A bootstrapping approach for computing multiple solutions of differential equations

Wenrui Hao* Jonathan D. Hauenstein† Bei Hu‡ Andrew J. Sommese§

November 23, 2012

Abstract

Discretizing systems of nonlinear algebraic differential equations yield polynomial systems. When using a fine discretization, the resulting polynomial system is often too large to solve using a direct solving approach. Our approach for solving such systems is to utilize a homotopy continuation based method arising from domain decomposition. This method solves polynomial systems arising from subdomains and then uses homotopy continuation to build solutions of the original polynomial system. We illustrate this approach on one- and two-dimensional problems.

Keywords: Domain decomposition, homotopy continuation, differential equations, multiple solutions, numerical algebraic geometry.
AMS Subject Classification: 65N55, 65M22,65H10,65L10.

1 Introduction

A common approach for approximating solutions to a system of nonlinear differential equations is to discretize and solve the resulting nonlinear system of equations. When the nonlinear equations are algebraic, the resulting system is a system of polynomials. Even though modern numerical codes for computing all solutions of a polynomial system can yield new solutions [5], the systems of polynomials (arising even from very sparse grids) are usually much too large for direct solution by these codes. The realization underlying this article is that domain decomposition gives guidance on how

*Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556 (whao@nd.edu). This author was supported by the Duncan Chair of the University of Notre Dame and NSF grant DMS-0712910.

†Department of Mathematics, North Carolina State University, Raleigh, NC 27695 (hauenstein@ncsu.edu, `www4.ncsu.edu/~jdhauens`). This author was supported by NSF grant DMS-1262428.

‡Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556 (b1hu@nd.edu, www.nd.edu/∼b1hu).

§Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556 (sommese@nd.edu, www.nd.edu/∼sommese). This author was supported by the Duncan Chair of the University of Notre Dame and NSF grant DMS-0712910.

to "bootstrap" from the solutions of many small polynomial systems to often many solutions of a polynomial system arising from a fine discretization based on a realistic grid. In summary, we focus on computing solutions to the polynomial system resulting from a realistic discretization, but do not discuss how one would obtain such a realistic discretization from their given system of differential equations.

A technique was described in [1] in which a very coarse initial grid is used (one can consider finite differences or other methods). After the system arising from the coarse discretization was solved and spurious solutions filtered out, the mesh is refined in a smooth way. The resulting solutions are used as the starting points for a homotopy moving to a finer discretization. This process can be repeated until the set of real solutions has stabilized. If one now seeks more accurate solutions, the available solutions can be extrapolated to finer meshes and used as starting points for another homotopy, which is now over the real numbers.

Domain decomposition is a powerful tool for devising parallel methods to solve partial differential equations. The basic idea of domain decomposition is to first decompose the domain into subdomains. Then, each subdomain is solved independently in parallel. The solutions from the subdomains are then merged together to form solutions of the original problem. One issue with this approach is computing approximate values to the subdomain boundary points. Since this is a major difficulty, there is a rich literature (see, for example, [4, 7, 9] and the references therein) which constructs schemes for approximating them using a time dependent systems of PDEs, typically for systems of parabolic differential equations.

In this article, we introduce a new homotopy method based on domain decomposition which we call the *bootstrapping method*. This method computes multiple solutions to discretized systems of nonlinear algebraic differential equations and is naturally parallelizable. We introduce this bootstrapping method for solving problems consisting of one- and two-space dimensions in Sections 2 and 4, respectively. Sections 3 and 5 examine the algorithm and provide numerical examples. Section 6 summarizes the results.

2 One-dimensional bootstrapping

The fundamental idea of domain decomposition and the bootstrapping approach is that the polynomial systems resulting from discretizing on the whole domain and each subdomain are structurally the same and only differ by values of parameters. In short, this means that coefficient-parameter continuation can be repeatedly used to solve such systems. To demonstrate this in the one-dimensional case, we will consider Laplace's equation. Suppose that $u : [0, 1] \to \mathbb{R}$ solves

$$\begin{cases} u_{xx} &= f(u) \quad \text{on } (0,1), \\ u(0) &= c_0, \\ u(1) &= c_1, \end{cases} \tag{2.1}$$

where $f(u)$ is a polynomial function of u, and $c_0, c_1 \in \mathbb{R}$. Consider discretizing the system using $N+1$ grid points located at $x_i = i/N$ for $i = 0, \ldots, N$ with a second order central difference scheme to approximate u_{xx}. The resulting polynomial system is

$$F_H(u_0, \ldots, u_N) = \begin{bmatrix} f(u_i) - H^{-2}(u_{i+1} - 2u_i + u_{i-1}), & 1 \le i \le N-1 \\ u_0 - c_0 \\ u_N - c_1 \end{bmatrix} \tag{2.2}$$

