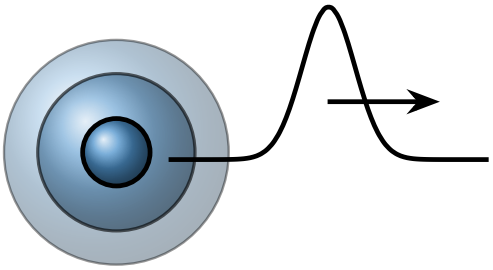
APECSS

(Acoustic Pulse Emitted by Cavitation in Spherical Symmetry)



Fabian Denner
Sören Schenke

21 October 2022

Contents

Chapter 1

About APECSS

APECSS is a software tool to compute pressure-driven bubble dynamics and the resulting acoustic emissions. It is written exclusively in C and has been developed with simplicity, versatility and performance in mind. The acronym APECSS stands for "Acoustic Pulse Emitted by Cavitation in Spherical Symmetry".

The main features of APECSS are:

- Includes widely-used models for the bubble dynamics (Rayleigh-Plesset, Keller-Miksis, Gilmore).
- Acoustic emissions of the bubble under different assumptions (incompressible, quasi-acoustic, fully compressible).
- Prediction of the formation and attenuation of shock fronts emitted by the bubble.
- Viscoelastic media (Kelvin-Voigt, Zener, Oldroyd-B).
- Lipid monolayer coating of the bubble as used for ultrasound contrast agents.
- All ODEs are solved with an in-built fifth-order Runge-Kutta scheme with fourth-order error estimate, based on Dormand and Prince [4].
- APECSS has no external dependencies, aside from some common C headers (math.h, stdio.h, stdlib.h, string.h).

The APECSS repository is located at `https://github.com/polycfd/apecss` and structured as follows:

- The `documentation/` folder contains a short documentation of APECSS, written in Latex.
- The `examples/` folder contains representative examples of how to use APECSS and to demonstrate the most important features of APECSS. These examples also serve to validate APECSS against results reported in the literature.
- The `include/` folder contains the `apecss.h` header file, in which all variable, macro and function of APECSS are defined.
- The `lib/` folder in which the APECSS library is compiled (at least if you follow the 2.1).
- The `src` folder contains all source files (`*.c`) of APECSS.
- The `.clang-format` file, which defines the formatting rules for the source code. A *clang* formatter (supported by most IDEs and editors) should be used for contributions to APECSS. The formatter should recognize this `.clang-format` file automatically.
- The `.gitignore` file telling *git* which folders and files to ignore.
- The `LICENSE.txt` file containing the Mozilla Public License Version 2.0.
- The `README.txt` file with the most important information about APECSS.

The development of APECSS has directly benefitted from research funding provided by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), grant number 441063377.

Chapter 2

Using APECSS

2.1 Installation

Installing APECSS is easy. After downloading APECSS in the directory `<path to APECSS>`, define the following environment variables:

- `APECSS_DIR` to the directory in which APECSS is located. Using `bash`, for instance, simply execute the command `export APECSS_DIR <path to APECSS>` or, even better, add this command to your `bash` profile.
- `USRLIB_DIR` to the directory in which libm.a or `libm.dylib` (the standard *math* library) is located. This may, for instance, be `/usr/lib64/` on Linux systems or `/usr/lib/` on MacOS systems.

Now navigate into the folder `$APECSS_DIR/lib` and execute `./compile_lib.sh`. This command will compile the APECSS library using the `cmake` with the `CMakeLists.txt` file provided in this folder. By default, APECSS is compiled with double precision and in *Release* mode, meaning all optimization flags are enabled. That's it, you've succesfully installed the APECSS library!

2.2 Running APECSS

There are several ways in which you can use the APECSS library. You can either incorporate selected features of APECSS into your own software or you can program an interface to use APECSS as a standalone program.

Some representative examples are given in the `$APECSS_DIR/examples` directory. Each directory contains the following:

- A `README.txt` file explaining the purpose and specificities of this/these example(s).
- A `src/` folder with a file called `*_apecss.c` that acts as the standalone interface to the APECSS library. This file contains the `main()` function and any additional functionality required to simulate a specific scenario.
- A `build/` folder containing the `CMakeLists.txt` file and a shell script `compile.sh` with which this example can be compiled using the command `./compile.sh`.
- One or several `*.apecss` files in which the options for a specific case are defined.

2.2.1 The `*.apecss` options file

The `*.apecss` file is the primary way of passing options, such as the size of the bubble, the density of the liquid or the type of results you want to have written out, to APECSS. In the examples provided

in the `$APECSS_DIR/examples` directory, the name of the `*.apecss` file is passed as an argument with the call to run APECSS, e.g. executing `./<APECSS-example> -options run.apecss` to use the an options file named `run.apecss`.

Any `*.apecss` file has the following sections:

- `BUBBLE`: Information related to the bubble, such as its initial radius R_0 or the Rayleigh-Plesset model that is used to solve its dynamics.
- `GAS`: Properties and equation of state of the gas.
- `LIQUID`: Properties, type (i.e. Newtonian or viscoelastic) and equation of state of the liquid.
- `INTERFACE`: Properties of the gas-liquid interface.
- `RESULTS`: Results of the bubble dynamics and the acoustic emissions that should be written out.
- `ODESOLVER`: Parameters of the ODE solver.

Each section is terminated with `END`. Any of these sections and any of the options that may be defined within each section are optional; they are read if they are present, otherwise the default values (typically set in `apecss_*_setdefaultoptions()`) or values set in the code calling it will be used. The options file is read by the function `apecss_options_readfile()`.

The relevant options that are available are discussed in the following chapters of this documentation in the context of the theoretical framework of APECSS. For instance, the available options used to define a certain Rayleigh-Plesset model are discussed in Section 3.1, where the theory of the implemented Rayleigh-Plesset models is described.

2.3 Units

APECSS assumes SI units or any appropriate combination of SI units at all times, e.g. when reading user-defined options and in all internal computations. To avoid any misunderstanding, the SI base units are the following:

- Time in seconds [s]
- Length in meter [m]
- Mass in kilogram [kg]
- Temperature in Kelvin [K]
- Electric current in Ampere [A]
- Amount of substance in mole [mol]
- Luminosity in candela [cd]

2.4 Programming in APECSS

All routines are placed in source files that relate to parts of the code, distinguished by physical phenomena (e.g. `nonspherical.c`), fluid type (e.g. `liquid.c`) or computational operations (e.g. `results.c`).

All decalarations and definitions are located in the header file `apecss.h`.

