Samoa

Oliver MEISTER [??]

Abstract.

Keywords.

1. Notation

The following notation conventions are used in this document:

- Variables, constants or fields:
  - a lowercase, thin letters are scalars
  - $\mathbf{a}$ lowercase, bold letters are vectors
  - $\mathbf{A}$ uppercase, bold letters are matrices
- Operators:
  - $\mathbf{a} \circ \mathbf{b}$ circles are dot products of two vectors or matrices
  - $\mathbf{A} \cdot \mathbf{b}$ center dots are either matrix-matrix, scalar-matrix or scalar-scalar products
  - $\dot{y}$ dots above function variables are the temporal derivative operator $\frac{\partial}{\partial t}$
  - ∇ nablas are spatial derivative operators, defined as $\nabla^T = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots \right)$
  - $\mathrm{div}(\mathbf{A})$ $\nabla^T \cdot \mathbf{A}$, the divergence of a vector field or matrix

2. Heat Equation

Let $\Omega \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$, be an open, bounded domain representing an arbitrary anisotropic metal plate (or chunk), u be the heat of a metal plate, c the heat conductivity of the metal plate and f an external heat source, where $u(\mathbf{x}, t), f(\mathbf{x}, t), c(\mathbf{x}) \in \mathbb{R}\ \forall \mathbf{x} \in \Omega\ \forall t \in \mathbb{R}^+$. Expressing the derivative of time $\dot{u}$ with respect to the spatial derivatives of u gives a *Partial Differential Equation* (PDE) for u with boundary conditions:

$$\dot{u} = \operatorname{div}(-c \cdot \nabla u) + f \qquad \text{in } \Omega \tag{1}$$

$$\nabla u \circ \vec{n} = g \qquad \text{in } \Gamma_N \subseteq \partial\Omega \tag{2}$$

$$u = h \qquad \text{in } \Gamma_D = \partial\Omega \setminus \Gamma_N \tag{3}$$

for $g : \Gamma_N \to \mathbb{R}$, $h : \Gamma_D \to \mathbb{R}$.

2.1. Algebraic solution

Directly solving the equation system cannot be done in general. However, for special choices of u, external heat source and boundary conditions, finding a solution is possible.

2.1.1. Collapsing initial condition

For instance, we assume $f = 0$, $c = -1$ and $u(\mathbf{x}, t) = s(t) \cdot u_0(\mathbf{x})$ for some nontrivial scalar functions s and u_0 with $s(0) = 1$. Inserting f and c, we can simplify the heat equation to the PDE $\dot{u} = \Delta u$. Next, one obtains $\dot{s}(t) \cdot u_0(\mathbf{x}) = s(t) \cdot \Delta u_0(\mathbf{x})$ or $\frac{\dot{s}(t)}{s(t)} = \frac{\Delta u_0(\mathbf{x})}{u_0(\mathbf{x})}$. For this equation to hold for each t and each $\mathbf{x}$, there must be a constant $a \in \mathbb{R}$ so that $\frac{\dot{s}(t)}{s(t)} = \frac{\Delta u_0(\mathbf{x})}{u_0(\mathbf{x})} = a$, since the left side of the equation is independent of space and the right side independent of time. As a result, $\dot{s} = a \cdot s$, $\Delta u_0 = a \cdot u_0$. Solving for s is straightforward: $s(t) = \exp(a \cdot t) \cdot s(0)$, so $u(\mathbf{x}, t) = \exp(a \cdot t) \cdot u_0(\mathbf{x})$. $\Delta u_0 = a \cdot u_0$ is a linear second-order PDE, which closely resembles linear ODEs for which the solution is known.
For instance, in a 2D case a solution for $\Omega := [0, 1]^2$ with $u = 0$ on $\partial\Omega$ can be obtained if $u_0(x, y) = \sin(\pi x) \cdot \sin(\pi y)$. It follows, that $\Delta u_0(x, y) = -2\pi^2 \cdot \sin(\pi x) \cdot \sin(\pi y)$, which means $a = -2\pi^2$. So $u(\mathbf{x}, t) = \exp(-2\pi^2 t) \cdot \sin(\pi x) \cdot \sin(\pi y)$.

2.2. Weak formulation

Special cases are of no use in solving the general case though, so one method to simplify the general problem is to solve a modified system given by the weak formulation of the PDE instead. A weak formulation for a given set of $n \in \mathbb{N}$ test functions $(\psi_j)_{j \in \{1,\dots,n\}}$, where $\psi_j : \Omega \to \mathbb{R}$, is obtained by the inner products of $\dot{u}$ with each ψ_j, $j \in \{1, \dots, n\}$ over Ω:

$$\int_\Omega \dot{u} \cdot \psi_j d\Omega = \int_\Omega (\operatorname{div}(-c \cdot \nabla u) + f) \cdot \psi_j d\Omega \tag{4}$$

Splitting the integral and applying the Divergence Theorem gives:

$$\int_\Omega \dot{u} \cdot \psi_j d\Omega = \int_\Omega \operatorname{div}(-c \cdot \nabla u) \cdot \psi_j d\Omega + \int_\Omega f \cdot \psi_j d\Omega \tag{5}$$

$$= \int_\Omega c \cdot (\nabla u \circ \nabla \psi_j) d\Omega - \int_{\partial\Omega} (c \cdot \nabla u \cdot \psi_j) \circ d\vec{S} + \int_\Omega f \cdot \psi_j d\Omega \tag{6}$$

$$\Rightarrow \int_\Omega \dot{u} \cdot \psi_j d\Omega = \int_\Omega c \cdot (\nabla u \circ \nabla \psi_j) d\Omega + \int_\Omega f \cdot \psi_j d\Omega - \int_{\Gamma_N} c \cdot g \cdot \psi_j d\Omega \tag{7}$$

The main difference here is, that while the strong form of the PDE requires the correctness of the solution in each point, a weak form only demands equality of integrals over the whole domain for a limited number of equations, depending on the choice of n. It is obvious that a solution of the strong formulation is also a solution of a weak formulation, the reverse implication is generally not true, however. Weak solutions even do not have to be strictly differentiable or continuous anymore, which in some cases allows a weak solution where a strong solution does not exist.

