# DUNE PDELab Tutorial 01

## Conforming FEM for a Nonlinear Poisson Equation

Peter Bastian

Interdisziplinäres Zentrum für Wissenschaftliches Rechnen
Im Neuenheimer Feld 205, D-69120 Heidelberg

August 31, 2021

# Motivation

This tutorial extends on tutorial 00 by

1) Solving a **nonlinear** stationary PDE
2) Using conforming finite element spaces of **arbitrary order**
3) Using **different types of (conforming) meshes** (simplicial, cubed and mixed)
4) Using **multiple types of boundary conditions**

# PDE Problem

We consider the problem

$$
\begin{aligned}
-\Delta u + q(u) &= f && \text{in } \Omega, && (1a)\\
u &= g && \text{on } \Gamma_D \subseteq \partial\Omega, && (1b)\\
-\nabla u \cdot \nu &= j && \text{on } \Gamma_N = \partial\Omega \setminus \Gamma_D. && (1c)
\end{aligned}
$$

- $q : \mathbb{R} \to \mathbb{R}$ is possibly nonlinear function
- $f : \Omega \to \mathbb{R}$ the source term
- $\nu$ unit outer normal to the domain

# Weak Formulation

$$\text{Find } u \in U \text{ s.t.:} \quad r^{\mathrm{NLP}}(u, v) = 0 \quad \forall v \in V, \tag{2}$$

with the continuous residual form

$$r^{\mathrm{NLP}}(u, v) = \int_{\Omega} \nabla u \cdot \nabla v + (q(u) - f)v \, dx + \int_{\Gamma_N} jv \, ds$$

and the function spaces

- $U = \{v \in H^1(\Omega) \,:\, \text{“}v = g\text{” on } \Gamma_D\}$ (affine space)
- $V = \{v \in H^1(\Omega) \,:\, \text{“}v = 0\text{” on } \Gamma_D\}$

We assume that a unique solution exists

# Algebraic Problem

$\rightarrow$ Solve weak formulation in finite-dimensional spaces

$$U_h = \mathrm{span}\{\phi_1, \ldots, \phi_n\}, \quad V_h = \mathrm{span}\{\psi_1, \ldots, \psi_m\}$$

Expanding the solution $u_h = \sum_{j=1}^n (z)_j \phi_j$ results in an algebraic equation for $z \in \mathbb{R}^n$:

$$\begin{aligned}
\text{Find } u_h \in U_h \text{ s.t.:} \qquad & & r(u_h, v) = 0 & \quad \forall v \in V_h \\
& \Leftrightarrow & r\left(\sum_{j=1}^n (z)_j \phi_j, \psi_i\right) = 0 & \quad \forall i = 1, \ldots, m \\
& \Leftrightarrow & R(z) = 0, &
\end{aligned}$$

where $R : \mathbb{R}^n \rightarrow \mathbb{R}^m$ given by $R_i(z) = r_h\left(\sum_{j=1}^n (z)_j \phi_j, \psi_i\right)$ is a nonlinear, vector-valued function.

# Solution of Algebraic Problem

Use *iterative* methods to solve $R(z) = 0$, fixed point iteration:

$$z^{(k+1)} = G(z^{(k)}) = z^{(k)} - \lambda^k W(z^{(k)}) R(z^{(k)}). \tag{3}$$

- $\lambda^k$ is a damping factor
- $W(z^{(k)})$ is a preconditioner matrix, e.g. in Newton's method one has

$$W(z^{(k)}) = (J(z^{(k)}))^{-1} \quad \text{where } (J(z^{(k)}))_{i,j} = \frac{\partial R_i}{\partial z_j}(z^{(k)})$$

  i.e. need to solve $J\left(z^{(k)}\right) w = R(z^{(k)})$

The following algorithmic building blocks are required:

**i)** residual evaluation $R(z)$,

**ii)** Jacobian evaluation $J(z)$ (or an approximation of it),

**iii)** matrix-free Jacobian application $J(z)w$ (or an approximation).

# Note on Matrix-free Evaluation

**Nonlinear case:**

$$(J(z)w)_i = \sum_{j=1}^{n} (J(z))_{i,j}(w)_j = \sum_{j=1}^{n} \frac{\partial}{\partial z_j} r_h\left(\sum_{l=1}^{n}(z)_l\phi_l, \psi_i\right)(w)_j.$$