2.4.1 Macros

Macros are used as shortcuts to define frequently-used constants (e.g. `APECSS_PI`), for frequently-used computational operations (e.g. `APECSS_MAX`) and for computational operations that depend on the chosen machine precision (e.g. `APECSS_SIN`). Furthermore, options related to different numerical models are represented by logically named flags.

2.4.1.1 Macros related to machine precision

APECSS can be used with different floating point precisions: double precision (default) and long double precision (`APECSS_PRECISION_LONGDOUBLE`).

Based on the chosen precision, `APECCS_FLOAT` is defined as the standard floating point type. In addition, the following precision-dependent computational operations are defined based on the chosen floating point precision:

- `APECSS_ABS(a)`: Absolute value of a.
- `APECSS_CEIL(a)`: a rounded to the nearest integer larger than a.
- `APECSS_COS(a)`: Cosine of a.
- `APECCS_EPS`: Returns a value that is close to machine epsilon.
- `APECSS_EXP(a)`: e to the power a.
- `APECSS_LOG(a)`: Natural logarithm of a.
- `APECSS_POW(a,b)`: Power b of a.
- `APECSS_SIN(a)`: Sine of a.
- `APECSS_SMALL`: Returns a number significantly smaller than machine.
- `APECSS_SQRT(a)`: Square root of a.
- `APECSS_STRINGTOFLOAT(a)`: Conversion of a string to float a.

To ensure compatibility for different floating point precisions, it is paramount to use the standard floating point type `APECCS_FLOAT` and the operator definitions given above consistently throughout APECSS.

2.4.1.2 Computational operations and predefined constants

Macros that provide a shortcut to frequently-used computational operations are:

- `APECSS_POW2(a)`: Returns a^2
- `APECSS_POW3(a)`: Returns a^3
- `APECSS_POW4(a)`: Returns a^4
- `APECCS_MAX(a,b)`: Returns the maximum of a and b.
- `APECSS_MAX3(a,b,c)`: Returns the maximum of a, b and c.
- `APECSS_MIN(a,b)`: Returns the minimum of a and b.
- `APECSS_MIN3(a,b,c)`: Returns the minimum of a, b and c.

Macros that provide a shortcut to frequently-used constants are:

- `APECSS_PI`: Returns π
- `APECSS_E`: Returns e
- `APECSS_ONETHIRD`: Returns $1/3$
- `APECSS_ONESIXTH`: Returns $1/6$
- `APECCS_AVOGADRO`: Returns the Avogadro constant.
- `APECSS_LN_OF_2`: Returns the natural logarithm of 2.
- `APECSS_LN_OF_10`: Returns the natural logarithm of 10.
- `APECSS_LARGE`: Returns a large number, defined as 10^{15}.

2.4.1.3 Flags for model options

All model options are represented by human-readable flags. If explicitly indicated as such, these flags are defined in such a way (with integer values being a multiple of 2), that a bit-wise comparison can be performed. Bit-wise comparison may be used for options that are checked frequently and for options that can have several building blocks.

Options of the Runge-Kutta scheme used to discretize the ODEs:

- `APECSS_RK54_7M`: RK5(4)7M (minimum truncation) coefficients of Dormand and Prince [4]
- `APECSS_RK54_7S`: RK5(4)7S (stability optimized) coefficients of Dormand and Prince [4]

Rayleigh-Plesset schemes:

- `APECSS_BUBBLEMODEL_RP`: Standard Rayleigh-Plesset model, (3.1)
- `APECSS_BUBBLEMODEL_RP_ACOUSTICRADIATION`: Rayleigh-Plesset model incl. acoustic radiation term, (3.2)
- `APECSS_BUBBLEMODEL_KELLERMIKSIS`: Keller-Miksis model, (3.3)
- `APECSS_BUBBLEMODEL_GILMORE`: Gilmore model, (3.4)

Equation of state of the gas:

- `APECSS_GAS_IG`: Ideal gas EoS
- `APECSS_GAS_HC`: Ideal gas EoS with van-der-Waals hardcore
- `APECSS_GAS_NASG`: Noble-Abel-stiffened-gas EoS

Equation of state of the gas:

- `APECSS_LIQUID_TAIT`: Tait EoS
- `APECSS_LIQUID_NASG`: Noble-Abel-stiffened-gas EoS

Viscoelasticity of the liquid:

- `APECSS_LIQUID_NEWTONIAN`: Newtonian liquid
- `APECSS_LIQUID_KELVINVOIGT`: Kelvin-Voigt solid
- `APECSS_LIQUID_ZENER`: Zener solid (standard linear solid model)
- `APECSS_LIQUID_OLDROYDB`: Oldroyd-B liquid

Lipid monolayer coating of the gas-liquid interface:

- `APECSS_LIPIDCOATING_NONE`: No lipid monolayer coating
- `APECSS_LIPIDCOATING_MARMOTTANT`: Lipid monolayer coating described by the model of Marmottant *et al.* [13]
- `APECSS_LIPIDCOATING_GOMPERTZFUNCTION`: Redefine the Marmottant model with a Gompertz function [6].

Acoustic excitation applied to the bubble:

- `APECSS_EXCITATION_NONE`: No external excitation.
- `APECSS_EXCITATION_SIN`: Sinusoidal excitation.

Model to compute the acoustic emissions of the bubble:

- `APECSS_EMISSION_NONE`: Emissions are not modelled.
- `APECSS_EMISSION_INCOMPRESSIBLE`: Emissions are assumed to occur in an incompressible fluid.
- `APECSS_EMISSION_FINITE_TIME_INCOMPRESSIBLE`: Emissions are assumed to occur in an incompressible fluid, but the finite propagation speed given by the speed of sound is taken into account.
- `APECSS_EMISSION_QUASIACOUSTIC`: Emissions are modelled under the quasi-acoustic assumption of Gilmore [5].
- `APECSS_EMISSION_KIRKWOODBETHE`: Emissions are modelled based on the Kirkwood-Bethe hypothesis.

2.4.1.4 Others

Other predefined macros are used to define the verbosity of APECSS, the length of strings and arrays, as well as to help with debugging:

- APECSS_DATA_ALLOC_INCREMENT: The increment for dynamics re-allocation of arrays.
- APECSS_STRINGLENGTH: The standard length of a string.
- APECSS_STRINGLENGTH_SPRINTF: The standard length of a string to written out in the terminal.
- APECSS_STRINGLENGTH_SPRINTF_LONG: The standard length of a long string to written out in the terminal.
- APECSS_WHERE: Outputs in the terminal the file name and line number where the macro is called.
- APECSS_WHERE_INT(a): Outputs in the terminal the file name and line number where the macro is called, plus the integer value a.
- APECSS_WHERE_FLOAT(a): Outputs in the terminal the file name and line number where the macro is called, plus the floating point value a.

