\left[\frac{u_{i-1}^{k} - 2u_i^{k} + u_{i+1}^{k}}{\Delta x^2}\right] \quad (1)$$

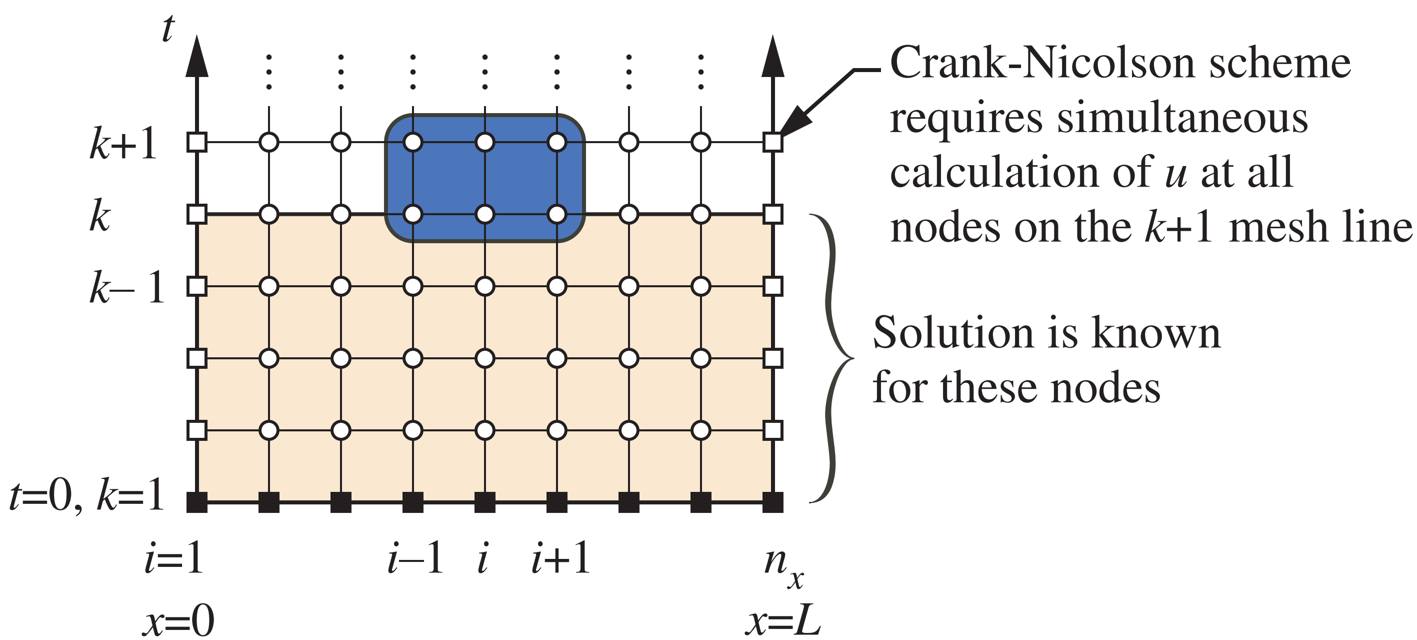
where

$$0 \le \theta \le 1$$

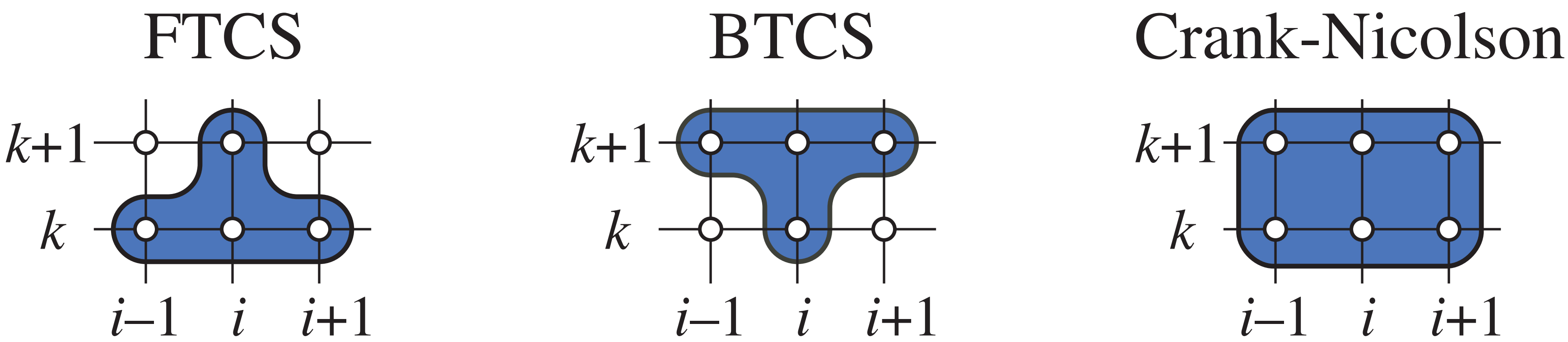
$$\theta = 0 \quad \Longleftrightarrow \quad \text{FTCS}$$

