This is a postprint version of the following published document:

Bernal, F., Kindelan, M. (2006). Meshless Simulation of Hele-Shaw Flow. In: Bonilla L.L., Moscoso M., Platero G., Vega J.M. (eds), *Progress in Industrial Mathematics at ECMI 2006*. Mathematics in Industry. (The European Consortium for Mathematics in Industry), vol 12. Springer, Berlin, Heidelberg. Pp. 677-682.

DOI: https://doi.org/10.1007/978-3-540-71992-2_113

© Springer-Verlag Berlin Heidelberg 2008.

A Meshless Solution to the p-Laplace Equation

F. Bernal¹, and M. Kindelan²

¹ fcoberna@math.uc3m.es, ² kinde@ing.uc3m.es

G. Millán Institute for Modeling, Simulation and Industrial Mathematics,
Universidad Carlos III de Madrid, 28911 Leganés.

November 30, 2007

Abstract

The p-Laplace equation is a non-linear elliptic PDE which plays an important role in the modeling of many phenomena in areas such as glaciology, non-Newtonian rheology or edge-preserving image deblurring. We have linearized it and applied a scheme introduced by G. Fasshauer which allows to solve it in the framework of Kansa's method. In order to confirm the validity of the approach, a 2D example (the pressure distribution in Hele-Shaw flow) has been numerically solved. The convergence and accuracy of the method are discussed, and an improvement based on smoothing up the linearized PDE is suggested.

1 Introduction

The motivation for this work is the simulation of injection molding, a process of industrial relevance whereby molten polymer is driven into a cavity (the mold) in order to manufacture small plastic parts. If the polymer viscosity obeys a power law and the mold is thin compared to its planar dimensions, the classical mathematical model of injection molding is the Hele-Shaw approximation ([13]). In the remainder of this paper, we will restrict ourselves to isothermal Hele-Shaw flows, which physically arise whenever the fluid viscosity does not depend on temperature. In this case it suffices to solve the following 2D, non-linear, elliptic equation

$$\operatorname{div}(|\nabla u|^\gamma \nabla u) = 0 \quad (1)$$

whose solution yields the pressure distribution $u(x, y)$ in the filled region of the mold. Exponent γ completely characterizes the polymer rheology, and is typically $\gamma \approx 1/2$. We will assume dimensionless units: the independent

variables in this problem have been scaled in such a way that the maximal fluid velocity is one (the reasons for this criterion will be evident in Section 3). If the pressure profile (p_{IN}) is set along the injection gates by the injection machine, the boundary conditions are

$$u = p_{IN} \text{ (injection)} \quad \partial u / \partial n = 0 \text{ (walls)} \quad u = 0 \text{ (front)} \quad (2)$$

From this pressure field, the average planar velocity $\langle \vec{v} \rangle$ can be computed and the location of the advancing front can be updated:

$$\langle \vec{v} \rangle = - |\nabla u|^\gamma \nabla u \quad (3)$$

Although the boundary conditions (BCs) (2) are the most usual in commercial software, the rate Q at which the polymer is fed can be alternatively used. In this case, assuming that there is a single injection segment Γ , and always in dimensionless units, $\oint_\Gamma |\nabla u|^\gamma (\partial u / \partial n) dl = Q$. Instead, we will make the further assumption that the profile of $\langle \vec{v} \rangle$ along the injection segment, q_{IN} , is known. The proper BCs for this situation are therefore non-linear,

$$- |\nabla u|^\gamma \partial u / \partial n = q_{IN} \text{ (injection)} \quad \partial u / \partial n = 0 \text{ (walls)} \quad u = 0 \text{ (front)} \quad (4)$$

We will refer to (2) and (4) as *Dirichlet-injection* and *Neumann-injection* BCs, respectively.

The numerical simulation of the Hele-Shaw flow requires coupling a) some method for solving equation (1) at every time step with b) some technique to advance the front to its new position, until the mold domain has been completely filled. In the state-of-the-art approach a) is accomplished through finite elements (FEM), whereas for b), either the volume-of-flow (VoF) method is used, or the nodes along the front are tracked to their new positions. The latter option entails remeshing around the front at every time step, while the former avoids it at the price of forgoing a sharp frontline. Another disadvantage of FEM is the fact that the numerical interpolant is not differentiable along element borders, so that an averaging, upwind process is required in order to compute gradients on element nodes. In [4], an alternative, meshless framework was proposed for solving this problem combining the method of asymmetric Radial Basis Function (RBF) collocation for pressure with Level Sets for capturing the front motion. We believe that this approach has the potential to overcome some difficulties inherent to the FEM formulation.

Eq.(1) is a *p-Laplace* (also called *p-harmonic*) equation of index $p = \gamma + 2$. The p-Laplace operator

$$-\Delta_p u := \text{div}(|\nabla u|^{p-2} \nabla u) \quad (5)$$

has been regarded as a counterpart to the Laplace operator for non-linear phenomena. The equation $\Delta_p u = 0$ has been the subject of extensive mathematical research for its own sake (see for instance [1]). In the remainder of

this paper, we will consider the 2D case only (*i.e.* $u \in \mathbb{R}^d$ with $d = 2$), and will report on our recent progress in solving the non-linear elliptic PDE (1) in a meshless numerical environment. We believe that this approach could also be applied to other topical two-dimensional PDEs involving the p-Laplace operator as the core nonlinearity, such as the Perona-Malik equation for non-linear (edge-preserving) image denoising or the problem of finding the minimal surface resting on a given boundary.

