Nektar1D Reference Manual

Jordi Alastruey[1], Jorge Aramburu[2], Peter Charlton[1], Weiwei Jin[1], Marie Willemet[1]
[1]Department of Biomedical Engineering, King's College London, UK
[2]Universidad de Navarra, Donostia–San Sebastián, Spain

August 30, 2022

Contents

1 Introduction

Nektar1D is our in-house computer code for solving the nonlinear, one-dimensional (1-D) equations of blood flow in a given network of compliant vessels subject to boundary and initial conditions. Nektar1D computes blood pressure, blood flow and luminal area waves in any point of the arterial network. It is an essential part of our research activities which are described in the group's website: `www.haemod.uk`.

This document describes how to compile Nektar1D (Section 2), create a text file containing all the input data for a specific simulation (Section 3), run the input file (Section 4), and interpret the output files containing the results of the simulation (Section 5). It also provides some examples of Nektar1D simulations used in our peer-reviewed publications (Section 6), and a brief summary of the source code structure (Section 7).

For a review on arterial pulse wave haemodynamics, a description of the 1-D equations, and the numerical scheme used in Nektar1D to solve them we refer to [1]. We have verified the accuracy of the 1-D formulation by comparison against (i) experimental data in a 1:1 scale cardiovascular simulator rig of the aorta and its larger branches made of silicone tubes [2], (ii) *in vivo* data in rabbits [3] and humans [4], and (iii) numerical data obtained by solving the full 3-D equations of blood flow in compliant vessels [4, 5, 6, 7]. In addition, we have developed pulse wave analysis tools to post-process simulated waveforms and understand underlying mechanisms affecting the shape of pulse waveforms [8, 9, 10, 11, 12].

Any comments/improvements on this document and Nektar1D can be sent to `jordi.alastruey-arimon@kcl.ac.uk` .

2 Compiling Nektar1D

The easiest way to compile Nektar1D is on a Ubuntu 64bit Linux operating system (https://ubuntu.com/download/desktop). To compile the code, the `g++` and `gfortran` compilers should be installed (they normally come with the Linux distribution). Otherwise, they can be installed by typing the following commands:

```
sudo apt-get install g++
sudo apt-get install gfortran
```

The code requires the compilation of two libraries in `/Hlib` and `/Veclib` and of the source code in `1DBio/src`. All makefiles assume that the application `make` is available and that the following symbolic links have been created:

```
ln -s ../Makefile Makefile       in nektar/Hlib/Linux
ln -s ../MakeHybrid MakeHybrid   in nektar/Hlib/Linux
ln -s GCC.inc Linux.inc          in nektar/Flags
```

The following tools also need to be installed:

- The Yacc compiler: `sudo apt-get install byacc`
- LAPACK and BLAS libraries for linear algebra operations:

```
sudo apt-get install liblapack-dev
```

Veclib can then be built using the following commands:

```
cd ~/nektar/Veclib
make
```

The file `libvec.a` should have been generated in `/Veclib`. Hlib can be built from the `Hlib/Linux` directory using:

```
cd ../Hlib/Linux
make dbx
make opt
```

The files `libhybridg.a` and `libhybridopt.a` should have been generated in `/Hlib/Linux`. Then, the Veclib library should be copied into the Hlib/Linux directory:

```
cd ../..
cp Veclib/libvec.a Hlib/Linux/.
```

Now Nektar1D can be compiled using:

```
cd ~/nektar/1DBio/Linux
make clean
make dbx
```

Once the compilation is complete, the Nektar1D executable file called `oneDbio` will be located in the folder `nektar/1DBio/Linux`. The file can be executed using the command `./oneDbio` .

It might be convenient to access `./oneDbio` from any directory. This can be achieved by exporting the path of `./oneDbio` into the `.bashrc` file which contains some initialisation commands for the shell, such as the definition of aliases and new environment variables, and the addition of directories to the `PATH` variable. The first step consists of opening the `.bashrc` file in a text editor:

```
vim ~/.bashrc
```

At the end of the file, the following line should be inserted:

```
export PATH=$PATH:/home/username/nektar/1DBio/Linux
```

The command `whoami` can be used to display the username if this is unknown. The following command refreshes the environment variables previously defined:

```
source ~/.bashrc
```

Your executable `oneDbio` should now be accessible from any directory.

3 Creating an Input File

This section describes how to write a text input file containing all the parameters of a specific simulation. Examples of input files can be found in the folder `nektar/examples`. At the bottom of each file you will find the command line required to execute the file and are described in Section 6. A full description on how to run an input file is given in Section 4.

The input file must have extension `.in` (*e.g.* `input_file_name.in`) and contain the following 5 sections in the given order:

1. Parameter list
2. Mesh definition
3. Boundary conditions
4. Initial conditions
5. History points

Below we describe each section and provide examples. Comments can be added to any lines of the input file and should be preceded by the character `#`.

3.1 Parameter List

The parameter list contains several general parameters of the simulation. The first line of the list must contain an integer indicating the total number of parameters in the list, followed by the list of parameters, each one written in a different line – the value of the parameter must be followed by the parameter identifier. The order of the parameters is not important, but their value must precede their identifier. For any simulation, the following parameters must always be included:

`EQTYPE`	Integer number with the value 0 for the nonlinear formulation and 1 for the linear formulation. Note that the linear formulation is NOT as developed as the nonlinear formulation
`DT`	Time step (in s)
`NSTEPS`	Number of time steps
`HISSTEP`	Number of time steps until the next step solution is dumped in the output file `input_file_name.his`
`Rho`	Blood density (in Kg m^{-3})
`Viscosity`	Blood viscosity (in Pa s)
`Alpha`	Velocity profile parameter α in the frictional force per unit length of the momentum equation, as defined in [1] (after Eq. (1); the velocity profile is given by Eq. (3)). `Alpha` = 4/3 for Poiseuille flow; `Alpha` = 1 for inviscid flow

The following parameters are optional:

