Scientific Computing WS 2020/2021

Slide lecture 7

Jürgen Fuhrmann

juergen.fuhrmann@wias-berlin.de

Finite volumes for nonlinear problems: motivation

Regard stationary second order PDE with Robin boundary conditions as a system of two first order equations in a Lipschitz domain Ω:

$$\nabla \cdot \vec{j} = f \qquad \text{continuity equation in } \Omega$$

$$\vec{j} = -D(u)\vec{\nabla} u \qquad \text{flux law in } \Omega$$

$$\vec{j} \cdot \vec{n} = \alpha u - g \qquad \text{on } \Gamma$$

- Derivation of the continuity equation was based on the consideration of species balances of an representative elementary volume (REV)
- Subdivide the computational domain into a finite number of REV's
  - Assign a value of u to each REV
  - Approximate $\vec{\nabla} u$ by finite differece of u values in neigboring REVs
  - ... call REVs "control volumes" or "finite volumes"

Discretization of continuity equation

- Stationary continuity equation: $\nabla \cdot \vec{j} = f$
- Integrate over control volume ω_k:

$$
\begin{aligned}
0 &= \int_{\omega_k} \nabla \cdot \vec{j}\, d\omega - \int_{\omega_k} f \; d\omega \\
&= \int_{\partial\omega_k} \vec{j} \cdot \vec{n}_\omega \; ds - \int_{\omega_k} f \; d\omega \\
&= \sum_{l \in \mathcal{N}_k} \int_{\sigma_{kl}} \vec{j} \cdot \vec{\nu}_{kl} \; ds + \sum_{m \in \mathcal{G}_k} \int_{\gamma_{km}} \vec{j} \cdot \vec{n}_m \; ds - \int_{\omega_k} f d\omega
\end{aligned}
$$

- This is exactly the same procedure as in the linear case

Approximation of flux between control volumes

Utilize flux law: $\vec{j} = -D(u)\vec{\nabla} u$

- Admissibility condition $\Rightarrow \vec{x}_k\vec{x}_l \parallel \nu_{kl}$
- Let $u_k = u(\vec{x}_k)$, $u_l = u(\vec{x}_l)$
- $h_{kl} = |\vec{x}_k - \vec{x}_l|$: distance between neigboring collocation points
- Finite difference approximation of normal derivative:

$$\vec{\nabla} u \cdot \vec{\nu}_{kl} \approx \frac{u_l - u_k}{h_{kl}}$$

$\Rightarrow$ flux between neigboring control volumes:

$$\int_{\sigma_{kl}} \vec{j} \cdot \vec{\nu}_{kl}\; ds = \frac{|\sigma_{kl}|}{h_{kl}} g(u_k, u_l)$$

where $g(\cdot, \cdot)$ is called flux function

Approximation of Robin boundary conditions

Utilize boundary condition $\vec{j} \cdot \vec{n} = \alpha u - g$

- Assume $\alpha|_{\Gamma_m} = \alpha_m$
- Approximation of $\vec{j} \cdot \vec{n}_m$ at the boundary of ω_k:

$$\vec{j} \cdot \vec{n}_m \approx \alpha_m u_k - g$$

- Approximation of flux from ω_k through Γ_m:

$$\int_{\gamma_{km}} \vec{j} \cdot \vec{n}_m \, ds \approx |\gamma_{km}|(\alpha_m u_k - g)$$

Approximation of right hand side

- $f_k = \frac{1}{|\omega_k|} \int_{\omega_k} f(\vec{x}) \; d\omega$
- Simple quadrature: $f_k = f(\vec{x}_k)$

Discrete system of equations

- The discrete system of equations then writes for $k \in \mathcal{N}$:

$$\sum_{l\in\mathcal{N}_k} \frac{|\sigma_{kl}|}{h_{kl}} g(u_k, u_l) + \sum_{m\in\mathcal{G}_k} |\gamma_{km}|\alpha_m u_k = |\omega_k| f_k + \sum_{m\in\mathcal{G}_k} |\gamma_{km}| g$$

- Flux approximation variants: $g(u_k, u_l) \approx D(u)(u_k - u_l)$
  - Averaging: $g(u_k, u_l) = D(\frac{u_k+u_l}{2})(u_k - u_l)$
  - Integrating: let $\mathcal{D}(u) = \int_{u_k}^{u} D(\xi)d\xi$.

$$\begin{aligned} g(u_k, u_l) &= \mathcal{D}(u_k) - \mathcal{D}(u_l) \\ &= D(\theta)(u_k - u_l) \text{ for some } \theta \in [u_k, u_l] \end{aligned}$$

For a 1D problem, the idea behind this variant is the solution of a two point boundary value problem along the grid edge x_k, x_l:

$$\begin{cases} D(u)u' &= g \\ u|_{x_k} &= u_k \\ u|_{x_l} &= u_l \end{cases}$$

This can be generalized to 2D and 3D problems.

Discrete system of equations II

- As a result, we have a nonlinear system of equations $A(u) = f$ with N unknowns and N equations.
- It's Jacobi matrix $A'(u)$ is needed in order to allow to implement Newton's method.
- $A'(u)$ is sparse, so we will be able to apply sparse linear solvers to solve the linear systems occuring during the solution
- M property of the Jacobi matrix is desirable and depends on the monotonicity property of $g(u_k, u_l)$: it needs to be monotonically increasing in the first argument, and monotonically decreasing in the second argument.

Nonlinear operator evaluation

- Given a vector u, calculate $A(u) - f$ based on information from the triangulation
- Necessary information:
  - List of point coordinates $\vec{x}_K$
  - List of triangles which for each triangle describes indices of points belonging to triangle
  - List of (boundary) segments which for each segment describes indices of points belonging to segment
- Assembly in two loops:
  - Loop over all triangles, calculate triangle contribution to nonlinear operator
  - Loop over all boundary segments, calculate boundary contributions
- Use Julia's dual number approach to assemble Jacobi matrix data at the same time. So in fact for given u we get the $r = A(u) - f$ and $A'(u)$ at once.

Generalization

The same approach can be generalized:

- Reaction-diffusion problems: $\nabla \cdot \vec{j} + r(u) = f$
- Time dependent (parabolic) problems: $\partial_t u(x,t) + \nabla \cdot \vec{j} = f$
- Convection-diffusion problems: $j = -D\nabla u + u\vec{v}$
- Systems of partial differential equations: $u(x) = (u_1(x) \dots u_m(x))$ describing the interaction of multiple species.
- In all these cases we have “building blocks” like $g(u_k, u_l)$, $r(u)$, which give rise to a programming interface allowing to describe rather sophisticated systems of partial differential equations
- `VoronoiFVM.jl` Julia package