December 12, 2012

# Introduction to discontinuous Galerkin finite element methods (DG-FEMs)

Hamdullah YÜCEL

Computational Methods in Systems and Control Theory
Max Planck Institute for Dynamics of Complex Technical Systems
Magdeburg

# Outline

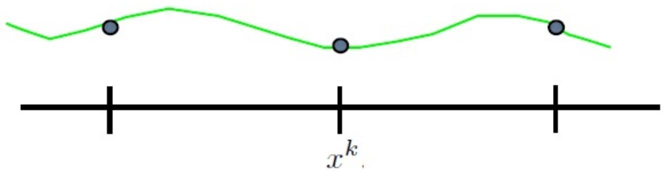
# Outline

# Outline

# Outline

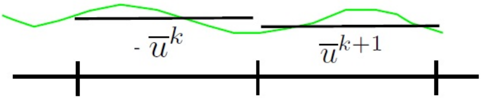
1. Motiviation
2. Elliptic Equations
3. Convection-diffusion Equations
4. Implementation

# Finite difference methods




- Main benefits
  - Simple to implement and fast
  - Explicit in time
  - Strong theory
- Main problem
  - Simple local approximation and geometric flexibility are not agreeable

# Finite volume methods



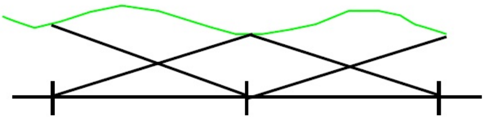

The local approximation is a cell average

$$\int_{x^{k-1/2}}^{x^{k+1/2}} u_h(x)dx = h^k \bar{u}^k$$

- Main benefits
  - Robust and fast due to locality
  - Complex geometries
  - Well suited for conservation laws
  - Explicit in time
- Main problem
  - Inability to archive high-order accuracy on general grids

# Finite element methods

We begin by splitting the solution into elements as



The solution is defined in a nonlocal manner

$$u_h(x) = \sum_{k=1}^{N} u_k \varphi_k(x)$$

- Main benefits
  - Higher-order accuracy and complex geometries can be combined
- Main problem
  - Implicit in time
  - Not well suited for problems with direction

# Summary

| | Complex geometries | Higher-order accuracy and *hp*-adaptivity | Local mass Conservation |
|---|---|---|---|
| FDM | $\times$ | ✓ | ✓ |
| FVM | ✓ | $\times$ | ✓ |
| FEM | ✓ | ✓ | $\times$ |
| DG | ✓ | ✓ | ✓ |

# Summary

| | Complex geometries | Higher-order accuracy and *hp*-adaptivity | Local mass Conservation |
|---|---|---|---|
| FDM | $\times$ | ✓ | ✓ |
| FVM | ✓ | $\times$ | ✓ |
| FEM | ✓ | ✓ | $\times$ |
| DG | ✓ | ✓ | ✓ |

What we need is a scheme that combines

# Summary

| | Complex geometries | Higher-order accuracy and *hp*-adaptivity | Local mass Conservation |
|---|---|---|---|
| FDM | $\times$ | $\checkmark$ | $\checkmark$ |
| FVM | $\checkmark$ | $\times$ | $\checkmark$ |
| FEM | $\checkmark$ | $\checkmark$ | $\times$ |
| DG | $\checkmark$ | $\checkmark$ | $\checkmark$ |

What we need is a scheme that combines

- The local-higher order/flexible element of FEM

# Summary

| | Complex geometries | Higher-order accuracy and *hp*-adaptivity | Local mass Conservation |
|---|---|---|---|
| FDM | $\times$ | $\checkmark$ | $\checkmark$ |
| FVM | $\checkmark$ | $\times$ | $\checkmark$ |
| FEM | $\checkmark$ | $\checkmark$ | $\times$ |
| DG | $\checkmark$ | $\checkmark$ | $\checkmark$ |

What we need is a scheme that combines

- The local-higher order/flexible element of FEM
- The local statement on the equation for FVM

# Summary

| | Complex geometries | Higher-order accuracy and *hp*-adaptivity | Local mass Conservation |
|---|---|---|---|
| FDM | × | ✓ | ✓ |
| FVM | ✓ | × | ✓ |
| FEM | ✓ | ✓ | × |
| DG | ✓ | ✓ | ✓ |

What we need is a scheme that combines

- The local-higher order/flexible element of FEM
- The local statement on the equation for FVM

These are exactly the components of the

Discontinuous Galerkin Finite Element Method

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

- Pros:

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

- Pros:
  - Flexibility for approximation order and complex meshes

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

- Pros:
  - Flexibility for approximation order and complex meshes
  - Local conservation of physical quantities such as mass, momentum, and energy