where $H = 1/N$ and u_i is considered to be an approximation of $u(x_i)$. If we are given that $u(x_i) = d_i$ and $u(x_{i+1}) = d_{i+1}$, consider discretizing each subinterval $[x_i, x_{i+1}]$ using $M+1$ grid points located at $x_{i,j} = x_i + j/(NM)$ for $j = 0, \ldots, M$ with a second order central difference scheme to approximate u_{xx}. The resulting polynomial systems is obtained from F by simply modifying the parameters N, c_0, and c_1, namely

$$G_{i,h}(u_{i,0}, \ldots, u_{i,M}) = \begin{bmatrix} f(u_{i,j}) - h^{-2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}), & 1 \le j \le M-1 \\ u_{i,0} - d_i \\ u_{i,M} - d_{i+1} \end{bmatrix} \tag{2.3}$$

where $h = (NM)^{-1}$ and $u_{i,j}$ approximates $u(x_{i,j})$. As mentioned in the Introduction, obtaining the values of d_i and d_{i+1} is a major difficulty.

The goal of the bootstrapping approach is to compute solutions of $F_{H'} = 0$ for a sufficiently small value of H', i.e., a fine discretization using large number of grid points, by solving $F_H = 0$ for large values of H, i.e., a coarse discretization using a small number of grid points, and building from solutions to the subsystems $G_{i,h} = 0$ using appropriately selected h and boundary values d_i and d_{i+1}.

We will use the values obtained from solving $F_H = 0$ for the boundary values of the subsystems. That is, the domain decomposition system under consideration is

$$F_{h,H}(\mathbf{u}) = \begin{bmatrix} f(u_{i,j}) - h^{-2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}), & 0 \le i \le N-1, 1 \le j \le M-1 \\ f(u_{i,0}) - H^{-2}(u_{i+1,0} - 2u_{i,0} + u_{i-1,0}), & 1 \le i \le N-1 \\ u_{0,0} - c_0 \\ u_{N,0} - c_1 \end{bmatrix} \tag{2.4}$$

where $H = N^{-1}$, $h = (NM)^{-1}$, $u_{i,M} = u_{i+1,0}$, and

$$\mathbf{u} = \left[\; u_{0,0}, \cdots, u_{0,M-1}, \cdots, u_{N-1,0}, \cdots, u_{N-1,M-1}, u_{N,0} \; \right].$$

The basic idea is that solutions to $F_{h,H} = 0$ can be computed by first solving $F_H = 0$ for small values of $N = H^{-1}$ and use the solutions for "boundary conditions" to solve $G_{i,h} = 0$ for small values of $M = (Nh)^{-1}$. Then, solutions to $F_h = 0$ are computed from solutions of $F_{h,H} = 0$ using the homotopy

$$\mathcal{F}(\mathbf{u}, t) = (1-t)F_h(\mathbf{u}) + \gamma t F_{h,H}(\mathbf{u}), \tag{2.5}$$

where γ is a random complex number (see [8] for more details). Full details of the bootstrapping method are as follows.

1. For small N, F_H defined by (2.2) can be solved directly where $H = N^{-1}$ yielding discretized solutions $u_{i,0}$ on a coarse grid. We denote the set of solutions as $\mathcal{S}_{N,0}$.

2. Each solution in $\mathcal{S}_{N,0}$ defines Dirichlet "boundary conditions" on each subdomain. Solve $G_{i,h}$ for $i = 0, \ldots, M-1$ where $h = (NM)^{-1}$ and let the set of solutions be $\mathcal{S}_{N,M}$. Typically, M is taken to be small, say $M = 2$ or $M = 3$.

3. Discard unreasonable solutions of $\mathcal{S}_{N,M}$ using a filtering condition. Our filtering condition ensures that the approximation solution is reasonably close to the real solution. Some filtering conditions are following:

   (a) **Variational form:** Let $v \in V_h$ be the set of continuous piecewise linear functions that vanish at $x = 0$ and $x = 1$. Then, the solutions satisfy the form
   $$-\int_0^1 u'v'dx = \int_0^1 f(u)vdx \text{ for all functions } v,$$
   corresponding to the residual $u'' - f(u)$ being orthogonal to the test function space V_h. We can pick a sequence $\{v_n\}_{n=1}^{\infty}$, such as $v_n = \frac{\sin(n\pi x)}{\sqrt{n}}$ or $v_n = \frac{x^n(1-x)}{\sqrt{n}}$, to obtain a series of filtering conditions. (Such a filtering approach is related to finite element methods). We use
   $$\left| \int_0^1 u'v'dx + \int_0^1 f(u)vdx \right| \leq \epsilon \text{ for } \|v\|_{H^1([0,1])} \leq 1,$$
   with a reasonably small ϵ.

