Analytical solution to diffusion-convection PDE in 1D

Nasser M. Abbasi

March 11, 2019

This is a diffusion-convection PDE.

\[ \frac{\partial F(t, z)}{\partial t} = k \frac{\partial^2 F(t, z)}{\partial z^2} + \frac{v}{\partial z} \frac{\partial F(t, z)}{\partial z} \quad (1) \]

Where \( k \) is the diffusion constant and \( v \) is the convection speed. Boundary conditions are

\[ F(t, 0) = 0 \]
\[ F(t, L) = 1 \]

Initial conditions are

\[ F(0, z) = \begin{cases} 
0 & 0 \leq z < L \\
1 & z = L 
\end{cases} \]

The first step is to convert the PDE to pure diffusion PDE using the transformation

\[ F(t, z) = A(t, z) u(t, z) \]

Substituting this back in (1) gives

\[ A_t u + A u_t = k \left( A_{zz} u + 2A_z u_z + A u_{zz} \right) + v \left( A_z u + A u_z \right) \]

Dividing by \( A \) and simplifying

\[ \frac{A_t u}{A} + \frac{u_t}{A} = k \left( \frac{A_{zz} u + 2A_z u_z + A u_{zz}}{A} \right) + \frac{A_z u + A u_z}{A} \]
\[ u_t = k A_{zz} A^{-1} + \frac{\partial A_z}{\partial z} + v A_{zz} A^{-1} u + \frac{2kA_z + vA}{A} u_z \quad (2) \]

To make (2) pure diffusion PDE, we want

\[ k \left( \frac{A_{zz}}{A} + \frac{\partial A_z}{\partial z} + v A_{zz} A^{-1} \right) u = 0 \quad (3) \]
\[ \left( \frac{2kA_z + vA}{A} \right) u_z = 0 \quad (4) \]

From (4) \( (2kA_z + vA) u_z = 0 \) or \( 2kA_z + vA = 0 \) or \( \frac{\partial A_z}{\partial z} + \frac{v}{\partial z} A = 0 \) which has the solution

\[ A(t, z) = C(t) e^{-\frac{v}{2k} z} \quad (5) \]

From (3) we want \( k \left( A_{zz} - \frac{\partial A_z}{\partial z} + v A_z \right) = 0 \). Substituting the result just obtained for \( A(t, z) \) in (3) gives
\[
\frac{\psi^2}{4k^2} C(t) \frac{e^{-\frac{\psi}{2}z}}{2k} - \frac{dC(t)}{dt} \frac{1}{k} \frac{e^{-\frac{\psi}{2}z}}{2k} - \frac{\psi^2}{2k} C(t) \frac{e^{-\frac{\psi}{2}z}}{2k} = 0
\]
\[
\frac{\psi^2}{4k^2} C(t) - \frac{1}{k} C'(t) - \frac{\psi^2}{2k} C(t) = 0
\]

Hence

\[ C(t) = C_1 e^{-\frac{\psi}{2}t} \]

For some constant \( C_1 \). The constant \( C_1 \) ends up canceling out at the very end. Hence we set it to 1 now instead of carrying along in all the derivation in order to simplify notations. Therefore \( C(t) = e^{-\frac{\psi}{2}t} \).

Substituting this into (5) gives the transformation function

\[ A(t, z) = e^{-\left(\frac{\psi^2}{4k} + \frac{\psi}{2k}\right)z} \]

Using this in (2) gives the pure diffusion PDE to solve

\[ u_t = k u_{zz} \quad (6) \]

Converting the original boundary conditions from \( F \) to \( u \) gives

\[
F(t, 0) = 0
\]
\[
A(t, 0) u(t, 0) = 0
\]
\[
e^{-\frac{\psi}{2k}} u(t, 0) = 0
\]
\[
u(t, 0) = 0
\]

And

\[
F(t, L) = 1
\]
\[
A(t, L) u(t, L) = 1
\]
\[
e^{-\frac{\psi}{2k}} u(t, L) = 1
\]
\[
u(t, L) = e^{\left(\frac{\psi^2}{4k} + \frac{\psi}{2k}\right)L}
\]

And for the initial conditions

\[
F(0, z) = \begin{cases} 0 & 0 \leq z < L \\ 1 & z = L \end{cases}
\]
\[
A(0, z) u(0, z) = \begin{cases} 0 & 0 \leq z < L \\ 1 & z = L \end{cases}
\]
\[
e^{-\frac{\psi}{2k}} u(0, z) = \begin{cases} 0 & 0 \leq z < L \\ 1 & z = L \end{cases}
\]
\[
u(0, z) = \begin{cases} 0 & 0 \leq z < L \\ e^{\frac{\psi}{2}} & z = L \end{cases}
\]

Therefore the new PDE to solve is

\[ u_t = k u_{zz} \]

With time varying boundary conditions
\[ u(t, 0) = 0 \]
\[ u(t, L) = e^{\left( \frac{x^2}{4k} + \frac{t^2}{4k} \right)} \]

