JOURNAL OF INTEGRAL EQUATIONS
AND APPLICATIONS
Volume 7, Number 2, Spring 1995

ACCURATE COMPUTATION OF THE FIELD IN PIPPARD'S NONLOCAL SUPERCONDUCTIVITY MODEL

TAO LIN AND ROBERT C. ROGERS

ABSTRACT. A Galerkin finite element method is analyzed for a class of the Fredholm type integro-differential equations. The method is applied to Pippard's nonlocal superconductivity model. Optimal H^1 norm error estimates are derived for the finite element solution of the current potential. A class of superconvergent post-processing techniques are developed to obtain more accurate approximations to the magnetic field from the finite element solutions. An H^1 semi-norm actual error indicator is derived and is used to generate an adaptive grid refinement procedure. Several numerical examples are presented.

1. Introduction. The purpose of this paper is to develop and analyze the Galerkin finite element solution of the following boundary value problem of a Fredholm type integro-differential equation:

$$Du(x) + \int_{-l}^{l} K(x,y)u(y)\,dy = f(x), \qquad x \in (-l,l), \tag{1.1}$$

$$a_2(-l)u'(-l) = b_0, \qquad a_2(l)u'(l) = b_1. \tag{1.2}$$

We assume throughout this paper that the kernel $K : [-l,l] \times [-l,l] \to \mathbf{R}$ is such that the mapping $\mathcal{K} : L^2(-l,l) \to L^2(-l,l)$ defined by

$$L^2(-l,l) \ni u \mapsto \mathcal{K}u(\cdot) := \int_{-l}^{l} K(\cdot,y)u(y)\,dy \in L^2(-l,l) \tag{1.3}$$

Received by the editors on May 10, 1994, and in revised form on September 1, 1994.
The research of the first author was supported in part by grant DMS-9403844 from the National Science Foundation.
The research of the second author was supported in part by grants DMS-9204304 and DMS-9403844 from the National Science Foundation.

is compact. A sufficient condition for this is that $K(x,y) \in L^2((-l,l)\times(-l,l))$, and thus $K(x,y)$ can have a weak singularity at $x=y$. The differential operator D is defined by

$$Du(x) := -(a_2(x)u'(x))' + a_0(x)u(x), \tag{1.4}$$

for given functions $a_0(x)$ and $a_2(x)$. The motivation of this work is to numerically solve a nonlocal superconductivity model developed by Pippard [**21**] (see also [**2**]). For an infinite slab of thickness $2l$ with magnetic field strength H applied parallel to the slab, the model reduces to

$$-A''(x) + C\int_{-l}^{l} K_1(x-y)A(y)\,dy = 0, \qquad x \in (-l,l), \tag{1.5}$$

$$A'(-l) = A'(l) = H. \tag{1.6}$$

Here $A(x)$ is the current potential and

$$K_1(x-y) := \left(\frac{|x-y|}{2\xi} - \frac{1}{2}\right)e^{-|x-y|/\xi} + \left(1 - \frac{|x-y|^2}{2\xi^2}\right)\int_{|x-y|}^{\infty} \frac{e^{-u/\xi}}{u}\,du.$$

Compared with works on Volterra type integro-differential equations (see Brunner [**6**] and the references there) there are relatively few publications about the boundary value problems of Fredholm type integro-differential equations. The available numerical methods can be roughly divided into two classes: indirect methods and direct methods. In indirect methods (see Baker [**1**] and Linz [**20**]) an integro-differential equation is first transformed into an integral equation of Fredholm type which is then solved by available numerical methods for integral equations. In direct methods (see Delves and Mohamed [**10**], Espinosa [**11**], Volk [**25**], and Garey [**12**]), certain projection methods such as Galerkin finite element or collocation methods are applied directly to the integro-differential equations.

However, efficient numerical approximations to the solution of the nonlocal superconductivity model considered here cannot be generated by simple applications of the above general approaches. The first difficulty in using the conventional methods is that the kernel in

the integral operator itself is given by an integration over an infinite interval. A numerical method inevitably has to spend a lot of time on forming the matrix involving this kernel, since many integrations in multiple dimensions have to be carried out. Using an indirect method will make this even worse because the kernel in the integral equation solved by this indirect method is the integration of the kernel of the integro-differential equation. The computational cost spent in forming the matrix involving the integral kernel becomes more critical when the more general nonlinear model developed recently by Brandon and Rogers [**3**] is solved by Newton type iteration.

The second difficulty is due to our desire to model the magnetic penetration depth of a superconductor. A superconductor in a weak applied field can expel the magnetic field from its interior. Beyond a thin layer from the surface of a superconductor, the magnetic field becomes close to zero. The thickness of this layer is called the magnetic penetration depth and is an important physical characteristic of superconductors. Thus, we expect that the derivative of the solution to (1.5) and (1.6) is close to zero in most of the interval, but tends to the $H \neq 0$ quickly in the vicinity of the boundaries. An accurate approximation of the field is needed in this boundary layer to understand the magnetic penetration depth of a superconductor. The conventional methods generate the approximations to the field by numerically differentiating the approximate current potential. Typically the accuracy is lost in the numerical differentiation. We present here a class of techniques to generate approximations to the field with better convergence rates by using the superconvergence property of the finite element solutions.

The superconvergence of the approximations to $A(x)$ for either integral and integro-differential equations can be found in [**22, 7, 6, 23, 26**]. For Fredholm type integro-differential equation, certain superconvergence of the numerical solution to $A'(x)$ based on a collocation method was given in [**13**]. Brunner [**4, 5**] showed interesting local superconvergence properties of the numerical solutions to $A'(x)$ generated by some collocation methods for the second order Volterra type integro-differential equations. However, we are unaware of any results on how to use an approximation to the potential to generate an approximation to the field $A'(x)$ with a better global convergence rate. It is our intention here to generate the interpolated finite element solution developed in [**14, 15, 16, 17, 18, 19**] for partial differential equations

to the Fredholm type integro-differential equations.

The outline of the next four sections is as follows. In Section 2, a Galerkin finite element method will be derived for (1.1) and (1.2). The standard error estimate of this Galerkin method will be given. Section 3 presents and analyzes several superconvergent post-processing procedures which can generate better global approximations to the field from a finite element approximation to the potential. The first part of Section 4 presents some numerical examples to show the effects of the post-processing techniques. The second part discusses how the superconvergent approximations to the field can be used to construct indicators for the actual errors and how to use these indicators to form adaptive grid refinement procedures. The last part of this section discusses how to implement the Galerkin method efficiently for the Pippard's nonlocal superconductivity model. Section 5 contains conclusions.

2. Galerkin finite element solutions and error estimates. In this paper we will use the standard Sobolev spaces $H^r(-l,l)$ whose norm is defined by

$$||v||_r := \left(\sum_{k=0}^{r} \int_{-l}^{l} |v^{(k)}(x)|^2 \, dx \right)^{1/2},$$

and whose semi-norm is defined by

$$|v|_r := \left(\int_{-l}^{l} |v^{(r)}(x)|^2 \, dx \right)^{1/2}.$$

The weak formulation of (1.1) and (1.2) is: Find $u(x) \in H^1(-l,l)$ such that

$$\begin{aligned} a(u,v) + (\mathcal{K}\, u, v) = (f,v) + b_1 v(l) - b_0 v(-l), \\ \forall\, v \in H^1(-l,l), \end{aligned} \tag{2.7}$$

where

$$(u,v) := \int_{-l}^{l} u(x) v(x) \, dx,$$

and

$$a(u,v) := \int_{-l}^{l} a_2(x)u'(x)v'(x)\,dx + \int_{-l}^{l} a_0 u(x)v(x)\,dx. \tag{2.8}$$

We begin with a standard abstract existence result.

