p-Laplacian solved, and line search

it's nice for SNES to have an objective

Ed Bueler
Dept. of Mathematics and Statistics, UAF

19 April 2016

outline for today

Chapter 5 of book:

- recall p-Laplacian equation

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

  - arises from minimizing objective $I[u] = \int_\Omega \frac{1}{p} |\nabla u|^p - fu$
  - discretized with Q^1 structured-grid finite element method
- code `c/ch5/plap.c`:
  - objective-only prototype implementation:
    - i.e. option `-snes_fd_function`
    - severe scalability problems
  - add residual to succeed:
    - compare `-snes_fd` and `-snes_fd_color`
    - show clear evidence of convergence
    - compare quadrature degree
- line search for Newton's method

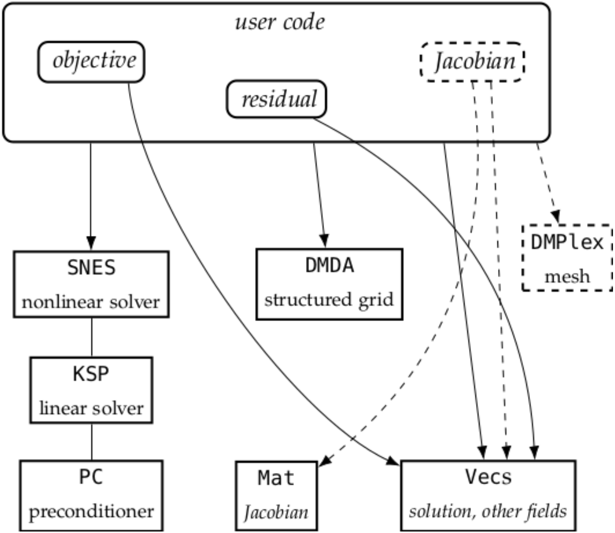
optimization vs solving equations

- it is possible to lose track of which "level" stuff is at
- a correspondence ($N =$ number of unknowns):

output dim	*optimization*	$\leftrightarrow$	*solving equations*
1	$\min I[u]$ <u>objective</u>		?
N	$\nabla I[u] = 0$ <u>gradient</u>		$\mathbf{F}(u) = 0$ <u>residual</u>
N^2	$H_{ij}(u) = \frac{\partial^2 I}{\partial u_i \partial u_j}(u)$ <u>Hessian</u>		$J_{ij}(u) = \frac{\partial F_i}{\partial u_j}(u)$ <u>Jacobian</u>

- `c/ch5/plap.c` implements
  - objective in `FormObjectiveLocal()`
  - gradient (=residual) in `FormFunctionLocal()`
  - but *not* Hessian (=Jacobian)

structure of `c/ch5/plap.c`



with objective-only: breaks easily

- try it out:

```
$ cd c/ch5
$ make plap
$ ./plap -snes_monitor -snes_fd_function -snes_fd
$ ./plap -snes_monitor -snes_fd_function -snes_fd_color
```

- flaky behavior under refinement; if `LEV`$= 1, 2, 3$ then

```
$ ./plap -snes_monitor -snes_fd_function -snes_fd_color \
    -snes_converged_reason -da_refine $LEV
```

  gives `DIVERGED` for 1,3 levels and `CONVERGED` for 2 level

- fixable with weaker Newton tolerance:

```
$ ./plap -snes_monitor -snes_fd_function -snes_fd_color \
    -snes_converged_reason -da_refine $LEV \
    -snes_linesearch_type basic -snes_rtol 1e-6
```

- ...but this is unsustainable (next slide)

with objective-only: too many objective evaluations

- count evaluations of `FormObjectiveLocal()` $= I^h[u]$
- for `LEV`$= 0, 1, 2, 3$ do:

```
$ ./plap -snes_fd_function -snes_fd_color \
     -snes_linesearch_type basic -snes_rtol 1e-6 \
     -log_view -da_refine $LEV |grep Eval
```

- result:

level	*grid*	*error*	*# of objective evals*
0	3×3	8.1×10^{-3}	1428
1	5×5	2.9×10^{-3}	4636
2	9×9	9.3×10^{-4}	12444
2	17×17	2.6×10^{-4}	104160