   (b) **Bounded condition:** A filter forcing $|u_{i,j}| < K$, where $K > 0$ is a upper bound of the PDE solutions. Clearly, a bounded filter can be applied to PDE systems for which bounds can be estimated.

   (c) **Other conditions:** Other filters can be found in [1], which is derived from known properties of the solutions of the problem at hand, such as derivatives, symmetry, positive or some other easily-detected behavior.

4. By construction, the points in $\mathcal{S}_{N,M}$ are solutions of $F_{h,H}$ which are start points for the homotopy $\mathcal{F}$ defined by (2.5) at $t = 1$. Track the solutions paths defined by $\mathcal{F}$ starting at the points in $\mathcal{S}_{N,M}$. The set of endpoints yields solutions of F_h. If h is sufficiently small, output the solutions. Otherwise, return to step 2 by identifying $\mathcal{S}_{N,M}$ with $\mathcal{S}_{NM,0}$ and updating N and H accordingly.

With the proper modifications of the polynomial systems, this is a general method that can be applied to other problems, including problems with Neumann and mixed boundary conditions.

3 One-dimensional examples

To demonstrate the approach, we apply the bootstrapping method to a modification of (2.1), namely

$$\begin{cases} u_{xx} &= f(u) \quad \text{on } (0,1), \\ u'(0) &= 0, \\ u(1) &= 0. \end{cases} \tag{3.6}$$

In the following examples, the Neumann boundary condition $u'(0) = 0$ is discretized using a second-order one-sided scheme of the form

$$u'(0) \approx \frac{u(2H) - 4u(H) + 3u(0)}{-2H}.$$

The "variational form" filtering condition was used in Example 1 while Example 2 utilized the "bounded condition" filtering since the solutions are bounded by $\sqrt{p}$ from [3].

3.1 Example 1

We consider $f(u) = -\lambda(1 + u^p)$ where $\lambda \geq 0$ is a real parameter and p is a nonnegative integer. Multiplying by u' and integrating over $[0, x]$, we obtain

$$\frac{1}{2}(u')^2(x) + F(u(x)) - F(u_0) = 0, \tag{3.7}$$

where $F(u) = \int_0^u \lambda(1 + s^p)ds$ and $u_0 = u(0)$. Since $u' < 0$ for $x > 0$, we have

$$\int_0^{u(x)} \frac{ds}{\sqrt{F(u_0) - F(s)}} = \sqrt{2}(1 - x). \tag{3.8}$$

By taking $x = 0$, we obtain

$$G(u_0) := \int_0^{u_0} \frac{ds}{\sqrt{F(u_0) - F(s)}} - \sqrt{2} = 0. \tag{3.9}$$

In particular, there exists λ^* such that

(i) for $0 < \lambda < \lambda^*$, there are two solutions,

(ii) the two solutions merge at $\lambda = \lambda^*$, and

(iii) for $\lambda > \lambda^*$, there are no solutions.

Figure 1 presents $G(u_0)$ for different values of λ with $p = 4$. With this setup, one can verify that $\lambda^* \approx 1.30107$. Exact values of the real solutions can be obtained from (3.8) which we compare with the numerical bootstrapping method. We implemented this method utilizing Bertini [2], which is a software package for solving polynomial system and tracking homotopy paths. To solve this problem, we used 12 computing

nodes each containing two Xeon 5410 processors running 64-bit Linux, i.e., each node consists of 8 processing cores.

This computation yielded two real solutions for $\lambda = 1.2$ and one solution for $\lambda = \lambda^*$, which are displayed in Figure 2. Table 1 lists the number of ODE solutions, N, M, numerical errors, and the time required for the computation.