Lemma 2.1. *Suppose $a_2(x)$ and $a_0(x)$ are bounded and sufficiently smooth, the operator $\mathcal{K} : L^2(-l,l) \to L^2(-l,l)$ is compact, and the bilinear form of $D+\mathcal{K}$ is V-elliptic, i.e., there exists a positive α such that*

$$a(u,u) + (\mathcal{K}u, u) \geq \alpha ||u||_1, \quad \forall\, u \in H^1(-l,l). \tag{2.9}$$

Then for any $f \in L^2(-l,l)$ the boundary value problem (1.1) *and* (1.2) *has a unique solution $u(x) \in H^2(-l,l)$.*

Proof. Without losing generality, we assume that the boundary value problem has a homogeneous boundary condition. Consider the differential operator D and the integral operator $\mathcal{K}$ as operators in $L^2(-l,l)$ with the same domain as follows:

$$\mathcal{D} := \{v \in H^2(-l,l) \mid v'(-l) = v'(l) = 0\}. \tag{2.10}$$

Then $\mathcal{K}$ is compact and D is normally solvable, i.e., it is a closed operator with closed range. Hence $D+\mathcal{K}$ is also normally solvable. Therefore, we have

$$L^2(-l,l) = \mathcal{R} \oplus \mathcal{N},$$

where $\mathcal{R}$ is the range and $\mathcal{N}$ is the null manifold of $D+\mathcal{K}$. Because of the V-ellipticity of $D+\mathcal{K}$, $\mathcal{N} = \{0\}$. This gives $\mathcal{R} = L^2(-l,l)$, consequently the result of this lemma. □

We now describe the finite element approximation to the exact solution. Let Π_h be a partition of the interval $[-l,l]$ formed by

$$-l = x_1 < x_2 < \cdots < x_{n-1} < x_n = l,$$

with

$$h := \max_{i=1,2,\ldots,n-1} |x_{i+1} - x_i|$$

being the largest diameter of the subintervals. With this partition, we introduce the following standard Lagrange type finite element spaces:

$$S_h^r := \{v(x) \in C[-l,l] \mid v(x)|_{[x_i,x_{i+1}]} \in P_r,\ i = 1, 2, \ldots, n-1\}, \tag{2.11}$$

where $r = 1$ or $r = 2$ throughout this paper, and P_r stands for the space formed by polynomials of degree at most r. Then S_h^r is a finite dimensional subspace of $H^1(-l,l)$ with the following approximation property [**24**]:

$$\begin{gathered} ||v - I_h^r v||_0 + h||v - I_h^r v||_1 \le Ch^{r+1}||v||_{r+1}, \\ \forall\, v \in H^{r+1}(-l,l), \end{gathered} \tag{2.12}$$

where $I_h^r v(x)$ is the interpolation of $v(x)$ in S_h^r. The finite element approximation $u_h(x) \in S_h^r$ is then defined by

$$\begin{gathered} a(u_h, v_h) + (\mathcal{K}\, u_h, v_h) = (f, v_h) + b_1 v_h(l) - b_0 v_h(-l), \\ \forall\, v_h \in S_h^r. \end{gathered} \tag{2.13}$$

The V-ellipticity of $a(u,v)$ ensures that (2.13) has a unique solution in S_h^r for sufficiently small h whose accuracy is stated in the following theorem.

Theorem 2.1. *Suppose that $a_2(x)$ and $a_0(x)$ are bounded, $f(x) \in L^2(-l,l)$ and the V ellipticity* (2.9) *is satisfied. If the exact solution $u(x)$ is in $H^{r+1}(-l,l)$, then the finite element solution given by* (2.13) *has the following error estimate*:

$$||u - u_h||_0 + h||u - u_h||_1 \le Ch^{r+1}||u||_{r+1}.$$

Proof. Since $\mathcal{K}$ is compact in $L^2(-l,l)$, the bilinear form $a(u,v) + (\mathcal{K}\, u, v)$ is continuous. Then, because of the V-ellipticity of $a(u,v) + (\mathcal{K}\, u, v)$ we can apply Cea's lemma [**9**] and the approximation property of S_h^r to obtain

$$h||u - u_h||_1 \le Ch^{r+1}||u||_{r+1}. \tag{2.14}$$

The result of Lemma (2.1) ensures that the Aubin-Nitsche's technique can be applied to obtain

$$||u - u_h||_0 \leq Ch^{r+1}||u||_{r+1} \tag{2.15}$$

from (2.14). The result of this theorem is a combination of (2.14) and (2.15). □

3. Approximate the field by post-processing. The error estimate given by Theorem 2.1 is optimal from the point of view of the approximation capability of S_h^r. Consequently, the derivative of $u_h(x)$ is a lower-order approximation to $u'(x)$ than $u_h(x)$ itself as an approximation to $u(x)$. For example, if linear finite elements are used, $u_h(x)$ is an $O(h^2)$ approximation to $u(x)$, but $u_h'(x)$ is only an $O(h)$ approximation to $u'(x)$. More accurate approximation to $u'(x)$ is desirable since in our computation on Pippard's model $u'(x)$ represents the field, which is an important characteristic in superconductivity modeling. Conventionally, better approximations to $u'(x)$ are obtained by either the 'h' approach or the 'p' approach. In the 'h' approach, smaller partition parameter h is used to maintain the accuracy, while in the 'p' approach finite elements with polynomials of higher degrees are used to maintain the accuracy. Each of these two approaches results in more expensive systems to solve.

However, the finite element solution $u_h(x)$ does contain more information about $u'(x)$ which is not revealed by $u_h'(x)$. This can be seen from the following lemma.

Lemma 3.1. *If all the conditions of Theorem* 2.1 *are satisfied, then*

$$||I_h^r u - u_h||_1 \leq Ch^{r+1}||u||_{r+1}. \tag{3.16}$$

Proof. Notice first the following well-known weak superconvergence result [**8**]:

$$|a(u - I_h^r u, v_h)| \leq Ch^{r+1}||u||_{r+1}|v_h|_1. \tag{3.17}$$

Then the V-ellipticity and continuity of $a(u,v)+(\mathcal{K}u,v)$, the approximation property of S_h^r, and Theorem 2.1 ensure

$$\begin{aligned} C||I_h^r u - u_h||_1^2 &\le a(I_h^r u - u_h, I_h^r u - u_h) + (\mathcal{K}(I_h^r u - u_h), I_h^r u - u_h) \\ &= a(I_h^r u - u, I_h^r u - u_h) + (\mathcal{K}(I_h^r u - u), I_h^r u - u_h) \\ &\le Ch^{r+1}||u||_{r+1}|I_h^r u - u_h|_1 + Ch^{r+1}||u||_{r+1}||I_h u - u_h||_0, \end{aligned}$$

which consequently gives (3.16). □

This lemma indicates that the first derivative of the finite element solution $u_h(x)$ is a better approximation to $(I_h^r u(x))'$ than to $u'(x)$ and suggests that $u_h(x)$ contains some information about $u'(x)$ that is not in $u_h'(x)$. The important question is how to extract more information about $u'(x)$ from $u_h(x)$. Notice that the derivative of the finite element solution is a piecewise polynomial with a lower degree than that of $u_h(x)$ and naturally does not have the same approximation capability as $u_h(x)$. This leads us to consider using $u_h(x)$ to construct piece-wise polynomials with higher degrees for better approximations to $u'(x)$. We present below several specific ways to realize this post-processing idea.