2.3. FEM discretization

Unfortunately, an algebraic solution of the weak form in equation (7) still is not possible in most cases, however there are means to approximate the solution. The Finite Element Method (FEM) suggests discretization of the temperature unknown u by a weighted sum over n basis functions $(\phi_i)_{i \in \{1,\dots,n\}}$, where $\phi_i : \Omega \to \mathbb{R}$.
So, assuming that $u = \sum_{i=1}^{n} u_i \cdot \phi_i$ for certain u_i with $u_i(t) \in \mathbb{R}\ \forall t \in \mathbb{R}^+, \forall i \in \{1, \dots, n\}$, it immediately follows that $\dot{u} := \sum_{i=1}^{n} \dot{u}_i \cdot \phi_i$ which, inserted into equation (7), gives for each $j \in \{1, \dots, n\}$:

$$\int_\Omega (\sum_{i=1}^{n} \dot{u}_i \cdot \phi_i) \cdot \psi_j d\Omega = \int_\Omega c \cdot (\nabla (\sum_{i=1}^{n} u_i \cdot \phi_i) \circ \nabla \psi_j) d\Omega + \int_\Omega f \cdot \psi_j d\Omega - \int_{\Gamma_N} c \cdot g \cdot \psi_j d\Omega \tag{8}$$

$$\Rightarrow \sum_{i=1}^{n} \dot{u}_i \cdot \left(\int_\Omega \phi_i \cdot \psi_j d\Omega \right) = \sum_{i=1}^{n} u_i \cdot \left(\int_\Omega c \cdot (\nabla \phi_i \circ \nabla \psi_j) d\Omega \right) + \int_\Omega f \cdot \psi_j d\Omega - \int_{\Gamma_N} c \cdot g \cdot \psi_j d\Omega \tag{9}$$

Since $j \in \{1, \dots, n\}$, equation (9) is actually a system of equations. It can be described more elegantly by a matrix notation:

$$\mathbf{M}\dot{\mathbf{u}} = \mathbf{A}\mathbf{u} + \mathbf{f} - \mathbf{b} \tag{10}$$

where the original terms have been replaced by the following variables:

- mass matrix $\mathbf{M} := \left(\int_\Omega \phi_i \cdot \psi_j d\Omega \right)_{i,j \in \{1,\dots,n\}} \in \mathbb{R}^{n \times n}$

- stiffness matrix $\mathbf{A} := \left(\int_\Omega c \cdot (\nabla \phi_i \circ \nabla \psi_j) d\Omega \right)_{i,j \in \{1,\dots,n\}} \in \mathbb{R}^{n \times n}$
- heat unknown vector $\mathbf{u} := (u_i)_{i \in \{1,\dots,n\}}$
- time derivative vector $\dot{\mathbf{u}} := (\dot{u}_i)_{i \in \{1,\dots,n\}}$
- external heat vector $\mathbf{f} := \left(\int_\Omega f \cdot \psi_j d\Omega \right)_{j \in \{1,\dots,n\}}$
- boundary vector $\mathbf{b} := \left(\int_{\Gamma_N} c \cdot g \cdot \psi_j d\Omega \right)_{j \in \{1,\dots,n\}} \in \mathbb{R}^n$

2.4. Time discretization

Again, a general solution is difficult, but only because the time-dependent external heat vector $\mathbf{f}$ is not known.
If we assume $\mathbf{f} = 0$, we have all we need to solve the equation. For that matter, we linearize the system by solving the time derivative $\mathbf{M}\ddot{\mathbf{u}} = \mathbf{A}\dot{\mathbf{u}}$ instead and setting $\mathbf{u} = \mathbf{A}^{-1}(\mathbf{M}\dot{\mathbf{u}} + \mathbf{b})$. The solution is $\dot{\mathbf{u}} = \exp(\mathbf{M}^{-1} \cdot \mathbf{A} \cdot t) \cdot \dot{\mathbf{u}}_0$, which means $\mathbf{u} = \mathbf{A}^{-1}\mathbf{M} \cdot \exp(\mathbf{M}^{-1} \cdot \mathbf{A} \cdot t) \cdot \mathbf{M}^{-1} \cdot (\mathbf{A} \cdot \mathbf{u}_0 - \mathbf{b}) + \mathbf{A}^{-1}\mathbf{b}$. This simplifies to:

$$\mathbf{u} = \exp(\mathbf{M}^{-1} \cdot \mathbf{A} \cdot t) \cdot (\mathbf{u}_0 - \mathbf{A}^{-1}\mathbf{b}) + \mathbf{A}^{-1}\mathbf{b} \tag{11}$$

If $\mathbf{f}$ is chosen arbitrarily, we have to discretize time as well. For example, one could use Euler's Method which approximates $\dot{\mathbf{u}}$ by either the forward difference $\dot{\mathbf{u}}(t) \approx \frac{1}{\delta t}(\mathbf{u}(t + \delta t) - \mathbf{u}(t))$ (explicit Euler) or the backward difference $\dot{\mathbf{u}}(t + \delta t) \approx \frac{1}{\delta t}(\mathbf{u}(t + \delta t) - \mathbf{u}(t))$ (implicit Euler). Solving for the update $\mathbf{u}(t + \delta t) - \mathbf{u}(t)$, we would obtain the following linear system in the explicit case:

$$\mathbf{M}(\mathbf{u}(t + \delta t) - \mathbf{u}(t)) = \delta t \cdot (\mathbf{A}\mathbf{u}(t) + \mathbf{f} - \mathbf{b}) \tag{12}$$

which must be modified only slightly for the implicit case:

$$(\mathbf{M} - \delta t \cdot \mathbf{A})(\mathbf{u}(t + \delta t) - \mathbf{u}(t)) = \delta t \cdot (\mathbf{A}\mathbf{u}(t) + \mathbf{f} - \mathbf{b}) \tag{13}$$

