BPX and HB preconditioner for PCG

- Max Schröder and Julian Roth -

1 Problem description and FEM discretization

We already implemented the multigrid method for two dimensional linear finite elements on triangular grids [1] and extended this code to be able to precondition the conjugate gradient method with the BPX and the HB preconditioner. In the following we will restrict our analysis to the two dimensional case.

The domain $\Omega := (-1,1)^2 \setminus (0,1)^2$ is an L-shape, $\Gamma_D := (0,1) \times \{0\} \cup \{0\} \times (0,1)$ is the homogeneous Dirichlet boundary and $\Gamma_N := \partial\Omega \setminus \Gamma_D$ is the homogeneous Neumann boundary. The weak form is given by:

Variational form

Find $u \in V := \{u \in H^1(\Omega) \mid u = 0 \text{ on } \Gamma_D\}$ such that

$$a(u,v) = l(v) \quad \forall v \in V \tag{1}$$

where $a : V \times V \to \mathbb{R}$ is the bilinear form defined as

$$a(u,v) := \int_\Omega \nabla u \nabla v \; dx$$

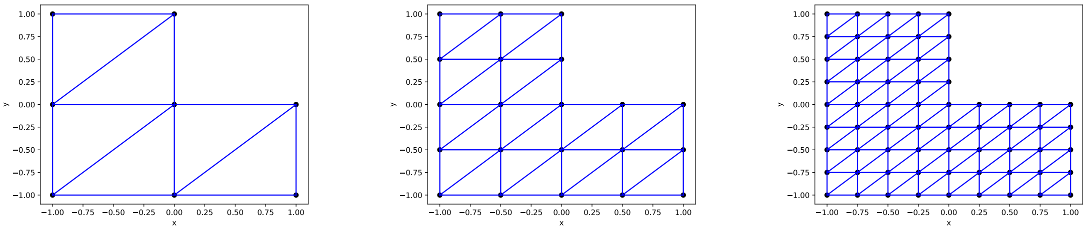
and the right hand side $l : V \to \mathbb{R}$ is a linear form defined as

$$l(v) := \int_\Omega fv \; dx.$$

Here, we use the right hand side function

$$f(x) := \begin{cases} -1 & \text{for } x \in (-1,0) \times (0,1) \\ 0 & \text{for } x \in (-1,0) \times (-1,0) \\ 1 & \text{for } x \in (0,1) \times (-1,0) \end{cases}.$$

We created a sequence of globally refined grids.



(a) Coarse Grid (b) Grid on level 1 (c) Grid on level 2

Figure 1: Level grids

These grids were refined by bisecting all edges of a triangle and drawing a new triangle out of the three new nodes.

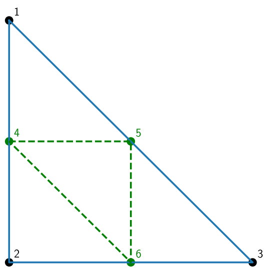


Figure 2: Refining a triangle

When refining, one needs to store the parents of the new nodes. The parents are the two end nodes of the edge that has been bisected, e.g. node 1 and node 2 are the parents of node 4. To create the prolongation matrix I_{l-1}^{l}, we follow the rules:

```
if (i.th node already exists on level l-1):
    (I_{l-1}^l)_{i,i} = 1
else:
    # get the indices of the parents of the i.th node
    (I_{l-1}^l)_{i,parent_1} = 1/2
    (I_{l-1}^l)_{i,parent_2} = 1/2
```

The problem (1) is equivalent to the solution of

$$K_h \underline{u}_h = \underline{f}_h \quad \text{where} \quad K_h = [a(\phi_i, \phi_j)]_{i,j=1}^{N},\, u_h = [\Phi]\underline{u}_h \text{ and } \underline{f}_h = [l(\phi_j)]_{j=1}^{N}.$$

This linear system must be solved, but its generally poor conditioned. Therefore we need to develop a good preconditioner C such that the condition number $\kappa(C^{-1}K_h)$ is small and we can solve the new system $C^{-1}K_h\underline{u} = C^{-1}\underline{f}$ in fewer iteration steps.