2 Asymmetric RBF Collocation

2.1 Kansa's method

The idea of using RBFs to solve PDEs was first introduced by Kansa [14, 15]. Consider the boundary-value problem (BVP) $L(u) = f(\vec{x})$ in domain Ω with boundary conditions along $\partial\Omega$ given by $G(u) = g(\vec{x})$, where L and G are linear operators. Ω is discretized into a set of $N = N_I + N_B$ scattered nodes (called *centers*) $\chi = \{\vec{x}_i \in \Omega, i = 1 \dots N_I\} \cup \{\vec{x}_j \in \partial\Omega, j = N_I + 1 \dots N_I + N_B\}$ and an approximate solution to the PDE is sought in the form of a linear combination of RBFs $\{\phi_k(\vec{x}), k = 1 \dots N\}$ centered at each of them,

$$u(\vec{x}) = \sum_{k=1}^N \alpha_k \phi_k(\vec{x}), \quad \phi_k(\vec{x}) \equiv \phi(\|\vec{x} - \vec{x}_k\|) \quad (6)$$

Having L and G operate on the RBF, the unknown coefficients α_k are determined by appropriate collocation of either the PDE or the BC on N points, which usually -but not necessarily- are the same set of centers:

$$\sum_{k=1}^N \alpha_k L\phi_k(\vec{x}_i) = f(\vec{x}_i), \quad i = 1, \dots, N_I \quad (7)$$

$$\sum_{k=1}^N \alpha_k G\phi_k(\vec{x}_j) = g(\vec{x}_j), \quad j = N_I + 1, \dots, N_I + N_B \quad (8)$$

Inversion of the linear system (7)-(8) is guaranteed for positive-definite RBFs. In the event of a non-positive-definite RBF, positive-definiteness can be restored by adding a low order polynomial to (7)-(8) plus suitable constraints for the additional coefficients. However, even if the system (7)-(8) is formally solvable, it may be extremely ill-conditioned in practice. This drawback is compounded by 'Schaback's uncertainty principle', which establishes a trade-off between accuracy and ill-condition in RBF collocation.

PDE collocation on boundary (PDEBC). Accuracy can be greatly improved (especially in presence of Neumann BCs) by enforcing the PDE on the boundary nodes also [12]. In this case expansion (6) must be supplemented with N_B extra RBF centers $\{\vec{x}_m, m = N + 1, \dots, N + N_B\}$ in order to match the N_B new collocation

equations. Since these extra centers are not to be collocated on, they may lie outside the PDE domain. With them, the RBF interpolant takes on the form

$$u(\vec{x}) = \sum_{k=1}^N \alpha_k \phi_k(\vec{x}) + \sum_{m=N+1}^{N+N_B} \alpha_m \phi_m(\vec{x}) \quad (9)$$

2.2 Nonlinear Equations

In the event of a non-linear PDE the collocation equations (7)-(8) give rise to a non-linear system of algebraic equations. An entirely different approach is the *operator-Newton method*, which was first introduced by Fasshauer in the context of meshless methods [9] [8], and which is sketched below:

Algorithm 1

- Let $Hu = 0$ be an elliptic non-linear PDE in Ω and L a linearization of it
- Pick an initial guess $u_{k=0}$ of solution. We seek v such that $H(u + v) = 0$
- For $k = 1, 2, \dots$ until convergence
 - Compute residual $R_k = -Hu_{k-1}$
 - Solve $L_k v_k = R_k$ by Kansa's method, where $L_k = L(u_{k-1})$
 - Perform the smoothing of the Newton update, $\tilde{v}_k = S_k v_k$
 - Update the previous iterate, $u_k = u_{k-1} + \tilde{v}_k$

The idea behind the operator-Newton method is therefore to recast the nonlinear elliptic PDE into a succession of linear elliptic problems, whose coefficients are determined by the previous iterate. The smoothing step is optional (if $S = I$, smoothing is skipped) and designed to counter the phenomenon of *lack of derivatives*, which takes place whenever the numerical solution of the linearized PDE prevents the Newton iterations from achieving full quadratic convergence rate [10] [11]. Since Kansa's method has been shown to possess spectral convergence [7] in the solution of linear elliptic PDEs, Algorithm 1 is well suited to the RBF collocation framework, especially when the solution domain Ω has an irregular shape (such as that of an expanding fluid).

3 Linearization of the p-Laplace Equation

3.1 Linearization of the PDE

Let us define *flow fluidity* as $S(u) := |\nabla u|^{p-2}$ so that eq. (1) may be rewritten as $H(u) := \text{div}(S(u)\nabla u) = 0$. In order to linearize this equation, we assume that

$$|\nabla u| \gg |\nabla v| \quad (10)$$

and expand $S(u)$ in a Taylor series around the origin, retaining only terms to first order in $|\nabla v| / |\nabla u|$,

$$S(u + v) \approx S(u) + \gamma |\nabla u|^{\gamma-2} (\nabla u \cdot \nabla v) \quad (11)$$

Therefore the truncation criterion is $\left(\frac{|\nabla v|}{|\nabla u|}\right)^2 \approx 0$. It will be useful to define $\vec{K}_u := \gamma |\nabla u|^{\gamma-2} \nabla u$, so that $S(u + v) \approx S(u) + \vec{K}_u \cdot \nabla v$. Now

$$H(u + v) \approx H(u) + \text{div}[S(u)\nabla v] + \text{div}[(\vec{K}_u \cdot \nabla v)\nabla u] + \text{div}[(\vec{K}_u \cdot \nabla v)\nabla v] \quad (12)$$

