Use of simulateBMP() v3.0

```
[sensorData, t, dt, pVariable, lx, ly] = ...
%          simulateBMP(imageFileName, scale, duration, ...
%          simMedium, simSource,recordVideo);
```

Simulates acoustic-wave propagation of pressure sources in 2D scenarios defined by an image file. Allows the use of multiple pressure sensors and multiple sources with free definition of their temporal development.

imageFileName is a string with the name or the BMP file.

Version 3.0 supports either BMP 256 or BMP 24 bit format.

It is convenient to select bit sizes that allow efficient computations. For them it is recommended to use previously the function checkFactors (of the kWave toolbox)
checkFactors(min_number, max_number)
Type 2^n sizes are efficient (128, 256, 512, 1024, etc.)
For example: using a bmp image of 256x128 is more efficient than 250x120.
White points represents air. Black points represents elements with total reflection (walls). Red points represents sources. Green points represents sensor. Brown points represents absorbent material. Version 3.0 supports absorption, but is still in testing. Supports a single value of Sabine's coefficient for all materials defined with brown color.

scale

It is the dimension of each minimum lattice jump in meters. The lattice expected by simulateBMP assumes that the spacing is equal in both x and y.
Using scale=0.01 (1 cm) gives a simulation sampling frequency of 114 kHz and a valid Nyquist frequency of 17 kHz for the propagating wave components.
If another value of scale is chosen, version 3.0 reports the available fNyquist at the start.

duration

It is the duration time of the simulation in seconds.
In normal use k-Wave does not require this parameter, since it is determined in relation to the maximum time that a wave can take to traverse the entire simulation space. The function simulateBMP() is intended to simulate waves with longer periods of time. It requires to define the duration in a mandatory way (there is no value defined by default).

simMedium

It is a structure (struct) that contains variables related to the simulation environment.
In version 3.0 it is possible to carry out simulations without specifying all the parameters, since it includes default values for almost each one. However, since this is a required argument in the function, it is essential to specify at least one of the following.
simMedium.c0 specifies the propagation speed in m/s in air (white areas in the image). If the c0 field does not exist in the structure, it assumes the default value of 344 m/s

simMedium.density specifies the density of air. The default value is 1.2 Kg/m3
simMedium.alpha specifies the Sabine absorption coefficient (not to be confused with the energy absorption coefficient used in k-Wave). An alpha=1 coefficient would mean total absorption. The default value if the field is not specified is alpha=0.5. However, it only has an effect on the simulation if the image contains brown dots.

simMedium.speedRatio specifies the ratio of propagation speeds between the absorbing medium and air, considering speedRadio= cAbsorb/c0. It is only used when there are brown points in the simulation. The default value is speedRatio=1.1.

simMedium.sign specifies the sign of the reflection (its phase). This is related to the fact that there is an uncertainty in determining a Sabine coefficient in relation to the phase of the reflection. In other words, a Sabine coefficient alpha=0.5 indicates that half the energy is absorbed, but the wave that is not absorbed could be reflected with the same pressure phase (closer to the reflection in a closed tube) or with the inverted phase (closer to a tube with an open end). The default value is sign=1 (pipe closed).

simMedium.CFL specifies the value of the Courant–Friedrichs–Lewy coefficient related to the stability of the solution of differential equations. The default value is CFL = 0.5.

simMedium.showAbsorptionMask is a logical variable that specifies to display the absorbing material mask (as dotted areas) in the simulation. The default value is true.

simMedium.showWallMask is a logical variable that specifies to display the reflective material mask in the simulation. The default value is true.

simMedium.showSourceMasks is a logical variable that specifies to display the source mask in the simulation. The default value is true.

simMedium.showSensorMasks is a logical variable that specifies to display the sensor mask in the simulation. The default value is true.

simMedium.showLegendMask is a logical variable that specifies to display the legend or comment mask (pure gray dots in the original bmp). The default value is true.

simSource

It is a structure (struct) that determines the properties of the sources used

simSource.type indicates the signal type of the sources. Depending on the type of signal selected, other parameters of the structure will be taken into account to define it completely, although all the necessary ones have values assigned by default to make fast simulations easier.

