Brain Dump of my Work in 1D

Maggie McCarthy

June 2023

1 Singularly Perturbed Problems

Singularly perturbed problems are differential equations that involve a small positive parameter ϵ. The solutions to such problems as ϵ approaches zero differ than the solutions at $\epsilon = 0$. Furthermore, the solutions to singularly perturbed problems tend to have boundary and/or interior layers which are regions of the domain where the solution changes rapidly (Farrell or MacLachlan). An example of a singularly perturbed problem is

$$-\epsilon^2 u^{''} + a(x)u^{'}(x) + b(x)u = f(x) \text{ on } (0, 1),\ u(0) = A,\ u(1) = B.$$

When $b = 0$, the problem is called a *convection-diffusion* problem. If $a = 0$ and $b \neq 0$ then the problem is called a *reaction-diffusion* problem (all MacLachlan). If the singularly perturbed problem has any nonlinearity, we may expect multiple solutions with boundary and/or interior layers (Kopteva). These boundary layers are caused by the solution having to satisfy the imposed boundary conditions. Such layers become more pronounced as the parameter ϵ is decreased. Consequently, it becomes more challenging to construct initial guesses that allow us to use Newton's Method to find solutions of singularly perturbed problems small values of ϵ.

With this analysis of singularly perturbed problems in mind, some questions may arise; How can one capture the multiple solutions to singularly perturbed problems? Furthermore, how can one resolve the layers of these problems without using large grids? The answers to such problems will be explored in this paper. To do so, we discuss adaptive mesh methods in 1D and 2D. These methods allow us to resolve the layers in the solutions to singularly perturbed problems using less points than what would be required to find an analogous solution on a uniform grid. Such discussion is focused around a linear singularly perturbed reaction diffusion problem. To capture the multiple solutions of singularly perturbed problems, we employ the the *Deflation method* (SEE ?). Those who have explored such method previously have used uniform grids. We go a step further by coupling deflation with adaptive mesh methods. This coupling of an adaptive mesh method with deflation will allow us to capture the multiple solutions using smaller grids than what is required when grid points are equally spaced. To display the success of such coupling, a 1D semilinear problem along with 2D ???? problem will be explored. Upon doing so, it will become clear that when coupling Deflation with an adaptive mesh method, multiple solutions of singularly perturbed problems can be found efficiently. Furthermore, the examples in this paper will display the possibility that the adaptive mesh method may assist Deflation in finding new solutions.

2 An Adaptive Mesh Method in 1D

To adapt a mesh in 1D we use *de Boor's Method* as described in [Huang and Russell]. Such method makes use of the *Equidistribution Principle* to adjust the mesh used to approximate a solution to a given problem. Given a mesh of N points and a continuous function $M = M(x) > 0$ on a bounded interval $[a, b]$, an equidistributed mesh is a mesh $T_h : x_0 = a < x_1 < \cdots < x_{N-1} = b$ such that the area under $M(x)$ is the same for every subinterval. In particular, T_h satisfies

$$\int_{x_0}^{x_1} M(x)\,dx = \cdots = \int_{x_{N-2}}^{x_{N-1}} M(x)\,dx \tag{1}$$

or equivantly,

$$\int_{a}^{x_j} M(x)\,dx = \frac{(j-1)}{(N-1)}\sigma \text{ for } j = 0, \ldots, N-1 \tag{2}$$

where

$$\sigma = \int_a^b M(x)\,dx. \tag{3}$$

The chosen $M(x)$ is called the *mesh density function* in [Huang and Russell]. It has been proven in [Huang and Russell] that if a given mesh density function is strictly positive for $N > 0$ points, there exists a unique T_h.