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

- Pros:
  - Flexibility for approximation order and complex meshes
  - Local conservation of physical quantities such as mass, momentum, and energy
  - Increase of the robustness and accuracy

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

- Pros:
  - Flexibility for approximation order and complex meshes
  - Local conservation of physical quantities such as mass, momentum, and energy
  - Increase of the robustness and accuracy
  - Facilitation of parallelization

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

- Pros:
  - Flexibility for approximation order and complex meshes
  - Local conservation of physical quantities such as mass, momentum, and energy
  - Increase of the robustness and accuracy
  - Facilitation of parallelization
- Cons:

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

- Pros:
  - Flexibility for approximation order and complex meshes
  - Local conservation of physical quantities such as mass, momentum, and energy
  - Increase of the robustness and accuracy
  - Facilitation of parallelization
- Cons:
  - Large number of degrees of freedom

# Discontinuous Galerkin Methods

DG is a class of FEMs which use discontinuous functions as the solution (and the test functions)

- Pros:
  - Flexibility for approximation order and complex meshes
  - Local conservation of physical quantities such as mass, momentum, and energy
  - Increase of the robustness and accuracy
  - Facilitation of parallelization
- Cons:
  - Large number of degrees of freedom
  - Ill-conditioning and denser global matrix with increasing approximation order

# Model Problem

- Let $\Omega \subset \mathbf{R}^2$ be a bounded open polygonal domain,
- Consider the following model problem

$$
\begin{aligned}
-\mathrm{div}(\varepsilon \nabla u(x)) + \alpha u(x) &= f(x) \qquad && x \in \Omega, \\
u(x) &= g_D(x) && x \in \Gamma_D, \\
\varepsilon \frac{\partial u(x)}{\partial n} &= g_N(x) && x \in \Gamma_N,
\end{aligned}
$$

  where $\Gamma = \Gamma_D \cup \Gamma_N$.
- $f \in L^2(\Omega)$, $g_D \in H^{1/2}(\Gamma_D)$, $g_N \in L^2(\Gamma_N)$ and $\varepsilon$ is symmetric and positive definite.

# DG Discretization

- Let $\{\mathscr{T}_h\}_h$ be a partition of a domain $\Omega$ with the conformity and shape regularity
- Let $\mathscr{E}_h$ be the set of all edges and the interior edges, Dirichlet and Neumann boundary edges are denoted by $\mathscr{E}_h^0$, $\mathscr{E}_h^{\partial D}$, $\mathscr{E}_h^{\partial N}$, respectively
- An element and an edge are denoted by $K$ and $E$, respectively
- Let $|K|$ denote the area of triangle $K$ and $|E|$ denote the length of edge $E$

# DG Discretization

The solution is represented as

$$u(x) = \sum_{m=1}^{N_{el}} \sum_{j=1}^{N_{loc}} u_j^m \varphi_m^j(x)$$

$N_{el}$: Number of elements
$N_{loc} = \frac{(p+1)(p+2)}{2}$ local dimension with $p$ approximation order

# DG Discretization

The solution is represented as

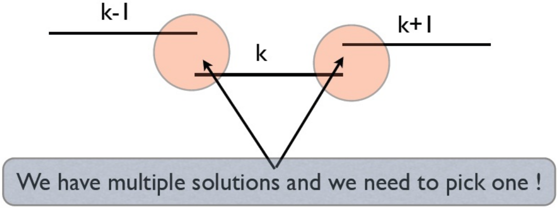
$$u(x) = \sum_{m=1}^{N_{el}} \sum_{j=1}^{N_{loc}} u_j^m \varphi_m^j(x)$$

$N_{el}$: Number of elements
$N_{loc} = \frac{(p+1)(p+2)}{2}$ local dimension with $p$ approximation order

Basic idea behind the construction of DGFEMs: Replace integration-by-parts over by:

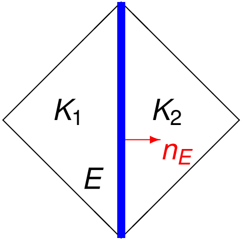
- element-wise integration-by-parts, and
- summing such formula over the elements in the finite element partition of the domain.

