A FINITE DIFFERENCE METHOD FOR FREE BOUNDARY PROBLEMS

BENGT FORNBERG*

Abstract. Fornberg and Meyer-Spasche proposed some time ago a simple strategy to correct finite difference schemes in the presence of a free boundary that cuts across a Cartesian grid. We show here how this procedure can be combined with a minimax-based optimization procedure to rapidly solve a wide range of elliptic-type free boundary value problems.

Key words. Finite differences, free boundary, minimax optimization.

1. Introduction. Both free and moving boundary problems arise in a vast number of physical settings, as surveyed for example in [3] and [12]. We focus in this study on free boundary problems of the type

$$Lu + f^{+}(u) = 0 \tag{1.1}$$

where L is a linear (or locally linearizable) second order elliptic differential operator, and

$$f^{+}(u) = \begin{cases} \lambda u + O(u^2) \text{ or } \mu + O(u) & \text{if } u > 0 \\ 0 & \text{if } u \le 0 \end{cases} . \tag{1.2}$$

Here u is a function of any number of space variables, and λ and μ are either constants or smooth functions of the space variables. The many important free boundary problems that can be cast in these forms include the Euler equations for steady flows [11], [4], [5] and the Grad-Shafranov equation describing magnetohydrodynamic (MHD) equilibria in plasma physics [1], [8], [10]. A common feature between free boundary problems in these as well as in others categories (such as flows in porous media, etc.) is that interfaces virtually always become very smooth, thereby offering great opportunities for highly effective numerical representations. This is in striking contrast to moving boundary problems, for which extreme interface complexities are common.

The simplest possible numerical approach for the present class of problems would be to discretize L with standard second order finite difference (FD2) approximations (a 5-point stencil in 2-D if L contains no mixed derivatives), and use for $f^{+}(u)$ its value at the FD2-stencil's center point. The two main problems with this approach are how to:

- Avoid the loss of accuracy that will arise because of inaccurate treatment of the interface on the fixed grid. Movements of the free boundary with less than the grid spacing may not be 'felt' at all by this numerical scheme.

  An extremely simple correction scheme (involving no u-values beyond those that are already part of the basic FD2 stencil) was initially proposed in [10], and then discussed further in [9]. Like for the immersed interface method [7], second order accurate results are obtained starting with standard FD2 approximations of L. We will find here that the free boundary influence in some cases will be sufficiently reduced that Richardson extrapolation can be used to improve the accuracy still further.

*University of Colorado, Department of Applied Mathematics, 526 UCB, Boulder, CO 80309, USA (fornberg@colorado.edu).

- Achieve a high and reliable rate of convergence in the numerical iterations.

  Since free boundary problems are nonlinear, some type of iterative scheme will be needed for their solution. Successful options include a regula falsi approach [6], Newton-type iterations [2], and fixed-area (or fixed-circulation) functional iterations [4]. We propose here the use of a numerical 'minimax' approach, implemented for example with the *fminimax* function in Matlab's optimization toolbox.

We introduce in Section 2 two test problems that were earlier considered in [9], and then proceed by

- Describing the free boundary correction procedure,
- Showing how this procedure reduces the numerical residual in case both the free boundary and the analytic solution u are known,
- Showing how we can obtain an accurate solution to u in case only the free boundary is given, and how this provides an easy-to-measure residual if the given boundary approximation is inaccurate,
- Showing how the free boundary approximation can be adjusted in order to minimize this residual.

The last of these steps completes the description of the present short and easy-to-implement Cartesian grid-based FD2 algorithm in which the presence of a free boundary only leads to a very small degradation of accuracy.

2. Two primary test problems. After having arrived at the numerical procedure in Section 6, we will use Problems 1 and 2 described below for the numerical tests given in Section 7. Euler flows give rise to free boundary equations of a similar type as Problem 1 while both types frequently arise in MHD contexts (tokamaks, solar physics, etc.). The task will be to numerically determine both the PDE solution $u(x, y)$ and the location of the free boundary. The solution $u(x, t)$ to Problem 1 will feature a discontinuous second derivative at the free boundary. The solution to Problem 2 is one order smoother - irregular first in its third derivative. Although this is not a limitation of the numerical approach, we restrict for simplicity our attention to problems in which the free boundary does not intersect any fixed boundary or any edge of the computational domain. As is explained in Section 8, only a very minor modification of the algorithm is needed if these two problems are replaced by the much more general ones outlined in (1.1) and (1.2). This is also illustrated in Figures 7-9 in [9]. Of the two present test problems, the first one has by far the most severe free boundary irregularity, and is therefore the most important case to achieve a free boundary correction for. We are not considering cases here when the solution is irregular first in the fourth derivative (or higher still) since the errors due to the interface then become negligible compared to standard second order discretization errors, and no interface correction procedure is called for.

2.1. Problem 1. Consider the PDE

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + H(u) = 0 \tag{2.1}$$

where

$$H(u) = \begin{cases} 1 & \text{if } u > 0 \\ 0 & \text{if } u \le 0 \end{cases} .$$

We will solve this elliptic free boundary problem over $-4 \le x \le 4$, $-4 \le y \le 4$ with boundary conditions (BC) along the four sides that match the analytic solution

$$u(x,y) = \begin{cases} 1 - r^2/4 & \text{if } r < 2 \\ 2\ln(2/r) & \text{if } r \ge 2 \end{cases}, \tag{2.2}$$

where $r = \sqrt{x^2 + y^2}$.

