Spectral-element methods

F. Auteri

Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano

Course of Applied CFD, a. a. 2014/15.

Objectives

Main objective of the lecture is to give an introduction to the spectral-element method, i.e.

- introduce the basis adopted
- introduce the main operations on the basis
- give an idea of advantages and drawbacks

Outline

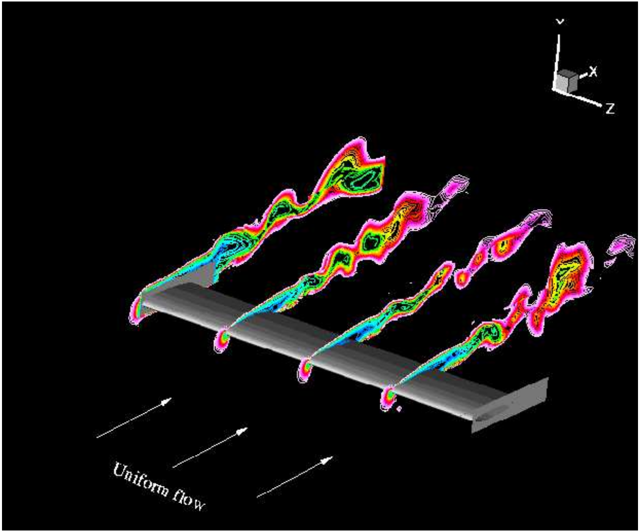
Introduction

Hypothesis: for unstable phenomena (strong dependence on initial and boundary data) high accuracy methods are needed to compute reliable results
Fact: real world phenomena take place in complex geometries

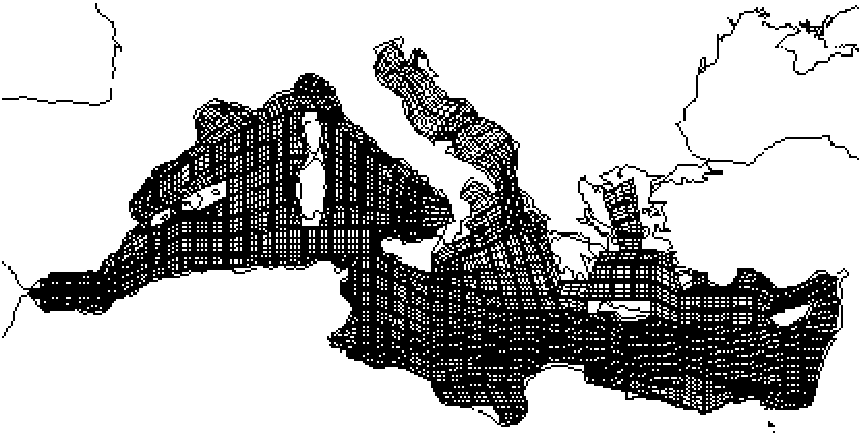
- Spectral methods provide the accuracy required
- Finite elements provide the geometrical flexibility

Spectral elements $=$ spectral accuracy $+$ geometrical flexibility
Spectral elements $=$ finite elements with high-order, numerically stable bases

Examples: Wing flow



Examples: Geophysical flow



Outline

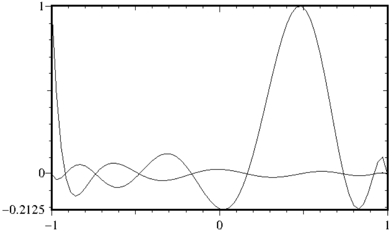
Two families of spectral elements

- Lagrangian-basis spectral elements: the basis is built by means of Lagrangian polynomials through Gauss(-Lobatto) points
- Hierarchical-basis spectral elements: the basis is built by means of Jacobi polynomials

This lecture is essentially devoted to hierarchical spectral elements

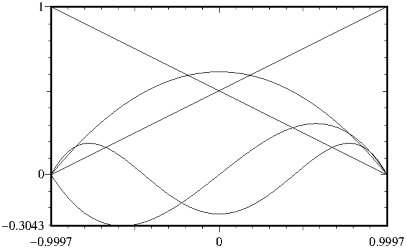
Lagrangian versus hierarchical bases

LAGRANGIAN



$$\mathcal{P}_N u = \sum_{i=0}^{N} \Phi_i^L u_i$$
$$\Phi_i^L = \prod_{k=0,k\neq i}^{N} \frac{(x-x_k)}{(x_i-x_k)}$$

HIERARCHICAL



$$\mathcal{P}_N u = \sum_{i=0}^{N} \Phi_i^H \hat{u}_i$$
$$\Phi_0^H = (1-x)/2$$
$$\Phi_1^H = (1+x)/2$$
$$\Phi_i^H = (1-x)^2/4 \;\; P_i^{1,1}(x)$$

Lagrangian versus hierarchical bases (2)

LAGRANGIAN

Advantages

+ mass matrix is easily lumped
+ unknowns are nodal values
+ well established method

Drawbacks

– hard to extend to n-simplex ($n > 1$)
– make difficult p-adaptivity
– full stiffness local operator

