DUNE PDELab — Mathematical Abstractions for the Numerical Solution of Partial Differential Equations

Peter Bastian
Universität Heidelberg,
Interdisziplinäres Zentrum für Wissenschaftliches Rechnen
Im Neuenheimer Feld 205, D-69120 Heidelberg
Peter.Bastian@iwr.uni-heidelberg.de

August 31, 2021

Abstract

This short note describes the mathematical abstractions underlying the software framework DUNE/PDELab for the numerical solution of partial differential equations. It starts with the basic building block, the stationary nonlinear problem, discusses the special case of the linear problem and then goes on to instationary problems and systems of partial differential equations. It ends with a short discussion of two fundamental approaches to parallel solution techniques.

1 Introduction

State-of-the-art numerical solvers for partial differential equations (PDEs) have a variety of aspects, among them:

1) Stationary and instationary solvers.

2) Linear and nonlinear solvers.

3) Various types of meshes in d space dimensions.

4) Error control and adaptive mesh refinement.

5) Sequential and parallel solvers (overlapping and nonoverlapping).

6) Matrix-based and matrix-free computations.

7) Low- and high-order methods.

DUNE/PDELab was designed to allow the implementation of a large variety of methods for problems from different application areas with the orthogonality of concepts in mind. This means that the different aspects listed above can be combined in an arbitrary way allowing huge flexibility. Flexibility, however, comes at a price: it requires the problem to be stated in its most abstract way to identify commonalities and differences. This little note describes these mathematical abstractions seperate from implementation aspects to highlight their interplay without cluttering the description with unnecessary detail.

2 Stationary Nonlinear Problems

The basic building block in PDELab is the solution of a stationary, nonlinear partial differential equation. The idea is to write the discretization scheme in a weighted residual formulation:

$$\text{Find } u_h \in U_h \text{ s.t.:} \quad r_h(u_h, v) = 0 \quad \forall v \in V_h. \tag{1}$$

Here U_h and V_h are finite-dimensional function spaces of dimension n and m and $r_h : U_h \times V_h \to \mathbb{K}$ ($\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$) is the residual form. The residual form is always linear in the second argument and it may be linear or nonlinear in the first argument. The subscript h on the residual form indicates that its evaluation may be mesh dependent. It turns out that a large class of finite element and finite volume schemes can be cast in this general form.

Example 1. Consider the nonlinear Poisson equation with Dirichlet and Neumann boundary conditions:

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

$\Omega \subset \mathbb{R}^d$ is a domain, $q : \mathbb{R} \to \mathbb{R}$ is a given, possibly nonlinear function and $f : \Omega \to \mathbb{R}$ is the source term and ν denotes the unit outer normal to the domain

The weak formulation of this problem reads

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

with the continuous residual form

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

The function spaces U and V are subspaces of the Sobolev space $H^1(\Omega)$ given by $U = \{v \in H^1(\Omega) : \text{“}v = g\text{” on } \Gamma_D\}$ and $V = \{v \in H^1(\Omega) : \text{“}v = 0\text{” on } \Gamma_D\}$. An element $u \in U$ may also be written as $u = u_g + w$ where $u_g \in H^1(\Omega)$ satisfies the Dirichlet boundary conditions and $w \in V$. A shorthand notation for U is then “$U = u_g + V$”.

The conforming finite element method [2, 3, 5] replaces the function spaces U and V by finite-dimensional versions U_h and V_h and uses the same residual form. □

Solving Stationary Nonlinear Problems

In order to solve the abstract problem (1) we use the fact that every finite-dimensional function space is spanned by a basis. So, assume that

$$U_h = \text{span}\{\phi_1, \ldots, \phi_n\}, \quad V_h = \text{span}\{\psi_1, \ldots, \psi_n\}.$$

Expanding the solution $u_h = \sum_{j=1}^{n} (z)_j \phi_j$ in the basis and hereby introducing the coefficient vector $z \in \mathbb{K}^n$ we can reformulate the problem as

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

where $R : \mathbb{K}^n \to \mathbb{K}^m$ 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.

The solution of the nonlinear algebraic equation $R(z) = 0$ is typically computed in an iterative fashion using e.g. a fixed-point iteration of the form

$$z^{(k+1)} = G(z^{(k)}) = z^{(k)} - W(z^{(k)}) R(z^{(k)}). \tag{2}$$

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

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

(we now assumed that $n = m$ and that the Jacobian $J(z^{(k)})$ is invertible). Newton's method requires the solution of the linear system $J(z^{(k)})w = R(z^{(k)})$ in each step which could be done using either direct or iterative methods. The implementation of the fixed-point scheme requires the following algorithmic building blocks:

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).