2.5. Mass matrix and stiffness matrix computation

2.5.1. Sparsity

Depending on the choice of the basis functions $(\phi_i)_{i \in \{1,\dots,n\}}$ and test functions $(\psi_j)_{j \in \{1,\dots,n\}}$, the mass matrix and stiffness matrix are likely to be sparse, since for any pair of $i, j \in \{1, \dots, n\}$, the matrix entries $\mathbf{M}_{i,j}$ and $\mathbf{A}_{i,j}$ can be nonzero only if $\phi_i \cdot \psi_j$ is nonzero. So if, for instance, all basis and test functions have a local support, then the matrices are nonzero only in entries where the supports of the respective basis and test function intercept.

2.5.2. *Reference functions*

Efficient computation of the matrix entries can be done if certain assumptions about the basis and test functions are made. Let there be a small (independent of n) set of $m \in \mathbb{N}$ "reference" basis functions $(\hat{\phi}_r)_{r\in\{1,\dots,m\}}$ and test functions $(\hat{\psi}_q)_{q\in\{1,\dots,m\}}$, and for each basis function ϕ_i and test function ϕ_j there is assumed to be a respective basis function $\hat{\phi}_r$ and test function $\hat{\psi}_q$, so that the following equations hold:

$$\phi_i(\mathbf{x}) = \hat{\phi}_r(\mathbf{t}^{-1}(\mathbf{x})) \qquad \forall \mathbf{x} \in \Omega_e \tag{14}$$

$$\psi_j(\mathbf{x}) = \hat{\psi}_q(\mathbf{t}^{-1}(\mathbf{x})) \qquad \forall \mathbf{x} \in \Omega_e \tag{15}$$

where $\mathbf{t}$ is an invertible transformation $\mathbf{t} : \Omega_0 \to \Omega_e$ with $\Omega_0 \subseteq \Omega$, $\Omega_e = \mathbf{t}(\Omega_0)$. The Jacobian of $\mathbf{t}$ is $\mathbf{J_t} : \Omega_0 \to \mathbb{R}^{d\times d}$, the Jacobian of $\mathbf{t}^{-1}$ is $\mathbf{J_{t^{-1}}} : \Omega_e \to \mathbb{R}^{d\times d}$. From simple analysis it's obvious, that $\mathbf{J_{t^{-1}}}(\mathbf{t}) = \mathbf{J_t}^{-1}$.
Computing the gradients yields:

$$\nabla\phi_i(\mathbf{x}) = \mathbf{J}^T_{\mathbf{t}^{-1}}(\mathbf{x}) \cdot \nabla\hat{\phi}_r(\mathbf{t}^{-1}(\mathbf{x})) = (\mathbf{J}_\mathbf{t}^{-T} \cdot \nabla\hat{\phi}_r)(\mathbf{t}^{-1}(\mathbf{x})) \qquad \forall \mathbf{x} \in \Omega_e$$

$$\nabla\psi_j(\mathbf{x}) = \mathbf{J}^T_{\mathbf{t}^{-1}}(\mathbf{x}) \cdot \nabla\hat{\psi}_q(\mathbf{t}^{-1}(\mathbf{x})) = (\mathbf{J}_\mathbf{t}^{-T} \cdot \nabla\hat{\psi}_q)(\mathbf{t}^{-1}(\mathbf{x})) \qquad \forall \mathbf{x} \in \Omega_e$$

2.5.3. *General transformation*

The partial (i, j)-th mass matrix entry can now be computed by:

$$\begin{aligned}
&\int_{\Omega_e} \phi_i \cdot \psi_j d\Omega = \int_{\Omega_e} (\phi_i \cdot \psi_j)(\mathbf{x}) d\mathbf{x} \\
&= \int_{\Omega_e} (\hat{\phi}_r \cdot \hat{\psi}_q)(\mathbf{t}^{-1}(\mathbf{x})) d\mathbf{x} \\
&= \int_{\mathbf{t}(\Omega_0)} \left|\det(\mathbf{J_t}(\mathbf{t}^{-1}(\mathbf{x})))\right| \cdot |\det(\mathbf{J_{t^{-1}}}(\mathbf{x}))| \cdot (\hat{\phi}_r \cdot \hat{\psi}_q)(\mathbf{t}^{-1}(\mathbf{x})) d\mathbf{x} \\
&= \int_{\Omega_0} |\det(\mathbf{J_t}(\mathbf{y}))| \cdot (\hat{\phi}_r \cdot \hat{\psi}_q)(\mathbf{y}) d\mathbf{y} \\
&= \int_{\Omega_0} |\det(\mathbf{J_t})| \cdot \hat{\phi}_r \cdot \hat{\psi}_q \, d\Omega
\end{aligned}$$