`INTTYPE`	Time integration order (1, 2 or 3) of the numerical scheme. By default, `INTTYPE` = 2
`IOSTEP`	Number of time steps until the next step solution is dumped in the output file `input_file_name.out`. By default `IOSTEP` = 0 and .out files are not generated.
`Beta`	Constant stiffness parameter $\beta = 4/3\sqrt{\pi}Eh/A_{\mathrm{d}}$ (in Pa m^{-1}) applied to all the arterial segments of the model, where E is the Young's modulus of the arterial wall, h is the wall thickness, and A_{d} is the luminal area at the reference pressure (usually the diastolic pressure) (see Section 3.2.1)
`Gamma`	Constant wall visco-elasticity parameter $\Gamma = 2/3\sqrt{\pi}\varphi h/A_{\mathrm{d}}$ (in Pa s m^{-1}) applied to all the arterial segments of the model, where φ is the wall viscosity, h is the wall thickness, and A_{d} is the luminal area at the reference pressure (usually the diastolic pressure) (see Section 3.2.1). Note that wall visco-elasticity is only available for the nonlinear solution (`EQTYPE` = 0)
`GammaII`	Constant wall visco-elasticity parameter related to Γ by $\Gamma = 1/2\beta$`GammaII` (in seconds), where Γ and β are defined in Section 3.2.1). This parameter is applied to all the arterial segments of the model and is only available for the nonlinear solution (`EQTYPE` = 0). It is used in `nektar/examples/Experimental/exp37.in` (see Section 6)
`Ao`	Constant luminal cross-sectional area A_{d} (in m^2) applied to all the arterial segments of the model
`pinf`	Pressure (in Pa) at the outflow of each lumped parameter model. By default `pinf` = 0
`Pext`	External pressure P_{ext} (in Pa) to the vessel wall (see Eq. (3.2)). By default `Pext` = 0
`Junc_losses`	1: Switch on energy losses at junctions; 0: no energy losses at junctions. By default `Junc_losses` = 0
`Bifurc_losses`	1: Switch on energy losses at bifurcations; 0: no energy losses at bifurcations. By default `Bifurc_losses` = 0
`Ischemic_Loss`	Pressure drop in Pa due to an ischemic attack. It is used in `nektar/examples/CoW/CoW_autoreg.in` to simulate a ischemic attack [13] and in `nektar/examples/FMD/FMD.in` to simulate inflation and deflation of the cuff in the upper forearm [14]
`Ischemic_TCompres`	Time it takes to reduce the pressure when `Ischemic_Loss` is activated
`NO_STN_DOM`	Domain number with the stenosis model. It is used in `nektar/examples/CCA_Stn/CCA_Stn.in` [15]
`ELM_SIZE`	Element size in the domain with the stenosis model. It is used in `nektar/examples/CCA_Stn/CCA_Stn.in` [15]
`STR_STN_ELM`	Domain number where the stenosis model starts. It cannot be the first element of the domain. It is used in `nektar/examples/CCA_Stn/CCA_Stn.in` [15]
`END_STN_ELM`	Domain number where the stenosis model ends. It cannot be the last element of the domain. It is used in `nektar/examples/CCA_Stn/CCA_Stn.in` [15]

`K_T`	Empirical constant of the stenosis model. It is used in `nektar/examples/CCA_Stn/CCA_Stn.in` [15]
`Periodic`	The inflow boundary condition is periodic. The value of the period is the number given in front of `Periodic` in seconds
`T_initial`	Initial time (in s) for the time-average calculations in `input_file_name_periodic.tex`. By default `T_initial` = 0
`T_final`	Final time (in s) for the time-average calculations in `input_file_name_periodic.tex`. By default `T_final` = `NSTEPS` * `DT`
`SkipDomain`	Number of domains (starting from domain 1) to be skipped when producing the Latex files `input_file_name.tex` and `input_file_name_periodic.tex`. By default `SkipDomain` = 0
`SCAL_F`	Scaling factor for the inflow waveform (see Section 3.3.1)
`Pulse_Centre`	Time of the peak (in s) for the Gaussian function/s used as inflow boundary condition/s (see Eq. (3.8)). By default `Pulse_Centre` = 1
`Pulse_Width`	Parameter controlling the width (in s^{-2}) of the Gaussian function/s used as inflow boundary condition/s (see Eq. (3.8)). By default `Pulse_Width` = 100
`FMDHL`	Half-life of cumulative shear exposure of the flow-mediated dilation (FMD) model described in [14]
`FMDCd`	Parameter controlling the time delay in the change in Young's modulus of the FMD model described in [14]
`FMDalp`	Parameter controlling the magnitude in the change in Young's modulus of the FMD model described in [14]
`RVar1`	Parameter controlling the variation in magnitude of the time-varying peripheral resistances of all the terminal Windkessels models coupled to the right arm 1-D model arterial segments of the FMD model. The nomenclature α_1 was used for this parameter in Eq. (6) of [14].
`RVar2`	Parameter controlling the variation in magnitude of the time-varying peripheral resistances of all the terminal Windkessels models coupled to the right arm 1-D model arterial segments of the FMD model. The nomenclature α_2 was used for this parameter in Eq. (6) of [14].
`RefT`	Time when the reference values were calculated in the FMD model described in [14].
`CarCyc`	Length of one cardiac cycle in the FMD model described in [14].

Additional parameters can be defined by the user; *e.g.* `ELASTIC` is used in `nektar/examples/Experimental/exp37.in` to define a constant Young's modulus for all arterial segments. (This model produces the results for the purely elastic case described in [2].) Note that the number π is defined as `PI` by default throughout any input file.

3.2 Mesh Definition

The geometrical and mechanical properties of all arterial segments – which are called 'domains' in Nektar1D – are defined in this section. The first line of the section must contain the string `Mesh` followed by `Ndomains =` *integer*, where the integer indicates the

number of domains present. If there is only one domain, then `Ndomains =` *integer* is not necessary.

Each domain definition is started with an opening line containing the number of finite elements (Nel) that make up the domain. For each element, a line is required with four numbers indicating the (i) lower spatial coordinate x; (ii) upper spatial coordinate x; (iii) polynomial order of the element, p; and (iv) quadrature order of the element, q. For example, a domain with first point $x = -0.075$ m and last point $x = 0.075$ m which is divided into three equispaced elements with a quadrature and polynomial order of 6 is defined as:

```
Mesh
3       # Nel
-0.075  -0.025  6  6  # x_lower x_upper p q
-0.025  0.025   6  6  # x_lower x_upper p q
0.025   0.075   6  6  # x_lower x_upper p q
```

Material and geometrical properties can be defined for each element of each domain in *Mesh Definition*, as detailed below for a purely elastic (Section 3.2.2) and visco-elastic (Section 3.2.3) arterial wall. If all domains have the same properties, their values can be defined in *Parameter List* (see Section 3.1 for more details). First, we present the most generic tube law – relating changes in blood pressure to changes in cross-sectional area – currently available in Nektar1D (Section 3.2.1).