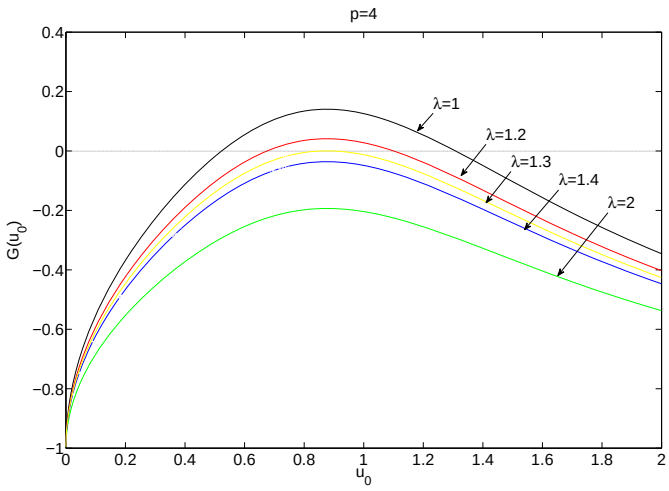

Figure 1: $G(u_0)$ for $p = 4$

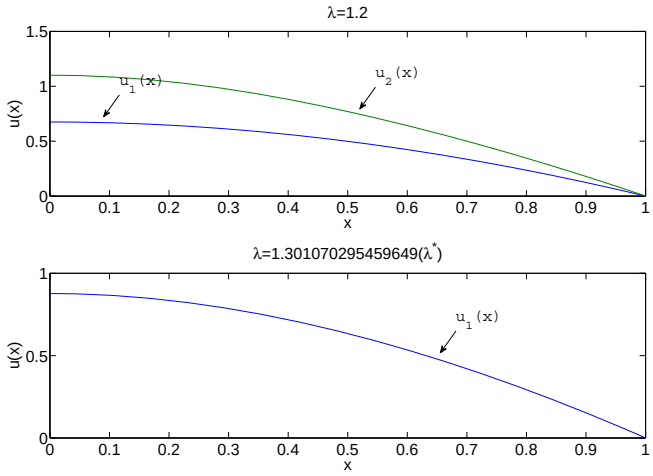

Figure 2: Two solutions for $\lambda = 1.2$ and unique solution for $\lambda = \lambda^*$

Table 1: Summary of solutions

λ	(N, M)	# solns	# $\mathcal{S}_{N,M}$	1^{st} soln Err.	2^{nd} soln Err.	Time
1.2	$(7, 3)$	2	4	1.105364e-4	3.714641e-4	4m5s
	$(21, 2)$	2	10	3.236483e-5	1.133685e-4	31m16s
	$(42, 2)$	2	16	8.384076e-6	3.029060e-5	5h39m58s
λ^*	$(7, 3)$	1	2	1.674890e-4		1m56s
	$(21, 2)$	1	6	3.862634e-5		14m46s
	$(42, 2)$	1	10	1.087360e-5		2h45s42s

3.2 Example 2

We next consider $f(u) = -\frac{\pi^2}{4}u^2(u^2 - p)$ where $p \geq 0$ is a real parameter [3]. We utilized the shooting method to confirm our results regarding the number of solutions. In particular, for each β, let $v_\beta(t)$ denote the solution of the initial value problem

$$\begin{cases} -v_{xx} &= \frac{\pi^2}{4}v^2(v^2 - p) \quad \text{on } (0, 1), \\ v'(0) &= 0, \\ v(0) &= \beta. \end{cases} \tag{3.10}$$

For each p, the goal is to compute how many values of β for which $v_\beta(1) = 0$. Clearly, for each such β, $v_\beta(t)$ solves (3.6). With $F(\beta) = v_\beta(1)$. Figure 3 plots F for $p = 1, 8, 10$. This clearly shows that F has 2, 4, and 8 roots, respectively.

Following the same setup utilized in Section 3.1, the bootstrapping method was used to approximate the solutions for these values of p. This computation produced the same number of solutions as the shooting method. Figure 4 plots these solutions and Table 2 summarizes the data of this computation.

Table 2: Summary of solutions

p	(N, M)	# solns	# $\mathcal{S}_{N,M}$	Time
1	(10,3)	2	48	1h23m
	$(30, 3)$	2	19	12m23s
8	$(10, 3)$	4	120	1h57m
	$(30, 3)$	4	48	27m56s
10	$(10, 3)$	12	490	3h56m
	$(30, 3)$	8	108	56m42s