$$\theta = 1 \quad \Longleftrightarrow \quad \text{BTCS}$$

Crank-Nicolson Computational Molecule



Compare Computational Molecules



Crank Nicolson Approximation to the Heat Equation

$$
\begin{aligned}
-\frac{\alpha}{2\Delta x^2}u_{i-1}^{k+1} + \left(\frac{1}{\Delta t} + \frac{\alpha}{\Delta x^2}\right)u_i^{k+1} - \frac{\alpha}{2\Delta x^2}u_{i+1}^{k+1} = \\
\frac{\alpha}{2\Delta x^2}u_{i-1}^{k} + \left(\frac{1}{\Delta t} - \frac{\alpha}{\Delta x^2}\right)u_i^{k} + \frac{\alpha}{2\Delta x^2}u_{i+1}^{k} \quad (2)
\end{aligned}
$$

Crank-Nicolson System of Equations

The system of equations has the same structure as BTCS

$$
\begin{bmatrix}
a_1 & b_1 & 0 & 0 & 0 & 0 \\
c_2 & a_2 & b_2 & 0 & 0 & 0 \\
0 & c_3 & a_3 & b_3 & 0 & 0 \\
0 & 0 & \ddots & \ddots & \ddots & 0 \\
0 & 0 & 0 & c_{nx-1} & a_{nx-1} & b_{nx-1} \\
0 & 0 & 0 & 0 & c_{nx} & a_{nx}
\end{bmatrix}
\begin{bmatrix}
u_1^{k+1} \\ u_2^{k+1} \\ u_3^{k+1} \\ \vdots \\ u_{nx-1}^{k+1} \\ u_{nx}^{k+1}
\end{bmatrix}
=
\begin{bmatrix}
d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{nx-1} \\ d_{nx}
\end{bmatrix}
\tag{3}
$$

where the coefficients of the interior nodes $(i = 2, 3, \ldots, N-1)$ are

$$
\begin{aligned}
a_i &= 1/\Delta t + \alpha/\Delta x^2 = 1/\Delta t - (b_i + c_i), \\
b_i &= c_i = -\alpha/(2\Delta x^2), \\
d_i &= -c_i u_{i-1}^k + (1/\Delta t + b_i + c_i)\, u_i^k - b_i u_{i+1}^k.
\end{aligned}
$$

`demoCN` Code

```
% --- Coefficients of the tridiagonal system
b = (-alfa/2/dx^2)*ones(nx,1);   %  Super diagonal: coefficients of u(i+1)
c = b;                           %  Subdiagonal:    coefficients of u(i-1)
a = (1/dt)*ones(nx,1) - (b+c);   %  Main Diagonal:  coefficients of u(i)
at = (1/dt + b + c);             %  Coefficient of u_i^k on RHS
a(1) = 1;     b(1) = 0;          %  Fix coefficients of boundary nodes
a(end) = 1;   c(end) = 0;
[e,f] = tridiagLU(a,b,c);        %  Save LU factorization

% --- Assign IC and save BC values in ub.  IC creates u vector
x = linspace(0,L,nx)';   u = sin(pi*x/L);   ub = [0 0];

% --- Loop over time steps
for k=2:nt
  % --- Update RHS for all equations, including those on boundary
  d =  - [0;       c(2:end-1).*u(1:end-2); 0] ...
       + [ub(1);  at(2:end-1).*u(2:end-1); ub(2)] ...
       - [0;       b(2:end-1).*u(3:end);   0];
  u = tridiagLUsolve(e,f,b,d);                   %  Solve the system
end
```