2.4.2 Structures

Structures (`struct`) are used in APECSS to group variables and functions, and to provide a modular layout of the code that enables reusing different parts of it.

The structure `APECSS_Bubble` is the central structure of APECSS as it contains all the information related to a bubble. There is, of course, no a priori limit on how many copies of this structure you can have, for instance, a multi-bubble simulation with 100 bubbles would naturally have 100 objects of type `struct APECSS_Bubble`. Aside from key information, such as the bubble radius, the `APECSS_Bubble` structure contains pointers to the properties of the liquid the bubble is immersed in (`struct APECSS_Liquid`), the properties of the gas the bubble contains (`struct APECSS_Gas`) and the properties of its interface (`struct APECSS_Interface`). If applicable, the `APECSS_Bubble` structure also points to the structure with the information of the driving acoustic excitation (`struct APECSS_Excitation`), the structure handling the acoustic emissions (`struct APECSS_Emissions`), and the structure containing the desired results (`struct APECSS_Results`). Lastly, the `APECSS_Bubble` structure has function pointers to functions related to one of its optional structures (e.g. for the emissions) that are directly called in the context of the bubble, i.e. where it is not a priori known if the optional structure is allocated or not.

The (optional) structure `APECSS_Emissions` is, as the name suggests, related to the acoustic emissions of a bubble. If allocated, it contains information about how to handle the acoustic emissions, function pointers referring to the functions used to advance the acoustic emissions using a Lagrangian approach and, very importantly, the linked list of emissions nodes (`struct APECSS_EmissionNode`) that carry the actual information of the acoustic emissions. The `APECSS_EmissionNode` structure holds the information (e.g. radial location, velocity, pressure) associated with a specific emission node.

The (optional) structure `APECSS_Results` holds all the results the user may want to have written out. For performance reasons, the results are in general not written to disk on-the-fly, but are stored in arrays and dumbed to disk at the end of the simulation. The `APECSS_Results` structure contains optional structures for the results of the Rayleigh-Plesset model (`struct APECSS_ResultsBubble`) and for the acoustic emissions (`APECSS_ResultsEmissions`).

2.4.3 A word on function pointers

APECSS uses function pointers extensively. Function pointers are an elegant means in `C` to add complexity and functionality to code yet still retain a slim code, avoid redudant code and, if nothing else, to avoid a large number of costly conditional statements. However, function pointers can quickly make a code unreadable and obscure what is actually happening if they are used without care. In order to keep the use of function pointers in APECSS transparent, the adopt the convention that all function pointers are set in `apecss_*_processoptions()` functions, e.g. `apecss_gas_processoptions()`.

2.4.4 Code formatting

To ensure a consistent formatting, please use a *clang* formatter that formats the file automatically upon saving. The file defining the formatting of the APECSS source code (`.clang-format`) is part of the respository.

Chapter 3

Bubble dynamics

The dynamic behaviour of the bubble is modelled with a Rayleigh-Plesset-type (RP) model, assuming spherical symmetry. This requires to choose a suitable RP model (Section 3.1) and define appropriate conditions for the gas (Section 3.2), the liquid (Section 3.3), the interface (Section 3.4), as well as at infinity (Section 3.5). The results that APECSS can write out based on the RP model are explained in Section 3.6.

All ordinary differential equations (ODEs) are solved using the embedded RK5(4) scheme of Dormand and Prince [4], whereby a fifth-order Runge-Kutta scheme is used to solve the ODEs and the corresponding fourth-order Runge-Kutta scheme is used to estimate the solution error. Based on this solution error, the time-step Δt used to advance the solution of the ODEs is chosen.

Section	Command	Description
`BUBBLE`	`InitialRadius <float>`	The initial radius R_0 of the bubble.
	`PressureAmbient <float>`	The ambient pressure p_0.
	`InitialGasPressure <float>`	The initial gas pressure $p_{G,0}$, if different from p_0 or the corresponding Laplace pressure.

3.1 Rayleigh-Plesset models

APECSS offers four Rayleigh-Plesset-type models to simulate pressure-driven bubble dynamics: the standard Rayleigh-Plesset model without and with acoustic radiation damping, the Keller-Miksis model and the Gilmore model.

Section	Command	Description
`BUBBLE`	`RPModel RP`	Standard Rayleigh-Plesset model, Eq. (3.1). This is the default.
	`RPModel RPAR`	Rayleigh-Plesset model including acoustic radiation damping, Eq. (3.2).
	`RPModel KM`	Keller-Miksis model, Eq. (3.3).
	`RPModel Gilmore`	Gilmore model, Eq. (3.4).

The standard Rayleigh-Plesset (RP) model is given as [11]

$$R\ddot{R} + \frac{3}{2}\dot{R}^2 = \frac{p_\mathrm{L} - p_\infty}{\rho_{\ell,\mathrm{ref}}}, \tag{3.1}$$

where R is the bubble radius, p_L is the pressure of the liquid at the bubble wall, p_∞ is the pressure

of the liquid at infinite distance from the bubble, p_G is the pressure of the gas inside the bubble and $\rho_{\ell,\mathrm{ref}}$ is the *constant* density of the liquid.

To incorporate acoustic radiation in the liquid and the associated damping, the modified Rayleigh-Plesset model is given as [1]

$$R\ddot{R} + \frac{3}{2}\dot{R}^2 = \frac{p_L - p_\infty}{\rho_{\ell,\mathrm{ref}}} + \frac{R\,\dot{p}_G}{\rho_{\ell,\mathrm{ref}}\,c_{\ell,\mathrm{ref}}}, \tag{3.2}$$

where $c_{\ell,\mathrm{ref}}$ is the *constant* reference speed of sound of the liquid. The last term on the right-hand side accounts for acoustic radiation in the liquid. This modified RP model is frequently used to simulate medical ultrasound applications [19] as well as sonoluminescence [1]. It follows directly from the Keller-Miksis model, Eq. (3.3), which incorporates the compressibility of the liquid, by assuming the Mach number of the bubble wall is small, $M_\ell = \dot{R}/c_{\ell,\mathrm{ref}} \ll 1$. Eq. (3.2) is, consequently, only valid for small Mach numbers $M_\ell = \dot{R}/c_{\ell,\mathrm{ref}} \ll 1$ [14, 16], although feasible results have frequently been obtained with Eq. (3.2) for Mach numbers $M_\ell \sim 1$ [1].

