The Discontinuous Galerkin method

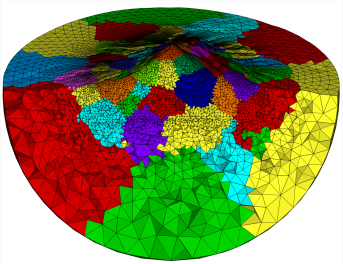
Heiner Igel

Department of Earth and Environmental Sciences
Ludwig-Maximilians-University Munich

Outline

Introduction

Motivation



- Efficiently solving the elastic wave equation on tetrahedral grids
- Easy implementation of local time stepping
- High accuracy of frictional boundary conditions for dynamic rupture problems

History

- Developed in the Los Alamos National Laboratories for the problem of neutron transport by Reed 1973 formulated on triangular meshes
- In the late eighties Cockburn connected discontinuous Galerkin method with high-order Runge-Kutta-type time integration schemes (Cockburn et al., 2000)
- First application to elastic wave propagation was published by Kaeser and Dumbser, 2006
- Kaeser et al., 2008 carried out a detailed analysis of the convergence properties of the discontinuous Galerkin method
- The local time-stepping approach by Dumbser et al., 2007b circumvents the problem of oversampling large parts of the model and thereby reducing the overall computations.
- De la Puente et al., 2009b reported a first analysis of scaling and synchronization of the discontinuous Galerkin method which raised the interest of the computational science community in Munich to optimize it.

History

- Developed in the Los Alamos National Laboratories for the problem of neutron transport by Reed 1973 formulated on triangular meshes
- In the late eighties Cockburn connected discontinuous Galerkin method with high-order Runge-Kutta-type time integration schemes (Cockburn et al., 2000)
- First application to elastic wave propagation was published by Kaeser and Dumbser, 2006
- Kaeser et al., 2008 carried out a detailed analysis of the convergence properties of the discontinuous Galerkin method
- The local time-stepping approach by Dumbser et al., 2007b circumvents the problem of oversampling large parts of the model and thereby reducing the overall computations.
- De la Puente et al., 2009b reported a first analysis of scaling and synchronization of the discontinuous Galerkin method which raised the interest of the computational science community in Munich to optimize it.

History

- Developed in the Los Alamos National Laboratories for the problem of neutron transport by Reed 1973 formulated on triangular meshes
- In the late eighties Cockburn connected discontinuous Galerkin method with high-order Runge-Kutta-type time integration schemes (Cockburn et al., 2000)
- First application to elastic wave propagation was published by Kaeser and Dumbser, 2006
- Kaeser et al., 2008 carried out a detailed analysis of the convergence properties of the discontinuous Galerkin method
- The local time-stepping approach by Dumbser et al., 2007b circumvents the problem of oversampling large parts of the model and thereby reducing the overall computations.
- De la Puente et al., 2009b reported a first analysis of scaling and synchronization of the discontinuous Galerkin method which raised the interest of the computational science community in Munich to optimize it.

History

- Developed in the Los Alamos National Laboratories for the problem of neutron transport by Reed 1973 formulated on triangular meshes
- In the late eighties Cockburn connected discontinuous Galerkin method with high-order Runge-Kutta-type time integration schemes (Cockburn et al., 2000)
- First application to elastic wave propagation was published by Kaeser and Dumbser, 2006
- Kaeser et al., 2008 carried out a detailed analysis of the convergence properties of the discontinuous Galerkin method
- The local time-stepping approach by Dumbser et al., 2007b circumvents the problem of oversampling large parts of the model and thereby reducing the overall computations.
- De la Puente et al., 2009b reported a first analysis of scaling and synchronization of the discontinuous Galerkin method which raised the interest of the computational science community in Munich to optimize it.

History

- Developed in the Los Alamos National Laboratories for the problem of neutron transport by Reed 1973 formulated on triangular meshes
- In the late eighties Cockburn connected discontinuous Galerkin method with high-order Runge-Kutta-type time integration schemes (Cockburn et al., 2000)
- First application to elastic wave propagation was published by Kaeser and Dumbser, 2006
- Kaeser et al., 2008 carried out a detailed analysis of the convergence properties of the discontinuous Galerkin method
- The local time-stepping approach by Dumbser et al., 2007b circumvents the problem of oversampling large parts of the model and thereby reducing the overall computations.
- De la Puente et al., 2009b reported a first analysis of scaling and synchronization of the discontinuous Galerkin method which raised the interest of the computational science community in Munich to optimize it.

History

- Developed in the Los Alamos National Laboratories for the problem of neutron transport by Reed 1973 formulated on triangular meshes
- In the late eighties Cockburn connected discontinuous Galerkin method with high-order Runge-Kutta-type time integration schemes (Cockburn et al., 2000)
- First application to elastic wave propagation was published by Kaeser and Dumbser, 2006
- Kaeser et al., 2008 carried out a detailed analysis of the convergence properties of the discontinuous Galerkin method
- The local time-stepping approach by Dumbser et al., 2007b circumvents the problem of oversampling large parts of the model and thereby reducing the overall computations.
- De la Puente et al., 2009b reported a first analysis of scaling and synchronization of the discontinuous Galerkin method which raised the interest of the computational science community in Munich to optimize it.