To solve a differential equation numerically the problem is discretized. Therefore, the integrals presented in (1) must be approximated. Consequently, the equidistributing mesh is to be approximated using numerical methods [Huang and Russell]. As previously mentioned, the numerical method used in 1D is *de Boor's Method* which computes an equidistributed mesh based on a piecewise constant approximation of $M(x)$. In particular, assume that $M(x)$ is known on some arbitrary mesh $T_* : y_0 = a < y_1 < ... < y_{N-1} = b$. This mesh may be the mesh used before any adaptation or it may be the current approximation of an equidistributed mesh [Huang and Russell]. On $T_*, M(x)$ is approximated by the piecewise constant function

$$m(x) = \begin{cases} \frac{1}{2}\left(M(y_0) + M(y_1)\right), & \text{for } x \in [y_0, y_1] \\ \frac{1}{2}\left(M(y_1) + M(y_2)\right), & \text{for } x \in (y_1, y_2] \\ \cdots & \\ \frac{1}{2}\left(M(y_{N-2}) + M(y_{N-1})\right), & \text{for } x \in (y_{N-2}, y_{N-1}]. \end{cases} \tag{4}$$

Letting

$$\mathrm{M}(y_j) = \int_a^x m(x)\,dx = \sum_{i=1}^{j}(y_i - y_{i-1})\frac{M(y_i)+M(y_{i-1})}{2},\ j = 1, ..., N-1,$$

ξ_j represent the jth mesh point on a uniform mesh of N points, and k be the index such that $\mathrm{M}(y_{k-1}) < \xi_j \mathrm{M}(b) \leq \mathrm{M}(y_k)$, the updated jth meshpoint is computed as

$$x_j = y_{k-1} + \frac{2(\xi_j \mathrm{M}(b) - \mathrm{M}(y_{k-1}))}{M(y_{k-1}) + M(y_k)} \tag{5}$$

DO I NEED MORE INFO HERE??

Although we only require that $M(x)$ is strictly positive, there are some popular choices for $M(x)$. One would like to choose $M(x)$ so that equidistribution places more points in the layers of the solutions of singularly perturbed problems. For example, [Kopteva] chooses the arclength formula

$$M(x,u) = \sqrt{1+|u_x|^2} \tag{6}$$

This function is chosen because at regions of large variation in $u(x)$, the derivative of u with respect to x will be large. Consequently, (6) will be large and more points will be placed in these regions where the solution varies greatly so that the mesh can satisfy the *Equidistribution Principle.* Another popular choice is the curvature formula

$$M(x,u) = \left(1+|u_{xx}|^2\right)^{\frac{1}{4}}. \tag{7}$$

[Huang and Russell] explore mesh density functions that minimize particular error bounds. These include the optimal mesh density function for an error bound computed using piecewise constant interpolation,

$$M(x,u) = \left(1+\frac{1}{a}|u_x|^2\right)^{\frac{1}{3}}, a = \left[\frac{1}{b-a}\int_a^b |u_x|^{\frac{2}{3}}\,dx\right]^3, \tag{8}$$

the optimal mesh density function for an error bound computed using the L^2 norm,

$$M(x,u) = \left(1+\frac{1}{a}|u_{xx}|^2\right)^{\frac{1}{5}}, a = \left[\frac{1}{b-a}\int_a^b |u_{xx}|^{\frac{2}{5}}\,dx\right]^5, \tag{9}$$

and the optimal mesh density function for an error bound computed using the H^1 semi-norm,

$$M(x,u) = \left(1+\frac{1}{a}|u_{xx}|^2\right)^{\frac{1}{3}}, a = \left[\frac{1}{b-a}\int_a^b |u_{xx}|^{\frac{2}{3}}\,dx\right]^3. \tag{10}$$

In the numerical experiments to follow, the structure of the solutions along with numerical experiments are used to choose the best of these mesh density functions for each problem.

3 A Linear Example

To explore solving singularly perturbed problems with adaptive meshing, we consider the reaction-diffusion problem given in [MacLachlan] as

$$-\epsilon^2 u'' + u = e^x \text{ on } (0,1),\ u(0) = u(1) = 0. \tag{11}$$

The analytical solution of this problem is known and is

$$u(x) = \frac{\left(1 - e^{1-\frac{1}{\epsilon}}\right) e^{\frac{-x}{\epsilon}} + \left(e - e^{\frac{-1}{\epsilon}}\right) e^{\frac{x-1}{\epsilon}}}{(\epsilon^2 - 1)\left(1 - e^{\frac{-2}{\epsilon}}\right)} - \frac{e^x}{(\epsilon^2 - 1)}. \tag{12}$$

