Journal of Universal Computer Science, vol. 25, no. 10 (2019), 1261-1278
submitted: 15/12/18, accepted: 17/5/19, appeared: 28/10/19 © J.UCS

High-Performance Simulation of Drug Release Model Using Finite Element Method with CPU/GPU Platform

Akhtar Ali
(Ibnu Sina Institute for Fundamental Science Studies (IIS), Universiti Teknologi Malaysia
Johor Bahro, Malaysia
utm.akhtar@gmail.com)

Imran Sarwar Bajwa
(Department of Computer Science, The Islamia University of Bahawalpur
Bahawalpur, Pakistan
Imran.sarwar@iub.edu.pk)

Rafaqat Kazmi
(Department of Computer Science, The Islamia University of Bahawalpur
Bahawalpur, Pakistan
rafaqutkazmi@gmail.com)

Abstract: This paper describes a hybrid CPU/GPU approach for solving a two-phase mathematical model numerically. The dynamic of drug release between the first phase (coating) and second phase (arterial tissue) is represented by a system of partial differential equations (PDEs). The system of equations is discretized by Finite Element Method. The whole discretized system involves a large sparse system of equation which requires a high computation. The CPU/GPU approach provides a platform to solve PDEs having extensive computations in parallel. Consequently, this platform can significantly reduce the solution times as compared to the implementation of CPU. This allows for more efficient investigation of different mathematical models, as well as, the governing parameters. In this paper, a significant parallel computing framework is presented to solve the governing equations numerically using the Graphics Processing Units (GPUs) with CUDA. This two-phase model investigates the impact of key parameters related to mass concentrations and drug release from tissue and coating layers. The identification and the role of major parameters such as (Filtration velocity, the ratio of accessible void volume to solid volume, the solid-liquid mass transfer rate) are tinted. Furthermore, the motivation and guidance for using parallel computing in order to handle computational complexities and large sparse system arise after discretizing the model equations are explained. We have designed a hybrid CPU/GPU solution of the proposed model by using Matlab. The parallel performance results show that CPU/GPU architecture is more efficient in large-scale problem simulations.

Keywords: Convection–diffusion–reaction equation, Finite element method, Parallel computing, CPU-GPU, Mass Transport, drug-eluting stent
Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1

1 Introduction

Endovascular drug eluting stents (DES) are being increasingly used for the prevention and cure of restenosis. Stents are devices inserted into arteries to widen their lumen, inhibit occlusion and restore blood flow perfusion to the tissues downstream. DES combine mechanical support of restricted lumen with local drug delivery. The drug delivery depends on different features, such as physico chemical properties and coating geometry, and drug properties, for example, solubility and diffusivity. In these stents, the therapeutic agent is loaded into biocompatible polymeric layers that are coated on the struts of the stent. These stents work under complex stress situations that vary in time and it is hard to accurately predict their efficiency and performance over extended periods of time [Khakpour et al., 2008]. There have been a few computational approaches to know about the mass transport in DES. The arterial wall considered as a porous medium in the form of a channel where the molecules of drug move progressively. Thus, many of these approaches model the arterial wall as a porous medium, including drug advection because of the plasma filtration through the tissue. An early one-dimensional porous medium model by Lovich and Edelman showed that when the arteries were uniformly loaded with heparin, most of the drug was cleared in a less than one hour, illustrating the need for sustained modes of delivery [Lovich et al., 1996]. Seo and Barakat studied the influence of various Reynolds numbers, drug diffusivities and stent diameters on drug deposition to design stents with minimum flow disturbance [Seo, T et al., 2005]. [Migliavacca, F et al., 2007] developed numerical models based on FEM in the presence of the atherosclerotic plaque, the coronary stent and artery. Mathematically, the arterial wall was considered as a porous medium on a macroscopic scale and the drug molecules transport was modeled by the convection-diffusion equation. These models considered mechanical effects of the stent expansion in addition to the drug delivery effect from the expanded stent into the artery wall. Kolachalama et al. [Kolachalama et al., 2009] applied a computational modeling approach to explore the impact of luminal flow patterns on arterial drug delivery and deposition, by considering the arterial wall to be a porous medium. A multi-domain and multi-scale mathematical model based on advection-diffusion reported by Vairo and co-workers, to investigate the dynamics of drug release in the polymeric substrate covering the stent, in the vessel lumen, and in the arterial wall [Vairo, G., et al., 2010]. A mathematical model developed by Pontrelli and co-workers for the diffusion transport of a drug with different properties and dimensions between two porous homogeneous media [Pontrelli, G et al., 2007]. The drug dynamics through two layers simulated and predicted to estimate the absorption rate of drug dose. The mathematical developments were able to incorporate the drug consumption effect due to binding of tissue cell. In a later study, Pontrelli developed a two-layer model to solve analytically the transient drug diffusion problem by considering porous layers of arterial wall faced with a DES [McGinty, S et al., 2015]. Even though the blood flow was coupled to the system, the simulations estimate the influence of solid-liquid parameters on mass and drug concentrations. In their study, they have not considered the effect of filtration velocity on mass and drug concentrations. In this study, we have considered the effect of filtration velocity along with other corresponding parameters. Furthermore, for numerical computations a parallel hybrid platform (CPU/GPU) together with the FEM is applied to simulate the model equations.

The parallelism is a tendency in current computing and, therefore, microprocessor manufacturers are focusing on adding cores rather than on increasing single-thread performance. However, there are a lot of parallel architectures, mostly widespread many-core architecture are the graphics processing units (GPUs), and the most general multi-core architecture is the central processing units (CPUs) [Šimkus, A et al., 2013].

However, as a consequence of the massive parallelism offered by GPU as compared to CPU, the tendency in scientific computing is to use GPU for accelerating the intensive and intrinsic parallel computations [Kirk, D.B, 2012], [Alias et al., 2017], [Alias, 2014]. Furthermore, it is preferred to use GPU as a co-processor to accelerate part or the whole code over the other architectures because its market is entrenched in the computer games industry. Although some researches are reported in literature are focused on completely porting the FEM subroutines to be executed in GPU [Šimkus, A et al., 2013], [Fu, Z., et al., 2014] and the other works are focused on porting parts of the FEM code to the GPU, such as the numerical integration [Knepley et al., 2013], the matrix assembly [Cecka, C et al., 2011] the global finite element matrices and the solution of large systems of equations [Ramírez-Gil et al., 2015]. Even though some former works accelerate the construction of sparse FE matrix [Dziekonski, A., et al., 2013]. These approaches require high GPU memory which means that the problem may be analyzed is limited in size or an expensive GPU with high memory capacity is needed. The code executed in GPU is compared with its respective serial implementation counterpart executed in one CPU core. In this study parallel computing capabilities of MATLAB together with finite element method are analyzed. The advantage of GPU implementation is observed by using curves.

