We consider the elliptic problem: Find $u(x)$ such that

$$\begin{aligned} -u''(x) &= 8 \qquad x \in (0,\, 1)\,, \\ u(0) &= -1\,, \\ u'(0) &= 0\,, \end{aligned} \tag{8.1}$$

and its corresponding variational formulation

$$\text{find } u \in V_0 : \qquad a(u,\, v) \;=\; \langle F,\, v\rangle \qquad \forall v \in V_0\,, \tag{8.2}$$

and the corresponding discrete problem

$$K_h\, \underline{u}_h \;=\; \underline{f}_h\,. \tag{8.3}$$

**43** Embed your MDS code from Tutorial 7 in a class `MDSPreconditioner` where you have a member function `solve` similar to the one in your existing Jacobi preconditioner class, e. g., `MDSPreconditioner::solve (const Vector& r, Vector& w)`. *Look to the pseudo code at the very end of this assignment.*

Solve system (8.3) using the preconditioned CG method with this MDS preconditioner.

**44** Solve same the system using the preconditioned conjugate gradient method with the following strategies:

a) without any preconditioner ($M = I$),
b) with the Jacobi preconditioner ($M = \operatorname{diag}(K_h)$),
c) with the MDS preconditioner on two, four, and eight levels ($L = 2,\, 4,\, 8$).

Let $h_f$ denote the mesh size of the **finest** grid in the whole computation, all the coarser grids are nested. Fill in the following table:

*Number of iterations*

| | no prec. | Jacobi | MDS ($L = 2$) | MDS ($L = 4$) | MDS ($L = 8$) |
|---|---|---|---|---|---|
| $h_f = 1/800$ | | | | | |
| $h_f = 1/1600$ | | | | | |

***optional:*** *CPU time*

| | no prec. | Jacobi | MDS ($L = 2$) | MDS ($L = 4$) | MDS ($L = 8$) |
|---|---|---|---|---|---|
| $h_f = 1/800$ | | | | | |
| $h_f = 1/1600$ | | | | | |

How do the number of iterations (and the CPU time) depend on the mesh size and the number of unknowns?

This finishes the tutorials on stationary problems. We consider now the abstract problem

$$\begin{aligned} \frac{d}{dt}(u(t),\, v)_H + a(u(t),\, v) &= \langle f(t),\, v\rangle \qquad \forall v \in V\,,\ \text{for } t \in (0,\, T) \text{ a.e.}, \\ u(0) &= u_0\,, \end{aligned} \tag{8.4}$$

where $V$ and $H$ are separable Hilbert spaces with $V \subset H$ dense and there exists $c > 0$

$$\|v\|_H \;\leq\; c\,\|v\|_V \qquad \forall v \in V\,.$$

Furthermore, let $a(\cdot,\,\cdot)$ be a bilinear form on $V$, $u_0 \in H$, and $f \in L_2((0,\,T),\,V^*)$.

**45** Show that for all $\lambda \in \mathbb{R}$: The function $u \in H^1((0,\,T);H)$ is a solution to (8.4) *if and only if* $u_\lambda \in H^1(0,\,T);H)$ is a solution to

$$\begin{aligned} \frac{d}{dt}(u_\lambda(t),\,v)_H + a_\lambda(u_\lambda(t),\,v) \;&=\; \langle f_\lambda(t),\,v\rangle \qquad \forall v \in V\,,\ \text{for } t \in (0,\,T) \text{ a.e.},\\ u_\lambda(0) \;&=\; u_0\,, \end{aligned}$$

with

$$u_\lambda(t) = e^{-\lambda t}u(t)\,, \qquad a_\lambda(w,\,v) + \lambda(w,\,v)_H\,, \qquad f_\lambda(t) = e^{-\lambda t}f(t)\,.$$

**46** In your lecture notes you find a theorem which states the existence and uniqueness of a solution to (8.4) under the condition that the bilinear form $a(\cdot,\,\cdot)$ is coercive and bounded, i. e., there exist constants $\mu_2 \geq \mu_1 > 0$ with

$$\begin{aligned} a(v,\,v) \;&\geq\; \mu_1\,\|v\|_V^2 \qquad &&\forall v \in V\,,\\ a(u,\,v) \;&\leq\; \mu_2\,\|u\|_V\,\|v\|_V \qquad &&\forall u,\,v \in V\,. \end{aligned}$$

*Using* that theorem and example **45**, show that a unique solution also exists if we replace the coercivity assumption by the following weaker condition (called *Gårding inequality*): There exists $\lambda \in \mathbb{R}$ and $\mu_1 > 0$ with

$$a(v,\,v) + \lambda\|v\|_H^2 \;\geq\; \mu_1\,\|v\|_V^2 \qquad \forall v \in V\,.$$

**47** Consider the bilinear form

$$a(w,\,v) := \int_0^1 a(x)\,w'(x)\,v'(x)\; + b(x)\,w'(x)\,v(x) + c(x)\,w(x)\,v(x)\,dx$$

in $H^1(0,\,1)$ with $a,\,b,\,c \in L_\infty(0,\,1)$ and $a_0 = \inf_{x\in(0,1)} a(x) > 0$. Show the Gårding inequality: There exist constants $\lambda \in \mathbb{R}$ and $\mu_1 > 0$ with

$$a(v,\,v) + \lambda\|v\|_{L_2(0,\,1)}^2 \;\geq\; \mu_1\,\|v\|_{H^1(0,\,1)}^2 \qquad \forall v \in H^1(0,\,1)\,.$$

**48** Assume that $a(\cdot,\,\cdot)$ is bounded an coercive with coercivity constant $\mu_1 > 0$. Show that

$$\|\theta_h(t)\|_H \;\leq\; \|\theta_h(0)\|_H\,e^{-\mu_1 t/c^2} + \int_0^t \|\rho_h'(s)\|_H e^{-\mu_1\,(t-s)/c^2}\,ds\,,$$

where $\theta_h$ and $\rho_h$ are defined according to the lecture notes.
*Hint:* Modify the proof in the lecture notes, use the coercivity, and investigate the term

$$\frac{d}{dt}\Big[\|\theta_h(t)\|_H\,e^{\mu_1\,t/c^2}\Big]\,.$$

**Hints for MDS:** In order to uncouple the problem and the solver, use (not necessarily exactly) the following structure. The comments indicate that you have to complete, fill in, or rewrite according to your current implementation.

```
typedef valarray<double> Vector;

class MDSPreconditioner
{
 public:

  MDSPreconditioner (int lvls)
    : levels(lvls), jacobi(lvls) // (reserve lvls entries in jacobi)
  { ; }

  void setJacobi (int l, const Vector& diag)
  { /* set jacobi[l] to diag; */ }

  void solve (const Vector& r, Vector& w)
  { /* call the routine MDS (...) on finest level */ }

 private:
  int levels;
  vector<JacobiPreconditioner> jacobi;

  void MDS (int l, const Vector& r, Vector& w);
    // (this is essentially the routine from Tutorial 7)
  {
    // jacobi[l].solve (r, w);
    if (l != 0)
    {
       Restrict (r, r_coarse);
       MDS (l-1, r_coarse, w_coarse);
       Prolongate (w_coarse, w_fine);
       w += w_fine;
    }
  }
}; // class MDSPreconditioner

void main ()
{
  ...
  MDSPreconditioner M(levels);
  // create coarsest mesh

  for (l=0; l<levels; l++)
  {
    // assemble stiffness matrix K, load vector f, implement B.C.
    // M.setJacobi (l, ...)
    // if (l != levels-1) refineUniform (...)
  }

  CG (K ... f ... M ...);
  ...
} // main
```