HIERARCHICAL

Advantages

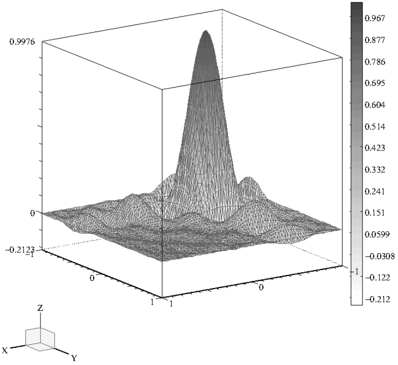
+ extend easily up to 3-D simplexes
+ allow different bases
+ well suited for p-adaptivity
+ sparse stiffness local operator

Drawbacks

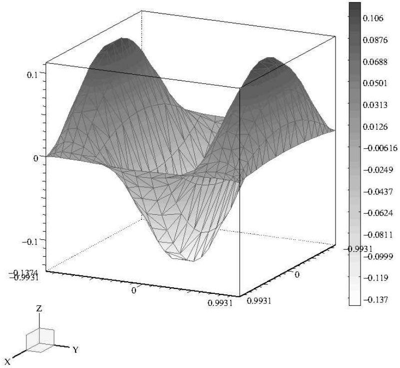
– need to transform to obtain nodal values
– mass lumping?
– loose sparsity in curved elements

Lagrangian versus hierarchical bases: 2-D examples

LAGRANGIAN



HIERARCHICAL



Fundamentals on Jacobi orthogonal polynomials

Orthogonal polynomials of degree n $p_0(x), p_1(x), \ldots, p_n(x)$ on (a, b):

1. $p_n(x)$ is a polynomial of precise degree n in which the coefficient of x^n is positive;
2. the system $\{p_n(x)\}$ is orthonormal, $\int_a^b p_n(x) p_m(x) w(x) dx = \delta_{nm}$

Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$ are the polynomial eigenfunctions of the following singular Sturm–Liouville problem $(\alpha, \beta > -1)$:

$$-(1 - x^2)\, u'' + ((\alpha + \beta + 2)x + \alpha - \beta)\, u' = \lambda u$$

whose eigenvalues are: $\lambda = n(n + \alpha + \beta + 1)$

A few properties of Jacobi polynomials

1. Orthogonality with respect to $w(x) = (1-x)^\alpha (1+x)^\beta$:

$$\int_{-1}^{1} P_n^{(\alpha,\beta)}(x) P_m^{(\alpha,\beta)}(x)(1-x)^\alpha (1+x)^\beta dx = \delta_{nm}$$

2. Eigenfunctions of the (singular) differential equation:
$\frac{d}{dx}\{(1-x)^{\alpha+1}(1+x)^{\beta+1}\ u'\} + n(n+\alpha+\beta+1)(1-x)^\alpha(1+x)^\beta\ u = 0$
3. Recurrence formula: $P_0^{(\alpha,\beta)}(x) = 1$,
$P_1^{(\alpha,\beta)}(x) = \frac{1}{2}(\alpha+\beta+2)x + \frac{1}{2}(\alpha-\beta)$,
$P_n^{(\alpha,\beta)}(x) = (\rho_n x + \sigma_n) P_{n-1}^{(\alpha,\beta)}(x) + \tau_n P_{n-2}^{(\alpha,\beta)}(x)$
4. $\frac{d}{dx} P_n^{(\alpha,\beta)}(x) = \frac{1}{2}(n+\alpha+\beta+1) P_{n-1}^{(\alpha+1,\beta+1)}(x)$

Quadrilateral spectral elements: the basis

In quadrilateral spectral elements the basis is defined on the standard $[-1,1]^2$ as tensor product of 1D bases:

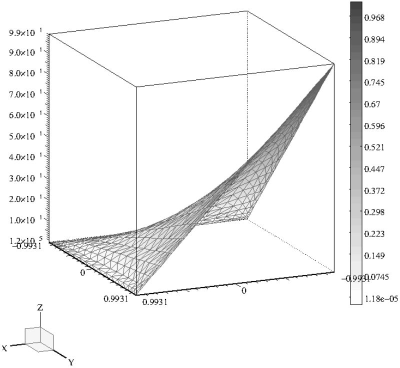
$$\mathcal{P}_N u(x,y) = \sum_{i=0}^{N} \sum_{j=0}^{M} \Phi_i(x)\, \Phi_j(y)\, \hat{u}_{i,j}$$

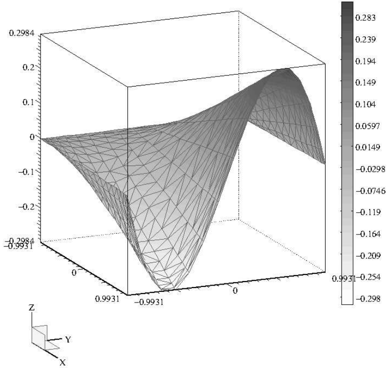
A nice basis for problems in Cartesian coordinates:

$$\begin{aligned}
\Phi_0(\xi) &= (1-\xi)/2 \\
\Phi_1(\xi) &= (1+\xi)/2 \\
\Phi_i(\xi) &= (1-\xi)^2/4 \;\; P_i^{1,1}(\xi)
\end{aligned}$$