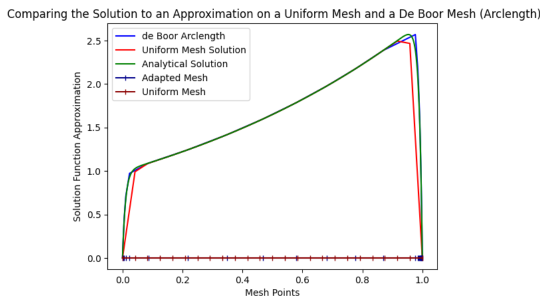
For this problem, the differential equation is discretized on an initial mesh using a finite difference method. The resulting linear system is then solved using Numpy's solver to yield the first approximation of the solution. Once the first approximation of the solution is found, the chosen mesh density function is computed and is used to compute a new mesh using (5). An iteration is set between the computation of a new solution approximation and the computation of a new mesh until the meshes converge (until $||oldmesh - newmesh||_2 < tol$ where tol is a user specified tolerance). Once the stopping criteria is met, the final mesh and the solution approximation on that mesh are returned as our solution approximation for the problem.

This linear example was solved at $\epsilon = 0.01$ on different meshes of $N = 25$ points. In particular, the problem was solved on a uniform mesh along with meshes adapted using (6), (7), (8), (9), and (10). Furthermore, the problem was solved on a Shishkin mesh [NEED REF]. The tolerance set for the convergence of meshes is $tol = 10^{-8}$. The solutions approximated on each mesh are compared to the analytical solution given in (12) computed on a fine mesh ($N = 1000$). The resulting plots are given in ????.

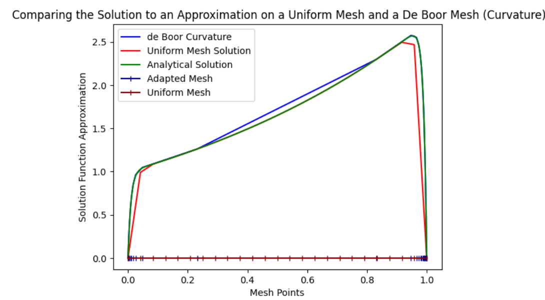
One can conclude from these plots that the solutions approximated on the meshes found using (9) provide better solution approximations than the solutions approximated on a uniform mesh along with the meshes found using (6), (7), (8), and (10). Although the solution found using (9) closely resembles the solution found on a Shishkin mesh, the adapted mesh will behave better in situations where the solution has layers that the user may have very little or no knowledge about (such as multiple interior layers). Consequently, meshes computed using de Boor's method and (9) will be used for the 1D example that follows.

4 The Deflation Method

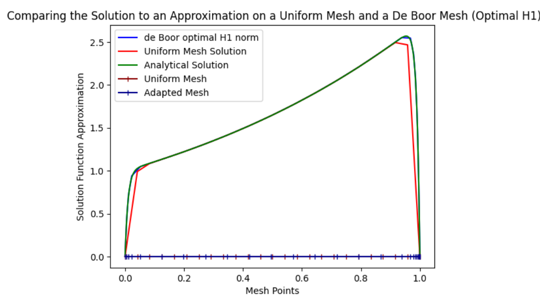
As we transition to nonlinear singularly perturbed problems, we must consider the possibility of multiple solutions. Consequently, we would like to discover as many solutions as we can given a nonlinear singularly perturbed problem. To solve nonlinear 1D singularly perturbed problems, the differential equation



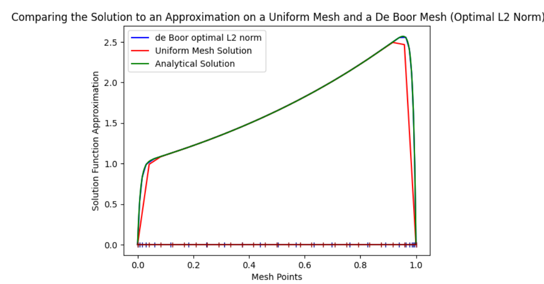
(a) Analyzing the results when (6) is used to compute the mesh



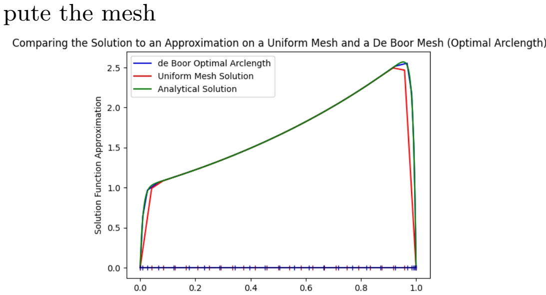
(b) Analyzing the results when (7) is used to compute the mesh