And initial conditions
\[ u(0, z) = \begin{cases} 
0 & 0 \leq z < L \\
e^{\frac{t^2}{4k}} & z = L 
\end{cases} \]

To solve this using separation of variables, the boundary conditions has to be homogenous. Therefore we use standard method to handle this as follows. Let
\[ u(t, z) = \phi(t, z) + u_E(t, z) \tag{7} \]

Where \( u_E(t, z) \) is the steady state solution which needs to only satisfy the boundary conditions and \( \phi(t, z) \) satisfies the PDE but with homogenous boundary conditions. Therefore
\[ u_E(t, z) = u(t, 0) + z \left( \frac{u(t, L) - u(t, 0)}{L} \right) \]
\[ u_E(t, z) = \frac{z}{L} e^{\left( \frac{x^2}{4k} + \frac{t^2}{4k} \right)} \]

And (7) becomes
\[ u(t, z) = \phi(t, z) + \frac{z}{L} e^{\left( \frac{x^2}{4k} + \frac{t^2}{4k} \right)} \]

Substituting the above in (6) gives
\[ \frac{\partial \phi}{\partial t} + z \frac{u^2}{L 4k} e^{\left( \frac{x^2}{4k} + \frac{t^2}{4k} \right)} = k \frac{\partial^2 \phi}{\partial z^2} \]
\[ \frac{\partial \phi}{\partial t} = k \frac{\partial^2 \phi}{\partial z^2} + z \frac{u^2}{L 4k} e^{\left( \frac{x^2}{4k} + \frac{t^2}{4k} \right)} \]

Or
\[ \frac{\partial \phi}{\partial t} = k \frac{\partial^2 \phi}{\partial z^2} + Q(t, z) \tag{8} \]

This is diffusion PDE with homogenous B.C. with source term
\[ Q(t, z) = -\frac{d}{dt}u_E(t, z) \]

Now we find \( \phi(t, z) \). Since this solution needs to satisfy homogenous boundary conditions, we know the solution to pure diffusion on bounded domain with source present is by given by the following eigenfunction expansion
\[ \phi(t, z) = \sum_{n=1}^{\infty} b_n(t) \sin \left( \sqrt{\lambda_n} z \right) \tag{8A} \]

Where eigenvalues are \( \lambda_n = \left( \frac{n \pi}{L} \right)^2 \) for \( n = 1, 2, \ldots \) and \( \sin \left( \sqrt{\lambda_n} z \right) \) are the eigenfunction. Substituting the above in (8) in order to obtain an ODE to solve for \( b_n(t) \) gives
\[ \sum_{n=1}^{\infty} b_n'(t) \sin \left( \sqrt{\lambda_n} z \right) \approx k \sum_{n=1}^{\infty} -b_n(t) \lambda_n \sin \left( \sqrt{\lambda_n} z \right) + Q(t, z) \tag{9} \]

Expanding \( Q(t, z) \) in terms of eigenfunctions
Applying orthogonality

\[ Q(t, z) = \sum_{n=1}^{\infty} q_n(t) \sin(\sqrt{\lambda_n} z) \]

But

\[ \int_0^L Q(t, z) \sin(\sqrt{\lambda_n} z) \, dz = q_n(t) \frac{L}{2} \]

Hence from (9A) we find

\[ q_n(t) = \frac{(-1)^n \nu^2 e^{\frac{\nu^2}{4L} t + \frac{\nu^2}{8L}}} {2kL\sqrt{\lambda_n}} \]

To solve the above ODE, the integrating factor is \( \mu = e^{k\lambda_n t} \), therefore

\[ b_n(t) e^{k\lambda_n t} = \int_0^t q_n(\tau) e^{k\lambda_n \tau} \, d\tau + C_n \]

\[ b_n(t) = e^{-k\lambda_n t} \int_0^t q_n(\tau) e^{k\lambda_n \tau} \, d\tau + C_n e^{-k\lambda_n t} \]

Using the above in (7) gives

\[ u(t, z) = \frac{z}{L} e^{\frac{\nu^2}{4L} + \frac{\nu^2}{8L}} \sum_{n=1}^{\infty} \left( \int_0^t q_n(\tau) e^{k\lambda_n(t-\tau)} \, d\tau + C_n e^{-k\lambda_n t} \right) \sin(\sqrt{\lambda_n} z) \]

\[ C_n \text{ is now found from initial conditions. At } t = 0 \text{ the above becomes} \]

\[ u(0, z) = \frac{z}{L} e^{\frac{\nu^2}{8L}} = \sum_{n=1}^{\infty} C_n \sin(\sqrt{\lambda_n} z) \]