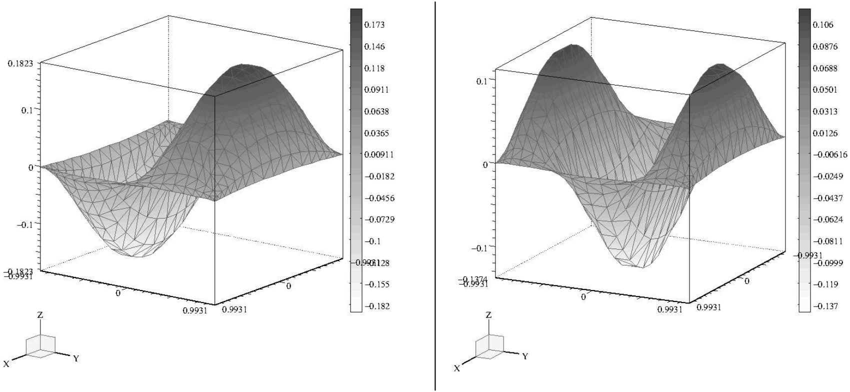
N.B.: spectral-element bases can be built in several ways according to the properties of the problem we want to solve.

Quadrilateral spectral elements: vertex and side modes

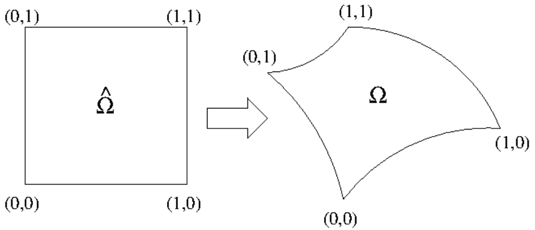




Quadrilateral spectral elements: internal modes



Operations: mapping from standard square to real element



Accuracy + generality $\Rightarrow$ isoparametric mapping

$$\begin{aligned} \boldsymbol{x} &= \Phi_0\Phi_0\hat{\boldsymbol{x}}_{0,0} + \Phi_1\Phi_0\hat{\boldsymbol{x}}_{1,0} + \Phi_0\Phi_1\hat{\boldsymbol{x}}_{0,1} + \Phi_1\Phi_1\hat{\boldsymbol{x}}_{1,1} \\ &+ \sum_i(\Phi_i\Phi_0\hat{\boldsymbol{x}}_{i,0} + \Phi_i\Phi_1\hat{\boldsymbol{x}}_{i,1}) + \sum_j(\Phi_0\Phi_j\hat{\boldsymbol{x}}_{0,j} + \Phi_1\Phi_j\hat{\boldsymbol{x}}_{1,j}) \end{aligned}$$

where the $\hat{\boldsymbol{x}}_{i,j}$ are the coefficients of the expansion of the boundaries.

Operations: quadrature

- Quadratures are performed numerically on standard square
- Use Gauss-type quadrature formulas
- Use (Legendre)-Gauss-(Lobatto) points and weights

Write:

$$\int_{\Omega} f(x,y)\, d\Omega = \int_{\hat{\Omega}} f(x(\xi,\eta), y(\xi,\eta)) \left| \frac{\partial(x,y)}{\partial(\xi,\eta)} \right| d\hat{\Omega}$$

$$= \sum_{h=1}^{h_G} \sum_{k=1}^{k_G} f(x(\xi_h,\eta_k), y(\xi_h,\eta_k)) \left| \frac{\partial(x,y)}{\partial(\xi,\eta)} \right|_{(\xi_h,\eta_k)} w_h w_k$$

Operations: forward transform

Unknowns, in HSE, are *not* the nodal values but the coefficients of the basis functions $\Rightarrow$ need to transform from the coefficients space to the physical space (forward) and viceversa (backward):

- Forward: multiply coefficients by basis functions computed at the selected point:

$$\mathcal{P}_N u(x,y) = \sum_{i=0}^{N} \sum_{j=0}^{M} \Phi_i(x)\, \Phi_j(y)\, \hat{u}_{i,j}$$

- Use sum–factorization technique to save computational cost when the function is needed at quadrature points:

$$\mathcal{P}_N u(x_m,y_n) = \sum_{i=0}^{N} \sum_{j=0}^{M} \Phi_i(x_m)\, \Phi_j(y_n)\, \hat{u}_{i,j} = \sum_{i=0}^{N} \Phi_i(x_m) \sum_{j=0}^{M} \Phi_j(y_n) \hat{u}_{i,j}$$

- Cost: naïf implementation: $O(N^2 M^2)$; sum factorization technique: $O(N^2 M + N M^2)$.

