PDEModelica and Breathing in an Avalanche

Jan Šilar*+, Filip Ježek#, Jiří Kofránek+

+ Institute of Pathological physiology, First Faculty of Medicine, Charles University, Prague, Czech republic
# Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
* Corresponding Author Institute of Pathological physiology, U nemocnice 5, Praha 2 128 00, Czech Republic, jansilar@jansilar.cz

Abstract

This paper presents an updated version of Modelica language extension for partial differential equations (PDE) called PDEModelica and implementation of its support in OpenModelica. This support is limited to 1-dimensional problems and the first and second partial derivatives. PDEModelica is introduced by a string equation model and later by a real life model of respiration during a snow burial. This model describes CO_2 advection and diffusion in snow described by advection-diffusion PDE.

PDEModelica, PDE, avalanche survival

Introduction

PDEModelica is a Modelica language extension for partial differential equations (PDE). It was designed by Levon Saldamli (Saldamli, 2006). This original extension is currently not supported by any tool. We focused on a subset of this extension for 1-dimensional models only and introduced several changes and enhancements. Support for the renewed extension has been implemented in OpenModelica. We present the extension using a simple string equation model at first and then a real-life problem of modelling respiration during a snow burial.

In the past four decades, avalanches were responsible for around 100 deaths annually in the European Alps only (Techel et al., 2016). When a victim is buried by an avalanche he or she repetitively inspires previously expired air as the motion of air in snow is restricted. The body metabolism consumes O_2 and produces CO_2 and thus the concentration of O_2 decreases and the concentration of CO_2 increases in the inspired and expired air. The concentrations of O_2 and CO_2 are partially restored by diffusion. But this process is not fast enough and if the victim is not rescued within approximately 15 minutes he or she may die of asphyxiation, i.e. a lack of oxygen supply to the cells. Asphyxia could be caused by a variety of situations, including excess of CO_2. More than 75 % of deaths in an avalanche are caused by asphyxia (McIntosh et al., 2007). However the content of oxygen in snow should satisfy the body needs – Radwin (Radwin et al., 2001) proved, that volunteers buried in snow with the removal of the expired gas did not have any problems even after an hour long burial. In contrast, no removal resulted in serious hypercapnia (i.e. an excessive amount of carbon dioxide in blood) within 10 minutes. In this paper, we focus on modeling of CO_2 diffusion only, as the O_2 is then a very similar problem.

Modeling task

It is assumed, that a potential cavity around the mouth and the nose significantly increase the chance of survival. Roubík et al. carried out an experiment (Roubík et al., 2015) where the volunteers were breathing through a tube whose end opened into a cavity in snow of various volumes. They proved that the size of the cavity has a significant impact on the concentration of O_2 and CO_2 in the inspired and expired air. There are at least two possible mechanisms causing this effect. First, the small cavity has a small surface of the air-snow boundary and so the resistance for the air flux is high. This causes an increase in the work of breathing, an increase in the metabolism rate and thus an increase in O_2 consumption and CO_2 production. Second, the expired air is mixed with more fresh air in the cavity and then the inspired air is also more fresh. Both mechanisms probably take place in the process. The question is which one dominates. We were asked to help with the investigation using a model.

Methods

PDEModelica

Let us introduce all new language elements of PDEModelica on the advection equation model:

```
1 model advection "advection equation"
2  parameter Real pi =
       Modelica.Constants.pi;
3  parameter DomainLineSegment1D omega(L =
       1, N = 100);
4  field Real u( domain= omega);
5 initial equation
6  u = sin(2*pi *omega.x);
7 equation
8  der(u) + pder(u, x) = 0 indomain omega;
9  u = 0 indomain omega.left;
10 u = extrapolateField(u)
             indomain omega.right;
11end advection;
```

- The Domain `omega` represents the geometrical domain where the PDE holds. The domain is defined using the built-in record `DomainLineSegment1D` (line 3). This record contains among others `L` – the length of the domain, `N` – the number of grid points, `x` – the coordinate variable and the regions `left`, `right` and `interior`, representing the left and right boundaries and the interior of the domain.

- The field variable `u` is defined using a new keyword `field` (line 4). The `domain` is a mandatory attribute to specify the domain of the field.