2 Processing of Mathematical Model

The two-phase system consists of two parts, a polymeric part that contains drug initially and arterial wall with smooth muscle cells where the drug is directed as shown in figure 2.

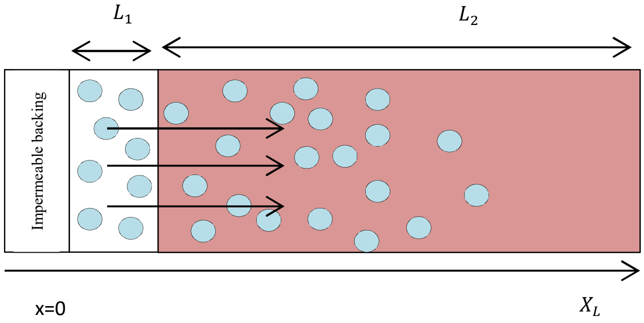


Figure 1: Mechanism of drug transport from DES into artery wall.

The layer (i) is mapped out as a thin planar slab and its one side is covered with a sealed backing (metallic strut) associated with the layer (ii). A membrane may exist at the interface of two layers that protects the polymer matrix and controls the rate. It is imperative to mention here that the dynamic of drug inseparably depend on the coupled system characteristics. Generally, the dynamics of mass takes place along the normal direction to the surface of the tissue and the model may logically be restricted to a one dimensional case. Let L_1 and L_2 be considered as the thicknesses of the first and second layers respectively, as represented in (Fig. 1).

2.1 Modeling of Drug Transport in Polymer and Tissue

The dynamics equations for the layer (i) coating and the layer (ii) arterial wall are defined as [Ali et al., 2017].

$$\frac{\partial c_1^b}{\partial t} = -\gamma_1\left(e_1 c_1^b - c_1^f\right) \; in \; (-L_1, 0) \tag{1}$$

$$\frac{\partial c_1^f}{\partial t} = D_1 \frac{\partial^2 c_1^f}{\partial x^2} + \gamma_1\left(e_1 c_1^b - c_1^f\right) \quad in \;\; (-L_1, 0) \tag{2}$$

$$\frac{\partial c_2^f}{\partial t} = \frac{\partial}{\partial x}\left(D_2 \frac{\partial c_2^f}{\partial x} - u c_2\right) - \delta_2\left(\frac{c_2^f}{K_1} - e_2 c_2^b\right) \qquad in \;\; (0, L_2) \tag{3}$$

$$\frac{\partial c_2^b}{\partial t} = \delta_2\left(\frac{c_2^f}{K_1} - e_2 c_2^b\right) \qquad in \;\; (0, L_2) \tag{4}$$

Where c_1^b, c_1^f, c_2^f and c_2^b are free and bound concentrations of drug in fluid and solid phases. Let γ_1 is the ratio of solid–liquid mass transfer coefficient, D_2 is the diffusivity of free drug and D_1 is the diffusion coefficients for first and second layers. The magnitude of the convection is denoted by u and it acts in the direction of positive x-axis. Consequently, the arterial drug release is directed by the reaction-convection diffusion equation.

2.2 Boundary Conditions

The model initial and boundary conditions are:

$$c_1^b(x,0) = C_1 \,, c_1^f(x,0) = c_2^f(x,0) = c_2^b(x,0) = 0$$

where C_1 denote the initial concentration of drug in the coating. At the interface of each layer, a flux continuity condition is assumed.

$$-D_1 \frac{\partial c_1^f}{\partial x} = -D_2 \frac{\partial c_2^f}{\partial x} + u c_2^f \qquad at \;\; x = 0$$

A membrane with permeability *P* is set up at the interface between the coating and wall to reduce the drug rate. The mass transport from the top coat can be defined as: [Kargol, A., et al., 1997].

$$-D_2 \frac{\partial c_2^f}{\partial x} = P\left(\frac{c_1^f}{k_1 \epsilon_1} - \frac{c_2^f}{k_2 \epsilon_2}\right) \qquad at \;\; x = 0$$

Applying no flux boundary conditions because the impermeable backing does not permit mass flux to elapse to the outer surrounding, so:

$$D_1 \frac{\partial c_1^f}{\partial x} = 0 \qquad at \;\; x = -L_1$$

Furthermore, it is assumed that there is no concentration at the outer boundary L_2

$$c_2^f = 0 \qquad at \;\; x = -L_2$$

2.3 Scaling of Variables

The scaling of variables is presented in Equation 5 as:

$$\bar{x} = \frac{x}{L_2}, \quad \bar{t} = \frac{D_1 t}{L_2^2}, \quad \overline{c_i^b} = \frac{c_i^b}{C_1}, \quad \overline{c_i^f} = \frac{c_i^f}{C_1} \tag{5}$$

Therefore, the reaction and convection terms are governed by the nondimension parameters given as:

$$D = \frac{D_1}{D_2}, \ L = \frac{L_1}{L_2}, \ \gamma = \frac{\gamma_1 L_2^2}{D_2}, \sigma = \frac{k_2 \epsilon_2}{k_1 \epsilon_1}, \phi = \frac{PL_2}{D_2 k_1 \epsilon_1}, Pe = \frac{uL_2}{D_2}, Da = \frac{\beta_2 L_2^2}{D_2}$$

The resulting nondimensional equations come as:

$$\frac{\partial c_1^b}{\partial t} = -\gamma\left(e_1 c_1^b - c_1^f\right) \qquad in \ \ (-L_1, 0) \tag{6}$$

$$\frac{\partial c_1^f}{\partial t} = D\frac{\partial^2 c_1^f}{\partial x^2} + \gamma\left(e_1 c_1^b - c_1^f\right) \qquad in \ \ (-L_1, 0) \tag{7}$$

$$\frac{\partial c_2^f}{\partial t} = \frac{\partial^2 c_2^f}{\partial x^2} - Pe\frac{\partial c_2^f}{\partial x} + Da\left(c_2^f - K_2 e_2 c_2^b\right) \qquad in \ \ (0, 1) \tag{8}$$

$$\frac{\partial c_2^b}{\partial t} = Da\left(c_2^f - K_2 e_2 c_2^b\right) \qquad in \ \ (0, 1) \tag{9}$$

where *Da* and *Pe* are the Damkohler and Peclet numbers, respectively. The main characteristic of drug that normally coated on stents is their tenacious binding characteristics. These characteristics make sure that the drug is intensely reserved in tissue and consequently employs the desired impact for longer. Actually, it means that the timescale for binding is considerably smaller than that of convection and diffusion. Usually, for the Damkohler number, to be higher than 1 and for the Peclet number to be smaller than 1 in [Zunino, P., et al., 2012] that shows that the reaction is dominant over diffusion which in turn is more important than convection. Furthermore, [Zunino, P., et al., 2004] has analysed the changes in the total amount of drug without and with convection and has obtained distinguishable results.

