Deflation Techniques for Finding Distinct Solutions of Nonlinear Partial Differential Equations

Article *in* SIAM Journal on Scientific Computing · October 2014

DOI: 10.1137/140984798 · Source: arXiv

3 authors, including:

Simon Funke
Simula Research Laboratory
34 PUBLICATIONS **1,017** CITATIONS

SEE PROFILE

DEFLATION TECHNIQUES FOR FINDING DISTINCT SOLUTIONS OF NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS

P. E. FARRELL*, Á. BIRKISSON†, AND S. W. FUNKE‡

Abstract. Nonlinear systems of partial differential equations (PDEs) may permit several distinct solutions. The typical current approach to finding distinct solutions is to start Newton's method with many different initial guesses, hoping to find starting points that lie in different basins of attraction. In this paper, we present an infinite-dimensional deflation algorithm for systematically modifying the residual of a nonlinear PDE problem to eliminate known solutions from consideration. This enables the Newton–Kantorovitch iteration to converge to several different solutions, even starting from the same initial guess. The deflated Jacobian is dense, but an efficient preconditioning strategy is devised, and the number of Krylov iterations are observed not to grow as solutions are deflated. The power of the approach is demonstrated on several problems from special functions, phase separation, differential geometry and fluid mechanics that permit distinct solutions.

Key words. deflation, Newton's method, distinct solutions, continuation.

AMS subject classifications. 65N30, 65N35, 65H99, 35B32

1. Introduction.

Nonlinear problems may permit nontrivial distinct solutions. This paper is concerned with a computational technique, called *deflation*, for finding several distinct solutions of nonlinear (systems of) partial differential equations.

Historically, the first application of related techniques was to finding distinct roots of scalar polynomials [46, pp. 78]. Let $p(x)$ be a scalar polynomial, and let $x^{[0]}, x^{[1]}, \ldots, x^{[n]}$ be roots of p identified with some iterative algorithm, such as Newton's method [20]. Then further roots of p may be found by considering the *deflated* function

$$q(x) = \frac{p(x)}{\prod_{i=0}^{n}(x - x^{[i]})},$$

and applying the same iterative algorithm to q.[1]

Brown and Gearhart [13] extended this deflation approach to systems of nonlinear algebraic equations, by considering *deflation matrices* $M(x; r)$ that transform the residual so that sequences that converge to a solution r of the original problem will not converge to that solution of the deflated problem. Let F be the residual of a system

*Mathematical Institute, University of Oxford, Oxford, UK. Center for Biomedical Computing, Simula Research Laboratory, Oslo, Norway (patrick.farrell@maths.ox.ac.uk).

†Mathematical Institute, University of Oxford, Oxford, UK (birkisson@maths.ox.ac.uk).

‡Center for Biomedical Computing, Simula Research Laboratory, Oslo, Norway and Department of Earth Science and Engineering, Imperial College London, London, UK. (s.funke09@imperial.ac.uk). This research is funded by EPSRC grant EP/K030930/1, an ARCHER RAP award, a Sloane Robinson Foundation Graduate Award associated with Lincoln College, by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant 291068, and a Center of Excellence grant from the Research Council of Norway to the Center for Biomedical Computing at Simula Research Laboratory. The authors would like to acknowledge useful discussions with E. Süli and L. N. Trefethen.

[1] In [42], Peters and Wilkinson draw a distinction between deflation (algebraically dividing the polynomial p by $(x - x^{[i]})$, and suppression (where the polynomial division is not performed explicitly, but the numerator and denominator are separately evaluated and floating point division is performed after the fact). In the subsequent literature, what these authors call suppression has come to be called deflation, and we follow that convention here.

of nonlinear algebraic equations, and let r be a computed solution of $F(x) = 0$. Of the deflation matrices considered by Brown and Gearhart, norm deflation extends most naturally, and is defined by choosing

$$M(x; r) \equiv \frac{I}{||x - r||}$$

as its deflation operator, where I is the appropriate identity matrix and $||\cdot||$ is some vector norm. This yields the modified residual function

$$G(x) = M(x; r)F(x) = \frac{F(x)}{||x - r||}.$$

Brown and Gearhart prove that the deflated Newton sequence will not converge to the previous solution, assuming r is simple. (More precisely, a rootfinding algorithm that employs the norm of the residual as its merit function will not converge to the previously-identified solution.)

This paper makes several novel contributions. First, we extend the theoretical framework of Brown and Gearhart to the case of infinite-dimensional Banach spaces, enabling the application of deflation techniques to systems of partial differential equations. Second, we introduce new classes of deflation operators to overcome some numerical difficulties of previous methods. Third, we discuss important details of applying these ideas in practice. Methods for solving PDEs typically exploit the sparsity of the residual Jacobian; the deflated Jacobian is dense, but we devise an efficient preconditioner for the dense deflated systems via the Sherman–Morrison formula. Finally, we demonstrate its widespread applicability on several problems of interest in the literature.

1.1. Other related techniques. There are two main alternative approaches to identifying distinct solutions of nonlinear systems: numerical continuation, and the approximate integration of the associated Davidenko differential equation.

The first approach, numerical continuation, is a well-established technique in the scientific computing literature [15, 16, 3, 4]. The essential idea of it is as follows: suppose a problem F with solution u is parameterised by a parameter λ:

$$F(u, \lambda) = 0. \tag{1.1}$$

Equation (1.1) could represent an algebraic problem, or an operator equation such as a PDE with boundary conditions. Respectively, u will either be a vector in $\mathbb{R}^n$ or a function in some function space. The parameter λ is usually a real scalar, but may be a vector of parameters. For a fixed value λ^*, there may exist zero, one, or many solutions u for which $F(u, \lambda^*) = 0$. For some problems, the parameter λ appears naturally, whereas for others it may be artificially introduced (such as in homotopy or incremental loading methods for solving difficult nonlinear equations). Studying how a solution u of (1.1) varies with λ is the subject of *bifurcation theory*.