The Keller-Miksis model [9, 16], which incorporates the compressibility of the liquid to first order, is given as

$$\left(1 - \frac{\dot{R}}{c_{\ell,\mathrm{ref}}}\right) R\ddot{R} + \frac{3}{2}\left(1 - \frac{\dot{R}}{3\,c_{\ell,\mathrm{ref}}}\right)\dot{R}^2 = \left(1 + \frac{\dot{R}}{c_{\ell,\mathrm{ref}}}\right)\frac{p_G - p_\infty}{\rho_{\ell,\mathrm{ref}}} + \frac{p_L - p_G}{\rho_{\ell,\mathrm{ref}}} + \frac{R\,\dot{p}_G}{\rho_{\ell,\mathrm{ref}}\,c_{\ell,\mathrm{ref}}}, \tag{3.3}$$

where $c_{\ell,\mathrm{ref}}$ is the speed of sound of the liquid. Both $\rho_{\ell,\mathrm{ref}}$ and $c_{\ell,\mathrm{ref}}$ are assumed to be constant, limiting the Keller-Miksis model to moderate liquid pressures ($p_L \lesssim 10^8\,\mathrm{Pa}$).

Based on the Kirkwood-Bethe hypothesis, Gilmore [5] derived a second-order ordinary differential equation describing the radial dynamics of a bubble in a compressible liquid,

$$\left(1 - \frac{\dot{R}}{c_L}\right) R\ddot{R} + \frac{3}{2}\left(1 - \frac{\dot{R}}{3c_L}\right)\dot{R}^2 = \left(1 + \frac{\dot{R}}{c_L}\right) H + \left(1 - \frac{\dot{R}}{c_L}\right)\frac{R\dot{H}}{c_L}, \tag{3.4}$$

where c_L is the speed of sound of the liquid at the bubble wall, and $H = h_L - h_\infty$ and $\dot{H} = \dot{h}_L - \dot{h}_\infty$ are the enthalpy difference between the bubble wall and infinity and its derivative, respectively. The enthalpy h and the speed of sound c are defined by an appropriate equation of state as a function of pressure, with $h_L = h(p_L)$, $h_\infty = h(p_\infty)$ and $c_L = c(p_L)$, detailed in 3.3.

3.2 The gas

In APECSS, every bubble needs to contain a gas, which requires to select an appropriate equation of state and defined meaningful properties.

Section	Command	Description
`GAS`	`EoS IG`	Ideal gas equation of state. This is the default.
	`EoS HC`	Ideal gas equation of state with van-der-Waals hardcore.
	`EoS NASG`	Noble-Abel-stiffened-gas equation of state [12].
	`PolytropicExponent <float>`	Polytropic exponent Γ_g
	`ReferencePressure <float>`	Reference pressure $p_{g,\mathrm{ref}}$
	`ReferenceDensity <float>`	Reference density $\rho_{g,\mathrm{ref}}$
	`HardcoreRadius <float>`	Hardcore radius r_{hc}
	`CoVolume <float>`	Co-volume b_g
	`TaitPressureConst <float>`	Pressure constant B_g
	`MolecularWeight <float>`	Molecular weight $\mathcal{M}_g$ of the gas.
	`MolecularDiameter <float>`	Molecular kinematic diameter $\mathcal{D}_g$ of the gas.

Using the ideal gas EoS, the pressure and its derivative are given as

$$p_{\mathrm{G}} = p_{\mathrm{G,ref}} \left(\frac{\rho}{\rho_{\mathrm{g,ref}}} \right)^{\Gamma_{\mathrm{g}}} \tag{3.5}$$

$$\dot{p}_{\mathrm{G}} = \frac{\dot{\rho}_{\mathrm{G}}\, \Gamma_{\mathrm{g}}\, p_{\mathrm{G}}}{\rho_{\mathrm{G}}}, \tag{3.6}$$

$$\tag{3.7}$$

including a van-der-Waals hardcore in the ideal gas model, the pressure and its derivative follow as

$$p_{\mathrm{G}} = p_{\mathrm{G,ref}} \left(\frac{\rho^3 - r_{\mathrm{hc}}^3}{\rho_{\mathrm{g,ref}}^3 - r_{\mathrm{hc}}^3} \right)^{\Gamma_{\mathrm{g}}} \tag{3.8}$$

$$\dot{p}_{\mathrm{G}} = -3\, \frac{p_{\mathrm{G}}\, \Gamma_{\mathrm{g}}\, R^2\, \dot{R}}{R^3 - r_{\mathrm{hc}}^3}, \tag{3.9}$$

and using the Noble-Abel-stiffened-gas EoS, the pressure and its derivative are [3]

$$p_{\mathrm{G}} = (p_{\mathrm{G,ref}} + B_{\mathrm{g}}) \left[\frac{\rho_{\mathrm{g}}\, (1 - b_{\mathrm{g}}\, \rho_{\mathrm{g,ref}})}{\rho_{\mathrm{g,ref}}\, (1 - b_{\mathrm{g}}\, \rho_{\mathrm{G}})} \right]^{\Gamma_{\mathrm{g}}} - B_{\mathrm{g}} \tag{3.10}$$

$$\dot{p}_{\mathrm{G}} = \frac{\dot{\rho}_{\mathrm{G}}\, \Gamma_{\mathrm{g}}\, (p_{\mathrm{G}} + B_{\mathrm{g}})}{\rho_{\mathrm{G}}\, (1 - b_{\mathrm{g}}\, \rho_{\mathrm{G}})} \tag{3.11}$$

For all three equations of state, the the gas density and its derivative are given by

$$\rho_{\mathrm{G}} = \rho_{\mathrm{g,ref}} \left(\frac{R_0}{R} \right)^3 \tag{3.12}$$

$$\dot{\rho}_{\mathrm{G}} = -3\, \rho_{\mathrm{G}}\, \frac{\dot{R}}{R}. \tag{3.13}$$

The hardcore radius r_{hc} and the co-volume b_{g} are set by default to -1. If the `HC` or `NASG` model is chosen, the user either has to pass values for the hardcore radius r_{hc} or the co-volume b_{g}, respectively. Alternatively, the molecular weight $\mathcal{M}_{\mathrm{g}}$ and the molecular kinematic diameter $\mathcal{D}_{\mathrm{g}}$ of the gas may be defined instead of r_{hc} or b_{g}, APECSS then computes, based on the bubble size, the correct hardcore radius r_{hc} or co-volume b_{g}. Based on the molecular weight $\mathcal{M}_{\mathrm{g}}$, the bubble contains

$$N_{\mathrm{G}} = N_{\mathrm{A}}\, \frac{\rho_{\mathrm{G},0} V_0}{\mathcal{M}_{\mathrm{g}}} \tag{3.14}$$

molecules, where N_{A} is the Avogadro constant (see macro `APECSS_AVOGADRO`), $\rho_{\mathrm{G},0}$ is the initial gas density and V_0 is the initial bubble volume. As per the molecular kinetic diameter $\mathcal{D}_{\mathrm{g}}$ of the gas molecules, the volume of each molecule is

$$V_{\mathrm{mol}} = \frac{\pi}{6}\, \mathcal{D}_{\mathrm{g}}^3. \tag{3.15}$$

The van-der-Waals hardcore radius is then readily defined as

$$r_{\mathrm{hc}} = \sqrt[3]{\frac{3}{4\pi} f_{\mathrm{mol}}\, V_{\mathrm{mol}}\, N_{\mathrm{G}}} \tag{3.16}$$

and the co-volume of the gas is given as

$$b_{\mathrm{g}} = f_{\mathrm{mol}}\, N_{\mathrm{A}}\, \frac{V_{\mathrm{mol}}}{\mathcal{M}_{\mathrm{g}}}. \tag{3.17}$$

The semi-empirical constant f_{mol} is based on the repulsive forces acting between the molecules [10, 18], and is typically taken to be $f_{\mathrm{mol}} = 4$.