2.2. Problem 2. Consider the PDE

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + u^+ = 0 \tag{2.3}$$

with

$$u^+ = \begin{cases} u & \text{if } u > 0 \\ 0 & \text{if } u \le 0 \end{cases}.$$

In this case, the analytical solution is

$$u(x,y) = \begin{cases} J_0(r) & \text{if } r < r_c \\ A\ \ln(r_c/r) & \text{if } r \ge r_c \end{cases}. \tag{2.4}$$

The radius r is defined as before, $r_c \approx 2.404826$ is the first zero of $J_0(r)$, and $A = -[r\ \frac{d}{dr} J_0(r)]_{r=r_c} = r_c J_1(r_c) \approx 1.248459$.

3. Concept behind the free boundary correction procedure. We start by considering Problem 1, and assume at first that we know both $u(x,y)$ and the location of the free boundary (along which $u = 0$). Standard FD2 approximations will be locally second order accurate (in the grid spacing h) whenever the free boundary does not cut across any of the four legs of the 5-point FD2 stencil. Consider next the situation in Figure 3.1 and let u_k denote the u-value at node k, $k = 0, 1, \ldots, 4$. We have in this case $u_0 < 0$, and the standard FD approximation becomes

$$(u_1 + u_2 + u_3 + u_4 - 4u_0)\frac{1}{h^2} = 0. \tag{3.1}$$

Locally approximating the free boundary by a straight line, we can introduce the (ξ, η)-system that is also shown in the figure. In this system, the PDE simplifies to

$$u_{\xi\xi} = \begin{cases} 0 & \text{where } \xi < 0 \\ -1 & \text{where } \xi > 0 \end{cases}. \tag{3.2}$$

The values of u at the nodes 1 and 2 are therefore smaller than what they otherwise would have been by the amounts $\frac{1}{2}d_1^2$ and $\frac{1}{2}d_2^2$, respectively, where d_1 and d_2 are the distances by which these nodes fall outside the free boundary. If we subtract $\frac{1}{2h^2}d_1^2 + \frac{1}{2h^2}d_2^2$ from the right hand side (RHS) of (3.1), local second order accuracy becomes restored. If only one leg of a stencil extends across the free boundary, we similarly need only one correction term of this kind. If the stencil center falls in the region where $u > 0$, any stencil leg that extends across the free boundary can likewise be corrected by just reversing the sign of the correction term. All that needs to be kept track of is therefore the distances between the free boundary and any nodes that fall on the other side of it relative to the stencil's center point (or some approximations for these distances).

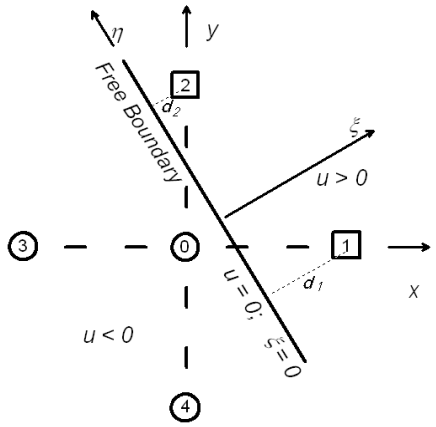

FIG. 3.1. *Schematic illustration of a FB cutting across two legs of a 2-D 5-point stencil. The nodes marked 1 and 2 fall here on the opposite side of the FB than the stencil center point (node 0).*

4. Calculation of the finite difference (FD) residual if the correct PDE solution and the free boundary are given. The top row of subplots in Figure 4.1 displays, from left to right, (a) the analytic solution (2.2) (with the free boundary marked), (b) the analytic Laplacian at all node points (taking the value -1 when $r < 2$, zero otherwise), and (c) the numerical Laplacian, as obtained by applying the 5-point stencil to the analytic solution at all node points. Superficially, the two last plots look quite similar, but the first subplot in the bottom row (d) reveals that they in fact differ quite dramatically and in a highly irregular manner along the free boundary. Subplot (e) displays what the correction procedure above tells that the RHS ought to have been adjusted with in order to account for the free boundary. The excellent agreement between the latter two results is confirmed in subplot (f), which shows their difference. As noted in the figure caption, the interface errors have in this case been reduced by a factor of around 20.

5. Calculation of the PDE solution if the free boundary is given. We next assume that only the free boundary is given, but not the solution $u(x,t)$. Based of knowing the free boundary, we can trivially detect when one or two legs of a stencil extends across it and adjust the PDE's right hand side accordingly. Figure 5.1 shows in subplot (a) the FD2 solution to the PDE if the corrections are not made, and in subplot (b) its difference to the analytical solution (2.2). With the correction included, subplot (c) shows that the errors due to the free boundary have been virtually eliminated. The discrepancies that remain come from standard second order truncation errors in the smooth regions. In Figure 5.2, we display the solution errors, in max norm over all the node points when we, instead of a 32×32 grid, use $n \times n$ grids, where n ranges from 5 to 401. Without correction, the error is seen to be highly erratic, roughly following an $O(h^{1.5})$ trend. It would have been $O(h^1)$ had it not been for the cancellations due to the randomly varying signs of the errors, as seen in Figure 4.1 (d). This is because, without cancellations, errors of size $O(1)$ at $O(1/h)$ locations (where the free boundary crosses legs of the FD2 stencil) would cause comparable inaccuracies as $O(h)$-sized errors at $O(1/h^2)$ (i.e. at all) node locations. If a Poisson equation right hand side is increased (or decreased) uniformly by $O(h)$,