Assume we have found one point u^* for which $F(u^*, \lambda^*) = 0$. Then, following the implicit function theorem in Banach spaces [35, theorem 13.22], under technical conditions there exist open neighbourhoods around λ^* and u^* and a unique function f for which u can be regarded as a function of λ, that is, $u = f(\lambda)$, and $F(f(\lambda), \lambda) = 0$. It is thus possible to define solution curves in the augmented solution space, which consist of points (u, λ) for which $F(u, \lambda) = 0$. Numerical continuation methods are

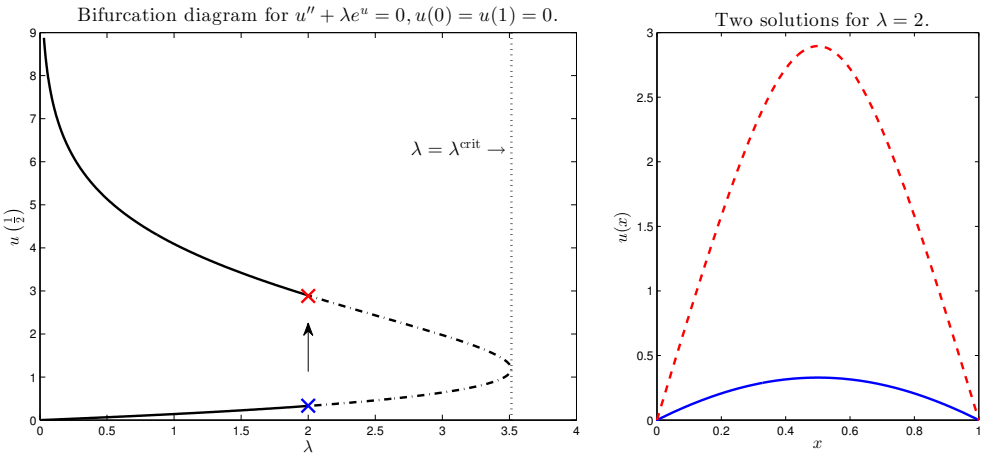

Fig. 1.1: A bifurcation diagram for the Bratu–Gelfand ODE (1.2). On the right, the two distinct solutions for $\lambda = 2$ are plotted. For a fixed value of λ, deflation enables the rootfinding algorithm to find one solution after the other without changing λ, whereas numerical continuation traces the curve around the turning point (dashed-dotted line).

concerned with tracing out these curves, which give rise to bifurcation diagrams, an example of which is shown in figure 1.1.

While the solution curves can often be parameterised locally by λ, it is well known that this parameterisation breaks down at points where the Fréchet derivative with respect to the solution becomes singular. Thus, techniques such as arclength or pseudo-arclength continuation are often applied, so that the continuation technique can extend beyond such points.

Consider the Bratu–Gelfand ordinary differential equation (ODE):

$$\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + \lambda e^u = 0, \quad u(0) = u(1) = 0. \tag{1.2}$$

It is well known [19] that for values of $\lambda < \lambda^{\text{crit}} \approx 3.51383$, (1.2) has two solutions; for $\lambda = \lambda^{\text{crit}}$, it has one solution, and no solutions exist for $\lambda > \lambda^{\text{crit}}$ (figure 1.1). Fix a value $\lambda^* < \lambda^{\text{crit}}$, and suppose one solution of the two is known. Numerical continuation traces the solution curve around the turning point, increasing and decreasing λ until the second solution for λ^* is identified. By contrast, deflation modifies the residual of the problem to eliminate the first solution from consideration, enabling the Newton–Kantorovitch iteration (henceforth referred to as Newton's method) to converge directly to the second solution without changing λ.

The second approach relies on the numerical integration of the Davidenko differential equation (DDE) associated with the original nonlinear problem $F(u) = 0$ [18, 11]. The DDE introduces a new arclength parameter s and considers the augmented system

$$\frac{\mathrm{d}F(u(s))}{\mathrm{d}s} + F(u(s)) = 0, \tag{1.3}$$

with initial condition $u(s = 0)$ given by the initial guess to the solution. This has

a strong connection to Newton's method: provided the Fréchet derivative of F with respect to u is nonsingular, the chain rule implies that

$$\frac{\mathrm{d}F(u(s))}{\mathrm{d}s} = F'(u)\frac{\mathrm{d}u}{\mathrm{d}s}.$$

Hence (1.3) can be rewritten as

$$\frac{\mathrm{d}u}{\mathrm{d}s} = -\left(F'(u)\right)^{-1} F(u), \tag{1.4}$$

and thus Newton's method is just the forward Euler discretisation of (1.4) with unit arclength step. Branin's method consists of considering the modified equation

$$\frac{\mathrm{d}F(u(s))}{\mathrm{d}s} \pm F(u(s)) = 0,$$

where the sign is changed whenever the functional determinant of the Fréchet derivative changes sign or whenever a solution is found. The major difficulty with implementing this method is that computing the determinant is impractical for large-scale problems, where matrix decompositions are not feasible. Limited computational experience indicates no performance benefit over deflation: applying the Newton-like forward Euler discretisation of Branin's method to the Allen–Cahn problem of section 4.2 finds no solutions (whereas deflation finds three), and applying it to the Yamable problem of section 4.3 finds two solutions (whereas deflation finds seven).

The deflation technique we present here is distinct from those algorithms (often also called deflation) which aim to improve the convergence of Newton's method towards *multiple solutions*, solutions at which the Fréchet derivative is singular [41, 30, 38]. The algorithm presented here is also distinct from deflation for eigenvalue problems.

2. Deflation for PDEs.

2.1. Deflation operators.

We now extend the results of [13] to the case of infinite-dimensional Banach spaces.

DEFINITION 2.1 (deflation operator on a Banach space [8]). *Let V, W and Z be Banach spaces, and U be an open subset of V. Let $\mathcal{F}{:}U \subset V \to W$ be a Fréchet differentiable operator with derivative $\mathcal{F}'$. For each $r \in U$, let $\mathcal{M}(u;r) : W \to Z$ be an invertible linear operator, defined for all $u \in U_r$, where U_r is an open subset of U such that r belongs to the closure of U_r. We say that $\mathcal{M}$ is a* deflation operator *if for any $\mathcal{F}$ such that $\mathcal{F}(r) = 0$ and $\mathcal{F}'(r)$ is nonsingular, we have*

$$\liminf_{i\to\infty} ||\mathcal{M}(u_i;r) \circ \mathcal{F}(u_i)||_Z > 0 \tag{2.1}$$

for any sequence $\{u_i\}$ converging to r, $u_i \in U_r$.

Informally, a deflation operator transforms the original equation $\mathcal{F}(u) = 0$ to ensure that r will not be found by any algorithm that uses the norm of the problem residual as a merit function. Once a solution $r^{[0]}$ has been found (by any means), we form the new nonlinear problem

$$\mathcal{F}^{[1]}(u) \equiv \mathcal{M}(u;r^{[0]}) \circ \mathcal{F}(u) = 0 \tag{2.2}$$

and apply the rootfinding technique to $\mathcal{F}^{[1]}$. Clearly, this procedure may be iterated until the rootfinding technique diverges for some $\mathcal{F}^{[i]}$.

Brown and Gearhart introduced a *deflation lemma* [13, lemma 2.1] for determining whether a matrix function can serve as a deflation matrix. We now extend this to deflation operators.

LEMMA 2.2 (sufficient condition for identifying deflation operators [8]). *Let $\mathcal{F}:U \subset V \to W$ be a Fréchet differentiable operator. Suppose that the linear operator $\mathcal{M}(u;r) : W \to Z$ has the property that for each $r \in U$, and any sequence $u_i \xrightarrow{U} r, u_i \in U_r$, if*

$$||u_i - r||\, \mathcal{M}(u_i;r)w_i \xrightarrow{Z} 0 \implies w_i \xrightarrow{W} 0 \tag{2.3}$$

for any sequence $\{w_i\}, w_i \in W$, then $\mathcal{M}$ is a deflation operator.