(c) Analyzing the results when (10) is used to compute the mesh



(d) Analyzing the results when (9) is used to compute the mesh



(e) Analyzing the results when (8) is used to compute the mesh

Figure 1: The Results for the Linear Example (11)

is discretized and a solution is approximated using some version of Newton's method. Newton's method requires an initial guess for the solution it is seeking. To find distinct solutions, distinct initial guesses are often used. However, even when numerous initial guesses are used, Newton's method may not find all of the solutions that the user hopes to find. There may also be solutions yet to be discovered that Newton's method cannot find without being given the 'perfect initial guess.' If the user has no knowledge of the structure of the solutions and/or a 'perfect initial guess,' how are they supposed to aid Newton's Method in discovering solutions? Furthermore, even when Newton's method is supplied with multiple initial guesses, each guess may converge to the same solution.

To help discover distinct solutions of the nonlinear systems that represent differential equations, the *Deflation Method* as described in [Original Deflation Paper] is a useful tool. The Deflation Methodology is an extension of the Deflation Method for finding roots of nonlinear equations to finding distinct solutions of nonlinear systems [Original Deflation Paper]. In particular, if a solution of the problem, r, is known, the residual of the original problem can be altered to create a *deflated problem.* To deflate the problem, the residual of the original problem is multiplied by a deflation operator $D(x;r)$. The deflation operator is constructed so that when Newton's method is applied to the deflated problem it will not converge to r [all Original Deflation paper].

For a singularly perturbed problem, the differential equation is written as

$$F(u) = 0$$

where F is the discretized problem. Assuming that a solution r_0 has been found, the deflated problem is constructed as

$$G_1(u) := D(u;r_0)F(u) = 0 \tag{13}$$

As given in [original deflation paper], deflation operators are formulated as

$$D_{p,\alpha}(u;r) = \frac{I}{||u-r||_U^p} + \alpha I \tag{14}$$

where I is the identity operator on the space that F maps to, p is the *power* coefficient, α is the *shift* coefficient, and $||\cdot||_U$ denotes the norm of the solution space [all Maclachlan and Farrell spdes]. In this paper, $||\cdot||_U$ is the grid-2 norm given in [Leveque].

If $n+1$ solutions have been found, $r_0, r_1, \ldots, r_n$, then the deflation operator used to attempt to discover solution $n+2$ is

$$G_n(u) := D_{p,\alpha}(u;r_n) \ldots D_{p,\alpha}(u;r_0).$$

These deflated problems are constructed so that any solution to the deflated problem is a solution of the undeflated problem, any undeflated solution of the original problem is a solution to the deflated problem, and if Newton converges from an initial guess, it converges to an undeflated solution [ALL MacLachlan and Farrell spdes]. Consequently, one can pair the Deflation Methodology with

Newton's method to discover multiple solutions of a singularly perturbed problem using a single initial guess. To help discover more solutions than one can find with a single initial guess, a pool of initial guesses may be used so that solutions with radically different behaviour may be captured with the help of Deflation.

One may wonder some of the advantages and disadvantages of the Deflation Method. The most obvious advantage is the method's ability to potentially discover multiple solutions by repeatedly initializing Newton's method from the same initial guess and deflating solutions as they are found. However, if more solutions can be found by repeatedly initializing Newton's method with numerous initial guesses, is Deflation any better than Newton's method alone? To answer such question, consider the case in which two initial guesses are used but both guesses lead Newton's method to converge to the same solution. In this case, if the solution found by one of those initial guesses is deflated out of the system, then that found solution cannot be found again when we initialize Newton's method from the second initial guess. Consequently, we may be able to discover a new solution. Therefore, the Deflation method is a useful tool for finding multiple solutions. Although the Deflation Method can aid in discovering new solutions, there is no guarantee that Deflation will discover new solutions. Furthermore, there is no guarantee that Deflation will discover all the solutions of a problem [Deflation for semismooth]. There is no standard way to choose the deflation parameters p and α. Rather, the parameters are often chosen through experimentation. It has been shown in [NEED PAPER] that the choices of p and α have a large effect on how many solutions are found. Therefore, experimenting to find p and α is important for helping capture as many solutions as possible via Deflation. As previously discussed, when the parameter, ϵ, in a singularly perturbed problem becomes smaller, the layers in the solutions become more localized. Consequently, it becomes harder to capture the solutions of the problem. To aid the Deflation Method in discovering solutions at smaller values of ϵ, Deflation may be paired with *Continuation*. Such pairing is used in this paper and will be explained in the sections to come.