3.2.1 Tube Law

The most generic relationship between changes in blood pressure and changes in cross-sectional area implemented in Nektar1D is a Voigt-type visco-elastic tube law given by

$$P = P_{\mathrm{e}}(A; x) + \frac{\Gamma}{\sqrt{A}} \frac{\partial A}{\partial t}, \tag{3.1}$$

with

$$P_{\mathrm{e}}(A, x) = P_{\mathrm{ext}} + \beta \left(\sqrt{A} - \sqrt{A_{\mathrm{d}}} \right), \tag{3.2}$$

$$\beta(x) = \frac{4}{3} \frac{\sqrt{\pi}\, Eh}{A_{\mathrm{d}}}, \tag{3.3}$$

$$\Gamma(x) = \frac{2}{3} \frac{\sqrt{\pi}\, \varphi h}{A_{\mathrm{d}}} = \frac{4}{3} \frac{\varphi h}{D_{\mathrm{d}} \sqrt{A_{\mathrm{d}}}}, \tag{3.4}$$

where $P(x,t)$ is blood pressure, $P_{\mathrm{e}}(x,t)$ is the elastic component of pressure, P_{ext} is the external pressure, $A(x,t)$ is the luminal cross-sectional area, $h(x)$ is the wall thickness, $E(x)$ is the Young's modulus, and $\varphi(x)$ is the wall viscosity. The reference area $A_{\mathrm{d}}(x)$ and diameter $D_{\mathrm{d}}(x)$ are the area and diameter, respectively, at $P = P_{\mathrm{ext}}$ and $\frac{\partial A}{\partial t} = 0$, with P_{ext} usually taken as the diastolic pressure. Note that the parameter $\beta(x)$ is related to the elasticity of the wall and $\Gamma(x)$ to the viscosity of the wall. Moreover, both $\beta(x)$ and $\Gamma(x)$ are independent of the transmural pressure $P - P_{\mathrm{ext}}$.

3.2.2 Purely-Elastic Simulations

By default $\Gamma = 0$; *i.e.* all domains have a purely-elastic arterial wall (see Eq. (3.1)). The mechanical properties for a purely-elastic arterial wall can be specified in three different ways; *i.e.* using:

- The stiffness parameter β;
- The product of wall Young's modulus and wall thickness, Eh;
- An empirical law relating the pulse wave velocity c and the reference diameter D_{d}.

The stiffness parameter β
For each domain, the opening line must contain the strings `Beta` and `Area`, as shown in the example below. For each element, β and A_{d} are defined in two different lines, starting with '`Beta =`' and '`Area =`' (or `Ao =`), respectively. (The order matters; first `Beta` and then `Area`.) Elements can have either constant β and A_{d} or varying β and A_{d} as a function of the axial coordinate x along the domain. For example,

```
2  # Nel domain 5  Beta Area
0.0  0.0175  6  6  # x_lower x_upper p q
Beta = 404.063553/(3.2000e-03 + -2.8571e-02*x)/(2.7628e-03 + -2.1984e-02*x)
Area = 4.7022e-06 + -6.7010e-05*x + 2.3874e-04*x*x
0.0175  0.035  6  6  # x_lower x_upper p q
Beta = 404.063553/(3.2000e-03 + -2.8571e-02*x)/(2.7628e-03 + -2.1984e-02*x)
Area = 4.7022e-06 + -6.7010e-05*x + 2.3874e-04*x*x
```

This example was taken from `nektar/examples/Rabbit/Rabbit.in`, which is the model of the rabbit systemic circulation used in [3].

Note that it is possible to define a constant β in *Parameter List* and a variable A_{d} in *Mesh Definition*, and viceversa.

Wall Young's modulus times wall thickness, Eh
We can prescribe the quantity Eh, with $E(x)$ the Young's modulus of the arterial wall and $h(x)$ the wall thickness. The value of $\beta(x)$ is then computed by Nektar1D using Eq. (3.3). For each domain, the opening line must contain the strings `Eh` and `Area`. For each element, Eh is then defined in a new line starting with '`Eh =`'. Note that A_{d} must be defined before Eh. For example:

```
2  nel  Eh Area
0.0  0.120685834705770  5  5  # x_lower x_upper p q
Area = 4.5239e-04
Eh = 480
0.120685834705770  0.241371669411541  5  5  # x_lower x_upper p q
Area = 4.5239e-04
Eh = 480
```

This example was taken from `nektar/examples/Aorta/Ao_Eh.in`, which is the single-vessel model of the upper thoracic aorta used in [5, 7]. The material properties in this model can also be prescribed using the stiffness parameter β as shown in `nektar/examples/Aorta/Ao.in`. Another example in which Eh is specified is `nektar/examples/Experimental/exp37.in`. In this case a constant Young's modulus (defined in *Parameter List* as `ELASTIC`) is used for all arterial segments [2].

Empirical law
Material properties can be prescribed through the local pulse wave velocity, $c(x,t)$, since this is directly related to $\beta(x)$ through $A(x,t)$,

$$\beta = \frac{2\,\rho\,c^2}{\sqrt{A}}. \tag{3.5}$$

Pulse wave velocities can be calculated using the following empirical relationship [16],

$$c = \frac{a}{(D_{\rm d})^b}, \tag{3.6}$$

where $D_{\rm d}$ is the luminal diameter (expressed in mm) at the reference pressure, and a and $b = 0.3$ are empirical coefficients. The value of β is computed by Nektar1D as

$$\beta = \frac{2\,\rho}{\sqrt{A_{\rm d}}}\,\frac{a^2}{(D_{\rm d})^{2b}} \tag{3.7}$$

with $D_{\rm d} = \sqrt{\frac{4A_{\rm d}}{\pi}}$ expressed in mm.
For each domain, the opening line must contain the strings `Empirical_I` and `Area`. For each element, a is defined in a line starting with '`a =`'. Note that the reference area $A_{\rm d}$ must be defined before a. For example:

```
1  nel domain 3  Empirical_I Area
0.0  0.023  3  3  # x_lower x_upper p q
Area = (4.94808594E-04 - 1.88563815E-03*x + 1.79646802E-03*x*x)
a = CSCAL_a*11.0
```

This example was taken from `nektar/examples/55art/55art_elas.in`, which corresponds to a model of the 55 larger systemic arteries [1]. In this example, `CSCAL_a` is defined by the user in *Parameter List*.