Proof. Assume (2.3) holds. If $\mathcal{M}$ is not a deflation operator, then there exists a Fréchet differentiable operator $\mathcal{F}:U \subset V \to W$ and an $r \in U$ such that $\mathcal{F}(r) = 0$, $\mathcal{F}'(r)$ nonsingular and

$$\liminf_{i\to\infty} ||\mathcal{M}(u_i;r) \circ \mathcal{F}(u_i)||_Z = 0$$

for some sequence $\{u_i\}$ converging to r, $u_i \in U_r$. Then there exists a subsequence $\{v_i\}$ such that $\mathcal{M}(v_i;r)\mathcal{F}(v_i) \xrightarrow{Z} 0$. Defining $\{w_i\} \in W$ such that

$$w_i = \frac{\mathcal{F}(v_i)}{||v_i - r||_U},$$

we have

$$||v_i - r||_U\, \mathcal{M}(v_i;r)w_i \xrightarrow{Z} 0.$$

By (2.3), $w_i \xrightarrow{W} 0$, i.e.

$$\frac{\mathcal{F}(v_i)}{||v_i - r||_U} \xrightarrow{W} 0. \tag{2.4}$$

Since $\mathcal{F}$ is Fréchet differentiable, we can expand it in a Taylor series around r to give

$$\begin{aligned}\mathcal{F}(v_i) &= \mathcal{F}(r) + \mathcal{F}'(r; v_i - r) + o(||v_i - r||_U^2) \\ &= \mathcal{F}'(r; v_i - r) + o(||v_i - r||_U^2),\end{aligned}$$

as $\mathcal{F}(r) = 0$. We then have that

$$\begin{aligned}\frac{\mathcal{F}(v_i)}{||v_i - r||_U} &= \frac{1}{||v_i - r||_U}\left[\mathcal{F}'(r; v_i - r) + o(||v_i - r||_U^2)\right] \\ &\approx \mathcal{F}'(r; \bar{v_i}),\end{aligned}$$

where

$$\bar{v_i} = \frac{(v_i - r)}{||v_i - r||_U} \in U_r$$

is a function with unit norm for all $v_i \in U_r$. But then (2.4) leads to a contradiction of the nonsingularity of $\mathcal{F}'$. ∎

The utility of this lemma is that it allows us to define candidate deflation operators $\mathcal{M}(u;r)$ and check whether property (2.3) holds; if it holds, then $\mathcal{M}(u;r)$ is indeed a deflation operator. In the next section, we introduce several classes of deflation operators to which lemma 2.2 applies.

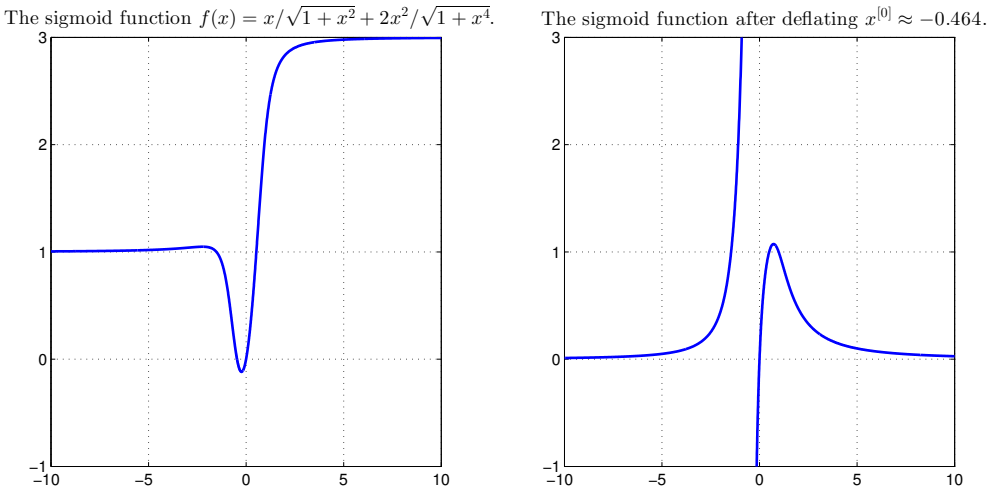

Fig. 2.1: A plot of the sigmoid function (2.7) used to motivate the development of shifted deflation. On the right, the function obtained after deflating the solution $x^{[0]} \approx -0.464$ with $p = 2$ is shown. The deflated function tends to zero away from the deflated solution.

2.2. Classes of deflation operators. The simplest kind of deflation, Brown and Gearhart's norm deflation, extends naturally to the infinite-dimensional case.

DEFINITION 2.3 (norm deflation). *Norm deflation specifies*

$$\mathcal{M}(u; r) = \frac{\mathcal{I}}{||u - r||_U}, \tag{2.5}$$

where $\mathcal{I}$ is the identity operator on W.

That (2.5) defines a deflation operator follows straightforwardly from lemma 2.2.

In computational practice it has occasionally proven useful to consider a generalisation of norm deflation, where the norm is taken to a power:

DEFINITION 2.4 (exponentiated-norm deflation). *Exponentiated-norm deflation specifies*

$$\mathcal{M}_p(u; r) = \frac{\mathcal{I}}{||u - r||_U^p}, \tag{2.6}$$

where $p \in \mathbb{R} \geq 1$.

For example, in the Painlevé example of section 4.1, the globalised Newton algorithm fails to find the second solution when norm deflation is used, but succeeds when squared-norm deflation ($p = 2$) is used.

While often successful, this class of deflation operator can sometimes induce numerical difficulties for a nonlinear rootfinding algorithm: the rootfinding algorithm can erroneously report convergence due to small residuals, even when u is far away from a solution. This can be clearly seen in the following example. Consider the problem of finding the solutions of the sigmoid function

$$f(x) = \frac{x}{\sqrt{1+x^2}} + \frac{2x^2}{\sqrt{1+x^4}}. \tag{2.7}$$

This function has two roots at $x = 0$ and $x = -\sqrt{\frac{1}{3}\left(\sqrt{7}-2\right)} \approx -0.464$, and is plotted in figure 2.1. Starting from the initial guess $x = -1$, undamped Newton iteration converges to the root $x \approx -0.464$ in 5 iterations. Suppose exponentiated norm deflation is applied to this problem with $p = 2$. When the Newton iteration is applied for a second time starting from $x = -1$, the algorithm finds that it can make the norm of the residual arbitrarily small by pushing x towards $-\infty$: at $x \approx -1.2 \times 10^8$, the deflated residual has norm on the order of 10^{-13}, and the algorithm erroneously reports successful convergence with a small residual. While this example is artificially constructed, similar erroneous behaviour has been observed in practice for more complicated problems.

