

MAP562 Optimal design of structures

by Grégoire Allaire, Thomas Wick

Ecole Polytechnique

Academic year 2016-2017

Gradient descent and Newton's method for minimization in a PDE / function space context

We recapitulate the basic steps to implement gradient descent (with a fixed step size) and Newton's method in order to minimize a nonlinear functional (here the p -Laplace problem) in a function space setting. This setting is a so-called unconstrained minimization problem.

The problem statement reads:

Problem 1. Let $\Omega \in \mathbb{R}^2$ be a domain and Γ_D and Γ_N Dirichlet and Neumann boundary parts, respectively. The function space we work with is $V := H^1(\Omega)$. Then the minimization problem reads:

$$\min_{u \in V} J(u),$$

where

$$J(u) = \int_{\Omega} \left(\frac{|\nabla u|^2}{2} + \frac{|\nabla u|^p}{p} \right) dx - \int_{\Omega} f \cdot u dx - \int_{\Gamma_N} g \cdot u ds,$$

and $f = 1$ and $g = -1$. Moreover, we choose $p = 4$. In this context, we must ensure that $p > 2$ (the more general problem $p < 2$ is possible, but difficult from a theoretical and numerical point of view).

In the following, we formulate the two algorithms (using the same notation as in the lecture notes, chapter 3). This notation is also (as far as possible) used in the FreeFem codes (updated on the webpage).

1 Gradient descent with fixed step

In this section, we discuss a gradient descent method with fixed step μ in order to minimize the above functional $J(u)$ in a function space setting.

Algorithm 1 (Gradient descent with fixed step size, chapter 3, page 34). Choose an initial guess $u^0 \in V$, e.g., $u^0 = 0$. For $n = 0, 1, 2, \dots$ compute

$$u^{n+1} = u^n - \mu J'(u^n).$$

We stop the iteration when

$$\sqrt{\int_{\Omega} (J'(u^n))^2} < TOL$$

where, e.g., $TOL = 1e-4$. In a function space context the crucial aspect is the evaluation of $J'(u^n)$.

A standard procedure to realize the above algorithm is to compute the projection $\delta u \in V$. In more detail we replace $J'(u^n)$ by δu such that

$$u^{n+1} = u^n - \mu \delta u$$

where $\delta u \in V$ is a finite element solution of the projection problem:

Problem 2 (Projection problem). For a given u^n , find $\delta u \in V$ such that

$$\int_{\Omega} \nabla \delta u \cdot \nabla \phi = \langle J'(u^n), \phi \rangle \quad \forall \phi \in V,$$

where we use a scalar product that is related to the V function space (here the H^1 space). Specifically the right hand side consists of known terms: u^n, f, g .

We could also have used

$$\int_{\Omega} \delta u \phi + \alpha \nabla \delta u \cdot \nabla \phi = \langle J'(u^n), \phi \rangle,$$

with some parameter $\alpha > 0$.

Specifically, the right hand side term of the projection problem is nothing else than calculating the directional derivative of $J(u)$ into direction $\phi \in V$ (neglecting the index n for the moment):

$$\begin{aligned} \langle J'(u), \phi \rangle &= J'(u)(\phi) = \int_{\Omega} \nabla u \cdot \nabla \phi + \int_{\Omega} |\nabla u|^{p-2} \cdot \nabla u \cdot \nabla \phi - \int_{\Omega} f \cdot \phi - \int_{\Gamma_N} g \cdot \phi \, ds, \\ &= \int_{\Omega} (1 + |\nabla u|^{p-2}) \cdot \nabla u \cdot \nabla \phi - \int_{\Omega} f \cdot \phi - \int_{\Gamma_N} g \cdot \phi \, ds. \end{aligned}$$

Remark 1. Please verify yourself by applying the technique presented in the lecture notes and also my computation on the black board.

Remark 2. The first-order derivative $J'(u)(\phi)$ is nothing else than a variational form in which u is the sought solution and ϕ the test function.

Next, we write the projection problem as a root finding problem:

Problem 3 (Projection as root-finding problem). For a given u (in the gradient-based context, u will be the previous iteration u^n), find $\delta u \in V$ such that:

$$- \int_{\Omega} \nabla \delta u \nabla \phi + J'(u)(\phi) = 0 \quad \forall \phi \in V.$$

To solve this (linear) problem, we introduce finite dimensional subspaces V_h and use the finite element method and then use for example an LU decomposition or an iterative solver (CG, GMRES) for linear problems. The obtained solution δu is then our current iterate used in the gradient descent method.

Problem 4 (Problem 3 realized in FreeFem). In FreeFem the previous setting is defined as

```

problem Jlin(du,phi) = -int2d(Th)(grad(du)'*grad(phi))
                    +int2d(Th)((1+sqrt(grad(u)'*grad(u))^(p-2))*grad(u)'*grad(phi))
                    -int2d(Th)(f*phi)
                    -int1d(Th,GammaN)(g*phi)
                    +on(GammaD,du=0);

```

To solve this problem we simply write

```
Jlin;
```

and can use the solution `du` to update the iteration:

```
u = u - stepsize * du;
```

Remark 3. In the above algorithm we have to solve at each iteration n such a linear problem. In FreeFem we use therefore the keyword `problem` as already done for linear problems in the first exercise.