The Discontinuous Galerkin Method in a Nutshell

All concepts that have been discussed previously enter in this method

1. Finite-difference extrapolation using e.g. the Euler scheme
2. Calculation of element-based stiffness and mass matrices
3. Flux calculations at the element boundaries as encountered in the finite-volume method
4. Exact nodal interpolation based on Lagrange polynomials
5. Numerical integration schemes using collocation points

Wave Equation

1st order wave equation

$$\begin{aligned} \rho \partial_t v &= \partial_x \sigma + f \\ \partial_t \sigma &= \mu \partial_x v \end{aligned}$$

Matrix-vector form

$$\partial_t Q + A \partial_x Q = f$$

Wave Equation

where $\sigma = \sigma_{xy} = \sigma_{yx}$ representing the only non-zero stress component, and implicitly assuming space-time dependencies, $Q = (v, \sigma)$ is the vector of unknowns and A contains the coefficients of the equation given by

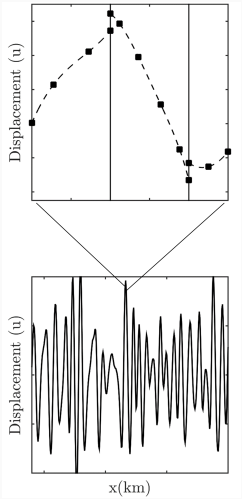
$$A = \begin{pmatrix} 0 & -1/\rho \\ -\mu & 0 \end{pmatrix}.$$

$\Longrightarrow$ Linear hyperbolic system, having the same form as the classic advection equation

The Discontinuous Galerkin Method in a Nutshell

- The Ansatz is to multiply the equation by an arbitrary test function combined with describing the unknown fields with the same set of basis functions (**Galerkin** principle).
- At the element boundaries the unknown fields can be **discontinuous**.
- Communication between the elements is achieved through a flux scheme based on solutions of the Riemann problem.

The Discontinuous Galerkin Method in a Nutshell



The wavefield inside each element is described by Lagrange polynomials exactly interpolating at appropriate collocation points. At each time step, a flux term F has to be evaluated at all element boundaries. The extrapolation scheme of the discontinuous Galerkin method can be expressed as

$$\partial_t Q^k(t) \;=\; (M^k)^{-1}(K^k Q^k(t) - (F_l^k - F_r^k)Q^k(t))$$

where $Q(t)$ is the vector of unknowns, M and K are the elemental mass and stiffness matrices, respectively, and $F_{r,l}$ are the flux terms at the left and right boundaries. The upper index k denotes the element (source term is omitted).

The Discontinuous Galerkin Method in a Nutshell

Implications of a boundary flux and no global system of equations to solve:

1. Obtaining a fully explicit scheme which lends itself to an **element-based parallelization** scheme
2. Choice of element size is arbitrary, it has no impact on the solution algorithm (**h-adaptivity**)
3. The polynomial order in each element can be arbitrarily chosen and no impact on the algorithm (**p-adaptivity**)
4. Considering the boundary points twice to calculate the fluxes implies an **increase in the number of degrees of freedom** that of course gets worse with increasing dimensionality

Ingredients

Source-free version of the coupled elastic wave equation in 1D

$$\begin{aligned} \partial_t \sigma - \mu \partial_x v &= 0 \\ \partial_t v - \frac{1}{\rho} \partial_x \sigma &= 0 \end{aligned}$$

Matrix notation

$$\partial_t Q + A \partial_x Q = 0$$

where $Q = (\sigma, v)$ is the vector of unknowns and matrix A contains the parameters

$$\mathbf{A} = \begin{pmatrix} 0 & -\mu \\ -\frac{1}{\rho} & 0 \end{pmatrix}$$

The Wave equation as Hyperbolic System

In the case of a quadratic matrix A with shape $m \times m$, this leads to an eigenvalue problem. If we were able to obtain eigenvalues λ_p such that

$$\mathbf{A}\mathbf{x}_p = \lambda_p \mathbf{x}_p \ , \quad p = 1, ..., m$$

we get a diagonal matrix of eigenvalues

$$\Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{pmatrix}$$

and the corresponding matrix $\mathbf{R}$ containing the eigenvectors $\mathbf{x_p}$ in each column

$$\mathbf{R} = (\mathbf{x_1}|\mathbf{x_2}|\ldots|\mathbf{x_p})$$

The Wave equation as Hyperbolic System

The Jacobian matrix $\mathbf{A}$ can now be expressed with the definitions

$$\mathbf{A} = \mathbf{R}\Lambda\mathbf{R}^{-1}$$
$$\Lambda = \mathbf{R}^{-1}\mathbf{A}\mathbf{R} \, .$$