3.3 The liquid

In the same way that every bubble needs a to contain a gas, in APECSS every bubble needs to be surrounded by a fluid, which requires to select an appropriate equation of state and fluid type, as well as define meaningful properties.

Section	Command	Description
LIQUID	EoS Tait	The Tait EoS is applied to the liquid. Only relevant for the Gilmore model and acoustic emissions based on the Kirkwood-Bethe hypothesis.
	EoS NASG	The Noble-Abel-stiffened-gas EoS is applied to the liquid. Only relevant for the Gilmore model and acoustic emissions based on the Kirkwood-Bethe hypothesis.
	LiquidType Newtonian	Newtonian fluid. This is the default.
	LiquidType KelvinVoigt	Kelvin-Voigt solid.
	LiquidType Zener	Zener solid.
	LiquidType OldroydB	Oldroyd-B (or upper-convected Maxwell) fluid.
	PolytropicExponent <float>	Polytropic exponent Γ_ℓ
	ReferencePressure <float>	Reference pressure $p_{\ell,\mathrm{ref}}$
	ReferenceDensity <float>	Reference density $\rho_{\ell,\mathrm{ref}}$
	ReferenceSpeedofSound <float>	Reference speed of sound $\rho_{\ell,\mathrm{ref}}$
	CoVolume <float>	Co-volume b_ℓ
	TaitPressureConst <float>	Pressure constant B_ℓ
	Viscosity <float>	Newtonian viscosity μ_ℓ
	PolymerViscosity <float>	Polymer viscosity η_ℓ associated with viscoelasticity
	ShearModulus <float>	Shear modulus G_ℓ associated with viscoelasticity
	RelaxationTime <float>	Relaxation time λ_ℓ associated with viscoelasticity

The pressure at the bubble wall of a Newtonian liquid is given as

$$p_\mathrm{L} = p_\mathrm{G} - \frac{2\sigma}{R} - 4\,\mu_\ell \frac{\dot{R}}{R}, \tag{3.18}$$

where p_G is the gas pressure, see Section 3.2, σ is the surface tension coefficient of the interface, see Section 3.4, and μ_ℓ is the liquid viscosity. The derivative of Eq. (3.18) follows as

$$\dot{p}_\mathrm{L} = \dot{p}_\mathrm{G} + \frac{2\,\sigma\,\dot{R}}{R^2} + 4\,\mu_\ell \frac{\dot{R}^2}{R^2} - 4\,\mu_\ell\,\frac{\ddot{R}}{R}. \tag{3.19}$$

3.3.1 Equation of state

For the Gilmore model (3.4) and the acoustic emissions based on the Kirkwood-Bethe hypothesis (see Section 4.3), an equation of state (EoS) for the liquid has to be defined. Two EoS are currently available in APECSS: the Tait EoS and the NASG EoS.

Since the seminal work of Gilmore [5], the Tait EoS is traditionally used to describe the properties of the liquid in Eq. (3.4). The Tait EoS defines the density ρ, enthalpy h and speed of sound c as

$$\rho = \rho_{\ell,\mathrm{ref}} \left(\frac{p + B_\ell}{p_{\ell,\mathrm{ref}} + B_\ell} \right)^{\frac{1}{\Gamma_\ell}} \tag{3.20}$$

$$h = \frac{\Gamma_\ell}{\Gamma_\ell - 1} \frac{p + B_\ell}{\rho} \tag{3.21}$$

$$c = \sqrt{(\Gamma_\ell - 1)\,h}, \tag{3.22}$$

respectively, where B_ℓ is a pressure constant, Γ_ℓ is the polytropic exponent, $p_{\ell,\mathrm{ref}}$ is the reference pressure and $\rho_{\ell,\mathrm{ref}}$ is the reference density. For water, typical values are $\Gamma_\ell = 7.15$, $B_\ell = 3.046 \times 10^8\,\mathrm{Pa}$, $\rho_{\ell,\mathrm{ref}} = 997\,\mathrm{kg/m^3}$ and $p_{\ell,\mathrm{ref}} = 10^5\,\mathrm{Pa}$. However, the Tait EoS cannot provide physically-meaningful temperature values [17].

Using the Noble-Abel stiffened-gas (NASG) EoS [12] instead of the Tait EoS, the fluid properties are defined as [3]

$$\rho = \frac{K\,(p+B_\ell)^{\frac{1}{\Gamma_\ell}}}{1+b_\ell\,K\,(p+B_\ell)^{\frac{1}{\Gamma_\ell}}} \tag{3.23}$$

$$h = \frac{\Gamma_\ell}{\Gamma_\ell - 1}\frac{p+B_\ell}{\rho} - \frac{\Gamma_\ell\, b_\ell}{\Gamma_\ell - 1}\,(p+B_\ell) + b_\ell\, p \tag{3.24}$$

$$c = \sqrt{\Gamma_\ell \frac{(p+B_\ell)}{\rho - b_\ell \rho^2}}, \tag{3.25}$$

with $K = \rho_{\ell,\mathrm{ref}}/[(p_{\ell,\mathrm{ref}}+B_\ell)^{1/\Gamma_\ell}\,(1-b_\ell\rho_{\ell,\mathrm{ref}})]$ describing a constant reference state, and where b_ℓ is the co-volume of the liquid molecules. The NASG EoS reduces to the Tait EoS for $b_\ell = 0$. Appropriate properties for water have, for instance, been proposed by Chandran and Salih [2] as $\Gamma_\ell = 1.19$, $B_\ell = 6.2178 \times 10^8\,\mathrm{Pa}$, $b_\ell = 6.7212 \times 10^{-4}\,\mathrm{m^3/kg}$, $\rho_{\ell,\mathrm{ref}} = 997\,\mathrm{kg/m^3}$ and $p_{\ell,\mathrm{ref}} = 10^5\,\mathrm{Pa}$.