The techniques in the first class are based on piecewise polynomial interpolation. For a linear finite element solution $u_h(x)$, we first use the nodes of Π_h to form a coarser partition Π_{2h} of $[-l,l]$:

$$\Pi_{2h} := \{a = z_1 < z_2 < \cdots < z_m\}, \tag{3.18}$$

where

$$m = \begin{cases} (n+1)/2, & \text{if } n \text{ is odd,} \\ (n+2)/2, & \text{if } n \text{ is even,} \end{cases} \tag{3.19}$$

and

$$z_i = x_{2i-1}, \qquad i = 1, 2, \ldots, m-1, \qquad z_m = x_n. \tag{3.20}$$

Most of the elements of Π_{2h} contains three nodes of Π_h, for example $[z_i, z_{i+1}]$ contains $x_{2i-1}, x_{2i}, x_{2i+1}$ if $i \le m-2$. We then define a function $I_q u_h(x)$ on $[z_i, z_{i+1}]$ to be the quadratic interpolation of $(x_{2i-1}, u_h(x_{2i-1})), (x_{2i}, u_h(x_{2i})), (x_{2i+1}, u_h(x_{2i+1}))$. If n

is odd, $I_q u_h(x)$ in $[z_{m-1}, z_m]$ can be defined in the same way as in the other intervals of Π_{2h}. For even n we define $I_q u_h(x)$ in $[z_{m-1}, z_m]$ to be the restriction of the quadratic interpolation of $(x_{n-2}, u_h(x_{n-2})), (x_{n-1}, u_h(x_{n-1})), (x_n, u_h(x_n))$. The function $I_q u_h(x)$ defined here obviously has the following properties:

$$\begin{aligned} &I_q u_h(x) \in C^0[-l, l], \\ &I_q u_h(x)|_{[x_i, x_{i+1}]} \text{ is a quadratic polynomial,} \\ &I_q u_h(x_i) = u_h(x_i), \qquad i = 1, 2, \ldots, n-1. \end{aligned} \tag{3.21}$$

We note that Q. Lin, H. Wang and T. Lin [**14, 15**] used the same procedure to post-process the linear finite element solution for partial differential equations.

We can post-process the quadratic finite element solution by a similar procedure which was developed for partial differential equations in [**16**]. Assume that the midpoint $x_{i+1/2}$ of each element $[x_i, x_{i+1}]$, $i = 1, 2, \ldots, n-1$, is also a node where the finite element space S_h^2 has a nodal basis function. Let

$$z_i = x_{(i+1)/2}, \qquad i = 1, 2, \ldots, 2n-1.$$

Then we form a new partition $\Pi_{3h/2}$ of $[-l, l]$ as follows:

$$\Pi_{3h/2} := \{-l = z_1 < z_4 < z_7 < \cdots < z_{m-1} < z_m \le z_{2n-1} = l\},$$

where

$$m = \begin{cases} 2n-2, & \text{if } \operatorname{mod}(n,3) = 0, \\ 2n-1, & \text{if } \operatorname{mod}(n,3) = 1, \\ 2n-3, & \text{if } \operatorname{mod}(n,3) = 2. \end{cases}$$

The subscript $3h/2$ in $\Pi_{3h/2}$ implies that $\Pi_{3h/2}$ is a partition of $[-l, l]$ whose elements are formed by one and one half elements of Π_h except the element on the right. In each $[z_{i-3}, z_i]$, $i = 4, 7, 11, \ldots, m$, we let $I_c u_h(x)$ be the cubic interpolation polynomial of $(z_{i-3}, u_h(z_{i-3})), (z_{i-2}, u_h(z_{i-2})), (z_{i-1}, u_h(z_{i-1})), (z_i, u_h(z_i))$. To finish the construction of $I_c u_h(x)$ over the rest of $[-l, l]$, we let $I_c u_h(x)$ be the restriction of the cubic interpolation polynomial of $(z_{2n-4}, u_h(z_{2n-4}))$, $(z_{2n-3}, u_h(z_{2n-3})), (z_{2n-1}, u_h(z_{2n-1})), (z_{2n-1}, u_h(z_{2n-1}))$ in $[z_{2n-2}, z_{2n-1}]$

if mod $(n,3)=0$ or in $[z_{2n-3}, z_{2n-1}]$ if mod $(n,3)=2$. It is easy to see that this new function $I_c u_h(x)$ has the following properties:

$$\begin{aligned} &I_c u_h(x) \in C^0[-l,l], \\ &I_c u_h(x)|_{[x_i,x_{i+1}]} \text{ is a cubic polynomial,} \\ &I_c u_h(z_i) = u_h(z_i), \qquad i = 1,2,\ldots,2n-1. \end{aligned} \tag{3.22}$$

The second class of post-processing techniques are based on spline interpolation. Notice that the values of the first derivative of the exact solution are available at the end points of the interval, and they can be used to do post-processing by spline interpolation. For the linear finite element solution $u_h(x) \in S_h^1$, we can define $I_q u_h(x)$ to be its quadratic spline interpolant such that

$$\begin{aligned} &(I_q u_h(-l))' = u'(-l), \qquad I_q u_h(x_i) = u_h(x_i), \\ &\qquad\qquad i = 1,2,\ldots,n. \end{aligned} \tag{3.23}$$

The quadratic spline function satisfying (3.23) is uniquely defined and satisfies (3.21). Obviously, the derivative condition in (3.23) can also be imposed on the right end point of the interval $[-l,l]$.

For the quadratic finite element solution $u_h(x) \in S_h^2$, we can define $I_c u_h(x)$ to be the unique clamped cubic spline interpolant of $u_h(x)$ such that

$$\begin{aligned} &I_c u_h(z_i) = u_h(z_i), \qquad i = 1,2,\ldots,2n-1, \\ &(I_c u_h(-l))' = u'(-l), \qquad (I_c u_h(l))' = u'(l), \end{aligned} \tag{3.24}$$

where the z_i's are the nodes of the quadratic finite elements defined on Π_h. It is easy to see that this clamped cubic spline satisfies (3.22). The natural cubic spline interpolation can also be used to post-process the quadratic finite element solutions.

We now turn to the error estimates of $I_q u_h(x)$ and $I_c u_h(x)$. As preparation, we first list some properties of the post-processing operators. As usual, the constant C in the following may have different values at different places.

Lemma 3.2. *If the post-processing operators I_q and I_c are defined by piecewise polynomial interpolation, then there exists a constant C such that for any $v \in H^1(-l,l)$,*

$$||I_q v||_1 \le C||v||_1, \qquad ||I_c v||_1 \le C||v||_1.$$

Proof. These can be obtained following the same arguments used in [**16**]. □

Lemma 3.3. *If the post-processing operators I_q and I_c are defined by spline interpolation, then there exists a constant C such that*

$$||I_q v||_1 \le C||v||_1, \qquad \forall\, v \in S_h^1 \cap H_0^1(-l,l),$$
$$||I_c v||_1 \le C||v||_1,$$

for all $v \in S_h^2 \cap H_0^1(-l,l)$ if $I_c v$ is a clamped cubic spline, and for all $v \in S_h^2$ if $I_c v$ is a natural cubic spline.