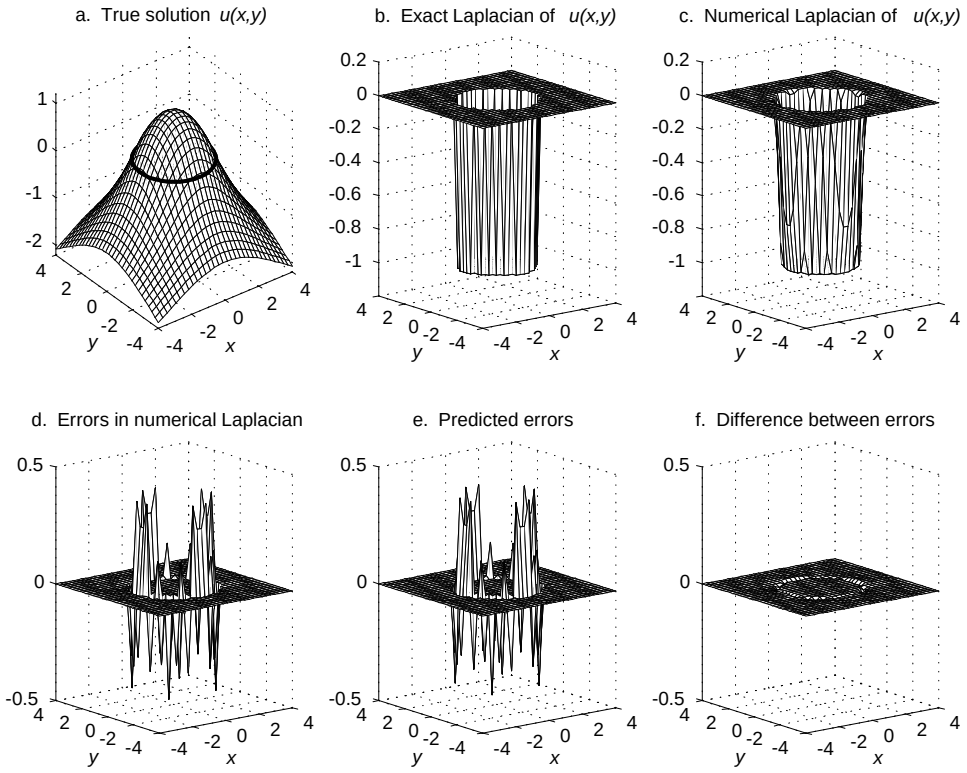

FIG. 4.1. *The true solution to Problem 1 on a* 32×32 *grid; Laplacians and residuals without and with the correction procedure. The maximum magnitudes of the entries in subplots d and f are 0.434 and 0.020, respectively, corresponding an error reduction by a factor of about 20.*

the solution will also be perturbed to the same order.

With the correction approach in place, the solid curve in Figure 5.2 very clearly shows the error trend to instead be $O(h^2)$, just as would have been the case if no free boundary had been present. To reach an error level of 10^{-3}, $n \approx 100$ suffices when the correction method is used vs. $n \approx 1000$ otherwise - a saving with a factor of 100 in total number of nodes.

6. Adjusting the free boundary to minimize the residual. The numerical algorithm. When the free boundary is given, we saw in the last section how we then readily can calculate the full solution $u(x, y)$. In particular, it is then easy (for example with Matlab's *interp2* routine) to inspect u along the free boundary. Ideally, in the absence of errors in both the free boundary location and in the numerical FD2 approximation, this ought to produce a perfectly zero result. When using the analytically correct free boundary, the subplots of Figure 6.1 show the numerically obtained values along it (displayed at equispaced angles, as seen from the origin) when $n = 21, 81, 321$. In each case, the solid curve shows the result with the free boundary correction in use, and the dashed curve without it. A thin dotted line marks the desired zero error level. The errors without the correction procedure are far greater than with it.

We are now ready to describe our proposed strategy for when neither the solution

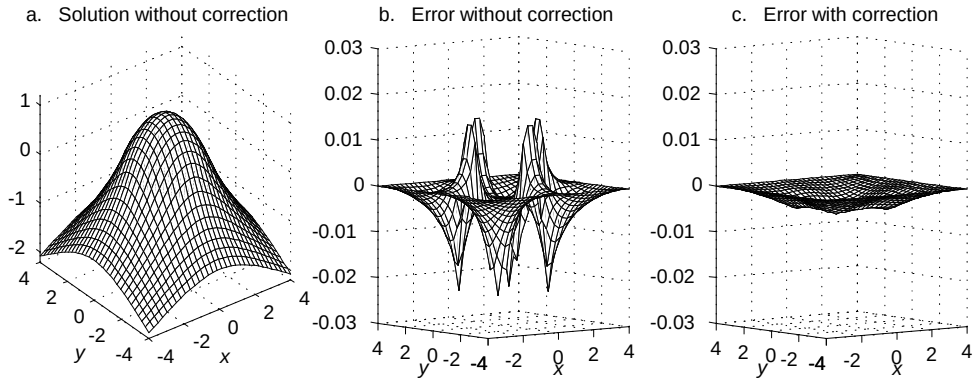

FIG. 5.1. *Numerical solution to Problem 1 when the FB is given; errors without and with the correction procedure.*

