Fire Technology manuscript No.
(will be inserted by the editor)

Response to "Unexpected Oscillations in Fire Modelling Inside a Long Tunnel" by Ang et al.

Kevin McGrattan · Randall McDermott

Received: date / Accepted: date

To the editor

This is a response to a letter published June 2020 in *Fire Technology* by Chin Ding Ang, Guillermo Rein and Joaquim Peiro, entitled "Unexpected Oscillations in Fire Modelling Inside a Long Tunnel". The letter describes oscillatory behavior of the pressure field in numerical simulations of tunnel fires performed with the Fire Dynamics Simulator (FDS), versions 6.1.1 through 6.6.0. Here we would like to explain the causes of these oscillations and recent efforts to improve the simulation of tunnel fires.

Before embarking on a description of the relevant equations, we will first provide a simple physical explanation for the problem. FDS uses a low-Mach number formulation of the Navier-Stokes equations, and one of the consequences of this approximation is that the speed of sound is essentially infinite in the model. For most building fire simulations, this raises no serious issues, but when a fire is located within a tunnel that might be several kilometers long, the impact of this assumption is significant. Here's why—imagine a large fire within a tunnel, pulsating turbulently. The heat release rate of such a fire is not perfectly steady, but rather fluctuates in time, causing fluctuations in the rate of gas expansion (velocity divergence). If the tunnel walls are sealed, the excess volume is discharged out the end of the tunnel instantaneously via an increased volumetric flow rate. Of course, in reality, the excess volume propagates at roughly the speed of sound as a pressure pulse. However, given that the speed of sound is roughly 300 m/s, the time step required to track

K. McGrattan
National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 20899
Tel.: +1 301 975 2712
E-mail: kevin.mcgrattan@nist.gov

R. McDermott
Tel.: +1 301 975 4710
E-mail: randall.mcdermott@nist.gov

these pressure pulses could be up to several orders of magnitude smaller than the time step required to resolve the fire-driven flow.

Now consider the governing equations. In its simplest form, the equation governing the gas velocity, $\mathbf{u}$, is given by

$$\frac{\partial \mathbf{u}}{\partial t} + \hat{\mathbf{F}} + \frac{1}{\rho}\nabla \tilde{p} = 0 \tag{1}$$

$\hat{\mathbf{F}}$ represents all the convective, diffusive, and force terms, ρ is the gas density, and $\tilde{p}$ is the pressure perturbation, the difference between the local and background pressure, $p - p_0$. If we apply the divergence operator to this equation, we obtain an elliptic partial differential equation for the pressure perturbation, $\tilde{p}$:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \tilde{p} \right) = - \left[\frac{\partial}{\partial t} (\nabla \cdot \mathbf{u}) + \nabla \cdot \hat{\mathbf{F}} \right] \tag{2}$$

Notice on the right hand side of this equation there is a time derivative of the gas divergence, $\nabla \cdot \mathbf{u}$. The divergence is a measure of the local expansion or contraction of the gas, and in a fire simulation, the local heat release rate is the principal component of the divergence.

The discretized form of Eq. (2) is a system of linear equations, $\mathrm{A}\tilde{p} = \mathbf{b}$, where A is an $N \times N$ matrix and N is the number of grid cells in the mesh. $\mathbf{b}$ represents the right hand side of Eq. (2). Because the elements of the matrix involve the density, ρ, and because ρ changes twice per time step, the matrix would require two time-consuming inversions per time step, a cost that would slow the simulations by an order of magnitude. To alleviate this problem, we rewrite Eq. (2) as follows:

$$\nabla^2 \left(\frac{\tilde{p}^n}{\rho} \right) = \nabla \cdot \left(\tilde{p}^{n-1} \nabla \frac{1}{\rho} \right) - \left[\frac{\partial}{\partial t} (\nabla \cdot \mathbf{u}) + \nabla \cdot \hat{\mathbf{F}} \right] \tag{3}$$

The superscript n represents the time step, meaning that at time step n, an updated pressure is found using its value from the previous time step. To increase the accuracy of this scheme, Eq. (3) is solved repeatedly forcing the old and new pressure terms to converge to within some preset tolerance. The advantage of Eq. (3) over Eq. (2) is that the matrix A does not change with each time step, and all necessary manipulation of A for the solution of the linear system of equations can be done once at the start of the simulation.