Proof. We only prove the case in which $I_c v$ is formed by the clamped cubic spline interpolation. The other cases follow in a similar fashion. In the clamped cubic spline case, $I_c v(x)$ has the following piecewise representation:

$$I_c v(x) = A_j \frac{(z_{j+1}-x)^3}{6s_{j+1}} + A_{j+1}\frac{(x-z_j)^3}{6s_{j+1}} + B_j(x-z_j) + C_j, \qquad \forall\, x \in [z_j, z_{j+1}],$$

for $j = 1, 2, \ldots, 2n-2$, $s_{j+1} = z_{j+1} - z_j$ and z_j's are defined as before. Then

$$|A_j| \le \max_{i=0}^{2n-1} |D_i|, \qquad j = 1, 2, \ldots, 2n-1,$$

where

$$D_j = \frac{6}{s_j + s_{j+1}} \left\{ \frac{v(z_{j+1}) - v(z_j)}{s_{j+1}} - \frac{v(z_j) - v(z_{j-1})}{s_j} \right\}, \qquad j = 2, 3, \ldots, 2n-2$$
$$D_0 = \frac{6}{s_1}\frac{v(z_2) - v(z_1)}{s_1},$$
$$D_{2n-1} = -\frac{6}{s_{2n-1}}\frac{v(z_{2n-1}) - v(z_{2n-2})}{s_{2n-1}}.$$

Using the relationships

$$B_j = \frac{v(z_{j+1}) - v(z_j)}{s_{j+1}} - \frac{s_{j+1}}{6}(A_{j+1} - A_j),$$
$$C_j = v(z_j) - A_j \frac{s_{j+1}^2}{6},$$

we have

$$\begin{aligned} ||I_c v||_1^2 &\le C \sum_{i=1}^{2n-2} \left| \frac{v(z_{i+1}) - v(z_i)}{s_{i+1}} \right|^2 \\ &\le C \sum_{i=1}^{2n-2} \int_{z_i}^{z_{i+1}} (v'(z))^2 \, dz \\ &\le C \int_{-l}^{l} (v'(z))^2 \, dz, \end{aligned}$$

which gives the result for clamped cubic spline interpolation. □

Lemma 3.4. *For any $v \in H^1(-l, l)$ and any $x \in [-l, l]$, we have*

$$I_q I_h^1 v(x) = I_q v(x), \qquad I_c I_h^2 v(x) = I_c v(x),$$

where $I_h^1 v(x)$ and $I_h^2 v(x)$ are the interpolation of $v(x)$ in the finite element spaces S_h^1 and S_h^2, respectively.

Proof. These are the consequences of (3.21) and (3.22). □

Using these results, we can show that the first derivatives of $I_q u_h(x)$ and $I_c u_h(x)$ have better convergence rates than $u_h'(x)$ for $u_h \in S_h^1$ and S_h^2, respectively.

Theorem 3.1. *Under the same conditions as in Lemma* 3.1*, we have*

$$||I_q u_h - u||_1 \le C h^2 ||u||_3, \tag{3.25}$$

for $u_h \in S_h^1$ and $u \in H^3(-l, l)$, and

$$||I_c u_h - u||_1 \le C h^3 ||u||_4, \tag{3.26}$$

for $u_h \in S_h^2$ and $u \in H^4(-l, l)$.

Proof. Again, we prove only (3.26) for the case in which I_c is defined by the cubic spline interpolation. The other case has a similar proof. First, for $u_h \in S_h^2$ we have

$$||I_c u_h - u||_1 \le ||I_c u_h - I_c I_h^2 u||_1 + ||I_c I_h^2 u - u||_1. \tag{3.27}$$

Then, by Lemma 3.3 and Lemma 3.1,

$$||I_c u_h - I_c I_h^2 u||_1 \le C||u_h - I_h^2 u||_1 \le Ch^3||u||_4. \tag{3.28}$$

According to Lemma 3.4 and the approximation capability of the cubic spline interpolant,

$$||I_c I_h^2 u - u||_1 = ||I_c u - u||_1 \le Ch^3||u||_4.$$

Applying the above and (3.28) to (3.27) yields (3.26). □

4. Numerical examples and application.

4.1. *Numerical examples for the superconvergence.* To see the effect of the post-processing techniques, we applied them to the following boundary value problem defined in the interval $x \in (-1, 1)$:

$$-((2+\sin(x))u'(x))' + u(x) + \int_{-1}^{1} (1+\sin(x+y))u(y)\,dy = f(x), \tag{4.29}$$

$$u'(-1) = b_0, \qquad u'(1) = b_1, \tag{4.30}$$

where $f(x)$, b_0 and b_1 were chosen such that $u(x) = \sin(x)$ was the exact solution of this boundary value problem. The errors in the H^1 semi-norm of the numerical results by both the linear and the quadratic finite elements are listed in Table 1 and Table 2, respectively. The post-processing techniques obviously generated better approximations to the first derivative. For example, the approximation to the first derivative by post-processing the linear finite element solution is almost as good as the derivative of the quadratic finite element finite element solution (see the third column of Table 1 and the second column of Table 2). Using the discrete Green's function, we can also show that the results in Theorem 3.1 are also true in uniform norm, i.e., we can show that

$$||(I_q u_h)' - u'||_{L^\infty(-l,l)} \le Ch^2||u||_{3,\infty}, \quad \text{for } u_h \in S_h^1 \tag{4.31}$$

and

$$||(I_c u_h)' - u'||_{L^\infty(-l,l)} \le Ch^3||u||_{4,\infty}, \quad \text{for } u_h \in S_h^2. \tag{4.32}$$

TABLE 1. Numerical results by linear finite elements.

h	$\|u' - u_h'\|_0$	$\|u' - (I_q u_h)'\|_0$	$\|(I_q u_h)' - u_h'\|_0$
2/6	0.13925066	0.06210226	0.12084604
2/10	0.06529805	0.02229592	0.06035870
2/14	0.03955717	0.01136743	0.03749395
2/18	0.02718226	0.00687463	0.02610515
2/22	0.02013842	0.00460136	0.01949615
2/26	0.01568580	0.00329419	0.01526757
2/30	0.01266199	0.00247418	0.01237208
2/34	0.01049849	0.00192620	0.01028793
2/38	0.00888780	0.00154199	0.00872923
2/42	0.00765058	0.00126224	0.00752768
2/46	0.00667593	0.00105225	0.00657841
2/50	0.00589196	0.00089062	0.00581305

TABLE 2. Numerical results by quadratic finite elements. $I_c u_h$ in the fourth and fifth columns are formed by the clamped cubic interpolation.

h	$\|u' - u_h'\|_0$	$\|u' - (I_c u_h)'\|_0$	$\|(u' - (I_c u_h)'\|_0$	$\|(I_c u_h)' - u_h'\|_0$
2/3	0.05990817	0.01589577	0.01380605	0.06020735
2/5	0.02103984	0.00346630	0.00299768	0.02105423
2/7	0.01065099	0.00126813	0.00109087	0.01064812
2/9	0.00642027	0.00059797	0.00051244	0.00641692
2/11	0.00428942	0.00032797	0.00028034	0.00428694
2/13	0.00306736	0.00019888	0.00016970	0.00306561
2/15	0.00230203	0.00012955	0.00011040	0.00230077
2/17	0.00179118	0.00008904	0.00007581	0.00179026
2/19	0.00143331	0.00006381	0.00005428	0.00143262
2/21	0.00117290	0.00004727	0.00004019	0.00117237
2/23	0.00097752	0.00003599	0.00003059	0.00097711
2/25	0.00082719	0.00002803	0.00002381	0.00082687

Instead of repeating the same arguments as in the proof of Theorem 3.1, we just present the errors in the uniform norm of the numerical results by both the linear and the quadratic finite elements in Table 3 and Table 4, respectively, where the norm $||\cdot||_\infty^*$ is defined by

$$||u||_\infty^* := \max_{i=1}^{n-1} \max_{j=1}^{10} |u(x_{i,j})|, \tag{4.33}$$

$$x_{i,j} := x_i + (j-1)h_i, \qquad j = 1, 2, \ldots, 10,$$
$$h_i = x_{i+1} - x_i.$$

Using linear regression one can easily see that the data in these tables satisfy (4.31) or (4.32).