Only one of the methods ii) and iii) is required depending on the chosen solution procedure.

Constraints

As illustrated in example 1 the function spaces involved are often subspaces of a larger function space. PDELab provides a general approach to building a subspace of a given function space through the application of constraints.

In order to construct a subspace $\tilde{U}_h$ of $U_h = \text{span}\{\phi_j : j \in J_h = \{1, \ldots, n\}\}$

i) select a subset of indices $\tilde{J}_h \subset J_h$

ii) and set $\tilde{U}_h = \text{span}\left\{\tilde{\phi}_j \,:\, j \in \tilde{J}_h\right\}$ where the new basis functions have the form

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

Thus, any subspace of U_h is characterized by $C = (\tilde{J}_h, B)$. This abstractions allows to represent Dirichlet conditions ($J_h \setminus \tilde{J}_h$ are the indices of the Dirichlet nodes and $B = 0$), hanging nodes ($J_h \setminus \tilde{J}_h$ are the indices of hanging nodes and B represents the interpolation conditions) or even rigid body modes.

More Structure to the Residual Form

So far the residual form was deliberately kept very abstract, except for the condition on linearity with respect to the second argument. Finite element and finite volume methods, however, are defined via appropriate weak formulations which involve integrals where the domain of integration naturally decomposes into subdomains and basis functions have local support.

To that end assume that the domain Ω is covered by a mesh $\mathcal{T}_h = \{T_1, \ldots, T_M\}$ consisting of elements which are closed point sets satisfying

$$\bigcup_{T \in \mathcal{T}_h} T = \overline{\Omega}, \quad \forall T, T' \in \mathcal{T}_h, T \neq T' : \mathring{T} \cap \mathring{T}' = \emptyset. \tag{3}$$

The nonempty intersections $F = T_F^- \cap T_F^+$ of codimension 1 form the interior skeleton $\mathcal{F}_h^i = \{F_1, \ldots, F_N\}$. Each intersection is equipped with a unit normal vector ν_F pointing from T_F^- to T_F^+. The intersections of an element $F = T_F^- \cap \partial\Omega$ with the domain boundary form the set of boundary intersections $\mathcal{F}_h^{\partial\Omega} = \{F_1, \ldots, F_L\}$. Each boundary intersection is equipped with a unit normal vector ν_F which coincides with the unit outer normal to the domain. Finally, we define the restriction of a function $u \in U$ to an element by

$$(R_T u)(x) = u(x) \quad \forall x \in \mathring{T}.$$

Note that the restriction of a function to element T is only defined in the interior of T. On interior intersections F, functions may be two-valued and appropriate limits from within the elements T_F^-, T_F^+ need to be defined.

With that notation in place PDELab assumes that the residual form has the following general structure:

$$\begin{aligned} r(u,v) = &\sum_{T \in \mathcal{T}_h} \alpha_T^V(R_T u, R_T v) + \sum_{T \in \mathcal{T}_h} \lambda_T^V(R_T v) \\ &+ \sum_{F \in \mathcal{F}_h^i} \alpha_F^S(R_{T_F^-} u, R_{T_F^+} u, R_{T_F^-} v, R_{T_F^+} v) \\ &+ \sum_{F \in \mathcal{F}_h^{\partial\Omega}} \alpha_F^B(R_{T_F^-} u, R_{T_F^-} v) + \sum_{F \in \mathcal{F}_h^{\partial\Omega}} \lambda_F^B(R_{T_F^-} v). \end{aligned} \tag{4}$$

The five terms comprise volume integrals (superscript V), interior skeleton integrals (superscript S) and boundary integrals (superscript B). Furthermore, the α-terms depend on trial and test functions whereas the λ-terms only depend on the test function and involve the data of the PDE.