3.2.3 Visco-Elastic Simulations

To define a domain with a visco-elastic arterial wall we must provide the visco-elastic parameter Γ using the string `Gamma` in the opening line of the domain definition. Then, Γ can be defined as a function of the axial coordinate x along the domain. If the stiffness parameter β is used for the purely-elastic part, then `Gamma =` must be located between the lines `Beta =` and `Area =` . For example:

```
1  nel  Beta Area Gamma
0.0  0.126  5  5  # x_lower x_upper p q
Beta = 1.7553E+07
Gamma = 1.8806E+05
Ao = 2.8274e-05
```

This example was taken from `nektar/examples/CCA/CCA_Beta_vw_mesh.in` which corresponds to a single-vessel model of the common carotid artery [5, 7].

If the wall Young's modulus times wall thickness, Eh, is used for the purely-elastic part, then `Gamma =` must be located after the lines `Area =` and `Eh =` . For example:

```
1  nel  Eh Area Gamma
0.0  0.126  5  5  # x_lower x_upper p q
Ao = 2.8274e-05
Eh = 3E-4*700E3
Gamma = 1.8806E+05
```

This example was taken from `nektar/examples/CCA/CCA_vw_mesh.in` which corresponds to a single-vessel model of the common carotid artery [5, 7].

If the empirical law is used for the purely-elastic part, then `Gamma =` must be located after the lines `Area =` and `a =` . For example:

```
1  nel domain 3  Empirical_I Area Gamma
0.0  0.023  3  3  # x_lower x_upper p q
Area = (4.94808594E-04 - 1.88563815E-03*x + 1.79646802E-03*x*x)
a = CSCAL_a*11.0
Gamma = 4/3*Varphi4*hD/(CSCAL_Ao*(4.94808594E-04 - 1.88563815E-03*x
    + 1.79646802E-03*x*x))^0.5
```

This example was taken from `nektar/examples/55art/55art.in` which corresponds to a model of the 55 larger systemic arteries [1]. In this example, `CSCAL_a` (a scaling factor for a in Eq. (3.7)), `Varphi4` (the value of φ in Eq. (3.4)), and `hD` (the value of the ratio h/D_{d} in Eq. (3.4)) are defined by the user in *Parameter List*.

3.3 Boundary Conditions

This section defines the boundary conditions (BCs) for all the domains (arterial segments) that can be prescribed in Nektar1D. The first line of this section must contain the string `Boundary`. Then, for each domain, at least four lines with BC information must be included: two with information on the BC at the inlet of the domain, and two with information at the outlet. Each line must start with a letter defining the type of BC. There are three types of BCs that can be prescribed at the inlet and outlet of each domain:

1. *Prescribed waveform* (usually at the inlet of the arterial network);
2. *Lumped parameter (0-D) model* (usually at the outlets of terminal branches);
3. *Connection with other domains.*

3.3.1 Prescribed Waveform

A blood flow, blood velocity, or blood pressure waveform can be prescribed as BC. Incoming waves can be treated in two different ways: they can be reflected back into the domain (*reflective BC*) or they can be fully absorbed by the BC (*absorbing BC*). Moreover, a pulse waveform can be defined in a text file by (i) providing the amplitude and phase angle of all its Fourier harmonics (for flow rate waves only) or (ii) the magnitude of the flow, velocity or pressure for each time step. The file must be named `input_file_name_IN.bcs` if there is only one domain or `input_file_name_IN_1.bcs` if there are multiple domains. If the harmonics definition is used, the first line in the

`.bcs` file must contain the following three values separated by a space: number of harmonics, cardiac cycle duration (in seconds), and mean blood flow rate (in m^3/s). This must be followed by a line for each harmonic containing the values of its amplitude and phase angle separated by a space. Alternatively, a waveform can be defined as an algebraic function of time in the input file `input_file_name.in`, using `t` to denote time. `AorticFlowWave` can be used to create aortic inflow waveforms under a range of conditions. The source code for this Matlab script is available from here.

The following table lists the Nektar1D commands for imposing a flow, velocity or pressure waveform as BC, in either a reflective or an absorbing way, and using either a text file .bcs or an algebraic equation in the input file.

		Reflective BC	Absorbing BC
Flow, q	File .bcs (time - flow rate)	`q 2`	`q 3`
	File .bcs (harmonics)	`F 0`	`F 8`
	Algebraic function	`q 0`	`q 1`
Velocity, u	File .bcs (time - velocity)	`u 2`	`u 3`
	Algebraic function	`u 0`	`u 1`
Pressure, p	File .bcs (time - pressure)	`p 2`	`p 3`
	Algebraic function	`p 0`	`p 1`

For example, to prescribe an *in vivo* flow waveform expressed as a sum of harmonics in a reflective way, the commands at the inlet (or outlet) of the domain are

```
F 0
F 0
```

This type of infow boundary condition must be accompanied by a text file called `input_file_name_IN.bcs` or `input_file_name_IN_1.bcs` with the harmonics information described above (`input_file_name_OUT.bcs` or `input_file_name_OUT_1.bcs` if the BC is prescribed at the outlet of the domain). See `nektar/examples/AoBif/AoBif.in` for an example of a model that uses this type of reflective BC and `nektar/examples/AoBif/AoBif_abs.in` for the equivalent model with an absorbent BC.

For and example on a flow waveform expressed as a time–flow rate text file .bcs, see `nektar/examples/Adan56/adan77.in` .

To prescribe a cosine pressure wave at the inlet of a domain, in a reflective way and using an algebraic function, the following commands must be used in the *Boundary conditions* section of the input file:

```
p 0
p = cos(PI*t)
p 0
p = 0
```

This example was taken from `nektar/examples/Sine/Sine.in` . An example with a prescribed velocity waveform, in a reflective way and using an algebraic function, is given by `nektar/examples/Sine/Sine_vw.in` .

A Gaussian inflow waveform described as

$$Q = ae^{-(t-b)^2/c} \tag{3.8}$$

can be prescribed using `F 7`. The constant parameters a, b and c must be specified in the parameter list using the identifiers `SCAL_F`, `Pulse_Centre` and `Pulse_Width`, respectively. Examples are provided in the folder `nektar/examples/Pulse/` .