Applying these definitions to the wave equation

$$\mathbf{R}^{-1}\partial_t Q + \mathbf{R}^{-1}\mathbf{R}\Lambda\mathbf{R}^{-1}\partial_x Q = 0$$

and introducing the solution vector $\mathbf{W} = \mathbf{R}^{-1}\mathbf{Q}$ results in

$$\partial_t \mathbf{W} + \Lambda\partial_x \mathbf{W} = 0 \, .$$

The Wave equation as Hyperbolic System

With $\lambda_{1,2} = \sqrt{\mu/\rho} = \pm c$, where c corresponds to the shear velocity, the eigenvectors can be obtained

$$\mathbf{r}_{1,2} = \begin{pmatrix} \pm\rho c \\ 1 \end{pmatrix}$$

interestingly enough containing as elements values of the *seismic impedance* $Z = \rho c$ that are relevant for the reflection behaviour of seismic waves. Thus the matrix $\mathbf{R}$ (and its inverse) are

$$\mathbf{R} = \begin{pmatrix} Z & -Z \\ 1 & 1 \end{pmatrix}, \; \mathbf{R}^{-1} = \frac{1}{2Z} \begin{pmatrix} 1 & Z \\ -1 & Z \end{pmatrix} .$$

The Wave equation as Hyperbolic System

The wave equation in the rotated eigensystem can be stated as

$$\partial_t \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \begin{pmatrix} -c & 0 \\ 0 & c \end{pmatrix} \partial_x \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = 0$$

with the simple general solution $w_{1,2} = w_{1,2}^{(0)}(x \pm ct)$, where the upper index 0 stands for the initial condition (i.e., waveform that is advected). The initial condition also fullfills $\mathbf{W}^{(0)} = \mathbf{R}^{-1}\mathbf{Q}^{(0)}$, we can therefore relate the so-called characteristic variables $w_{1,2}$ to the initial conditions of the physical variables as

$$\begin{aligned} w_1(x,t) &= \frac{1}{2Z}(\sigma^{(0)}(x+ct) + Zv^{(0)}(x+ct)) \\ w_2(x,t) &= \frac{1}{2Z}(-\sigma^{(0)}(x-ct) + Zv^{(0)}(x-ct)) \end{aligned}$$

The Wave equation as Hyperbolic System

To obtain the final analytical solution for velocity v and stress σ using $\mathbf{Q} = \mathbf{RW}$ as

$$\begin{aligned}\sigma(x,t) &= \frac{1}{2}(\sigma^{(0)}(x+ct) + \sigma^{(0)}(x-ct)) \\ &\quad + \frac{Z}{2}(v^{(0)}(x+ct) - v^{(0)}(x-ct)) \\ v(x,t) &= \frac{1}{2Z}(\sigma^{(0)}(x+ct) - \sigma^{(0)}(x-ct)) \\ &\quad + \frac{1}{2}(v^{(0)}(x+ct) + v^{(0)}(x-ct))\,.\end{aligned}$$

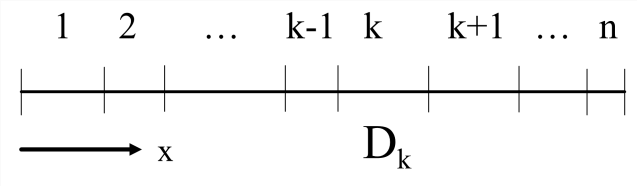
In compact form, the solution can be expressed as

$$\mathbf{Q}(x,t) = \sum_{p=1}^{m} w_p(x,t) r_p$$

Solution to Scalar Advection Equation

Denoting $q(x,t)$ for the unknown *scalar* solution and a for the given (possibly space dependent) advection (wave) velocity to obtain the source-free wave equation

$$\partial_t q(x,t) + a\,\partial_x q(x,t) = 0$$



The physical domain of each element is denoted by D_k with left and right boundaries x_l^k and x_r^k, respectively. Note the varying element sizes (h-adaptivity).

Solution to Scalar Advection Equation

The k elements are **non-overlapping** which implies a higher number of degrees of freedom. The solution vector in 1D is

$$q_{N_p} = \begin{pmatrix} q_1^1 & q_1^2 & \dots & q_1^n \\ q_2^1 & q_2^2 & \dots & q_2^n \\ \vdots & \vdots & \ddots & \vdots \\ q_{N_p}^1 & q_{N_p}^2 & \dots & q_{N_p}^n \end{pmatrix}$$

where N_p is the number of points per element and n the overall number of elements. The collocation points are Gauss-Lobatto-Legendre stored as

$$x_{N_p} = \begin{pmatrix} x_1^1 & x_1^2 = x_{N_p}^1 & \dots & x_1^n \\ x_2^1 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_p}^1 = x_1^2 & x_{N_p}^2 & \dots & x_{N_p}^n \end{pmatrix}.$$

Solution to Scalar Advection Equation

The global solution requires the adoption of flux concepts. Therefore a special sign is introduced that links the element-level to the global solution as

$$q(x,t) \approx q_h(x,t) = \bigoplus_{k=1}^{n} q_h^k(x,t)$$

where $\bigoplus$ indicates global assembly, n is the overall number of elements and h is the size of element k.