Operations: backward transform

- Transform from physical space to coefficient space: use Galerkin projection, i.e. $\forall h, k$:

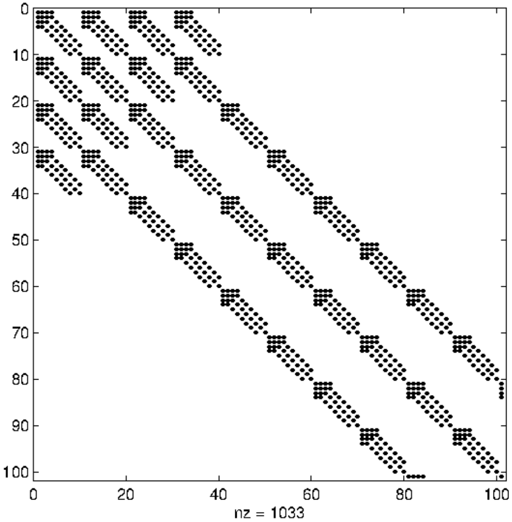
$$(\Phi_h(x)\, \Phi_k(y), \sum_{i=0}^{N} \sum_{j=0}^{M} \Phi_i(x)\, \Phi_j(y)\, \hat{u}_{i,j}) = (\Phi_h(x)\, \Phi_k(y), u(x, y))$$

- Solve a mass problem:

$$\boldsymbol{M}\hat{\boldsymbol{u}} = \boldsymbol{u}$$

where $\boldsymbol{M}_{m,n} = (\Phi_h(x)\, \Phi_k(y), \Phi_i(x)\, \Phi_j(y))$, $\hat{\boldsymbol{u}}_n = \hat{u}_{i,j}$, $\boldsymbol{u}_m = (\Phi_h(x)\, \Phi_k(y), u(x, y))$, $m = M * i + j$ and $n = M * h + k$

Mass operator pattern (rectangle)



Operations: derivative

Use the Jacobian matrix to compute real element derivatives from standard square ones:

$$\left\{ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{array} \right\} = \left[\frac{\partial(r,s)}{\partial(x,y)} \right] \left\{ \begin{array}{c} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \end{array} \right\} = \left[\frac{\partial(x,y)}{\partial(r,s)} \right]^{-1} \left\{ \begin{array}{c} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \end{array} \right\} =$$

Two strategies to compute $\partial/\partial r$ and $\partial/\partial s$:

1. In the physical space by Lagrange interpolation:

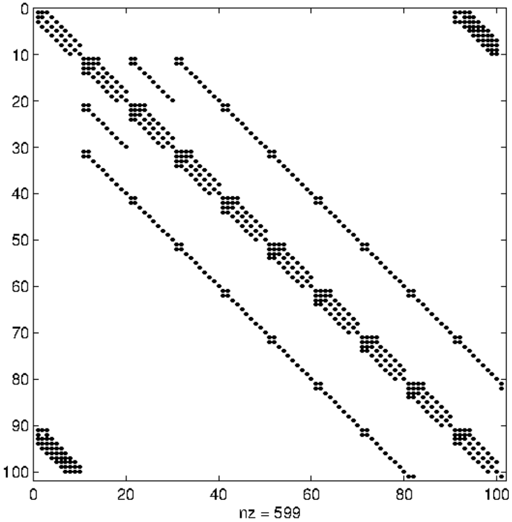
$$\partial u/\partial r|_{r_k,s_l} = \sum_i u_{i,l} \;\; h_i'(r)\big|_{r_k} ; \qquad \partial u/\partial s|_{r_k,s_l} = \sum_i u_{i,l} \;\; h_l'(s)\big|_{s_l}$$

2. In the transformed space by deriving the basis functions:

$$\partial u/\partial r = \sum_m \sum_n \hat{u}_{mn} \, \Phi_m'(r) \, \Phi_n(s); \; \partial u/\partial s = \sum_m \sum_n \hat{u}_{mn} \, \Phi_m(r) \, \Phi_n'(s)$$

use sum–factorization for the latter.

Stiffness-operator pattern (rectangle)



Triangular spectral elements: Dubiner basis

To allow a straightforward enforcement of continuity among different elements, the basis is built with three kind of modes:

1. Vertex modes:

$$g^{v-a} = \left(\frac{1-\zeta}{2}\right)\left(\frac{1-s}{2}\right); \quad g^{v-b} = \left(\frac{1+\zeta}{2}\right)\left(\frac{1-s}{2}\right); \quad g^{v-c} = \left(\frac{1+s}{2}\right)$$

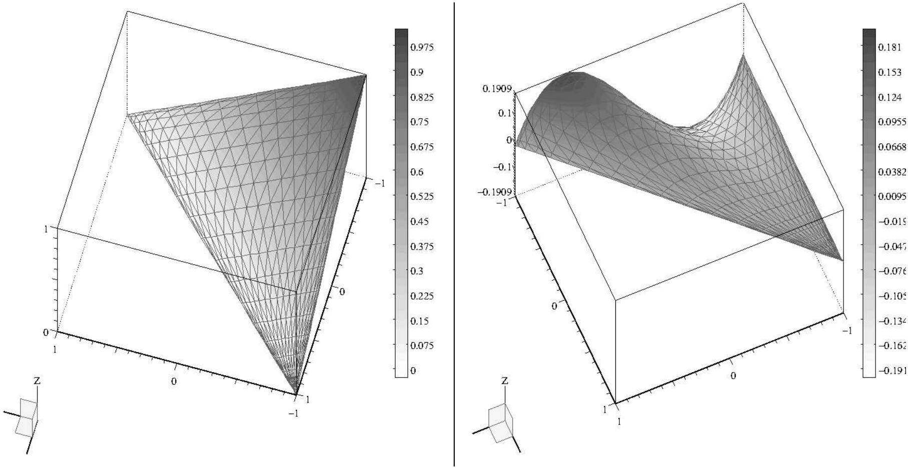
2. Side modes:

$$\begin{aligned}
g^{s-1}_{m0} &= \left(\frac{1+\zeta}{2}\right)\left(\frac{1-\zeta}{2}\right) P^{1,1}_{m-2}(\zeta)\left(\frac{1-s}{2}\right)^m, \quad 2 \le m < M \\
g^{s-2}_{1n} &= \left(\frac{1+\zeta}{2}\right)\left(\frac{1-s}{2}\right)\left(\frac{1+s}{2}\right) P^{1,1}_{n-1}(s), \quad 1 \le n < M-1 \\
g^{s-3}_{1n} &= \left(\frac{1-\zeta}{2}\right)\left(\frac{1-s}{2}\right)\left(\frac{1+s}{2}\right) P^{1,1}_{n-1}(s), \quad 1 \le n < M-1
\end{aligned}$$

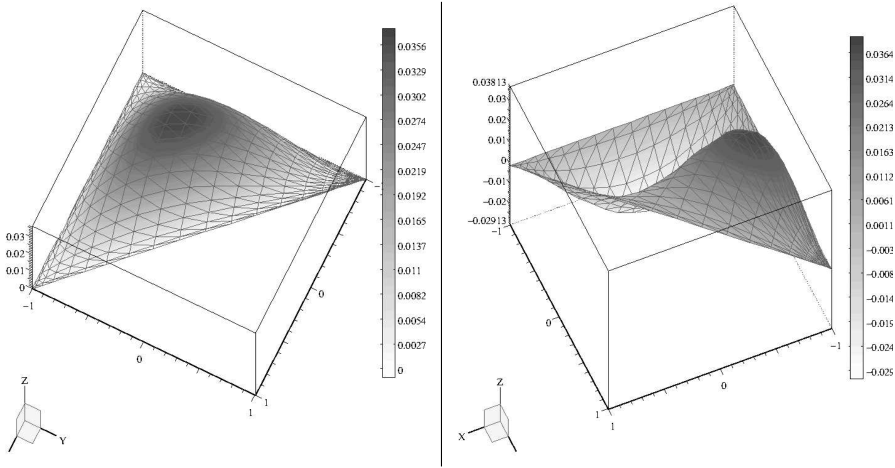
3. Internal modes: $2 \le m < M$; $1 \le n < M$; $m+n < M$

$$g^{i}_{mn} = \left(\frac{1-\zeta^2}{4}\right) P^{1,1}_{m-2}(\zeta)\left(\frac{1-s}{2}\right)^m \left(\frac{1+s}{2}\right) P^{2m-1,1}_{n-1}(s)$$

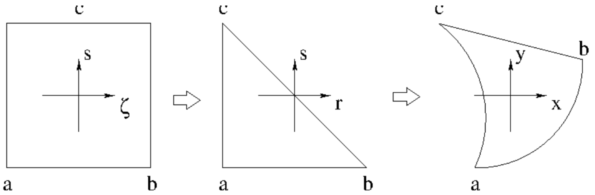
Triangular spectral elements: vertex and side modes



Triangular spectral elements: internal modes



Operations: coordinate transformation



1. Transformation from the square to the standard triangle:

$$r = \frac{(1+\zeta)(1-s)}{2} - 1; \quad s = s$$

2. Transformation from the standard triangle to the physical one: using an isoparametric mapping

$$\boldsymbol{x}(\zeta, s) = \sum_{i=1}^{3} \left(\boldsymbol{x}_{v-i}\, g^{v-i} + \sum_{j=1}^{M-1} \hat{\boldsymbol{x}}_j^{s-i}\, g_j^{s-i} \right)$$

Operations: quadrature

- Quadratures are performed numerically on standard square
- Use Gauss-type quadrature formulas
- Use Legendre/Jacobi-Gauss-(Lobatto/Radau) points and weights

$\Rightarrow$ L-G-(L) in ζ ($w^{0,0}$)

$\Rightarrow$ J-G-(L/R) with $\alpha = 1$ and $\beta = 0$ in s and halve weights ($\hat{w}^{1,0} = w^{1,0}/2$) to include the Jacobian $(1 - s)/2$

$$\int_\Omega f(x,y)\, d\Omega = \int_{\hat{\Omega}} f(x(\zeta,s), y(\zeta,s))(1-s)/2 \left| \frac{\partial(x,y)}{\partial(r,s)} \right| d\hat{\Omega}$$

$$= \sum_{h=1}^{h_G} \sum_{k=1}^{k_G} f(x(\zeta_h, s_k), y(\zeta_h, s_k)) \left| \frac{\partial(x,y)}{\partial(r,s)} \right|_{(\zeta_h, s_k)} w_h^{0,0}\, \hat{w}_k^{1,0}$$

Operations: forward transform

- Forward: multiply coefficients by basis functions computed at the selected point:

$$\mathcal{P}u(x,y) = \sum_{mn} \overset{1-2}{g}_{mn}(\zeta(x,y), s(x,y))\ \hat{u}_{mn}$$

- Use sum–factorization technique to save computational cost when the function is needed at quadrature points:

$$\sum_{mn} \overset{1-2}{g}_{mn}(\zeta, s)\ \hat{u}_{mn} = \sum_{m} \overset{1}{g}_{m}\ (\zeta) \sum_{mn}\ \overset{2}{g}_{mn}\ (s)\hat{u}_{mn}$$

- Cost: naïf implementation: $O(M^4)$; sum factorization technique: $O(M^3)$.