2 Algorithm

Let

$$\underline{z}^{(k)} = \underline{u}^{(k)} - \underline{u}^* \Leftrightarrow z^{(k)} = u^{(k)} - u^*$$

be the **error** of the **k-th iterate** $u^{(k)}$ and u^* denotes the **exact solution**. The **residuum** is

$$\underline{r}^{(k)} = K_h \underline{u}^{(k)} - \underline{f}_h.$$

In the following we will consider decompositions

$$\mathbb{V}_h = \mathbb{V}_0 + \mathbb{V}_1 + \cdots + \mathbb{V}_m \tag{2}$$

with $\mathbb{V}_i \subset \mathbb{V}_h$ and $\dim \mathbb{V}_i = N_i$. Then there exist matrices $V_i \in \mathbb{R}^{N \times N_i}$ such that

$$\mathbb{V}_i = \operatorname{span}[\Phi]\, V_i.$$

2.1 Additive Schwarz Method (ASM)

A map $\mathcal{P}_i : \mathbb{V}_h \to \mathbb{V}_i$ is an **orthoprojector** with respect to $a(\cdot,\cdot)$, if

$$a(\mathcal{P}_i u, v) = a(u, v) \quad \forall v \in \mathbb{V}_i, u \in \mathbb{V}_h. \tag{3}$$

Now $\mathcal{P}_i$ can be represented in the standard basis. Therefore let

$$u_i = \mathcal{P}_i u = [\Phi]\, V_i \underline{u}_i, \quad v = [\Phi]\, V_i \underline{v} \quad \text{and} \quad u = [\Phi] \underline{u}.$$

Then one obtains

$$a([\Phi]\, V_i \underline{u}_i, [\Phi]\, V_i \underline{v}) = a(\mathcal{P}_i u, v) \overset{(3)}{=} a(u, v) = a([\Phi]\underline{u}, [\Phi]\, V_i \underline{v}).$$

This is equivalent to

$$\underline{u}_i^T V_i^T K_h V_i \underline{v} = \underline{u}^T K_h V_i \underline{v} \quad \forall \underline{v} \in \mathbb{R}^{N_i}$$

Using $\underline{v} = (1, \dots, 1)$ and the symmetry of the system matrix K_h, we get

$$\begin{aligned} && \underline{u}_i^T V_i^T K_h V_i &= \underline{u}^T K_h V_i \\ &\Leftrightarrow & V_i^T K_h V_i \underline{u}_i &= V_i^T K_h \underline{u} \\ &\Leftrightarrow & \underline{u}_i &= (V_i^T K_h V_i)^{-1} V_i^T K_h \underline{u} \end{aligned}$$

Plugging this into $u_i = \mathcal{P}_i u = [\Phi]\, V_i \underline{u}_i$, yields the matrix representation P_i of $\mathcal{P}_i$ in the standard basis

$$P_i := [\mathcal{P}_i]_{[\Phi]} = V_i (V_i^T K_h V_i)^{-1} V_i^T K_h \tag{4}$$

We define the **additive Schwarz projector** $\mathcal{P}$ of the decomposition

$$\mathbb{V}_h = \mathbb{V}_0 + \mathbb{V}_1 + \cdots + \mathbb{V}_m$$

as

$$\mathcal{P} = \mathcal{P}_0 + \mathcal{P}_1 + \cdots + \mathcal{P}_m$$

and

$$w^{(k)} = \mathcal{P} z^{(k)} = \mathcal{P}\left(u^{(k)} - u^*\right)$$

is the **preconditioned residuum** of the iterative method.