The only remaining term which is not linear in v is the last one. Notice that

$$|(\vec{K}_u \cdot \nabla v)\nabla v| \leq |\vec{K}_u| \cdot |\nabla v|^2 = \gamma |\nabla u|^{\gamma+1} \left(\frac{|\nabla v|}{|\nabla u|}\right)^2 = \gamma | < \vec{v} > | \left(\frac{|\nabla v|}{|\nabla u|}\right)^2 \approx 0 \quad (13)$$

and therefore

$$\text{div}[(\vec{K}_u \cdot \nabla v)\nabla v] \approx 0 \quad (14)$$

as long as $| < \vec{v} > | = O(1)$. Hence the convenience for normalizing the velocity field when recasting the independent variables into dimensionless form. The third term may be rewritten as

$$\text{div}[(\vec{K}_u \cdot \nabla v)\nabla u] = \text{div}[(\nabla u \cdot \nabla v)\vec{K}_u] \quad (15)$$

It is also worthwhile noticing that

$$\vec{K}_u \cdot (\nabla v \cdot \nabla) \nabla u = \nabla v \cdot (\vec{K}_u \cdot \nabla) \nabla u = \nabla v \cdot \nabla S(u) \quad (16)$$

After some manipulation we finally arrive at

$$H(u + v) \approx H(u) + S(u)\nabla^2 v + \nabla v \cdot \nabla S(u) + \vec{K}_u \cdot (\nabla u \cdot \nabla) \nabla v + \nabla S(u) \cdot \nabla v - \frac{(2 - \gamma)\nabla u \cdot \nabla S(u) - \gamma S(u)\nabla^2 u}{|\nabla u|^2} (\nabla u \cdot \nabla v) \quad (17)$$

Now particularize to $\mathbb{R}^2$ and define

$$A(x, y) = S\left[1 + \frac{\gamma}{|\nabla u|^2} \left(\frac{\partial u}{\partial x}\right)^2\right] \quad (18)$$

$$B(x, y) = \frac{2\gamma S}{|\nabla u|^2} \left(\frac{\partial u}{\partial x}\right) \left(\frac{\partial u}{\partial y}\right) \quad (19)$$

$$C(x, y) = S\left[1 + \frac{\gamma}{|\nabla u|^2} \left(\frac{\partial u}{\partial y}\right)^2\right] \quad (20)$$

$$\vec{D}(x, y) = 2\nabla S + \frac{(\gamma - 2)\nabla u \cdot \nabla S + \gamma S\nabla^2 u}{|\nabla u|^2} \nabla u \quad (21)$$

where we have dropped the u in $S(u)$. Then, in order for $H(u + v) = 0$, the correction v must obey the following *linear* PDE:

$$A \frac{\partial^2 v}{\partial x^2} + B \frac{\partial^2 v}{\partial x \partial y} + C \frac{\partial^2 v}{\partial y^2} + \vec{D} \cdot \nabla v = R \quad (22)$$

where $R(x, y) := -H(v)$ is the residual of the current iterate u to the nonlinear PDE (1). We will refer to R as *nonlinear residual* (to distinguish it from the *linear residual* r , i.e. the residual of the RBF approximation to (22), which will appear in the ensuing discussion). Since $\Delta = B^2 - 4AC = -4(1 + \gamma)S^2 < 0$, (22) is elliptic everywhere and can be always reduced to canonical form. As a guess of the solution (iteration 0), it seems natural to solve a Laplace equation (which would correspond to $\gamma = 0$) with the same BCs as in (2) or (4).

3.2 Linearization of the BCs

Dirichlet-injection case. In this case no linearization is needed and the BCs for the correction are homogeneous if the guess complies with (2),

$$v = p_{IN} - u \text{ (injection)} \quad \frac{\partial v}{\partial n} = -\frac{\partial u}{\partial n} \text{ (walls)} \quad v = -u \text{ (front)} \quad (23)$$

Neumann-injection case. Here, the BC operator $Gu := |\nabla u|^\gamma \frac{\partial u}{\partial n}$ must be linearized

$$G(u + v) \approx |\nabla u|^\gamma \frac{\partial u}{\partial n} + \gamma \frac{\partial u}{\partial n} |\nabla u|^{\gamma-2} (\nabla u \cdot \nabla v) \quad (24)$$

Therefore, the BCs for v are

$$|\nabla u|^\gamma \frac{\partial v}{\partial n} + \gamma \frac{\partial u}{\partial n} |\nabla u|^{\gamma-2} (\nabla u \cdot \nabla v) = q_{IN} - |\nabla u|^\gamma \frac{\partial u}{\partial n} := R_{BC} \text{ (injection)} \quad (25)$$

$$\frac{\partial v}{\partial n} = -\frac{\partial u}{\partial n} \text{ (walls)} \quad v = -u \text{ (front)} \quad (26)$$

where R_{BC} is the residual to the non-linear BC. As the iterations progress and both $R \rightarrow 0$ and $R_{BC} \rightarrow 0$, v becomes the solution of a Laplace PDE with homogeneous BCs (if u_0 complies with the BCs of the non-linear problem) and vanishes.