So the boundary conditions become

$$\frac{\partial c_1^f}{\partial x} = 0, \quad at \ x = -L_1 \tag{10}$$

$$-D\frac{\partial c_1^f}{\partial x} = \frac{\partial c_2^f}{\partial x} \qquad at \ \ x = 0 \tag{11}$$

$$-\frac{\partial c_2^f}{\partial x} = \phi\left(\sigma c_1^f - c_2^f\right) \ \text{ at } \ x = 0 \tag{12}$$

$$c_2^f = 0 \qquad at \ \ x = 1 \tag{13}$$

$$c_1^b = 1, c_1^f = c_2^f = c_2^b = 0 \quad at \ \ t = 0 \tag{14}$$

The following table represents the values of parameters which are used in the simulations of the model equations.

	Parameter	Suggestive Range
Diffusion coefficient in the coating	D_1	$10^{-10}\ cm^2 s^{-1}$
Effective diffusivity of free drug	D_2	$7.10^{-8}\ cms^{-1}$
Topcoat Permeability	P	$10^{-6}\ cms^{-1}$
Porosity coefficient in tissue	ϵ_2	0.61
Partition coefficient in the coating	k_1	1
Transmural Velocity	u	$5.8 \times 10^{-6}\ cm/s$
Thicknesses of coating	L_1	$5.10^{-4}\ cm$
Equilibrium dissociation constant K_2	$\frac{\delta_2}{\beta_2}$	0.1
Porosity coefficient in the coating	ϵ_1	0.5
The ratio of accessible voide volume to solid volume in the first layer e_1	$\frac{k_1\epsilon_1}{1-\epsilon_1}$	1
Peclet Number	Pe	0.1,0.2,0.3
Unbinding rate constants	δ_2	$10^{-6}s^{-1}$
binding rate constants	β_2	$10^{-3}s^{-1}$
Ratio of accessible voide volume to solid volume in second layer e_2	$\frac{k_2\epsilon_2}{1-\epsilon_2}$	1.5
Thicknesses of tissue	L_2	$5.10^{-2}\ cm$
Partition coefficient in tissue	k_2	1

Table 1: Values of dimensional parameters used in the simulations of the model equations. The selected values are taken from the literature.

3 Sequential Results and Numerical Simulations

In this section the model equations (5-8) with prescribed conditions (9-14) are solved by FEM. For discretization, dividing the interval $(-L, 0)$ into $n + 1$ equidistance nodes $x_k = (k - n)h_1$,$k = 0,1, \dots n$, and the interval (0,1) with $n_1 + 1$ nodes $x_k = kh_2$, $k = 0,1, \dots, n_1$. Here, h_1 and h_2 represents the spacing for first and second layers, respectively. The system of PDE's is reduce to the system of ordinary differential equations (ODEs) after applying spatial discretization, given in the form

$$\frac{dX}{dt} = B(X) \tag{15}$$

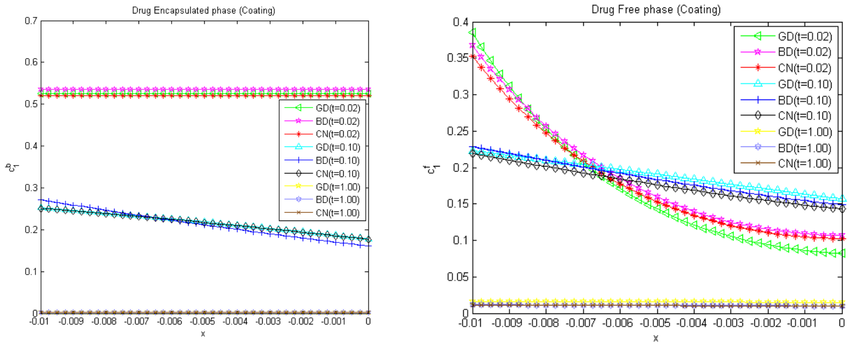
where $X = \left(c_1^{b,0}, c_1^{b,1} \dots c_1^{b,n}, c_1^{f,0}, c_1^{f,1} \dots c_1^{f,n}, c_2^{f,0}, c_2^{f,1} \dots c_2^{f,n_1}, c_2^{b,0}, c_2^{b,1} \dots c_2^{b,n_1}\right)^T$ and $B(X)$ consist of $2(n + n_1)$ equations. To verify the exactness of the numerical solution, (15) is solved by three time marching techniques namely the backward difference (BD), Galerekin difference (GD) and the Crank Nicolson (CN) [Ali et al., 2017], [Alias, N., et al., 2016], [Alias, N., et al., 2016]. The mass of drug is simply calculated in each layer by using the concentration integral of Equation 16.

$$M_j(t) = \int j(x,t), \qquad j = c_1^b, c_1^f, c_2^f, c_2^b \tag{16}$$

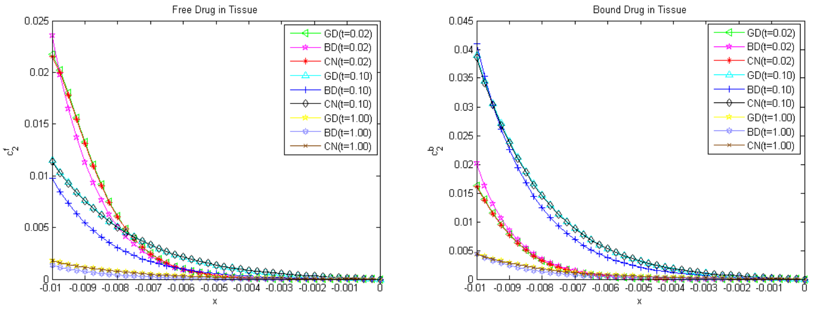
The main objective of this research is to examine the consequences of solid-liquid mass transport in the coating on the drug concentrations that are attained within the artery wall. Thus, we are focusing to analyze the impact of varying γ_1, the filtration velocity and e_1, on the resulting drug mass and drug release profiles within each layer. Initially, we select $e_1 = 1$ corresponding to the values $\epsilon_1 = 0.5, k_1 = 1$