Operations: backward transform

- Transform from physical space to coefficients space: use Galerkin projection, i.e. $\forall h, k$:

$$\left(\overset{1-2}{g}_{hk}(\zeta, s), \sum_{mn} \overset{1-2}{g}_{mn}(\zeta, s)\, \hat{u}_{mn} \right) = \left(\overset{1-2}{g}_{hk}(\zeta, s), u(\zeta, s) \right)$$

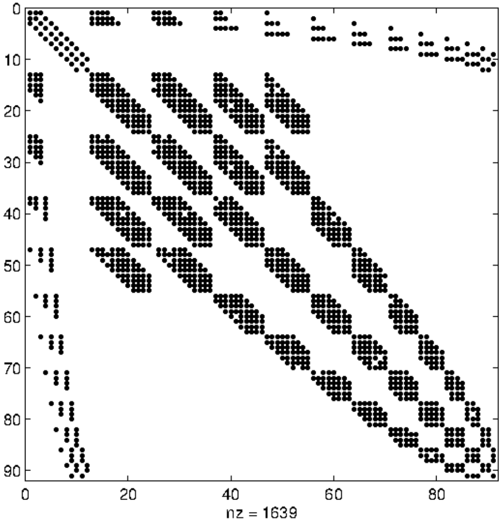
- Solve for a consistent mass problem:

$$\boldsymbol{M}\hat{\boldsymbol{u}} = \boldsymbol{u}$$

where $\boldsymbol{M}_{m,n} = \left(\overset{1-2}{g}_{hk}(\zeta, s), \overset{1-2}{g}_{ij}(\zeta, s) \right)$, $\hat{\boldsymbol{u}}_n = \hat{u}_{i,j}$,

$\boldsymbol{u}_m = (\overset{1-2}{g}_{hk}, u(x, y))$, $m = M * i + j$ and $n = M * h + k$

Mass-operator pattern (straight-sided triangle)



Operations: derivative computation

Use the Jacobian matrix to compute real-element derivatives from standard-triangle ones (as for quadrilateral elements):
Two strategies to compute $\partial/\partial r$ and $\partial/\partial s$:

1. In the physical space by Lagrange interpolation:

$$\left.\frac{\partial u(r_k, s_l)}{\partial r}\right|_s = \sum_i \frac{2}{1-s_l} u_{il} \left.\frac{\partial h_i^{0,0}(\zeta)}{\partial \zeta}\right|_{\zeta_k}$$

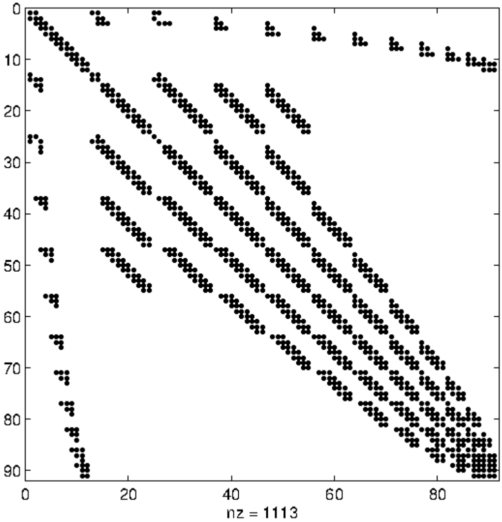
and:

$$\left.\frac{\partial u(r_k, s_l)}{\partial s}\right|_r = \sum_i \frac{1+\zeta_k}{1-s_l} u_{il} \left.\frac{\partial h_i^{0,0}(\zeta)}{\partial \zeta}\right|_{\zeta_k} + \sum_j u_{kj} \left.\frac{\partial h_j^{1,0}(s)}{\partial s}\right|_{s_l}$$

2. In the transformed space by deriving the basis functions:

$$\left.\frac{\partial u}{\partial r}\right|_s = \sum_m \sum_n \frac{2}{1-s} \hat{u}_{mn} \frac{\partial \overset{1}{g}_m(\zeta)}{\partial \zeta} \overset{2}{g}_{mn}(s)$$

Stiffness-operator pattern (straight-sided triangle)



Connection between elements and adaptivity

- These are conformal elements, i.e. the basis on a multi-element domain is continuous, provided that degrees of side modes on the coincident sides are the same
- Full adaptivity is allowed *without* loosing conformity
- Polymorphic tessellations (of triangles and quadrilaterals) are possible again *without* loosing conformity

Outline

Applications: advection-diffusion equation

$$\frac{\partial \phi}{\partial t} + \vec{U} \cdot \boldsymbol{\nabla}\, \phi - \boldsymbol{\nabla} \cdot (\nu\, \boldsymbol{\nabla} \phi) = f$$

supplemented with initial: $\phi(x, y, 0) = \phi_0(x, y)$ and boundary conditions:

$$\phi|_{\Gamma_D} = a, \quad (\nabla \phi \cdot \vec{n})\big|_{\Gamma_N} = b$$

Variational formulation: *Given* $f \in L^2\{(0, T) \times \Omega\}$ *and* $\phi_0 \in L^2(\Omega)$*, find* $u \in L^2\{(0, T); H^1_D(\Omega)\} \cap C^0([0, T]; L^2(\Omega))$ *such that:*

$$\frac{d}{dt}(\phi(t), v) + a(\phi(t), v) = f((t), v) \quad \forall v \in H^1_D(\Omega); \quad \phi(0) = \phi_0$$