  - evidence for convergence at $O(\Delta x^2)$ or $O(\Delta x^{1.5})$??
  - but it is too much work per grid point

back to calculus: get gradient of objective

- we can differentiate the thing we are optimizing:

$$\begin{aligned} I[u+\epsilon v] - I[u] &= \int_\Omega \frac{1}{p}|\nabla u + \epsilon \nabla v|^p - \frac{1}{p}|\nabla u|^p - \epsilon f v \\ &= \epsilon \left(\int_\Omega |\nabla u|^{p-2} \nabla u \cdot \nabla v - f v \right) + O(\epsilon^2). \end{aligned}$$

  - if u is minimum of $I[u]$ then $I[u+\epsilon v] - I[u] \geq 0$
- get formula for the *gradient*:

$$\nabla I[u](v) = \lim_{\epsilon \to 0} \frac{I[u+\epsilon v] - I[u]}{\epsilon} = \int_\Omega |\nabla u|^{p-2} \nabla u \cdot \nabla v - f v$$

  - for each $u \in W^{1,p}(\Omega)$ it's a map $\nabla I[u] : W^{1,p}(\Omega) \to \mathbb{R}$

p-Laplacian weak formulation: $\mathbf{F}(u) = 0$

- the *equations* we want to solve are "derivative $= 0$":

$$\nabla I[u](v) = 0 \quad \forall v \qquad \iff \qquad \mathbf{F}(u) = 0$$

- equation is the *weak form*

$$\int_\Omega |\nabla u|^{p-2} \nabla u \cdot \nabla v - fv = 0$$

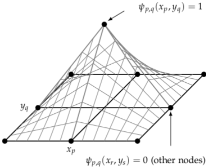
  - each $v \in W_0^{1,p}(\Omega)$ gives an equation which unknown $u \in W_g^{1,p}(\Omega)$ must satisfy
  - $F_i(u) = \int_\Omega |\nabla u|^{p-2} \nabla u \cdot \nabla v_i - fv_i$ is a scalar component of the residual

- additional integration-by-parts gives *strong form*

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

discrete weak form is residual

- hat functions $\psi_{pq}(x, y)$ form basis of S_0^h



- we want $u \in S_g^h$ so that $\nabla I^h[u](\psi_{pq}) = 0$ for all p, q
- integral over Ω is sum over elements $\square_{ij}$:

$$\sum_{i,j} \int_{\square_{ij}} |\nabla u|^{p-2} \nabla u \cdot \nabla \psi_{pq} - f \psi_{pq} = 0$$

- each grid location p, q gives one component of residual:

$$F_{pq}(u) = \sum_{i,j} \int_{\square_{ij}} |\nabla u|^{p-2} \nabla u \cdot \nabla \psi_{pq} - f \psi_{pq}$$

residual on one element

- need residual on element $\square_{ij}$

$$H_{ij}^{pq}(x,y) = \Big[|\nabla u|^{p-2}\nabla u\cdot\nabla\psi_{pq} - f\psi_{pq}\Big]_{\square_{ij}}$$

- …but we want it on the ref. element, for quadrature
- only four ψ_{pq} are nonzero on $\square_{ij}$
  - correspond to element corners $\ell = 0,1,2,3$:

$$\psi_{pq} = \chi_\ell$$

  - so define:

$$H_{ij}^{\ell}(\xi,\eta) = \left[|\nabla u|^{p-2}\nabla u\cdot\nabla\chi_\ell - f\chi_\ell\right]_{\square_*}$$

  - details in Exercise 5.8

discrete weak form with quadrature

- recall Gauss-Legendre quadrature

$$\int_{\square_{ij}} v(x,y)\,dx\,dy = \frac{h_x h_y}{4} \int_{\square_*} v(\xi,\eta)\,d\xi\,d\eta$$
$$\approx \frac{h_x h_y}{4} \sum_{r=0}^{n-1} \sum_{s=0}^{n-1} w_r w_s v(z_r, z_s)$$