Lastly, the following inflow waveforms have been hard-coded in Nektar1D as reflective BCs:

`F 1` – Flow waveform at the aortic root of the *in vitro* model described in [2] and used in `nektar/examples/Experimental/exp37.in` ;

`F 2` – Flow waveform at the rabbit aortic root as described in [3] and used in `nektar/examples/Rabbit/Rabbit.in` ;

`F 3` – Flow waveform at the human aortic root as described in [1] and used in `nektar/examples/55art/55art.in` ;

`F 4` – Flow waveform at the human aortic root as described in [17] and used in `nektar/examples/CoW/CoW.in` ;

`F 6` – Flow waveform at the human common carotid artery as described in [5, 7] and used in `nektar/examples/CCA/CCA.in` ;

3.3.2 Lumped Parameter Model

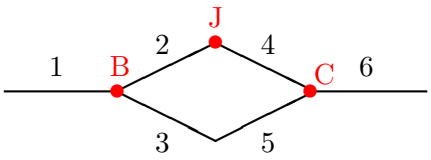
The main types of outlet BCs in Nektar1D are summarised hereunder, together with the two lines of commands that are required in the *Boundary condition* section of the input file. Further details on this type of BCs are provided in [18].

Reflection coefficient	`T` #	if # = 0, complete absorption of the incoming wave
	`T` #	if # = 1, complete reflection of the incoming wave
Single resistance (R)	`R` #	# is the vascular resistance value in [Pa s m^{-3}]
	`R` #	
2-element windkessel (C-R)	`w` $\#_C$	$\#_C$ is the vascular compliance value in [m^3 Pa^{-1}]
	`w` $\#_R$	$\#_R$ is the vascular resistance value in [Pa s m^{-3}]
3-element windkessel (R_1-C-R_2)	`W` $\#_C$	$\#_C$ is the vascular compliance value
	`W` $\#_R$	$\#_R$ equals the sum of vascular resistances $R_1 + R_2$
4-element windkessel (R_1-C-L-R_2)	`Z` $\#_C$	$\#_C$ is the vascular compliance value
	`Z` $\#_R$	$\#_R$ equals the sum of vascular resistances $R_1 + R_2$
		The inductance value is specified in *Parameter List* as `inductance`

For the 3-element windkessel BC, the value of the first resistance (R_1) is by default computed as the characteristic impedance of the end point of the domain (Z_0). If a numeric value is specified after the total resistance value (*i.e.* `W` $\#_R$ $\#_r$), the value of the first resistance R_1 will be multiplied by the absolute value of this factor: $R_1 = |r| * Z_0$. By default, $r = 1$.

3.3.3 Connection with Other Domains

Several types of domain connections can be defined in Nektar1D. The commands for each type are described below and illustrated using arterial network examples.



- **J: Connection between two domains:** At the outlet of Domain 2 the following two lines of commands must be written to indicate that Domain 2 is connected to Domain 4:
  `J 4 4`
  `J 4 4`

- **B: Splitting Flow Bifurcation:** At the outlet of Domain 1 the following two lines of commands must be written to indicate that Domain 1 is connected to the daughter Domains 2 and 3:
  `B 2 3`
  `B 2 3`
  Similarly, at the inlet of Domain 3 the following two lines of commands must be written to specify the number of the parent domain (Domain 1) and the other daughter domain (Domain 2):
  `B 1 2`
  `B 1 2`

- **C: Merging Flow Bifurcation:** At the outlet of Domain 4, the following two lines of commands must be written to indicate that Domain 4 is connected to Domains 5 and 6:
  `C 5 6`
  `C 5 6`
  Similarly, at the inlet of Domain 6, the commands are:
  `C 4 5`
  `C 4 5`

3.4 Initial Conditions

This section contains the initial values for luminal cross-sectional area (A_0) and blood velocity (U_0) in all the domains of the simulation. The first line of the section must contain the string `Initial condition`. Next, two lines must be included for each domain: the first provides the value of A_0 and the second the value of U_0. Both A_0 and U_0 can either be constant values or functions of x. For example,

```
Initial condition
a = PI*1E-4          #Initial value of area (in m^2)
u = 0                #Initial value of flow velocity (in m/s)
```

The initial area A_0 can be prescribed to be equal to the area A_d defined in *Mesh Definition* by using `a = Ao`. For example,

```
Initial condition
a = Ao               #Initial value of area equal to the area in Mesh Definition (in m^2)
u = 0                #Initial value of flow velocity (in m/s)
```

However, the initial area is usually computed as the area that yields the prescribed area A_d in *Mesh Definition* at a given pressure P_d. This is achieved by replacing $P_\mathrm{e} = 0$, $P_\mathrm{ext} = P_\mathrm{d}$, and $A = A_0$ in Eq. (3.2), which yields

$$A_0 = \left(\sqrt{A_\mathrm{d}} - \frac{P_\mathrm{d}}{\beta} \right)^2 . \tag{3.9}$$

Nektar1D can calculate A_0 using Eq. (3.9) if the simulation is launched using the flag `-i`, followed by a positive real number with the value of the pressure P_d (in Pa). For example,

```
./oneDbio -i 12666.66 55art.in
```

Using the flag `-i` overwrites all values of A_0 in *Initial Conditions*. Note, however, that the section *Initial Conditions* must still be included in the input file. This example is taken from `nektar/examples/55art/55art.in` [1].

3.5 History Points

This section specifies the points in the arterial network – called *history points* – where the computed haemodynamic waveforms are dumped. The first line must contain the string `History` and the second must contain the number of domains with history points. Next, for each domain with history points, two lines must be included. The first must have an integer with the number of history points within the domain, followed by another integer with the number of the domain. The second must contain the x position of each history point within the domain. The text below is an example on how to specify history points at the inlet of Domain 1 and at three points in Domain 17. It is taken from `nektar/examples/Rabbit/Rabbit.in`.

```
History Pts
2                    #Number of domains with history points
1 1                  #Number of points and domain identifier
0.0
3 17                 #Number of points and domain identifier
0.0433 0.065 0.0866
```

4 Running an Input File

To execute Nektar1D simply open a terminal window, change to the directory `nektar/1DBio/Linux` and type `./oneDbio` followed by the name and extension of the input file; *e.g.*

```
./oneDbio input_file_name.in
```

To execute Nektar1D from the directory that contains the input file, just export the path of `./oneDbio` to the shell, as described at the end of Section 2. All output files (see Section 5) will be dumped in the directory from which `./oneDbio` is executed.