Example 2. The residual form from example 1 can be cast into the following form:

$$r^{\mathrm{NLP}}(u,v) = \sum_{T\in\mathcal{T}_h}\int_T \nabla u\cdot\nabla v + q(u)v\,dx - \sum_{T\in\mathcal{T}_h}\int_T fv\,dx + \sum_{F\in\mathcal{F}_h^{\partial\Omega}\cap\Gamma_N}\int_F jv\,ds$$

and involves α_T^V, λ_T^V and λ_F^B. □

Example 3. As a another example consider the cell-centered finite volume method with two-point flux approximation applied to the problem treated in example 1. In this method the discrete function space involved is the space of piecewise constant functions on the mesh

$$W_h = \{w \in L^2(\Omega) : w|_T = \text{const for all } T \in \mathcal{T}_h\}$$

and the residual form reads:

$$\begin{aligned} r_h^{\mathrm{CCFV}}(u_h,v) = &\sum_{T\in\mathcal{T}_h} q(u_h(x_T))v(x_T)|T| - \sum_{T\in\mathcal{T}_h} f(x_T)v(x_T)|T| \\ &- \sum_{F\in\mathcal{F}_h^i} \frac{u_h(x_{T_F^+}) - u_h(x_{T_F^-})}{\|x_{T_F^+} - x_{T_F^-}\|}\big(v(x_{T_F^-}) - v(x_{T_F^+})\big)|F| \\ &+ \sum_{F\in\mathcal{F}_h^{\partial\Omega}\cap\Gamma_D} \frac{u_h(x_{T_F^-})}{\|x_F - x_{T_F^-}\|}v(x_{T_F^-})|F| \\ &- \sum_{F\in\mathcal{F}_h^{\partial\Omega}\cap\Gamma_D} \frac{g(x_F))}{\|x_F - x_{T_F^-}\|}v(x_{T_F^-})|F| + \sum_{F\in\mathcal{F}_h^{\partial\Omega}\cap\Gamma_N} j(x_F)v(x_{T_F^-})|F|. \end{aligned}$$

The cell-centered finite volume method in this form is applicable on axi-parallel meshes and is based in applying Gauss' theorem, approximating the derivative in normal direction by a finite difference (two-point flux approximation) and approximating integrals by the mid-point rule (x_T, x_F denotes the center point of an element or face and $|T|$, $|F|$ denote the measure of an element or face). In this case all five different terms of the residual form are involved. Note that both terms in the last line are part of λ_F^B as they do not involve the unknown function u_h. Also note that here no constraints on the function space are necessary. Dirichlet as well as Neumann boundary conditions are built into the residual form. □

Linear Stationary Problems as a Special Case

A linear stationary problem can be considered as a special case of the nonlinear problem where the residual form reads

$$r(u_h,v) = a(u_h,v) - l(v) \tag{5}$$

with a bilinear form a_h and a linear form l_h. The corresponding algebraic function

$$R(z) = Az - b$$

is now affine linear, the matrix A and vector b are defined by

$$(A)_{i,j} = a(\phi_j,\psi_i) \quad \text{and} \quad (b)_i = l(\psi_i)$$

and the Jacobian $J(z) = A$ is independent of z. Computing the solution u_h now amounts to solving the linear system $Az = b$.

3 Instationary Problems

PDELab assumes that an instationary (nonlinear) problem can be cast into the general form

$$\text{Find } u_h(t) \in U_h \text{ s.t.:} \quad d_t m_h(u_h(t), v; t) + r_h(u_h(t), v; t) = 0 \quad \forall v \in V_h \tag{6}$$

after semi-discretization in space (method of lines approach). One way to arrive at a fully discrete scheme is now to apply the one-step-θ rule which reads:

$$\begin{aligned}\text{Find } u_h^{k+1} \in U_h \text{ s.t.:} \quad & \frac{1}{\Delta t_k}(m_h(u_h^{k+1}, v; t^{k+1}) - m_h(u_h^k, v; t^k)) + \\ & \theta r_h(u_h^{k+1}, v; t^{k+1}) + (1-\theta) r_h(u_h^k, v; t^k) = 0 \quad \forall v \in V_h.\end{aligned} \tag{7}$$

Here, $0 = t^0 < t^1 < \ldots < t^K = t_F$ is a subdivision of the time interval $\Sigma = (0, t_F)$, $\Delta t_k = t^{k+1} - t^k$ is the time step and u_h^k is an approximation of $u_h(t^k)$. For $\theta = 1$ we obtain the implicit Euler method, for $\theta = 1/2$ the Crank-Nicolson method and for $\theta = 0$ the explicit Euler method. With u_h^k given and u_h^{k+1} unknown, the left hand side of equation (7) can be considered a new residual form built up from a linear combination of the temporal residual form $m_h(u_h, v; t)$ and the spatial residual form $r_h(u_h, v; t)$ evaluated at different points in time. Various explicit and diagonally implicit Runge-Kutta [1] methods result in variants of equation (7) which yield higher-order accuracy in time under appropriate stability properties.