**Linear case:** $r_h(u,v) = a(u,v) - l(v)$, $a$ BLF, $l$ LF

$$\begin{aligned}
(J(z)w)_i &= \sum_{j=1}^{n} \frac{\partial}{\partial z_j} r_h\left(\sum_{l=1}^{n}(z)_l\phi_l, \psi_i\right)(w)_j \\
&= \sum_{j=1}^{n} \frac{\partial}{\partial z_j}\left(a_h\left(\sum_{l=1}^{n}(z)_l\phi_l, \psi_i\right) - l_h(\psi_i)\right)(w)_j \\
&= \sum_{j=1}^{n} \frac{\partial}{\partial z_j}\left(\sum_{l=1}^{n}(z)_l a_h(\phi_l, \psi_i)\right)(w)_j \\
&= \sum_{j=1}^{n} a_h(\phi_j, \psi_i)(w)_j = a_h\left(\sum_{j=1}^{n}(w)_j\phi_j, \psi_i\right) = (Aw)_i
\end{aligned}$$

# Recall Finite Element Mesh Notation

**i)** Ordered sets of vertices and elements:

$$\mathcal{X}_h = \{x_1, \dots, x_N\}, \quad \mathcal{T}_h = \{T_1, \dots, T_M\}$$

**ii)** Partitioning of vertex index set $\mathcal{I}_h = \{1, \dots, N\}$ into $\mathcal{I}_h = \mathcal{I}_h^{int} \cup \mathcal{I}_h^{\partial\Omega}$:

$$\mathcal{I}_h^{int} = \{i \in \mathcal{I}_h \,:\, x_i \in \Omega\}, \quad \mathcal{I}_h^{\partial\Omega} = \{i \in \mathcal{I}_h \,:\, x_i \in \partial\Omega\}.$$

**iii)** For every element $T \in \mathcal{T}_h$ a local-to-global map

$$g_T : \{0, \dots, n_T - 1\} \to \mathcal{I}_h$$

**iv)** For every element $T \in \mathcal{T}_h$ an element transformation map

$$\mu_T : \hat{T} \to T$$

$\mu_T$ is differentiable with invertible Jacobian and consistent with $g_T$:

$$\forall i \in \{0, \dots, n_T - 1\} : \mu_T(\hat{x}_i) = x_{g_T(i)}$$

# Conforming Finite Element Space

with **polynomial degree** $k$ in **dimension** $d$ on **mesh** $\mathcal{T}_h$:

$$V_h^{k,d}(\mathcal{T}_h) = \left\{ v \in C^0(\overline{\Omega}) \,:\, \forall T \in \mathcal{T}_h : v|_T = \mu_T \circ p_T \wedge p_T \in \mathbb{P}_T^{k,d} \right\}$$

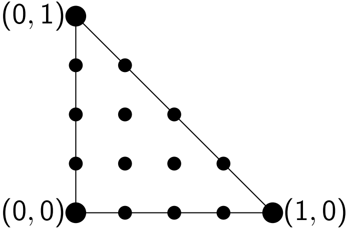
where the multivariate polynomials $\mathbb{P}$ depend on element type:

$$\mathbb{P}_T^{k,d} = \begin{cases} \left\{ p \,:\, p(x_1, \ldots, x_d) = \sum_{0 \le \|\alpha\|_1 \le k} c_\alpha x_1^{\alpha_1} \cdot \ldots \cdot x_d^{\alpha_d} \right\} & \hat{T} = \hat{S} \text{ (simplex)}, \\ \left\{ p \,:\, p(x_1, \ldots, x_d) = \sum_{0 \le \|\alpha\|_\infty \le k} c_\alpha x_1^{\alpha_1} \cdot \ldots \cdot x_d^{\alpha_d} \right\} & \hat{T} = \hat{C} \text{ (cube)} \end{cases}$$

The dimension of $\mathbb{P}_T^{k,d}$ is:

$$n_{\hat{C}}^{k,d} = (k+1)^d \text{ (cube)}, \quad n_{\hat{S}}^{k,d} = \begin{cases} 1 & k = 0 \vee d = 0 \\ \sum_{i=0}^{k} n_{\hat{S}}^{i,d-1} & \text{else} \end{cases} \quad \text{(simplex)}$$

# Local Lagrange Basis




Lagrange points and poynomials (shape functions) on $\hat{T}$:

$$L_{\hat{T}} = \left\{ \hat{x}_0^{\hat{T}}, \ldots, \hat{x}_{n_{\hat{T}}^{k,d}-1}^{\hat{T}} \right\}, \quad P_{\hat{T}} = \left\{ p_0^{\hat{T}}, \ldots, p_{n_{\hat{T}}^{k,d}-1}^{\hat{T}} \right\}$$