4.2. *An adaptive grid refinement procedure.* As in any other numerical simulations, it is critical to know whether a particular finite element solution $u_h(x)$ is within a given error tolerance tol > 0 before we can make any further decision based on $u_h(x)$. Even though the *a priori* error estimates given in Theorem 2.1 describe the asymptotic behavior of the finite element solutions, they cannot tell us accurately how large the actual error might be for a particular computation because the constant C depends on the unknown $u(x)$. Without losing generality, we assume that the H^1 semi-norm is used in checking the error tolerance. The immediate question is how to measure the actual error $u_h(x) - u(x)$ in the H^1 semi-norm.

Notice that $u_h(x)$ is an $O(h^{r+1})$ approximation to $u(x)$, but $u_h'(x)$ is only an $O(h^r)$ approximation to $u'(x)$. Let $G_h(x)$ be either $I_q u_h(x)$ or $I_c u_h(x)$. Theorem 3.1 has shown that $G_h'(x)$ is a higher order approximation of $u'(x)$ than $u_h'(x)$. Therefore we can use the computable quantity $||u_h' - G_h'||_0$ as an indicator to measure $|u_h - u|_1$ because

$$\begin{aligned} u_h' - u' &= (u_h' - G_h') + (G_h' - u') \\ &\approx (u_h' - G_h') + O(h^{r+1}), \quad \text{for } u_h \in S_h^r, \end{aligned}$$

i.e., $u_h' - G_h'$ is the dominant part of the actual error $|u_h - u|_1$. A comparison of the second columns with the last columns of Tables 1–4 confirms that the error indicators formed by using the post-processing techniques in the above way are very accurate.

TABLE 3. Numerical results by linear finite elements.
The norm $||\cdot||_\infty^*$ is defined by (4.44).

h	$\|u' - u_h'\|_\infty^*$	$\|u' - (I_q u_h)'\|_\infty^*$	$\|(I_q u_h)' - u_h'\|_\infty^*$
2/6	0.13310356	0.03961181	0.10241148
2/10	0.08172053	0.01411752	0.07153395
2/14	0.05889190	0.00716220	0.05393553
2/18	0.04602173	0.00432334	0.04310547
2/22	0.03776503	0.00289987	0.03584794
2/26	0.03201928	0.00207639	0.03066412
2/30	0.02779058	0.00155866	0.02678232
2/34	0.02454831	0.00121250	0.02376906
2/38	0.02198343	0.00096983	0.02136322
2/42	0.01990375	0.00079323	0.01939845
2/46	0.01818351	0.00066100	0.01776392
2/50	0.01673695	0.00055953	0.01638298

TABLE 4. Numerical results by quadratic finite elements. $I_c u_h$ in the fourth and fifth columns are formed by the piecewise cubic interpolation.
The norm $||\cdot||_\infty^*$ is defined by (4.44).

h	$\|u' - u_h'\|_\infty^*$	$\|u' - (I_c u_h)'\|_\infty^*$	$\|u' - (I_c u_h)'\|_\infty^*$	$\|(I_c u_h)' - u_h'\|_\infty^*$
2/3	0.04548986	0.01360068	0.00653876	0.04456780
2/5	0.01514414	0.00297448	0.00140638	0.01514020
2/7	0.00745968	0.00108656	0.00053553	0.00747676
2/9	0.00442348	0.00051136	0.00025479	0.00443348
2/11	0.00292334	0.00028000	0.00013903	0.00292886
2/13	0.00207434	0.00016956	0.00008361	0.00207753
2/15	0.00154779	0.00011049	0.00005476	0.00154973
2/17	0.00119891	0.00007607	0.00003769	0.00120016
2/19	0.00095594	0.00005455	0.00002696	0.00095677
2/21	0.00077997	0.00004043	0.00001990	0.00078055
2/23	0.00064847	0.00003077	0.00001517	0.00064888
2/25	0.00054761	0.00002396	0.00001183	0.00054791

Once a finite element space is chosen, the accuracy of the finite element solution can be maintained by using an appropriate partition. From the point of view of both the accuracy and efficiency, more mesh points should be put at the places where the exact solution changes quickly than where the solution changes slowly. The superconvergence property of $G_h(x)$ not only allows us to know more accurately how exact the solution changes, but also can be used in the error indicator above to generate an adaptive grid refinement procedure for a given error tolerance, tol .

Let us consider the quadratic finite element solution and the related post-processing operator I_c. According to the optimal error estimate (2.14), we can assume that for any subinterval $[a, b] \subset [-l, l]$ there exists a constant C such that

$$|(u_h - u)|_{[a,b]}|_1 = Ch^2,$$

where $(u_h - u)|_{[a,b]}$ stands for the restriction of $u_h - u$ in $[a, b]$. We first use a trial step size $\tilde{h}$ to form a trial partition $\Pi_{\tilde{h}}$:

$$-l = \tilde{x}_0 < \tilde{x}_1 < \cdots < \tilde{x}_{\tilde{n}} = l,$$

with $\tilde{h} = \max_{i=1}^{\tilde{n}-1}\{x_{i+1} - x_i\}$. Then we partition $[\tilde{x}_i, \tilde{x}_{i+1}]$ into smaller subintervals $\tilde{x}_i = x_{i1} < x_{i2} < \cdots < x_{in_i} = \tilde{x}_{i+1}$, such that $x_{i(j+1)} - x_{ij} = q(\tilde{x}_{i+1} - \tilde{x}_i) = q\tilde{h}_i$, $j = 1, 2, \ldots, n_i$, only if the restriction of the actual error $u_{\tilde{h}} - u$ in $[\tilde{x}_i, \tilde{x}_{i+1}]$ seems to be larger than the error tolerance tol , i.e.,

$$C\tilde{h}_i^2 = |(u_{\tilde{h}} - u)|_{[\tilde{x}_i,\tilde{x}_{i+1}]}|_1 \approx |(u_{\tilde{h}} - I_c u_{\tilde{h}})|_{[\tilde{x}_i,\tilde{x}_{i+1}]}|_1 \geq \text{tol}\,.$$

If we use $\{x_{ij} \mid i = 1, 2, \ldots, n, j = 1, 2, \ldots, n_i\}$ to form a new partition Π_h, then, by the superconvergence property of $I_c u_{\tilde{h}}(x)$, the related finite element solution $u_h \in S_h^2$ satisfies

$$\begin{aligned} |(u_h - u)|_{[\tilde{x}_i,\tilde{x}_{i+1}]}|_1 &= C(q\tilde{h}_i)^2 = q^2(C\tilde{h}_i^2) \\ &= q^2|(u_{\tilde{h}} - u)|_{[\tilde{x}_i,\tilde{x}_{i+1}]}|_1 \\ &\approx q^2|(u_{\tilde{h}} - I_c u_{\tilde{h}})|_{[\tilde{x}_i,\tilde{x}_{i+1}]}|_1. \end{aligned}$$

Therefore, the parameter q which determines the new partition Π_h from the trial partition $\Pi_{\tilde{h}}$ can be decided by

$$q \leq \left(\frac{\text{tol}}{|(u_{\tilde{h}} - I_c u_{\tilde{h}})|_{[\tilde{x}_i,\tilde{x}_{i+1}]|_1}}\right)^{1/2},$$

and the finite element solution $u_h(x) \in S_h^2$ will approximately satisfy

$$|(u_h - u)|_{[\tilde{x}_i, \tilde{x}_{i+1}]}|_1 \leq \text{tol}\,.$$

Again, we can assess the accuracy of the numerical solution $u_h(x)$ by checking the error indicator $|u_h - I_c u_h(x)|_1$. If $|u_h - I_c u_h(x)|_1$ is larger than the error tolerance tol, then the above grid refinement procedure can be repeated until the error tolerance is approximately satisfied. The method presented here is only a simple way to control the actual error by using the superconvergence property of the finite element solution. More sophisticated procedures may be used together with the error indicator here to generate more robust and efficient adaptive methods. Obviously, similar grid refinement procedure based on the post-processing operator I_q can be derived for the linear finite element solution.