Similarly, the partial (i, j)-th stiffness matrix entry is:

$$\int_{\Omega_e} c \cdot (\nabla\phi_i \circ \nabla\psi_j) d\Omega = \int_{\Omega_e} c(\mathbf{x}) \cdot (\nabla\phi_i \circ \nabla\psi_j)(\mathbf{x}) d\mathbf{x}$$

$$= \int_{\Omega_e} c(\mathbf{x}) \cdot \left((\mathbf{J}_\mathbf{t}^{-T} \cdot \nabla\hat{\phi}_r)^T \cdot (\mathbf{J}_\mathbf{t}^{-T} \cdot \nabla\hat{\psi}_q) \right) \left(\mathbf{t}^{-1}(\mathbf{x}) \right) d\mathbf{x}$$

$$= \int_{\Omega_e} c(\mathbf{x}) \cdot \left(\nabla\hat{\phi}_r^T \cdot \mathbf{J}_\mathbf{t}^{-1} \cdot \mathbf{J}_\mathbf{t}^{-T} \cdot \nabla\hat{\psi}_q \right) \left(\mathbf{t}^{-1}(\mathbf{x}) \right) d\mathbf{x}$$

$$= \int_{\mathbf{t}(\Omega_0)} c(\mathbf{x}) \cdot \left| \det(\mathbf{J}_\mathbf{t}(\mathbf{t}^{-1}(\mathbf{x}))) \right| \cdot |\det(\mathbf{J}_{\mathbf{t}^{-1}}(\mathbf{x}))| \cdot \left(\nabla\hat{\phi}_r^T \cdot (\mathbf{J}_\mathbf{t}^T \cdot \mathbf{J}_\mathbf{t})^{-1} \cdot \nabla\hat{\psi}_q \right) \left(\mathbf{t}^{-1}(\mathbf{x}) \right) d\mathbf{x}$$

$$= \int_{\Omega_0} c(\mathbf{t}(\mathbf{y})) \cdot |\det(\mathbf{J}_\mathbf{t}(\mathbf{y}))| \cdot \left(\nabla\hat{\phi}_r^T \cdot (\mathbf{J}_\mathbf{t}^T \cdot \mathbf{J}_\mathbf{t})^{-1} \cdot \nabla\hat{\psi}_q \right) (\mathbf{y}) \, d\mathbf{y}$$

$$= \int_{\Omega_0} c(\mathbf{t}) \cdot |\det(\mathbf{J}_\mathbf{t})| \cdot \nabla\hat{\phi}_r^T \cdot (\mathbf{J}_\mathbf{t}^T \cdot \mathbf{J}_\mathbf{t})^{-1} \cdot \nabla\hat{\psi}_q d\Omega$$

Already, a big advantage of the substitution is noticeable: The integrals still have to be computed for each pair of n basis and test functions, but only the m reference functions and the transformation $\mathbf{t}$ with its Jacobian $\mathbf{J}_\mathbf{t}$ are actually required to do so. It's also obvious, that if further assumptions about $\mathbf{t}$ are made, one might even be able to completely reduce integral evaluations to the m reference functions.

2.5.4. Affine map

So instead of choosing $\mathbf{t}$ arbitrarily, it makes sense now to investigate how common choices of $\mathbf{t}$ affect the integrals:
An obvious help is a constant Jacobian $\mathbf{J}_\mathbf{t}$, which is given iff $\mathbf{t}$ is affine. This allows for a slight modification of the equations:

$$\int_{\Omega_e} \phi_i \cdot \psi_j d\Omega = |\det(\mathbf{J}_\mathbf{t})| \cdot \int_{\Omega_0} \hat{\phi}_r \cdot \hat{\psi}_q \, d\Omega$$

and

$$\int_{\Omega_e} c \cdot (\nabla\phi_i \circ \nabla\psi_j) d\Omega = |\det(\mathbf{J}_\mathbf{t})| \cdot \int_{\Omega_0} c(\mathbf{t}) \cdot \nabla\hat{\phi}_r^T \cdot (\mathbf{J}_\mathbf{t}^T \cdot \mathbf{J}_\mathbf{t})^{-1} \cdot \nabla\hat{\psi}_q d\Omega$$

Assuming c is constant in $\mathbf{t}(\Omega_0) = \Omega_e$ and using the identity $\mathbf{v}^T \cdot \mathbf{S} \cdot \mathbf{w} = \mathbf{S} \circ (\mathbf{v} \cdot \mathbf{w}^T)$ for any $\mathbf{v}, \mathbf{w} \in \mathbb{R}^d$, $\mathbf{S} \in \mathbb{R}^{d \times d}$, one obtains:

$$\int_{\Omega_e} c \cdot (\nabla\phi_i \circ \nabla\psi_j) d\Omega = c_e \cdot |\det(\mathbf{J}_\mathbf{t})| \cdot \left((\mathbf{J}_\mathbf{t}^T \cdot \mathbf{J}_\mathbf{t})^{-1} \circ \int_{\Omega_0} \nabla\hat{\phi}_r \cdot \nabla\hat{\psi}_q^T d\Omega \right)$$

Evaluation of all $\mathcal{O}(n^2)$ integrals thus requires the computation of $\mathcal{O}(m^2 \cdot d^2)$ integrals, a small number for lower order basis and test functions usually, that - most importantly - does not scale with n.

2.5.5. Conformal map

Another possible choice for $\mathbf{t}$ is a conformal (angle-preserving) map, which means $\mathbf{J_t} = h \cdot \mathbf{R}$ for $h : \mathbb{R}^d \to \mathbb{R}$ and $\mathbf{R} : \mathbb{R}^d \to \mathbb{R}^{d \times d}$, $\mathbf{R}^T \cdot \mathbf{R} = \mathbf{I}$. Some interesting properties of $\mathbf{t}$ result:

$$|\det(\mathbf{J_t})| = |\det(h\mathbf{R})| = h^d \, |\det(\mathbf{R})| = h^d \tag{16}$$

$$\mathbf{J}_\mathbf{t}^T \cdot \mathbf{J_t} = (h\mathbf{R})^T h\mathbf{R} = h^2 \mathbf{R}^T \mathbf{R} = h^2 \mathbf{I} \tag{17}$$