# DG Discretization-One Dimension




- The jump operator $[\upsilon]_{x_k} = \upsilon|_{I_k}(x_k) - \upsilon_{I_{k+1}}(x_k)$
- The average operator $\{\upsilon\}_{x_k} = \frac{1}{2}(\upsilon_{I_k}(x_k) + \upsilon_{I_{k+1}}(x_k))$

# DG Discretization-Two Dimensions




- Scalar functions $y$

$$[\![y]\!] = (y|_{K_1^E} - y|_{K_2^E})\mathbf{n_E}, \quad \{\{y\}\} = \frac{1}{2}(y|_{K_1^E} + y|_{K_2^E}).$$

- Vector field $\nabla y$

$$[\![\nabla y]\!] = (\nabla y|_{K_1^E} - \nabla y|_{K_2^E}) \cdot \mathbf{n_E} \quad \{\{\nabla y\}\} = \frac{1}{2}(\nabla y|_{K_1^E} + \nabla y|_{K_1^E}).$$

# DG Solution

The DG finite element spaces on $\mathscr{T}_h$

$$V_h = U_h = \{u_h \in L^2(\Omega) \mid u|_K \in \mathbb{P}_n(E), \ \forall K \in \mathscr{T}_h\}$$

**DG approximation of the solution**

*Find $u_h \in V_h$ such that*

$$a_h(u_h, v_h) = l_h(v), \qquad \forall v_h \in V_h(\Omega).$$

# Interior Penalty Galerkin Methods

$$a_h(u,v) \quad = \quad \sum_{K \in \mathscr{T}_h} \int_K \varepsilon \nabla u \cdot \nabla v \, dx + \sum_{K \in \mathscr{T}_h} \int_K \alpha u v \, dx$$

# Interior Penalty Galerkin Methods

$$
\begin{aligned}
a_h(u,v) \quad = \quad & \sum_{K\in\mathcal{T}_h}\int_K \varepsilon\nabla u\cdot\nabla v\,dx + \sum_{K\in\mathcal{T}_h}\int_K \alpha uv\,dx \\
& - \sum_{E\in\mathcal{E}_h^0\cup\mathcal{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla u\}\}\cdot[\![v]\!]\,ds
\end{aligned}
$$

# Interior Penalty Galerkin Methods

$$\begin{aligned} a_h(u,v) &= \sum_{K\in\mathscr{T}_h}\int_K \varepsilon\nabla u\cdot\nabla v\,dx + \sum_{K\in\mathscr{T}_h}\int_K \alpha uv\,dx \\ &\quad - \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla u\}\}\cdot[\![v]\!]\,ds + \kappa \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla v\}\}\cdot[\![u]\!]\,ds \end{aligned}$$

# Interior Penalty Galerkin Methods

$$
\begin{aligned}
a_h(u,v) \;=\; & \sum_{K\in\mathscr{T}_h}\int_K \varepsilon\nabla u\cdot\nabla v\,dx + \sum_{K\in\mathscr{T}_h}\int_K \alpha uv\,dx \\
& - \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla u\}\}\cdot[[v]]\,ds + \kappa \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla v\}\}\cdot[[u]]\,ds \\
& + \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}} \frac{\sigma\varepsilon}{|E|^{\beta_0}}\int_E [[u]]\cdot[[v]]\,ds
\end{aligned}
$$

# Interior Penalty Galerkin Methods

$$
\begin{aligned}
a_h(u,v) \;=\; & \sum_{K\in\mathscr{T}_h}\int_K \varepsilon\nabla u\cdot\nabla v\,dx + \sum_{K\in\mathscr{T}_h}\int_K \alpha uv\,dx \\
& - \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla u\}\}\cdot[[v]]\,ds + \kappa \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla v\}\}\cdot[[u]]\,ds \\
& + \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}} \frac{\sigma\varepsilon}{|E|^{\beta_0}}\int_E [[u]]\cdot[[v]]\,ds
\end{aligned}
$$

with $\sigma$ penalty parameter and $\beta_0$ superpenalization parameter.

# Interior Penalty Galerkin Methods

$$
\begin{aligned}
a_h(u,v) \;=\; & \sum_{K\in\mathscr{T}_h}\int_K \varepsilon\nabla u\cdot\nabla v\,dx + \sum_{K\in\mathscr{T}_h}\int_K \alpha uv\,dx \\
& - \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla u\}\}\cdot[\![v]\!]\,ds + {\color{red}\kappa} \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla v\}\}\cdot[\![u]\!]\,ds \\
& + \sum_{E\in\mathscr{E}_h^0\cup\mathscr{E}_h^{\partial D}} \frac{\sigma\varepsilon}{|E|^{\beta_0}}\int_E [\![u]\!]\cdot[\![v]\!]\,ds
\end{aligned}
$$

with $\sigma$ penalty parameter and $\beta_0$ superpenalization parameter.

- if $\kappa=-1$, **SIPG**, i.e., symmetric interior penalty Galerkin,
- if $\kappa=1$, **NIPG**, i.e., nonsymmetric interior penalty Galerkin,
- if $\kappa=0$, **IIPG**, i.e., incomplete interior penalty Galerkin.

