

# Optimization and an FEM

a 2D finite element method applied to the  $p$ -Laplacian

Ed Bueler

Dept. of Mathematics and Statistics, UAF

12 April 2016

# outline for today

## Chapter 5 of book:

- ▶ the  $p$ -Laplacian equation

$$-\nabla \cdot (|\nabla u|^{p-2} \nabla u) = f$$

- $p > 1$ ; eqn is nonlinear if  $p \neq 2$
- arises from minimizing objective  $I[u] = \int_{\Omega} \frac{1}{p} |\nabla u|^p - fu$
- generalizes Poisson equation ( $p = 2$  case) in Chap 2
- ▶ introduce  $Q^1$  structured-grid finite element method
- ▶ show code `c/ch5/plap.c`:
  - objective-only implementation with `-snes_fd_function`

# continuum problem: minimize power of gradient

- ▶  $\Omega \subset \mathbb{R}^2$  is (open) domain with nice boundary
  - also given  $f \in C(\overline{\Omega})$
- ▶ functions with integrable gradient:

$$W^{1,p}(\Omega) = \left\{ w : \int_{\Omega} |w|^p < \infty \text{ \& \, } \int_{\Omega} |\nabla w|^p < \infty \right\}$$

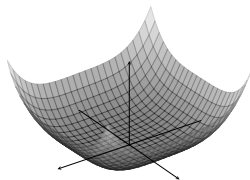
- actually want functions with given boundary values,  $u|_{\partial\Omega} = g$ , so consider this subspace:

$$W_g^{1,p}(\Omega) = \{ w \in W^{1,p}(\Omega) \text{ \& \, } w|_{\partial\Omega} = g \}$$

- ▶ *goal*: minimize  $\infty$ -dimn'l functional

$$I[u] = \int_{\Omega} \frac{1}{p} |\nabla u|^p - fu$$

- ▶ think: minimize a polynomial surface



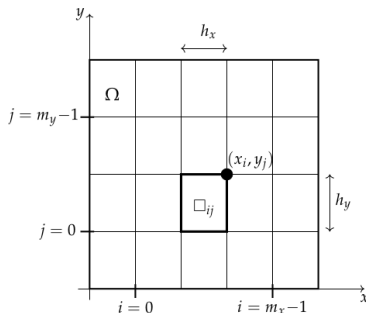
## just do it!

- ▶ go no further ... can we just find the minimum now, without doing any more calculus?
- ▶ steps:
  1. write code that computes the functional  $I[u]$  from a representation  $\mathbf{u} = u_{ij}$  of  $u$  based on a grid
  2. have PETSc find derivative  $\mathbf{F}(\mathbf{u}) \approx I'[u]$  by finite differences
  3. have PETSc solve nonlinear equations  $\mathbf{F}(\mathbf{u}) = 0$ 
    - have PETSc compute Jacobian
    - *also* by finite differences ... dodgy?
- ▶ yes, this sort of works
  - it is a path *toward* a scalable implementation
  - a secondary benefit ... line search (next week)
- ▶ provides concrete transition to finite element method

# grid of elements

- ▶ domain is unit square  
 $\Omega = (0, 1) \times (0, 1)$
- ▶ consider grid  $(x_i, y_j)$  with *interior* locations as unknowns  $u_{ij}$ 
  - boundary values from  $g$
  - $N = m_x m_y$  unknowns
- ▶  $(m_x + 1)(m_y + 1)$  *elements*  $\square_{ij}$ 
  - dimensions  $h_x \times h_y$
  - indexed by upper-right corner
  - functional can be computed element-by-element:

$$I[u] = \sum_{i=0}^{m_x} \sum_{j=0}^{m_y} \int_{\square_{ij}} \frac{1}{p} |\nabla u|^p - fu$$