Weak formulation

To obtain the weak formulation of the scalar advection equation we multiply it

$$\partial_t q(x,t) + a\,\partial_x q(x,t) = 0$$

with a general test function $\phi_j(x)$, integrate over the k-th element domain D_k to obtain

$$\int_{D_k} \partial_t q(x,t)\phi_j(x)dx + \int_{D_k} a\partial_x q(x,t)\phi_j(x)dx = 0$$

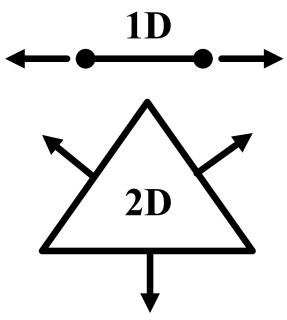
Weak formulation

And partially integrate replacing the term containing the space derivative by

$$\int_{D_k} a\partial_x q(x,t)\phi_j(x)dx = [aq(x,t)\phi_j(x)]_{x_l}^{x_r} - \int_{D_k} aq(x,t)\partial_x\phi_j(x)dx$$

where x_r and x_l are the right and left boundaries of element k, and we assume constant velocity a inside the element.

Weak formulation



What does the rule of partial integration look like with more than one dimension? The definition is known as

$$\int_\Omega \partial_{x_i} uv d\Omega = \int_\Gamma uvn_i d\Gamma - \int_\Omega u\partial_{x_i} v d\Omega$$

where u, v are arbitrary space-dependent functions, Ω denotes the entire volume, Γ its boundary, and n_i is a vector normal to the boundary.

Weak formulation

Putting this into the advection equation we obtain for element k

$$\int_{D_k} \partial_t q(x,t)\phi_j(x)dx - \int_{D_k} aq(x,t)\partial_x \phi_j(x)dx = -\left[aq(x,t)\phi_j(x)\right]_{x_l}^{x_r}$$

Replacing the unknown field $q(x,t)$ by a finite polynomial representation in terms of a weighted sum over Lagrange polynomials inside each element k of order N denoted as $\ell_i(x), i = 1,\ldots,N+1$ defined in the interval $x \in [-1,1]$.

Weak formulation

With the exact interpolation property of the Lagrange polynomials at the Gauss-Legendre-Lobatto collocation points x_i, here illustrated for an arbitrary function $y_i = y(x_i)$

$$y_i = \sum_{j=1}^{N_p} y_j \ell_j(x_i) = \sum_{j=1}^{N_p} y_j \delta_{ij}$$

we obtain for element k

$$q(x,t)|_{x=x_i} = \sum_{i=1}^{N_p} q_i(t)\ \ell_i(x)$$

where N_p indicates that in each element the polynomial order may vary, $x = x_i$ are the collocation points.

Weak formulation

In addition we also replace the test function by Lagrange polynomials. Combining this we yield

$$\int_{D_k} \partial_t \sum_{i=1}^{N_p} q_i(t)\ell_i(x)\ell_j(x)dx - \int_{D_k} a \sum_{i=1}^{N_p} q_i(t)\ell_i(x)\partial_x\ell_j(x)dx$$

and after re-ordering (omitting space dependencies of polynomials)

$$\sum_{i=1}^{N_p} \left[\left[\int_{D_k} \ell_i(x)\ell_j(x)dx \right] \partial_t q_i(t) - \left[\int_{D_k} a\ell_i(x)\partial_x\ell_j(x)dx \right] q_i(t) \right]$$

Weak formulation

Recognizing the elemental mass M_{ij} and stiffness K_{ij} matrices in this equation. Assuming implicit matrix-vector operations we obtain

$$M\ \partial_t q(t)\ =\ K^T\ q(t)$$

where the matrices are given by

$$M_{ij}\ =\ \int_{D_k} \ell_i(x)\ell_j(x)dx$$
$$K_{ij}\ =\ \int_{D_k} a\ell_i(x)\partial_x\ell_j(x)dx$$

assuming constant velocity a inside element k. Note that the lower index D_k indicates that we are still in physical space and we have to map to the local coordinate system.

