Coordinate-free numerics: all your variation points for free?

Variation points

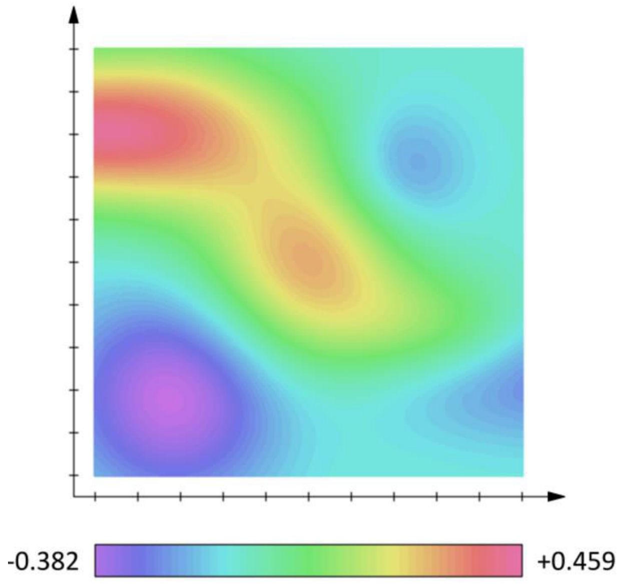
- Design for change/easily accomodated to change
- Different PDEs
- Discretisations
    - Numerical methods
    - Time stepping technique
    - Underlying grid
- Computational simplifications
    - Dimensions
    - Coordinate system
    - Symmetries
    - Problem size
    - Sequential vs parallel

Coordinate free numerics

- Algebraic specifications
- Core abstractions
  - Scalar field
  - Tensor field

Scalar-field level

- Boundary beetween continous level and discrete level
- Coordinate free
- Value at every point in space
- Operators:
  - (a + b)[p] = a[p] + b[p]
  - Partial differential operator (Rate of change along a direction)
  - Integration



Scalar-field level

- Discretisations (FDM, FEM, FVM)
- Example FDM
  - Data values sampled at regular intervals along each axis
  - Pointwise operations performed elementwise (a+b)[i,j] = a[i,j] + b[i,j]
  - Partial derivatives use neighbouring data.

- Problems with physical symmetries:
  - Represent 3D as 2D
  - 1000^3 -> 1000^2, Saving a factor of 1000

Tensor level

- Boundary between coordinate-free and coordinate-aware level
- Different ranks
- Built up of scalar-fields, vector-fields and covector-fields
- Used in formulation of coordinate-free PDE solvers
- Coordinate free operators
  - Tensor product: K1 and K2 rank -> K1 x K2
  - Tensors can be added together if same components and rank
  - Differentiation operations: Divergence, Gradient and Covariant derivatives

    $\nabla \cdot \mathbf{F}$ ∇f $\nabla_{\mathbf{v}} f$
  - Integration only on scalar tensor-fields

Tensor level

- Basic data structure for a tensor-field is a multi-indexed collection of scalar-fields together with vector/covector composition info
- Coordinate system coded into tensor implementation
- Linear operators are collective operators on component scalar-fields
- Derivation is partial derivation on component scalar-fields

PDE solver level

$$
\begin{aligned}
\rho \frac{\partial^2 \vec{u}}{\partial t^2} &= \nabla \cdot \sigma + \vec{f}(t), \\
\sigma &= \Lambda(e), \\
e &= \mathcal{L}_{\vec{u}}(g).
\end{aligned}
$$

Elastic wave equation

$$
\begin{aligned}
\rho_{11} \frac{\partial^2 \vec{u}}{\partial t^2} + \rho_{12} \frac{\partial^2 \vec{U}}{\partial t^2} &= \nabla \cdot \sigma + \vec{f}(t) + b \left(\frac{\partial \vec{U}}{\partial t} - \frac{\partial \vec{u}}{\partial t} \right) \\
\rho_{12} \frac{\partial^2 \vec{u}}{\partial t^2} + \rho_{22} \frac{\partial^2 \vec{U}}{\partial t^2} &= \nabla s + \vec{f}(t) - b \left(\frac{\partial \vec{U}}{\partial t} - \frac{\partial \vec{u}}{\partial t} \right), \\
\sigma &= \Lambda(e) + Q(\epsilon), \\
s &= M(e) + R(\epsilon), \\
e &= \mathcal{L}_{\vec{u}}(g), \\
\epsilon &= \nabla \cdot \vec{U}.
\end{aligned}
$$

Poro-elastic wave equation

- Can reprogram one to the other by changing 600-700 lines of code
- Full program 13500 lines

Mix and match

Mesh	D 2000				
	S 2000	P 2700			
Tn	D 1700				
	SFD 1600				
Bn	D 1500				
	S 1900	U 2100			
Tensor	SI 2200	STI 2200	STA 3000	CI 2200	CTI 2300
Seismod	SE 600	PE 700			

- Mesh – storing discrete data in arrays
- Tn – Scalar field
- Bn – Boundary handling
- Seismod – Time stepping seismic equation solver