There are a number of qualitative differences between this problem and the test problem analyzed in [8]. First, the present nonlinearity is a *differential operator* rather than a *function* of the solution. Secondly, both Dirichlet and Neumann BCs must be enforced, instead of only Dirichlet. Finally, the highest gradients take place along the boundary. In order to meet the latter two features, we have slightly modified Fasshauer's Algorithm 1 to incorporate PDEBC.

4 The Test Problem

4.1 Motivation

In order to validate the proposed method, we have solved (1) in a non-trivial domain. It is a square box $[0, 1] \times [-\frac{1}{2}, \frac{1}{2}]$ with an elliptical insert $(\frac{x-1/2}{a})^2 + (\frac{y}{b})^2 = 1$, with $a = 2, b = 3$. The meshless discretization of such a domain is shown (for the injection-Dirichlet case) in Fig.(1). Most of the nodes stem from an $n \times n = 30 \times 30$ grid from which those inside the insert have been removed and $2n = 60$ further nodes have been added in order to model the internal, elliptical wall. There is also an additional layer of nodes at half-grid-constant distance of the boundary, which is intended to improve the accuracy of the RBF approximation close to the boundary, especially along the portions of it with Neumann BCs. As long as there are no coincidental nodes, no minimal distance among nodes has been enforced (it happens to be $h_{min} = 0.0018$). The set of RBF centers is the same as that of collocation nodes plus a set of extra centers required for PDEBC, which are placed outside the mold at a distance $h = 1/29$ (the grid constant) along the outward vector. The exponent γ is 0.6 (which models polyethylene). The mold has been taken from [4] with minor modifications, the most important of which is that the side $x = 1$ has been removed to be replaced by a freely moving frontline.

The RBF that we have used is the scaled Matérn function $M_{2,11,c}$ (see the Appendix for the definition). Since it is positive definite, there is no need for including polynomial terms into the RBF interpolant. The same RBF was used in [17] and found to significantly outperform MQs. Here, we have chosen to restrict c below the ill-condition threshold and found a nearly optimal value of $c = 0.1$, which gives rise (in our collocation point set) to a condition number of about 10^{13} . It performs indeed better than a MQ RBF with an equal or lower condition number (in the best case, the MQ delivers half of the accuracy of $M_{2,11,0.1}$) and, more importantly, is more stable with respect to the condition number. However, the Matérn RBF is more expensive to compute and has two tunable parameters, instead of just one as the MQ.

The injection gate is the segment $x = 0, -\xi \leq y \leq \xi, \xi = 0.15$, highlighted with an arrow in Fig.(1). Along it we have enforced either Neumann (4) and Dirichlet (2) BCs. In the former case, the profile is

$$q_{IN}(y) = 0.9 \left(1 - 3 \left(\frac{y}{\xi} \right)^2 + 2 \left(\frac{|y|}{\xi} \right)^3 \right) \quad (27)$$

such that the exact solution is smooth at both ends $(0, \pm\xi)$ of the injection segment. In the injection-Dirichlet, the prescribed pressure $p_{IN}(y)$ is the exact solution of the problem with injection-Neumann BCs and profile like (27). Consequently, both versions of the problem have an identical solution. In lack of an analytical solution, a FEM approximation computed over 75209 triangles has been used as reference. It is shown in Fig.(2).

Figure 1: Pointset 30×30 in the Dirichlet-injection case. Green/blue/red: PDE/Neumann BCs/Dirichlet BCs nodes. Crosses: RBF centers.

Figure 2: FEM solution.

4.2 Numerical Results

In order to trigger the Newton iterations, an initial guess of the solution must be provided to Algorithm 1. For simplicity, we have used as a guess the solution of the Laplace equation on the same domain and under identical BCs as the non-linear problem. However, other numerical experiments (not reported here) suggest that the scheme is reasonably robust with respect to the starting guess, especially in the Dirichlet-injection case.

Fig.(3) shows the performance of the proposed method throughout the first seven iterations. Concretely, at iteration k , the error ϵ to the FEM solution, the non-linear residual R , and the linear residual (to the linearized equation) r are monitored over a fine mesh of 2228 nodes scattered throughout the domain (and different from the collocation point set). The 'exact' solution values over this evaluation mesh have been linearly interpolated from the much finer FEM computational mesh. The goodness of the estimates ϵ , R , and r , is characterized by its root mean square (RMS) values, which is defined for a vector a of N elements as

$$RMS(a) = \sqrt{\frac{\sum_{i=1}^{i=N} a_i^2}{N}} \quad (28)$$

We first focus on the Dirichlet-injection case (solid line in Fig.(3)). Starting from ≈ 0.01 , the $RMS(\epsilon)$ drops by two orders of magnitude in the two first Newton iterations (exhibiting a clearly superlinear convergence rate), and then stalls (although actually it keeps on dropping for a few more iterations, albeit at a negligible rate). Figs.(6)-(9) demonstrate how the operator-Newton scheme accurately predicts the correction required to match the current error -notice

Figure 3: Convergence of the operator-Newton scheme. Solid/broken line: Dirichlet/Neumann injection BCs. Iteration 0 corresponds to harmonic guess of solution.

that the algorithm itself of course ignores the true solution. Concerning R , it increases at first, and then also drops steadily until leveling off at about the 3rd – 4th iteration. The fact that the *interpolation* non-linear residual R does not drop down to zero (or the error ϵ , for that purpose) is not surprising since the RBF method enforces the sequence of linear PDEs (and therefore the non-linear PDE) on a finite set of collocation nodes only. Indeed, the residual to the non-linear PDE averaged among the collocation nodes *does* drop to about 10^{-10} in some ten Newton iterations (not shown). However, the global accuracy depends on the collocation point set which models the domain and on the RBF interpolation space containing the numerical solution.