simSource.type = 'impulse' is an impulse signal (only a unit value in the first element of the vector with the rest of the duration equal to zero). Use the parameters simSource.amplitude (amplitude in pascals), and simSource.mode (specifying whether the source will be in 'additive' mode (in which the source pressure will be added to whatever the rest of the simulation specifies for that point)) or 'dirichlet' mode (in which the selected pressure is a fixed boundary condition for the point where the source is defined).

simSource.type = 'nCycles' is a single frequency component signal with a specified number of cycles emitted. Logically the number of cycles specified should be less than the total duration of the simulation. If this is not true, the simulation is still carried out, but the result would be that of a pure sinusoidal. Additionally, k-Wave in this case issues a warning indicating that the source data is greater than the simulation times. It uses the following parameters: simSource.amplitude (amplitude in pascals), simSource.n=2 (number of cycles), simSource.f0 = 1000 (frequency in Hz), and simSource.mode.

simSource.type = 'whiteNoise' defines the use of white noise with a certain duration that, if it is less than the total duration of the simulation, allows reverberation times to be analyzed. It uses simSource.amplitude (amplitude in pascals), simSource.duration (noise duration in seconds from the start of the simulation), and simSource.mode.

simSource.type = '(text with math equation) ' defines the pressure signal using the equation enclosed in single quotes. The duration of this signal is equal to the total duration of the simulation. The equation contains all the necessary information so it is only necessary to specify the mode through simSource.mode. Usage example: simSource.type = '4*sin(2*pi*1000*t+pi/4)';

simSource.type = 'audio' defines the pressure signal by any variable defined in the MATLAB Workspace. The variable to use will be defined by setting simSource.p equal to that variable (for example, simSource.p=audioSamples). It is also required to specify simSource.fs (sampling frequency of the audio included in the MATLAB variable), and simSource.mode. It is important to mention that in this case the program, before simulating, performs an automatic resampling to adapt the audio sampling frequency (simSource.fs) to the simulation sampling frequency (fs, which is usually much higher in

value) . When using this type of source, the program, once the simulation is finished, resample the signals captured by the sensors to return the result in the sampling frequency of the audio. In short, sensorData returned by simulateBMP() will in this case have the same sample rate as the input audio.

simSource.fCut specifies the cutoff frequency of an antialias filter. If not specified, it takes a value less than the Nyquist frequency, which can be calculated as fNyquist=cMin/2/dx. For cMin = 344 m/s and dx=0.01 m; fNyquist=17.2 kHz and fCut=15 kHz.

simSource.order specifies the order of the antialiased filter generated with the butter() function. The default value is order=15.

Default values of simSource fields that specify parameters

simSource. mode = 'additive' (takes this value when mode is not specified, but also when mode is specified incorrectly, although in this case it warns with a message that the additive mode was set).

simSource.amplitude = 10;

simSource.fs=44100; (only used with simSource.type = 'audio')

simSource.f0=1000; (only used with simSource.type = 'nCycles')

simSource.n=3; (only used with simSource.type = 'nCycles')

simSource.duration=duration/5; (only used with simSource.type = 'whiteNoise')

In the simulateBMP() 3.0 version it is possible to use an array of sources with individual definition of the signal of each source. In that case the variable source.p must have a number of rows equal to the number of source and a number of columns equal to the number of signal samples of each source. Additionally, it is necessary that the order of the temporal source definitions coincide with the order in which Matlab places the points corresponding to the sources in an array (the kgrid space of kWave).

Sample script tu use simulateBMP()

```
% Minimum version(using default values)
%% 1-Setup parameters
ImageFile='HelloWorld.bmp';
scale=1e-2; % 1 cm
duration = 1e-3; % 1 ms
simSource.type='nCycles';
recordVideo=false;

%% 2-Call simulateBMP
[sensor_data, t, dt, pVariable, lx, ly] = ...
simulateBMP(imageFileName, scale, duration, ...
simMedium, simSource,recordVideo);

%% 3-Plot the results
tVariable=(0:length(pVariable)-1)*dt;
plot(t,sensor_data)
title('Sensor'); xlabel('t [s]'); ylabel('p [Pa]'); grid on
```