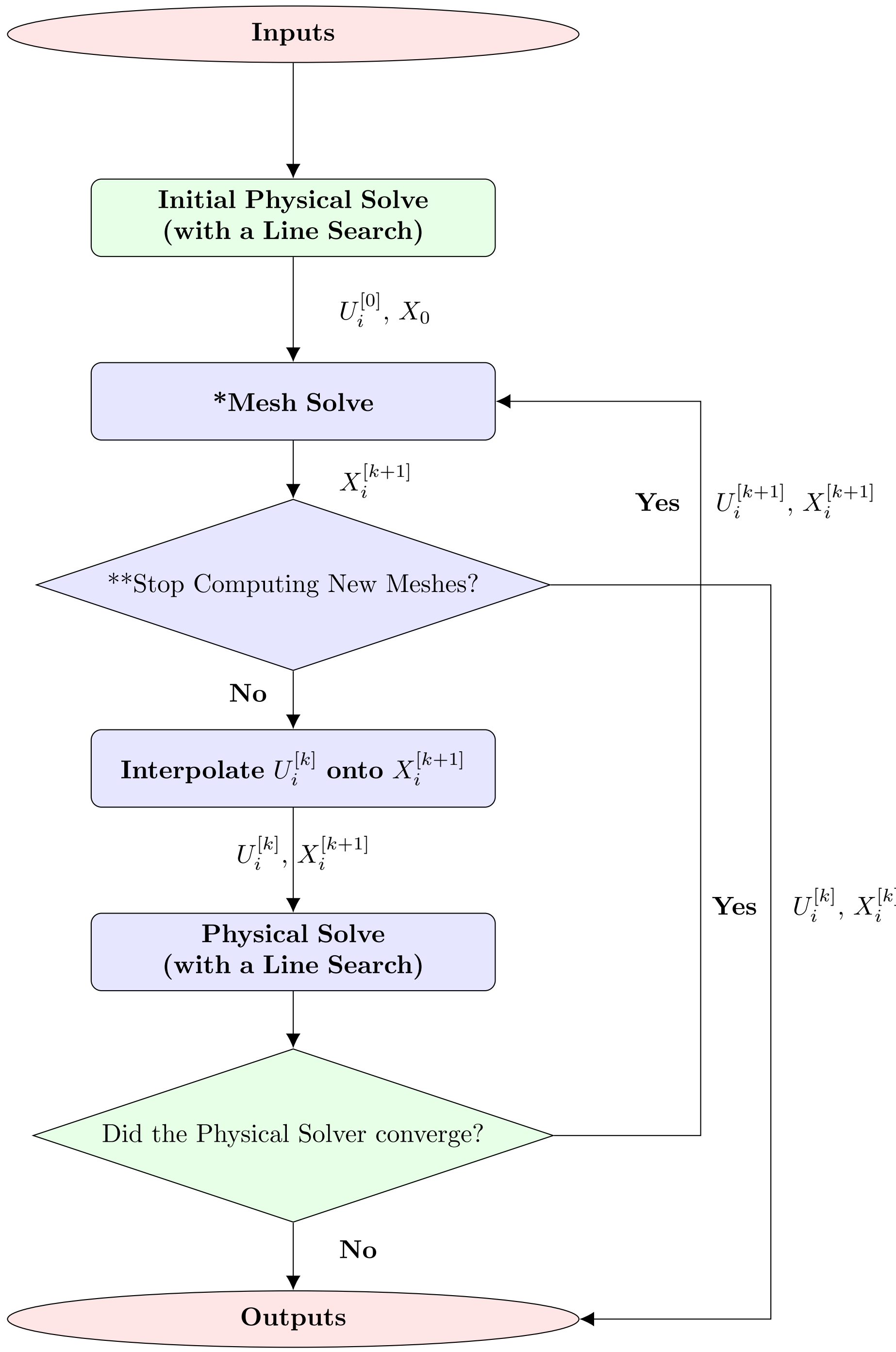
5 An Algorithm For Capturing Multiple Solutions in 1D