3.1 Filtration Velocity Impact on Drug and Mass Concentrations

The impact of filtration velocity on drug and mass concentrations is investigated by fixing the $\gamma_1 = 10^{-2}$, $e_1 = 1$ and varying Peclet number for different time values. To illustrate the impact of filtration velocity on the concentration of the drug, a value $\delta_2 = 10^{-6}$ cms^{-1} is selected, in agreement with measurements considered by [Migliavacca, F et al., 2007], [Meyer et al., 1996]. The figures (2-7) highlight the impact of filtration velocity which is connected with non-dimensional parameter Pe.



(a) (b)



(c) (d)

Figure 2: Normalized concentration profiles for different times when high porosity in (a) drug encapsulated phase (b) drug-free phase in a coating (c) free drug in tissue (d) bound drug in tissue for $\gamma = 10^{-2}, e_1 = 1, \epsilon_1 = 0.5, Pe = 0.1$.

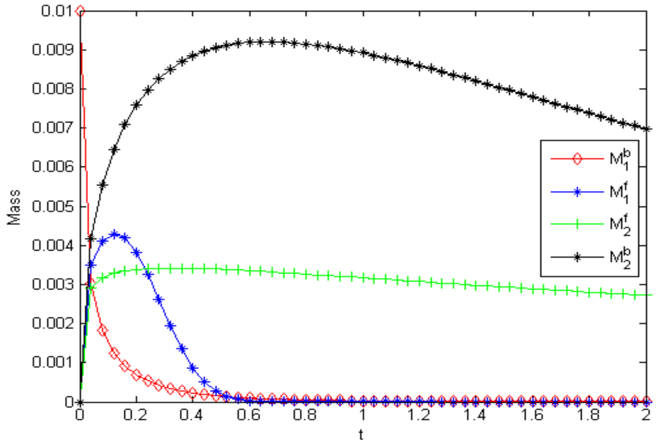
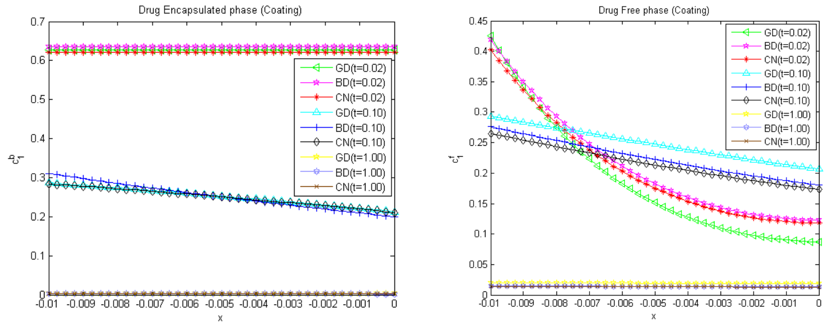
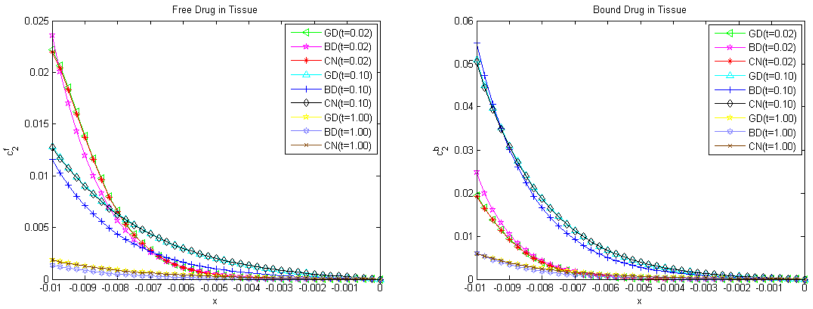


Figure 3: Variation of the normalized drug mass transport in arterial wall and coating for $\gamma = 10^{-2}, e_1 = 1, \epsilon_1 = 0.5, Pe = 0.1$.



(a) (b)



(c) (d)

Figure 4: Normalized concentration profiles for various times when high porosity in (a) drug encapsulated phase (b) drug-free phase in the coating (c) free drug in tissue (d) bound drug in tissue for $\gamma = 10^{-2}, e_1 = 9, \epsilon_1 = 0.9, Pe = 0.2$.

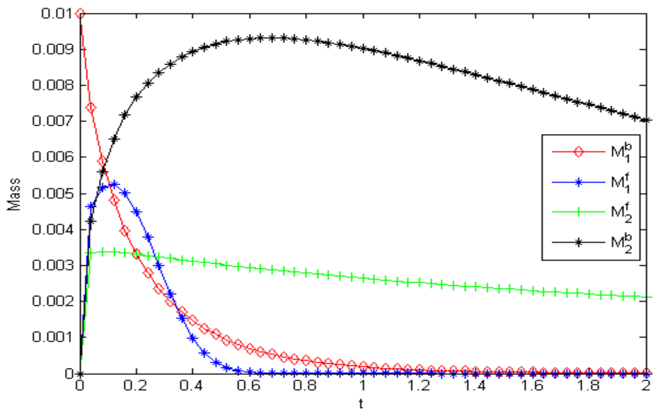
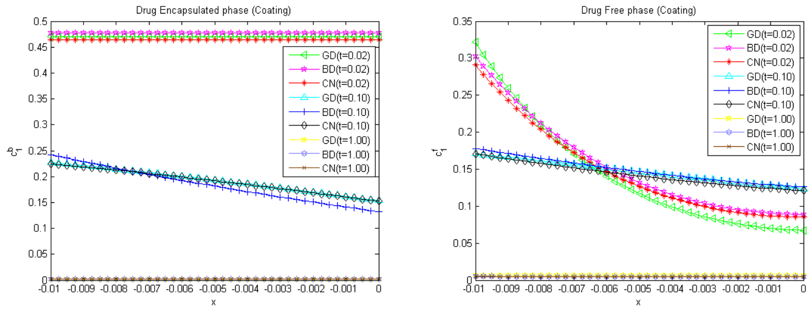
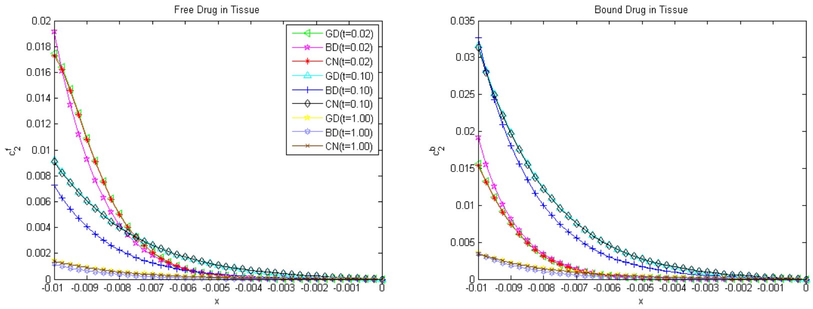