3.3.2 Viscoelasticity

Currently, APECSS supports three widely-used models for viscoelastic media: the Kelvin-Voigt model, the Zener model and the Oldroyd-B model. While the Kelvin-Voigt model merely yields an additional term in the expression for the pressure at the bubble wall, Eq. (3.18), the Zener and Oldroyd-B models each require to solve two additional ODEs.

3.3.2.1 Kelvin-Voigt model

To model a Kelvin-Voigt medium, the elasticity of the medium is modelled by the term

$$\frac{4}{3}\,G\,\frac{R^3 - R_0^3}{R^3}, \tag{3.26}$$

which contributes to Eq. (3.18) to obtain

$$p_\mathrm{L} = p_\mathrm{G} - \frac{2\sigma}{R} - 4\,\mu_\ell\frac{\dot{R}}{R} - \frac{4}{3}\,G_\ell\,\frac{R^3 - R_0^3}{R^3}, \tag{3.27}$$

where G_ℓ is the elastic shear modulus. The derivative of the liquid pressure at the bubble wall is given as

$$\dot{p}_\mathrm{L} = \dot{p}_\mathrm{G} + \frac{2\,\sigma\,\dot{R}}{R^2} + 4\,\mu_\ell\frac{\dot{R}^2}{R^2} - 4\,\mu_\ell\,\frac{\ddot{R}}{R} - 4\,G_\ell\,\frac{R_0^3\dot{R}}{R^4}. \tag{3.28}$$

3.3.2.2 Zener model

A more sophisticated viscoelastic model than the Kelvin-Voigt model is the Zener model, also known as standard linear solid model. With the Zener model, the stresses in the medium surrounding the bubble are incorporated in the liquid pressure at the bubble wall as [7]

$$p_\mathrm{L} = p_\mathrm{G} - \frac{2\sigma}{R} + 3\varsigma \tag{3.29}$$

where

$$\varsigma = \int_R^\infty \frac{\tau_{rr}(r,t)}{r}\,\mathrm{d}r \tag{3.30}$$

is an auxiliary variable associated with the rr-component of the viscous stress tensor $\tau_{rr}(r,t)$. The auxiliary stress variable is governed by

$$\lambda_\ell \dot{\varsigma} + \varsigma + \lambda_\ell \frac{\dot{R}}{R}\tau_{rr|R} = -\frac{S}{3}, \tag{3.31}$$

with

$$S = \frac{4}{3}G_\ell\left(1 - \frac{R_0^3}{R^3}\right) + 4\mu_\ell\frac{\dot{R}}{R} \tag{3.32}$$

the combined viscous and elastic contributions, where λ_ℓ is the relaxation time, G_ℓ is the shear modulus and μ_ℓ the viscosity. The stress at the bubble wall, $\tau_{rr|R}$, evolves as

$$\lambda_\ell \dot{\tau}_{rr|R} + \tau_{rr|R} = -S. \tag{3.33}$$

The question is now how to solve the ODEs for ς and τ_{rr} in such a way that we always obtain a meaningful result, even if $\lambda_\ell = 0$. In order for a customary ODE solver to handle this correctly, we need to rearrange Eqs. (3.31) and (3.33). Under the discrete assumption

$$\dot{\varsigma} = \frac{\varsigma_{n+1} - \varsigma_n}{\Delta t}, \tag{3.34}$$

Eq. (3.31) becomes

$$\lambda_\ell \frac{\varsigma_{n+1} - \varsigma_n}{\Delta t} + \varsigma + \lambda_\ell \frac{\dot{R}}{R}\tau_{rr|R} = -\frac{S}{3} \tag{3.35}$$

so that

$$\varsigma_{n+1} = \varsigma_n + \Delta t \frac{-\frac{S}{3} - \lambda_\ell \frac{\dot{R}}{R}\tau_{rr|R,n} - \varsigma_n}{\lambda_\ell + \Delta t}. \tag{3.36}$$

Similarly, Eq. (3.33) follows as

$$\tau_{rr|R,n+1} = \tau_{rr|R,n} + \Delta t \frac{-S - \tau_{rr|R,n}}{\lambda_\ell + \Delta t}. \tag{3.37}$$

Even in the limit $\lambda_\ell = 0$, we can now obtain a meaningful answer, that is Eq. (3.36) reduces to

$$\varsigma = -\frac{S}{3}. \tag{3.38}$$

After inserting Eq. (3.38) into Eq. (3.29) we recover the Kelvin-Voigt model. For $\lambda_\ell = 0$, Eq. (3.37) becomes redundant.

3.3.2.3 Oldroyd-B model

The Oldroyd-B model is a widely used constitutive model for viscoelastic fluids. Following the work of Jiménez-Fernández and Crespo [8], the liquid pressure at the bubble wall including the Oldroyd-B model is given as

$$p_\mathrm{L} = p_\mathrm{G} - \frac{2\sigma}{R} - 4\mu_\ell\frac{\dot{R}}{R} + \mathcal{S}. \tag{3.39}$$

The polymer stress $\mathcal{S} = \mathcal{S}_1 + \mathcal{S}_2$ is split into two constitutive ODEs,

$$\lambda_\ell \dot{\mathcal{S}}_1 + \mathcal{S}_1 + 4\lambda_\ell \frac{\dot{R}}{R}\mathcal{S}_1 = -2\eta_\ell\frac{\dot{R}}{R} \tag{3.40}$$

$$\lambda_\ell \dot{\mathcal{S}}_2 + \mathcal{S}_2 + \lambda_\ell \frac{\dot{R}}{R}\mathcal{S}_2 = -2\eta_\ell\frac{\dot{R}}{R} \tag{3.41}$$

where η_ℓ is the polymer viscosity. These ODEs are reformulated in a similar manner as for the Zener model shown above, to yield

$$\mathcal{S}_{1,n+1} = \mathcal{S}_{1,n} + \Delta t \frac{-\left(4\lambda_\ell \frac{\dot{R}}{R} + 1\right)\mathcal{S}_{1,n} - 2\eta_\ell \frac{\dot{R}}{R}}{\lambda_\ell + \Delta t} \tag{3.42}$$

$$\mathcal{S}_{2,n+1} = \mathcal{S}_{2,n} + \Delta t \frac{-\left(\lambda_\ell \frac{\dot{R}}{R} + 1\right)\mathcal{S}_{2,n} - 2\eta_\ell \frac{\dot{R}}{R}}{\lambda_\ell + \Delta t}. \tag{3.43}$$

For $\lambda_\ell = 0$ Eqs. (3.42) and (3.43) still give a meaningful result and reduce to a Newtonian fluid with $\mathcal{S} = -4\eta_\ell \dot{R}/R$.