To capture multiple solutions of nonlinear singularly perturbed problems in 1D, we first discretize the nonlinear problem and construct the nonlinear system $F(u) = 0$. To solve such system numerically for an approximation of u, we use Newton's method which makes use of a line search [?] to determine a possible damping parameter. This pairing of Newton's method with a line search will be called the *Physical Solver*. To approximate a solution on an adapted mesh, the Physical Solver is paired with de Boor's Method. Such pairing allows us to compute a solution to the nonlinear singularly perturbed problem on an adapted mesh. Similar to the scheme used to solve (11), an iteration is set between the Physical Solver and the chosen adaptive mesh method (de Boor's method). In particular, given an initial guess U_0 on an initial mesh X_0, the Physical Solver is used to compute the first solution approximation, $U_i^{[0]}$, on X_0. Here the index i indicates the ith solution being found for the problem and the superscript represents the kth approximation of that ith solution. Once this first solution approximation is found, $U_i^{[0]}$ and X_0 are passed to the adaptive mesh method where X_0 is adapted to become $X_i^{[1]}$ using (5). Before updating the solution approximation using the Physical Solver, $U_i^{[0]}$ is interpolated onto $X_i^{[1]}$ using piecewise linear interpolation. Then the interpolated solution function approximation is passed to the Physical Solver as an initial guess for Newton's method. This iteration between the Physical solver and de Boor's Method is repeated until either the meshes converge, the iteration count reaches a set maximum, or Newton's method diverges. If the meshes converge, the most recent solution approximation and the corresponding mesh is returned as the final solution approximation. If either the iteration count reaches a maximum or Newton's method diverges, the algorithm is considered to have failed to find a solution. This algorithm is called *Algorithm (A)* and is illustrated in 5.1.