Example 4. As an example consider the nonlinear heat equation

$$\begin{aligned} \partial_t u - \Delta u + q(u) &= f && \text{in } \Omega \times \Sigma, \\ u &= g && \text{on } \Gamma_D \subseteq \partial\Omega, \\ -\nabla u \cdot \nu &= j && \text{on } \Gamma_N = \partial\Omega \setminus \Gamma_D, \\ u &= u_0 && \text{at } t = 0. \end{aligned}$$

Here, the parameter functions f, g, j may also depend on time (and, with some restrictions, the subdivision into Dirichlet and Neumann boundary can be time-dependent). The initial condition u_0 is a function of $x \in \Omega$. Then the residual forms involved in the conforming finite element method are

$$\begin{aligned} m^{\mathrm{NLH}}(u, v) &= \int_\Omega uv \, dx, \\ r^{\mathrm{NLH}}(u, v) &= \int_\Omega \nabla u \cdot \nabla v + (q(u) - f)v \, dx + \int_{\Gamma_N} jv \, ds. \end{aligned}$$

Note that the spatial residual form r_h^{NLH} is the same as before which allows for a very easy implementation. □

All considerations of Section 2 apply here as well. The nonlinear solver can be applied to the new residual form and for linear problems a linear algebraic problem is obtained in each time step.

Explicit Time Stepping Schemes

Considering the case of the explicit Euler method in (7) results in

$$\text{Find } u_h^{k+1} \in U_h \text{ s.t.:} \quad m_h(u_h^{k+1}, v; t) - m_h(u_h^k, v; t) + \Delta t_k r_h(u_h^k, v; t) = 0 \quad \forall v \in V_h.$$

For certain spatial schemes, e.g. finite volume or discontinuous Galerkin, and assuming m_h to be bilinear, the corresponding algebraic system to be solved is diagonal:

$$Dz^{k+1} = s^k - \Delta t^k q^k. \tag{8}$$

Moreover, a stability condition restricting the time step Δt^k has to be obeyed. The maximum allowable time step can be computed explicitly for the simplest schemes but depends on the mesh $\mathcal{T}_h$. For explicit time-stepping schemes therefore the following algorithm is employed:

i) While traversing the mesh assemble the vectors s^k and q^k separately and compute the maximum time step Δt^k.

ii) Form the right hand side $b^k = s^k - \Delta t^k q^k$ and "solve" the diagonal system $Dz^{k+1} = b^k$ (can be done in one step).

This procedure can be applied also to more general time-stepping schemes such as strong stability preserving Runge-Kutta methods [7].

4 Systems of Partial Differential Equations

All the considerations above carry over to the case of systems of partial differential equations when Cartesian products of functions spaces are introduced, i.e. the abstract stationary problem then reads

$$\text{Find } u_h \in U_h = U_h^1 \times \ldots \times U_h^s \text{ s.t.:} \quad r_h(u_h, v) = 0 \quad \forall v \in V_h = V_h^1 \times \ldots \times V_h^s \tag{9}$$

with s the number of components in the system. Again the concepts are completely orthogonal meaning that r_h might be linear or nonlinear in its first argument and the instationary case works as well as is shown in the following example.

Example 5. As an example for a system we consider the wave equation with reflective boundary conditions:

$$\begin{aligned}
\partial_{tt} u - c^2 \Delta u &= 0 && \text{in } \Omega \times \Sigma, \\
u &= 0 && \text{on } \partial\Omega, \\
u &= q && \text{at } t = 0, \\
\partial_t u &= w && \text{at } t = 0.
\end{aligned}$$

Renaming $u_0 = u$ and introducing $u_1 = \partial_t u_0 = \partial_t u$ we can write the wave equation as a system of two equations:

$$\begin{aligned}
\partial_t u_1 - c^2 \Delta u_0 &= 0 && \text{in } \Omega \times \Sigma, \\
\partial_t u_0 - u_1 &= 0 && \text{in } \Omega \times \Sigma, \\
u_0 &= 0 && \text{on } \partial\Omega, \\
u_0 &= q && \text{at } t = 0, \\
u_1 &= w && \text{at } t = 0.
\end{aligned}$$

Note that there are no constraints on the function space for u_1 and neither are there boundary conditions for the equation $u_1 = \partial_t u_0$. We note that different formulations of the wave equation as a system are possible. [4] uses $\Delta u_1 = \Delta \partial_t u_0$ as an equation to define u_1 and the formulation as a first order hyperbolic system in the equations of linear acoustics allows the use of appropriate upwinding techniques [6].