# $Q^1$ finite element method (FEM)

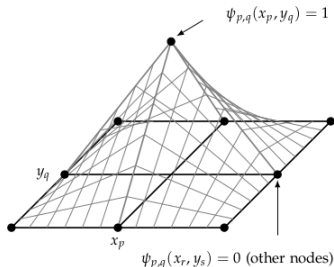
- ▶ in a  $Q^1$  FEM, each function is piecewise-bilinear

$$u(x, y) = a + bx + cy + dxy$$

- ▶ let  $S^h$  be space of *continuous* piecewise-bilinear functions
  - $S^h \subset W^{1,p}(\Omega)$
- ▶ also define  $S_g^h \subset S^h$  with boundary values  $g$ 
  - $\dim(S_g^h) = N = (\text{number of interior nodes})$

- ▶ hat function  $\psi_{p,q}(x, y) \in S^h$ 
  - equals one at  $(x_p, y_q)$
  - zero at other nodes
  - $\{\psi_{p,q}\}$  is basis of  $S^h$
- ▶ for  $v \in S_0^h$ :

$$v(x, y) = \sum_{i=0}^{m_x-1} \sum_{j=0}^{m_y-1} v_{i,j} \psi_{i,j}(x, y)$$



# formulas on one element

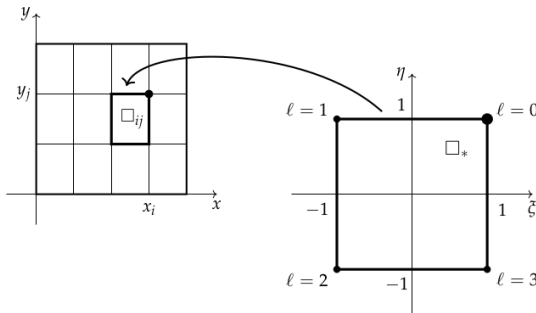
- ▶ define *reference element*

$$\square_* = [-1, 1] \times [-1, 1]$$

- corners labeled  $\ell = 0, 1, 2, 3$
- ▶ basis for bilinear functions on  $\square_*$ :

$$\chi_\ell(\xi, \eta) = \frac{1}{4} (1 + \xi_\ell \xi) (1 + \eta_\ell \eta)$$

- monomial basis  $\{1, \xi, \eta, \xi\eta\}$  is not convenient



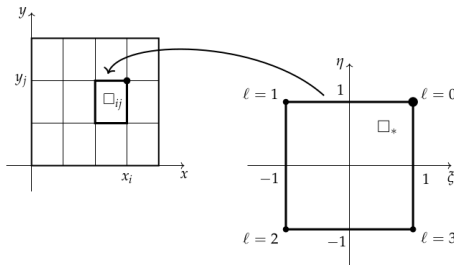
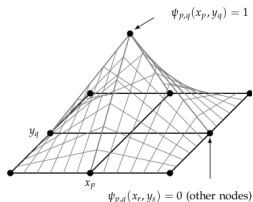
## formulas on one element 2

- important formulas on reference element:

$$v(\xi, \eta) = \sum_{\ell=0}^3 v_{\ell} \chi_{\ell}(\xi, \eta)$$

$$\psi_{p,q}(x(\xi, \eta), y(\xi, \eta)) = \chi_{\ell}(\xi, \eta)$$

$$\det \frac{\partial(x, y)}{\partial(\xi, \eta)} = \det \begin{bmatrix} \frac{h_x}{2} & 0 \\ 0 & \frac{h_y}{2} \end{bmatrix} = \frac{h_x h_y}{4}$$





## quadrature on quadrilaterals

- ▶  $n$  point Gauss-Legendre quadrature on interval  $[-1, 1]$ :

$$\int_{-1}^1 f(z) dz \approx \sum_{q=0}^{n-1} w_q f(z_q)$$

- ▶ on reference element  $\square_*$ :