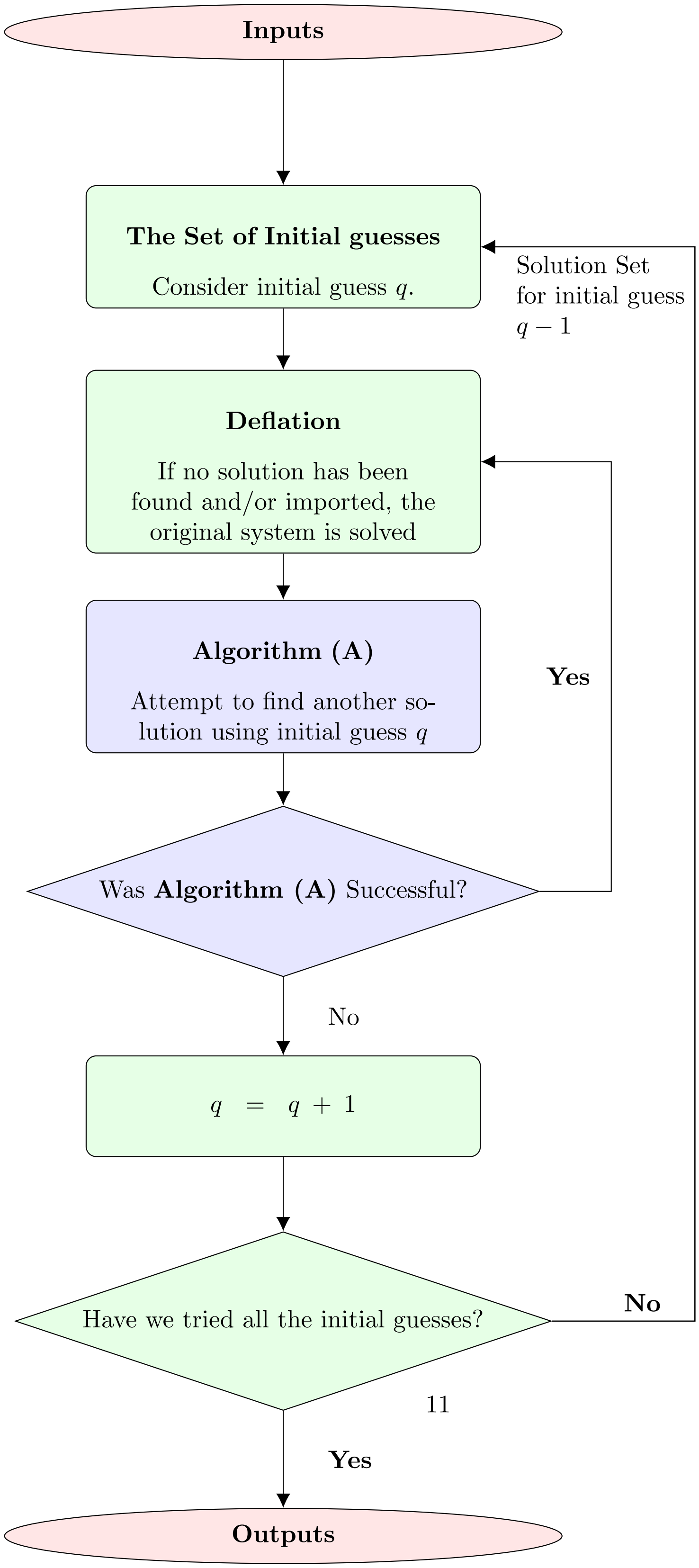
To capture multiple solutions of a nonlinear singularly perturbed problem, we introduce Deflation into Algorithm (A). Since multiple initial guesses can aid in finding more solutions than just a single guess, multiple initial guesses will be used. In particular, given an initial guess, any solutions that have been found thus far at the current parameters are deflated out of the problem. The resulting Deflated problem is then passed to Algorithm (A) in an attempt to compute a new solution. If Algorithm (A) computes a new solution then that found solution is deflated out of the problem and Algorithm (A) is used on the resulting Deflated problem. If Algorithm (A) fails to find a new solution, we move on to the next initial guess and repeat the process by deflating out any solutions found at the previous initial guess. This coupling of Deflation and Algorithm (A) is repeated until we run out of initial guesses. Once the guesses have all been used, the current set of solutions and their corresponding meshes are returned. This process is illustrated further in 5.2 where it is called *Algorithm (B)*.

As previously discussed, for small values of ϵ the layers of the solutions of singularly perturbed problems become more localized. Consequently, the solutions of singularly perturbed problems become harder to capture as ϵ is decreased. In order to capture solutions of singularly perturbed problems at smaller values of ϵ, we will pair *Parameter Continuation* with Algorithm (A) and Algorithm (B). Such pairing of continuation and Deflation has been shown to be useful and can be further explored [HERE].For our numerical experiments we begin with an initial parameter, ϵ_0, a final parameter, ϵ_f, and the desired change in parameter, $\Delta\epsilon$. Starting with ϵ_0, a set of solutions is computed using Algorithm (B). Once a set of solutions is computed at ϵ_0, ϵ is decreased and becomes $\epsilon_1 = \epsilon_0 - \Delta\epsilon$. Using the solutions found at ϵ_0 as initial guesses, we attempt to find the corresponding solution set at ϵ_1 using Algorithm (A). The solutions found at ϵ_1 are then given to Algorithm (B) where they are deflated out of the problem so that we can search for more solutions at ϵ_1. This process of decreasing ϵ, using the solution set for the previous ϵ as initial guesses to find the solutions at the decreased ϵ, and then deflating those solutions out of the problem so that we can look for more solutions, is repeated until we reach ϵ_f. Once we reach that final epsilon, the solutions and their meshes are returned for each ϵ. This process comprises what we will call *Algorithm (C)* which is described further in 5.3. This algorithm makes use of the Algorithm (A) and (B) to capture multiple solutions at smaller values of ϵ. The key components and structure of this algorithm were created and described in [LIQUID CRYSTALS] and [CONTINUED DEFLATION]. The algorithms given in [LIQUID CRYSTALS] and [CONTINUED DEFLATION] use uniform meshes and/or mesh refinement. Consequently, our addition to these algorithms is the adaptive mesh method. We shall explore the success of our version of the algorithms using a 1D semilinear singularly perturbed reaction diffusion problem. Then we will construct an analogous algorithm in 2D.