This phenomenon of the deflation factor causing the residual to go to zero away from the solution motivates the development of shifted deflation:

DEFINITION 2.5 (shifted deflation). *Shifted deflation specifies*

$$\mathcal{M}_{p,\alpha}(u;r) = \frac{\mathcal{I}}{||u-r||_U^p} + \alpha\mathcal{I}, \tag{2.8}$$

where $\alpha \geq 0$ is the shift.

The extra term $\alpha\mathcal{I}, \alpha > 0$ ensures that the norm of the deflated residual does not artificially go to zero as $||u-r|| \longrightarrow \infty$. Computational experience has found that shifted deflation is often more robust when seeking distinct solutions.

3. Implementation: sparsity and preconditioning. Exploiting sparsity of the (discretised) Jacobian in the Newton step is critical for computational efficiency for many discretisation techniques for PDEs, such as finite differences and finite elements. Consider one application of deflation to yield

$$\mathcal{G}(u) = \frac{\mathcal{F}(u)}{\eta(u)},$$

where the exact form of $\eta : U \mapsto \mathbb{R}$ depends on the kind of deflation operator employed. The Fréchet derivative of $\mathcal{G}$ in a direction δu is given by

$$\mathcal{G}'(u;\delta u) = \frac{\mathcal{F}'(u;\delta u)}{\eta(u)} - \frac{\mathcal{F}(u)}{\eta(u)^2}\eta'(u;\delta u).$$

Suppose the problem is discretised so that the solution u is sought in $\mathbb{R}^N$, where N is the number of degrees of freedom. The discretisation $J_G \in \mathbb{R}^{N\times N}$ of $\mathcal{G}'(u)$ will be dense even if the discretisation $J_F \in \mathbb{R}^{N\times N}$ of $\mathcal{F}'(u)$ is sparse, as J_G is a rank-one perturbation of J_F:

$$J_G(u) = \frac{J_F(u)}{\eta(u)} - \frac{F(u)}{\eta(u)^2} \otimes d(u) \tag{3.1}$$

where $F(u) \in \mathbb{R}^N$ is the discretisation of $\mathcal{F}(u)$ and $d(u) \in \mathbb{R}^N$ is the discretisation of $\eta'(u;\cdot)$.

This has two practical implications for a software implementation. First, the discretised deflated Jacobian J_G should not be stored explicitly, but rather its action should be computed matrix-free via (3.1). Second, some care must be taken in preconditioning linear systems involving J_G (when Newton–Krylov methods are used). Suppose that a left preconditioner P_F is available for the matrix J_F, i.e. that the undeflated Newton–Krylov method approximately solves

$$P_F^{-1}J_F x = P_F^{-1}b \tag{3.2}$$

using some Krylov method. Neglecting dependence on u for notational brevity, this suggests using the preconditioner

$$P_G = P_F - \frac{F}{\eta^2} \otimes d, \tag{3.3}$$

for the deflated linear system:

$$P_G^{-1} J_G x = P_G^{-1} b. \tag{3.4}$$

The action of P_G^{-1} may be computed via the Sherman–Morrison formula [7, 31]:

$$(P_F + f d^T)^{-1} = P_F^{-1} - \frac{P_F^{-1} f d^T P_F^{-1}}{1 + d^T P_F^{-1} f},$$

where $f = -F/\eta^2$, provided $1 + d^T P_F^{-1} f \neq 0$. This allows for the matrix-free computation of actions of P_G^{-1} in terms of actions of P_F^{-1}.

We hope that if P_F is a useful preconditioner for J_F, then P_G will be a useful preconditioner for J_G. Expanding, we find

$$\begin{aligned}
P_G^{-1} J_G &= (P_F + f d^T)^{-1} (J_F + f d^T) \\
&= (P_F^{-1} - \frac{P_F^{-1} f d^T P_F^{-1}}{1 + d^T P_F^{-1} f})(J_F + f d^T) \\
&= P_F^{-1} J_F + P_F^{-1} f d^T - \frac{P_F^{-1} f d^T P_F^{-1} J_F + P_F^{-1} f d^T P_F^{-1} f d^T}{1 + d^T P_F^{-1} f} \\
&= P_F^{-1} J_F + P_F^{-1} f d^T - \frac{P_F^{-1} f d^T + P_F^{-1} f d^T P_F^{-1} f d^T}{1 + d^T P_F^{-1} f} - \frac{P_F^{-1} f d^T \left(P_F^{-1} J_F - I\right)}{1 + d^T P_F^{-1} f} \\
&= P_F^{-1} J_F + P_F^{-1} f d^T - \frac{P_F^{-1} f \left(1 + d^T P_F^{-1} f\right) d^T}{1 + d^T P_F^{-1} f} - \frac{P_F^{-1} f d^T \left(P_F^{-1} J_F - I\right)}{1 + d^T P_F^{-1} f} \\
&= P_F^{-1} J_F - \frac{P_F^{-1} f d^T \left(P_F^{-1} J_F - I\right)}{1 + d^T P_F^{-1} f}.
\end{aligned}$$

This expression has two consequences. First, if $P_F^{-1} = J_F^{-1}$ (computed with an LU or Cholesky decomposition) so that (3.2) is solved in one iteration, then (3.4) will also be solved in one iteration, as the error term is multiplied by $P_F^{-1} J_F - I$. Second, on taking matrix norms, we find

$$\begin{aligned}
\left|\left|P_G^{-1} J_G - P_F^{-1} J_F\right|\right| &= \left|\frac{1}{1 + d^T P_F^{-1} f}\right| \left|\left|P_F^{-1} f d^T \left(P_F^{-1} J_F - I\right)\right|\right| \\
&\leq \left|\frac{1}{1 + d^T P_F^{-1} f}\right| \left|\left|P_F^{-1} f d^T\right|\right| \left|\left|P_F^{-1} J_F - I\right|\right|.
\end{aligned}$$

Far from the previously found solution, $P_F^{-1} f$ will be small (as it is divided by the square of the deflation factor η). Hence, heuristically, the difference between the deflated and undeflated preconditioned operators will be small if P_F is a good preconditioner for J_F in the sense that $\left|\left|P_F^{-1} J_F - I\right|\right|$ is small.

This approach to preconditioning the deflated systems may be extended to the case of multiple deflation with the Woodbury formula [47]. Computational experience with this preconditioned Newton–Krylov method is reported in the examples of sections 4.2 and 4.3.

