

Solving variational problems with uniform gradient bounds by p -Laplacian approximation

Implementation using the deal.II finite elements library

Felipe Alvarez¹, Salvador Flores^{2*}

¹*DIM, Universidad de Chile, Beauchef 851, Santiago, Chile*

²*Centro de Modelamiento Matemático, Universidad de Chile, Beauchef 851, Santiago, Chile*

Abstract

We describe the finite element implementation of the method of Alvarez and Flores (2015) for computing solutions and Lagrange multipliers associated to the uniform constraint in the family of variational problems

$$\inf \left\{ \int_{\Omega} \left(\frac{1}{2} W(|\nabla v|^2) - f(v) \right) \mid \|\nabla v\|_{\infty, \Omega} \leq 1, v \in g + W_0^{1,s}(\Omega) \right\}$$

by solving the p -Laplacian penalized approximation:

$$\begin{aligned} -\operatorname{div}((W'(|\nabla u_p|^2) + |\nabla u_p|^{p-2})\nabla u_p) &= f'(u_p) & \text{in } \Omega \\ u_p &= g & \partial\Omega. \end{aligned}$$

With the default parameters, the software gives an approximate solution and the corresponding Lagrange multiplier for the problem of the elastoplastic torsion of a cylindrical bar of section Ω :

$$\min_{v \in K_0} \int_{\Omega} \left(\frac{1}{2} |\nabla v(x)|^2 - 4v(x) \right) dx$$

for $K_0 = \{v \in H_0^1(\Omega) \mid |\nabla v(x)| \leq 1 \text{ a.e } x \in \Omega\}$ and $\Omega = \text{unit disk}$.

*Tel.: +56 22 97 80 611; fax: +56 26 88 97 05.

E-mail address: sflores@dim.uchile.cl

1. Introduction

We discuss the computational aspects of solving the class of scalar Dirichlet problems by the finite elements method

$$\inf\{J(v) \mid \|\nabla v\|_{\infty,\Omega} \leq 1, v \in g + W_0^{1,s}(\Omega; \mathbb{R})\}, \quad (1)$$

where $\|\nabla v\|_{\infty,\Omega} = \text{ess-sup}\{|\nabla v(x)| : x \in \Omega\}$, $g \in W^{1,\infty}(\Omega; \mathbb{R}^m) \cap C(\bar{\Omega}; \mathbb{R})$ and

$$J(v) = \int_{\Omega} \left(\frac{1}{2} W(|\nabla v|^2) - f(v) \right) dx \quad (2)$$

for some functions W and f .

The model case of (1) is the problem of the elastoplastic torsion of a cylindrical bar of section Ω :

$$\min_{v \in K_0} \frac{1}{2} \int_{\Omega} (|\nabla v(x)|^2 - h(x)v(x)) dx \quad (3)$$

for $K_0 = \{v \in H_0^1(\Omega) \mid |\nabla v(x)| \leq 1 \text{ a.e } x \in \Omega\}$.

Let us consider the penalized problem

$$\min \left\{ \frac{1}{2} \int_{\Omega} W(|\nabla v|^2) + \frac{1}{p} \int_{\Omega} |\nabla v|^p - \int_{\Omega} f(v) : v \in g + H_0^1(\Omega) \right\}. \quad (4)$$

By the convexity assumptions on the functions W and ϕ , that problem has a unique solution u_p which is a weak solution of the Euler-Lagrange equation:

$$- \text{div}((W'(|\nabla u_p|^2) + |\nabla u_p|^{p-2})\nabla u_p) = f u_p. \quad (5)$$

Alvarez and Flores (2015) showed that the solutions u_p of (4) converge as $p \rightarrow \infty$ to a solution of the constrained problem (1), and moreover $|\nabla u_p|^{p-2}$ converges to the Lagrange Multiplier associated to the pointwise constraint $|\nabla u| \leq 1$. We are thus led to solve the quasilinear elliptic equations (5) for large values of p . However, for large p the convergence and stability of such an iterative procedure is a delicate issue. In the following we describe some techniques allowing to compute good approximations for values of p in the thousands and/or under mild departures from the hypothesis of Alvarez and Flores (2015).