Elemental Mass and Stiffness Matrices

The matrices (nodal or modal approach) for arbitrary test functions $\phi_i(x)$ are defined by

$$M_{ij} = \int_{D_k} \phi_i(x)\phi_j(x)\, dx$$

$$K_{ij} = a \int_{D_k} \phi_i(x)\partial_x\phi_j(x)\, dx$$

containing integrals over (derivatives of) the test functions $\phi(x)$. Replacing the integrals by a weighted sum over the function values $f(x_i)$ at carefully chosen points x_i inside the elements

$$\int_\Omega f(x)\, dx \approx \sum_{i=1}^{N} w_i\, f(x_i)$$

Elemental Mass and Stiffness Matrices

We need to map our physical coordinates into an element-based system. In 1D this is quite simple using as local variable ξ and transforming via

$$x^k(\xi) = x_l^k + \frac{(1+\xi)}{2} dx \quad \xi \in [-1, 1]$$

where x_l^k and x_r^k are the left and right physical boundaries of element k. In general the mapping of the differential used to evaluate integrals is called the Jacobian defined for element k as

$$J^k = \frac{dx}{d\xi}, \quad J^k = \frac{x_r^k - x_l^k}{1 - (-1)} = \frac{h^k}{2}$$

where h^k is the size of element k.

Elemental Mass and Stiffness Matrices

For the elemental matrices we obtain for general test functions

$$
\begin{aligned}
M_{ij} &= \int_{D_k} \phi_i(\xi)\phi_j(\xi)J^k \, d\xi \\
K_{ij} &= a \int_{D_k} \phi_i(\xi)\partial_\xi((J^k)^{-1})J^k\phi_j(\xi) \, d\xi \\
&= a \int_{D_k} \phi_i(\xi)\partial_\xi\phi_j(\xi) d\xi
\end{aligned}
$$

Elemental Mass and Stiffness Matrices

Finally, we can replace the test function with the Lagrange polynomials of order N_p leading to the definition of the mass and stiffness matrices

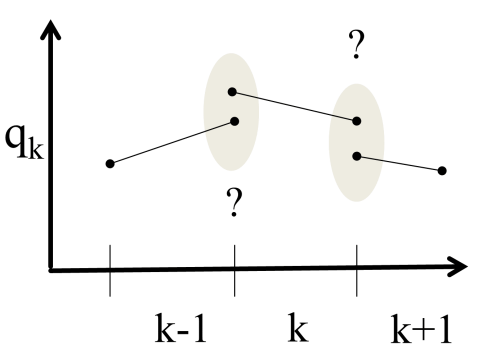
$$
\begin{aligned}
M_{ij} = \int_{-1}^{1} \ell_i(\xi)\ell_j(\xi)\, J^k \, d\xi &= \sum_{m=1}^{N_p} w_m \, \ell_i(x_m)\ell_j(x_m)\, J^k \\
&= \sum_{m=1}^{N_p} w_m \delta_{im} \, \delta_{jm} \, J^k \\
&= \begin{cases} w_i \, J^k & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}
\end{aligned}
$$

Elemental Mass and Stiffness Matrices

$$
\begin{aligned}
K_{ij} = \int_{-1}^{1} \ell_i(\xi)\partial_x \ell_j(\xi)\, d\xi &= \sum_{m=1}^{N_p} a\, w_m\, \ell_i(x_m)\partial_x \ell_j(x_m) \\
&= \sum_{m=1}^{N_p} a\, w_m \delta_{im} \partial_x \ell_j(x_m) \\
&= a\, w_i \partial_x \ell_j(x_i)\ .
\end{aligned}
$$

The Flux Scheme

The key task in the discontinuous Galerkin scheme concerns the question what values to allocate to the points at the element boundaries. This involves the use of flux schemes.



The Flux Scheme

Starting with the r.h.s. of this equation

$$\int_{D_k} \partial_t q(x,t)\phi_j(x)dx - \int_{D_k} aq(x,t)\partial_x\phi_j(x)dx = -\left[aq(x,t)\phi_j(x)\right]_{x_l}^{x_r}$$

The originial form holds also for higher dimensions

$$\int_{\partial D_k} a\, q(x,t)\phi_j(x)\ \mathbf{n}\ dx$$

where $\mathbf{n} = \pm 1$ denotes the vector normal to the boundary, in 1D taking the values $n = -1$ and $n = 1$ at the left and right boundaries.

The Flux Scheme

Replacing the space-dependent part of $q(x,t)$ by a sum over Lagrange polynomials to obtain

$$\sum_{i=1}^{N_p} \int_{\partial D_k} \ell_j(x)\, \ell_i(x)\, a\, q_i(t)\, \mathbf{n}\, dx$$

and also replacing the arbitrary test function with the same function. The orthogonality of the Lagrange polynomials and the fact that we are integrating over the surface ∂D_k leads to

$$\sum_{i=1}^{N_p} (\ell_i(x_r^k)\, \ell_j(x_r^k)(a\, q(x_r^k))^* - \ell_i(x_l^k)\, \ell_j(x_l^k)(a\, q(x_l^k))^*)$$

$$= \ell_j(x_r^k)(a\, q(x_r^k))^* - \ell_j(x_l^k)(a\, q(x_l^k))^*$$

where we introduced the *starred* terms $(a\, q(x_{r,l}^k))^*$ that are the values at the element boundaries.