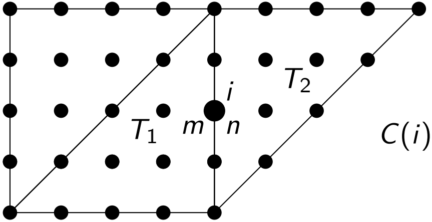
such that

$$p_i^{\hat{T}}(\hat{x}_j^{\hat{T}}) = \delta_{i,j}$$

Extend local to global map:

$$g_T : \{0, \ldots, n_{\hat{T}}^{k,d} - 1\} \to \mathcal{I}\left(V_h^{k,d}(\mathcal{T}_h)\right) = \left\{0, \ldots, \dim V_h^{k,d}(\mathcal{T}_h) - 1\right\}$$

# Global Lagrange Basis




$$C(i) = \{(T_1, m), (T_2, n)\}$$

Define inversion of the global map:

$$C(i) = \{(T, m) \in \mathcal{T}_h \times \mathbb{N} \, : \, g_T(m) = i\}$$

then the global Lagrange basis functions are

$$\phi_i(x) = \begin{cases} p_m^{\hat{T}}(\mu_T^{-1}(x)) & x \in T \wedge (T, m) \in C(i) \\ 0 & \text{else} \end{cases} \quad , \quad i \in \mathcal{I}\left(V_h^{k,d}(\mathcal{T}_h)\right).$$

corresponding to the global Lagrange points

$$\mathcal{X}_h^{k,d} = \left\{x_i \in \overline{\Omega} \, : \, x_i = \mu_T(\hat{x}_m^{\hat{T}}) \wedge (T, m) \in C(i)\right\}$$

# Dirichlet Boundary Conditions

Indices of Lagrange points on the Dirichlet boundary are:

$$\mathcal{I}^D\left(V_h^{k,d}(\mathcal{T}_h)\right) = \left\{i \in \mathcal{I}\left(V_h^{k,d}(\mathcal{T}_h)\right) \,:\, x_i \in \mathcal{X}_h^{k,d} \cap \Gamma_D\right\}.$$

Then the test space with zero Dirichlet condition is:

$$V_{h,0}^{k,d}(\mathcal{T}_h) = \left\{v \in V_h^{k,d}(\mathcal{T}_h) \,:\, v(x_i) = 0 \quad \forall i \in \mathcal{I}^D\left(V_h^{k,d}(\mathcal{T}_h)\right)\right\}$$

For the trial space choose any extension

$$u_{h,g} = \sum_{i \in \mathcal{I}\left(V_h^{k,d}(\mathcal{T}_h)\right)} u_g(x_i)\phi_i \qquad u_g(x_i) = g(x_i)\ \forall i \in \mathcal{I}^D\left(V_h^{k,d}(\mathcal{T}_h)\right)$$

Then

$$U_h^{k,d}(\mathcal{T}_h) = \left\{u \in V_h^{k,d}(\mathcal{T}_h) \,:\, u = u_{h,g} + w \wedge w \in V_{h,0}^{k,d}(\mathcal{T}_h)\right\}$$

# General Constraints

**Task:** Given $U_h = \text{span}\left\{\phi_j \,:\, j \in J_h = \{1, \ldots, n\}\right\}$ construct $\tilde{U}_h \subseteq U_h$

This is how it is done in PDELab:

1) Assume $U_h = \text{span}\{\phi_i \,:\, i \in J_h\}$
2) Select a subset of indices $\tilde{J}_h \subset J_h$
3) Set $\tilde{U}_h = \text{span}\left\{\tilde{\phi}_j \,:\, j \in \tilde{J}_h\right\}$, where the new basis functions have the form

$$\tilde{\phi}_j = \phi_j + \sum_{l \in J_h \setminus \tilde{J}_h} (B)_{j,l} \phi_l \quad \forall j \in \tilde{J}_h.$$

Any subspace is thus characterized by $C = (\tilde{J}_h, B)$

# Element-wise Computations

Come back to the residual form which is element-wise

$$r^{\mathrm{NLP}}(u,v) = \sum_{T\in\mathcal{T}_h} \alpha_T^V(u,v) + \sum_{T\in\mathcal{T}_h} \lambda_T^V(v) + \sum_{F\in\mathcal{F}_h^{\partial\Omega}} \lambda_F^B(v)$$

with

$$\alpha_T^V(u,v) = \int_T \nabla u \cdot \nabla v + q(u)v \, dx$$
$$\lambda_T^V(v) = -\int_T fv \, dx$$
$$\lambda_F^B(v) = \int_{F\cap\Gamma_N} jv \, ds$$

$\mathcal{F}_h^{\partial\Omega}$: intersections of elements with the domain boundary, i.e.

$$F = T_F^- \cap \partial\Omega$$

with $T_F^- \in \mathcal{T}_h$ the "minus" element associated with $F$

# $\lambda$ Volume Term

For any $(T, m) \in C(i)$ we obtain

$$\lambda_T^V(\phi_i) = -\int_T f\phi_i\,dx = -\int_{\hat{T}} f(\mu_T(\hat{x}))p_m^{\hat{T}}(\hat{x})|\mathrm{det}J_{\mu_T}(\hat{x})|\,d\hat{x}.$$