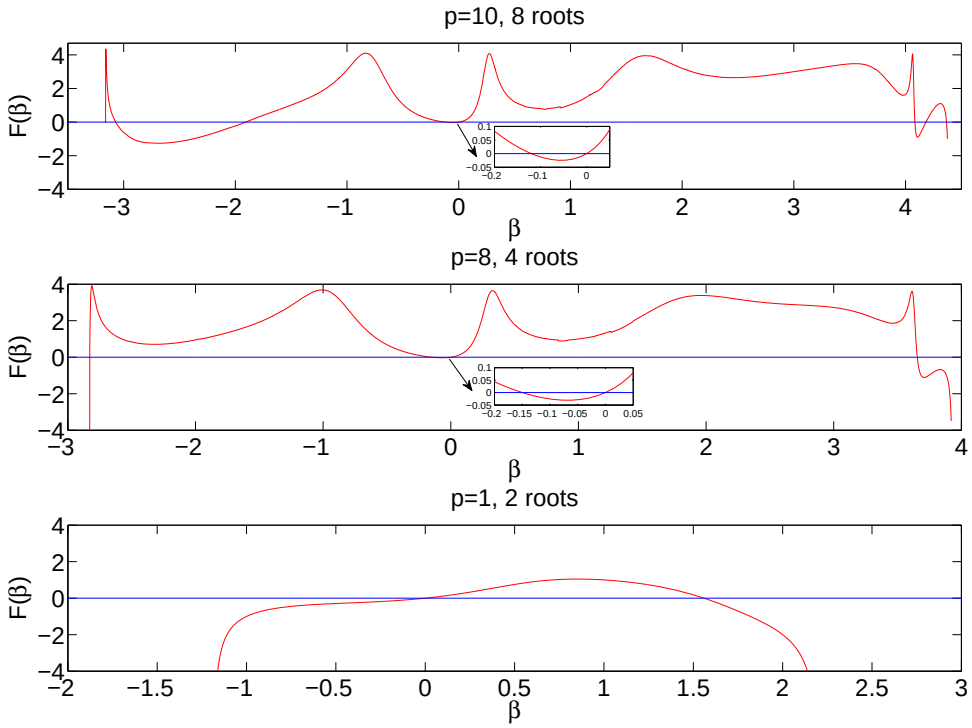

Figure 3: Plots of F using the shooting method

4 Two-dimensional bootstrapping

We extend the bootstrapping method from Section 2 to two-dimensional problems by considering Laplace's equation on the unit square. Let $\Omega = (0,1) \times (0,1)$ and suppose that $u(x,y)$ defined on the unit square $\overline{\Omega}$ solves

$$\begin{cases} \Delta u &= f(u) & \text{on } \Omega, \\ u(x,y) &= g(x,y) & \text{on } \partial\Omega. \end{cases} \tag{4.11}$$

Analogous to Section 2, we decompose $\overline{\Omega}$ into equal subdomains and then decompose each subdomain. Let N_x and N_y be positive integers with $H_x = \frac{1}{N_x}$ and $H_y = \frac{1}{N_y}$, and grid points $(x_{i_1}, y_{i_2}) = (i_1 H_x, i_2 H_y)$. The grid points in each domain $[x_{i_1}, x_{i_1+1}] \times [y_{i_2}, y_{i_2+1}]$ are

$$(x_{i_1,j_1}, y_{i_2,j_2}) = (i_1 H_x + j_1 h_x, i_2 H_y + j_2 h_y)$$

where $h_x = \frac{1}{N_x M_x}$ and $h_y = \frac{1}{N_y M_y}$ with positive integers M_x and M_y. To simplify notation, we will write $(i,j) = ((i_1,i_2),(j_1,j_2))$. As before, the values $u_{i,j}$ will be

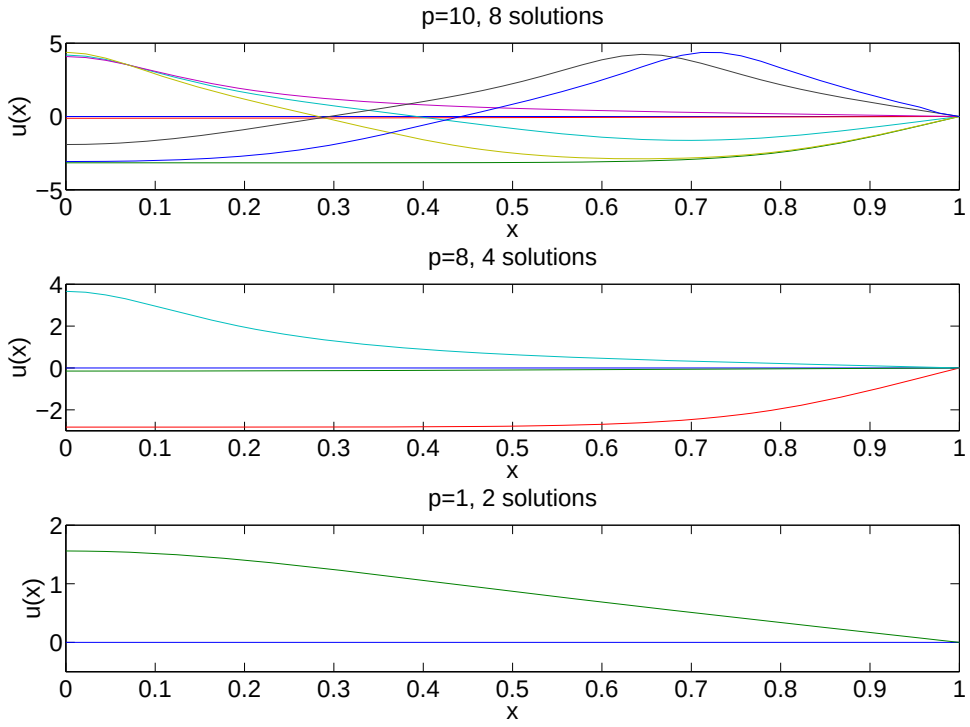

Figure 4: Solutions for different values of p

numerical approximations of $u(x_{i_1,j_1}, y_{i_2,j_2})$ obtained via solving a polynomial system resulting from discretization.