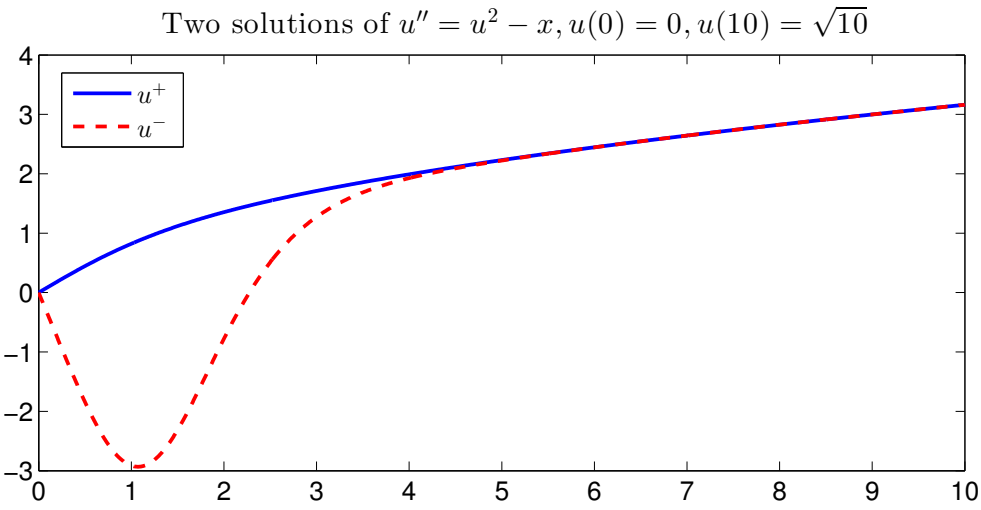

Fig. 4.1: The positive- and negative-slope solutions u^+ and u^- of the Painlevé boundary value problem (4.2).

4. Examples.

4.1. Special functions: Painlevé. A well-studied example of an ordinary differential equation boundary-value problem that permits distinct solutions is based on the first Painlevé transcendent [34, 40, 28], and is given by

$$\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = u^2 - x, \quad u(0) = 0, \quad u(x) \sim \sqrt{x} \text{ as } x \to \infty, \tag{4.1}$$

where the latter condition means that the solution should asymptote to $\sqrt{x}$. In [32], it was shown that exactly two solutions exist that satisfy these conditions, one with a positive slope at $x = 0$, and the other with a negative slope at $x = 0$. We refer to these solutions as u^+ and u^-. The first solution is easy to compute, while computing the second is far more computationally challenging: almost all convergent initial guesses attempted converge to u^+. Here, we truncate (4.1) to the interval $[0, 10]$, and consider the task of finding the two solutions to the boundary-value problem:

$$\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = u^2 - x, \quad u(0) = 0, \quad u(10) = \sqrt{10}. \tag{4.2}$$

Using the `chebop` class of Chebfun [9, 22], the u^+ solution can be obtained via spectral methods [45], using the affine-covariant globalised `NLEQ-ERR` Newton algorithm of [20, pg. 148] in function space. All linear systems arising were solved with the `lu` function of MATLAB. The initial guess for the Newton iteration was set to the linear function that satisfies the boundary conditions, and solved with the following lines of MATLAB code:

```
% Define the interval to solve the problem on:
domain = [0, 10];
% Construct a chebop object, representing the differential operator:
N = chebop(@(x,u) diff(u, 2) − u.^2 + x, domain);
```

```
% Impose Dirichlet boundary conditions:
N.lbc = 0; N.rbc = sqrt(10);
% Solve using overloaded \ method:
uplus = N\0;
```

No damping was required for the Newton iteration to converge from the initial guess used. Deflation was then applied with $p = 2$ and $\alpha = 0$, and the u^- solution was computed starting from the same initial guess. Globalised damped Newton iteration was applied to achieve convergence to the u^- solution, requiring 12 steps to converge. (Deflation with $p = 1$ failed to converge, even with globalisation.) In this case, globalisation was necessary: applying the undamped exact Newton iteration to the deflated problem diverged for every value of p and α attempted. The two solutions obtained are plotted in figure 4.1.

4.2. Phase separation: Allen–Cahn. The Allen–Cahn equation [2] was proposed to model the motion of boundaries between phases in alloys, and is well-known to permit several steady solutions.

The equation considered is the steady Allen–Cahn equation:

$$-\delta\nabla^2 u + \delta^{-1}(u^3 - u) = 0, \tag{4.3}$$

where $u = +1$ corresponds to one material, and $u = -1$ the other, and $\delta = 0.04$ is a parameter relating the strength of the free surface tension to the potential term in the free energy. The equation is solved on the unit square $\Omega = (0, 1)^2$ with boundary conditions:

$$\begin{aligned} u &= +1 \quad \text{on } x = 0,\ x = 1,\ 0 < y < 1, \\ u &= -1 \quad \text{on } y = 0,\ y = 1. \end{aligned} \tag{4.4}$$

The equation was discretised with piecewise linear finite elements on a 100×100 mesh via FEniCS [39]. Undamped Newton iteration was applied to solve the nonlinear problem.

This example was studied by [24] in the context of identifying minimal action pathways and transition times between stable states. In the absence of noise, a system evolves to one of its stable states and stays there indefinitely; by contrast, when stochastic noise is present, the system may switch between metastable states when the noise perturbs the system out of the basin of attraction for its steady solution. The string method of E, Ren and Vanden–Eijnden [23, 24, 25] attempts to identify the most probable transition path between two metastable states, and hence to characterise the transition timescales. However, this requires some foreknowledge of the steady solutions, so that the endpoints of the string may be initialised appropriately. Deflation may be used to identify the different solutions of the associated steady problem.

Starting each time from an initial guess of zero, Newton's method with deflation ($p = 1$, $\alpha = 0$) finds three solutions (figure 4.2): one unstable solution (figure 4.2a), and two stable solutions (figures 4.2b, 4.2c). This information about the stable solutions would be sufficient to initialise the string method to find the minimum energy pathway (which, in this case, happens to pass through the unstable solution).

The linear systems arising in the Newton algorithm were solved using GMRES [43] and the GAMG classical algebraic multigrid algorithm [1] via PETSc [6], with two smoother iterations of Chebyshev and SOR. The relative and absolute Krylov solver tolerances were both set to 10^{-12}. The average number of Krylov iterations required

(a) Unstable solution (b) Negative stable solution (c) Positive stable solution

Fig. 4.2: Solutions of the Allen–Cahn equation (4.3)–(4.4); c.f. figure 4.1 of [24], left column.

# of deflations	average Krylov iterations per solve
0	10.84
1	10.58
2	10.53

Table 4.1: The performance of the preconditioning strategy of section 3 on the Allen–Cahn problem, section 4.2. As more solutions are deflated, the number of Krylov iterations required does not increase, indicating that the preconditioning strategy suggested is effective.

in the Newton iterations for each solution are listed in table 4.1. The number of Krylov iterations required for the deflated solves stays approximately constant. This suggests that the preconditioning strategy proposed in section 3 is effective, even as several solutions are deflated.