For the Neumann-injection case (broken line), the starting guess is shown in Fig.(5). Compared to the FEM solution in Fig.(2), the 0th error is apparent to the naked eye. Next, the $RMS(\epsilon)$ decreases superlinearly during the first

Figure 4: $A(x,y)$ it=2 (Neumann injection).

Figure 5: Initial guess (Neumann injection).

Figure 6: Error it=0.

Figure 7: Correction it=1.

Figure 8: Error it=1.

Figure 9: Correction it=2.

three iterations, but then ‘bounces’ and levels off, after some oscillations, at about 0.001. In both injection versions, the gain in accuracy from the guess is of the same order of magnitude. The performance of the numerical scheme in the Neumann case is understandably poorer, since derivative BCs are traditionally more difficult to deal with and because the accuracy of the RBF interpolation worsens roughly by an order of magnitude per derivative -which also fits the gap between the $RMSs(\epsilon)$: 0.0001 and 0.0013. On the other hand, the asymptotic non-linear residual (chiefly made up of second derivatives) is very similar in both cases: 0.054 and 0.062. The quality of the approximation provided by the RBF-Newton scheme can be assessed by comparison with the accuracy of interpolation of the FEM solution with the same set of nodes and RBFs (without the exterior centers for PDEBC). Such interpolation values are: $RMS(\text{interpolation error})=5.9 \times 10^{-5}$, $RMS(\text{non-linear residual})=0.52$, and $MAX(|\text{nonlinear residual}|)=4.73$ (the residual is based on derivating the interpolation solution).

Finally, it is apparent that the plots of the linear residual r merge with those of R . Moreover, the lines for R and r may cross but their collapsing together neatly signals the end of convergence of the error to the FEM solution. Since both estimates are available without knowledge of the exact solution, this phenomenon could furnish an efficient stopping criterion (without need to wait for the collocation non-linear residual to converge). The numerical equivalence of R and r means that the RBF scheme is no longer capable of capturing the features of the current linear PDE, because they take place *among* collocation nodes. Since the linear residual is defined as

$$r = R - A \frac{\partial^2 v}{\partial x^2} - B \frac{\partial^2 v}{\partial x \partial y} - C \frac{\partial^2 v}{\partial y^2} - \vec{D} \cdot \nabla v \quad (29)$$

the fact that $r = R$ means that the RBF method is effectively ‘seeing’ a Laplace equation, whose solution is zero by virtue of the homogeneous BCs. Therefore, v vanishes and the convergence stagnates. Inversely, R is in turn affected by the linear residual r , which shows overshoots in intricated or oversampled regions (like corners or where several RBF centers are too closely packed). Such spurious oscillations contaminate the non-linear residual R throughout the iterations until eventually masking it up. The final non-linear residual that remains in the pointset after convergence has stalled resembles noise, for it has zero average and is evenly distributed, except for the overshoots -see Fig.(12). The situation could be loosely summed up with the insight that the non-linear asymptotic residual settles at a value which is *not* the true R , but a value of it contaminated with the linear residual of the iterations prior to convergence.

Based on the above analysis, at least two improvement strategies, which turn out to be complementary, may be devised, namely: a) reducing the linear residual at each linearized iteration, and b) filter the spurious features out of the coefficients of the iterated PDEs. Regarding the first approach, it may be

implemented by adaptively choosing the collocation nodes and the RBF centers, while keeping the size of the discretization support approximately constant, or by simply refining the point set (the effect of this on the injection-Neumann problem can be seen in the columns labeled I in Table (2). As for b), we have tried out a simple idea related to the smoothing step of Algorithm 1, which will be explained in the following section.

5 Implicit Smoothing

5.1 Noise Removal in Interpolation Problems

Loosely speaking, the basic idea of denoising is to remove from a set of data those features whose characteristic length is below a given threshold. Convolution of the data with a low-pass filter is one such technique which allows a global interpretation of the denoising process: it attenuates (or supresses) the high frequencies present in the Fourier transform of the data. The usefulness of low-pass filters lies in the following two facts: the spectral components of the noise are typically higher than those of the data (which amounts to a shorter physical scale), and, if the highest frequency corresponding to a genuine structure in the signal can be established *a priori*, the filter cut-off frequency can be adjusted accordingly.

