## 1. Adaptive wavelet schemes for elliptic problems

1.1. **Basis case.** In this section we briefly show how wavelet bases can be used to treat elliptic operator equations of the form

$$\mathcal{A}u = f, \tag{1.1}$$

where we will assume $\mathcal{A}$ to be a boundedly invertible operator from some Hilbert space $\mathcal{H}$ into its normed dual $\mathcal{H}'$, i.e.,

$$\|\mathcal{A}u\|_{\mathcal{H}'} \sim \|u\|_{\mathcal{H}}, \quad u \in \mathcal{H}.$$

In applications $\mathcal{H}$ is typically a Sobolev space $W_2^t(\Omega)$ on some domain $\Omega \subset \mathbb{R}^n$. We shall mainly focus on the special case where

$$a(v, w) := \langle \mathcal{A}v, w \rangle$$

defines a *symmetric* bilinear form on $\mathcal{H}$ which is *elliptic* in the sense that

$$a(v, v) \sim \|v\|_{\mathcal{H}}^2. \tag{1.2}$$

Usually, operator equations of the form (1.1) are solved by a Galerkin scheme, i.e., one defines an increasing sequence of finite-dimensional approximation spaces $S_{\Lambda_l} := \operatorname{span}\{\eta_\mu : \mu \in \Lambda_l\}$, where $S_{\Lambda_l} \subset S_{\Lambda_{l+1}}$, and projects the problem onto these spaces, i.e.,

$$\langle \mathcal{A}u_{\Lambda_l}, v \rangle = \langle f, v \rangle \quad \text{for all} \quad v \in S_{\Lambda_l}.$$

To compute the actual Galerkin approximation, one has to solve a linear system

$$\mathbf{G}_{\Lambda_l}\mathbf{c}_{\Lambda_l} = \mathbf{f}_{\Lambda_l}, \quad \mathbf{G}_{\Lambda_l} = (\langle \mathcal{A}\eta_{\mu'}, \eta_\mu \rangle)_{\mu,\mu' \in \Lambda_l}, \quad \mathbf{f}_{\Lambda_l} = (\langle f, \eta_\mu \rangle)_{\mu \in \Lambda_l}.$$

Then the question arises how to choose the approximation spaces in a suitable way, for doing that in a somewhat clumsy fashion would yield a very inefficient scheme. One natural idea would be to use an *adaptive* scheme, i.e., an updating strategy which essentially consists of the following three steps:

| solve | – | estimate | – | refine |
|---|---|---|---|---|
| $\mathbf{G}_{\Lambda_l}\mathbf{c}_{\Lambda_l} = \mathbf{f}_{\Lambda_l}$ | | $\|u - u_{\Lambda_l}\| =?$ a posteriori error estimator | | add functions if necessary. |

Already the second step is highly nontrivial since the exact solution $u$ is unknown, so that clever a posteriori error estimators are needed. Then another challenging task is to show that the refinement strategy leads to a convergent scheme and to estimate its order of convergence, if possible. In recent years, it has been shown that both tasks can be solved if wavelets are used as basis functions for the Galerkin scheme as we shall now explain.

The first step is to transform (1.1) into a discrete problem. We assume the collection

$$\mathbf{\Psi} := \{\psi_{\boldsymbol{\lambda}} := \psi_{\lambda_1} \otimes \cdots \otimes \psi_{\lambda_d} : \boldsymbol{\lambda} \in \boldsymbol{\mathcal{J}}\},$$

and its renormalized version $\{\big(\sum_{i=1}^n 4^{t|\lambda_i|}\big)^{-1/2}\psi_{\boldsymbol{\lambda}} : \boldsymbol{\lambda} \in \boldsymbol{\mathcal{J}}\}$ to be Riesz bases for $\mathcal{V}$ and $\mathcal{H}$, respectively. Here, $\boldsymbol{\mathcal{J}}$ is an index set and $\mathcal{H} \subset \mathcal{V} = \mathcal{V}' = \mathcal{H}'$ a Gelfand triple. In applications typically it is $\mathcal{V} = L_2(\Omega)$. By using the Riesz property of the renormalized wavelet basis in $\mathcal{H}$ it is easy to see that (1.1) is equivalent to

$$\mathbf{A}\mathbf{u} = \mathbf{f} \tag{1.3}$$

where

$$\mathbf{A} := \mathbf{D}^{-1}\langle \mathcal{A}\mathbf{\Psi}, \mathbf{\Psi}\rangle^T\mathbf{D}^{-1},\ \mathbf{u} := \mathbf{D}\mathbf{c},\ u = \mathbf{c}^T\mathbf{\Psi},\ \mathbf{f} := \mathbf{D}^{-1}\langle f, \mathbf{\Psi}\rangle^T,$$

and $\mathbf{D} = \big(\big(\sum_{i=1}^n 4^{t|\lambda_i|}\big)^{-1/2}\delta_{\boldsymbol{\lambda},\boldsymbol{\lambda}'}\big)_{\boldsymbol{\lambda},\boldsymbol{\lambda}'\in\boldsymbol{\mathcal{J}}}$ is a diagonal scaling matrix.