3.4 The interface

APECSS readily supports the gas-liquid interface, also often referred to as the *bubble wall*, to be either clean, for which only the surface tension coefficient has to be defined, or coated with a lipid monolayer.

Section	Command	Description
`INTERFACE`	`SurfaceTensionCoeff <float>`	Surface tension coefficient σ_c of the clean interface.
	`LipidCoatingModel None`	No lipid coating model is applied. This is the default.
	`LipidCoatingModel Marmottant`	The lipid coating model of Marmottant *et al.* [13] is applied.
	`LipidCoatingModel Gompertz-Marmottant`	The continuous variant of the lipid coating model of Marmottant proposed by Gümmer *et al.* [6] is applied.
	`SigmaInit <float>`	Initial surface tension coefficient σ_0 of the lipid coating model at R_0.
	`Elasticity <float>`	Elasticity χ of the lipid coating model.
	`DilatationalViscosity <float>`	Dilatational viscosity κ_s of the lipid coating model.

The influence of the surface tension, the rheology of the lipid-monolayer coating and the viscous dissipation in the liquid is accounted for through the definition of the liquid pressure at the bubble wall, given as [13]

$$p_L = p_G - \frac{2\sigma}{R} - 4\,\mu_\ell \frac{\dot{R}}{R} - 4\,\kappa_s \frac{\dot{R}}{R^2}, \tag{3.44}$$

where σ is the surface tension coefficient, μ_ℓ is the dynamic viscosity of the liquid and κ_s is the surface dilatational viscosity of the lipid monolayer.

The clean gas-liquid interface has a surface tension coefficient of $\sigma = \sigma_c$ and a surface dilatational viscosity of $\kappa_s = 0$.

The surface tension coefficient is given by the model introduced by Marmottant *et al.* [13] as

$$\sigma = \begin{cases} 0 & \text{for } R \le R_{buck} \\ \chi\left(\frac{R^2}{R_{buck}^2} - 1\right) & \text{for } R_{buck} < R < R_{rupt} \\ \sigma_c & \text{for } R \ge R_{rupt} \end{cases} \tag{3.45}$$

where χ is the surface elasticity of the lipid monolayer. When the radius of the bubble becomes smaller than [15]

$$R_{\mathrm{buck}} = \frac{R_0}{\sqrt{1+\sigma_0/\chi}}, \tag{3.46}$$

where σ_0 is the surface tension coefficient of the lipid-coated bubble at $R = R_0$, the lipid monolayer cannot compress any further and begins to buckle, as a result of which the surface tension effectively vanishes. In contrast, when the bubble expands to a radius larger than

$$R_{\mathrm{rupt}} = R_{\mathrm{buck}} \sqrt{1+\frac{\sigma_{\mathrm{c}}}{\chi}}, \tag{3.47}$$

the lipid monolayer ruptures and, as a consequence, the clean gas-liquid interface is laid bare.

The radius-dependent surface tension coefficient of the Marmottant model [13], defined in Eq. (3.45), contains two discontinuities at $R = R_{\mathrm{buck}}$ and $R = R_{\mathrm{rupt}}$. These discontinuities render the Marmottant model sensitive to the applied time-step when numerically solving the primary ordinary differential equation [19]. A continuously differentiable form of the Marmottant model a Gompertz function of the form $f(x) = a\,\mathrm{e}^{-b\,\mathrm{e}^{-cx}}$, a special case of the generalized logistics function, was proposed by Gümmer *et al.* [6]. Using this Marmottant-Gompertz model, the surface tension coefficient is defined as

$$\sigma = \sigma_{\mathrm{c}}\,\mathrm{e}^{-b\,\mathrm{e}^{c(1-R/R_{\mathrm{buck}})}}, \tag{3.48}$$

with $a = \sigma_{\mathrm{c}}$ and $x = R/R_{\mathrm{buck}} - 1$, and where the buckling radius R_{buck} is given by Eq. (3.46). The derivative of the surface tension coefficient follows as

$$\dot{\sigma} = \sigma\, b\, c\, \mathrm{e}^{c(1-R/R_{\mathrm{buck}})}\,\frac{\dot{R}}{R}. \tag{3.49}$$

Enforcing σ_0 for R_0, the coefficient b is readily given as

$$b = -\frac{\ln(\sigma_0/\sigma_{\mathrm{c}})}{\mathrm{e}^{c(1-R_0/R_{\mathrm{buck}})}}. \tag{3.50}$$

Assuming, additionally, that the maximum slope of the Gompertz function is equal to the derivative of the surface tension coefficient given by the Marmottant model at $R = R_{\mathrm{buck}}\sqrt{1+\sigma_{\mathrm{c}}/(2\chi)}$, the coefficient c follows as

$$c = \frac{2\chi\mathrm{e}}{\sigma_{\mathrm{c}}}\sqrt{1+\frac{\sigma_{\mathrm{c}}}{2\chi}}. \tag{3.51}$$

The Marmottant-Gompertz model reproduces the main features of the original Marmottant model [6], but with a smooth transition between the surface tension regimes, using the same set of input parameters (σ_0, σ_{c}, χ) as the original Marmottant model.

3.5 Infinity

The pressure at infinity, p_∞ is used to apply a driving pressure difference for the bubble dynamics. Presently, APECSS readily supports a constant ambient pressure $p_\infty = p_0$, which may also be replaced by a pressure defined on-the-fly, or a sinusoidal excitation.

The sinusoidal exictation is defined as $p_\infty = p_0 - \Delta p_{\mathrm{a}}\,\sin(2\pi f_{\mathrm{a}} t)$, where f_{a} and Δp_{a} are the frequency and pressure amplitude of the excitation. In order to use the sinusoidal excitation, the user has to allocate the pointer `*Excitation`, which is part of the structure `APECSS_Bubble`, with `struct APECSS_Excitation` and define the desired values for f_{a} and Δp_{a}. The example found in the directory `example/ultrasound/` provides an example of how to do this.

3.6 Results

The results of the bubble dynamics can be written to disk, if the user wants that. Note that APECSS does not write any results to disk unless it is specifically ask to do so.

Section	Command	Description
`RESULTS`	`Bubble`	Results of the bubble dynamics are written to file.
	`OutputFreqRP <float>`	Results of the bubble dynamics are stored every so many time steps (default: 1).
	`OutputPath <string>`	Path to the directory where all the results should be written in to (default: `./`).
	`OutputDigits <float>`	Results are written out with as many digits (default: 6).