Please note that an input file transferred from a DOS-platform into a Linux-based system might create errors at the execution. This is because both systems have different character encodings and end-of-line commands. The `fromdos` command converts text files between DOS and UNIX formats. You should be able to install it using the command:

```
sudo apt-get install tofrodos
```

`fromdos` should be used once to convert an input file: `fromdos input_file_name.in` .

The following optional flags are available after the command `./oneDbio`:

1. `-p #`: Assigns the value given in `#` as the polynomial order for all the elements in the simulation. This flag replaces any values given in the input file.

2. `-q #`: Assigns the value given in `#` as the quadrature order of all the elements in the simulation. This flag replaces any values given in the input file.

3. `-N #`: Divides all domains into the number of equispaced elements given in `#`. This flag replaces the number of elements given in the input file.

4. `-O`: Generates output files (with extension `.out`) containing the variables of the simulation evaluated at the quadrature points of all the domains for different times. See Section 5 for more details.

5. `-L`: Generates output files (with extension `.lum`) containing the variables of the lumped parameter models which change with time. See Section 5 for more details.

6. `-A`: Dumps the luminal cross-sectional area, A – instead of the blood pressure, P – in the output file with extension `.out` .

7. `-B`: Generates output files (with extension `.bcs`) containing time-varying characteristic information (including the Riemann variables) at all the boundaries of the arterial network. See Section 5 for more details.

8. `-a`: Dumps the following additional variables in the history files: (i) forward- and backward-travelling Riemann (or characteristic) variables (in m s^{-1}); (ii) spatial-averaged blood pressure (Pa), blood flow velocity (m s^{-1}), blood flow rate (m s^{-3}), and luminal cross sectional area (m^2) across each domain; (iii) forward and backward-travelling components of pressure (Pa) and velocity (m s^{-1}); and (iv) space derivatives: pressure gradient term (Pa m^{-1}), convective acceleration term (m s^{-2}), flow gradient term ($m^2 s^{-1}$). These additional variables were used for the pulse wave analysis tools described in Willemet *et al.* [12].

   Without this flag, the default variables are: time (s), blood pressure (Pa), blood flow velocity (m s^{-1}), blood flow rate (m s^{-3}), luminal cross sectional area (m^2), and an integer that refers to the point label indicated in the heading (see Section 5).

9. `-d`: Dumps a report file called `input_file_name.txt` containing several parameters of the simulation, including haemodynamic variables at the initial time. The file is generated without running the simulation.

10. `-t`: Dumps a LaTeX report file called `input_file_name.tex` containing several parameters of the simulation, including haemodynamic variables at the initial time. The file is generated without running the simulation.

11. `-R`: Dumps space-averaged variables for the whole arterial network in a file called `input_file_name.avg`. See Section 5 for more details.

12. `-r #`: Scales all peripheral resistances by multiplying them by the value specified in `#`.

13. `-c #`: Scales all peripheral compliances by multiplying them by the value specified in `#`.

14. `-i #`: Sets the initial areas (A_0) in all elements to the values that will produce the areas A_d (specified in *Mesh Definition*) at the pressure given by the value specified in `#`. Equation (3.9) is used to calculate A_0.

The first four flags were used in `nektar/examples/Pulse/.` . The last four flags were used in `nektar/examples/55art/55art.in` .

5 Output Files

By default, Nektar1D generates the following output text files:

1. *LaTeX report file*: It is called `input_file_name.tex` and, when compiled in LaTeX, displays several tabulated parameters of the simulation and haemodynamic variables at the initial time.

2. *Report file*: It is called `input_file_name.txt` and contains similar information to the previous file in a format that can be read by a normal text editor.

3. *Period LaTeX report file*: It is called `input_file_name_period.tex` and, when compiled in LaTeX, displays several tabulated parameters of the simulation and haemodynamic variables for the time period starting at `T_initial` and ending at `T_final`, with `T_initial` and `T_final` defined in *Parameter List* (Section 3.1).

4. *Property file*: It is called `input_file_name.prp` and contains the following information for each domain: length, inlet radius, outlet radius, inlet wave speed, outlet wave speed, inlet area, outlet area, inlet Γ, and outlet Γ. Radii, wave speeds, and areas are given for the initial time.

5. *Period property file*: It is called `input_file_name_period.prp` and contains the following information for each domain: length, inlet, midpoint and outlet radii, inlet, midpoint and outlet wave speeds, inlet, midpoint and outlet areas, and arterial compliance for the time period starting at `T_initial` and ending at `T_final`, with `T_initial` and `T_final` defined in *Parameter List* (Section 3.1).

6. *Stiffness parameter file*: A single file called `input_file_name.bet` that contains, for each domain, the domain number, the number of history points, and the value of the stiffness parameter β at the x position of each history point (each value in a different line). Domains without a history point are assigned a β value of zero.

7. *History file/s*: They are called `input_file_name.his` if the model consists of a single domain or `input_file_name_#.his` if the model consists of multiple domains, with `#` the number of each domain with history points defined in the *History Points* section of the input file (see Section 3.5). Each history file consists of a header with information on the number of history points in the domain and their x location. After the header there is a matrix of numbers with the following information:

time (in s) in the first column; blood pressure (Pa) in the second; blood flow velocity (m s^{-1}) in the third; blood flow rate (m s^{-3}) in the fourth; luminal cross sectional area (m^2) in the fifth; and an integer in the sixth referring to the point label indicated in the heading. For a visco-elastic tube law, the elastic component of pressure, P_e (see Eq. (3.2)), is dumped in the third column (in Pa), followed by flow velocity, flow rate, etc. The temporal spacing of these quantities is defined by `HISSTEP` in *Parameter List* (Section 3.1). History files can be converted into Matlab format, ready for analysis, using `ConvertHistoryFiles` which can be dowloaded from here.

8. *History points location*: A single file called `input_file_name.loc` that, for each domain, contains the domain number, the number of history points, and the x position of each history point (each value in a different line).