# Linear form

$$
\begin{aligned}
l_h(v) \quad = \quad & \sum_{K \in \mathscr{T}_h} \int_K f v \, dx + \sum_{E \in \mathscr{E}_h^{\partial D}} \frac{\sigma \varepsilon}{|E|^{\beta_0}} \int_E g_D \mathbf{n} \cdot [\![ v ]\!] \, ds \\
& + \kappa \sum_{E \in \mathscr{E}_h^{\partial D}} \int_E g_D \{\!\{ \varepsilon \nabla v \}\!\} \, ds + \sum_{E \in \mathscr{E}_h^{\partial N}} \int_E g_N v \, ds
\end{aligned}
$$

# Properties of the DG methods

| Method | Cons. | A.C. | Stab. | $H^1$ | $L^2$ |
|---|---|---|---|---|---|
| Brezzi et al. [2000] | $\checkmark$ | $\checkmark$ | $\checkmark$ | $h^p$ | $h^{p+1}$ |
| LDG [Cockburn-Shu,1998] | $\checkmark$ | $\checkmark$ | $\checkmark$ | $h^p$ | $h^{p+1}$ |
| CDG [Peraire-Persson,2008] | $\checkmark$ | $\checkmark$ | $\checkmark$ | $h^p$ | $h^{p+1}$ |
| SIPG [Arnold 1982] | $\checkmark$ | $\checkmark$ | $\checkmark$ | $h^p$ | $h^{p+1}$ |
| Bassi et al. [1997] | $\checkmark$ | $\checkmark$ | $\checkmark$ | $h^p$ | $h^{p+1}$ |
| NIPG [Riviere 1999] | $\checkmark$ | $\times$ | $\checkmark$ | $h^p$ | $h^p$ |
| Babuška-Zlámal [1973] | $\times$ | $\times$ | $\checkmark$ | $h^p$ | $h^{p+1}$ |
| Baumann-Oden ($p = 1$) [1999] | $\checkmark$ | $\times$ | $\times$ | $\times$ | $\times$ |
| Baumann-Oden ($p \geq 2$) [1999] | $\checkmark$ | $\times$ | $\times$ | $h^p$ | $h^p$ |
| IIPG [Wheeler 2004] | $\checkmark$ | $\times$ | $\checkmark$ | $h^p$ | $h^p$ |

*D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini,* Unified analysis of discontinuous Galerkin methods for elliptic problems, *2002.*

# Model Problem

- Consider the following convection-diffusion equation

$$
\begin{aligned}
-\varepsilon\Delta u(x) + \beta\cdot\nabla u(x) + \alpha u(x) &= f(x) \qquad && x\in\Omega,\\
u(x) &= g_D(x) && x\in\Gamma_D,\\
\varepsilon\frac{\partial u(x)}{\partial n} &= g_N(x) && x\in\Gamma_N,
\end{aligned}
$$

  where $\Gamma = \Gamma_D \cup \Gamma_N$.

- $f\in L^2(\Omega)$, $g_D\in H^{3/2}(\Gamma_D)$, $g_N\in L^2(\Gamma_N)$, $\beta\in\left(W^{1,\infty}(\Omega)\right)^2$, $\alpha\in L^\infty(\Omega)$

- The boundary edges are decomposed into the **inflow** and **outflow** edges;

$$\Gamma^{-} = \{x \in \partial\Omega : \ \beta \cdot \mathbf{n}(x) < 0\},$$
$$\Gamma^{+} = \{x \in \partial\Omega : \ \beta \cdot \mathbf{n}(x) \geq 0\}$$

- The boundary edges are decomposed into the **inflow** and **outflow** edges;

$$\Gamma^{-} = \{x \in \partial\Omega : \ \beta \cdot \mathbf{n}(x) < 0\},$$
$$\Gamma^{+} = \{x \in \partial\Omega : \ \beta \cdot \mathbf{n}(x) \geq 0\}$$

- We use upwind-discretization to discretize convection term

$$u = \begin{cases} u|_{K^1}, & \text{if } \beta \cdot \mathbf{n_E} < 0, \\ u|_{K^2}, & \text{if } \beta \cdot \mathbf{n_E} \geq 0, \end{cases} \qquad u^e = \begin{cases} u|_{K^2}, & \text{if } \beta \cdot \mathbf{n_E} < 0, \\ u|_{K^1}, & \text{if } \beta \cdot \mathbf{n_E} \geq 0. \end{cases}$$