Remark 4 (Relationship to ‘standard’ linear problems, Poisson). In fact Problem 3 has exactly the same structure as we know from solving the linear Poisson problem. Also here we defined a root-finding problem: Find $\delta u \in V$ such that

$$a(\delta u, \phi) - l(\phi) = 0 \quad \forall \phi \in V,$$

where

$$\begin{aligned} a(\delta u, \phi) &= (\nabla \delta u, \nabla \phi), \\ l(\phi) &= (f, \phi) + (g, \phi)_{\Gamma_N}, \end{aligned}$$

where (u, v) denotes the L^2 scalar product $\int_{\Omega} u \cdot v \, dx$.

Remark 5. In general we **hope** that the sequence $(u^n)_{n \in \mathbb{N}}$ converges towards the optimal solution. Under several assumptions this convergence can be justified also from the theoretical point of view. In most cases (and this is standard in practice!) a theoretical convergence cannot be proven because it is too difficult, however the algorithm is nevertheless used in practice and yields satisfying results.

2 Newton's method

In this section we turn our attention to Newton's method. The crucial difference is that Newton's method converges (locally) much faster than gradient descent, however we need

- to calculate the second derivative of $J(u)$;
- to solve a linear equation system, which is in most cases an expensive operation.

Another difficulty is that Newton's method is quite sensitive if the initial guess u^0 is too bad. Common strategies to globalize the method is to introduce a line search parameter or to work with so-called trust region methods (see the remark below).

Algorithm 2 (Newton's method, chapter 3, page 37). Choose an initial guess $u^0 \in V$, e.g., $u^0 = 0$. For $n = 0, 1, 2, \dots$:

$$u^{n+1} = u^n + \delta u$$

where $\delta u \in V$ is obtained from solving the linear system:

$$\text{Find } \delta u \in V : \quad J''(u^n)(\delta u, \phi) = -J'(u^n)(\phi) \quad \forall \phi \in V$$

using e.g., a finite element method in a finite dimensional subspace V_h . We stop the iteration when

$$\sqrt{\int_{\Omega} (\delta u)^2} < TOL$$

where, e.g., $TOL = 1e - 4$.

To compute the second-order derivative we start with $J'(u)(\phi)$,

$$\begin{aligned} J'(u)(\phi) &= \int_{\Omega} \nabla u \cdot \nabla \phi + \int_{\Omega} |\nabla u|^{p-2} \cdot \nabla u \cdot \nabla \phi - \int_{\Omega} f \cdot \phi - \int_{\Gamma_N} g \cdot \phi \, ds, \\ &= \int_{\Omega} (1 + |\nabla u|^{p-2}) \cdot \nabla u \cdot \nabla \phi - \int_{\Omega} f \cdot \phi - \int_{\Gamma_N} g \cdot \phi \, ds \end{aligned}$$

and obtain (check yourself !!) by differentiating u now into direction δu using multiple times the chaine rule and also the product rule:

$$\begin{aligned} J''(u)(\delta u, \phi) &= \int_{\Omega} \nabla \delta u \cdot \nabla \phi + \int_{\Omega} (p-2) |\nabla u|^{p-4} \nabla u \cdot \nabla \delta u \nabla u \cdot \nabla \phi + \int_{\Omega} |\nabla u|^{p-2} \cdot \nabla u \cdot \nabla \phi \\ &= \int_{\Omega} (1 + |\nabla u|^{p-2}) \cdot \nabla \delta u \cdot \nabla \phi + \int_{\Omega} (p-2) |\nabla u|^{p-4} \nabla u \cdot \nabla \delta u \nabla u \cdot \nabla \phi. \end{aligned}$$

Having now the Hessian matrix $J''(u)(\delta u, \phi)$ and the first order derivative $J'(u)(\phi)$ we define again a root-finding problem to solve the linear equation system:

Problem 5. For a given u^n (previous solution), find $\delta u \in V$ such that

$$J''(u^n)(\delta u, \phi) + J'(u^n)(\phi) = 0 \quad \forall \phi \in V.$$

In FreeFem we use again the `problem` functionality to solve this problem. Please verify yourself each term.

Problem 6 (Problem 5 realized in FreeFem). In FreeFem the previous setting is defined as

```
problem Jlin(du,phi)=int2d(Th)((1+sqrt(grad(u)'*grad(u))^(p-2))*(grad(du)'*grad(phi))
+(p-2)*sqrt(grad(u)'*grad(u))^(p-4)*(grad(u)'*grad(du))*(grad(u)'*grad(phi)))
+int2d(Th)((1+sqrt(grad(u)'*grad(u))^(p-2))*(grad(u)'*grad(phi)))
-int2d(Th)(f*phi)
-int1d(Th,GammaN)(g*phi)
+on(GammaD,du=0);
```

To solve this problem we simply write

```
Jlin;
```

and can use the solution `du` to update the iteration:

```
u = u + du;
```

Remark 6 (Line search). A simple method to increase the convergence radius of Newton's method (mainly for nonconvex problems) is to introduce a line search parameter $\mu \in [0, 1]$ such that:

$$u^{n+1} = u^n + \mu \delta u$$

For $\mu = 1$ we make a full Newton step and choosing $\mu < 1$ yields a damped Newton method which has an increased convergence radius (but which again does not work in all possible situations well), and also has a reduced order of convergence (not quadratically anymore). In FreeFem this would read:

```
u = u + stepsize * du;
```