2. Numerical Implementation

Let us consider the variational formulation for the p -Laplacian penalized problem (4). Alvarez and Flores (2015) proposed to solve it by a descent algorithm using the full Newton direction w^N , computed as

$$\begin{aligned} \int_{\Omega} (G(|\nabla u_{p,n}|^2) + (p-1)|\nabla u_{p,n}|^{p-2})\nabla w_n^N \nabla v = \\ - \int_{\Omega} (W'(|\nabla u_{p,n}|^2) + |\nabla u_{p,n}|^{p-2})\nabla u_{p,n} \nabla v + \int_{\Omega} f v \quad \forall v \in \mathcal{V}^h, \end{aligned} \quad (6)$$

where \mathcal{V}^h stands for the finite element space under consideration and $G(s) := W'(s) + 2sW''(s)$ is supposed to be a positive function for any $s > 0$. In our implementation, the functions

```
double W (double Du2) const;
```

```
double Wp (double Du2) const;
```

```
double G (double Du2) const;
```

should provide the values of the functions $W(|\nabla u|^2)$, $W'(|\nabla u|^2)$ and $G(|\nabla u|^2)$ respectively. The simplest implementation of this Newton algorithm is described in Algorithm 1.

Algorithm 1: Inner iterations on a given mesh and for a fixed p .

Given $p > 2$ and an initial point $u_{p,0} \in \mathcal{V}^h$, choose c_1, ε and `max_inner`
Set $n := 0$ and iterate:

1. Compute the multiplier $\lambda_{p,n} = |\nabla u_{p,n}|^{p-2}$.
2. Find the descent direction w_n^N by solving (6).
3. Perform a line-search with sufficient decrease condition, *i.e.*, find $\alpha_n > 0$ satisfying

$$J_p(u_{p,n} + \alpha_n w_n^N) \leq J_p(u_{p,n}) + c_1 \alpha_n J'_p(u_{p,n})[w_n^N] \quad (7)$$

4. Set $u_{p,n+1} = u_{p,n} + \alpha_n w_n^N$.
 5. If $\|J'_p(u_{p,n+1})\| \leq \varepsilon$ or $n \geq \text{max_inner}$ stop.
Otherwise update $n = n + 1$ and go to step 1.
-

It is customary to define the function $\phi(\alpha) = J_p(u_{p,n} + \alpha_n w_n^N)$. In this way, (7) can be simplified to

$$\phi(\alpha) \leq \phi(0) + (c_1 \phi'(0))\alpha \quad (8)$$

A step-length satisfying the sufficient decrease condition (8) is found by a home made line search procedure based on quadratic interpolation. The detailed implementation is presented in Algorithm 2; it follows the general guidelines from Nocedal and Wright (2006). For large values of p the objective function can take on infinity values; in these cases the line-search algorithm performs Armijo's steps until a finite value is reached.

Algorithm 1 can be greatly improved by using adaptive mesh refinements. Numerical results by Alvarez and Flores (2015) show that they are indeed necessary in order to obtain good accuracy.

It can be sensitive with respect to the initial point too, particularly for large p and/or when a non-homogeneous boundary condition is given. A way around this difficulty is to adopt a path-following strategy, which consists in running initially the algorithm on a coarse mesh and for a low value of p , and then increasing p and adaptively refining the mesh until reaching a target p . Some extra mesh refinements can be eventually performed once the target p has been achieved. Our implementation is outlined in Algorithm 3. The adaptive refinements proceed by refining a percentage of the cells with the highest *a posteriori* gradient approximation error according to the estimator by Kelly et al. (1983) (see also Ainsworth and Oden, 1997) provided by the `deal.II` library. The descent directions are computed solving the systems by the conjugate gradient algorithm with a SSOR preconditioner.