Since the action of a low-pass filter also removes the sharp features from the signal, convolution with a filter is also known as smoothing. The linear nature of the RBF representation enables a simple and unexpensive form of filtering through basic function substitution called *implicit smoothing*. We will just sketch the procedure and refer the reader to [2] for details. Consider the RBF expansion of a function u

$$u = \sum_{i=1}^N \lambda_i \phi(|\vec{x} - \vec{x}_i|) \quad (30)$$

By linearity, convolution with a linear filter ψ is

$$\tilde{u} := u \star \psi = \sum_{i=1}^N \lambda_i \left(\phi(|\vec{x} - \vec{x}_i|) \star \psi \right) \quad (31)$$

Therefore, if

$$\varphi := \phi \star \psi \quad (32)$$

is another RBF, denoising amounts to a change of basis in the interpolation space. In order to perform the filtering, one only has to interpolate the noisy data with the RBF ϕ , retain the expansion coefficients and replace ϕ by the smoothed up RBF φ . Both RBFs are implicitly related by the smoothing parameters (such as the cut-off frequency). The applicability of this technique is restricted to the triplets of viable functions (ϕ, ψ, φ) for which the relation (32) holds. A list

of the possibilities currently available may be found in [2]. Among them, the scaled Matérn RBF is particularly simple, since

$$M_{d,\alpha,c} \star M_{d,\beta,c} = M_{d,\alpha+\beta,c} \quad (33)$$

The scaling parameter c can be interpreted in the physical space as the distance below which detail becomes severely blurred. For a gridded set of collocation points, it is natural to identify $c \approx h$ (with h the grid constant), because any feature below that length should be safely removable without loss of information, since the point set is not expected to capture it in the first place. For a point set made up of scattered data, determination of c is more elusive. We stress the heuristic nature of this approach, which performs surprisingly well in practical interpolation problems [6]. We will next try to take advantage of it in the PDE setting.

5.2 Application to PDEs

In order to illustrate the idea of smoothing up PDE coefficients, consider the following Poisson equation:

$$\nabla^2 u(x, y) = \text{RHS}, (x, y) \in \Omega \quad u(x, y) = f(x, y), (x, y) \in \partial\Omega \quad (34)$$

The domain Ω is a box $[0, 1] \times [-\frac{1}{2}, \frac{1}{2}]$, discretized into 30×30 gridded collocation nodes. $h = 1/29$ is the grid constant and an external layer of RBFs has been added at a distance h off the boundary for PDEBC. The RBF is $M_{2,5,c=h}$ and $f(x, y) = e^{-x^2} \cos(y)$. In order to assess the effect of noise on the RHS of (34), we have carried out three numerical experiments:

- $\text{RHS} = \nabla^2 f(x, y)$ (Clean RHS)
- $\text{RHS} = \nabla^2 f(x, y) + J\delta$ (Noisy RHS without smoothing)
- $\text{RHS} = (\nabla^2 f(x, y) + J\delta) \star M_{2,2,c=h}$ (Noisy RHS smoothed up)

$J = 5$ is the amplitude of the noise and δ is a Gaussian distribution (of mean zero and unit variance and standard deviation). The results of the tests are shown in Table (1). It can be seen that the effect of the filter is small on the accuracy but evident on the residual. In the noisy and unsmoothed case, the residual is dominated by the RHS noise. The Matérn filter $M_{2,2,c=h}$ removes a fair amount of this noise from the residual without essentially affecting the signal, for the errors remain the same. The histogram in Fig.(10) shows the normalized distribution of the 'extracted' noise compared with the (also normalized) noise distribution. Fig.(11) shows the 'remaining' noise in the RHS after smoothing. It is remarkable how the filter has eliminated all of the noise components except those of the smallest amplitude, even though the noise amplitude J is not considered among the smoothing parameters. The points of interest to the application of implicit smoothing to the operator-Newton method are however these two: a) the residual to the elliptic PDE is easily masked up by the PDE

coefficient noise, and b) the convolution with a proper filter can remove an important part of the noise both from the coefficients and from the residual.

Figure 10: RHS(after smoothing) minus RHS(noisy source)

Figure 11: RHS(after smoothing) minus RHS(clean source)

Table 1: Poisson equation with a noisy RHS

	Clean RHS	Noisy RHS without smoothing	Noisy RHS smoothed up
RMS(ϵ)	4.5×10^{-6}	0.012	0.013
MAX(ϵ)	1.9×10^{-5}	0.036	0.038
RMS(r)	0.02	4.80	1.46
MAX(r)	0.11	18.54	6.86

5.3 Operator-Newton Method with PDE Coefficient Smoothing

We have introduced implicit smoothing into the operator-Newton iterations in a different fashion than Algorithm 1. Instead of smoothing up the correction at every iteration, it is the set of coefficients to each linearized PDE that are smoothed up.

Regarding the implementation, we have adopted an approach in which the RBF involved in the operator-Newton iterations is independent from those involved in the smoothing process. At each iteration, the nodal values of each PDE coefficient are computed through Kansa's method with the RBF $M_{2,11,c=0.1}$, and captured on the same set of collocation nodes by the first scaled Matérn in the implicit smoothing triplet, the $M_{2,5,c=l_1}$. Then, they are interpolated on the same nodes by the 'smoothed up' RBF $M_{2,7,c=l_1}$, and this new vector of nodal values represents the smoothed up PDE coefficient.

As long as the collocation pointset remains fixed throughout the iterations, the smoothing of any quantity defined by a nodal vector reduces to a matrix-vector multiplication, where the (interpolation) matrix is computed at the be-

ginning and stored. This makes of implicit smoothing a particularly efficient regularization technique.