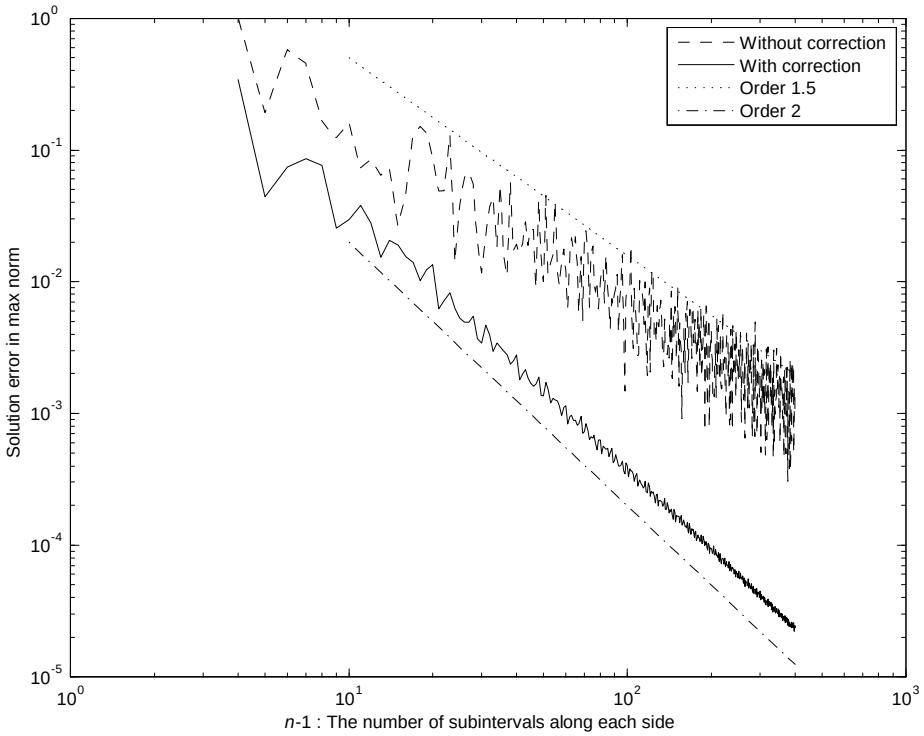

FIG. 5.2. *Max norm errors when solving Problem 1 with the FB location given. The step size h satisfies $h = 8/(n-1)$.*

$u(x, y)$, nor the free boundary are provided. The only two components that are needed are:

- A routine which, given a guess for the free boundary, returns the residual (accurate computed values of $u(x, y)$) along it.
- An optimization routine, such as Matlab's *fminimax*, that can be used to automatically vary the parameters describing the free boundary location in

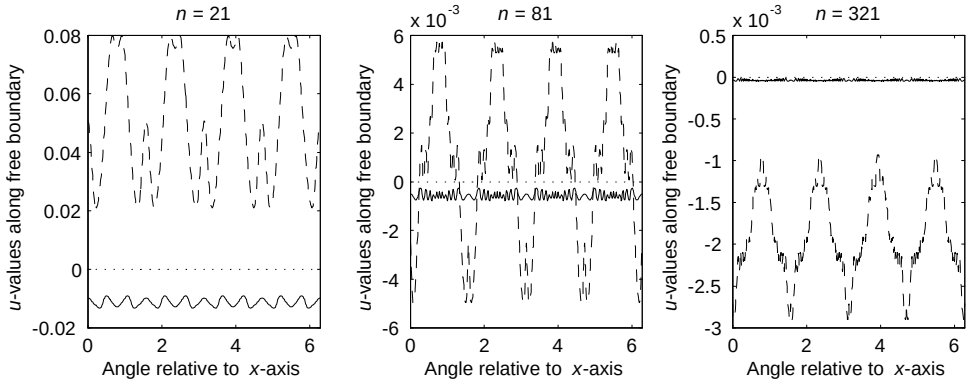

FIG. 6.1. *Values of $u(x, y)$ along the free boundary (here parametrized by the angle in polar coordinates). Dashed line: without the correction, solid line: with correction. The ideal curves should be identically zero - marked by a dotted line. As before, n denotes the number of nodes along each side.*

such a way that this residual gets minimized.

Since a free boundary tend to be very smooth in the present problem class, some 8-12 free parameters usually suffices to describe it well (e.g. Fourier coefficients if the free boundary is represented in polar form, or sample points along the free boundary in case we use periodic splines). We smooth out the high frequency noise in the residuals (the 'jitter' in the solid curves in Figure 6.1) and then sample the smoothed residual at a somewhat larger set of points than the number of free parameters used to represent the free boundary. This setup is perfectly suited for the *fminimax* routine, with the option set to minimize in max norm.

Effective multivariate optimization is a very non-trivial task for which very sophisticated library routines are readily available, with many choices provided in Matlab's optimization and genetic algorithm tool boxes. The main alternative to using *fminimax* would seem to be to create a scalar-valued residual at each iteration (rather than the vector-valued residual we are using and which returns of the values of the computed $u(x, y)$ at a relatively large number of sample points along the free boundary). A natural choice for such a scalar residual would be to form the 2-norm (or the infinity norm) of our vector valued residual. Although we have not exhaustively explored the effectiveness such scalar residual-based optimizers in the present context (especially since they typically require 'fine tuning' of a large number of parameters in order to be most effective), the preliminary tests we have carried out do not seem encouraging. The full vector residual we have available at each iteration contains a large amount of information which *fminimax* puts to very effective use, but which is totally lost whenever the residual vector is condensed into a single scalar quantity before being presented to the numerical optimizer.