Now (1.2) implies that

$$\|\mathbf{A}\|_{\mathcal{L}(\ell_{\mathbf{2}}(\boldsymbol{\mathcal{J}}))},\ \|\mathbf{A}^{-\mathbf{1}}\|_{\mathcal{L}(\ell_{\mathbf{2}}(\boldsymbol{\mathcal{J}}))} < \infty,$$

and the computation of the Galerkin approximation amounts to solving the system

$$\mathbf{A}_\Lambda\mathbf{u}_\Lambda = \mathbf{f}_\Lambda := \mathbf{f}|_\Lambda, \qquad \mathbf{A}_\Lambda := (\mathbf{D}^{-1}\langle \mathcal{A}\mathbf{\Psi}, \mathbf{\Psi}\rangle^T\mathbf{D}^{-1})|_\Lambda.$$

Now, ellipticity (1.2) and Riesz property yield

$$\|\mathbf{u} - \mathbf{u}_\Lambda\|_{\ell_2(\boldsymbol{\mathcal{J}})} \sim \|\mathbf{A}(\mathbf{u} - \mathbf{u}_\Lambda)\|_{\ell_2(\boldsymbol{\mathcal{J}})} \sim \|\mathbf{f} - \mathbf{A}\mathbf{u}_\Lambda\|_{\ell_2(\boldsymbol{\mathcal{J}})} \sim \|\mathbf{r}_\Lambda\|_{\ell_2(\boldsymbol{\mathcal{J}})},$$

so that the $\ell_2$-norm of the *residual* $\mathbf{r}_\Lambda$ serves as an a posteriori error estimator. Each individual coefficient $(\mathbf{r}_\Lambda)_{\boldsymbol{\lambda}}$ can be viewed as a local error indicator. Therefore a natural adaptive strategy would consist in catching the bulk of the residual, i.e., to choose the new index set $\hat{\Lambda}$ such that

$$\|\mathbf{r}_\Lambda|_{\hat{\Lambda}}\|_{\ell_2(\boldsymbol{\mathcal{J}})} \geq \zeta\|\mathbf{r}_\Lambda\|_{\ell_2(\boldsymbol{\mathcal{J}})}, \quad \text{for some} \quad \zeta \in (0,1).$$

However, such a scheme would not be implementable since the residual involves infinitely many coefficients. To transform this idea into an implementable scheme, the following three subroutines can be utilized:

- $\mathbf{RHS}[\varepsilon, \mathbf{g}] \to \mathbf{g}_\varepsilon$: determines for $\mathbf{g} \in \ell_2(\boldsymbol{\mathcal{J}})$ a finitely supported $\mathbf{g}_\varepsilon \in \ell_2(\boldsymbol{\mathcal{J}})$ such that

$$\|\mathbf{g} - \mathbf{g}_\varepsilon\|_{\ell_2(\boldsymbol{\mathcal{J}})} \leq \varepsilon.$$

- $\mathbf{APPLY}[\varepsilon, \mathbf{A}, \mathbf{v}] \to \mathbf{w}_\varepsilon$: determines for a finitely supported $\mathbf{v} \in \ell_2(\boldsymbol{\mathcal{J}})$ a finitely supported $\mathbf{w}_\varepsilon$ such that

$$\|\mathbf{A}\mathbf{v} - \mathbf{w}_\varepsilon\|_{\ell_2(\boldsymbol{\mathcal{J}})} \leq \varepsilon.$$

- **COARSE**$[\varepsilon, \mathbf{v}] \to \mathbf{v}_\varepsilon$: determines for a finitely supported $\mathbf{v} \in \ell_2(\boldsymbol{\mathcal{J}})$ a finitely supported $\mathbf{v}_\varepsilon \in \ell_2(\boldsymbol{\mathcal{J}})$ with at most $M$ significant coefficients, such that

$$\|\mathbf{v} - \mathbf{v}_\varepsilon\|_{\ell_2(\boldsymbol{\mathcal{J}})} \leq \varepsilon. \tag{1.4}$$

  Moreover, $M \lesssim M_{\min}$ holds, $M_{\min}$ being the minimal number of entries for which (1.4) is valid.