Figure 5: Variation in the normalized drug mass in the coating and arterial wall for $\gamma_1 = 10^{-2}, e_1 = 9, \epsilon_1 = 0.9, Pe = 0.2$.



(a) (b)



(c) (d)

Figure 6: Normalized concentration profiles for various times when high porosity in (a) drug encapsulated phase (b) drug-free phase in the coating (c) free drug in tissue (d) bound drug in tissue for $\gamma = 10^{-2}, e_1 = 1, \epsilon_1 = 0.5, Pe = 0.3$.

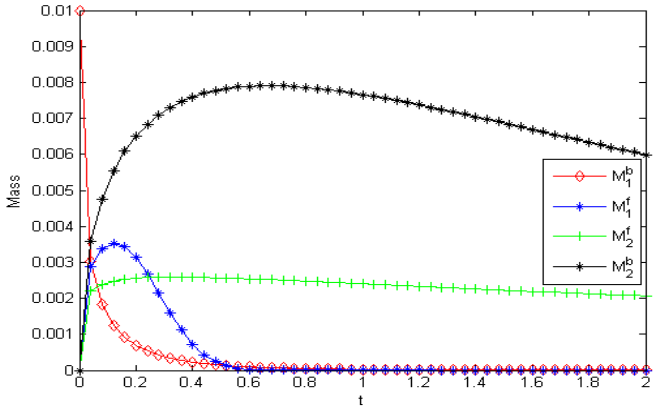


Figure 7: Variation of the normalized drug mass in arterial wall and coating for $\gamma = 10^{-2}, e_1 = 1, \epsilon_1 = 0.5, Pe = 0.3$.

Simulations for three different values of Pe within a compatible range are taken to explore the impact of advection velocity on drug transport. Consequently, a relatively small advection brings down the concentration profiles for all times as shown in Figures 2, 4 and 6. This is due to the convection velocity sweeps the drug away from the wall, where it is spread. The comparison between Figures 2, 4 and 6 show the concentration curves generally maintaining the same shape, the peaks are now sharper by increasing Pe. For all time spans, a decreasing Pe basically brings about a sharp drop in the value of peak concentration.

The Figures 3, 5 and 7 represent the free and bound drug mass concentrations for different values of Peclet number. The drug mass in all cases is first rising up to a peak value and then decreases asymptotically to x-axis. The mass peak becomes lower as the values of Pe increases. The profiles may appear expanded at transitional and later instants, therefore, a more uniform concentration is assured.

3.2 Implementation of CPU and CPU-GPU (Hybrid Platform)

The main aim in this section is to make a comparison between the computational accuracy and efficiency of GPU-based results and CPU-based results of the proposed model. The programming code is developed using Matlab 2017a. Simulations are done on a computer that has 2.10 GHz. Intel(R) Xeon(R) CPU E5-2620 v2, 24 cores and Windows 10, 64-bit operating system. The system has two GPU cards, NVIDIA Quadro K4000, and NVIDIA Tesla K20c. These cards have 3.0 and 3.5 compute capabilities respectively. The CUDA cores of NVIDIA Quadro K4000 and NVIDIA Tesla K20c GPU cards are 768 and 2496 respectively having 3 GB memory. The memory clock rate for both cards is 2808 Mhz and 2600 Mhz and memory bandwidth is 134 GB/sec. Table 2 and 3 represents the sequential performance of numerical results for two test cases and Figure 8 represents the Sequential and parallel algorithm for processing the data using FEM.