We applied this adaptive procedure to the following boundary value problem:

$$-((2+\cos(\pi x))u'(x))' + u(x) + \int_{-1}^{1} (1+\cos(3\pi x)+y^2)u(y)\,dy = f(x),$$
$$x \in (-1,1), \qquad u'(-1) = b_0, \qquad u'(1) = b_1,$$

where $f(x)$, b_0 and b_1 were chosen such that $u(x) = \sin(\pi x)e^{3x}$ was the exact solution of this boundary value problem. We started with a uniform partition $\Pi_{\tilde{h}}$ whose step size was $\tilde{h} = 1/19$. The actual error of the corresponding finite element solution in H^1 semi-norm was $|u_{\tilde{h}} - u|_1 = 0.6799$. For the error tolerance tol $= 10^{-2}$, the adaptive procedure formed a new partition Π_h with 146 mesh points which were not uniformly distributed in $[-1,1]$. We observed, as we expected, that more mesh points were automatically added at the places where the solution changed quickly. The actual error of the finite element solution found in this new partition was $|u_h - u|_1 = 0.003318$. Since the estimated actual error was $|u_h - I_c u_h|_1 = 0.003283$ which was less than tol $= 10^{-2}$, the program terminated the adaptive procedure. The exact solution and the final numerical solution together with all the partitions used were plotted in Figure 1.

4.3. *Application to Pippard's model.* For the Pippard's nonlocal superconductivity model (1.5) and (1.6), we first notice that the kernel

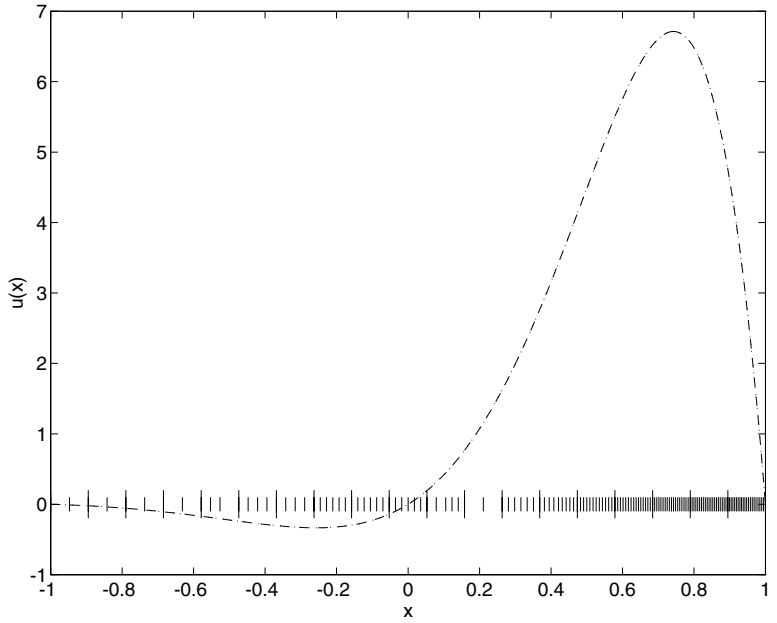

FIGURE 1. The dashed line is the true solution, the dotted line is the finite element solution generated by the adaptive procedure. The longer vertical line segments represent the trial partition, while the long and short vertical line segments together represent the refined partition for the given error tolerance.

$K(x,y) = K_1(x-y)$ is in $L^2((-l,l)\times(-l,l))$ and is symmetric. Hence the related integral operator is in compact in $L^2(-l,l)$. However, we cannot easily claim that bilinear form $a(u,v) + (\mathcal{K}u, v)$ is positive definite because both $a(u,v)$ and $(\mathcal{K}u, v)$ are only positive semi-definite. To avoid this, we can consider a modified boundary value problem as follows:

$$-A''(x) + \int_{-l}^{l} A(x)\,dx + C\int_{-l}^{l} K_1(x-y)A(y)\,dy = 0, \quad x \in (-l,l), \tag{4.34}$$

$$A'(-l) = A'(l) = H. \tag{4.35}$$

Lemma 4.1. *The modified problem* (4.34) *and* (4.35) *has a unique solution which is odd.*

Proof. Without losing generality, we assume that $H = 0$. The related bilinear form of this modified problem is

$$\tilde{a}(u,v) := \int_{-l}^{l} u'v'\,dx + \int_{-l}^{l} u\,dx \int_{-l}^{l} v\,dx + (\mathcal{K}\,u, v), \qquad \forall\, u, v \in H^1(-l,l), \tag{4.36}$$

where $\mathcal{K}_1$ is the operator defined by the kernel $K_1(x,y)$. An easy modification of the standard proof of Poincaré's inequality shows that there exists a positive constant C such that

$$\int_{-l}^{l} u'u'\,dx + \int_{-l}^{l} u\,dx \int_{-l}^{l} u\,dx \geq C||u||_1^2, \qquad \forall\, u \in H^1(-l,l).$$

The bilinear form $\tilde{a}(u,v)$ is then positive definite since $\mathcal{K}_1$ is positive semi-definite. Thus (4.34) and (4.35) has a unique solution in $H^1(-l,l)$. The operator $\mathcal{T}$ defined by

$$(\mathcal{T}\,u, v) := \tilde{a}(u,v), \qquad u, v \in H^1(-l,l), \tag{4.37}$$

is then strongly monotone and is also Lipschitz continuous. Therefore, there exists a constant $\varepsilon > 0$ such that

$$\mathcal{T}_{\varepsilon} u := u - \varepsilon \mathcal{T}\,u, \tag{4.38}$$

is contractive. By Banach fixed-point theorem, the sequence

$$u_{n+1} = \mathcal{T}_{\varepsilon} u_n, \qquad n = 0, 1, 2, \ldots,$$

converges to the unique solution of (4.34) and (4.35) starting from any $u_0 \in H^1(-l,l)$. But it is obvious that if $u_0(x)$ is an odd function, then every $u_n(x)$ in the fixed point iteration sequence above is also odd. Thus the solution of (4.34) and (4.35) which is the limit of $\{u_n\}_0^{\infty}$ must be odd. □

Since the solution of (4.34) and (4.35) is odd, we have $\int_{-l}^{l} A\,dx = 0$. Thus $A(x)$ is also a solution of (1.5) and (1.6). Since (4.34) has a V-elliptic bilinear form, the techniques of the last two sections can be

applied. Notice that the cost for solving (4.34) and (4.35) is comparable to solving (1.5) and (1.6).