- The `indomain` operator specifies where the equation containing the field variable holds. It is utilised in the initial conditions (IC) of the fields, in the PDE and in the boundary conditions (BC). The syntax is
  `equation indomain domain.region`
  If the `.region` is omitted, `.interior` is the default.

- The IC of the field variable `u` is written using an expression containing the coordinate variable `omega.x.` (line 6).

- The PDE contains a partial space derivative written using the `pder` operator (line 8). Also the second derivative is allowed (not in this example), the syntax is e.g. `pder(u,x,x)`. It is not necessary to write e.g. `omega.x` in `pder`, even though `x` is a member of `omega`.

- The BC is on line 9. The current limitation is that BCs may be written only in terms of variables that are spatially differentiated.

- All fields that are spatially differentiated must have at each boundary either BC or extrapolation. This extrapolation should be done automatically by the compiler, but this has not been implemented yet. The current workaround is the usage of the `extrapolateField()` operator directly in the model.

Comparison to the original version of PDEModlica

Our extension is restricted to 1-dimensional models only. This allows much simpler domain definition using the built-in `DomainLineSegment1D` record compared to the original extension which enables arbitrary geometry domain definition in multiple dimensions.

`pder()` is used instead of `der()` for partial derivatives. A shortcut to leave out the full qualification of the `x` coordinate is established. This was probably intended in the original extension also, but was not explicitly mentioned.

`indomain` is used instead of `in` as it is suggested in (Fritzson, 2015) because `in` is already utilized in `for` loops. `indomain` is mandatory not only in the BCs but also in the ICs and the PDEs here.

Field literals are written as expressions containing the coordinate variable `x` and thus the special syntax for the field literal constructor of the original extension was suppressed.

Solution process

The PDEs are solved using the method of lines (MOL) (Schiesser, 2012): during flattening of the model, the fields are replaced by arrays and the space derivatives

DOI 10.3384/ecp17132367

are replaced by finite differences (currently only the central difference is implemented)

$$\frac{\partial u}{\partial x} \rightarrow \frac{u[i+1] - u[i-1]}{2 \cdot dx},$$

$$\frac{\partial^2 u}{\partial x^2} \rightarrow \frac{u[i+1] - 2 \cdot u[i] + u[i-1]}{dx^2}$$

and thus the PDEs are converted into a system of ODEs. This resulting system may be written in standard Modelica and is solved by current OpenModelica solvers. The combination of the central space difference with an implicit Euler time solver results in a backward-time centered-space (BTCS) scheme which is a common finite difference method. The usage of the trapezoid time solver results in the Crank-Nicolson method (Strikwerda, 2004). Other time solvers may be also successful, even though the resulting methods were not investigated. A selection of a proper time solver is important.

It is also substantial to select a proper time step, so that the Courant–Friedrichs–Lewy (CFL) condition (Courant et al., 1967) is fulfilled.

To enable PDEModelica in OpenModelica, the compiler flag `--grammar=PDEModelica` must be set.

The features for the plotting fields (arrays) have not been implemented in OpenModelica yet. We use Octave to load the result file and plot the desired variables.

Model of breathing in snow

For the first stages of the research, the problem is simplified into a gas flow to and from a spherical snowball (see Figure 1). Due to small pressure differences, the gas is modeled as incompressible. The only significant pressure difference could occur at the boundary between the cavity and the snow, but Roubík et al. (Roubík et al., 2015) did experience only small pressure differences.

The snow is a porous material, formed by ice and air. The CO_2 could flow and diffuse across the snow through the air gaps. Given the ice density (916 kg/m3) and the density of snow (100 - 400 kg/m3) the snow consists of at least 55 % of air. Therefore, the air could penetrate through the snow and mix with the air captured within the snow. For simplification, we exclude the solubility in ice and possible melted water. The gas has a volumetric concentration of CO_2, the O_2 is omitted, but it follows the same principles. The gas transport in snow is modeled using the advection-diffusion equation.