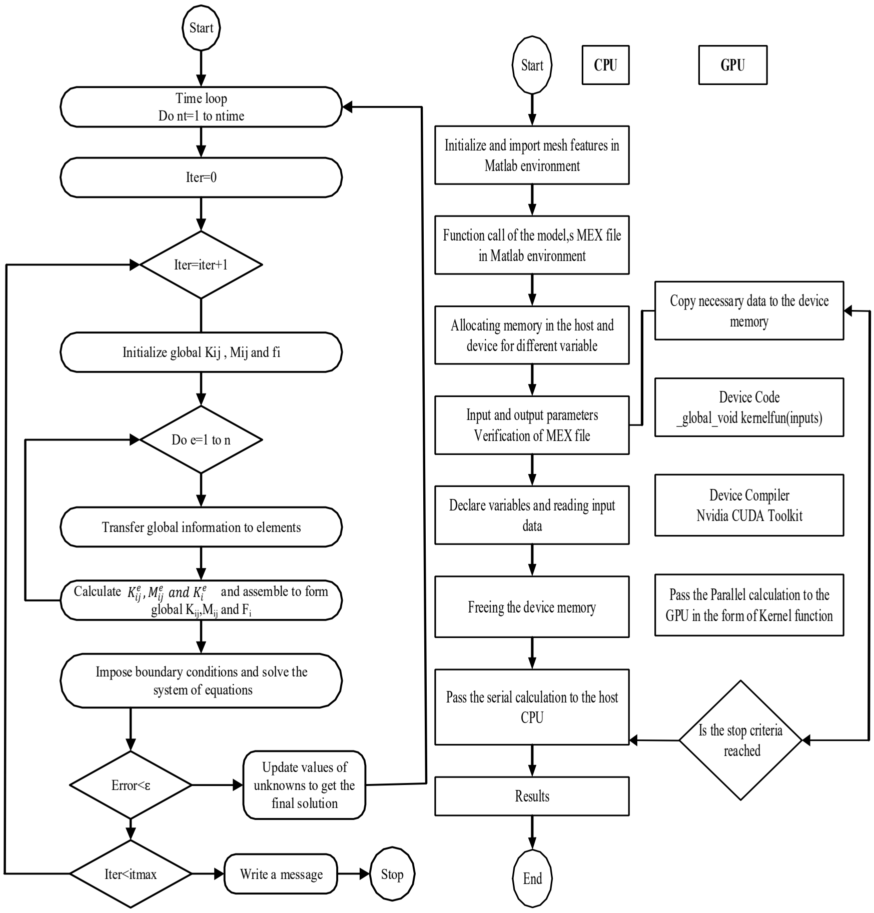

Start
Time loop
Do nt=1 to ntime
Iter=0
Iter=iter+1
Initialize global Kij , Mij and fi
Do e=1 to n
Transfer global information to elements
Calculate K_{ij}^e, M_{ij}^e and K_i^e and assemble to form global K_{ij},M_{ij} and F_i
Impose boundary conditions and solve the system of equations
Error<ε
Update values of unknowns to get the final solution
Iter<itmax
Write a message
Stop
Start
CPU
GPU
Initialize and import mesh features in Matlab environment
Function call of the model,s MEX file in Matlab environment
Allocating memory in the host and device for different variable
Input and output parameters Verification of MEX file
Declare variables and reading input data
Freeing the device memory
Pass the serial calculation to the host CPU
Results
End
Copy necessary data to the device memory
Device Code
_global_void kernelfun(inputs)
Device Compiler
Nvidia CUDA Toolkit
Pass the Parallel calculation to the GPU in the form of Kernel function
Is the stop criteria reached

Figure 8: Sequential and parallel algorithm for processing the data using FEM.

Method	**Backward Method**	**Galerkin Method**	**Crank Nicolson Method**
Execution Time Percentage (%)	11994.08 48.18%	14341.21 38.04%	23147.11
No of iterations	100	100	100
Maximum Error	6.83e-03	4.1043e-03	2.64e-04
RMSE	8.36e-04	3.5373e-04	5.42e-05

Table 2: Numerical results for sequential performance for a first test case with matrix size (20004×20004).

Method	**Backward Method**	**Galerkin Method**	**Crank Nicolson Method**
Execution Time Percentage (%)	19706.44 44.78%	24136.74 32.36%	35691.35
No of iterations	100	100	100
Maximum Error	5.46e-04	2.24e-04	9.55e5-05
RMSE	8.25e-05	5.62e-06	3.12e-06

Table 3: Numerical results for the sequential algorithm for the 2nd test case with matrix size (24004×24004).

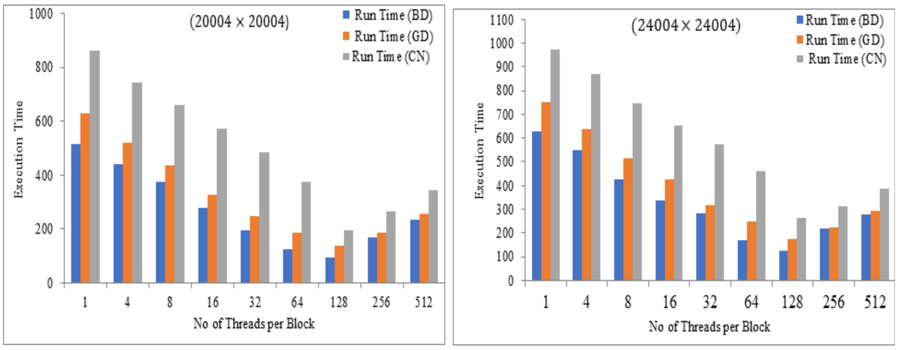
Tables 2 and 3 represent the numerical results of the sequential algorithm for BD, GD and CN methods. Two test cases are considered to measure parallel performance evaluation. The first test case has 20004 unknowns and 2nd test case has 24004 numbers of unknowns. The CN method is used as a benchmark numerical algorithm to compare its numerical results with other numerical schemes. The BD scheme is 48.18% faster than the CN scheme, while the GD scheme is 38.04% faster than the CN scheme for the first test case. On the other hand, the BD scheme is 44.78% faster than the CN scheme for the 2nd test case. The GD scheme is 32.36% faster than the CN scheme for the 2nd test case. The BD method is more efficient based on run time. The Tables 2 and 3 also show the direct proportion of matrix size, as the matrix size increases the computational time of CPU also increases. Moreover, the CN method has less error as compared to other methods. The CN method is computationally costly. The CN method in both test cases has least RMSE and maximum error as compared to BD and GD methods which shows that CN is a more accurate method.

3.3 Parallel Results and Discussion

In this section, the improvements brought by the GPU acceleration and developed for governing equations are presented. The parallel performance results for three-time marching numerical schemes are shown in Tables 4 and 5 and Figures (9-10). The tests are carried out for large sparse matrices using 2500 and 3000 number of finite elements. The corresponding numbers of unknowns are 20004 and 24004 respectively. For the time measurement, 100-time iterations are used by storing the solution to generate a detailed animation to visualize the drug movement. The initialization and set-up times are also excluded from the time measurements. Thus, the reported execution times represent the computational times spent in solving the model equations. The number of threads per block is taken from 1 to 512 for execution of the kernel. The CN method is used as a control method. The percentages are calculated using the formula given as:

$$\% \, w \, r \, t \, CN = \left| \frac{T_c - T_o}{T_c} \right| \times 100\% \qquad (17)$$

Where T_c is the execution time for the CN method and T_o is for other methods. Figure 9 indicates the execution time for each numerical scheme. The execution time decreases as the number of threads per block increases upto 128, subsequently, the execution time increases along the number of threads per block. This happens because of communication overhead process. The Table 4 and 5 illustrate the details of execution time as well as the percentage for all numerical methods.



(a) (b)

Figure 9: Execution time for all schemes for matrix size (a) (20004×20004) *and (b)* (24004×24004) *versus the number of Threads per block.*

Table 4 explains the parallel performance measures of three methods such as BD, GD, and CN. The parallel performance measurement was carried out for two test cases. The first test case has 20004 unknowns and 2500 number of elements. The performance measures in terms of run time depending on the run time of CPU and GPU versus a number of threads per block. The run time increases from thread 1 to 128 and later decreases due to communication overhead (cache ram, register values replacement, bandwidth). Table 4 also offers a good comparison between three numerical methods.