Table 2: Effect of smoothing on Newton iterations

it	RMS(ϵ)					
	20×20		30×30		40×40	
0	0.10304		0.10195		0.10205	
1	0.02920		0.03266		0.03382	
2	0.00498	0.00519	0.00322	0.00438	0.00309	0.00358
3	0.00259	0.00253	0.00019	0.00187	0.00032	0.00329
4	0.00387	0.00252	0.00188	0.00172	0.00164	0.00245
5	0.00335	0.00276	0.00112	0.00088	0.00072	0.00208
6	0.00357	0.00267	0.00148	0.00120	0.00117	0.00193
7	0.00348	0.00266	0.00130	0.00097	0.00094	0.00127
8	0.00352	0.00266	0.00139	0.00103	0.00106	0.00153
9	0.00350	0.00264	0.00135	0.00095	0.00100	0.00076
10	0.00351	0.00264	0.00137	0.00095	0.00103	0.00123
11	0.00351	0.00263	0.00136	0.00091	0.00101	0.00047
12	0.00351	0.00263	0.00136	0.00089	0.00102	0.00101
13	0.00351	0.00264	0.00136	0.00087	0.00102	0.00032
14	0.00351	0.00265	0.00136	0.00086	0.00102	0.00085
15	0.00351	0.00266	0.00136	0.00085	0.00102	0.00026
	I	S	I	S	I	S

Table (2) shows the effect of PDE coefficient smoothing on the error in the injection-Neumann BC version. For three different point sets based on 20×20 , 30×30 , and 40×40 grids, RMS(ϵ) is listed without (columns labeled I) and with smoothing (label S). In all cases, the iterated PDEs are solved with the RBF $M_{2,11,0.1}$, whereas the implicit smoothing is implemented with the triplet $M_{2,5,t}$ (capturing RBF), $M_{2,2,t}$ (filter), and $M_{2,7,t}$ (smoothed RBF).

The smoothing length t (scaling parameter of the filter Matérn $M_{2,2,t}$) is set to the corresponding value of the grid constant, *i.e.* $1/19$, $1/29$, and $1/39$. Implicit smoothing is applied from the second Newton iteration only. The reason is that for the two first iterations a superlineal convergence rate is indeed observed and therefore there should be no need for regularization. The effect of smoothing is different depending on the PDE coefficient. Coefficient $A(x, y)$, for instance, is fairly smooth (see Fig.(4)) so that the application of smoothing only slightly ‘files’ the central peak. The effect on R , on the other hand, is quite remarkable as it can be seen in Figs.(12) and (13). Altogether, filtering the coefficients seems to lead to some improvement in the injection-Neumann case, despite the simplicity of the idea. However, in the Dirichlet-injection case (not shown), smoothing lacks any noticeable effect. It might be due to the fact that the error is already very close (twice as much) to the error of interpolating of the exact solution.

Figure 12: Residual it=2.

Figure 13: Smoothed up residual it=2.

6 Conclusions

This paper builds up on our previous work on the Hele-Shaw flow which models plastic injection molding [3] [5] [4]. This problem features a non-linear, unsteady, and free-boundary flow confined inside a potentially irregular 2D geometry, and therefore well suited to meshless discretizations. Here, we have focused on the numerical solution of the p-Laplace equation for the pressure. The numerical solution of non-linear PDEs through asymmetric RBF collocation (Kansa's method) is still under-represented in the literature. Most authors have relied on general-purpose routines such as Powell's method or Levenberg-Marquardt's. We have preferred to investigate the operator-Newton algorithm proposed by G. Fasshauer, which recasts the non-linear PDE into a sequence of linear ones. Thus, the original problem remains inside the conceptual framework of Kansa's method, and can take advantage of its excellent performance in solving elliptic PDEs. Although the linearization of the p-Laplace equation is far from trivial, our numerical examples show good results, with the accuracy of solving the non-linear PDE coming very close to that of interpolating the exact solution after just a few operator-Newton iterations.

In order to improve the results without affecting efficiency, we are currently following two lines of research. The first one looks into an adaptive RBF approximation along the lines of the Greedy Algorithm in [16], and is still under investigation. The second one aims at regularization of the Newton iterates. A particularly unexpensive and RBF-friendly way of performing it is through implicit smoothing, which we have explored in this work with promising results.

7 Acknowledgements

This work has been supported by the Spanish MECD grants MAT2005-05730 and FIS2007-62673 and by Madrid Autonomous Region grant S-0505. F. Bernal acknowledges E. Kansa, G. Fasshauer, R. Beatson and L. Ling for their most useful comments.

A Appendix. The Scaled Matérn RBF and its Derivatives

The scaled Matérn RBF in $\mathbf{R}^d$ is defined as

$$M_{d,\alpha,c}(r) = \frac{1}{c^d \pi^{d/2} 2^{(d+\alpha-2)/2} \Gamma(\alpha/2)} \left(\frac{r}{c}\right)^{(\alpha-d)/2} K_{(\alpha-d)/2}\left(\frac{r}{c}\right) \quad (35)$$

where c is a scaling parameter and $K_\nu(z)$ is the modified Bessel function of the second kind. The parameter $\nu > 0$ controls the smoothness of the given Matérn function, such that $z^\nu K_\nu(z)$ is $\lfloor \nu - 1 \rfloor$ times differentiable.