Now the definition of $w^{(k)}$ gives us a first preconditioner C^{-1}, namely

$$\begin{aligned} \underline{w}^{(k)} &= P\left(\underline{u}^{(k)} - \underline{u}^*\right) \\ &= P K_h^{-1} K_h \left(\underline{u}^{(k)} - \underline{u}^*\right) \\ &= P K_h^{-1} \left(K_h \underline{u}^{(k)} - K_h \underline{u}^*\right) \\ &= P K_h^{-1} \underline{r}^{(k)} \end{aligned}$$

Therefore

$$C^{-1} = P K_h^{-1} \quad \Leftrightarrow \quad P = C^{-1} K_h$$

or using C^{-1} and K_h as operators

$$C^{-1} = \mathcal{P} K_h^{-1}$$

and

$$\begin{aligned} C^{-1} &\overset{(4)}{=} \sum_{i=0}^{m} V_i (V_i^T K_h V_i)^{-1} V_i^T K_h K_h^{-1} \\ &= \sum_{i=0}^{m} V_i (V_i^T K_h V_i)^{-1} V_i^T. \end{aligned}$$

2.2 Bramble/Pasciak/Xu (BPX) and Hierarchical Basis (HB) preconditioner

Going back to our choice of domain decomposition (2), lets consider the finite element space $\mathbb{V}_h = \mathbb{V}^L = \operatorname{span}\{\phi_i^{(L)}\}_{i=1}^{N_L}$ on the finest triangulation $\mathcal{T}_{h(L)}$ of the coarse mesh $\mathcal{T}_{h(0)}$. We can divide this space into its less refined predecessors $\mathbb{V}^l$, $l = 0, \ldots, L-1$, where they are themselves finite element spaces on the triangulation $\mathcal{T}_{h(l)}$. This domain decomposition leads to the construction of the following preconditioner.

Bramble/Pasciak/Xu (BPX) preconditioner [2]

Let $\mathbb{V}_h = \mathbb{V}^L = \text{span}\{\phi_i^{(L)}\}_{i=1}^{N_L}$, where $\mathbb{V}^l$, $l = 0, \cdots, L$ is the (linear/bilinear) finite element space of triangulation $\mathcal{T}_{h(l)}$ which is optained form the uniform/local refinement of the coarse mesh $\mathcal{T}_{h(0)}$. $\phi_i^{(l)}$ is the nodal basis function of the node i on level l. Then, we can introduce the ASM-splitting

$$\mathbb{V}^{(L)} = \sum_{l=0}^{L} \sum_{i=0}^{N_l} \mathbb{V}_i^{(l)} \tag{5}$$

with

$$\mathbb{V}_i^{(l)} = \text{span}\{\phi_i^{(l)}\}$$

This decomposition was introduced by Bramble/Pasciak/Xu 1991 [2].

Instead of always considering the full finite element space on the triangulation $\mathcal{T}_{h(l)}$, only observe the new nodes gained by the refinement of the triangulation $\mathcal{T}_{h(l-1)}$. With this domain decomposition of $\mathbb{V}_h$ we achieve a direct sum and we can derive the following preconditioner.

Hierarchical Basis (HB) preconditioner [3]

Consider the notation from the previous preconditioner and let $\mathcal{B}_l$ be the set of all nodes which are new on level l. The hierarchic decomposition

$$\mathbb{V}^{(L)} = \sum_{l=0}^{L} \sum_{i\in\mathcal{B}_l} \mathbb{V}_i^{(l)} \tag{6}$$

was introduced by Yserentant [3].

Using the orthoprojectors

$$P_i^{(l)} : \mathbb{V}^{(L)} \mapsto \mathbb{V}_i^{(l)} \text{ defined by } a(P_i u, v) = a(u, v) \quad \forall v \in \mathbb{V}_i^{(l)}, \tag{7}$$

the ASM-Projector (5) reads

$$P_{\text{BPX}} := P_L = \sum_{l=0}^{L} \sum_{i=0}^{N_l} P_i^{(l)}.$$

By definition of the orthoprojectors, we have

$$[P_i^{(l)}]_{[\Phi_L]} = \mathcal{I}_l^L \underline{e}_i (\underline{e}_i^T K_l \underline{e}_i)^{-1} \underline{e}_i^T \mathcal{I}_L^l K_L.$$