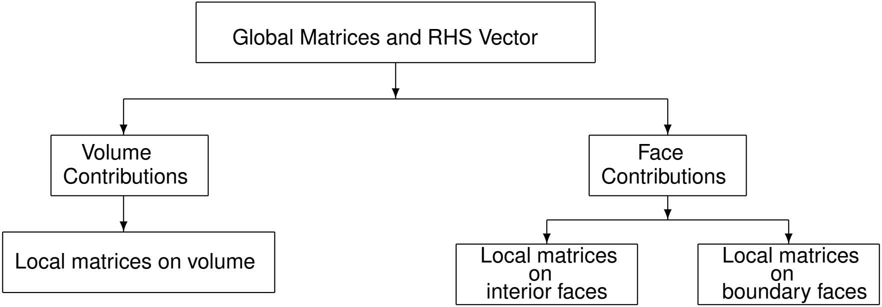
# (Bilinear Form

$$
\begin{aligned}
a_h(u,v) \;=\; & \sum_{K\in\mathcal{T}_h}\int_K \varepsilon\nabla u\cdot\nabla v\,dx \\
& - \sum_{E\in\mathcal{E}_h^0\cup\mathcal{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla u\}\}\cdot[\![v]\!]\,ds + {\color{red}\kappa}\sum_{E\in\mathcal{E}_h^0\cup\mathcal{E}_h^{\partial D}}\int_E \{\{\varepsilon\nabla v\}\}\cdot[\![u]\!]\,ds \\
& + \sum_{E\in\mathcal{E}_h^0\cup\mathcal{E}_h^{\partial D}}\frac{\sigma\varepsilon}{|E|^{\beta_0}}\int_E [\![u]\!]\cdot[\![v]\!]\,ds + \sum_{K\in\mathcal{T}_h}\int_K \beta\cdot\nabla uv+\alpha uv\,dx \\
& + \sum_{K\in\mathcal{T}_h}\int_{\partial K^-\setminus\Gamma^-}\beta\cdot\mathbf{n}(u^e-u)v\,ds - \sum_{K\in\mathcal{T}_h}\int_{\partial K^-\cap\Gamma^-}\beta\cdot\mathbf{n}uv\,ds
\end{aligned}
$$

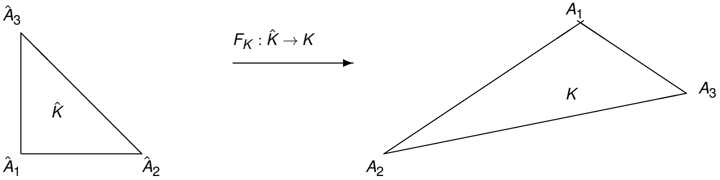
# Linear Form

$$
\begin{aligned}
l_h(v) \quad = \quad & \sum_{K \in \mathscr{T}_h} \int_K f v \, dx \\
& + \sum_{E \in \mathscr{E}_h^{\partial D}} \frac{\sigma \varepsilon}{|E|^{\beta_0}} \int_E g_D \mathbf{n} \cdot [\![ v ]\!] \, ds + \kappa \sum_{E \in \mathscr{E}_h^{\partial D}} \int_E g_D \{\{ \varepsilon \nabla v \}\} \, ds \\
& - \sum_{K \in \mathscr{T}_h} \int_{\partial K^- \cap \Gamma^-} \beta \cdot \mathbf{n} \, g_D v \, ds + \sum_{E \in \mathscr{E}_h^{\partial N}} \int_E g_N v \, ds.
\end{aligned}
$$

# The Implementation Structure



# Mapping between Physical and Reference Triangle




$$F_K \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad x = \sum_{i=1}^{3} x_i \hat{\phi}_i(\hat{x},\hat{y}), \quad y = \sum_{i=1}^{3} x_i \hat{\phi}_i(\hat{x},\hat{y}),$$

where

$$\hat{\phi_1}(\hat{x},\hat{y}) = 1 - \hat{x} - \hat{y}, \quad \hat{\phi_2}(\hat{x},\hat{y}) = \hat{x}, \quad \hat{\phi_2}(\hat{x},\hat{y}) = \hat{y}.$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = F_K \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = B_K \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + b_K,$$

where $B_K$ is an invertible matrix and $b_K$ is a translation vector

$$B_K = \begin{pmatrix} a_{11}^K & a_{12}^K \\ a_{21}^K & a_{22}^K \end{pmatrix} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix}, \quad b_K = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}.$$

# Local Matrices on Volume

For a fixed element $K$ :

$$(D_K)_{i,j} = \int_K \varepsilon \nabla \phi_{j,K} \cdot \nabla \phi_{i,K} dx, \ (C_K)_{i,j} = \int_K \beta \cdot \nabla \phi_{j,K} \phi_{i,K} dx, \ (R_K)_{i,j} = \int_K \alpha \phi_{j,K} \phi_{i,K} dx.$$

$\forall \, 1 \leq i, j, \leq N_{loc}$.
After a change of variable with the mapping $F_K$,

$$\begin{aligned}
(D_K)_{i,j} &= 2|K| \int_{\hat{K}} \varepsilon (B_K^T)^{-1} \hat{\nabla} \hat{\phi}_i \cdot (B_K^T)^{-1} \hat{\nabla} \hat{\phi}_j \, dx, \\
(C_K)_{i,j} &= 2|K| \int_{\hat{K}} \beta \cdot (B_K^T)^{-1} \hat{\nabla} \hat{\phi}_j \hat{\phi}_i \, dx, \\
(R_K)_{i,j} &= 2|K| \int_{\hat{K}} (\alpha \circ F_K) \hat{\phi}_j \hat{\phi}_i \, dx.
\end{aligned}$$

The local right-hand side $b_K$ are $(b_K)_i = \int_K f \phi_{i,K} \, dx$.