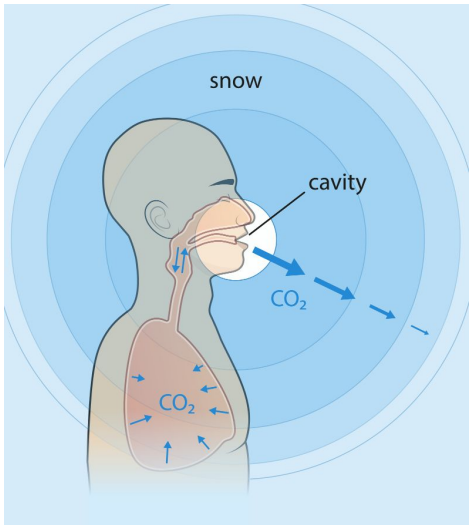

Figure 1 – Model schematics. The organism is producing CO_2 in a constant rate and it is concentrating in the lungs. The lungs expire to and inspire from a cavity, in which the air is ideally mixed. The air flux is given. The partial concentration of CO_2 in the cavity is drained by advection and diffusion through the snow. The dead volume in the airways is omitted.

Advection-diffusion equation and its formulation in PDEModelica

The advection-diffusion equation assuming the incompressible gas flow is

$$\frac{\partial c}{\partial t} + u \cdot \nabla c - D \triangle c = 0$$

where c is the concentration, u is the velocity of advection and D is a diffusion coefficient. We express this equation in the spherical coordinates. As our problem is spherically symmetrical, all derivatives except the derivatives in a radial direction are equal to zero. Then we obtain

$$\frac{\partial c}{\partial t} + \left(u - \frac{2D}{r}\right)\frac{\partial c}{\partial r} - D\frac{\partial^2 c}{\partial r^2} = 0,$$

$$u = \frac{q}{4\pi r^2}.$$

where r is the radius and q is a volumetric flow given by the lungs. This equation contains two partial derivatives. Using the principles described in the paragraphs above, the formulation of the advection-diffusion equation in PDEmodelica is written in the Code listing 1. The full model of CO_2 breathing is available online[1].

```
model sb1m
  (...)
  Real C_CS "concentration on cavity-snow
interface";
  DomainLineSegment1D omega(L = 0.5, N =
100, x0 = R_C) "x is actually r, center on
the left";
  field Real C_S(domain = omega)
"concentration of CO2 in snow";
(...)
//Left BC during exhalation, extrapolation
during inhalation
  C_S = if exhale then C_CS else
extrapolateField(C_S) indomain omega.left;
//The advection-diffusion equation
  der(C_S) + (q / (4 * pi * omega.x ^ 2)
       - 2 * D_S / omega.x) * pder(C_S, x)
       - D_S * pder(C_S, x, x) = 0
       indomain omega;
end sb1m;
```

Code listing 1: the advection-diffusion equation formulation in PDEModelica. New language elements are highlighted in purple.

Note, that the boundary conditions are switched with extrapolation every breathing cycle as the flux direction changes. This demonstrates the acausality of the proposed approach.

Results

In the presented model, we use the arbitrary parameter values to demonstrate the principles of CO_2 distribution. The exact identification of the values is a subject of additional research. However, the resulting trends are consistent with expectation and plausibility personally confirmed by the authors of (Roubík et al., 2015).

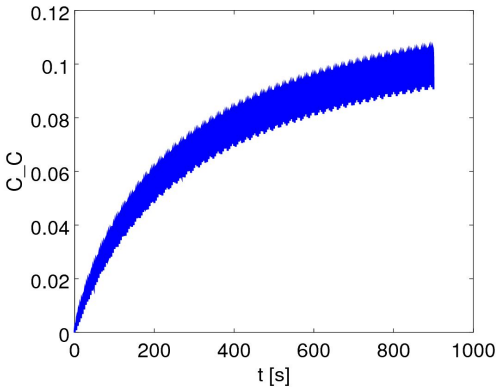

Figure 2 Concentration (fraction) in the cavity C_C (the Cavity volume 1 L) is changing between the inhale and the exhale.

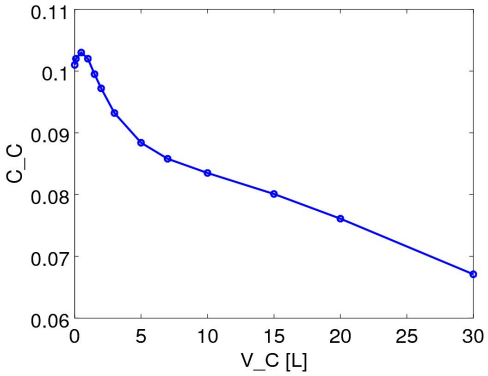

Figure 3: Average concentration C_C (average over 4 full breathing periods) in the snow cavity in time for various cavity sizes (V_C) at 15min.