Discretized form

⇒ Using an Euler-implicit scheme for diffusion and an Euler explicit scheme for advection:

$$(\phi_h^{i+1}, v_h) + a_d(\phi_h^{i+1}, v_h) = (f^{i+1}, v_h) - a_c(\phi_h^i, v_h); \quad \phi_h^0 = \phi_{0,h}$$

Introducing spectral element basis obtain:

$$([M] + [K])\{\phi^{i+1}\} = \{f^{i+1}\} - [C^i]\{\phi^i\}$$

Discretization in space

To obtain the global system first build local operators:

- Mass: $\boldsymbol{M}_{loc} = \boldsymbol{G}^T\ \boldsymbol{W}\ \boldsymbol{J}\ \boldsymbol{G}$;
- Stiffness: $\boldsymbol{K}_{loc} = \boldsymbol{G}_{/x}^T\ \boldsymbol{W}\ \boldsymbol{J}\ \boldsymbol{G}_{/x} + \boldsymbol{G}_{/y}^T\ \boldsymbol{W}\ \boldsymbol{J}\ \boldsymbol{G}_{/y}$;

Then build local contribution to the rhs:

$$\boldsymbol{F}_{loc} = \boldsymbol{G}^T\ \boldsymbol{W}\ \boldsymbol{J}\ \boldsymbol{f}^{i+1} - \boldsymbol{G}^T\ \boldsymbol{W}\ \boldsymbol{J}\ (\boldsymbol{U}_x\ \boldsymbol{G}_{/x} + \boldsymbol{U}_y\ \boldsymbol{G}_{/y})\{\phi^i\}$$

Use sum-factorization for efficiency.

Coefficient matrix assembling

Assembling the global matrix: add the coefficients of the local operators in the global matrix at the right place.
A simple algebraic interpretation:

$$\boldsymbol{A}_G = \boldsymbol{Z}^T \boldsymbol{A}_l \boldsymbol{Z}$$

where: $\boldsymbol{Z}$ is a rectangular matrix with as many rows as the total number of local degrees of freedom (dof) and as many columns as the number of unknowns. Each line contains a 1 in the row associated with the dof and column corresponding to the unknown associated with that dof. $\boldsymbol{A}_l$ is a block diagonal matrix containing local operators on the diagonal.
Global operators are stored in compressed form!

Boundary conditions

$\Rightarrow$ Dirichlet boundary conditions: essential boundary conditions are treated according the standard variational formulation:

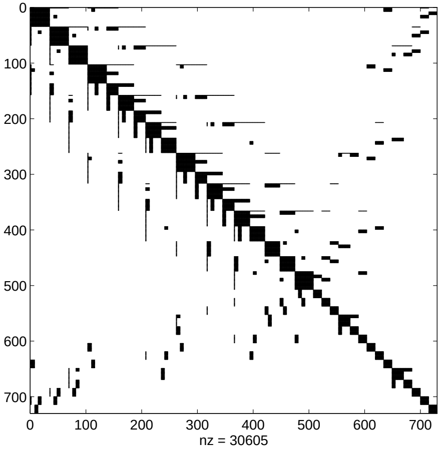
- Eliminate the unknowns related to Dirichlet boundaries
- Compute a discrete trace lifting of the Dirichlet datum (this allows us to determine the aforementioned unknowns)
- Perturb the right hand side of the system by this lifting

The lifting is determined by a mixed approach: vertex modes are computed by collocation, side modes by a Galerkin orthogonal projection.

$\Rightarrow$ Neumann boundary conditions: simply add to the rhs the contribution due to boundary integrals:

$$\int_{\partial\Omega} v \; \partial\phi/\partial n \; d(\partial\Omega)$$

Linear system solution



Coefficient matrix does *not* depend on time:

- $\Rightarrow$ Factor it once and for all
- $\Rightarrow$ Backward substitution at every time step

Factorization performed by a multifrontal method.
Other kind of iterative techniques have been employed.

Cost and memory requirement

- cost of the algorithm resides mainly in the the solution of linear systems.
- cost of building the right hand side is undoubtedly greater than for low order methods, but it is under control thanks to the sum factorization technique.
- The memory requirement is also a major drawback of the present method since the factorization must be stored. It is almost unaffordable for 3-D cases $\rightarrow$ multigrid.

Applications: Navier-Stokes I.B.V.P.