7. Numerical tests.

7.1. Results for Problem 1. Figure 7.1 shows how the resulting errors both in free boundary position and for $u(x, y)$ (both in max norm, compared to the analytical results) vary with n. The present procedure clearly solves the free boundary problem to full second order accuracy. We do not include any results here for the uncorrected FD2 scheme, since the residual then varies so erratically with small changes in free

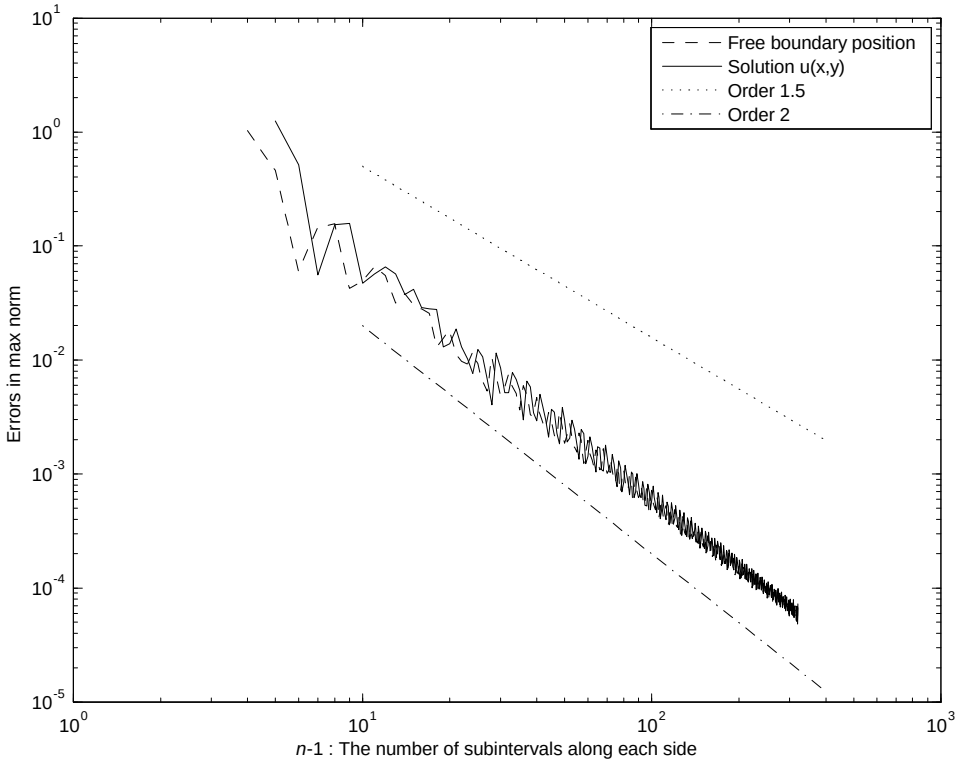

FIG. 7.1. *Max norm errors in FB position and in solution u(x,t) for the corrected FD2 scheme. The step size is* $h = 8/(n-1)$. *The lines marking the slopes* $O(h^{1.5})$ *and* $O(h^2)$ *are the same as in Figure 5.2.*

boundary position that *fminimax* either fails to converge, or becomes quite ineffective. With the correction procedure included, the *fminimax* procedure proved to be highly effective in terms of the number of function evaluations it requires. For its default error tolerance of 10^{-6}, and a reasonably good initial guess, it typically converges in 40-100 iterations (almost independently of n). In most applications, one is not interested in solving a single free boundary problem in complete isolation, but rather a sequence of such problems for which some physical parameters vary in a gradual manner. In such cases, good initial guesses are readily available, and 20-60 iterations may be more typical. Figure 7.2 illustrates that there is some convergence penalty, but not a particularly serious one, if the initial guess deviates significantly from the true solution.

Each iteration (residual calculation) requires only the solution of a single FD2-discretized Poisson equation. A large number of effective Poisson solvers are readily available for this task. In the present work, we used sparse Gaussian elimination (as implemented through Matlab's "\" operator), but any other standard approach, such as GMRES, multigrid solvers, etc., would also have worked perfectly well. When using sparse Gaussian elimination, the system can be LU-factorized once and for all, making the repeated solutions extremely fast. Iterative methods can utilize the fact that the solutions will differ very little between successive calls from the *fminimax* routine.