The value at the origin is

$$M_{d,\alpha,c}(0) = \lim_{r \rightarrow 0} M_{d,\alpha,c}(r) = \frac{\Gamma(\frac{\alpha-d}{2})}{(2c \sqrt{\pi})^d \Gamma(\frac{\alpha}{2})} \quad (36)$$

In order to simplify the notation, define

$$S_\mu(z) = z^\mu K_\mu(z) \quad (37)$$

$$C_{d,\alpha} = \pi^{d/2} 2^{(d+\alpha-2)/2} \Gamma(\alpha/2) \quad (38)$$

$$\mu = \frac{\alpha - d}{2} \quad (39)$$

The derivatives of the scaled Matérn RBF are

$$M_{d,\alpha,c}(r) = \frac{1}{c^d C_{d,\alpha}} S_\mu\left(\frac{r}{c}\right) \quad (40)$$

$$\frac{\partial M_{d,\alpha,c}(r)}{\partial x} = -\frac{1}{c^{d+1} C_{d,\alpha}} \left(\frac{x - x_j}{c}\right) S_{\mu-1}\left(\frac{r}{c}\right) \quad (41)$$

$$\frac{\partial M_{d,\alpha,c}(r)}{\partial y} = -\frac{1}{c^{d+1} C_{d,\alpha}} \left(\frac{y - y_j}{c}\right) S_{\mu-1}\left(\frac{r}{c}\right) \quad (42)$$

$$\frac{\partial^2 M_{d,\alpha,c}(r)}{\partial x^2} = -\frac{1}{c^{d+2} C_{d,\alpha}} \left[S_{\mu-1}\left(\frac{r}{c}\right) - \left(\frac{x - x_j}{c}\right)^2 S_{\mu-2}\left(\frac{r}{c}\right) \right] \quad (43)$$

$$\frac{\partial^2 M_{d,\alpha,c}(r)}{\partial y^2} = -\frac{1}{c^{d+2} C_{d,\alpha}} \left[S_{\mu-1}\left(\frac{r}{c}\right) - \left(\frac{y - y_j}{c}\right)^2 S_{\mu-2}\left(\frac{r}{c}\right) \right] \quad (44)$$

$$\frac{\partial^2 M_{d,\alpha,c}(r)}{\partial x \partial y} = \frac{1}{c^{d+2} C_{d,\alpha}} \frac{(x - x_j)(y - y_j)}{c^2} S_{\mu-2}\left(\frac{r}{c}\right) \quad (45)$$

$$\nabla^2 M_{d,\alpha,c}(r) = \frac{1}{c^{d+2} C_{d,\alpha}} \left[\left(\frac{r}{c}\right)^2 S_{\mu-2}\left(\frac{r}{c}\right) - 2S_{\mu-1}\left(\frac{r}{c}\right) \right] \quad (46)$$

References

- [1] G. Aronsson and U. Janfalk, *On Hele-Shaw flow of power-law fluids*, European J. Appl. Math. **3** 343-336 (1992)
- [2] R. K. Beatson and H.-Q. Bui, *Mollification formulas and implicit smoothing*, Research report UCDMS 2003/19, (2003).
- [3] F. Bernal and M. Kindelan, *An RBF meshless method for injection molding modeling*, Lecture Notes in Computational Science and Engineering, Springer, (2006).
- [4] F. Bernal, M. Kindelan, *RBF meshless modeling of non-Newtonian Hele-Shaw flow* Eng. Anal. Boun. Elem., **31**, 863-874 (2007)
- [5] F. Bernal and M. Kindelan, *Meshless Simulation of Hele-Shaw Flow*, 14th European Conference on Mathematics for Industry ECMI-2006, to appear in Progress in Industrial Mathematics at ECMI 2006, Mathematics in Industry (2006).
- [6] J. C. Carr, R. K. Beatson, B. C. McCallum, W. R. Fright, T. J. McLennan, and T. J. Mitchell, 2003. *Smooth surface reconstruction from noisy range data*. In GRAPHITE 2003, ACM Press, New York, 119126. (2003)
- [7] A. H. D. Cheng, M. A. Golberg, E. J. Kansa, and G. Zanghato *Exponential convergence and h-c multiquadric collocation method for partial differential equations*. Numer. Meth. Partial Diff. Eqns., 2003; 19(5):571594.
- [8] G. E. Fasshauer, *Newton iteration with multiquadrics for the solution of non-linear PDEs*, Comput. Math. Appls. **43**, 423-438 (2002).
- [9] G. E. Fasshauer, *Solving differential equations with radial basis functions: Multilevel methods and smoothing*, Adv. in Comput. Math. **11**, 139-159 (1999).
- [10] G. E. Fasshauer, *On Smoothing for Multilevel Approximation with Radial Basis Functions*, Approximation Theory XI, Vol.II: Computational Aspects, C. K. Chui, and L. L. Schumaker (eds.), Vanderbilt University Press, 55-62 (1999).
- [11] G. E. Fasshauer, E. C. Gartland, and J. W. Jerome *Algorithms Defined by Nash Iteration: Some Implementations via Multilevel Collocation and Smoothing* J. Comp. Appl. Math. **119**, 161-183 (2000).

- [12] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa, *Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary*, Comput. Math. Appl. **43**, 439-455 (2002).
- [13] C. A. Hieber and S. F. Shen, *A finite-element/finite difference simulation of the injection-molding filling process*, Journal of Non-Newtonian Fluid Mechanics, **7**, 1-32 (1979).
- [14] E. J. Kansa, *Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates*, Comput. Math. Appls. **19**, 127-145 (1990).
- [15] E. J. Kansa, *Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations*, Comput. Math. Appls. **19**, 147-161 (1990).
- [16] L. Ling, R. Opfer, and R. Schaback, *Results on Meshless Collocation Techniques*, Eng. Anal. Bound. Elems. **30** (4), 247-253 (2006).
- [17] C. T. Mouat and R. K. Beatson, *RBF Collocation*, Research report UCDMS 2002/3, (2002).