4.3. Differential geometry: Yamabe. In 1960, Hidehiko Yamabe [48] posed the following problem: given a compact manifold M of dimension $n \geq 3$ with Riemannian metric g, is it possible to find a metric $\tilde{g}$ conformal to g (a multiplication of g by a positive function) that has constant scalar curvature? Yamabe showed that solving this problem is equivalent to finding a u such that

$$-a\nabla^2 u - Su + \lambda u^{p-1} = 0, \tag{4.5}$$

where $a = 4(n-1)/(n-2)$, $p = 2n/(n-2)$, S is the scalar curvature of g, and λ is the constant scalar curvature of $\tilde{g}$. This problem has a solution; for a full account, see [37]. This is an instance of a more general class of critical exponent problems [12, 26], which often permit distinct solutions.

Following [26], we consider a Yamabe-like problem arising from $n = 3$, but posed on a two-dimensional domain Ω:

$$-8\nabla^2 u - \frac{1}{10}u + \frac{1}{r^3}u^5 = 0, \tag{4.6}$$

where Ω is an annulus centred on the origin of inner radius 1 and outer radius 100,

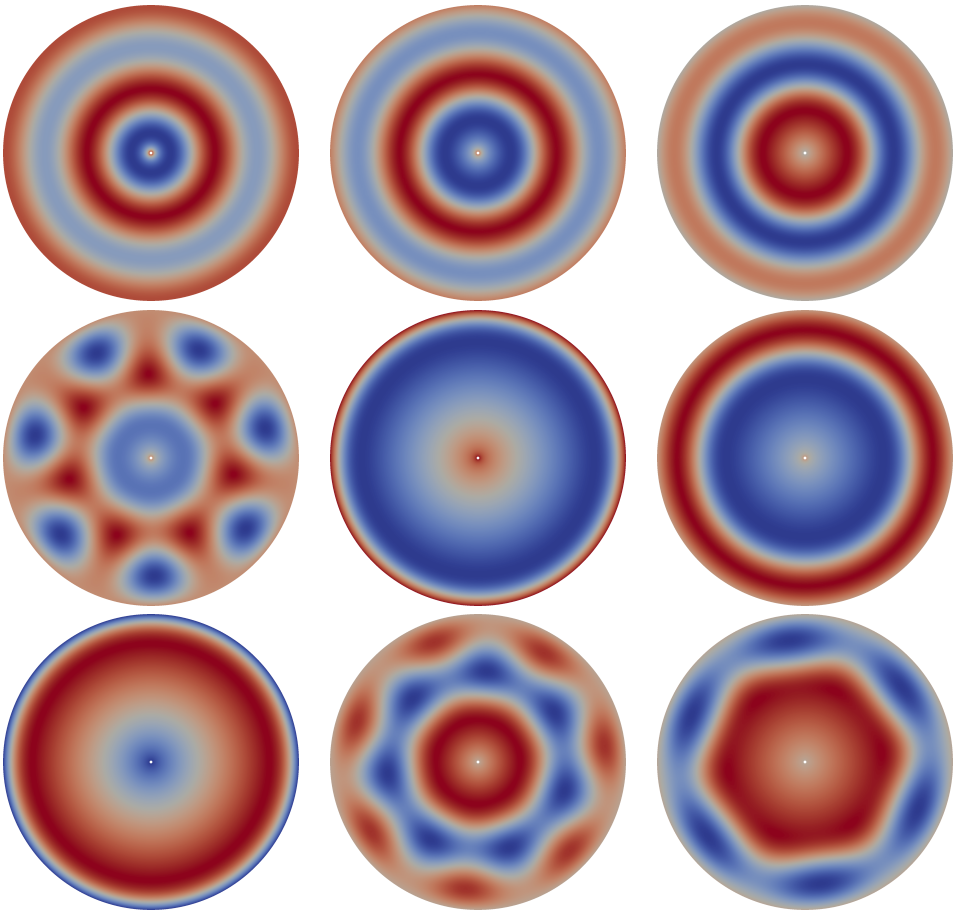

Fig. 4.3: Solutions of the Yamabe equation, found using deflation and negation. (All subfigures are plotted with different colour bars.)

and r is the distance to the origin. The system is closed by imposing that $u = 1$ on the boundaries. In [26], the authors are concerned with devising schemes to solve the nonlinear problem (4.6) subject to the constraint that $u \geq 0$; here, the nonnegativity constraint is relaxed, and solutions of any sign are sought using deflation.

The equation (4.6) was discretised using standard linear finite elements via FEniCS. An undamped Newton iteration was used to solve the nonlinear problem, with at most 100 iterations and an absolute residual termination tolerance of 10^{-10}. The mesh of 15968 vertices was generated with gmsh [29]. Starting from the constant initial guess of 1, deflation was successively applied with shift parameter $\alpha = 10^{-2}$. After each deflation, the nonlinear solver tolerance was reduced by a factor of 10^{-2}. With this configuration, deflation found 7 solutions, all starting from the same initial guess. Finally, further solutions were identified by using the negative of each solution in turn as the initial guess for the (deflated) solver; the negatives of solutions 4 and 6 yielded two additional distinct solutions. All of the 9 solutions identified with this procedure are plotted in figure 4.3. The nonnegative solution was the 7^{th} to be

# of deflations	average Krylov iterations per solve
0	15.2
1	17.1
2	15.1
3	16.9
4	11.2
5	12.4
6	10.9
7	15.5
8	13.9

Table 4.2: The performance of the preconditioning strategy of section 3 on the Yamabe problem, section 4.3. As more solutions are deflated, the number of Krylov iterations required does not significantly increase, indicating that the preconditioning strategy suggested is effective.

identified. It is known that other solutions exist that this procedure has not found.

The linear systems arising in the Newton algorithm were solved using GMRES and the GAMG classical algebraic multigrid algorithm via PETSc, with two smoother iterations of Chebyshev and SOR. The relative and absolute Krylov solver tolerances were both set to 10^{-12}. The average number of Krylov iterations required in the Newton iterations for each solution are listed in table 4.2. The number of Krylov iterations required for the deflated solves does not change significantly. This suggests that the preconditioning strategy proposed in section 3 is effective, even as several solutions are deflated.

While this example demonstrates the power of the deflation technique presented here, it also emphasises its drawbacks: the algorithm is highly sensitive to the choice of the shift parameter. Varying the shift parameter from $10^0, 10^{-1}, \ldots, 10^{-7}$, the deflation procedure identifies between 1 and 7 solutions. At present we are unable to give *a priori* guidance on the choice of the shift parameter, and reluctantly resort to numerical experimentation.

4.4. Flow bifurcation: Navier–Stokes. The previous examples have demonstrated the utility of deflation for finding distinct solutions from the same initial guess. In this example, we combine continuation (in Reynolds number) and deflation to trace out the solution branches of the flow in a pipe that undergoes a sudden expansion [27, 21, 17]. Continuation and deflation are natural complements: continuation uses the solutions identified for previous parameter values to trace out known solution branches, whereas deflation attempts to find unknown solutions which may or may not lie on unknown solution branches.