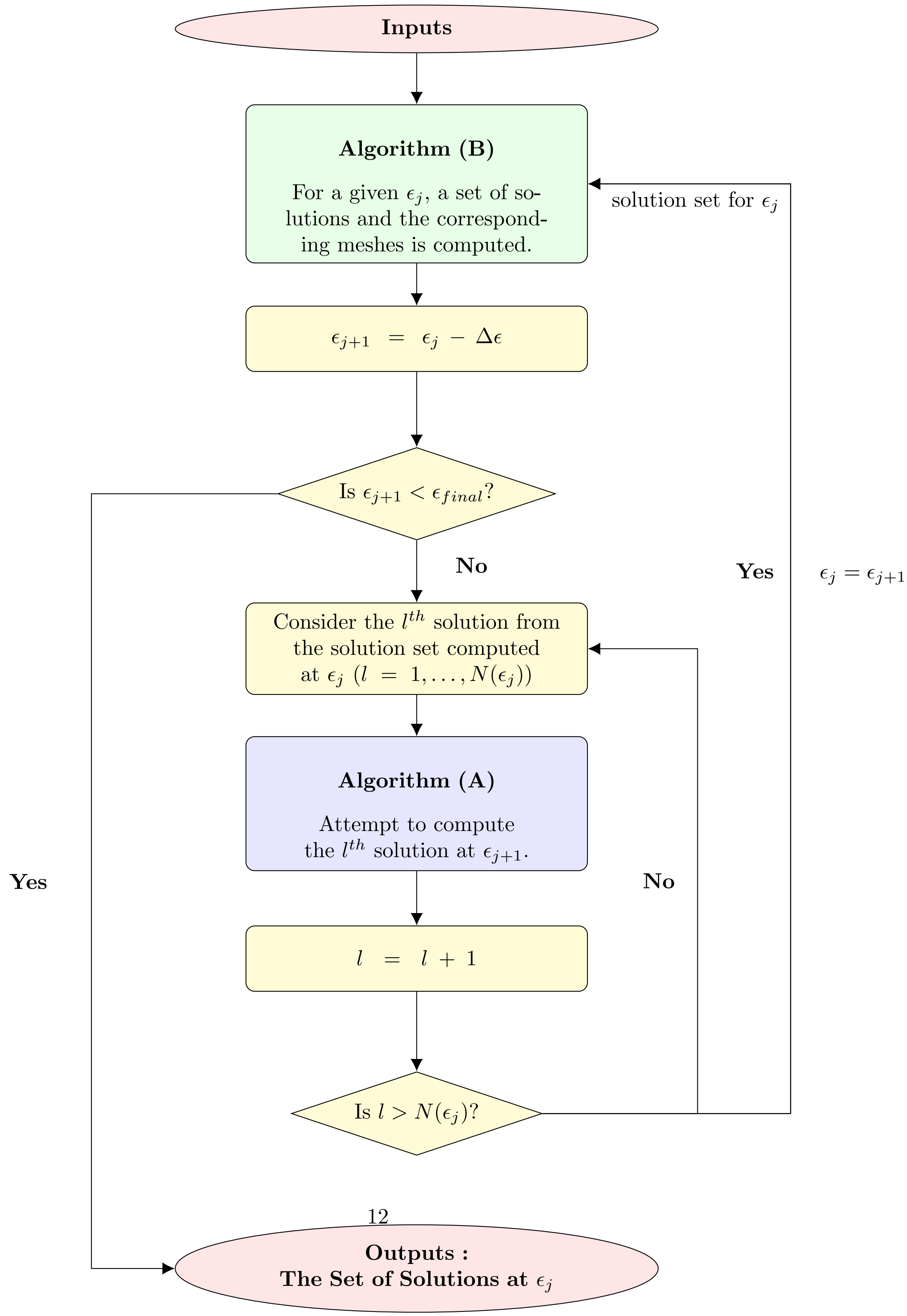
5.1 Algorithm (A) : How to Compute a Single Solution



5.2 Algorithm (B) : How to Compute Multiple Solutions



5.3 Algorithm (C) : Continued Deflation



...

6 Extending To 2D

...

7 A 2D Example

...