- putting it all together, we loop over elements i, j and add each element's contribution to the p, q residual:

$$F_{p,q}(u) \;\mathrel{+}=\; \frac{h_x h_y}{4} \sum_{r=0}^{n-1} \sum_{s=0}^{n-1} w_r w_s H_{ij}^{\ell}(z_r, z_s),$$

  - *if* the ℓ corner of $\square_{ij}$ is node p, q
  - details in `FormFunctionLocal()`
- then we solve $N = m_x m_y$ equations

$$\mathbf{F}(u) = 0$$

it makes all the difference

- we can count evaluations of `FormObjectiveLocal()` $= I^h[u]$ and `FormFunctionLocal()` $= \mathbf{F}(u)$
- objective-only:

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

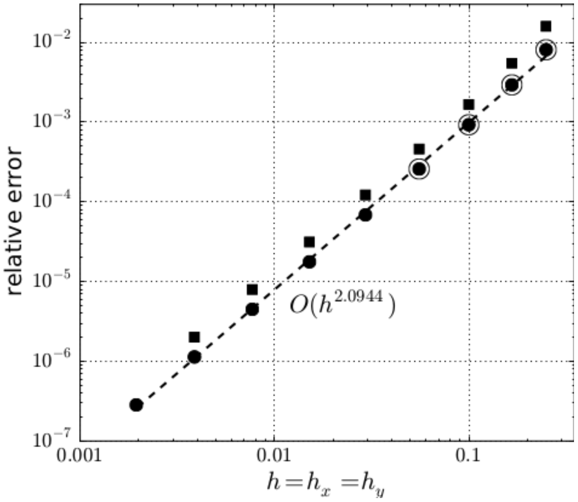
  gives 5333 objective evals
- with residual **F**:

  ```
  $ ./plap -da_refine 1 -log_view |grep Eval
  ```

  gives 8 objective (why?) and 45 function evals

- two versions give same numerical error
- residual version gives fewer Newton steps (why?)

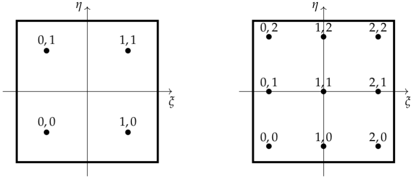
convergence



- dots for residual version, circles for objective-only
- convergence rate: good!

effect of quadrature

- recall we implemented $n = 1, 2, 3$ order Gauss-Legendre quadrature in `plap.c`



- compare numerical error:

```
$ timer ./plap -da_refine 4 -plap_quaddegree 1
numerical error:  |u-u_exact|/|u_exact| = 1.219e-04
real 0.20
$ timer ./plap -da_refine 4 -plap_quaddegree 2
numerical error:  |u-u_exact|/|u_exact| = 6.863e-05
real 0.44
$ timer ./plap -da_refine 4 -plap_quaddegree 3
numerical error:  |u-u_exact|/|u_exact| = 6.863e-05
real 0.86
```

- squares on prev graph show $n = 1$ quadrature results
- conclude: $n = 2$ good . . . thus it is the default

line search

- recall Newton's method:

$$J_{\mathbf{F}}(\mathbf{x}_k)\,\mathbf{s} = -\mathbf{F}(\mathbf{x}_k)$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}$$

- full step $\mathbf{s}_k$ may not reduce $I[\mathbf{x}]$ ... *if* there is an objective function $I[\mathbf{x}]$ at all!
- options:
  - if there is objective function $I[\mathbf{x}]$ then do line search

$$\min_{\lambda>0} I[\mathbf{x}_k + \lambda \mathbf{s}_k]$$

  - if not, define *merit function* $\phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|_2^2$ and then do line search

$$\min_{\lambda>0} \phi(\mathbf{x}_k + \lambda \mathbf{s}_k)$$

line search control in PETSc

- options for `-snes_linesearch_type`:

Name	Summary
`basic`	no line search; use full Newton step $\lambda_k = 1$
`bt` [*default*]	polynomial-fit back-tracking; uses obj. if present
`cp`	assume **F** is gradient of obj. and find crit. point
`l2`	secant-line minimization; fixed repeats

- use `-snes_linesearch_monitor` to see action