The Flux Scheme

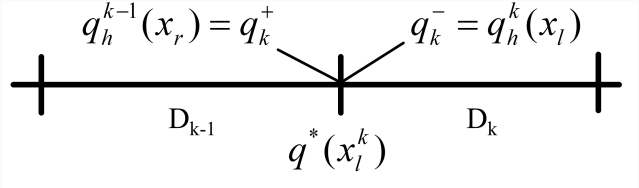
Therefore, we introduce the general flux vector with elements $F_j(a, q^k(x,t)), j = 1, .., N_p$ as

$$F_j(a, q^k(x,t)) \;=\; [\ell_j(x)\ (a\ q(x,t))^*]_{x_l^k}^{x_r^k}$$

In vector form

$$\mathbf{F} = \begin{pmatrix} F_1 \\ 0 \\ \vdots \\ 0 \\ F_{N_p} \end{pmatrix} = \begin{pmatrix} -(a\ q(x,t))^*(x_l^k) \\ 0 \\ \vdots \\ 0 \\ (a\ q(x,t))^*(x_r^k) \end{pmatrix}.$$

The Flux Scheme



At the left boundary the outside value in element $k-1$ is denoted with "'+"' and the value inside element k with "'-"'. The starred value $q*(x_l^k)$ is the flux that needs to be defined as a function of the boundary values of the adjacent elements.

The Flux Scheme

Taking the average of the values from both sides of the boundaries - also called the *central flux* - can be expressed as

$$F_1^c = \frac{1}{2}a^k(q(x_r^{k-1}, t) + q(x_l^k, t))$$
$$F_{N_p}^c = \frac{1}{2}a^k(q(x_r^k, t) + q(x_l^{k+1}, t))$$

The simplemost, stable choice is the so called *upwind flux*

$$F_1^{up} = \begin{cases} a^k \; q(x_l^k) & \text{if} \quad a^k \leq 0 \\ a^{k-1} \; q(x_r^{k-1}) & \text{if} \quad a^{k-1} > 0 \end{cases}$$

$$F_{N_p}^{up} = \begin{cases} a^{k+1} \; q(x_l^{k+1}) & \text{if} \quad a^{k+1} \leq 0 \\ a^k \; q(x_r^k) & \text{if} \quad a^k > 0 \end{cases}$$

The Flux Scheme

Centered and upwind fluxes can be formulated in a compact way. This formulation reads

$$F_1 = -a\frac{1}{2}(q(x_l^k) + q(x_r^{k-1})) - \frac{|a|}{2}(1-\alpha)(q(x_r^{k-1}) - q(x_l^k))$$
$$F_{N_p} = a\frac{1}{2}(q(x_r^k) + q(x_l^{k+1})) + \frac{|a|}{2}(1-\alpha)(q(x_r^k) - q(x_l^{k+1}))$$

where $\alpha = 0$ corresponds to the upwind flux and $\alpha = 1$ to the centered flux scheme.

The Discontinuous Galerkin Method Put To Action

The Discontinuous Galerkin Method Put To Action

In matrix notation we yielded for one element

$$M\partial_t q(t) \; - \; K^T q(t) \; = \; -F(a, q(t))$$

requiring an extrapolation scheme of the form

$$\partial_t q(t) \; = \; M^{-1}(K^T q(t) \; - F(a, q(t)))$$

where $F(a, q(t))$ is the flux vector as defined above. We seek to extrapolate the system from some initial conditions and obtain using the simple Euler method for each element

$$q(t_{n+1}) \; \approx \; q(t_n) \; + dt \left[M^{-1}(K^T q(t_n) \; - \; F(a, q(t)) \right]$$

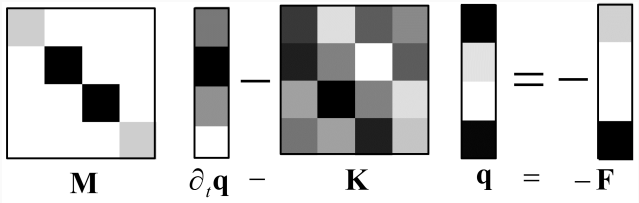
where for the flux scheme F() we use the upwind approach.

The Discontinuous Galerkin Method Put To Action

Employing a high-order extrapolation procedure known as *predictor-corrector* method (or Heun's method, or two-stage Runge-Kutta method). At time step t_i using time increment dt

$$
\begin{aligned}
k_1 &= f(t_i, y_i) \\
k_2 &= f(t_i + dt, y_i + dtk_1) \\
y_{i+1} &= y_i + \frac{1}{2} dt(k_1 + k_2)
\end{aligned}
$$

The Discontinuous Galerkin Method Put To Action



The matrix-vector form of the discontinuous Galerkin method. The system of questions at an elemental level is illustrated by plotting the absolute matrix/vector values. N = 3, N_p = N + 1 = 4