**Algorithm 1: Computing local contributions from element K**

initialize $D_K = 0$, $C_K = 0$, $R_K = 0$
initialize the quadrature weights $w$ and points $s$
loop over quadrature points : for k=1 to $N_G$ do
    compute determinant of $B_K$
    for i=1 to $N_{loc}$ do
        compute values of basis functions $\phi_{i,K}(s(k))$
        compute derivatives of basis functions $\nabla_{i,K}(s(k))$
    end
    compute global coordinates x of quadrature points s(k)
    compute source function f(x)
    for i=1 to $N_{loc}$ do
        for j=1 to $N_{loc}$ do
            $D_K(i,j) = D_K(i,j) + w(k)det(B_K)\varepsilon\nabla\phi_{j,K}(s(k)) \cdot \nabla\phi_{i,E}(s(k))$
            $C_K(i,j) = C_K(i,j) + w(k)det(B_K)\beta \cdot \nabla\phi_{j,K}(s(k))\phi_{i,K}(s(k))$
            $R_K(i,j) = R_K(i,j) + w(k)det(B_K)\alpha\phi_{i,K}(s(k))\phi_{j,K}(s(k))$
        end
        $b_K(i) = b_K(i) + w(k)det(B_K)f(x)\phi_{i,K}(s(k))$
    end
end

# Local matrices on faces-Diffusion Part

Let $E \in \mathscr{E}_h^0$,

$$T_D = -\int_E \{\varepsilon \nabla u_h \cdot n_E\}[\upsilon] + \kappa \int_E \{\varepsilon \nabla \upsilon \cdot n_E\}[u_h] + \frac{\sigma\varepsilon}{|E|^{\beta_0}} \int_E [u_h][\upsilon].$$

$$(D_E^{11})_{i,j} = -\frac{1}{2}\int_E \varepsilon \nabla \phi_{j,K_E^1} \cdot n_E \phi_{i,K_E^1}\, ds + \frac{\kappa}{2}\int_E \varepsilon \nabla \phi_{i,K_E^1} \cdot n_E \phi_{j,K_E^1}\, ds + \frac{\sigma\varepsilon}{|E|^{\beta_0}} \int_E \phi_{j,K_e^1}\phi_{i,K_e^1}\, ds,$$

$$(D_E^{22})_{i,j} = \frac{1}{2}\int_E \varepsilon \nabla \phi_{j,K_E^2} \cdot n_E \phi_{i,K_E^2}\, ds - \frac{\kappa}{2}\int_E \varepsilon \nabla \phi_{i,K_E^2} \cdot n_E \phi_{j,K_E^2}\, ds + \frac{\sigma\varepsilon}{|E|^{\beta_0}} \int_E \phi_{j,K_E^2}\phi_{i,K_E^2}\, ds,$$

$$(D_E^{12})_{i,j} = -\frac{1}{2}\int_E \varepsilon \nabla \phi_{j,K_E^2} \cdot n_E \phi_{i,K_E^1}\, ds - \frac{\kappa}{2}\int_E \varepsilon \nabla \phi_{i,K_E^1} \cdot n_E \phi_{j,K_E^2}\, ds - \frac{\sigma\varepsilon}{|E|^{\beta_0}} \int_E \phi_{j,K_E^2}\phi_{i,K_E^1}\, ds,$$

$$(D_E^{21})_{i,j} = \frac{1}{2}\int_E \varepsilon \nabla \phi_{j,K_E^1} \cdot n_E \phi_{i,K_E^2}\, ds + \frac{\kappa}{2}\int_e \varepsilon \nabla \phi_{i,K_E^2} \cdot n_E \phi_{j,K_E^1}\, ds - \frac{\sigma\varepsilon}{|E|^{\beta_0}} \int_E \phi_{j,K_E^1}\phi_{i,K_E^2}\, ds.$$

# Local matrices on faces-Convection Part

$$T_C = \sum_{K\in\mathscr{T}_h} \int_{\partial K^-\setminus\Gamma^-} \beta\cdot\mathbf{n}(u^e-u)v\,ds = c_E^{11}+c_E^{22}+c_E^{12}+c_E^{21}.$$