2.1 Test problems

We provide three different pre-defined domains for solving the problem

$$\min \left\{ J(u) := \frac{1}{2} \int |\nabla u|^2 - \int hu \mid \begin{array}{l} |\nabla u| \leq 1 \text{ a.e in } \Omega \\ u = g \text{ on } \partial\Omega \end{array} \right\}, \quad (9)$$

Algorithm 2: Path-following algorithm for solving (4)

```

Data: Parameters: max_LS_it;initial steplength;
        Global variables: old- $\alpha, \phi(0), \phi'(0), \text{old-}\phi'(0)$ ;
/* Initializations                                     */
initialize mesh;
refine mesh No of initial refinements times;
initialize finite element space;
cycle=0;
p=init_p;
 $u_p = u_0$ ;
/* First loop                                         */
/* Prepare initial condition increasing p and refining mesh */
while  $p < \text{actual}_p$  do
    run Algorithm 1 with  $u_p$  as starting point ;
    refine mesh adaptively;
    transfer  $u_p$  to new mesh;
    ++cycle;
     $p += \Delta p$ ;
/* Second loop                                       */
/* Solve problem for target p refining mesh only    */
p=actual_p;
while  $\text{cycle} < \text{adapt}_\text{ref}$  do
    run Algorithm 1 ;
    refine mesh adaptively;
    transfer  $u_p$  to new mesh;
    ++cycle;

```

Algorithm 3: Line-search used in step 3. of Algorithm 1

Data: Parameters: max_LS_it;initial steplength;
Global variables: old- α , $\phi(0)$, $\phi'(0)$, old- $\phi'(0)$;
Procedure arguments: inner iteration;

Result: step-size α

```

/*
/* initializations
/*
done=false;
it=0;
if inner iteration=0 then
|  $\alpha$ =initial steplength
else /* there exists an steplength from previous inner iter.
*/
|  $\alpha = \min \left( 1.45 \text{ old-}\alpha \frac{\text{old-}\phi'(0)}{\phi'(0)}, 1.0 \right);$ 
/*
/* main loop
/*
while (!done) and (it < max. LS. it.) do
| /* new try obtained by quadratic interpolation
|  $\alpha^+ = -\frac{\phi'(0)\alpha^2}{2(\phi(\alpha) - \phi(0) - \phi'(0)\alpha)}$ ;
| if  $\alpha^+ < 10^{-3}\alpha$  or  $|\alpha^+ - \alpha|/\alpha < 10^{-8}$  then
| |  $\alpha^+ = \alpha/2$ ;
| else if  $\phi(\alpha) - \phi(0) > 1e3|\phi(0)|$  then
| |  $\alpha^+ = \alpha/10$ ;
|  $\alpha = \alpha^+$ ;
|  $\phi(\alpha)$ =evaluate  $J_p(u_{p,n} + \alpha w_n)$ ;
| done=checkWolfe( $\alpha$ ,  $\phi(\alpha)$ ); /* check condition (8) */
| it=it+1;
Return  $\alpha$ ;
```

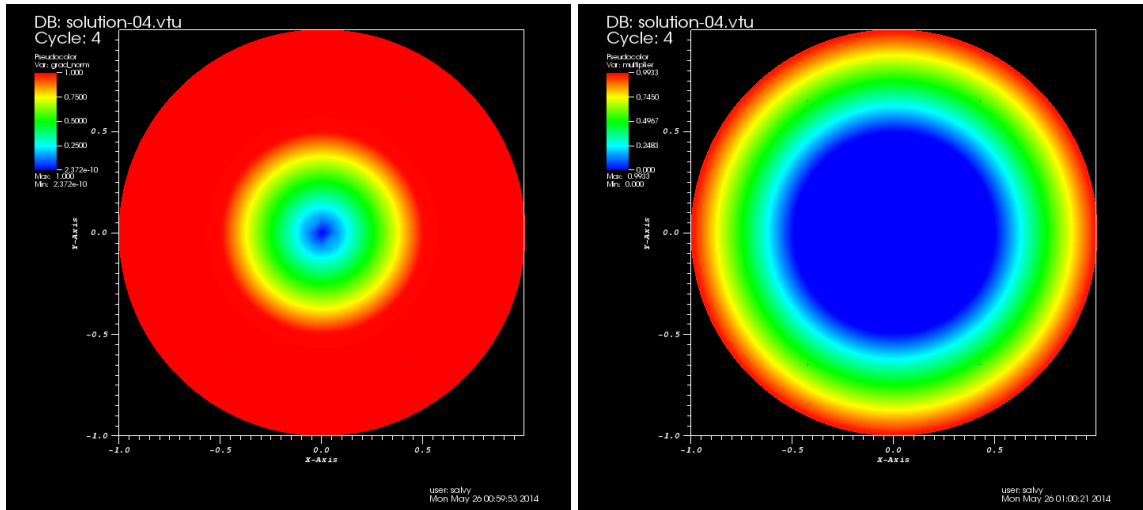


Figure 1: Plot of the norm of the gradient $|\nabla u_p|$ and the multiplier $\lambda_p = |\nabla u_p|^{p-2}$ on a circle.