which allow to simplify the equations:

$$\int_{\Omega_e} \phi_i \cdot \psi_j d\Omega = \int_{\Omega_0} h^d \cdot \hat{\phi}_r \cdot \hat{\psi}_q d\Omega$$

and

$$\begin{aligned}
&\int_{\Omega_e} c \cdot (\nabla \phi_i \circ \nabla \psi_j) d\Omega \\
&= \int_{\Omega_0} c(\mathbf{t}) \cdot h^d \cdot (\nabla \hat{\phi}_r^T \cdot (h^2 \mathbf{I})^{-1} \cdot \nabla \hat{\psi}_q) d\Omega \\
&= \int_{\Omega_0} c(\mathbf{t}) \cdot h^{d-2} \cdot (\nabla \hat{\phi}_r \circ \nabla \hat{\psi}_q) d\Omega
\end{aligned}$$

Assuming c is piecewise constant, this could be the point to stop for static problems in 2D, as the mass matrix is not required there and the term h^{d-2} vanishes if $d = 2$.
In all other cases, it is also necessary to assume h to be constant to continue, which effectively makes $\mathbf{t}$ an affine conformal map. The terms can thus be updated one last time:

$$\int_{\Omega_e} \phi_i \cdot \psi_j d\Omega = h^d \cdot \int_{\Omega_0} \hat{\phi}_r \cdot \hat{\psi}_q d\Omega$$

$$\int_{\Omega_e} c \cdot (\nabla \phi_i \circ \nabla \psi_j) d\Omega = c_e \cdot h^{d-2} \cdot \int_{\Omega_0} (\nabla \hat{\phi}_r \circ \nabla \hat{\psi}_q) d\Omega$$

At this point, evaluation of all $\mathcal{O}(n^2)$ integrals requires the computation of no more than $\mathcal{O}(m^2)$ integrals.

2.5.6. Combination of maps

Since it is very common for a FEM discretization to have lots of similar basis functions, one might also think about combining a general map with an affine conformal map to obtain a small set of expensive integrals, which can be precomputed. The other integrals then require only a scaling by the factors h^d or h^{d-2} respectively, which is a cheap operation.

3. Shallow Water Equations

Let $\Omega \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$, be an open, bounded domain representing a water tank, h be the water height, $\mathbf{u}$ the wave velocity and $g \approx 9.81$ the gravitational constant, where $h(\mathbf{x}, t) \in \mathbb{R}$, $\mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^d \; \forall \mathbf{x} \in \Omega \; \forall t \in \mathbb{R}^+$.
Expressing the derivatives of time h_t and $\mathbf{u}_t$ with respect to the spatial derivatives of h and $\mathbf{u}$ gives a hyperbolic PDE system for h and $\mathbf{u}$:

$$\partial_t h + \operatorname{div}(h \cdot \mathbf{u}) = 0 \qquad \text{in } \Omega \tag{18}$$

$$\partial_t (h \cdot \mathbf{u}) + \operatorname{div}(h \cdot \mathbf{u} \cdot \mathbf{u}^T) + \frac{1}{2} \cdot \nabla (g \cdot h^2) = 0 \qquad \text{in } \Omega \tag{19}$$

With

$$\mathbf{q} := \left(h, h \cdot \mathbf{u}^T\right) \in \mathbb{R}^{1 \times (d+1)}$$

$$\mathbf{F}\left(\left(h, h \cdot \mathbf{u}^T\right)\right) := \left(h \cdot \mathbf{u}, h \cdot \mathbf{u} \cdot \mathbf{u}^T + \frac{1}{2} \cdot g \cdot h^2 \cdot \mathbf{I}\right) \in \mathbb{R}^{d \times (d+1)}$$

these equations can be reduced to a single multidimensional equation:

$$\dot{\mathbf{q}} + \operatorname{div}(\mathbf{F}(\mathbf{q})) = 0 \qquad \text{in } \Omega \tag{20}$$

3.1. Weak formulation

Let $\Omega_e \subseteq \Omega$ be an open subset of the domain. An element-wise weak formulation for a given set of $n \in \mathbb{N}$ test functions $(\psi_j)_{j \in \{1,\dots,n\}}$, where $\psi_j : \Omega_e \to \mathbb{R}$, for all $j \in \{1, \dots, n\}$, is obtained by the inner products of $\dot{\mathbf{q}}$ with each ψ_j over Ω_e:

$$\int_{\Omega_e} \dot{\mathbf{q}} \cdot \psi_j d\Omega = -\int_{\Omega_e} \operatorname{div}(\mathbf{F}(\mathbf{q})) \cdot \psi_j d\Omega \tag{21}$$

Applying the Divergence Theorem gives:

$$\int_{\Omega_e} \dot{\mathbf{q}} \cdot \psi_j d\Omega = \int_{\Omega_e} \nabla \psi_j^T \cdot \mathbf{F}(\mathbf{q}) d\Omega - \int_{\partial\Omega_e} \mathbf{F}(\mathbf{q}) \cdot \psi_j \cdot d\vec{S} \tag{22}$$