Now we consider some issues in implementing the Galerkin finite element method for the nonlocal superconductivity model (4.34) and (4.35). Let $\{\phi_i(x)\}_{i=1}^N$ be a basis of S_h^r associated with the partition nodes. Then

$$N = \begin{cases} n, & \text{for } r = 1, \\ 2n-1, & \text{for } r = 2, \end{cases} \tag{4.39}$$

and $u_h = \sum_{j=1}^N u_j \phi_j(x)$ with $\vec{u}_h := (u_1, u_2, \dots, u_N)$ determined by

$$(T_h + K_h)\vec{u}_h = \vec{F}_h. \tag{4.40}$$

Here T_h is the standard stiffness matrix of the finite element spaces,

$$K_h := (K_{i,j})_{i,j=1}^N, \tag{4.41}$$

with

$$K_{i,j} := \int_{-l}^{l} \phi_i(x)\,dx \int_{-l}^{l} \phi_j(x)\,dx + \int_{-l}^{l} \phi_i(x) \left(\int_{-l}^{l} K_1(x,y)\phi_j(y)\,dy \right) dx. \tag{4.42}$$

Matrices T_h can be formed in the standard way as in the finite element method for differential equations. However, forming matrix K_h needs special attention because this is the most time consuming part.

First, since Pippard's kernel itself is defined through integration, forming the entries of K_h involves triple integrals. Reducing the computation cost in this part may greatly improve the whole computation efficiency, especially when we apply the techniques here to solve the more general nonlinear model developed recently by Brandon and Rogers [**3**]. Second, the kernel has a singularity at $x = y$ which requires special quadrature rules. These two difficulties can be treated together. Let $\widetilde{K}_j(x) := \int_{-l}^{l} \phi_j(x)\,dx + \int_{-l}^{l} K_1(x,y)\phi_j(y)\,dy$, then the j-th column of the matrix K_h is the load vector of the finite element space S_h^r with respect to the function $\widetilde{K}_j(x)$. The singularity in the kernel can be avoided by using a quadrature rule for

$$\int_{-l}^{l} \phi_i(x)\widetilde{K}_j(x)\,dx$$

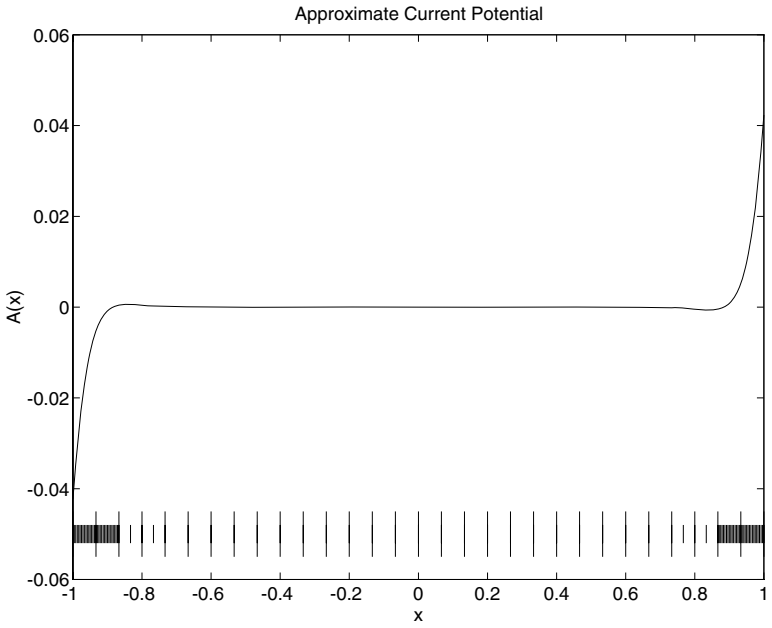

FIGURE 2. A linear finite element approximation to the current potential in Pippard's nonlocal superconductivity model. Here $\xi = 0.1$ and $C = 2500$. The longer vertical line segments represent the trial partition, while the long and short vertical line segments together represent the refined partition for the given error tolerance.

which is different from those used for evaluating $\widetilde{K}_j(x)$. For example, Gaussian quadrature rules of different degrees can be applied. To match the accuracy of $O(h^{r+1})$ of the finite element space S_h^r, the Gaussian quadrature rules which has the degree of precision greater than or equal to $2r-2$ can be used on each element. Then the most time consuming part is the evaluation of $K_1(x,y)$ at x and y which are Gaussian nodes of different degrees of the elements, because $K_1(x,y)$ is also defined by integration. For example, we generally have to evaluate $K_1(x,y)$ $2(n-1)^2$ times if Gaussian quadrature rules of degree 1 and 2 are used for the x and y integrations in $K_{i,j}$, respectively. But this may be reduced dramatically. For example, if the partition is uniform, we need to evaluate $K_1(x,y)$ only $2(n-1)$ times. Also, the discussion here indicates that the collocation methods might be more efficient alternatives for computing $A(x)$, and the related superconvergence property desires further investigation.

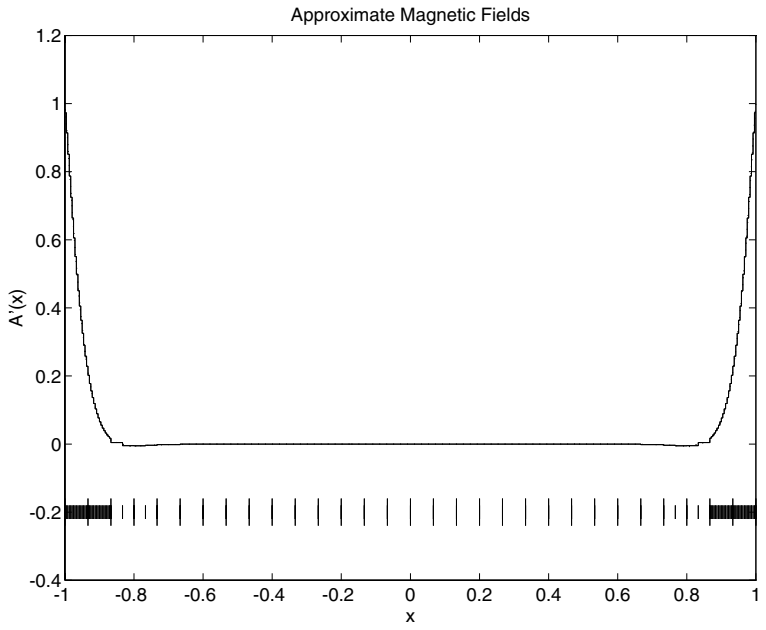

FIGURE 3. Approximations to the magnetic field in Pippard's nonlocal superconductivity model. The solid line represents the approximation based on piecewise quadratic post-processing, the dotted line represents the approximation given by the linear finite element. The longer vertical line segments represent the trial partition, while the long and short vertical line segments together represent the refined partition for the given error tolerance.

Figure 2 contains the linear finite element approximation to the current potential $A(x)$. Figure 3 contains approximations to the magnetic field $A'(x)$ generated by the derivatives of $u_h(x)$ and $I_q u_h(x)$, respectively. At the place where the exact solution seems to be flat, these two approximations to the field disagree very little. However, in the vicinity of the boundary, the approximation to the field based on the post-processing technique is obviously superior (see Figure 4). The adaptive grid refinement procedure in Section 4.1 was used in the these computations. We first used a uniform partition of 31 mesh points. The adaptive procedure automatically generated a new partition with 95 mesh points by adding more mesh points in the vicinity of the boundary than other places. To achieve a similar accuracy given here by a uniform partition, one has to solve a much larger finite element system.