$J_{\mu_T}$ is the Jacobian of the element map $\mu_T$

This integral is computed using numerical quadrature

Collect all contributions of $T$ in a small vector:

$$(\mathcal{L}_T^V)_m = -\int_{\hat{T}} f(\mu_T(\hat{x}))p_m^{\hat{T}}(\hat{x})|\mathrm{det}J_{\mu_T}(\hat{x})|\,d\hat{x}.$$

# $\lambda$ Boundary Term

For $F \in \mathcal{F}_h^{\partial\Omega}$ with $F \subseteq \Gamma_N$ and $(T_F^-, m) \in C(i)$ we obtain

$$\lambda_T^B(\phi_i) = \int_F jv\, ds = \int_{\hat{F}} j(\mu_F(\hat{x}))p_m^{\hat{T}}(\eta_F(\hat{x}))\sqrt{|\det(J_{\mu_F}^T(\hat{x})J_{\mu_F}(\hat{x}))|}\, ds$$

The integration is more involved here because it is over a face:

- $\mu_F : \hat{F} \to F$ maps the reference element of the face to the face
- $\eta_F : \hat{F} \to \hat{T}_F^-$ maps reference element of the face to reference element of $T_F^-$

Collect all contributions of $F$ in a small vector:

$$(\mathcal{L}_T^B)_m = \int_{\hat{F}} j(\mu_F(\hat{x}))p_m^{\hat{T}}(\eta_F(\hat{x}))\sqrt{|\det J_{\mu_T}^T(\hat{x})J_{\mu_T}(\hat{x})|}\, ds.$$

Numerical quadrature is applied to compute the integral

# $\alpha$ Volume Term

For any $(T, m) \in C(i)$ we get

$$\begin{aligned}
\alpha_T^V(u_h, \phi_i) &= \int_T \nabla u \cdot \nabla \phi_i + q(u)\phi_i \, dx, \\
&= \int_T \sum_j (z)_j \left(\nabla \phi_j \cdot \nabla \phi_i\right) + q\left(\sum_j (z)_j \phi_j\right) \phi_i \, dx, \\
&= \int_{\hat{T}} \sum_n (z)_{g_T(n)} (J_{\mu_T}^{-1}(\hat{x}) \hat{\nabla} p_n^{\hat{T}}(\hat{x})) \cdot (J_{\mu_T}^{-1}(\hat{x}) \hat{\nabla} p_m^{\hat{T}}(\hat{x})) \\
&\qquad + q\left(\sum_n (z)_{g_T(n)} p_n^{\hat{T}}(\hat{x})\right) p_m^{\hat{T}}(\hat{x}) |\mathrm{det} J_{\mu_T}(\hat{x})| \, dx
\end{aligned}$$

and again collect all contributions from $T$ in a small vector:

$$(\mathcal{R}_T^V(R_T z))_m = \alpha_T^V\left(\sum_n (R_T z)_n p_n^{\hat{T}}, p_m^{\hat{T}}\right)$$

# Putting It All Together

With these local contributions the evaluation of the algebraic residual is

$$R(z) = \sum_{T \in \mathcal{T}_h} R_T^T \mathcal{R}_T^V(R_T z) + \sum_{T \in \mathcal{T}_h} R_T^T \mathcal{L}_T^V + \sum_{F \in \mathcal{F}_h^{\partial\Omega} \cap \Gamma_N} R_T^T \mathcal{L}_F^B$$

where $R_T$ is the "picking out" matrix of element $T$

The Jacobian of $R$ is

$$(J(z))_{i,j} = \frac{\partial R_i}{\partial z_j}(z) = \sum_{(T,m,n):(T,m) \in C(i) \wedge (T,n) \in C(j)} \frac{\partial (\mathcal{R}_T^V)_m}{\partial z_n}(R_T z)$$

Note that:

**i)** Entries of the Jacobian can be computed element by element.

**ii)** The derivative is independent of the $\lambda$-terms

**iii)** Jacobian entries may be computed by numerical differentiation

# Implementation Overview

1) `tutorial01.ini` holds parameters controlling the execution
2) `tutorial01.cc` includes the necessary C++, DUNE and PDELab header files; contains `main` function calling `driver`
3) Function `driver` in `driver.hh` instantiates the necessary PDELab classes for solving a nonlinear stationary problem and finally solves the problem.
4) File `nonlinearpoissonfem.hh` contains class `NonlinearPoissonFEM` realizing a PDELab local operator
5) File `problem.hh` contains a “parameter class” which encapsulates the user-definable part of the PDE problem
6) Finally, the tutorial provides some mesh files.