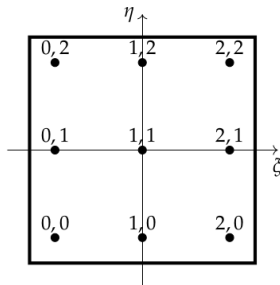
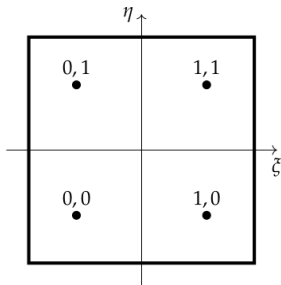
$$\int_{\square_*} v(\xi, \eta) d\xi d\eta \approx \sum_{r=0}^{n-1} \sum_{s=0}^{n-1} w_r w_s v(z_r, z_s)$$

- ▶ we can go back to  $x, y$  element  $\square_{ij}$  by change of variables:

$$\int_{\square_{ij}} v(x, y) dx dy = \frac{h_x h_y}{4} \int_{\square_*} v(\xi, \eta) d\xi d\eta$$

## $n = 1, 2, 3$ Gauss-Legendre quadrature

$n$	nodes $z_q$	weights $w_q$
1	0	2
2	$-\frac{1}{\sqrt{3}}, +\frac{1}{\sqrt{3}}$	1, 1
3	$-\sqrt{\frac{3}{5}}, 0, +\sqrt{\frac{3}{5}}$	$\frac{5}{9}, \frac{8}{9}, \frac{5}{9}$



# approximation to functional

- ▶ let

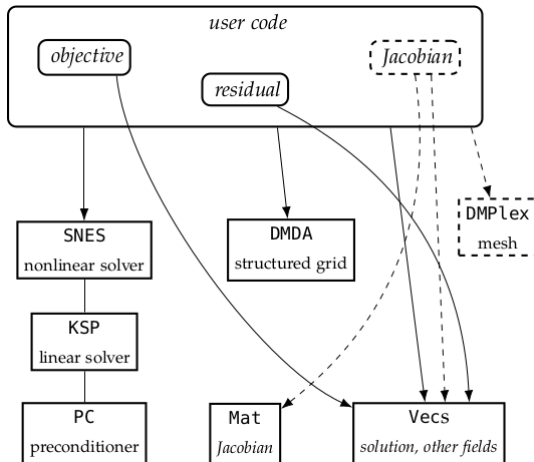
$$G_{ij}(\xi, \eta) = \left[ \frac{1}{p} |\nabla u|^p - fu \right]_{\square_*}$$

- details in Exercise 5.7 in book
- ▶ putting it all together, we have a computable approximation to the functional which we want to minimize
- ▶ this is our *objective*:

$$I^h[u] = \frac{h_x h_y}{4} \underbrace{\sum_{i=0}^{m_x} \sum_{j=0}^{m_y}}_{\text{sum over elements}} \underbrace{\sum_{r=0}^{n-1} \sum_{s=0}^{n-1}}_{\text{sum over quadrature points}} w_r w_s G_{ij}(z_r, z_s)$$

## structure of `c/ch5/plap.c`

- ▶ dashed lines denote un-implemented code parts
- ▶ diagram like this appears for all remaining book examples



# run the code with objective-only

- ▶ look at `c/ch5/plap.c`

- ▶ do:

```
$ cd c/ch5
```

```
$ make plap
```

```
$ ./plap -snes_monitor -snes_fd_function -snes_fd
```

```
$ ./plap -snes_monitor -snes_fd_function -snes_fd \  
-da_refine 1
```

```
$ ./plap -snes_monitor -snes_fd_function -snes_fd \  
-da_refine 2
```

- ▶ hmmm ...

- ▶ evaluations:

```
$ ./plap -snes_monitor -snes_fd -snes_fd_function \  
-log_view |grep Eval
```

```
$ ./plap -snes_monitor -snes_fd -snes_fd_function \  
-log_view -da_refine 1 |grep Eval
```

- ▶ wow!