The Discontinuous Galerkin Method Put To Action

```
% Initialize vectors, matrices
Minv = zeros(N+1,N+1,ne);
K = zeros(N+1,N+1,ne);
q = zeros(N+1,ne);
(...)
for k=1:ne,
for i=1:N+1,
Minv(i,i,k)=1./(w(i)*J(k);
end
end
(...)
for k=1:ne,
for i=1:N+1,
for j=1:N+1,
K(i,j,k)=a(k)*w(i)*l1d(j,i);
end
end
end
```

The most important initialization step is the calculation of the elemental matrices, mass M and stiffness K. The following code part illustrates a possible implementation looping through all elements *ne* and calculating these matrices as a function of the Jacobian $J(k) = h^k/2$ where h^k is the size of element k:

The Discontinuous Galerkin Method Put To Action

```
% Flux calculation
F = zeros(N+1,ne);
(...)
for k=1:ne,
F(1,k)= -a*(q(1,k)+q(N+1,k-1))/2 - abs(a)*(1-alpha)/2*(q(N+1,k-1)-q(1,k));
F(N+1,k)= a*(q(N+1,k)+q(1,k+1))/2 + abs(a)*(1-alpha)/2*(q(N+1,k)-q(1,k+1));
end
```

The flux vector has to be calculated new for each time step (or intermediate step when using high-order extrapolation schemes) as it depends on the current values of the wavefield q. *alpha* can be used to change the flux scheme from central ($alpha = 1$) to upwind ($alpha = 0$).

The Discontinuous Galerkin Method Put To Action

The element-wise system of equations can be extrapolated by the Euler scheme as

```
% Time extrapolation
for it = 1:nt,
(...)
% Initialize flux vectors F (...)
for k=1:ne,
q(:,k) = q(:,k) + dt .* (Minv(:,k) * (K(:,k)'*q(:,k)-F(:,k)));
end (...)
end
```

where *nt* is the overall number of time steps, *dt* is the global time increment, and *ne* is the number of elements. Note that can directly update the solution vector *q* without intermediate storage at different time level(s).

The Discontinuous Galerkin Method Put To Action

A high-order extrapolation scheme like the predictor-corrector method can be implemented as

```
% Time extrapolation
for it = 1:nt,
(...)
% Predictor corrector scheme
% Initialize flux vectors F for all k
(...)
% First step (predictor)
for i=1:k,
k1(:,k)= Minv(:,k) * (K(:,k)'*q(:,k)-F(:,k));
end
(...)
```

```
% Initialize flux vectors F for q+dt*k1
(...)
% Second step (corrector)
for i=1:k,
k2(:,k)=  Minv(:,k)  *  (K(:,k)'*(q(:,k)+dt*k1(:,k))-
F(:,k));
end
(...)
% Update
for i=1:k,
q(:,k)= q(:,k) + dt (k1(:,k) + k2(:,k));
end
(...)
end
```

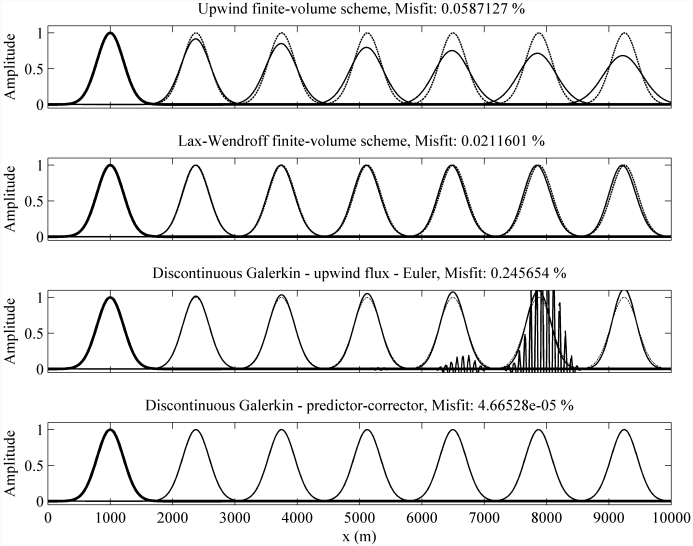
Example

Table 1: Simulation parameters for 1D discontinuous Galerkin advection

Parameter	Value	Meaning
ne	200	Elements
N	3	Order
a	2500 m/s	Velocity
x_{max}	10000 m	x-domain
dx_{min}	13.82 m	Increment
dt	4.4e-4 s	Time step
eps	0.08	Courant
σ	300 m	Gauss width
x_0	1000 m	Source x
t_{max}	3 s	Duration

Initiating the simulation with a spatial initial condition using a Gaussian function $e^{-1/\sigma^2(x-x_0)^2}$. A source term can be added to the system of equations in a straight forward way.