Multiplying the first equation with test function v_0 and the second with test function v_1 and using integration by parts we arrive at the weak formulation: Find $(u_0(t), u_1(t)) \in U_0 \times U_1$ s.t.

$$d_t \left[(u_0, v_1)_{0,\Omega} + (u_1, v_0)_{0,\Omega}\right] + \left[c^2(\nabla u_0, \nabla v_0)_{0,\Omega} - (u_1, v_1)_{0,\Omega}\right] = 0 \quad \forall (v_0, v_1) \in U_0 \times U_1$$

where we readily identify the temporal and spatial residual forms:

$$\begin{aligned} m^{\text{WAVE}}((u_0, u_1), (v_0, v_1)) &= (u_0, v_1)_{0,\Omega} + (u_1, v_0)_{0,\Omega}, \\ r^{\text{WAVE}}((u_0, u_1), (v_0, v_1)) &= c^2(\nabla u_0, \nabla v_0)_{0,\Omega} - (u_1, v_1)_{0,\Omega}. \end{aligned}$$

Here we used the notation of the L^2 inner product $(u, v)_{0,\Omega} = \int_\Omega uv \, dx$. □

5 Solving in Parallel

In this section we consider the solution of PDEs on parallel computers from an implementation point of view.

One approach relies on an additive splitting of the function space U_h into subspaces $U_{h,i}$:

$$U_h = U_{h,1} + \ldots + U_{h,p} \tag{10}$$

The subspaces may overlap, i.e. the splitting of any function $u_h = u_{h,1} + \ldots + u_{h,p}$ is not unique.

Given such a splitting of the trial space as well as the test space the additive subspace correction method [9] computes a new iterate as

$$u_h^{k+1} = u_h^k + \sum_{i=1}^{p} w_{h,i}, \quad r_h(u_h^k + w_{h,i}, v_i) = 0 \qquad \forall v_i \in V_{h,i}.$$

A particular choice for the space splitting is the overlapping domain decomposition method where $U_{h,i} = \text{span}\{\phi_j : j \in I_{h,i}\}$ and $I_h = \bigcup_{i=1}^p I_{h,i}$ is a possibly nonoverlapping splitting of the index set. Finite element basis functions usually have local support and the space decomposition gives rise to an overlapping domain decomposition $\Omega = \bigcup_{i=1}^p \Omega_i$ where $\Omega_i = \bigcup_{j \in I_{h,i}} \text{supp}(\phi_j)$. Even if the splitting of the index set is nonoverlapping (which corresponds to a block Jacobi solver) the domain decomposition is overlapping. In terms of the underlying matrices the subspace correction approach corresponds to a rowwise decomposition of the matrix. An advantage of the overlapping formulation is that the evaluation of the residual (4) can be performed without communication when u_h is known in each subdomain.

A second approach relies on an additive splitting of the residual form

$$r_h(u_h, v) = \sum_{i=1}^{p} r_{h,i}(u_h, v). \tag{11}$$

This is accomplished naturally by a nonoverlapping partitioning of the mesh:

$$\mathcal{T}_h = \bigcup_{i=1}^{p} \mathcal{T}_{h,i}, \quad \mathcal{T}_{h,i} \cap \mathcal{T}_{h,j} = \emptyset \qquad \forall i \neq j.$$

The corresponding subdomains $\overline{\Omega}_i = \bigcup_{T \in \mathcal{T}_{h,i}} T$ are then nonoverlapping. The evaluation of the skeleton terms in the general residual form (4) requires some additional thought and can be either achieved through communication during evaluation of the residual or through the introduction of additional overlap solely for this purpose (ghost cells in DUNE's data decomposition model). The splitting of the residual form (11) naturally carries over to an additive splitting of the Jacobian matrix.

The residual form decomposition is the natural way to go with nonoverlapping domain decomposition methods (or Schur complement methods) such as balancing or FETI-DP [8]. For finite volume and discontinuous Galerkin methods the space decomposition and residual form decomposition approach lead to very similar methods.

References

[1] R. Alexander. Diagonally implicit Runge-Kutta methods for stiff O. D. E.'s. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

[2] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 1994.

[3] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, 2005.

[4] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations*. Cambridge University Press, 1996. `http://www.csc.kth.se/~jjan/private/cde.pdf`.

[5] A. Ern and J.-L. Guermond. *Theory and practice of finite element methods*. Springer, 2004.

[6] R. J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[7] C. W. Shu. Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.*, 9:1073, 1988.

[8] A. Toselli and O. Widlund. *Domain Decomposition Methods – Algorithms and Theory*. Springer, Berlin Heidelberg, 2005.

[9] Jinchao Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613, December 1992.