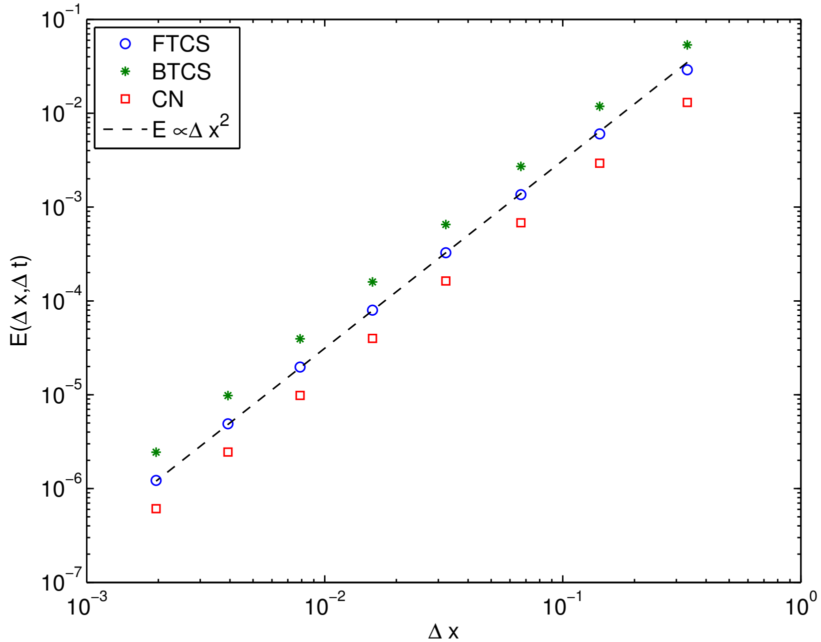
Convergence of FTCS, BTCS and CN

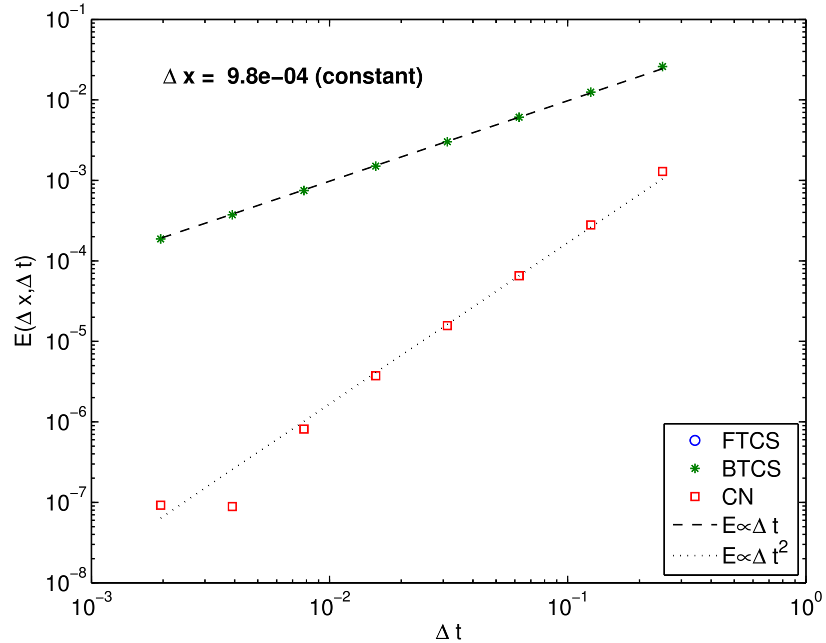
Truncation Errors

Scheme	Spatial	Temporal
FTCS	Δx^2	Δt
BTCS	Δx^2	Δt
C-N	Δx^2	Δt^2

```
Reduce both dx and dt within the FTCS stability limit

                      -------------  Errors  ------------
  nx        nt          FTCS        BTCS          CN
   4         5      2.903e-02    5.346e-02    1.304e-02
   8        21      6.028e-03    1.186e-02    2.929e-03
  16        92      1.356e-03    2.716e-03    6.804e-04
  32       386      3.262e-04    6.522e-04    1.630e-04
  64      1589      7.972e-05    1.594e-04    3.984e-05
 128      6453      1.970e-05    3.939e-05    9.847e-06
 256     26012      4.895e-06    9.790e-06    2.448e-06
 512    104452      1.220e-06    2.440e-06    6.101e-07

Reduce dt while holding dx = 9.775171e-04 (L=1.0, nx=1024) constant

                      -------------  Errors  ------------
  nx        nt          FTCS        BTCS          CN
1024         8            NaN    2.601e-02    1.291e-03
1024        16            NaN    1.246e-02    2.798e-04
1024        32            NaN    6.102e-03    6.534e-05
1024        64            NaN    3.020e-03    1.570e-05
1024       128            NaN    1.502e-03    3.749e-06
1024       256            NaN    7.492e-04    8.154e-07
1024       512            NaN    3.742e-04    8.868e-08
1024      1024            NaN    1.871e-04    9.218e-08
```

Convergence of FTCS, BTCS and CN





In the plot of truncation error versus Δt (right hand plot), there is an irregularity at $\Delta t \sim 3.9 \times 10^{-3}$. At that level of Δt, and for the chosen Δx (which is constant), the truncation error due to Δx is no longer negligible. Further reductions in Δt alone will not reduce the total truncation error.

Summary for the Crank-Nicolson Scheme

- The Crank-Nicolson method is more accurate than FTCS or BTCS.
  Although all three methods have the same spatial truncation error (Δx^2), the better *temporal* truncation error for the Crank-Nicolson method is big advantage.
- Like BTCS, the Crank-Nicolson scheme is unconditionally stable for the heat equation.
- Like BTCS, a system of equations for the unknown u_i^k must be solved at each time step. The tridiagonal solver for the 1D heat equation obtains an efficient solution of the system of equations.
- The Crank-Nicolson scheme is recommended over FTCS and BTCS.