3.2. DG discretization

Let there be n basis functions $(\phi_i)_{i \in \{1,\dots,n\}}$, where $\phi_i : \Omega_e \to \mathbb{R}$.
Assuming that $\mathbf{q} = \sum_{i=1}^{n} \mathbf{q}_i \cdot \phi_i$ and $\mathbf{F}(\mathbf{q}) = \sum_{i=1}^{n} \mathbf{F}_i \cdot \phi_i$ in Ω_e with $\mathbf{q}_i(t) \in \mathbb{R}^{d+1}$, $\mathbf{F}_i(t) \in \mathbb{R}^{(d+1) \times d} \; \forall t \in \mathbb{R}^+ \; \forall i \in \{1, \dots, n\}$:

$$\int_{\Omega}(\sum_{i=1}^{n}\dot{\mathbf{q}}_i\cdot\phi_i)\cdot\psi_j d\Omega=\int_{\Omega_e}(\sum_{i=1}^{n}\mathbf{F}_i\cdot\phi_i)\cdot\nabla\psi_j d\Omega-\int_{\partial\Omega_e}((\sum_{i=1}^{n}\mathbf{F}_i^T\cdot\phi_i)\cdot\psi_j)\cdot d\vec{S} \tag{23}$$

$$\Rightarrow\sum_{i=1}^{n}\dot{\mathbf{q}}_i\cdot\int_{\Omega}\phi_i\cdot\psi_j d\Omega=\sum_{i=1}^{n}\mathbf{F}_i\cdot\int_{\Omega_e}\phi_i\cdot\nabla\psi_j d\Omega-\sum_{i=1}^{n}\mathbf{F}_i^T\cdot\int_{\partial\Omega_e}\phi_i\cdot\psi_j\cdot d\vec{S} \tag{24}$$

The respective tensor notation is:

$$\mathbf{M}\dot{\mathbf{q}}=\mathfrak{A}\mathbf{F}-\mathfrak{B}\mathbf{F}^T \tag{25}$$

4. Porous Media Flow

Transport equations for water and oil saturation S_w, S_n for two phase-flow:

$$\Phi \dot{S}_w + \mathrm{div}(-\lambda_w(S_w) K \nabla p) = 0 \tag{26}$$

$$\Phi \dot{S}_n + \mathrm{div}(-\lambda_n(S_n) K \nabla p) = 0 \tag{27}$$

The mobility λ_α is defined as $\lambda_\alpha := \frac{\kappa_\alpha}{\mu_\alpha}$ for the dynamic fluid viscosity μ_α and the relative permeability κ_α. With $S_w + S_n = 1$, set $S := S_w$, then $S_n = 1 - S$. Insertion and addition of both equations gives:

$$\Phi \dot{S} + \mathrm{div}(-\lambda_w(S) K \nabla p) = 0 \tag{28}$$

$$-\Phi \dot{S} + \mathrm{div}(-\lambda_n(1-S) K \nabla p) = 0 \tag{29}$$

$$\Rightarrow \mathrm{div}(-(\lambda_w(S) + \lambda_n(1-S)) K \nabla p) = 0 \tag{30}$$

With Darcy's law $\mathbf{u}_\alpha = -\lambda_\alpha K \nabla p$ and $\mathbf{u}_t := \mathbf{u}_w + \mathbf{u}_n$ this gives $\mathrm{div}(\mathbf{u}_t) = 0$.
A simple, nonlinear choice for the relative permeability terms would be [ref]:

$$\kappa_w(S) = S^2 \tag{31}$$

$$\kappa_n(1-S) = (1-S)^2 \tag{32}$$

The Brooks-Corey Model, which is motivated by capillary pressure, suggests instead:

$$\kappa_w(S) = S^2 \cdot S^{\frac{2}{\lambda}+1} \tag{33}$$

$$\kappa_n(1-S) = (1-S)^2 \cdot (1 - S^{\frac{2}{\lambda}+1}) \tag{34}$$

for an empirical constant $\lambda \in [0.2, 3.0]$.

5. Sierpinski grids

Definition 5.1 (Grid)**.** A *2D grid* $G = (V, E, C)$ is a plane graph $P = (V, E, F)$ with a set of *cells* $C \subseteq F$, such that each edge in E is adjacent to exactly two faces in F of which at least one is a cell.

Definition 5.2. We define the following terms:

- An *inner edge* is an edge that is adjacent to exactly two cells.
- A *boundary edge* is an edge that is adjacent to exactly one cell.
- An *inner vertex* is a vertex that is adjacent to inner edges only.
- A *boundary vertex* is a vertex that is adjacent to at least one boundary edge.

Let c be the number of cells in a grid. The number of inner edges is e_i, the number of boundary edges e_b. It follows, that $e = e_i + e_b$ is the total number of edges. Similarly, the number of inner vertices is v_i, the number of boundary vertices v_b and $v = v_i + v_b$ is the total number of vertices.

Lemma 5.1. *The following statements hold for any grid G:*

1. *Each boundary edge $e \in E$ in a grid $G = (V, E, C)$ is adjacent to exactly one face $f \in F \setminus C$.*
2. *A face $f \in F \setminus C$ cannot be adjacent to inner edges.*

Proof.
1. Since each edge is adjacent to exactly two faces and e is adjacent to exactly one cell, the second face cannot be a cell and must be in $F \setminus C$.
2. Each edge is adjacent to exactly two faces, for an inner edge both of them must be cells and cannot be in $F \setminus C$.

□

Lemma 5.2. *$v_b \leq e_b$ for any grid G.*

Proof. We show the inequality by proving first, that each boundary vertex is adjacent to at least two boundary edges:
Suppose in a grid $G = (V, E, C)$ there is a vertex $w \in V$, that is adjacent only to one edge $k \in E$. w can be adjacent only to a single face $f \in F$. The same is true for k, as k must be adjacent to a subset of the faces adjacent to w. However, since an edge in a grid cannot be adjacent only to one face, the supposition was wrong and each vertex is adjacent to at least two edges.
This means a boundary vertex $v \in V$ is adjacent to at least one boundary edge and at least one additional edge. A face adjacent to v must thus be adjacent to at least two edges, that are adjacent to v. Of these two edges, one must be an inner edge, since only one boundary edge is adjacent to v. According to Lemma 5.1, this is not possible.
In conclusion at least two boundary edges are adjacent to each boundary vertex.