In addition, the following output text files can be generated using the flags described in Section 4:

1. *Output file/s*: Files containing the variables of the simulation evaluated at the quadrature points of all the domains for different times. They are created using the flag `-O`. They are called `input_file_name.out` if the model consists of a single domain or `input_file_name_#.out` if the model consists of multiple domains, with `#` the number of each domain. Each output file consists of a header with information on the number of elements, points dumped (total and for each element) and time steps dumped. The number of time steps dumped is defined by `IOSTEP` in *Parameter List* (Section 3.1). For each time step, there is a matrix of numbers with the following information: x location (m) in the first column; blood pressure (Pa) in the second; blood flow velocity (m s^{-1}) in the third; forward characteristic (m s^{-1}) in the fourth; and backward characteristic (m s^{-1}) in the fifth. The flag `-A` (in addition to `-O`) dumps the luminal cross-sectional area (A), instead of blood pressure (P), in the second column.

2. *Lumped parameters file/s*: They are obtained using the flag `-L`. We can have a single file called `input_file_name_out.lum`, if there is only one domain with an outflow boundary condition, or multiple files called `input_file_name_out_#.lum`, if there are multiple domains with outflow boundary conditions, with `#` the number of each terminal domain. Each file contains a matrix of numbers with the following information: time (s) in the first column; blood pressure (Pa) at the inflow of the lumped parameter model in the second; blood flow rate (m s^{-3}) at the inflow of the lumped parameter model in the third; blood pressure (Pa) at the compliance in the fourth; and blood flow rate (m s^{-3}) at the outflow of the lumped parameter model in the fifth.

3. *Inflow characteristic information file/s*: They are obtained using the flag `-B`. We can have a single file called `input_file_name_IN.bcs` in a model with only one domain or multiple files called `input_file_name_IN_#.bcs` in a model with multiple domains, with `#` the number of each domain with an inflow boundary condition. Each file contains a matrix of numbers with the following information calculated at the first point of the domain: time (s) in the first column; $\rho c W_f$ (Pa) in the second, with ρ blood density, c pulse wave velocity, and W_f the forward characteristic variable; $-\rho c W_b$ (Pa) in the third, with W_b the backward characteristic

variable; $W_f/2$ (m s^{-1}) in the fourth; $W_b/2$ (m s^{-1}) in the fifth; and $-W_b/W_f$ in the sixth. These information was used in [8] to study the effect of inflow boundary conditions on the shape of the pressure and flow waveforms.

4. *Outflow characteristic information file/s*: They are obtained using the flag `-B`. It is called `input_file_name_OUT.bcs` in a model with only one domain or `input_file_name_OUT_#.bcs` in a model with multiple domains, with `#` the number of each domain with an outflow boundary condition. Each file contains a matrix of numbers with the following information calculated at the last point of the domain: time (s) in the first column; $\rho c W_f$ (Pa) in the second, with ρ blood density, c pulse wave velocity, and W_f the forward characteristic variable; $-\rho c W_b$ (Pa) in the third, with W_b the backward characteristic variable; $W_f/2$ (m s^{-1}) in the fourth; $W_b/2$ (m s^{-1}) in the fifth; and $-W_b/W_f$ in the sixth. These information was used in [8] to study the effect of outflow boundary conditions on the shape of the pressure and flow waveforms.

5. *Average parameters file*: It is a single file obtained using the flag `-R`. It is called `input_file_name.avg` and contains the space-average information described in the header and in [9].

6 Examples

We provide input files for the following examples of 1-D simulations described in our peer-reviewed publications. These can be found in the folder `nektar/examples/`. At the end of each file there is the command line required to execute the file.

`55art/` – Model of the 55 larger systemic arteries in the human under normal physiological conditions [1]. Two input files are provided: cd55art_elas.in simulates the arterial wall as a purely-elastic material and `55art.in` as a visco-elastic material.

`116art/` – Models of the 116 larger systemic arteries in the human under normal physiological conditions for the 25 and 65 year-old baseline virtual subjects described in [19]. The input files are called `116art_25yo.in` and `116art_65yo.in`, respectively.

`Adan56/` – Model of the 56 larger systemic arteries in the human described in [5]. The input file is called `adan77.in` (77 indicates the number of domains used in the Nektar1D simulation).

`AoBif/` – Model of the human aortic bifurcation described in [5, 7]. The following cases are provided: `AoBif.in` (reflective boundary condition); `AoBif_abs.in` (absorbent boundary condition); and `AoBif_vw.in` (visco-elastic arterial wall case; reflective boundary condition).

`Aorta/` – Model of the human upper thoracic aorta described in [5, 7]. The following cases are provided: `Ao.in` (mechanical properties described using the stiffness parameter β); `Ao_Eh.in` (mechanical properties described using the stiffness parameter Eh); `Ao_vw.in` (visco-elastic arterial wall case; mechanical properties described using the stiffness parameter Eh); and `Ao_Beta_vw.in` (visco-elastic arterial wall case; mechanical properties described using the stiffness parameter Eh).

`CCA/` – Model of the human common carotid artery described in [5, 7]. The following cases are provided: `CCA.in` (purely-elastic arterial wall case; mechanical properties described using the stiffness parameter Eh); `CCA_Beta.in` (purely-elastic arterial wall case; mechanical properties described using the stiffness parameter β); `CCA_vw.in` (visco-elastic arterial wall case; mechanical properties described using the stiffness parameter β); `CCA_vw_mesh.in` (visco-elastic arterial wall case with Γ described in *Mesh Definition*; mechanical properties described using the stiffness parameter Eh); `CCA_Beta_vw.in` (visco-elastic arterial wall case; mechanical properties described using the stiffness parameter β); and `CCA_Beta_vw_mesh.in` (visco-elastic arterial wall case with Γ described in *Mesh Definition*; mechanical properties described using the stiffness parameter β).

`CCA_Stn/` – Model of the human common carotid artery with a stenosis described in [15].

`CoW/` – Model of the upper thoracic aorta and larger arteries of the upper body, including the circle of Willis. The following cases are provided: `CoW.in` (purely elastic arterial wall model published in [10]); and `CoW_AutoReg.in` (model with cerebral autoregulation described in [13]).

`Experimental/` – 37-artery network simulating blood flow in the cardiovascular simulator rig described in [2, 5]. The following cases are provided: `exp37.in` (purely elastic arterial wall case) and `exp37_vw.in` (visco-elastic arterial wall case).

`FMD/` – Model of the 116 larger systemic arteries in the human coupled to a flow-mediated dilation model, as described in [14]. The input file is called `FMD.in` .