The system considered is the nondimensionalised steady incompressible Newtonian Navier–Stokes equations:

$$-\frac{1}{\mathrm{Re}}\nabla^2 u + u \cdot \nabla u + \nabla p = 0, \tag{4.7}$$

$$\nabla \cdot u = 0, \tag{4.8}$$

where u is the vector-valued velocity, p is the scalar-valued pressure, and Re is the Reynolds number. The geometry is the union of two rectangles, $\Omega = (0, 2.5) \times (-1, 1) \cup$

$(2.5, 150) \times (-6, 6)$. Poiseuille flow is imposed at the inflow boundary on the left; an outflow boundary condition ($\nabla u \cdot n = pn$, where n is the unit outward normal) is imposed on the right; and no-slip ($u = 0$) imposed on the remaining boundaries. This configuration is symmetric around $y = 0$, and permits a symmetric solution. It is well known (see e.g. [17]) that for low Reynolds numbers, this symmetric solution is stable. At a critical Reynolds number, the system undergoes a pitchfork bifurcation, and the system permits an unstable symmetric solution and (possibly several) pairs of asymmetric solutions. The system (4.7) is discretised using the Taylor–Hood finite element pair [44] via FEniCS; undamped Newton iteration is used for the resulting nonlinear problems, with the undeflated linear systems solved on 12 cores with MUMPS [5] via PETSc.

The approach to trace out the solution branches proceeds as follows. The continuation path starts at $\mathrm{Re} = 10$ and uses a zero initial guess to find the first (symmetric) solution branch. Then continuation steps of $\Delta\mathrm{Re} = 0.5$ are made. The initial guess for the Newton iteration for (4.7) is set to each solution identified for the previous value of Re in turn; once each solution for the new parameter value is found, the residual is deflated with $p = 1$, $\alpha = 1$. Once all known solution branches have been continued, the Newton iteration is initialised with the average of the previous solutions, and an attempt is made to locate any nearby unknown solutions via deflation. We emphasise that more sophisticated continuation algorithms such as pseudo-arclength continuation [36, 3] could naturally be combined with deflation.

The algorithm was terminated at $\mathrm{Re} = 100$, having found 6 solutions (figure 4.4). As we expect one symmetric solution and pairs of asymmetric solutions, clearly the procedure has not found all possible solutions. This is confirmed by initialising a Newton iteration with the reflection in the line $y = 0$ of each asymmetric solution found; this identifies one extra solution (the second-last of figure 4.4). All identified solutions persist under grid refinement.

5. Conclusion. We have presented a deflation algorithm for computing distinct solutions of systems of partial differential equations. By systematically modifying the residual of the problem, known solutions may be eliminated from consideration and new solutions sought. While deflation does not guarantee that all solutions of the problem will be found, it has proven itself useful on several nonlinear PDEs drawn from special functions, phase separation, differential geometry and fluid mechanics. It provides a powerful complement to continuation methods, and motivates further developments of sophisticated nonlinear solvers [14].

A possible future application of deflation is to the computation of multiple local optima of PDE-constrained optimisation problems, possibly with additional equality constraints [33, 10], by means of computing distinct solutions of the associated Karush–Kuhn–Tucker equations.

References.

[1] M. F. Adams, H. H. Bayraktar, T. M. Keaveny, and P. Papadopoulos, *Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom*, in ACM/IEEE Proceedings of SC2004: High Performance Networking and Computing, Pittsburgh, Pennsylvania, 2004.

[2] S. M. Allen and J. W. Cahn, *A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening*, Acta Metallurgica, 27 (1979), pp. 1085–1095.

[3] E. L. Allgower and K. Georg, *Continuation and path following*, Acta Numerica, 2 (1993), pp. 1–64.

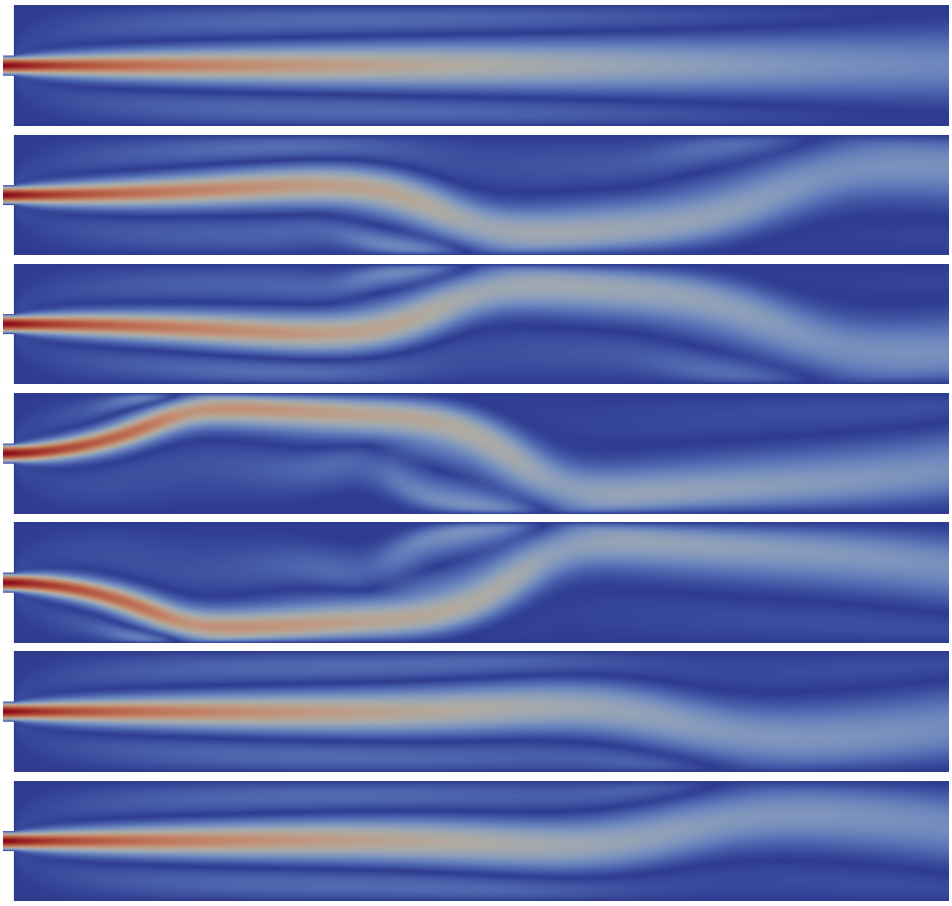

Fig. 4.4: The velocity magnitude of the solutions of the Navier–Stokes sudden expansion problem, at $\mathrm{Re} = 100$, found using deflation and reflection.

[4] ——, *Introduction to Numerical Continuation Methods*, SIAM, Philadelphia, 2003.

[5] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 15–41.

[6] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhäuser Press, 1997, pp. 163–202.