If we count boundary vertices by counting both vertices adjacent to each boundary edge, we count each boundary vertex at least twice, since we know each boundary vertex is adjacent to at least two boundary edges. So $v_b \leq \frac{2e_b}{2} = e_b$. □

5.1. Triangular grids

Definition 5.3. *Triangular grids* are grids where each cell is adjacent to exactly 3 edges.

Theorem 5.3. *The number of edges in a triangular grid is* $e = \frac{3}{2}c + \frac{1}{2}e_b$.

Proof. Each cell is adjacent to exactly 3 edges. If we count each adjacent edge once per cell, all inner edges will be counted twice and all boundary edges once, since each inner edge is adjacent to two cells and each boundary edge adjacent to one cell. That means $2e_i + e_b = 3c$. With $e = e_i + e_b$, we get $2e = 3c + e_b$ or $e = \frac{3}{2}c + \frac{1}{2}e_b$. □

Corollary 5.4. *The number of inner edges in a triangular grid is* $e_i = \frac{3}{2}c - \frac{1}{2}e_b$.

This gives an upper bound for the number of inner edges in a grid, namely $e_i \leq \frac{3}{2}c$, since $e_b \geq 0$.

5.2. Full triangular grids

Definition 5.4 (Full triangular grid)**.** A full triangular grid is defined as a connected triangular grid where the number of faces in a corresponding planar graph $f = c + 1$.

Lemma 5.5. *In a full triangular grid* $v = \frac{1}{2}c + \frac{1}{2}e_b + 1$.

Proof. Euler's formula for connected planar graphs states that $f - e + v = 2$. Then $v = 2 + e - f = 2 + \frac{3}{2}c + \frac{1}{2}e_b - f = 2 + \frac{3}{2}c + \frac{1}{2}e_b - c - 1 = \frac{1}{2}c + \frac{1}{2}e_b + 1$. □

Corollary 5.6. *The number of inner vertices in a full triangular grid is* $v_i = \frac{1}{2}c + \frac{1}{2}e_b - v_b + 1$

5.3. Sierpinski grids

Conjecture 5.1. Let s_{maxl} be the maximum stack size for the left side (right side respectively). Then the following upper bounds hold: $e_{bl} \leq 2 * s_{maxl}$ and $v_{bl} \leq 2 * s_{maxl} - 1$

Conjecture 5.2. Additionally, the following upper bounds hold:

1. $e_b \leq c + 2$
2. $e_{cr} \leq c - 1$
3. $e_{co} \leq \frac{1}{2}c$
4. $v_i \leq \frac{1}{2}c + 1$

With $e_i = e_{cr} + e_{co}$ lower bounds can be derived:

Corollary 5.7. e_{cr} *and* e_{co} *are limited as follows:* $c - \frac{1}{2}e_b \leq e_{cr} \leq c - 1$ *and* $\frac{1}{2}c - \frac{1}{2}e_b + 1 \leq e_{co} \leq \frac{1}{2}c$

Definition 5.5 (Grid partition)**.** Let $G = (V, E, C)$ be a grid. There is a dual plane graph $\bar{G} = (C, E', V)$ of G. Let $N \subseteq \mathcal{P}(C)$ be a partition of C with connected subsets. We call N a *grid partition* of G.

Lemma 5.8. *Let N be a partition of a grid G and P the corresponding dual graph. Then P is well-defined and planar.*

Proof. Since $G = (V, E, C)$ is planar, an induced plane graph $G' = (V, E', F)$ exists with a dual plane graph $\bar{G} = (F, E', V)$. Obviously, $P = (C, E' \cap Cchoose2, V)$ is a subgraph of $\bar{G}$ It remains to be shown that $P = (C, E, V)$ is the dual of □

Lemma 5.9. *Let $P = (V, E, F)$ be a connected plane graph with $|F| > 1$, then $|E| \leq 3|V| - 6$.*

Proof. let a be the number of edge-face pairs in P. Then $a = 2|E|$ since two vertices are adjacent to an edge. Also, each face in P has more than two adjacent edges, so $a \geq 3|F|$. Together $2|E| \geq 3|F|$. Euler's Polyeder fomula for planar graphs states that $|F| + |V| - |E| = 2$. Then $\frac{2}{3}|E| + |V| - |E| \geq 2 \Rightarrow |E| \leq 3|V| - 6$. □

Lemma 5.10. *Let $P = (V, E, F)$ be a connected plane graph with $|F| > 1$, then $|F| \leq 2|V| - 4$.*

Proof. Euler's Polyeder fomula for plane graphs states that $|F| + |V| - |E| = 2$. Then $|F| = |E| - |V| + 2 \leq 3|V| - 6 - |V| + 2 = 2|V| - 4$. □

Lemma 5.11. *For any partition N of a grid G and its corresponding grid partition graph $P = (N, I, F, w)$ with limited w, the following upper bound holds:*

$$\sum_{i \in I} w(i) < 3|N| \sup(w)$$

Proof. Let N be a partition of G and $P = (N, I, F, w)$ its corresponding grid partition graph. From Lemma 5.11 it follows, that P is planar. That means we can apply Lemma 5.9, which states that $|I| \leq 3|N| - 6 < 3|N|$. Then $\sum_{i \in I} w(i) \leq \sum_{i \in I} \sup(w) < 3|N| \sup(w)$. □

If we define w as the number of shared edges between two partitions, this leads to the following theorem:

Theorem 5.12. *The number of communications between edges shared by two processes in a grid is limited by $3 \sup(w)|N|$.*

Additionally, the number of of communications between vertices shared by two subsets cannot be higher than for edges, and thus is also limited by $3 \sup(w)|N|$.