Then, employing the key idea outlined above, the resulting fundamental algorithm reads as follows:

**Algorithm 1.1. SOLVE**$[\varepsilon, \mathbf{A}, \mathbf{f}] \to \mathbf{u}_\varepsilon$

$\Lambda_0 := \emptyset$; $\mathbf{r}_{\Lambda_0} := \mathbf{f}$; $\varepsilon_0 := \|\mathbf{f}\|_{\ell_2(\boldsymbol{\mathcal{J}})}$; $j := 0$; $u_0 := 0$;
While $\varepsilon_j > \varepsilon$ do
  $\varepsilon_j := 2^{-(j+1)}\|\mathbf{f}\|_{\ell_2(\boldsymbol{\mathcal{J}})}$; $\Lambda_{j,0} := \Lambda_j$; $\mathbf{u}_{j,0} := \mathbf{u}_j$;
  For $k = 1, ..., K$ do
    Compute Galerkin approximation $\mathbf{u}_{\Lambda_{j,k-1}}$ for $\Lambda_{j,k-1}$;
    Compute $\tilde{\mathbf{r}}_{\Lambda_{j,k-1}} := \mathbf{RHS}[c_1\varepsilon_{j+1}, \mathbf{f}] - \mathbf{APPLY}[c_2\varepsilon_{j+1}, \mathbf{A}, \mathbf{u}_{\Lambda_{j,k-1}}]$;
    Compute smallest set $\Lambda_{j,k}$ s.t. $\|\tilde{\mathbf{r}}_{\Lambda_{j,k-1}}|_{\Lambda_{j,k}}\|_{\ell_2(\boldsymbol{\mathcal{J}})} \geq \frac{1}{2}\|\tilde{\mathbf{r}}_{\Lambda_{j,k-1}}\|_{\ell_2(\boldsymbol{\mathcal{J}})}$;
  od
  **COARSE**$[c_3\varepsilon_{j+1}, \mathbf{u}_{\Lambda_{j,k}}] \to (\Lambda_{j+1}, \mathbf{u}_{j+1})$;
  $j := j + 1$;
od

*Remark* 1.2. (i) We shall not discuss in detail the concrete numerical realization of the three fundamental subroutines. The subroutine **COARSE** consists of a thresholding step, whereas **RHS** essentially requires the computation of a best $N$-term approximation. The most complicated building block is **APPLY**. The subroutine can be realized and optimality of the resulting algorithm can be proved up to order $s^*$, if the stiffness matrix $\mathbf{A}$ is $s^*$-*computable*, i.e., there exists matrices $\mathbf{A}^J$, $J \in \mathbb{N}$, with

$$\|\mathbf{A} - \mathbf{A}^J\|_{\mathcal{L}(\ell_2(\boldsymbol{\mathcal{J}}))} \lesssim M_J^{-s}$$

for all $s < s^*$, where $\mathbf{A}^J$ has $\mathcal{O}(M_J)$ nontrivial entries per column whose combined computation needs at most the same order of operations.

(ii) In **ALGORITHM**1.1, $c_1, c_2$ and $c_3$ denote some suitably chosen constants whose concrete values depend on the problem at hand. Also the parameter $K$ has to be chosen in a suitable way. We refer to [1] for details.

It can be shown that **ALGORITHM** 1.1 has the following basic properties:

- **ALGORITHM** 1.1 is guaranteed to converge for a huge class of problems, i.e.,
$$\|\mathbf{u}-\mathbf{u}_\varepsilon\|_{\ell_2(\boldsymbol{\mathcal{J}})} \lesssim \varepsilon.$$
- The order of convergence of **ALGORITHM** 1.1 is *optimal* in the sense that it asymptotically realizes the convergence order of best $N$-term wavelet approximation, i.e., if the best $N$-term approximation satisfies $\mathcal{O}(N^{-s})$, then
$$\|\mathbf{u}-\mathbf{u}_\varepsilon\|_{\ell_2(\boldsymbol{\mathcal{J}})} = \mathcal{O}((\#\mathrm{supp}\mathbf{u}_\varepsilon)^{-s}).$$
- The number of arithmetic operations stays proportional to the number of unknowns, that is, the number of floating point operations needed to compute $\mathbf{u}_\varepsilon$ satisfies $\mathcal{O}(\#\mathrm{supp}\mathbf{u}_\varepsilon)$.