By the upwind discretization [Lesaint,Raviert,1974-Reed,Hill,1973]

$$u = \begin{cases} u|_{K^1}, & \text{if } \beta\cdot\mathbf{n_E}<0,\\ u|_{K^2}, & \text{if } \beta\cdot\mathbf{n_E}\geq 0,\end{cases} \qquad u^e = \begin{cases} u|_{K^2}, & \text{if } \beta\cdot\mathbf{n_E}<0,\\ u|_{K^1}, & \text{if } \beta\cdot\mathbf{n_E}\geq 0.\end{cases}$$

The local matrices: $\forall E\in\mathscr{E}_h^0$ satisfying $\beta\cdot n_E<0$,

$$(C_E^{11})_{i,j} = -\int_E \beta\cdot\mathbf{n_E}\phi_{j,K_E^1}\phi_{i,K_E^1}, \qquad (C_E^{12})_{i,j} = \int_E \beta\cdot\mathbf{n_E}\phi_{j,K_E^2}\phi_{i,K_E^1}$$

and $\forall E\in\mathscr{E}_h^0$ satisfying $\beta\cdot n_E\geq 0$,

$$(C_E^{22})_{i,j} = -\int_E \beta\cdot\mathbf{n_E}\phi_{j,K_E^2}\phi_{i,K_E^2}, \qquad (C_E^{21})_{i,j} = \int_E \beta\cdot\mathbf{n_E}\phi_{j,K_E^1}\phi_{i,K_E^2}.$$

**Algorithm 2: Computing local contributions from interior edges**

initialize $D_E^{11} = D_E^{22} = D_E^{12} = D_E^{21} = 0, C_E^{11} = C_E^{22} = C_E^{12} = C_E^{21} = 0$
initialize the quadrature weights $w$ and the points $s$ on $[-1,1]$
compute edge length $|E|$, normal vector $n_E$
get face neighbors $K_E^1$ and $K_E^2$
loop over quadrature points: for k=1 to $N_G$ do
    compute local coordinates $ss1$ on $K_E^1$ and $ss2$ on $K_E^2$ of quadrature point $s(k)$
    for i=1 to $N_{loc}$ do
        compute values of basis functions $\phi_{i,K_E^1}(s(k))$ and $\phi_{i,K_E^2}(ss1)$
        compute derivatives of basis functions $\nabla\phi_{i,K_E^1}(s(k))$ and $\nabla\phi_{i,K_E^2}(ss2)$
    end
    for i=1 to $N_{loc}$ do
        for j=1 to $N_{loc}$ do

$$\begin{aligned}
D_E^{11}(i,j) &= D_E^{11}(i,j) - 0.5w(k)|E|\phi_{i,K_E^1}(s(k))(\nabla\phi_{j,K_E^1}(s(k))\cdot n_E) \\
D_E^{11}(i,j) &= D_E^{11}(i,j) + 0.5\varepsilon w(k)|E|\phi_{j,K_E^1}(s(k))(\nabla\phi_{i,K_E^1}(s(k))\cdot n_E) \\
D_E^{11}(i,j) &= D_E^{11}(i,j) - \frac{\sigma\varepsilon}{|E|^{\beta_0}} w(k)|E|\phi_{i,K_E^1}(s(k))\phi_{j,K_E^1}(s(k)) \\
&\dots \\
D_E^{21}(i,j) &= D_E^{21}(i,j) + 0.5w(k)|E|\phi_{i,K_E^2}(s(k))(\nabla\phi_{j,K_E^1}(s(k))\cdot n_E) \\
D_E^{21}(i,j) &= D_E^{21}(i,j) + 0.5\varepsilon w(k)|E|\phi_{j,K_E^1}(s(k))(\nabla\phi_{i,K_e^2}(s(k))\cdot n_E) \\
D_E^{21}(i,j) &= D_E^{21}(i,j) - \frac{\sigma\varepsilon}{|E|^{\beta_0}} w(k)|E|\phi_{i,K_E^2}(s(k))\phi_{j,K_E^1}(s(k))
\end{aligned}$$

        end
    end

if face is influx face do
for i=1 to $N_{loc}$ do
for j=1 to $N_{loc}$ do

$$
\begin{aligned}
C_E^{11}(i,j) &= C_E^{11}(i,j) - w(k)|E|\beta \cdot n \phi_{i,K_e^1}(s(k))\phi_{j,K_E^1}(s(k)) \\
C_E^{12}(i,j) &= C_E^{12}(i,j) + w(k)|E|\beta \cdot n \phi_{i,K_e^2}(s(k))\phi_{j,K_E^1}(s(k))
\end{aligned}
$$

end
end
else if face is outflux face do
for i=1 to $N_{loc}$ do
for j=1 to $N_{loc}$ do