By comparing the CN method and the BD method based on run time, Table 4 shows, the BD method is faster (39.95-67.13%) than the CN method. The GD method is also faster than the CN method in terms of run time. The percentage of run time for the GD method is (27.11-51.00) %. The BD method remains faster as compared to GD and CN methods after increasing the number of elements. This is because of the difference between the computational cost of the two methods. Table 5 represents the measures of parallel results for a 2nd test case, based on run time by increasing the number of elements. In this case, the number of elements are 3000, and 24004 are the number of unknowns.

No of Threads Percentage (%)	**Backward Difference**	**Galerkin Difference**	**Crank Nicolson**
1	517.14 **39.95**	627.70 **27.11**	861.23
4	442.54 **40.47**	522.31 **29.74**	743.41
8	376.53 **42.82**	438.85 **33.38**	658.62
16	278.37 **51.44**	326.17 **43.10**	573.29
32	197.71 **59.07**	246.35 **49.01**	483.15
64	123.85 **67.13**	184.62 **51.00**	376.80
128	96.54 **50.99**	138.73 **29.57**	197.00
256	169.97 **36.04**	185.44 **30.21**	265.74
512	237.19 **31.38**	259.07 **25.10**	345.70

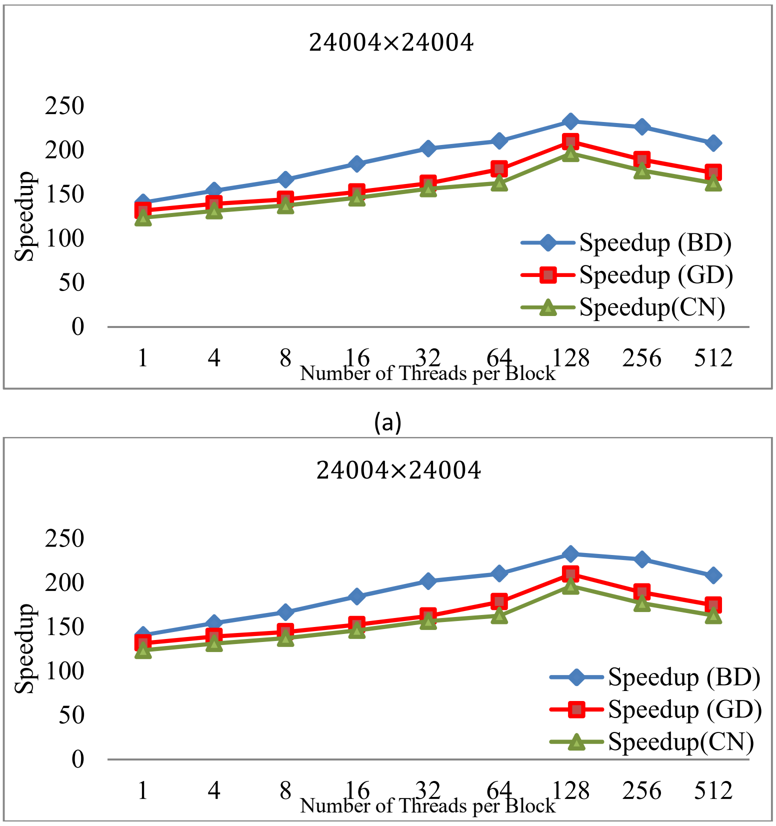
Table 4: Parallel performance results of a model problem for global matrix size (20004×20004).

Table 5 represents the parallel performance measures of BD, GD and CN methods for the 2nd test case. The BD method has faster execution time (35.40-63.32)% as compared to other methods because this method has fewer computations and communication time. The other performance metrics such as speedup, temporal performance, effectiveness, and efficiency are analyzed in Figure 10. The Parallel Performance Indicator speedup, depending on the run time of CPU, GPU and number of threads per block. The improvement in speedup by increasing number of threads is presented in Figure 10. Two data sets used to compare the impact of speedup on parallel performance results. The first data set has a matrix dimension (20004×20004) and the 2nd data set has a matrix dimension(24004×24004).

No of Threads Percentage (%)	**Backward Difference**	**Galerkin Difference**	**Crank Nicolson**
1	629.66 **35.40**	749.45 **23.11**	974.76
4	546.88 **37.12**	637.63 **26.67**	869.66
8	426.28 **42.70**	513.92 **31.00**	743.98
16	338.08 **48.10**	428.19 **34.26**	651.38
32	284.57 **50.39**	315.63 **44.97**	573.67
64	169.10 **63.32**	247.53 **46.32**	461.12
128	126.95 **51.68**	176.78 **32.71**	262.75
256	219.01 **30.29**	225.88 **28.11**	314.20
512	280.33 **27.51**	294.01 **23.97**	386.73

Table 5: Parallel performance results of a model problem for global matrix size is (24004×24004).

The speedup for the BD method increases from 100.22% to 209.36% for the first test case (20004×20004). The speedup increases from 140.93% to 232.58% for the 2nd test case(24004×24004). For GD method speedup increases from 95.33% to 201.07% for first test case while the speedup increases from 131.63% to 209.55% for the 2nd test case. The speedup of the CN method increases from 88.90% to 179.830% for the first test case, while the speedup increases from 123.60% to 1965.36% for the 2nd test case. The speedup is increasing from thread 1 to 128, later it decreases due to communication overhead in both cases. The trends of speedup are maximum when a number of threads are 128, later speedup decreases significantly along the number of threads, because of inter-process communication overhead (cache ram, register values replacement, bandwidth). The speedup for the BD method is better than for the GD and CN methods. This is because the GD and CN methods have more computational complexity as compared to the BD method. The speedup for the 2nd test case is better as compared to the first test case in all methods. This comparison shows that the hybrid platform is efficient for big size of data.