`Pulse/` – Single pulse propagation in a straight reflection-free vessel as described in [5, 10]. The following cases are provided: `Pulse.in` (inviscid fluid and inviscid wall); `Pulse_sym.in` (inviscid fluid and inviscid wall, with the pulse wave propagated from the outlet); `Pulse_vf.in` (viscous fluid and inviscid wall); `Pulse_vf_sym.in` (viscous fluid and inviscid wall, with the pulse wave propagated from the outlet); `Pulse_vw.in` (viscous wall and inviscid fluid); `Pulse_vw_Mesh.in` (viscous wall and inviscid fluid, with geometrical and material properties defined in 'Mesh definition'); `Pulse_vw_sym.in` (viscous wall and inviscid fluid, with the pulse wave propagated from the outlet); `Pulses.in` (inviscid fluid and inviscid wall with a pulse prescribed at the inlet and another at the outlet) and `Pulses_vw.in` (inviscid fluid and viscous wall with a pulse prescribed at the inlet and another at the outlet).

`Rabbit/` – Model of the 59 larger systemic arteries in the rabbit described in [3]. The input file is called `Rabbit.in` .

`Sine/` – Propagation of a single frequency, sinusoidal wave in a straight reflection-free vessel for which an analytical solution exists [10]. The following cases are provided: `Sine.in` (inviscid fluid and inviscid wall); `Sine_vf.in` (viscous fluid and inviscid wall); and `Sine_vw.in` (viscous wall and inviscid fluid).

7 Source Code Structure

The main functions of the code are located in `nektar/1DBio/src/main.C` and the input file is read by functions in `nektar/1DBio/src/setup.C` . The header files are located in `nektar/1DBio/include` and the external libraries in `nektar/Hlib` and `nektar/Veclib`.

References

[1] Alastruey J, Parker K, Sherwin S. *Arterial pulse wave haemodynamics. In Anderson (Ed.) 11th International Conference on Pressure Surges*, chap. 7. Virtual PiE Led t/a BHR Group (ISBN: 978 1 85598 133 1), 2012; 401–442.

[2] Alastruey J, Khir A, Matthys K, Segers P, Sherwin S, Verdonck P, Parker K, Peiró J. Pulse wave propagation in a model human arterial network: Assessment of 1-D visco-elastic simulations against *in vitro* measurements. *J. Biomech.* 2011; **44**:2250–2258.

[3] Alastruey J, Nagel S, Nier B, Hunt A, Weinberg P, Peiró J. Modelling pulse wave propagation in the rabbit systemic circulation to assess the effects of altered nitric oxide synthesis. *J. Biomech.* 2009; **42**:2116–2123.

[4] Alastruey J, Xiao N, Fok H, Schaeffter T, Figueroa C. On the impact of modelling assumptions in multi-scale, subject-specific models of aortic haemodynamics. *J. R. Soc. Interface* 2016; **13**:1–17.

[5] Boileau E, Nithiarasu P, Blanco P, Müller L, Fossan F, Hellevik L, Donders W, Huberts W, Willemet M, Alastruey J. A benchmark study of numerical schemes for one-dimensional arterial blood flow modelling. *Int. J. Numer. Meth. Biomed. Eng.* 2015; **31**:1–31, doi:10.1002/cnm.2732.

[6] Flores J, Alastruey J, Poire EC. A novel analytical approach to pulsatile blood flow in the arterial network. *Ann. Biomed. Eng.* 2016; **44**:3047–3068.

[7] Xiao N, Alastruey J, Figueroa C. A systematic comparison between 1-D and 3-D hemodynamics in compliant arterial models. *Int. J. Numer. Meth. Biomed. Eng.* 2014; **30**:204–231.

[8] Alastruey J, Parker K, Peiró J, Sherwin S. Analysing the pattern of pulse waves in arterial networks: a time-domain study. *J. Eng. Math.* 2009; **64**:331–351.

[9] Alastruey J. On the mechanics underlying the reservoir–excess separation in systemic arteries and their implications for pulse wave analysis. *Cardiov. Eng.* 2010; **10**:176–189.

[10] Alastruey J, Passerini T, Formaggia L, Peiró J. Physical determining factors of the arterial pulse waveform: theoretical analysis and estimation using the 1-D formulation. *J. Eng. Math.* 2012; **77**:19–37.

[11] Alastruey J, Hunt A, Weinberg P. Novel wave intensity analysis of arterial pulse wave propagation accounting for peripheral reflections. *Int. J. Numer. Meth. Biomed. Eng.* 2014; **30**:249–279.

[12] Willemet M, Alastruey J. Arterial pressure and flow wave analysis using time-domain 1-D hemodynamics. *Ann. Biomed. Eng.* 2015; **43**:190–206.

[13] Alastruey J, Moore S, Parker K, David T, Peiró J, Sherwin S. Reduced modelling of blood flow in the cerebral circulation: Coupling 1-D, 0-D and cerebral auto-regulation models. *Int. J. Numer. Meth. Fluids* 2008; **56**:1061–1067.

[14] Jin W, Chowienczyk P, Alastruey J. An in silico simulation of flow-mediated dilation reveals that blood pressure and other factors may influence the response independent of endothelial function. *Am. J. Physiol. Heart Circ. Physiol.* 2020; **318**:H1337–H1345.

[15] Jin W, Alastruey J. Arterial pulse wave propagation across stenoses and aneurysms: Assessment of 1-D simulations against 3-D simulations and in vitro measurements. *J. R. Soc. Interface* 2021; **18**(20200881):1–17.

[16] Reymond P, Merenda F, Perren F, Rüfenacht D, Stergiopulos N. Validation of a one-dimensional model of the systemic arterial tree. *Am. J. Physiol. Heart Circ. Physiol.* 2009; **297**:H208–H222.

[17] Alastruey J, Parker K, Peiró J, Byrd S, Sherwin S. Modelling the circle of Willis to assess the effects of anatomical variations and occlusions on cerebral flows. *J. Biomech.* 2007; **40**:1794–1805.

[18] Alastruey J, Parker K, Peiró J, Sherwin S. Lumped parameter outflow models for 1-D blood flow simulations: effect on pulse waves and parameter estimation. *Commun. Comput. Phys.* 2008; **4**:317–336.

[19] Charlton P, Mariscal-Harana J, Vennin S, Li Y, Chowienczyk P, Alastruey J. Modeling arterial pulse waves in healthy aging: a database for in silico evaluation of hemodynamics and pulse wave indexes. *Am. J. Physiol. Heart Circ. Physiol.* 2019; **317**:H1062–H1085.