$$
\begin{aligned}
C_E^{22}(i,j) &= C_E^{22}(i,j) - w(k)|E|\beta \cdot n \phi_{i,K_E^2}(s(k))\phi_{j,K_E^2}(s(k)) \\
C_E^{21}(i,j) &= C_E^{21}(i,j) + w(k)|E|\beta \cdot n \phi_{i,K_E^1}(s(k))\phi_{j,K_E^2}(s(k))
\end{aligned}
$$

end
end
end

end

# Volume Contributions

**Algorithm 3: Volume Contributions**

initialize k=0
loop over the elements: for k=1 to $N_{el}$ do
  compute local matrices $D_{K_k}$, $C_{K_k}$, $R_{K_k}$ and $b_{K_k}$
  for i=1 to $N_{el}$ do
    ie=i+k
    for j=1 to $N_{el}$ do
      je=j+k
      $D_{global}(ie, je) = D_{global}(ie, je) + D_K(i, j)$
      $C_{global}(ie, je) = C_{global}(ie, je) + C_K(i, j)$
      $R_{global}(ie, je) = R_{global}(ie, je) + R_K(i, j)$
    end
    $b_{global}(ie) = b_{global}(ie) + b_{E_k}(i)$
    k=k+Nloc
  end
end

# Face Contributions

**Algorithm 4: Face Contributions**

loop over the edges: for k=1 to $N_{face}$ do
  get face neighbors $K_k^1$ and $K_k^2$
  if face is an interior face do
  compute local matrices $D_k^{11}$, $D_k^{22}$, $D_k^{12}$, $D_k^{21}$, $C_k^{11}$, $C_k^{22}$, $C_k^{12}$, $C_k^{21}$
  assemble $D_k^{11}$ and $C_k^{11}$ contributions:
    for i=1 to $N_{loc}$ do
      $ie = i + (K_k^1 - 1)N_{loc}$
      for j=1 to $N_{loc}$ do
        $je = j + (K_k^1 - 1)N_{loc}$
        $D_{global}(ie, je) = D_{global}(ie, je) + D_k^{11}(i, j)$
        $C_{global}(ie, je) = C_{global}(ie, je) + C_k^{11}(i, j)$
      end
    end
  assemble $D_k^{21}$ and $C_k^{21}$ contributions:
    for i=1 to $N_{loc}$ do
      $ie = i + (K_k^2 - 1)N_{loc}$
      for j=1 to $N_{loc}$ do
        $je = j + (K_k^1 - 1)N_{loc}$
        $D_{global}(ie, je) = D_{global}(ie, je) + D_k^{21}(i, j)$
        $C_{global}(ie, je) = C_{global}(ie, je) + C_k^{21}(i, j)$
      end
    end
  end

end

# Matlab Implementation

- Goal: Faster program with less memory storage
- Problem: the number of loops
- Solution:
  - Sparse Matrix structure
  - Vectorization coding style
  - Multiple matrix multiplications (MULTIPROD)

# Multiple matrix multiplications (MULTIPROD)

- Generalization for N-D arrays of the MATLAB matrix multiplication operator (*)
- Perform any kind of multiple scalar-by-matrix or matrix multiplication:
  - Arrays of scalars by arrays of scalars, vectors (*) or matrices.
  - Arrays of vectors (*) by arrays of scalars, vectors (*) or matrices.
  - Arrays of matrices by arrays of scalars, vectors (*) or matrices.

(*) internally converted by MULTIPROD into row or column matrices

*P. d. Leva,* MULTIPROD TOOLBOX, Multiple matrix multiplications, with array expansion enabled, *University of Rome Foro Italico, Rome.*

# Example for MULTIPROD

- Consider

```
% Building A and B
A = rand(2, 5);
B = rand(5, 3, 1000, 10);
% Multiplying A by all the matrices in B
for i = 1:1000
    for j = 1:10
        C(:,:,i,j) = A * B(:,:,i,j);
    end
end
```

- C = MULTIPROD(A, B),
- Performance 380 times better.

# References

D. N. Arnold, *An interior penalty finite element method with discontinuous elements*, SIAM J. Numer. Anal., 19:724-760, 1982.

D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, 2002.

B. Ayuso, and L. D. Marini, *Discontinuous Galerkin methods for advection-diffusion-reaction problems*, SIAM J. Numer. Anal., 47:1391-1420, 2009.

J. S. Hesthaven, and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. Volume 54, Springer Texts in Applied Mathematics*, Springer Verlag, New York, 2008.

B. Rivière, *Discontinuous Galerkin methods for solving elliptic and parabolic equations, Theory and implementation*, SIAM Volume 35, Frontiers in Applied Mathematics, 2008.

# THANK YOU !