```cpp
//----------------------------------------------------------------------------
// Figure 3.7 Page 60
float Exp_FD_Euro_Call(float K, float T, float S, float sig, float r, float div,
                       int N, int Nj, float dx)
{
  int i, j;
  float dt, nu, edx, pu, pm, pd;
  float St[51], C[51][51];

  dt = T/N;
  nu = r-div-0.5*sig*sig;
  edx = exp(dx);

  pu = 0.5*dt*(sig*sig/(dx*dx)+nu/dx);
  pm = 1.0-dt*(sig*sig/(dx*dx))-r*dt;
  pd = 0.5*dt*(sig*sig/(dx*dx)-nu/dx);

  St[Id(-Nj)] = S*exp(-Nj*dx);
  for (j=-Nj+1;j<=Nj;j++)
  {
    St[Id(j)] = St[Id(j-1)]*edx;
  }

  for (j=-Nj;j<=Nj;j++)
  {
    C[Id(N)][Id(j)] = max(0, St[Id(j)]-K);
  }

  for (i=N-1;i>=0;i--)
  {
    for (j=-Nj+1;j<=Nj-1;j++)
    {
      C[Id(i)][Id(j)] = pu*C[Id(i+1)][Id(j+1)]+pm*C[Id(i+1)][Id(j)]+
                        pd*C[Id(i+1)][Id(j-1)];
    }
    C[Id(i)][Id(-Nj)] = C[Id(i)][Id(-Nj+1)];
    C[Id(i)][Id(Nj)] = C[Id(i)][Id(Nj-1)]+St[Id(Nj)]-St[Id(Nj-1)];
  }
  return( C[Id(0)][Id(0)] );
}
```

```cpp
//----------------------------------------------------------------------------
// Figure 3.9 Page 62
float Exp_FD_Amer_Put(float K, float T, float S, float sig, float r, float div,
                       int N, int Nj, float dx)
{
  int i, j;
  float dt, nu, edx, pu, pm, pd;
  float St[51], C[51][51];

  dt = T/N;
  nu = r-div-0.5*sig*sig;
  edx = exp(dx);

  pu = 0.5*dt*(sig*sig/(dx*dx)+nu/dx);
  pm = 1.0-dt*(sig*sig/(dx*dx))-r*dt;
  pd = 0.5*dt*(sig*sig/(dx*dx)-nu/dx);

  St[Id(-Nj)] = S*exp(-Nj*dx);
  for (j=-Nj+1;j<=Nj;j++)
  {
    St[Id(j)] = St[Id(j-1)]*edx;
  }

  for (j=-Nj;j<=Nj;j++)
  {
    C[Id(N)][Id(j)] = max(0, K-St[Id(j)]);
  }

  for (i=N-1;i>=0;i--)
  {

    C[Id(i)][Id(Nj)] = C[Id(i)][Id(Nj-1)]+St[Id(Nj)]-St[Id(Nj-1)];
    C[Id(i)][Id(-Nj)] = C[Id(i)][Id(-Nj+1)];

    for (j=-Nj+1;j<=Nj-1;j++)
    {
      C[Id(i)][Id(j)] = pu*C[Id(i+1)][Id(j+1)]+pm*C[Id(i+1)][Id(j)]+
                        pd*C[Id(i+1)][Id(j-1)];
    }

    for (j=-Nj;j<=Nj;j++)
    {
      C[Id(i)][Id(j)] = max( C[Id(i)][Id(j)], K-St[Id(j)] );
    }
  }
  return( C[Id(0)][Id(0)] );
}
```

```
//---------------------------------------------------------------------------
// Figure 3.13 Page 69
float Imp_FD_Amer_Put(float K, float T, float S, float sig, float r, float div,
                       int N, int Nj, float dx)
{
  int i, j;
  float dt, nu, edx, pu, pm, pd;
  float lamda_L, lamda_U;
  float St[51], C[51][51];

  void solve_implicit_tridiagonal_system(float C[51][51], float pu, float pm,
              float pd, float lamda_L, float lamda_U, int Nj );

  dt = T/N;
  nu = r-div-0.5*sig*sig;
  edx = exp(dx);

  pu = -0.5*dt*(sig*sig/(dx*dx)+nu/dx);
  pm = 1.0+dt*(sig*sig/(dx*dx))+r*dt;
  pd = -0.5*dt*(sig*sig/(dx*dx)-nu/dx);

  St[Id(-Nj)] = S*exp(-Nj*dx);
  for (j=-Nj+1;j<=Nj;j++)
  {
    St[Id(j)] = St[Id(j-1)]*edx;
  }

  for (j=-Nj;j<=Nj;j++)
  {
    C[Id(N)][Id(j)] = max(0, K-St[Id(j)]);
  }

  lamda_L = -1*( St[-Nj+1] - St[-Nj] );
  lamda_U = 0.0;

  solve_implicit_tridiagonal_system( C, pu, pm, pd, lamda_L, lamda_U, Nj );

  for (i=N-1;i>=0;i--)
  {
    for (j=-Nj;j<=Nj;j++)
    {
      C[Id(i)][Id(j)] = max( C[Id(i)][Id(j)], K-St[Id(j)] );
    }
  }
  return( C[Id(0)][Id(0)] );
}
```

```
void solve_implicit_tridiagonal_system(float C[51][51], float pu, float pm, float pd,
                                       float lamda_L, float lamda_U, int Nj )
{
  int j;
  float pmp[51], pp[51];

  pmp[Id(-Nj+1)] = pm + pd;
  pp[Id(-Nj+1)] = C[0][-Nj+1] + pd*lamda_L;

  for(j=-Nj+2;j<=Nj-1;j++)
  {
    pmp[Id(j)] = pm-pu*pd/pmp[Id(j-1)];
    pp[Id(j)] = C[Id(0)][Id(j)]-pp[Id(j-1)]*pd/pmp[Id(j-1)];
  }

  C[Id(1)][Id(Nj)] = (pp[Id(Nj-1)]+pmp[Id(Nj-1)]*lamda_U)/(pu+pmp[Id(Nj-1)]);
  C[Id(1)][Id(Nj-1)] = C[Id(1)][Id(Nj)] - lamda_U;

  for(j=Nj-2;j>=-Nj+1;j--)
  {
    C[Id(1)][Id(j)] = (pp[Id(j)]-pu*C[Id(1)][Id(j+1)])/pmp[Id(j)];
  }
}

```