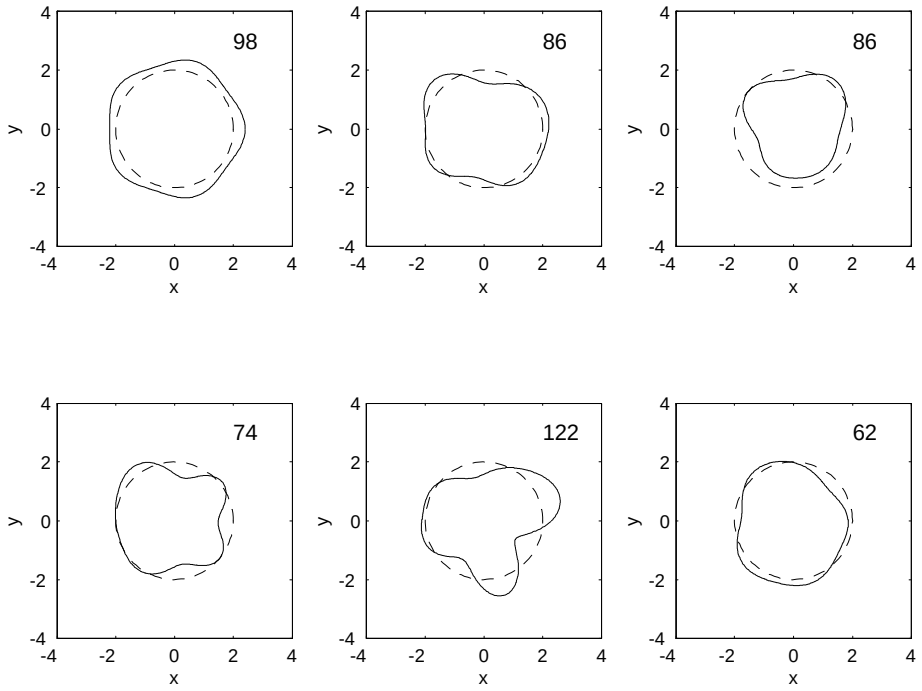

FIG. 7.2. *The numer if iterations (residual evaluations) required for Test Problem 1 when starting with various initial free boundary approximations (solid curves). This test was carried out with an $n = 81$ grid, and fminimax converged in all cases to its default precision of 10^{-6}.*

7.2. Results for Problem 2. There are three main differences compared to Problem 1:

- Equation (3.2) becomes in this case replaced by

$$u_{\xi\xi} = \begin{cases} 0 & \text{where } \xi < 0 \\ -u & \text{where } \xi > 0 \end{cases} . \tag{7.1}$$

  Further noting that u here can be closely approximated by $|\nabla u|\xi$ (with ∇u denoting the local gradient of u), the correction for a node k on the opposite side to the free boundary than the stencil center point becomes now $\frac{|\nabla u|}{6h^2} d_k^3$ instead of previously $\frac{1}{2h^2} d_k^2$.

- Some rough approximation of $|\nabla u|$ is needed. The whole numerical procedure is iterative, and the result from a previous iteration is easily of sufficient accuracy (doing no boundary correction at all produces second order of accuracy in this case, so already a first order accurate approximation to the correction is enough to bring the boundary-caused errors well below the regular FD2 error level).

- Another consequence of the $-u$ term in (7.1) is that the weight at the center point of the 5-point stencil becomes slightly modified for nodes inside the free boundary. Again, this is easily accommodated for by standard numerical

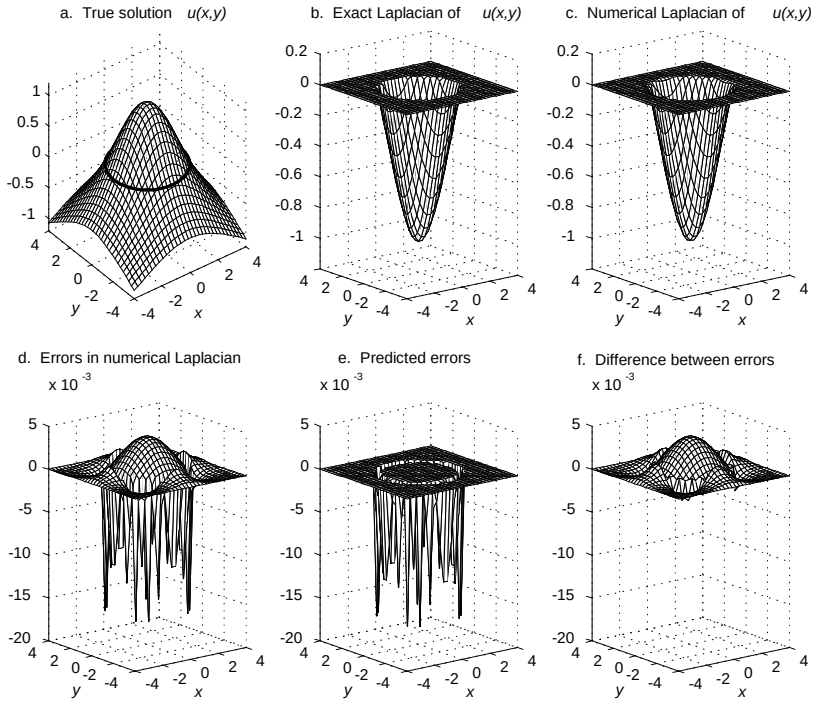

FIG. 7.3. *The counterpart to Figure 4.1, but for Problem 2.*

Poisson solvers.

Figure 7.3 shows the same information for Problem 2 as what Figure 4.1 displayed for Problem 1 The errors in the (uncorrected) numerical Laplacian along the free boundary are now much smaller in size (the solution $u(x,t)$ is discontinuous first in the third derivative, rather than in the second derivative for Problem 1) and they do not oscillate in sign. The correction scheme is again seen to provide excellent predictions for the FD2 scheme's errors along the free boundary and, with it eliminated, subplot (f) shows that regular truncation errors in the smooth regions then totally dominate any free boundary induced errors.