Lemma 5.13. *For any partition of a grid G with bounded degree and its corresponding grid partition graph $P = (N, I, F, w)$, the following upper bound holds:*

$$\sum_{f \in F} |\{\{x,y\} \in \binom{N}{2}; x \neq y, x \textit{ and } y \textit{ are adjacent to } f\}| < \triangle(G)^2|N|$$

Proof. The number of vertices in N which are adjacent to a face f in F must be limited by $\triangle(G)$, which is a direct consequence of the duality of $P = (N, I, F)$ to G. With $|F| < 2|N|$ (5.10), we get:

$$\begin{aligned}&\sum_{f \in F} |\{\{x,y\} \in \binom{N}{2}; x \neq y, x \text{ and } y \text{ are adjacent to } f\}| \\ \leq &\sum_{f \in F} \binom{\triangle(G)}{2} < 2|N| \binom{\triangle(G)}{2} < \triangle(G)^2|N|\end{aligned}$$

□

Lemma 5.13 states, that the number of communications between vertices shared by more than two subsets is limited by $\triangle(G)^2|N|$. Since the number of communications between vertices shared by two subsets is limited by $3\sup(w)|N|$ (Lemma 5.11), this leads to the following theorem:

Theorem 5.14. *The number of communications between vertices shared by two or more processes is limited by* $(3\sup(w) + \triangle(G)^2)|N|$.

5.4.1. Load Balancing

Definition 5.6 (load estimator)**.** Let G be a grid and $l : G \to \mathbb{R}$ be a function that estimates the computational load of the grid. We call l *load estimator*.

Lemma 5.15. *Let G be a full triangular grid and $l : G \to \mathbb{R}$ be a load estimator for G with $l(G) = xc + ye + zv + l_0, x, y, z, l_0 \in \mathbb{R}$. Then there are $x', y', z' \in \mathbb{R}$ with $l(G) = x'c + y'e_b + z'$.*

Proof. Since $e = \frac{3}{2}c + \frac{1}{2}e_b$ and $v = \frac{1}{2}c + \frac{1}{2}e_b + 1$:

$$\begin{aligned}l(G) =& xc + ye + zv + l_0 \\ =& xc + y(\frac{3}{2}c + \frac{1}{2}e_b) + z(\frac{1}{2}c + \frac{1}{2}e_b + 1) + l_0 \\ =& (x + \frac{3}{2}y + \frac{1}{2}z)c + (\frac{1}{2}y + \frac{1}{2}z)e_b + (z + l_0)\end{aligned}$$

Set $x' := x + \frac{3}{2}y + \frac{1}{2}z$, $y' := \frac{1}{2}y + \frac{1}{2}z$ and $z' := z + l_0$, then the lemma follows. □

This lemma states that in order to get an affine estimate of the load depending on the number of grid cells, edges and vertices, it is sufficient to use the number of cells and boundary edges instead, effectively reducing the number of degrees of freedom by one.

Lemma 5.16. *Let $x \in \mathbb{R}$ and $k \in \mathbb{N}$. Then the following statements hold:*

- $\lfloor x \rfloor \leq k \Leftrightarrow x < k + 1$

- $\lfloor x \rfloor > k \Leftrightarrow x \geq k+1$
- $\lfloor x \rfloor < k \Leftrightarrow x < k$
- $\lfloor x \rfloor \geq k \Leftrightarrow x \geq k$

Proof. □

Lemma 5.17 (Concurrent synchronization)**.** *Let there be a partition of a grid G on a processor divided into $n \in \mathbb{N}$ sections. Let c_i be the computation time and s_i the boundary send time for each section $i \in [n]$. Computation and sending of a section may be concurrent. Then the overall wall-clock time for computation and sending is* $t_n = \max\left(\sum_{i=1}^{n} c_i, \sum_{i=1}^{n} s_i\right)$.

Proof. While all sections are computed, all boundary data is sent. Thus, the overall wall clock time is the maximum of computation and send time. □

Lemma 5.18 (Staggered synchronization)**.** *Let there be a partition of a grid G on a processor divided into $n \in \mathbb{N}$ sections. Let c_i be the computation time and s_i the boundary send time for each section $i \in [n]$. Computation and sending of a section are mutually exlusive. Then the overall wall-clock time for computation and sending is* $t_n = \sum_{i=1}^{n} c_i + \sum_{i=1}^{n} s_i$.

Proof. All sections are computed and all boundary data is sent sequentially. Thus, the overall wall clock time is the sum of computation and send time. □

Lemma 5.19 (Pipelined synchronization)**.** *Let there be a partition of a grid G on a processor divided into $n \in \mathbb{N}$ sections. Let c_i be the computation time and s_i the boundary send time for each section $i \in [n]$. As soon as computation of a section is finished, the section starts asynchronuous sending which takes cpu time s_i. Concurrently, computation of the next section starts. Then the overall wall-clock time for computation and sending is* $t_n = \max_{k \in [n]}\left(\sum_{i=1}^{k} c_i + \sum_{i=k}^{n} s_i\right)$.

Proof. Let $C_0 := 0$, $S_0 := 0$ and for $k \in [n]$ let C_k be the wall clock time where computation of the k-th section has ended and S_k be the wall clock time where sending of the k-th section has ended. Since computation for section k starts iff all previous computations have finished, $C_{k+1} = C_k + c_{k+1}$. Synchronization for section k starts iff computation of section k has finished and all previous sending has ended. Thus $S_{k+1} = \max(C_{k+1}, S_k) + s_{k+1}$.
It is obvious that $C_k = \sum_{i=1}^{k} c_i$. Via induction, one can show that $S_k = \max_{j \in [k]}\left(\sum_{i=1}^{j} c_i + \sum_{i=j}^{k} s_i\right)$.
Since $S_n = \max(C_n, S_{n-1}) + s_n \geq C_n$, $t_n = S_n$ is the overall wall clock time. □