Example


Upwind finite-volume scheme, Misfit: 0.0587127 %
Lax-Wendroff finite-volume scheme, Misfit: 0.0211601 %
Discontinuous Galerkin - upwind flux - Euler, Misfit: 0.245654 %
Discontinuous Galerkin - predictor-corrector, Misfit: 4.66528e-05 %
Amplitude
0
0.5
1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
x (m)

Summary

- The discontinuous Galerkin method is a finite-element type method. The main difference is that the solution fields are not continuous at the element boundaries.
- The elemental mass and stiffness matrices are formulated very similar to the classic finite-element schemes. However, they are never assemble to a global system of equations. Therefore no large system matrix needs to be inverted.
- Elements are linked by a flux scheme, similar to the finite volume method. This scheme leads to an entirely local algorithm in the sense that all calculations are carried out at an elemental level. Communication happens only to direct neighbours.
- The discontinuous Galerkin scheme can easily to higher orders, keeping the local nature of the solution scheme. This leads to high efficiency when parallelizing the scheme.

Summary

- The solution fields can be extended using nodal and modal approaches. Modal approaches are preferred for tetrahedral schemes. Nodal solutions are preferred for hexahedral meshes.
- The discontinuous behaviour at the element boundaries and the associated discretization of the element boundaries increases the number of degrees of freedom compared to other methods.
- The flexibility with polynomial order, element size, local time stepping leads to a formidable problem when parallelizing a discontinuous Galerkin method: load balancing the computational task is not easy! Solution to this problem requires tight cooperation with computational scientists.
- In seismology, the discontinuous Galerkin method is useful for problem with highly complex geometries (by using tetrahedral meshes) and for problem with non-linear internal boundary conditions (e.g., dynamic ruputure problems).

Comprehensive Questions

1. What were the key points that led to the development of the discontinuous Galerkin method in seismology? Discuss the pros and cons of the method compared to finite-element type methods and the finite-difference method.
2. Explain qualitatively the difference between nodal and modal approaches.
3. Explain why the discontinuous Galerkin method lends itself to parallel implementation on supercomputer hardware.
4. What is $p-$ and $h-$adaptivity? Why is it straight forward to have this adaptivity with the discontinuous Galerkin method and not with others? Give examples in seismology where this adaptivity can be exploited and why.
5. What is local time-stepping? For what classes of Earth models and/or problems in seismology might it be useful?
6. What is the problem that arises on computers when using algorithms with h-/p-adaptivity and local time stepping?
7. What is meant by conservative and non-conservative properties in the context of advection problems?

Theoretical Problems

8. Show that the advection problem $\partial_t q + a\partial_x q = 0$ has a hyperbolic form.
9. The coupled 1D wave equation for longitudinal velocity v and pressure p can be formulated with compressibility K and density ρ as

$$\partial_t p + K\partial_x v = 0$$
$$\partial_t v + \frac{1}{\rho}\partial_x p = 0 \,.$$

   Formulate the Jacobian matrix of the coupled system of equations and calculate its eigenvalues.
10. Show that the rule of partial integration corresponds to Gauss' theorem in higher dimensions (assuming on of the functions under the integral to be unity). Explain the relevance of this for the discontinuous Galerkin method.
11. Reformulate the nodal discontinuous Galerkin solution allowing the velocity to be variable inside the element (Note: Compare with the spectral-element formulation).
12. Show that setting $\alpha = 0$ leads to the upwind flux scheme.
13. Search in the literature for the *classical* 4-term Runge-Kutta method. Formulate a pseudo-code for the scalar advection problem for this extrapolation scheme.

Programming Exercises

14. Using the sample code *dg*1*d* find numerically the Courant limit (for a fixed time extrapolation to t_{max}), keeping all other parameters constant, increasing the global spatial order N of the scheme.
15. Formulate an upwind finite-difference scheme for the scalar advection problem and write a computer programm. Discuss the diffusive behaviour.
16. Modify the sample code *dg*1*d* such that each element can have its own polynomial order (p-adaptivity) and size ($h-$adaptivity). (Suggestion: Initialize the shape of the solution and other matrices using the maximum number of degrees of freedom N^p_{max}).
17. Extend the sample code *dg*1*d* for the scalar advection problem to the 4-term Runge-Kutta method. Compare the accuracy of the method with the lower order extrapolation scheme as a function of spatial order N inside the elements.

Programming Exercises

18. Formulate the analytical solution to the advection problem and plot it along with the numerical solution in $dg1d$ each time you visualize it during extrapolation. Formulate an error between analytical and numerical result. Analyze the solution error as a function of propagation distance for the Euler scheme and the predictor-corrector scheme.
19. Explore the $p-$ and $h-$ adaptivity of the discontinuous Galerkin method in the following way. Using an appropriate Gaussian function defined on the entire physical domain decrease the element size by a factor of 5 towards the center of the domain. Find an appropriate variation of the order inside the elements to obtain a reasonable computational scheme (in the sense that the grid point distance does not vary too much). Hint: Use high order schemes at the edges of the physical domain and low(est) order schemes at the centre of the domain.