The first step of the bootstrapping homotopy method solves the polynomial system arising from the discretization with respect to H_x and H_y. For the grid displayed in Figure 5, this computes approximations at the red star points.

The second step uses the solutions computed in Step 1 to setup "boundary conditions" for the subdomains. Consider the subdomain $[x_i, x_{i+1}] \times [0, 1]$ with M_x and N_y points in the x and y directions, respectively. For the grid displayed in Figure 5, solving on these grids yield approximations at the green circle points. Now, consider the subdomain $[0, 1] \times [y_i, y_{i+1}]$ with $N_x M_x$ and M_y points in the x and y directions, respectively. For the grid displayed in Figure 5, solving on these grids yield approximations at the black square points.

The third step applies a filter to discard unreasonable solutions. The fourth step uses a homotopy to compute approximations on $\overline{\Omega}$ with $N_x M_x$ and $N_y M_y$ grid points in the x and y directions, respectively, using the polynomial system and approximations constructed in the second step.

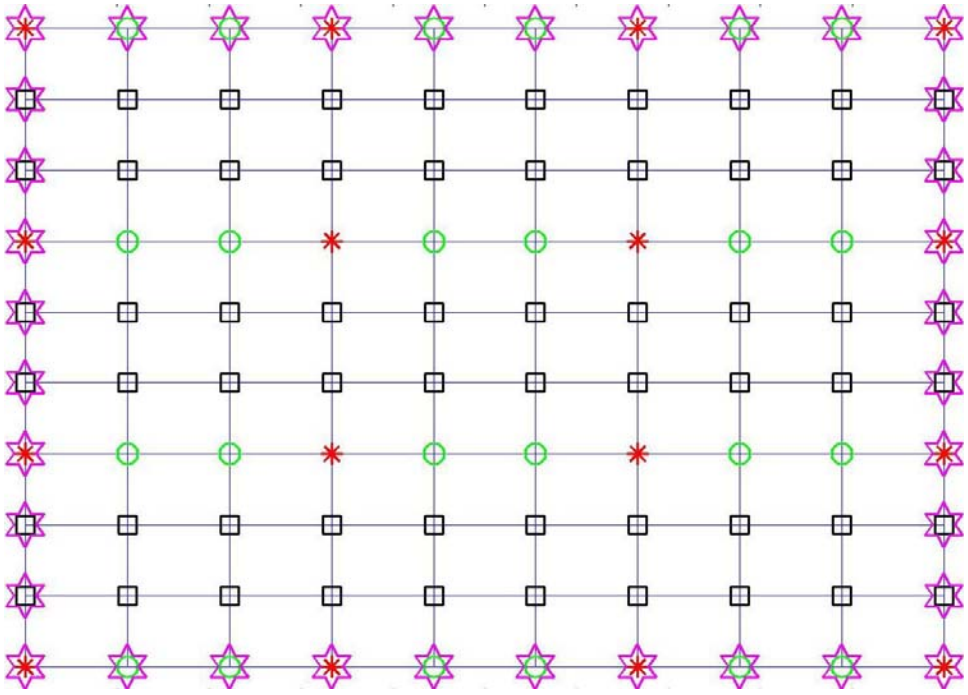

Figure 5: Mesh grid with $N_x = N_y = M_x = M_y = 3$

5 Two-dimensional example

We apply the two-dimensional bootstrapping method to a tumor model without a necrotic core that was discussed in [6]. Let Ω denote the tumor region, σ denote the concentration of nutrients, p denote the pressure, $\widetilde{\sigma}$ denote the concentration of nutrients needed for sustainability, μ denote the aggressiveness of the tumor, and κ denote the mean curvature. A normalized steady state system of a free boundary problem modeling tumor growth is given by

$$\begin{cases} -\Delta\sigma &= -\sigma & \text{in } \Omega \\ -\Delta p &= \mu(\sigma - \widetilde{\sigma}) & \text{in } \Omega \\ \\ \sigma &= 1 & \text{on } \partial\Omega \\ p &= \kappa & \text{on } \partial\Omega \\ \frac{\partial p}{\partial n} &= 0 & \text{on } \partial\Omega, \end{cases} \tag{5.12}$$

It should be emphasized here that Ω is unknown in advance yielding a free boundary problem. To handle the free boundary, we developed a novel discretization approach to allow the length of the grid to change in coordination with the boundary. That is, let $R(\theta)$ be the length of tumor in the θ direction which changes independently and models the free boundary in this direction. Using the floating grid presented in [6],

we apply a third-order finite difference scheme to setup a discretization of the system (5.12), which yields a polynomial system. In [6], nonradial solutions were computed by using parameter continuation with respect to μ starting from bifurcation branching points. A more difficult question is to compute other nonradial solutions for a given value of μ. The bootstrapping method allows us to compute such solutions with the goal of computing as many as possible.