The CO_2 concentration in the cavity rises with each expiration (Figure 2) and is rising towards an equilibrium. However, when the concentration of CO_2 is about 2 % the victim feels respiratory stimulation (here approx. 200s), at 6 % starts mental confusion (approx. 400s), followed by unconsciousness at 10% (approx. 800s) and later by death.

In Figure 3 we investigate the influence of the cavity size - the larger cavity, the longer the subject could survive (i.e. the lower cavity CO_2 concentration). If the volume of the cavity is smaller than 1L, CO_2 concentration does not change significantly. The size of the cavity has a huge impact from 1 to 5 L, but then the response becomes nearly linear. Unfortunately, the CO_2 concentration remains at unsatisfactory high levels.

[1] https://github.com/jansilar/snowbreathing/

DOI 10.3384/ecp17132367

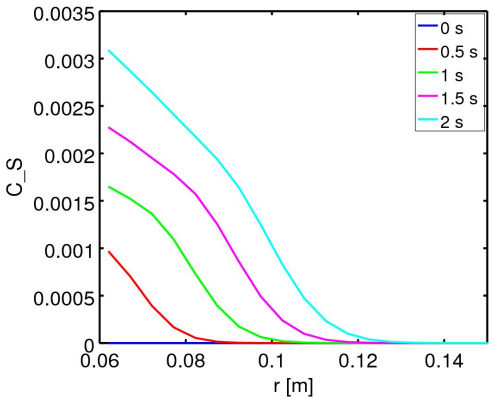

Figure 4 The concentration gradient dependable on the radius of the snowball during the first exhale (a half of breath period), the cavity volume 1 L.

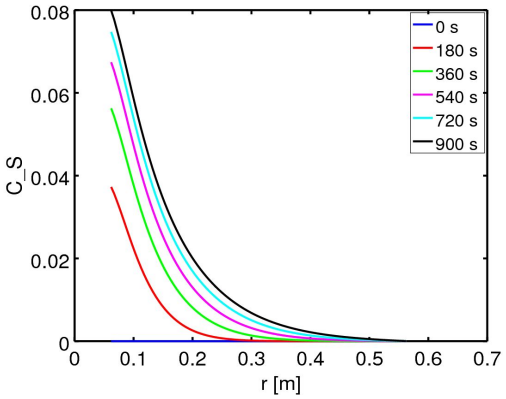

Figure 5 The concentration gradient dependable on the radius of the snowball during a longer periods, at the end of the breath period (i.e. after the exhale), the cavity volume 1 L.

We can see advection and diffusion of CO_2 into snow during the first breath out close to the cavity (Figure 4). Thanks to the r^2 attenuation of advection velocity, the CO_2 concentration is virtually zero within a few centimeters of the snow. Note that for long time periods the CO_2 proceeds further. Despite, the concentration is negligible in around 50 cm as the distribution volume grows rapidly with the radius (Figure 5).

Solution process performance

The model was translated and simulated several times with a different number of grid points. The trapezoid solver was used. The time step was chosen proportionally to the space step. The stop time (model time) was 5 minutes. The translation and simulation time and the size of the model binary file are in Table 1.

N	Step	Trans	Simul	Size
50	0.02	3,5	8,2	252
100	0.01	4,9	13,7	418
200	0.005	5,5	39,2	759
500	0.002	12,2	224,9	1843

Table 1 Performance comparison: N – number of grid points, step – the time step (s), trans – the time of translation (s), simul – the time of simulation (s), size – the size of the model binary (kB).

The simulation time increase substantially with increasing number of grid points. The results seem satisfying even for the simulation using 100 grid points (plotted). On the other hand this results were not verified by comparison with a different PDE simulation tool.

Discussion

The snow-breathing model

A mathematical model could help to study the countermeasures to avoid asphyxiation. This work supports the usage of devices for CO_2 removal and explains the underlying processes with the goal to contribute to their construction. Some CO_2 removal devices already exists, including a tube device to divert the CO_2-rich exhale (Margid et al., 1998), but additional data are needed to prove their efficiency.