Figure 7.4 corresponds to Figure 5.2 for Problem 1. Even without the free boundary correction, the errors are this time on the $O(h^2)$ level. Because they are of one sign only (cf. Figure 7.3 d), they happen to partly balance out the errors seen in the center region. Hence we actually see in Figure 7.4 slightly smaller errors without the correction than with it (top two curves). However, there is a lot more fine-scale jitter in the uncorrected case, making Richardson extrapolation much less reliable. When we look at the extrapolated curves in Figure 7.4 (based on meshes with grid sizes h and $2h$), it becomes clear that the present correction procedure combined with extrapolation is by far the most effective option.

The *fminimax* optimization applies again without any difficulties, producing accuracy results completely analogous to the residual results just shown in Figure 7.4.

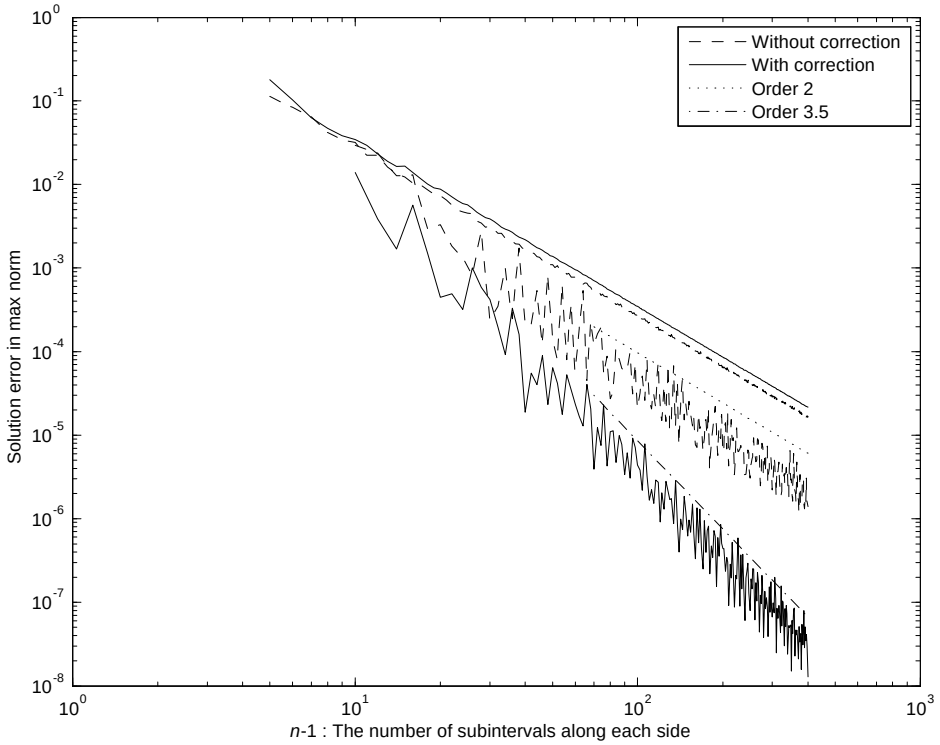

FIG. 7.4. *Max norm errors when solving Problem 2 with the FB location given. The step size* h *satisfies* $h = 8/(n-1)$*. The top two curves show corrected and the uncorrected FD2 results. The two jagged curves show the previous results following a standard Richardson extrapolation. The trends for increasing* n *(decreasing* h*) seem now to agree well with* $O(h^2)$ *and* $O(h^{3.5})$*, respectively.*

8. Generalization to other PDEs in 2-D and in 3-D. The proposed correction procedure, together with the use of *fminimax*, generalizes immediately to the wider class of problems that was referred to in (1.1), (1.2), as well as to PDEs in more than two space variables. The cases of 2-D and 3-D are discussed separately below.

8.1. 2-D cases. In the general 2-D case, the differential operator L takes the form

$$L = a(x,y)\frac{\partial^2 u}{\partial x^2} + b(x,y)\frac{\partial^2 u}{\partial x \partial y} + c(x,y)\frac{\partial^2 u}{\partial y^2} + d(x,y)\frac{\partial u}{\partial x} + e(x,y)\frac{\partial u}{\partial y}. \tag{8.1}$$

Locally within each stencil, we can (to the required leading order of accuracy) replace all the variable coefficients with constant coefficients based on the values at the stencil's center point. Referring to Figure 3.1, the (ξ,η)-system is translated a small amount (which is of no consequence) and then rotated an angle θ relative to the (x,y)-system. The rotation transforms the operator to

$$(a\cos^2\theta + b\cos\theta\sin\theta + c\sin^2\theta)\frac{\partial^2 u}{\partial \xi^2} + (\cdots)\frac{\partial^2 u}{\partial \xi \partial \eta} + (\cdots)\frac{\partial^2 u}{\partial \eta^2} + (\cdots)\frac{\partial u}{\partial \xi} + (\cdots)\frac{\partial u}{\partial \eta}. \tag{8.2}$$

It suffices here to consider the case of Test Problem 1, in which a constant of one enters the PDE the moment the free boundary is crossed (the analysis for second case

is entirely equivalent). The change we encounter in u will then, to leading order, be equivalent to the deviation from identical zero encountered for the ODE

$$(a\cos^2\theta + b\cos\theta\sin\theta + c\sin^2\theta)\, u_{\xi\xi} + 1 = 0 \tag{8.3}$$

with the initial conditions $u(0) = u'(0) = 0$. Compared to (3.2), the only difference is the presence of a constant scalar multiplier

$$M = (a\cos^2\theta + b\cos\theta\sin\theta + c\sin^2\theta)\ . \tag{8.4}$$

In the Laplace case, $a = c = 1$ and $b = 0$. M evaluates then to one for all values of θ, and this multiplier could consequently be ignored. In other cases (independently of the character of the nonlinear term), the corrections we use should just be divided by this factor M relative to their values in the Laplace operator case. Since we, at each stage of our iterations, have an approximation available for the location of the free boundary, the values for θ are immediately available.