Now to the problem at hand. The letter authors note that their simulations of a 50 MW tunnel fire, in the presence of ventilation fans, exhibit very large oscillations in the mass flow exiting the tunnel. The cause of the problem is related to the iterative pressure solution. Eq. (3) is solved twice per time step. In the first part of the time step, or predictor phase, the pressure, $\tilde{p}$, is used to estimate the velocity, $\mathbf{u}$, at the next time step using a first-order accurate forward Euler update derived from Eq. (1). In the second part of the time step, or corrector phase, the estimated velocity is "corrected" and made second-order accurate in time. This basic predictor-corrector updating scheme is referred to as a second-order Runge-Kutta method. The spurious oscillations the authors notice are caused by the fact that the iterative pressure scheme in FDS had been initialized by the pressure field from the previous substep. It had been assumed that the pressure field obtained in the corrector phase of the previous time step should be used to initialize the iterative scheme in the predictor phase of the new time step and vice versa. With the exception of long tunnels, the pressure fields obtained in the two phases of the time step are fairly similar, and the aforementioned strategy allows for second-order convergence in time without iteration for some manufactured solutions to the governing equations. However, for tunnels, it has been observed that the predictor and corrector pressure fields are very different. As a result, the initial estimates for the pressure in both the predictor and corrector phases were inaccurate, leading to an excessive number of iterations. By default, FDS would stop after 10 iterations, not enough to overcome the initial bad estimate of pressure.

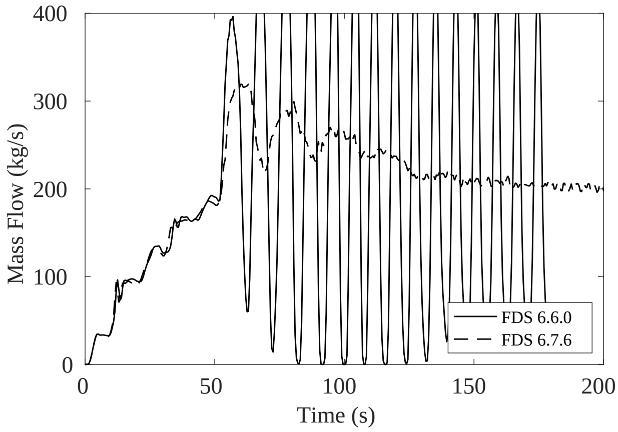

Fig. 1 Mass flow exiting the open end of a tunnel with a 50 MW fire and ventilation fans. The problematic oscillations are exhibited in the earlier version of FDS.

A simple one line code change has solved the problem first identified by Ang and his colleagues. The predictor and corrector values of H are stored separately and the initial predictor value for time step n is taken as the final predictor value from $n-1$, and similarly for the corrector. This change was made in FDS version 6.7.6, released in June 2021 (see Fig. 1).

The authors' sample case demonstrating the problem raises yet another challenging issue related to the simulation of tunnel fires; that is, multiple meshes and parallel computing. Ang et al. used a single numerical mesh to span the entire length of the tunnel, leading to extremely long run times because FDS can only exploit a handful of processors to solve the governing equations on a single mesh. If the mesh is broken up into multiple, abutting meshes, the simulation can be sped up considerably, assuming that the equation for pressure (3) can be solved quickly and efficiently on multiple meshes. The

Poisson equation solver used in FDS can only be applied on a single mesh, and these meshes are processed in parallel by either multiple processes on a single computer or over a fast computer network. FDS exploits the same iterative pressure-solving technique described above to "stitch together" the pressure fields of the multiple meshes. The more the pressure field fluctuates from one time step to another, the more iterations are necessary to obtain an accurate solution.