[7] M. S. Bartlett, *An inverse matrix adjustment arising in discriminant analysis*, The Annals of Mathematical Statistics, 22 (1951), pp. 107–111.

[8] Á. Birkisson, *Numerical Solution of Nonlinear Boundary Value Problems for Ordinary Differential Equations in the Continuous Framework*, PhD thesis, University of Oxford, 2014.

[9] Á. Birkisson and T. A. Driscoll, *Automatic Fréchet differentiation for the numerical solution of boundary-value problems*, ACM Transactions on Mathe-

matical Software, 38 (2012), pp. 26:1–26:29.
[10] A. Borzì and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, 2011.
[11] F. H. Branin, *Widely convergent method for finding multiple solutions of simultaneous nonlinear equations*, IBM Journal of Research and Development, 16 (1972), pp. 504–522.
[12] H. Brezis and L. Nirenberg, *Positive solutions of nonlinear elliptic equations involving critical Sobolev exponents*, Communications on Pure and Applied Mathematics, 36 (1983), pp. 437–477.
[13] K. M. Brown and W. B. Gearhart, *Deflation techniques for the calculation of further solutions of a nonlinear system*, Numerische Mathematik, 16 (1971), pp. 334–342.
[14] P. Brune, M. G. Knepley, B. Smith, and X. Tu, *Composing scalable nonlinear algebraic solvers*, preprint ANL/MCS-P2010-0112, Argonne National Laboratory, 2013.
[15] K.-S. Chao, D.-K. Liu, and C.-T. Pan, *A systematic search method for obtaining multiple solutions of simultaneous nonlinear equations*, IEEE Transactions on Circuits and Systems, 22 (1975), pp. 748–753.
[16] M.-J. Chien, *Searching for multiple solutions of nonlinear systems*, IEEE Transactions on Circuits and Systems, 26 (1979), pp. 817–827.
[17] K. A. Cliffe, E. J. C. Hall, and P. Houston, *hp-adaptive discontinuous Galerkin methods for bifurcation phenomena in open flows*, Computers & Mathematics with Applications, 67 (2014), pp. 796–806.
[18] D. F. Davidenko, *On a new method of numerical solution of systems of nonlinear equations*, Doklady Akademii Nauk SSSR, 88 (1953), pp. 601–602. (In Russian).
[19] H. T. Davis, *Introduction to Nonlinear Differential and Integral Equations*, Dover, 1962.
[20] P. Deuflhard, *Newton Methods for Nonlinear Problems*, vol. 35, Springer-Verlag, Berlin, Heidelberg, 2011.
[21] D. Drikakis, *Bifurcation phenomena in incompressible sudden expansion flows*, Physics of Fluids, 9 (1997), pp. 76–87.
[22] T. A. Driscoll, N. Hale, and L. N. Trefethen, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.
[23] W. E, W. Ren, and E. Vanden-Eijnden, *String method for the study of rare events*, Physical Review B, 66 (2002), p. 052301.
[24] ——, *Minimum action method for the study of rare events*, Communications on Pure and Applied Mathematics, 57 (2004), pp. 637–656.
[25] ——, *Simplified and improved string method for computing the minimum energy paths in barrier-crossing events*, Journal of Chemical Physics, 126 (2007).
[26] J. Erway and M. Holst, *Barrier methods for critical exponent problems in geometric analysis and mathematical physics*, 2011. arXiv:1107.0360 [math.PH].
[27] R. M. Fearn, T. Mullin, and K. A. Cliffe, *Nonlinear flow phenomena in a symmetric sudden expansion*, Journal of Fluid Mechanics, 211 (1990), pp. 595–608.
[28] B. Fornberg and J. A. C. Weideman, *A numerical methodology for the Painlevé equations*, Journal of Computational Physics, 230 (2011), pp. 5957–5973.
[29] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-D finite element mesh gen-*

erator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering, 79 (2009), pp. 1309–1331.

[30] A. Griewank, *On solving nonlinear equations with simple singularities or nearly singular solutions*, SIAM Review, 27 (1985), pp. 537–563.

[31] W. Hager, *Updating the inverse of a matrix*, SIAM Review, 31 (1989), pp. 221–239.

[32] S. Hastings and W. Troy, *On some conjectures of Turcotte, Spence, Bau, and Holmes*, SIAM Journal on Mathematical Analysis, 20 (1989), pp. 634–642.

[33] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE constraints*, vol. 23 of Mathematical Modelling: Theory and Applications, Springer, 2009.

[34] P. Holmes and D. Spence, *On a Painlevé-type boundary-value problem*, The Quarterly Journal of Mechanics and Applied Mathematics, 37 (1984), pp. 525–538.

[35] J. K. Hunter and B. Nachtergaele, *Applied Analysis*, World Scientific, Singapore, 2001.

[36] H. B. Keller, *Numerical solution of bifurcation and nonlinear eigenvalue problems*, in Applications of Bifurcation Theory, P. H. Rabinowitz, ed., New York, 1977, Academic Press, pp. 359–384.

[37] J. M Lee and T. H Parker, *The Yamabe problem*, Bulletin of the American Mathematical Society, 17 (1987), pp. 37–91.

[38] A. Leykin, J. Verschelde, and A. Zhao, *Newton's method with deflation for isolated singularities of polynomial systems*, Theoretical Computer Science, 359 (2006), pp. 111–122.

[39] A. Logg, K. A. Mardal, G. N. Wells, et al., *Automated solution of differential equations by the finite element method*, Springer, 2011.

[40] V. A. Noonburg, *A separating surface for the Painlevé differential equation* $x'' = x^2 - t$, Journal of Mathematical Analysis and Applications, 193 (1995), pp. 817–831.

[41] T. Ojika, S. Watanabe, and T. Mitsui, *Deflation algorithm for the multiple roots of a system of nonlinear equations*, Journal of Mathematical Analysis and Applications, 96 (1983), pp. 463–479.

[42] G. Peters and J. H. Wilkinson, *Practical problems arising in the solution of polynomial equations*, IMA Journal of Applied Mathematics, 8 (1971), pp. 16–35.

[43] Y. Saad and M. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[44] C. Taylor and P. Hood, *A numerical solution of the Navier-Stokes equations using the finite element technique*, Computers & Fluids, 1 (1973), pp. 73–100.

[45] L. N. Trefethen, *Spectral Methods in* Matlab, vol. 14 of Software, Environments, and Tools, SIAM, Philadelphia, 2000.

[46] J. H. Wilkinson, *Rounding errors in algebraic processes*, vol. 32 of Notes on Applied Science, H.M.S.O., 1963.

[47] M. A. Woodbury, *Inverting modified matrices*, Tech. Report Memorandum 42, Statistical Research Group, Princeton University, Princeton, New Jersey, 1950.

[48] H. Yamabe, *On a deformation of Riemannian structures on compact manifolds*, Osaka Mathematical Journal, 12 (1960), pp. 21–37.