Plugging this into $C_{\mathrm{BPX}}^{-1} = P_L K_L^{-1}$ yields

$$[C_{BPX}^{-1}]_{[\Phi_L]} = [P_L K_L^{-1}]_{[\Phi_L]} = \sum_{l=0}^{L} \sum_{i=1}^{N_l} \mathcal{I}_l^L \underline{e}_i (\underline{e}_i^T K_l \underline{e}_i)^{-1} \underline{e}_i^T \mathcal{I}_L^l$$

$$= \sum_{l=0}^{L} \mathcal{I}_l^L D_l^{-1} \mathcal{I}_L^l := C_{\mathrm{BPX}}^{-1}$$

Similarly we get for the hierarchical basis (6) preconditioner

$$C_{\mathrm{HB}}^{-1} = \sum_{l=0}^{L} \sum_{i \in \mathcal{B}_l} \mathcal{I}_l^L \underline{e}_i (\underline{e}_i^T K_l \underline{e}_i)^{-1} \underline{e}_i^T \mathcal{I}_L^l = \sum_{l=0}^{L} \mathcal{I}_l^L \tilde{D}_l^{\dagger} \mathcal{I}_L^l$$

with the diagonal matrix $\tilde{D}_l = [d_{ij}^l]_{i,j=1}^{N_l}$ and their entries

$$d_{ij}^{(l)} = \begin{cases} a(\phi_i^{(l)}, \phi_i^{(l)}) & i = j, i \in \mathcal{B}_l \\ 0 & \text{else} \end{cases}$$

2.3 Implementation of the preconditioners

BPX algorithm [2]

```
def BPX(r ∈ ℝ^{N_L}):
    w^(L) = r

    # 1. RESTRICTION
    for l in [L,L-1,...,2,1]:
        w^(l-1) = I_l^{l-1} w^(l)

    # 2. DIAGONAL SCALING
    for l in [0,1,...,L-1,L]:
        w^(l) = D_l^{-1} w^(l)

    # 3. INTERPOLATION
    for l in [1,2,...,L-1,L]:
        w^(l) = w^(l) + I_{l-1}^l w^(l-1)

    return w := w^(L) ∈ ℝ^{N_L} with C_{BPX}^{-1} r = w
```

HB algorithm [3]

```
def HB(\underline{r} \in \mathbb{R}^{N_L}):
    \underline{w} := (\underline{w}_0, ..., \underline{w}_L) = \underline{r}

    # 1. RESTRICTION
    for l in [L,L-1,...,2,1]:
        \underline{w}_{0,...,l-1} = I_l^{l-1} \underline{w}_{0,...,l}

    # 2. DIAGONAL SCALING
    \underline{w} = \hat{D}^{-1} \underline{w}  # \hat{D}^{-1} := \sum_{l=0}^{L} \tilde{D}_l^{\dagger}

    # 3. INTERPOLATION
    for l in [1,2,...,L-1,L]:
        \underline{w}_{0,...,l} = \begin{pmatrix} 0 \\ \underline{w}_l \end{pmatrix} + I_{l-1}^{l} \underline{w}_{0,...,l-1}

    return \underline{w} \in \mathbb{R}^{N_L} with C_{HB}^{-1} \underline{r} = \underline{w}
```

2.4 Remarks for implementation

1. For Dirichlet boundary conditions, the corresponding entries in $\hat{D}^{-1}$ and D_l^{-1} are set to zero. This guarantees that no update is obtained for the corresponding component within the PCG-method.

2. For the diagonal scaling in the BPX- and the HB-preconditioner the diagonal of the system matrix on the coarser levels can be approximated by the diagonal of the system matrix on the finest level [4], i.e.

$$a(\phi_i^{(l)}, \phi_i^{(l)}) \approx a(\phi_i^{(L)}, \phi_i^{(L)}).$$

   Hence, we only need to assemble the system matrix on the finest level.