The unit circle, the unit rectangle and the domain depicted in Figure 3 (Glowinski et al., 1981, Figure 3.4). The domains can be modified or added in the function `ElastoplasticTorsion<dim>::init_mesh()`.

In Figures 2 and 3 we show the solutions of Problem 9 in a rectangle and a domain with an interior corner, respectively.

Denote by D the unit disk of \mathbb{R}^2 , i.e $D = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 < 1\}$. When $\Omega = D$ and h is constant, (9) has an explicit solution. If $h \equiv 4$ and $g \equiv 0$ the solution is given by (Glowinski et al., 1981):

$$u(x) = \begin{cases} 1 - r & \text{if } 1/2 \leq r \leq 1 \\ -r^2 + 3/4 & \text{if } 0 \leq r \leq 1/2 \end{cases} \quad (10)$$

where $r = \sqrt{x^2 + y^2}$. Since Ω is convex, in this case the multiplier λ is continuous (Brézis, 1972). In fact we obtained its explicit expression, which is given by

$$\lambda(x) = \begin{cases} 2r - 1 & \text{if } 1/2 \leq r \leq 1 \\ 0 & \text{if } 0 \leq r \leq 1/2. \end{cases} \quad (11)$$

The norm of the gradient of the computed solution and the multiplier are plot in Figure 1. For this problem it is possible to compute error tables. This is done only if the parameter `known_solution` is set `True` (cf. Table 1).

3. Connections with overdetermined boundary value problems

A solution u_p to the boundary-value problem (4) should satisfy the relation:

$$\partial_\tau u_p = \partial_\tau g \quad \text{on } \partial\Omega$$

where ∂_τ denotes tangent derivative. Altogether with $\|\nabla u_p\|_\infty \leq 1$, this implies

$$|\partial_\nu u_p|^2 \leq 1 - \|\partial_\tau g\|^2.$$

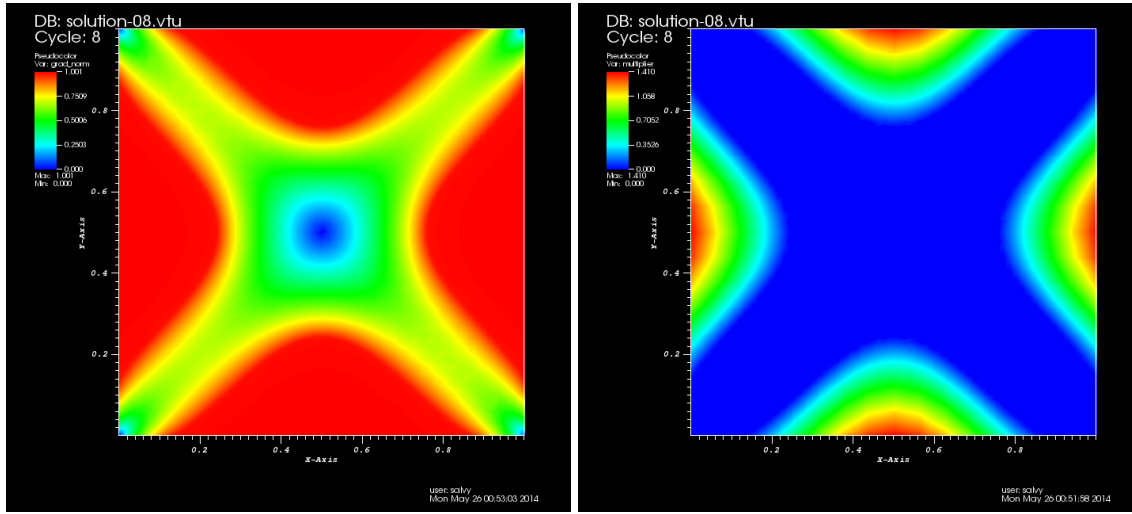


Figure 2: Plot of the norm of the gradient $|\nabla u_p|$ and the multiplier $\lambda_p = |\nabla u_p|^{p-2}$ on a rectangle.