While iterating, we also have approximations for $u(x, y)$ available. Even right across an interface, the first derivatives of $u(x, y)$ are continuous, and $u(x, y)$ can therefore locally be well approximated by a plane. This offers us another convenient opportunity for approximating θ and thereby for computing the quantity M. Using local centered finite differences within the stencil, we obtain readily approximations δ_x for $\partial u/\partial x$ and δ_y for $\partial u/\partial y$. It then holds that $\cos\theta = \delta_x/\sqrt{\delta_x^2 + \delta_y^2}$ and $\sin\theta = \delta_y/\sqrt{\delta_x^2 + \delta_y^2}$, providing us with the alternative approximation

$$M = (a\ \delta_x^2 + b\ \delta_x\delta_y + c\ \delta_y^2)/(\delta_x^2 + \delta_y^2). \tag{8.5}$$

This formulation is equivalent to equation (15) in [9] (where it was stated without any derivation; Figures 7-9 in that reference illustrated its accuracy).

8.2. 3-D cases. From the 2-D discussion above, it is clear that we only need to consider the second order derivatives in L. Rather than using spherical coordinates to describe how the interface surface is oriented in the (x, y, z)-coordinate system, it becomes easier in this case to focus directly on obtaining a generalization of (8.5). With L locally taking the form

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2} + c\frac{\partial^2 u}{\partial z^2} + d\frac{\partial^2 u}{\partial x\partial y} + e\frac{\partial^2 u}{\partial x\partial z} + f\frac{\partial^2 u}{\partial y\partial z} + \{\text{lower order terms}\},$$

one obtains similarly in place of (8.5)

$$M = (a\ \delta_x^2 + b\ \delta_y^2 + c\ \delta_z^2 + d\ \delta_x\delta_y + e\ \delta_x\delta_z + f\ \delta_y\delta_z)/(\delta_x^2 + \delta_y^2 + \delta_z^2).$$

The only remaining 3-D issue is the technicality of finding a suitable numerical representation of the free surface in terms of a relatively low number of parameters. In case this surface is approximately spherical in nature, an expansion in spherical harmonics will provide uniform resolution in the same way as a Fourier expansion does in 2-D polar coordinates.

9. Conclusions. We have in this study revisited a procedure that was first proposed in 1991 and which already then was shown to strongly reduce FD2 errors in the vicinity of a free boundary. It has now been combined with minimax optimization, producing an effective and easy-to-use approach for solving a wide range of free

boundary problems. If the solution features a jump in the second derivative at the free boundary, full second order accuracy is obtained (in both solution and in the free boundary location). If the irregularity occurs first in the third derivative, Richardson extrapolation becomes available, and gives better than third order accuracy. In either case, the correction procedure greatly improves the accuracy of the original FD2 scheme or, if a certain accuracy level is required, this level may be reached with a fraction of the number of node points that otherwise would have been needed.

10. Acknowledgements. The present work was supported by the NSF Grants DMS-0611681, DMS-0914647 and ATM-0620068. A significant part of the present work was carried out while the author was a Vising Fellow at OCCAM (Oxford Centre for Collaborative Applied Mathematics) under support provided by Award No. KUK-C1-013-04 to the University of Oxford, UK, by King Abdullah University of Science and Technology (KAUST).

REFERENCES

[1] Bateman, G., MHD instabilities, MIT Press, Cambridge, MA / London (1980).
[2] Borja, R.I. and Kishnani, S.S., On the solution of elliptic free-boundary problems via Newton's method, Computer Math. in Appl. Mech. Eng. 88 (1991), 341-361.
[3] Crank, J., Free and moving boundary problems, Clarendon Press, Oxford (1984).
[4] Elcrat, A., Fornberg, B., Horn, M. and Miller, K., Some steady vortex flows past a circular cylinder, J. Fluid Mech. 409 (2000), 13-27.
[5] Elcrat, A., Fornberg, B. and Miller, K., Some steady axisymmetric vortex flows past a sphere, J. Fluid Mech. 433 (2001), 315-328.
[6] Fox, L. and Sankar, R., The regula-falsi method for free-boundary problems, IMA J. Appl. Math. 12 (1973), 49-54.
[7] LeVeque, R.J. and Li, Z., The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994), 1019-1044.
[8] Miller, K., Fornberg, B., Flyer, N. and Low, B.C., Magnetic relaxation in the solar corona, The Astrophysical Journal, 609 (2009), 720-733.
[9] Fornberg, B. and Meyer-Spasche, R., A finite difference procedure for a class of free boundary problems, J. Comput. Phys., 102 (1992), 72-77.
[10] Meyer-Spasche, R. and Fornberg, B., Discretization errors at free boundaries of the Grad-Schlüter-Shafranov equation, Numer. Math. 59 (1991), 683-710.
[11] Saffman, P.G., Vortex Dynamics, Cambridge University Press (1992).
[12] Scardovelli, R. and Zaleski, S., Direct numerical simulation of free-surface and interfacial flow, Annual Rev. Fluid Mech. 31 (1999), 567-603.