3. If the dimension of the coarse space $\mathbb{V}^{(0)}$ is large, then a diagonal scaling leads to a large condition number. Then a coarse grid solver for K_0 is required, e.g. we choose

$$\hat{C}_{\mathrm{BPX}}^{-1} = \sum_{l=1}^{L} \mathcal{I}_l^L D_l^{-1} \mathcal{I}_L^l + \mathcal{I}_0^L K_0^{-1} \mathcal{I}_L^0$$

2.5 Preconditioned Conjugate Gradient method

Preconditioned conjugate gradient (PCG) method [5, pp. 324/325] & [6, p. 12]

```
def PCG(
        system matrix      K ∈ ℝ^{n×n},
        right hand side    f ∈ ℝ^n,
        start vector       u^(0) ∈ ℝ^n,
        preconditioner     C^{-1} ∈ ℝ^{n×n}
    ):

  # Start conditions
  r^(0) = f − K u^(0)
  d^(0) = p^(0) = C^{-1} r^(0)

  # Iteration
  for k in [0,1,...]:
      α^(k) = (r^(k)T p^(k)) / (d^(k)T K d^(k))
      u^(k+1) = u^(k) + α^(k) d^(k)
      r^(k+1) = r^(k) − α^(k) K d^(k)
      p^(k+1) = C^{-1} r^(k+1)
      β^(k) = (r^(k+1)T p^(k+1)) / (r^(k)T p^(k))
      d^(k+1) = p^(k+1) + β^(k) d^(k)

    return solution u^(k+1), iteration number k+1
```

3 Theoretical results

Let us assume that the mesh $\mathcal{T}_l$ is obtained by uniform refinement of the coarse grid $\mathcal{T}_0$ and the solution of a problem with a symmetric, bounded, coercive bilinear form and bounded right hand side, similar to (1), satisfies

$$\|u\|_2 \leq c\|f\|_0.$$

Let u be defined by $u = \sum_{l=0}^{L} u_l$ with $u_l = P_l u - P_{l-1} u$ where P_l for $l = 0, \ldots, L$ - 1 is similar to (7) and $P_L = I$, $P_{\text{-}1} = 0$. We can further decompose u_l into $u_l = \sum_{i=1}^{N_l} u_i^{(l)}$, this gives us $u = \sum_{l=0}^{L} \sum_{i=1}^{N_l} u_i^{(l)}$. Then, there exists a constant $c > 0$ independent of h such that

$$\sum_{l=0}^{L} \sum_{i=1}^{N_l} a(u_i^{(l)}, u_i^{(l)}) \leq ca(u, u) \quad \forall u \in \mathbb{V}^{(l)}.$$

For any decomposition

$$u_h = \sum_{l=0}^{L} \sum_{i=1}^{N_l} u_i^{(l)}, \quad u_i^{(l)} \in \mathbb{V}_i^{(l)}$$

we have

$$a(u_h, u_h) \leq c \sum_{l=0}^{L} \sum_{i=1}^{N_l} a(u_i^{(l)}, u_i^{(l)}) \quad \forall u_h \in \mathbb{V}^{(L)}.$$

Combining these two theorems yields the following estimate

We have

$$c_1 C_{BPX} \leq K_h \leq c_2 C_{BPX}$$

where $c_1, c_2 > 0$ are generic constants. Moreover, $C_{BPX}^{-1} \underline{v} = \underline{w}$ requires $O(N_L)$ operations.

This gives us the final result for BPX:

$$\kappa\left(C_{BPX}^{-1} K_L\right) = O(1) \tag{8}$$

Since we restricted our analysis to the case $d = 2$, we get the final result for HB:

$$\kappa\left(C_{HB}^{-1} K_h\right) = O(1 + \log^2 h) \tag{9}$$

4 Numerical experiments

Note that our problem (1) is not H^2-regular. Hence, we can't necessarily expect the results from above. However, one can still see that we almost get the optimal results from the theory. Starting with the coarsest triangulation possible of our domain and successive uniform refinements, we see that the number of BPX iterations is almost constant, which we would expect from (8). Furthermore, the number of HB iterations shows a slight h-dependency, which fits the result (9).