For the bubble dynamics, the following quantities as a function of time are written into a text file, named by the RP model and (if applicable) the excitation parameters used:

- Time-step number
- Time, t
- Time-step, Δt
- Bubble radius, R
- Velocity of the bubble wall, $\dot{R}$
- Pressure of the gas, p_G
- Pressure of the liquid at the bubble wall, p_L
- Speed of sound of the liquid at the bubble wall, c_L, if the Gilmore model is applied.
- The result of any additional user-defined ODE solved, if applicable.

An example of this output is shown in Figure 3.1 for the example `/example/rayleighcollapse/`, executed with the options file `simple.apecss`.

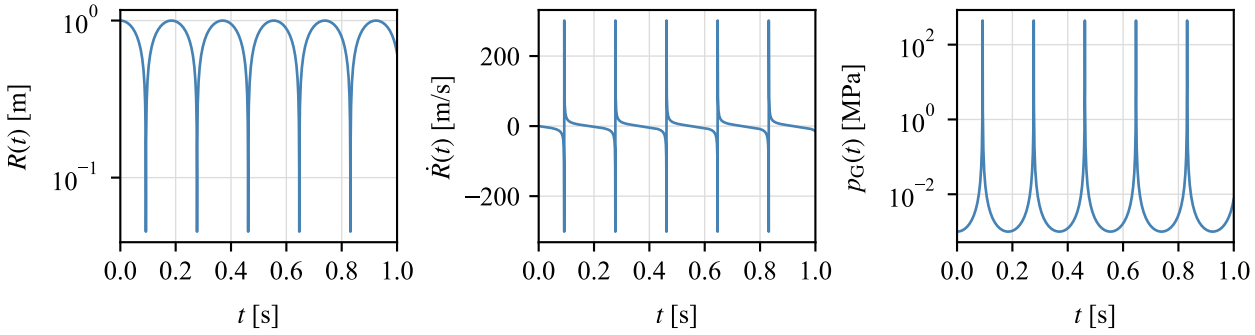


Figure 3.1: Bubble radius R, bubble wall velocity $\dot{R}$ and gas pressure p_G as a function of time t of a bubble undergoing a repeated Rayleigh collapse.

Chapter 4

Acoustic emissions

Section	Command	Description
BUBBLE	`Emissions Incompressible`	Computes the acoustic emissions under the common incompressible assumption.
	`Emissions FTI`	Computes the acoustic emissions under the assumption of an incompressible fluid but propagating the emissions with the speed of sound.
	`Emissions QA`	Computes the acoustic emissions using the quasi-acoustic model of Gilmore [5].
	`Emissions KB`	Computes the acoustic emissions based on the Kirkwood-Bethe hypothesis.
	`KBIterTolerance <float>`	Sets the tolerance for the iterative pressure evaluation when the emissions are computed based on the Kirkwood-Bethe hypothesis.

4.1 Incompressible assumption

4.2 Quasi-acoustic model

4.3 Emissions based on the Kirkwood-Bethe hypothesis

4.4 Results

Bibliography

[1] Brenner, M. P., Hilgenfeldt, S., and Lohse, D. (2002). Single-bubble sonoluminescence. *Reviews of Modern Physics*, **74**(2), 425–484.

[2] Chandran, J. and Salih, A. (2019). A modified equation of state for water for a wide range of pressure and the concept of water shock tube. *Fluid Phase Equilibria*, **483**, 182–188.

[3] Denner, F. (2021). The Gilmore-NASG model to predict single-bubble cavitation in compressible liquids. *Ultrasonics Sonochemistry*, **70**, 105307.

[4] Dormand, J. and Prince, P. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, **6**(1), 19–26.

[5] Gilmore, F. R. (1952). The growth or collapse of a spherical bubble in a viscous compressible liquid. Technical Report 26-4, California Institute of Technology, Pasadena, California, USA.

[6] Gümmer, J., Schenke, S., and Denner, F. (2021). Modelling Lipid-Coated Microbubbles in Focused Ultrasound Applications at Subresonance Frequencies. *Ultrasound in Medicine & Biology*, **47**(10), 2958–2979.

[7] Hua, C. and Johnsen, E. (2013). Nonlinear oscillations following the Rayleigh collapse of a gas bubble in a linear viscoelastic (tissue-like) medium. *Physics of Fluids*, **25**(8), 083101.

[8] Jiménez-Fernández, J. and Crespo, A. (2005). Bubble oscillation and inertial cavitation in viscoelastic fluids. *Ultrasonics*, **43**(8), 643–651.

[9] Keller, J. B. and Miksis, M. (1980). Bubble oscillations of large amplitude. *The Journal of the Acoustical Society of America*, **68**(2), 628–633.

[10] Kontogeorgis, G. M., Privat, R., and Jaubert, J.-N. (2019). Taking Another Look at the van der Waals Equation of State–Almost 150 Years Later. *Journal of Chemical & Engineering Data*, **64**(11), 4619–4637.

[11] Lauterborn, W. and Kurz, T. (2010). Physics of bubble oscillations. *Reports on Progress in Physics*, **73**(10), 106501.

[12] Le Métayer, O. and Saurel, R. (2016). The Noble-Abel Stiffened-Gas equation of state. *Physics of Fluids*, **28**(4), 046102.

[13] Marmottant, P., van der Meer, S., Emmer, M., Versluis, M., de Jong, N., Hilgenfeldt, S., and Lohse, D. (2005). A model for large amplitude oscillations of coated bubbles accounting for buckling and rupture. *The Journal of the Acoustical Society of America*, **118**(6), 3499–3505.

[14] Neppiras, E. A. (1980). Acoustic cavitation. *Physics Reports*, **61**(3), 159–251.

[15] Overvelde, M., Garbin, V., Sijl, J., Dollet, B., de Jong, N., Lohse, D., and Versluis, M. (2010). Nonlinear Shell Behavior of Phospholipid-Coated Microbubbles. *Ultrasound in Medicine & Biology*, **36**(12), 2080–2092.

[16] Prosperetti, A. and Lezzi, A. (1986). Bubble dynamics in a compressible liquid. Part 1. First-order theory. *Journal of Fluid Mechanics*, **168**, 457–478.

[17] Radulescu, M. I. (2020). Compressible flow in a Noble–Abel stiffened gas fluid. *Physics of Fluids*, **32**(5), 056101.

[18] van der Waals, J. (1998). The equation of state for gases and liquids. In S. Lundqvist, editor, *Nobel Lectures in Physics (1901-1921)*, pages 254–265. World Scientific, Singapore.

[19] Versluis, M., Stride, E., Lajoinie, G., Dollet, B., and Segers, T. (2020). Ultrasound Contrast Agent Modeling: A Review. *Ultrasound in Medicine & Biology*, **46**(9), 2117–2144.