(a)

(b)

Figure 10: Parallel performance metric speedup for two sets of data (a)(20004×20004) *and (b)* (24004×24004) *versus a number of threads per block.*

4 Conclusion

In this paper, a hybrid CPU/GPU solution for a two-phase mathematical model is offered to investigate the impact of parameters related to drug concentrations. The role of the Filtration velocity is highlighted, and the proposed model is solved numerically by FEM. The model equations are discretizing spatially by FEM, for temporal discretization the three-time marching schemes are applied namely Galerkin finite difference, backward finite difference, and the Crank Nicolson schemes. It is found that the filtration velocity has a certain effect on the drug mass and drug release that is achieved within the tissue. However, it is observed that for varying the Peclet number Pe, it has a great impact on drug mass and drug release profiles in the artery wall. Specifically, by varying the value of Pe the timing of peak drug concentrations and the speed of drug release can be controlled. Furthermore, this paper explained our motivation for using GPU accelerators in order to handle computational complexities in the model. We have designed a hybrid CPU/GPU solution for the bi-phase model solution and obtained satisfactory parallel performance results such as run time and speed ups. The parallel performance results also show that GPU solver is more efficient in large-scale problem simulations. The performance results showed significant accelerations by using GPU devices. Even though the GPU is used only in several parts of the algorithm, the speed-ups are encouraging, which allows us to continue in this direction.

References

[Ali et al., 2017] Ali, A., N. Alias, and W. Al Rahmi, "*Numerical Simulation of Two Phase Mathematical Model for Transportation of Mass and Drug from Drug Eluting Stents";* Mathematical Modelling of Natural Phenomena, **12,** 5 (2017) 162-179.

[Alias, 2014] Alias, N., et al. "*High Performance Nanotechnology Software (HPNS) for Parameter Characterization of Nanowire Fabrication and Nanochip System"*. in *International Conference on Intelligent Software Methodologies, Tools, and Techniques*, Springer (2014).

[Alias, N., et al., 2016] Alias, N., et al., "*Numerical analysis on mathematical model for drug delivery system on blood flow in external magnetic fields by magnetic nanoparticles";* Jurnal Teknologi, **78**, 4 (2016) 31-37.

[Alias, N., et al., 2016] Alias, N., et al., *Parallel Computing of Numerical Schemes and Big data analytic for solving real life applications.* Jurnal Teknologi. **78**(8-2): (2016), 151-162.

[Alias et al., 2017] Alias, N., R. Sahnoun, and V. Malyshkin, "*High-performance computing and communication models for solving the complex interdisciplinary problems on DPCS";* ARPN Journal of Engineering and Applied Sciences, 12, 2 (2017) 356-364.

[Cecka, C et al.,] Cecka, C., A.J. Lew, and E. Darve, "*Assembly of finite element methods on graphics processors";* International journal for numerical methods in engineering, **85,** 5 (2011) 640-669.

[Dziekonski, A., et al., 2013] Dziekonski, A., et al., "*Generation of large finite-element matrices on multiple graphics processors";* International Journal for Numerical Methods in Engineering, **94**, 2 (2013) 204-220.

[Fu, Z., et al., 2014] Fu, Z., et al., "*Architecting the finite element method pipeline for the GPU";* Journal of computational and applied mathematics, **257** (2014) 195-211.

[Kargol, A., et al., 1997] Kargol, A., M. Kargol, and S. Przestalski, "*The Kedem-Katchalsky equations as applied for describing substance transport across biological membranes";* Cellular and Molecular Biology Letters, **2,** 2 (1997) 117-124.

[Khakpour et al., 2008] Khakpour, M. and K. Vafai, "*Critical assessment of arterial transport models";* International Journal of Heat and Mass Transfer, **51**, 3 (2008), 807-822.

[Kirk, D.B et al., 2012] Kirk, D.B. and W.H. Wen-mei, "*Programming massively parallel processors: a hands-on approach";* Elsevier (2012).

[Knepley et al., 2013] Knepley, M.G. and A.R. Terrel, "*Finite element integration on GPUs";* ACM Transactions on Mathematical Software (TOMS), **39**, 2 (2013): 1-11.

[Kolachalama et al., 2009] Kolachalama, V.B., et al., "*Luminal flow patterns dictate arterial drug deposition in stent-based delivery";* Journal of Controlled Release, **133**, 1 (2009) 24-30.

[Lovich et al., 1996] Lovich, M.A. and E.R. Edelman, "*Computational simulations of local vascular heparin deposition and distribution";* American Journal of Physiology-Heart and Circulatory Physiology, **271,** 5 (1996) H2014-H2024.

[McGinty, S et al., 2015] McGinty, S. and G. Pontrelli, "*On the influence of solid–liquid mass transfer in the modelling of drug release from stents";* Journal of Coupled Systems and Multiscale Dynamics, **3**, 1 (2015) 47-56.

[Meyer et al., 1996] Meyer, G. and A. Tedgui, "*Effects of pressure-induced stretch and convection on low-density lipoprotein and albumin uptake in the rabbit aortic wall";* Circulation research, **79,** 3 (1996), 532-540.

[Migliavacca, F et al., 2007] Migliavacca, F., et al., "*Expansion and drug elution model of a coronary stent";* Computer methods in biomechanics and biomedical engineering, **10**, 1 (2007) 63-73.

[Pontrelli, G et al., 2007] Pontrelli, G. and F. de Monte, "*Mass diffusion through two-layer porous media: an application to the drug-eluting stent".* International Journal of Heat and Mass Transfer, **50**, 17 (2007) 3658-3669.

[Ramírez-Gil et al., 2015] Ramírez-Gil, F.J., M.d.S.G. Tsuzuki, and W. Montealegre-Rubio, *Global finite element matrix construction based on a CPU-GPU implementation,* arXiv preprint arXiv:15, 01 (2015) 47-84.

[Seo, T et al., 2005] Seo, T., L.G. Schachter, and A.I. Barakat, "*Computational study of fluid mechanical disturbance induced by endovascular stents";* Annals of biomedical engineering, **33**, 4 (2005) 444-456.

[Šimkus, A et al., 2013] Šimkus, A. and S. Turskienė, "*Parallel computing for the finite element method in MATLAB";* Computational Science and Techniques, **1,** 2 (2013) 214-221.

[Vairo, G., et al., 2010] Vairo, G., et al., "*Drug release from coronary eluting stents: a multidomain approach";* Journal of Biomechanics, **43**, 8 (2010) 1580-1589.

[Zunino, P., et al., 2004] Zunino, P., "*Multidimensional pharmacokinetic models applied to the design of drug-eluting stents";* Cardiovascular Engineering: An International Journal, **4,** 2 (2004) 181-191.

[Zunino, P., et al., 2012] Zunino, P.,Tzafriri, A.R., et al., "*Stent elution rate determines drug deposition and receptor-mediated effects";* Journal of Controlled Release, **161,** 3 (2012) 918-926.