		BPX		HB	
Refinements	DoFs	Iter.	Computation time	Iter.	Computation time
1	21	6	69 ms	6	104 ms
2	65	17	90 ms	22	126 ms
3	225	22	121 ms	34	157 ms
4	833	25	207 ms	46	258 ms
5	3201	27	517 ms	57	717 ms
6	12545	28	1 s 952 ms	67	5 s 376 ms
7	49665	29	7 s 510 ms	78	24 s 50 ms
8	197633	30	29 s 782 ms	87	1 min 52 s 208 ms
9	788481	30	1 min 58 s 832 ms	96	8 min 33 s 915 ms

Additionally, we mentioned that a coarse grid solver should be used when the coarse grid is very fine. To show the benefit of using a coarse grid solver, we refined the coarsest grid possible 4 times to compute our new coarse grid. Asymptotically we have a similar number of iteration steps. On the first glance it might also seem that the coarse grid solver is not useful, since some of the iteration numbers also increased. However, we are not using the grids with 8, 21, 65 and 225 DoFs, which would help reduce the number of iteration steps. Thus these results should be compared to the number of iteration steps needed for a fine coarse grid without a coarse grid solver. Without a coarse grid solver BPX and HB usually needed more than 100 steps, after which we stopped our computation. Thus we can observe that it is advisable to use a coarse grid solver when working with fine coarse grids.

		BPX		HB	
Refinements	DoFs	Iter.	Computation time	Iter.	Computation time
1	3201	22	5 s 468 ms	27	6 s 118 ms
2	12545	32	9 s 67 ms	47	12 s 65 ms
3	49665	38	16 s 119 ms	66	26 s 724 ms
4	197633	42	39 s 823 ms	86	1 min 26 s 446 ms

References

[1] J. Roth and M. Schröder. *Multigrid's documentation.* 2020. URL: https://julianroth.org/documentation/multigrid/index.html (visited on 06/22/2020).

[2] J. H. Bramble, J. E. Pasciak, and J. Xu. "Parallel Multilevel Preconditioners". In: *Mathematics of Computation* 55.191 (1990), pp. 1–22. ISSN: 00255718, 10886842. URL: http://www.jstor.org/stable/2008789.

[3] H. Yserentant. "On the multi-level splitting of finite element spaces". In: *Numerische Mathematik* 49.4 (July 1986), pp. 379–412. ISSN: 0945-3245. DOI: 10.1007/BF01389538. URL: https://doi.org/10.1007/BF01389538.

[4] Xuejun Zhang. "Multilevel Schwarz methods". In: *Numerische Mathematik* 63.1 (Dec. 1992), pp. 521–539. ISSN: 0945-3245. DOI: 10.1007/BF01385873. URL: https://doi.org/10.1007/BF01385873.

[5] T. Richter and T. Wick. *Einführung in die Numerische Mathematik - Begriffe, Konzepte und zahlreiche Anwendungsbeispiele.* Berlin Heidelberg New York: Springer-Verlag, 2017. ISBN: 978-3-662-54178-4.

[6] S. Häfner and C. Könke. "MULTIGRID PRECONDITIONED CONJUGATE GRADIENT METHOD IN THE MECHANICAL ANALYSIS OF HETEROGENEOUS SOLIDS". In: ed. by K. Gürlebeck and C. Könke. 2017. DOI: 10.25643/bauhaus-universitaet.2962. URL: http://euklid.bauing.uni-weimar.de/ikm2006/index.php_lang=de%5C&what=papers.html.

[7] S. Beuchler. *Lecture notes in 'Multigrid and domain decomposition'.* Apr. 2020.

[8] J. Roth. "Geometric Multigrid Methods for Maxwell's Equations". Bachelor's Thesis. Leibniz Universität Hannover, 2020. URL: https://julianroth.org/res/bachelorarbeit.pdf.