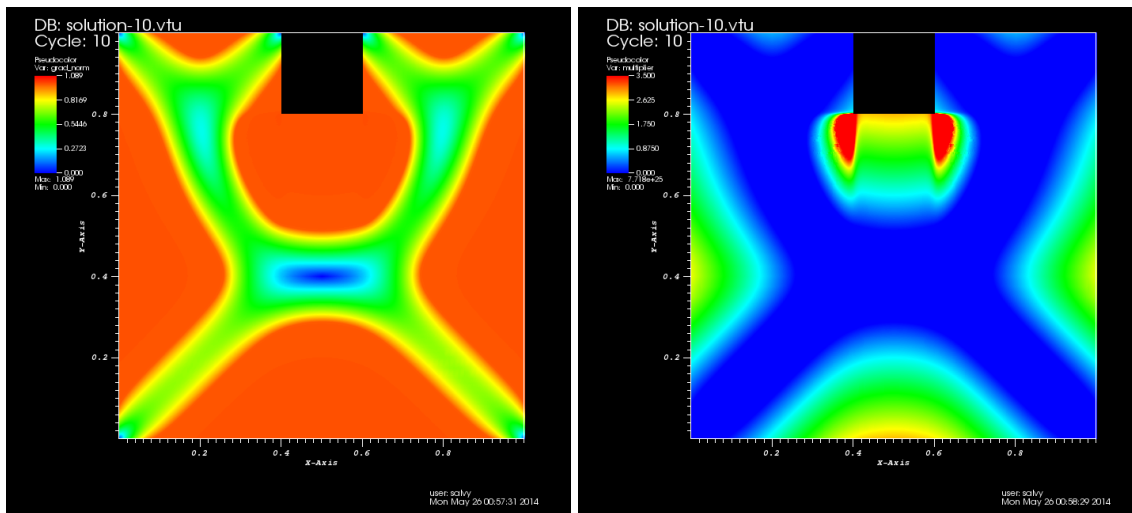


Figure 3: Plot of the norm of the gradient $|\nabla u_p|$ and the multiplier $\lambda_p = |\nabla u_p|^{p-2}$ on a domain with an interior corner. The scale in the plot of the multiplier is truncated.

Therefore, if g is such that $\|\partial_\tau g\| = 1$ in a neighborhood of $\Gamma \subseteq \partial\Omega$, then u_p solves indeed a partially overdetermined problem

$$u_p = g \text{ on } \partial\Omega, \quad \partial_\nu u_p = 0 \text{ on } \Gamma.$$

For fully overdetermined problems such as

$$u = 0 \text{ and } |\nabla u| = c \text{ on } \partial\Omega$$

it is known (Farina and Kawohl, 2008) that if the problem has a solution then Ω must be a ball.

To the best of our knowledge, partially overdetermined problems have not been considered yet in the literature. We invite the curious reader to try the following boundary condition,

$$g(\theta) = \begin{cases} \theta^2 & \text{if } \theta \leq 0.5 \\ \theta - 0.25 & \text{if } 0.5 < \theta \leq \pi - 0.5 \\ (\pi - 0.75 - (\theta - (\pi - 0.5)))^2 & \text{if } \pi - 0.5 < \theta \leq \pi + 0.5 \\ (2\pi - \theta) - 0.25 & \text{if } \pi + 0.5 < \theta \leq 2\pi - 0.5 \\ (\theta - 2\pi)^2 & \text{otherwise} \end{cases} \quad (12)$$

which has a large portion of the boundary where $\|\nabla_\tau u\| = 1$. This function is included (but commented) in the code.

4. Parameters

Most of the parameters involved in Algorithms 1, 2 and 3 are defined in the parameters file `EPT.prm`, and can be easily changed without need to recompile the code. The full list of the parameters is in Table 1.