We built a filtering condition based on the fact that the distance to the boundary in each angular direction, denoted $R(\theta)$, must be positive. This was enforced by requiring $R(\theta) > -\epsilon$. Here we choose $\epsilon = 1$ by taking numerical error into consideration. Using the same floating grids described in [6] along with the difference scheme, we employed the bootstrapping algorithm utilizing Bertini on 20 computing nodes of the type described in Section 3.1.

Let N_R and N_θ denote the number of subdomains in the radial, R, and angular, θ, direction, respectively. Each subdomain was decomposed using M_R and M_θ grid points in the R and θ directions, respectively. For small values of N_R and N_θ, e.g., $N_R = 2$ and $N_\theta = 3$, Bertini computed all solutions of the polynomial system arising from the discretization. This polynomial system has a natural multi-homogeneous structure which we utilized to reduce the bound on the number of isolated solutions over $\mathbb{C}$ from 3^{17} =129,140,163 to 923,440. This was the starting point for the bootstrapping method which built numerical solutions on the whole domain using more grid points. Table 3 lists number of grid points, number of solutions, and time needed for the computation.

Table 3: Summary for solving (5.12)

μ	N_R	N_θ	M_R	M_θ	Num. of solns	Time
3	2	3	3	1	56	1h46m41s
	8	5	1	1	117	2d2h47m50s
2	2	3	3	1	34	1h12m13s
	8	5	1	1	61	23h35m45s
1	2	3	3	1	4	54m34s
	8	5	1	1	6	11h56m23s
0.9	2	3	3	1	2	44m21s
	8	5	1	1	2	7h3m25s
0.8	2	3	3	1	1	22m13s
	8	5	1	1	1	3h46m12s
0.5	2	3	3	1	1	22m11s
	8	5	1	1	1	3h45m50s

After obtaining a non-radial solution using the bootstrapping homotopy method, we utilized higher-order difference schemes and finer grids to refine the solutions. The solutions for the higher-order schemes were computed using a homotopy starting from the solutions obtain via the bootstrapping approach. After refinement, some "fake"

solutions, which arose as artifacts of the discretization, were identified and discarded. For example, with $\mu = 3$, we found 24 solutions using this process which are presented in Figure 6. In this figure, it shows the pressure distribution among the region Ω and the maximum and minimum radius $R(\theta)$. We note that solution 21 presented in Figure 6 is the radial solution for $\mu = 3$. Starting from solutions 2 and 4 using a parameter continuation with respect to μ, we found that these solutions arise from the bifurcation at μ_2 (see [6]) for more details regarding this bifurcation point). Figure 7 shows the solution behavior of these paths starting from solutions 2 and 4 which intersect the radial solution branch at μ_2.

For smaller values of μ, e.g., $\mu = 0.5$ and $\mu = 0.8$, this bootstrapping process only found the radial solution. This suggests that no nonradial solutions exist for these values of μ, which is consistent with the theoretical results regarding this tumor model [6].

6 Conclusion

A bootstrapping approach arising from domain decomposition was presented for computing multiple solutions of differential equations for one- and two-dimensional problems. The computations needed in this method are parallelizable and allow for the computation of solutions of extremely large polynomial systems. The bootstrapping algorithm for three-dimensional case shall be considered in the future along with a detailed analysis of how to pick the starting value of N as well as M in the algorithm. Another line of research is developing and testing filtering conditions on a variety of problems.

References

[1] E.L. Allgower, D.J. Bates, A.J. Sommese, and C.W. Wampler, Solution of Polynomial Systems Derived from Differential Equations, *Computing*, Vol. 76, 1–10 (2005).

[2] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Bertini: Software for numerical algebraic geometry. Available at `www.nd.edu/∼sommese/bertini`.

[3] C. Chen, Z. Xie, Structure of multiple solutions for nonlinear differential equations, *Sci. China Ser. A – Math*, Vol. 47, 172–180 (2004).

[4] C.N. Dawson, Q. Du, and T.F. Dupont, A finite difference domain decomposition algorithm for numerical solution of the Heat equation, *Math. Comp.*, Vol. 57, 63–71 (1991).

[5] W. Hao, J.D. Hauenstein, B. Hu, Y. Liu, A.J. Sommese, and Y.-T. Zhang, Multiple stable steady states of a reaction-diffusion model on zebrafish dorsal-ventral patterning, *Discret. Contin. Dyn. – S*, Vol. 4, 1413–1428 (2011).