Applying orthogonality

\[ \int_0^L \left( u(0, z) - \frac{z}{L} e^{\frac{\nu^2}{8L}} \right) \sin(\sqrt{\lambda_n} z) \, dz = \int_0^L \frac{C_n}{2} \sin(\sqrt{\lambda_n} z) \, dz \]

\[ \int_0^L u(0, z) \sin(\sqrt{\lambda_n} z) \, dz - \int_0^L \frac{z}{L} e^{\frac{\nu^2}{8L}} \sin(\sqrt{\lambda_n} z) \, dz = C_n \frac{L}{2} \]
But \( \int_0^L u(0, z) \sin(\sqrt{\lambda_n} z) \, dz = 0 \) since \( u(0, z) \) is zero everywhere except at the end point. And

\[
- \int_0^L z e^{\frac{\sqrt{\lambda_n} z}{L}} \sin(\sqrt{\lambda_n} z) \, dz = \frac{(-1)^n e^{\frac{\sqrt{\lambda_n} L}{L}}}{\lambda_n}
\]

Therefore

\[
\frac{(-1)^n e^{\frac{\sqrt{\lambda_n} L}{L}}}{\lambda_n} = C_n \frac{L}{2}
\]

\[
C_n = \frac{2(-1)^n e^{\frac{\sqrt{\lambda_n} L}{L}}}{L} \quad (10A)
\]

And the solution (10) becomes

\[
u(t, z) = \frac{z}{L} e^{\left(\frac{\sqrt{\lambda_n} t}{L}\right) e^{\frac{\sqrt{\lambda_n} z}{L}}} + \sum_{n=1}^{\infty} \left( \int_0^t q_n(\tau) e^{k\lambda_n(\tau-t)} d\tau + C_n e^{k\lambda_n t} \right) \sin(\sqrt{\lambda_n} z) \quad (11)
\]

But

\[
\int_0^t q_n(\tau) e^{k\lambda_n(\tau-t)} d\tau = \int_0^t \frac{(-1)^n v^2 e^{\frac{\sqrt{\lambda_n} z}{L}}}{2kL\sqrt{\lambda_n}} e^{k\lambda_n(\tau-t)} d\tau = \frac{2(-1)^n v^2 e^{\frac{\sqrt{\lambda_n} z}{L}}}{n\pi (4k^2\lambda_n + v^2)} - 1
\]

Hence (11) becomes

\[
u(t, z) = \frac{z}{L} e^{\left(\frac{\sqrt{\lambda_n} t}{L}\right) e^{\frac{\sqrt{\lambda_n} z}{L}}} + \sum_{n=1}^{\infty} \frac{2(-1)^n v^2 e^{-\frac{\sqrt{\lambda_n} z}{L}}}{n\pi (4k^2\lambda_n + v^2)} - 1 + \frac{2(-1)^n e^{\frac{\sqrt{\lambda_n} L}{L}}}{L} e^{k\lambda_n t} \sin(\sqrt{\lambda_n} z) \quad (12)
\]

We now convert back to \( F(t, z) \). Since \( F(t, z) = A(t, z) u(t, z) \) and \( A(t, z) = e^{-\left(\frac{\sqrt{\lambda_n} t}{L}\right) e^{\frac{\sqrt{\lambda_n} z}{L}}} \) then the final solution is

\[
F(t, z) = e^{-\left(\frac{\sqrt{\lambda_n} t}{L}\right) e^{\frac{\sqrt{\lambda_n} z}{L}}} u(t, z)
\]

The following is animation of the solution for 30 seconds, side-by-side with numerical solution.

The following is the code used

```
ClearAll[t, z, L0, k, lam, n, v, f]

L0 = 10;
v = 1/2;
k = 1/2;
ode = D[f[t, z], t] == k*D[f[t, z], (z, 2)] + v*D[f[t, z], z];
ic = f[0, z] == Piecewise[{{1, z == L0}, {0, True}}];
bc = {f[t, 0] == 0, f[t, L0] == 1};
sol = NDSolve[{ode, bc, ic}, f, {t, 0, 100}, {z, 0, 10}];

λ[n_] := (n^2*v^2)/(4*k) + (v*L0)/(2*k)

 Manipulate[Grid[{{Row["t": t], SpanFromLeft},
{Plot[Evaluate[f[t, z] /. sol], {z, 0, 10},
PlotRange -> {{0, 10}, {-0.2, 1}},
ImageSize -> 300, PlotLabel -> "NDSolve solution"},
{Plot[λ[n] /. sol, {n, 1, max},
PlotRange -> {{0, 100}, {-10, 10}}] + (2/L0)*(((-1)^n*Exp[(L0*v)/(2*k)])/Sqrt[λ[n]])
Exp[(-λ[n])*("k"t)]
Sin[Sqrt[λ[n]]*z], {n, 1, max}, {z/L0}]
Exp[(v^2*t)/(4*k) + (v*L0)/(2*k)]
}
}, Frame -> True, Alignment -> Center]
```
Plot[Evaluate[fAnalytical[t, z]], {z, 0, 10},
PlotRange -> {{0, 10}, {-0.2, 1}},
ImageSize -> 300, PlotLabel -> "Analytical solution",
PlotStyle -> Red]],
{{t, 0, "t"}, 0, 30, 0.01},
TrackedSymbols :> {t}]

References