For a given body force $\boldsymbol{f}$ (possibly dependent on time) and a prescribed divergence-free initial velocity field $\boldsymbol{u}_0$, find a velocity field $\boldsymbol{u}$ and a pressure field p (per unit density) so that at $t = 0$, $\boldsymbol{u} = \boldsymbol{u}_0$, and at all subsequent times

$$\mathcal{P} \begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} - \nu \nabla^2 \boldsymbol{u} + (\boldsymbol{u} \cdot \boldsymbol{\nabla}) \boldsymbol{u} + \boldsymbol{\nabla} p = \boldsymbol{f}, \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \end{cases}$$

$$\begin{cases} \boldsymbol{u}_{|\partial\Omega_1} = \boldsymbol{b}_1; & \\ \boldsymbol{u} \cdot \boldsymbol{n}_{|\partial\Omega_2} = \boldsymbol{b}_2 \cdot \boldsymbol{n}, & (\alpha\, \boldsymbol{n} \times \boldsymbol{u} + \boldsymbol{\nabla} \times \boldsymbol{u}) \times \boldsymbol{n}_{|\partial\Omega_2} = 0; \\ \boldsymbol{u} \times \boldsymbol{n}_{|\partial\Omega_3} = \boldsymbol{b}_3 \times \boldsymbol{n}, & p_{|\partial\Omega_3} = c_3; \end{cases}$$

Variational formulation

For $\boldsymbol{f} \in W^{2,\infty}(0,T;L^2(\Omega)^d)$ and $\boldsymbol{u}_0 \in V \cap \boldsymbol{H}^2(\Omega)^d$ find a pair $(\boldsymbol{u}, p)$, $\boldsymbol{u} \in L^\infty(0,T;\boldsymbol{H}) \cap L^2(0,T;\boldsymbol{U})$, $\boldsymbol{u}_t \in L^2(0,T;\boldsymbol{H}^{-1})$, $p \in L^2(0,T;M)$, $\forall t > 0$

$$\begin{cases} \left(\dfrac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v}\right) + a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{v}) - (p, \nabla\cdot \boldsymbol{v}) \\ \qquad\qquad = (\boldsymbol{f}, \boldsymbol{v}) - \displaystyle\int_{\partial\Omega_3} c_3\ \boldsymbol{v}\cdot\boldsymbol{n}, & \forall \boldsymbol{v} \in \boldsymbol{X}_0, \\ (\nabla\cdot \boldsymbol{u}, q) = 0, & \forall q \in M. \end{cases}$$

with: $\boldsymbol{X} = \{\boldsymbol{v} \in \boldsymbol{H}^1(\Omega)\ \ \boldsymbol{v}_{|\partial\Omega_1} = 0,\ \boldsymbol{v}\cdot\boldsymbol{n}_{|\partial\Omega_2} = 0,\ \boldsymbol{v}\times\boldsymbol{n}_{|\partial\Omega_3} = 0\}$,
$V = \{\boldsymbol{v} \in X, \nabla\cdot \boldsymbol{v}\}$, $M = L^2(\Omega)$, $\boldsymbol{H} = \{\boldsymbol{v} \in L^2(\Omega)^d, \nabla\cdot \boldsymbol{v}\}$

Projection method in discretized form

Viscous step:

For $l \geq 1$, find $\boldsymbol{u}_h^{l+1} \in \boldsymbol{X}_{\boldsymbol{b}^{l+1},h}$ such that, $\forall\ \boldsymbol{v}_h \in \boldsymbol{X}_{0,h}$,

$$\left(\frac{\boldsymbol{u}_h^{l+1} - \boldsymbol{u}_h^l}{\delta t}, \boldsymbol{v}_h \right) + a(\boldsymbol{u}_h^{l+1}, \boldsymbol{v}_h) = (\boldsymbol{f}^{l+1}, \boldsymbol{v}_h) - b(\boldsymbol{u}_h^l, \boldsymbol{u}_h^l, \boldsymbol{v}_h) - (\boldsymbol{\nabla}(2p_h^l - p_h^{l-1}), \boldsymbol{v}_h).$$

Projection step:

For $l \geq 0$, find $p_h^{l+1} \in N_{c_{3,h}^{l+1},h}$ such that, $\forall\ q_h \in N_{0,h}$,

$$(\boldsymbol{\nabla}(p_h^{l+1} - p_h^l), \boldsymbol{\nabla} q_h) = -\frac{(\boldsymbol{\nabla} \cdot \boldsymbol{u}_h^{l+1}, q_h)}{\delta t}.$$

where $N_{0,h} \equiv M_h \subset L^2(\Omega)$ is an internal approximation to $H^1_{0,\partial\Omega_3}(\Omega)$ and similarly for $N_{c_{3,h}^{l+1},h}$.

Algebraic problem

At each time step one has to solve three linear systems, two for the velocity components and one for the pressure:

$$(\boldsymbol{M} + \boldsymbol{K})\boldsymbol{u}_x = \boldsymbol{f}_x$$

$$(\boldsymbol{M} + \boldsymbol{K})\boldsymbol{u}_y = \boldsymbol{f}_y$$

$$\boldsymbol{K}p = \boldsymbol{f}_p$$

Remark: since the convective term is advanced explicitly the method is only conditionally stable $\Rightarrow$ CFL condition on the time step.

Solution strategy

- Also in this case the coefficient matrices do not depend on time step
- It is possible to factor them once and for all at the beginning
- Good efficiency but quite memory consuming

Outline

Some references

- P. Fischer, *High-order methods for incompressible fluid flow*, Cambridge University Press, 2002.
- G. Karniadakis & S. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, 1999.