Changes in the human metabolism as a consequence of the increasing hypercapnia and hypoxia during the snow burial are not included in the model. Thus the presented model cannot describe the real process of breathing into snow. Development of the full model that includes the human physiology as well is the aim of the subsequent work. The current model has been greatly simplified by omitting oxygen, dead space in airways, solubility of CO2 in other body compartments and in water contained in snow and also rising breath work, which produces more CO_2. Thanks to Modelica implementation, it is planned to connect it directly with the most extensive open model of human physiology, the Physiomodel (Mateják and Kofránek, 2015).

The current parameters of the model are set arbitrarily and are not confirmed by the measurements. Therefore, the results may be taken as demonstrative only.

Alternative to PDEModelica

Instead of the presented solution, some other methods of using the PDEs in Modelica exists. One could make a usage of creating the PDEs in some other tool and then import them via FMI (Stavåker and Fritzson, 2014). For seamless Modelica integration, a dedicated library PDELib (Dshabarow et al., 2007) was developed. However, as it is not maintained, the examples are not working in the recent OpenModelica and Dymola versions.

We could have employed a manual PDE discretization and thus converting the PDEs into an ODE or DAE system. However using the manual discretization is in conflict with Modelica declarative philosophy. Utilising the language extension the modeller may focus on the model itself rather than its numerical solution. Any model written using the extension is more understandable and maintainable compared to using the manual discretization.

Conclusion

The presented model of breathing while buried in an avalanche has several limitations. The main purpose of this contribution was to demonstrate the ability of PDEModelica to solve PDE models. This enhancement is documented on the advection equation and then the advection-diffusion equation modelling breathing in snow after the avalanche burial. PDEModelica was able to successfully express these example models and the extended OpenModelica was able to solve them. Nevertheless the project is not finished and more work should be done. Automatic extrapolation on boundaries must be implemented. Both PDEModelica language extension and its implementation in OpenModelica have to be yet tested thoroughly on several different models and by comparison of results with reference PDE simulation tools.

References

Courant, R., Friedrichs, K., Lewy, H., 1967. On the Partial Difference Equations of Mathematical Physics. IBM J. Res. Dev. 11, 215–234.

Dshabarow, F., Cellier, F.E., Zimmer, D., Dshabarow, F., Com, C., Ch, I.E., 2007. Support for Dymola in the modeling and simulation of physical systems with distributed parameters. In: Proceedings of the 6th International Modelica Conference. pp. 683–690.

Fritzson, P., 2015. Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach. John Wiley & Sons.

Margid, J., Beidleman, N., Harmston, C., 1998. O2 and CO2 levels with the Black Diamond AvaLung during human snow burials lasting up to one hour. Proceedings of the.

Mateják, M., Kofránek, J., 2015. Physiomodel - an integrative physiology in Modelica. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). ieeexplore.ieee.org, pp. 1464–1467.

McIntosh, S.E., Grissom, C.K., Olivares, C.R., Kim, H.S., Tremper, B., 2007. Cause of death in avalanche fatalities. Wilderness Environ. Med. 18, 293–297.

Radwin, M.I., Grissom, C.K., Scholand, M.B., Harmston, C.H., 2001. Normal oxygenation and ventilation during snow burial by the exclusion of exhaled carbon dioxide. Wilderness Environ. Med. 12, 256–262.

Roubík, K., Sieger, L., Sykora, K., 2015. Work of Breathing into Snow in the Presence versus Absence of an Artificial Air Pocket Affects Hypoxia and Hypercapnia of a Victim Covered with Avalanche Snow: A Randomized Double Blind Crossover Study. PLoS One 10, e0144332.

Saldamli, L., 2006. PDEModelica A High-Level Language for Modeling with Partial Differential Equations (PhD Thesis.). Linkopings universitet.

Schiesser, W.E., 2012. The Numerical Method of Lines: Integration of Partial Differential Equations. Elsevier.

Stavåker, K., Fritzson, P. et al, 2014. PDE Modeling With Modelica Via FMI Import Of Hiflow3 C++ Components With Parallel Multi-core Simulations. Proceedings of the 55th Scandinavian Conference on Simulation and Modeling.

Strikwerda, J.C., 2004. Finite Difference Schemes and Partial Differential Equations. SIAM.

Techel, F., Jarry, F., Kronthaler, G., Mitterer, S., Nairz, P., Pavšek, M., Valt, M., Darms, G., 2016. Avalanche fatalities in the European Alps: long-term trends and statistics. Geogr. Helv. 71, 147–159.

DOI 10.3384/ecp17132367