1. Combine finite difference approximations for $\partial u/\partial t$ at $x = x_i$

$$\left.\frac{\partial u}{\partial t}\right|_{t_k, x_i} = \frac{u_i^{k+1} - u_i^k}{\Delta t} + \mathcal{O}(\Delta t). \tag{1}$$

   and $\partial^2 u/\partial x^2$ at time t_k

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{t_k, x_i} = \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x^2} + \mathcal{O}(\Delta x^2). \tag{2}$$

   To get

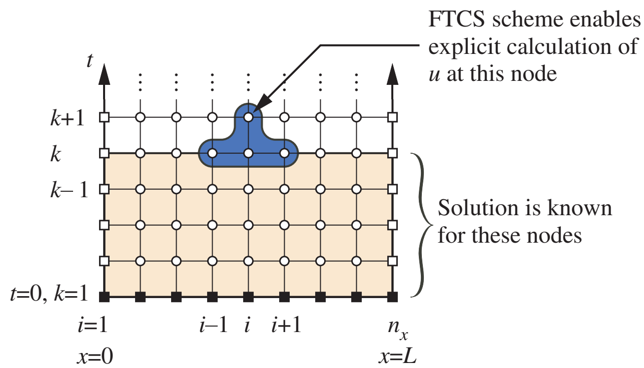
$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \alpha \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x^2} + \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2) \tag{3}$$

   Drop the truncation error terms and solve for u_i^{k+1}

$$u_i^{k+1} = r u_{i+1}^k + (1 - 2r) u_i^k + r u_{i-1}^k \tag{4}$$

   where $r = \alpha \Delta t/\Delta x^2$. Equation (4) is the computational formula for the FTCS scheme. It is an *explicit* scheme because it provides a simple formula to update u_i^{k+1} independently of the other nodal values at t_{k+1}.

2. Computational Molecule



3. Matlab implementation: code from `demoFTCS`

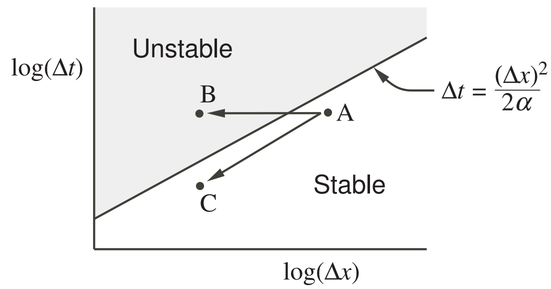
```
% --- Assign IC and BC.  u is initialized to a vector that includes BC
x = linspace(0,L,nx)';    u = sin(pi*x/L);

% --- Loop over time steps
for k=2:nt
  uold = u;    % prepare for next step
  for i=2:nx-1
    u(i) = r*uold(i-1) + r2*uold(i) + r*uold(i+1);
  end
end
```

   A more general implementation is in `heatFTCS`.

4. Stability: A computational scheme is *stable* if small perturbations in the values of the dependent variable do not grow unboundedly. The perturbations may arise in the initial conditions, boundary conditions, or from roundoff.

   The FTCS is *conditionally stable* for the heat equation when
$$r = \frac{\alpha \Delta t}{\Delta x^2} < 1/2$$



5. Measuring truncation error: When an analytical solution is known, we can compare the numerical solution (in this case from FTCS) with the exact solution.

   Define:
$$E(n_x, n_t) = \frac{1}{\sqrt{n_x}} \| u_i^k - u(x_i, t_k) \|_2 \tag{5}$$
   where n_x is the number of x-direction nodes in the mesh.

   The *local error* at $x = x_i$ and $t = t_k$ is
$$e_i^k = u_i^k - u(x_i, t_k). \tag{6}$$
   Let $\bar{e}_k$ as an *RMS average error per node* at time step t_k
$$\bar{e}^k \equiv \left[\frac{1}{n_x} \sum_{i=1}^{n_x} (e_i^k)^2 \right]^{1/2} \tag{7}$$
   This definition of the error is consistent with the *order of accuracy of the solution.* A little algebra shows that $E(n_x, n_t) = \bar{e}^k$.

   We are most interested in the *rate* at which $\bar{e}$ or $E(n_x, n_t)$ approach zero. Applying FTCS to the heat equation gives
$$E(n_x, n_t) = \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2)$$

```
>> convFTCS

    nx      nt       error      E(j)/E(j-1)     p
     8      21     6.028e-03        NaN       0.0000
    16      92     1.356e-03      0.2249      2.1524
    32     386     3.262e-04      0.2406      2.0553
    64    1589     7.972e-05      0.2444      2.0329
   128    6453     1.970e-05      0.2471      2.0170
   256   26012     4.895e-06      0.2485      2.0085
```