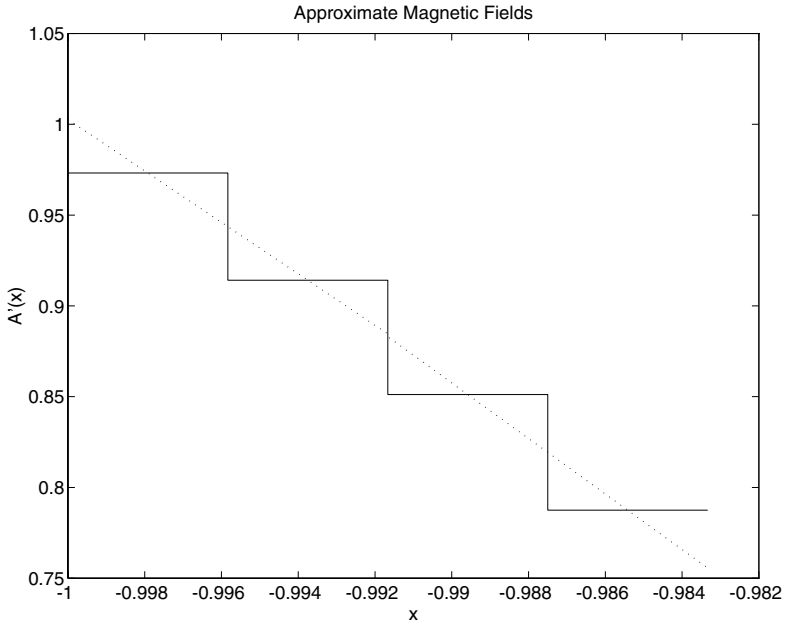

FIGURE 4. Approximations to the magnetic field in the vicinity of the left boundary. The solid line represents the approximation based on piecewise quadratic post-processing, the dotted line represents the approximation given by the linear finite element.

5. Conclusions. In this paper we have shown that a Galerkin finite element solution $u_h(x)$ to a class of Fredholm type integro-differential equations contains information about $u'(x)$ (the first derivative of the exact solution), which is not revealed by $u_h'(x)$. Several post-processing techniques were then developed to extract more information about $u'(x)$ from a finite element solution $u_h(x)$. These superconvergent post-processing techniques have the following features:

1. The first derivative of the post-processed results $I_q u_h(x)$ and $I_c u_h(x)$ are higher order approximations to $u'(x)$ than $u_h'(x)$ itself, and constructing $I_q u_h(x)$ or $I_c u_h(x)$ is less expensive than solving another finite element system.

2. The superconvergence property of $I_q u_h(x)$ and $I_c u_h(x)$ stated in Theorem 3.1 is global, i.e., $(I_q u_h(x))'$ or $(I_c u_h(x))'$ is a higher order approximation to $u'(x)$ than $u_h'(x)$ almost everywhere in the interval $[-l, l]$. This allowed us to generate more accurate approximations of

the magnetic field over the whole domain for the Pippard's nonlocal superconductivity model, and to obtain *a posteriori* H^1 semi-norm error indicators to measure the accuracy of the finite element solutions.

3. However, the post-processing procedures themselves are local because the construction of $I_q u_h(x)$ and $I_c u_h(x)$ at $x \in [-l, l]$ involves only two elements of Π_h locally. This local property was used to derive an adaptive grid refinement procedure which can control the actual error of the finite element solution within a given error tolerance by placing the mesh points appropriately.

Acknowledgments. The authors would like to thank Professor H. Brunner for his suggestion and encouragement. The authors would also like to thank the referee for the helpful comments.

REFERENCES

1. C. Baker, *The numerical treatment of integral equations*, Oxford University Press, Fair Lawn, 1977.

2. J. Bardeen, *Theory of superconductivity*, in *Encyclopedia of physics*, Vol. 15, Edited by S. Flügge, Springer-Verlag, New York, 1959.

3. D. Brandon and R. Rogers, *Nonlocal superconductivity*, Z. Angew. Math. Phys. **45** (1994), 135–152.

4. H. Brunner, *Implicit Runge-Kutta-Nyström methods for general second order Volterra integro-differential equations*, Comput. Math. Appl. **14** (1987), 549–559.

5. ———, *The approximate solution of initial-value problems for general Volterra integro-differential equations*, Computing **40** (1988), 125–137.

6. H. Brunner and P.J. van der Houwen, *The numerical solution of Volterra equations*, North-Holland, Amsterdam, 1986.

7. G.A. Chandler, *Superconvergence for second kind integral equations*, in *The application and numerical solution of integral equations* (R.S. Anderssen, F.R. de Hoog and M.A. Lukas, eds.), Sijthof & Noordhoff, Leyden, 1980, 103–117.

8. S.-S. Chow and R.D. Lazarov, *Superconvergence analysis of flux computations for non-linear problems*, Bull. Austral. Math. Soc. **40** (1989), 465–479.

9. P.G. Ciarlet, *The finite element method for elliptic problems*, North-Holland, Amsterdam, 1978.

10. L.M. Delves and J.L. Mohamed, *Computational methods for integral equations*, Cambridge University Press, Cambridge, 1985.

11. R.J. Espinosa, *Collocation solutions of integro-differential equations*, in *Constructive and computational methods for differential and integral equations* (D.L. Colton and R.P. Gilbert, eds.), Springer-Verlag, New York, 1974, 154–183.

12. L.E. Garey, K.A. Gilmore and C.J. Gladwin, *Direct methods for second order Fredholm integro-differential equations with two point boundary conditions*, in Proceedings of the Eighteenth Manitoba Conference on Numerical Mathematics and Computing, May 1989, 133–146.

13. R.. Hangelbroek, H.G. Kaper and G.K. Leaf, *Collocation methods for integro-differential equations*, SIAM J. Numer. Anal. **14** (1977), 377–390.

14. Q. Lin, *Global error expansion and superconvergence for higher order interpolation of finite elements*, J. Comput. Math., Suppl. Issue (1992), 286–289.

15. Q. Lin, H. Wang and T. Lin, *Interpolated finite element methods for second order hyperbolic equations and their global super-convergence*, System Sci. Math. **4** (1993), 331–340.

16. T. Lin and H. Wang, *A class of globally super-convergent post-processing techniques for approximating the gradients of solutions to elliptic boundary value problems*, submitted.

17. ———, *Solving a source term inverse problem of the diffusion equations by the interpolated quadratic finite element method*, Computational Methods in Water Resources IX, June 1992, Vol. I: Numerical Methods in Water Resources, 413–420.

18. ———, *Recovering the gradients of solutions to the second-order hyperbolic equations by interpolating the finite element solutions*, J. Computational Acoustics, to appear.

19. ———, *A class of error estimates based on interpolating the finite element solutions for reaction-diffusion equations* in Proceedings of the IMA Summer Program on Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations, July, 1993, to appear.

20. P. Linz, *A method for the approximate solution of linear integro-differential equations*, SIAM J. Numer. Anal. **11** (1974), 137–144.

21. A. Pippard, *An experimental and theoretical study of the relation between magnetic field and current in a superconductor*, Proc. Roy. Soc. London **216** (1953), 547–568.

22. G.R. Richter, *Superconvergence of piecewise polynomial Galerkin approximations for Fredholm integral equations of the second kind*, Numer. Math. **31** (1978), 63–70.

23. I.H. Sloan, *Superconvergence*, in *Numerical solution of integral equations* (M.A. Golberg, ed.), Plenum Press, New York, 1990.

24. G. Strang and G. Fix, *An analysis of the finite element method*, Prentice-Hall, Englewood Cliffs, 1971.

25. W. Volk, *The numerical solution of linear integro differential equations by projection methods*, J. Integral Equations Appl. **9** (1985), 171–190.

26. ———, *The iterated Galerkin method for linear integro-differential equations*, J. Comput. Appl. Math. **21** (1988), 63–74.

Department of Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061