References

- Ainsworth, M., Oden, J., 1997. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering* 142 (1–2), 1 – 88.
- Alvarez, F., Flores, S., 2015. Existence and approximation for variational problems under uniform constraints on the gradient by power penalty. *SIAM Journal on Mathematical Analysis* 47 (5), 3466–3487.
- Bangerth, W., Hartmann, R., Kanschat, G., 2007. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.* 33 (4), 24/1–24/27.
- Barrett, J. W., Liu, W. B., 1993. Finite element approximation of the p -Laplacian. *Mathematics of Computation* 61 (204), 523–537.
- Bermejo, R., Infante, J.-A., 2000. Multigrid algorithm for the p -Laplacian. *SIAM Journal on Scientific Computing* 21 (5), 1774–1789.
- Brézis, H., 1972. Multiplicateur de Lagrange en torsion elasto-plastique. *Arch. Rational Mech. Anal.* 49, 32–40.
- Ciarlet, P. G., 1978. The finite element method for elliptic problems. Vol. 4 of *Studies in Mathematics and its Applications*. North-Holland Publishing Co., Amsterdam-New York-Oxford.

- Farina, A., Kawohl, B., 2008. Remarks on an overdetermined boundary value problem. *Calc. Var.* 31, 351–357.
- Gago, J. P., Kelly, D. W., Zienkiewicz, O. C., Babuška, I., 1983. A posteriori error analysis and adaptive processes in the finite element method. II. Adaptive mesh refinement. *Internat. J. Numer. Methods Engrg.* 19 (11), 1621–1656.
- Glowinski, R., Lions, J.-L., Trémolières, R., 1981. Numerical analysis of variational inequalities. Vol. 8 of *Studies in Mathematics and its Applications*. North-Holland Publishing Co., Amsterdam, translated from the French.
- Huang, Y., Li, R., Liu, W., 2007. Preconditioned descent algorithms for p -Laplacian. *Journal of Scientific Computing* 32 (2), 343–371.
- Kelly, D. W., Gago, J. P., Zienkiewicz, O. C., Babuška, I., 1983. A posteriori error analysis and adaptive processes in the finite element method. I. Error analysis. *Internat. J. Numer. Methods Engrg.* 19 (11), 1593–1619.
- Nocedal, J., Wright, S. J., 2006. Numerical optimization, 2nd Edition. Springer Series in Operations Research and Financial Engineering. Springer, New York.

Table 1: Parameters used in Algorithm 3, by section.

parameter name	default value	pattern	description
Global Parameters			
p	100	Double(2,1)	Penalization parameter.
known_solution	true	Bool()	Whether the exact solution is known.
Mesh & Refinement Parameters			
Code for the domain	0	Integer(0,2)	Number identifying the domain in which we solve the problem 0: circle, 1: rectangle, 3: glowinski
No of initial refinements	4	Integer(0)	Number of global mesh refinement steps applied to initial coarse grid
No of adaptive refinements	8	Integer(0)	Number of global adaptive mesh refinements
top_fraction_of_cells	0.25	Double(0)	refinement threshold
bottom_fraction_of_cells	0.05	Double(0)	coarsening threshold
Algorithm Parameters			
Descent_direction	0	Integer(0,1)	0: Preconditioned descent, 1: Newton Method
init_p	10	Double(2)	Initial p
delta_p	50	Double(0)	increase of p
Max_CG_it	1500	Integer(1)	Maximum Number of CG iterations
CG_tol	1e-10	Double(0)	Tolerance for CG iterations
max_LS_it	45	Integer(1)	Maximum Number of LS iterations
line_search_tolerance	1e-6	Double(0)	line search tolerance constant
init_step_length	1e-2	Double(0)	initial step length in line-search
Max_inner	800	Integer(1)	Maximum Number of inner iterations
eps	1.0e-8	Double(0)	Threshold on norm of the derivative to declare optimality achieved
hi_eps	1.0e-9	Double(0)	Threshold on norm of the derivative to declare optimality achieved in for highly refined mesh
hi_th	8	Integer(0)	Number of adaptive refinement before change convergence threshold