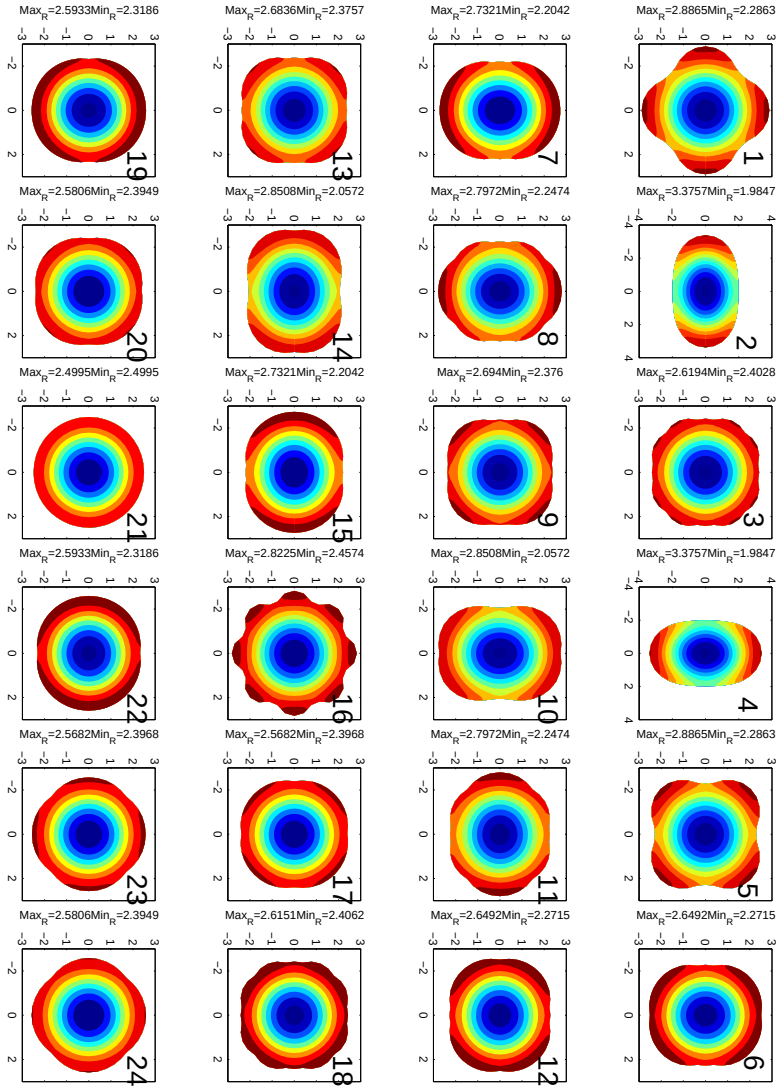

Figure 6: Solutions for $\mu = 3$

[6] W. Hao, J.D. Hauenstein, B. Hu, Y. Liu, A.J. Sommese, and Y.-T. Zhang, Continuation along bifurcation branches for a tumor model with a necrotic core,

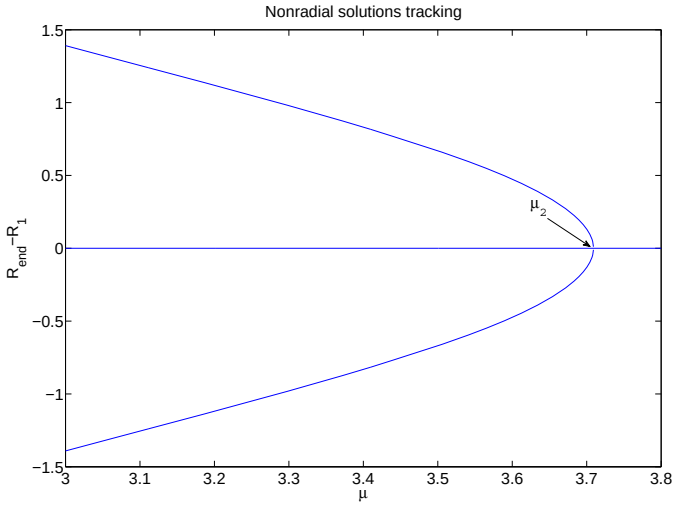

Figure 7: Tracking solution paths back to the bifurcation at μ_2

J. Sci. Comput., Vol. 53, 395–413 (2012).

[7] W. Hao and S. Zhu, Parallel iterative methods for parabolic equations, *Int. J. Comput. Math*, Vol. 86, 431–440 (2009).

[8] A.J. Sommese and C.W. Wampler *Numerical solution of systems of polynomials arising in engineering and science*, World Scientific, Singapore (2005).

[9] S. Zhu, Conservative domain decomposition procedure with unconditional stability and second-order accuracy, *Appl. Math. Comput.*, Vol. 216, 3275–3282 (2010)