One way to speed up convergence of the algorithm has recently been developed. A detailed description of this method can be found in the FDS Technical Reference Guide that accompanies the latest release of FDS (6.7.6). First, we write Eq. (3) in an equivalent form as it is written in the FDS documentation:

$$\nabla^2 H = -\left[\frac{\partial}{\partial t}(\nabla\cdot\mathbf{u}) + \nabla\cdot\mathbf{F}\right] \quad ; \quad H = \frac{\tilde{p}}{\rho} + \frac{\|\mathbf{u}\|}{2} \tag{4}$$

The vector $\mathbf{F}$ is similar to $\hat{\mathbf{F}}$ from above, but it now absorbs the extra pressure term. The combination of $\tilde{p}/\rho$ and $\|\mathbf{u}\|/2$ is a convenient way to save on the cost of finite differencing. Now we decompose H as follows:

$$H(\mathbf{x}) = \bar{H}(x) + H'(\mathbf{x}) \tag{5}$$

where $\mathbf{x} = (x, y, z)$ and x is the lengthwise spatial coordinate of the tunnel. The equation for $\bar{H}$ becomes:

$$\frac{\mathrm{d}^2\bar{H}}{\mathrm{d}x^2} = -\left[\frac{\partial}{\partial t}\overline{(\nabla\cdot\mathbf{u})} + \overline{\nabla\cdot\mathbf{F}}\right] \tag{6}$$

The bars over the terms on the right side of the equation denote integration of these terms in the lateral directions; thus, these terms are functions solely of x and form a simple ordinary differential equation that can be solved over the length of the entire tunnel with a simple, fast tri-diagonal linear solver. The full three-dimensional Poisson equation for H' can then be solved on each mesh in parallel using $\bar{H}$ as an initial estimate, or *preconditioner*.

An example of the new pressure solving technique is shown in Fig. 2. In this example, a 1600 m long tunnel with a square cross section that is 10 m by 10 m has a 60 MW fire at its mid point and a ventilation current of 2 m/s blowing towards its open end. The FDS simulation is performed with 80 separate meshes spanning the length of the tunnel. The left hand plot of Fig. 2 displays the mass flow rate through the tunnel. The upstream mass flow rate is ramped up quickly to approximately 240 kg/s at time $t = 0$. Without the preconditioning technique in the pressure solver discussed above, the downstream mass flow rate oscillates wildly, but with the preconditioner, the mass flow rate is as expected, eventually leveling off at 240 kg/s as the tunnel gas and surface temperatures reach steady-state. The right hand plot of Fig. 2 displays the steady-state profile of perturbation pressure, $\tilde{p}$, along the length of the tunnel for three grid resolutions. Note that even with the preconditioner, there is still a considerable fluctuation in the pressure due to the low Mach

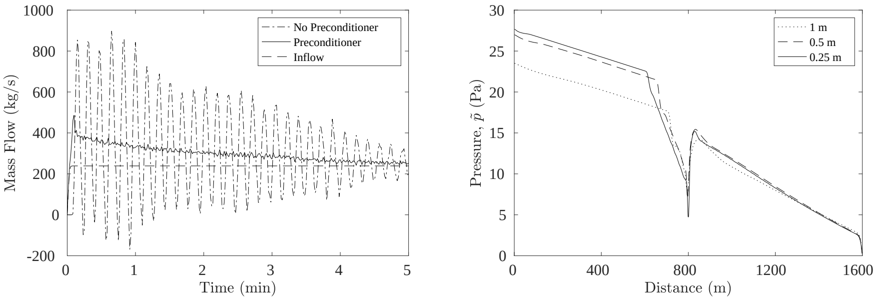

Fig. 2 (Left) Mass flow rate as a function of time through a tunnel of length 1600 m with a 60 MW fire and 2 m/s ventilation velocity. (Right) Perturbation pressure, $\tilde{p}$, along the length of the tunnel. The tunnel is open at 1600 m. The calculation is performed at three grid resolutions.

number assumption, but the accelerated convergence of the pressure solver successfully handles it.

This pressure solving feature has been implemented in FDS version 6.7.6. It is limited to cases where multiple meshes span a tunnel with no open boundaries along its lateral walls, ceiling and floor. The ends of the tunnel can be specified as either passive openings or as forced flow boundaries. If the tunnel does have openings along the lateral walls, one can still create openings to an ambient void to represent leakage, access points, and so on. In FDS, these openings tend to dampen pressure oscillations because they provide a pressure relief that absorbs excess volume generated by the pulsating fire.