1.2. **Frame case.** The construction of wavelet bases particularly on more involved domains is often very difficult. In this cases the discretiztion of the PDE via a *(Hilbert) frame for* $\mathcal{V}$ sounds more promising. The latter is a collection of functions
$$\boldsymbol{\Psi} = \{\psi_{\boldsymbol{\lambda}} : \boldsymbol{\lambda} \in \boldsymbol{\mathcal{J}}\},$$
fulfilling the frame property
$$c||f||_{\mathcal{V}} \leq ||\{\langle f, \psi_{\boldsymbol{\lambda}}\rangle_{\mathcal{V}}\}_{\boldsymbol{\lambda}\in\boldsymbol{\mathcal{J}}}||_{\ell_2(\boldsymbol{\mathcal{J}})} \leq C||f||_{\mathcal{V}} \quad \text{for all } f \in \mathcal{V}, \tag{1.5}$$
with constants $c, C > 0$. A frame can be seen as a more generalized Riesz basis as it allows for redundancies. Analogue to the basis case one aims at the discrezitation
$$\mathbf{A}\mathbf{u} = \mathbf{f} \tag{1.6}$$
where
$$\mathbf{A} := \mathbf{D}^{-1}\langle \mathcal{A}\boldsymbol{\Psi}, \boldsymbol{\Psi}\rangle^T\mathbf{D}^{-1},\ \mathbf{u} := \mathbf{D}\mathbf{c},\ u = \mathbf{c}^T\boldsymbol{\Psi},\ \mathbf{f} := \mathbf{D}^{-1}\langle f, \boldsymbol{\Psi}\rangle^T.$$
The main difference to the basis scheme is that due to the redundancy of the frame $\boldsymbol{\Psi}$, the system matrix $\mathbf{A}$ has a non-trivial kernel, so that (1.6) is not uniquely solvable. Straightforward Galerkin-type approaches might hence run into stability problems.

Nonetheless, classical iterative schemes like the damped Richardson iteration
$$\mathbf{u}^{(j+1)} := \mathbf{u}^{(j)} + \omega(\mathbf{f} - \mathbf{A}\mathbf{u}^{(j)}), \quad 0 < \omega < \frac{2}{\|\mathbf{A}\|_{\mathcal{L}(\ell_2(\Lambda))}}, \quad j = 0, 1, \ldots \tag{1.7}$$
or variations thereof, like steepest descent or conjugate gradient iterations, can still be applied in a numerically stable way, and the associated expansions $u^{(j)} = \mathbf{c}^{(j)T}\boldsymbol{\Psi}$ will converge to the solution $u$ under quite general assumptions. By judiciously choosing the respective tolerances, convergence can even be preserved under perturbation of the exact iterations when, e.g., each evaluation of the infinite-dimensional right-hand side $\mathbf{f}$ and each matrix-vector product $\mathbf{A}\mathbf{v}$ are replaced by

the numerical approximations **COARSE** and **APPLY**. The resulting algorithm reads as follows:

**Algorithm 1.3. R_SOLVE**$[\varepsilon, \mathbf{A}, \mathbf{f}] \to \mathbf{u}_\varepsilon$

% Let $\theta < 1/3$ and $K \in \mathbb{N}$ be fixed such that $3\rho^K < \theta$.
% $i := 0,\ \mathbf{v}^{(0)} := 0,\ \varepsilon_0 := \|(\mathbf{A}|_{\mathrm{ran}(\mathbf{A})})^{-1}\|\|\mathbf{f}\|_{\ell_2(\Lambda)}$
While $\varepsilon_i > \varepsilon$ do
  $i := i + 1$
  $\varepsilon_i := 3\rho^K \varepsilon_{i-1}/\theta$
  $\mathbf{f}^{(i)} := \mathbf{RHS}[\mathbf{f}, \frac{\theta\varepsilon_i}{6\alpha K}]$
  $\mathbf{v}^{(i,0)} := \mathbf{v}^{(i-1)}$
  For $j = 1, \ldots, K$ do
    $\mathbf{v}^{(i,j)} := \mathbf{v}^{(i,j-1)} - \alpha(\mathbf{APPLY}[\mathbf{A}, \mathbf{v}^{(i,j-1)}, \frac{\theta\varepsilon_i}{6\alpha K}] - \mathbf{f}^{(i)})$
  od
  $\mathbf{v}^{(i)} := \mathbf{COARSE}[\mathbf{v}^{(i,K)}, (1-\theta)\varepsilon_i]$
od
$\mathbf{u}_\varepsilon := \mathbf{v}^{(i)}$

It can be shown that **ALGORITHM** 1.3 has the same basic properties as **ALGORITHM** 1.1.

## References

1. A. Cohen, W. Dahmen, and R. DeVore, *Adaptive wavelet methods for elliptic operator equations – Convergence rates*, Math. Comput. **70** (2001), no. 233, 27–75.