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

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

and $\frac{\partial^2 u}{\partial x^2}$ at time $t_k$

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

to get

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

Drop the truncation error terms, shift the $k$ subscripts by one, and move all $u_{k+1}^i$ terms to the left hand side

$$\left[ -\frac{\alpha}{\Delta x^2} \right] u_{i-1}^{k+1} + \left[ \frac{1}{\Delta t} + \frac{2\alpha}{\Delta x^2} \right] u_i^{k+1} + \left[ -\frac{\alpha}{\Delta x^2} \right] u_{i+1}^{k+1} = \frac{1}{\Delta t} u_i^k \quad (4)$$

Equation (4) is the computational formula for the BTCS scheme. It is an implicit scheme because all $u_{k+1}^i$ values are coupled and must be updated simultaneously.

2. Computational Molecule

![Diagram of computational molecule]

3. The BTCS method is unconditionally stable for the heat equation.

The benefit of stability comes at a cost of increased complexity of solving a linear system of equations at each time step.
4. For the one-dimensional heat equation, the linear system of equations for the BTCS method can be organized into a tridiagonal matrix.

\[
\begin{bmatrix}
  a_1 & b_1 & 0 & 0 & 0 \\
  c_2 & a_2 & b_2 & 0 & 0 \\
  0 & c_3 & a_3 & b_3 & 0 \\
  0 & 0 & \ddots & \ddots & \ddots \\
  0 & 0 & 0 & c_{n_x-1} & a_{n_x-1} & b_{n_x-1} \\
  0 & 0 & 0 & 0 & c_{n_x} & a_{n_x} \\
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  \vdots \\
  u_{n_x-1} \\
  u_{n_x} \\
\end{bmatrix}
= 
\begin{bmatrix}
  d_1 \\
  d_2 \\
  d_3 \\
  \vdots \\
  d_{n_x-1} \\
  d_{n_x} \\
\end{bmatrix}
\]

where the coefficients of the matrix are stored compactly as vectors

\[
a_i = \left(\frac{1}{\Delta t}\right) + \left(\frac{2\alpha}{\Delta x^2}\right), \quad b_i = c_i = -\frac{\alpha}{\Delta x^2}, \quad d_i = \left(\frac{1}{\Delta t}\right)u_k.
\]

This system of equations is efficiently solved with a form of LU factorization. The LU factors need to be computed only once before the first time step.

5. MATLAB implementation: code from demoBTCS

```matlab
% --- Coefficients of the tridiagonal system
b = (-alfa/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)
a(1) = 1; b(1) = 0; % Fix coefficients of boundary nodes
c(end) = 0; % Fix coefficients of boundary nodes
[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
    d = [ub(1); u(2:nx-1)/dt; ub(2)]; % Update RHS, preserve BC
    u = tridiagLUsolve(e,f,b,d); % Solve the system
end
```

A more general implementation is in heatBTCS.

6. The demoBTCS function demonstrates the correct behavior of truncation error as \(\Delta x\) and \(\Delta t\) are reduced.

```matlab
>> convBTCS
```

<table>
<thead>
<tr>
<th>nx</th>
<th>nt</th>
<th>error</th>
<th>E(j)/E(j-1)</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>5.346e-02</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>1.186e-02</td>
<td>0.2219</td>
<td>2.1723</td>
</tr>
<tr>
<td>16</td>
<td>92</td>
<td>2.716e-03</td>
<td>0.2290</td>
<td>2.1268</td>
</tr>
<tr>
<td>32</td>
<td>386</td>
<td>6.522e-03</td>
<td>0.2401</td>
<td>2.0581</td>
</tr>
<tr>
<td>64</td>
<td>1589</td>
<td>1.594e-04</td>
<td>0.2444</td>
<td>2.0326</td>
</tr>
<tr>
<td>128</td>
<td>6453</td>
<td>3.939e-05</td>
<td>0.2471</td>
<td>2.0168</td>
</tr>
<tr>
<td>256</td>
<td>26012</td>
<td>9.790e-06</td>
<td>0.2485</td>
<td>2.0084</td>
</tr>
<tr>
<td>512</td>
<td>104452</td>
<td>2.440e-06</td>
<td>0.2493</td>
<td>2.0042</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>nx</th>
<th>nt</th>
<th>error</th>
<th>E(j)/E(j-1)</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>6.444e-02</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>2.737e-02</td>
<td>0.4248</td>
<td>1.2383</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>1.276e-02</td>
<td>0.4661</td>
<td>1.1014</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>6.172e-03</td>
<td>0.4833</td>
<td>1.0475</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>3.037e-03</td>
<td>0.4921</td>
<td>1.0230</td>
</tr>
<tr>
<td>128</td>
<td>128</td>
<td>1.507e-03</td>
<td>0.4961</td>
<td>1.0113</td>
</tr>
<tr>
<td>256</td>
<td>256</td>
<td>7.504e-04</td>
<td>0.4981</td>
<td>1.0056</td>
</tr>
<tr>
<td>512</td>
<td>512</td>
<td>3.745e-04</td>
<td>0.4990</td>
<td>1.0028</td>
</tr>
</tbody>
</table>