B-Splines Finite element and Physic-Based preconditioning for Tokamak Plasma

E. Franck[1], M. Gaja[2], J.Lakhlili[2], M. Mazza[2],
A. Ratnani[2], S. Serra-Capizzano[3], E. Sonnendrücker[2]

[1]Inria Nancy Grand Est and IRMA Strasbourg, France
[2]Max-Planck-Institut für Plasmaphysik, Garching, Germany
[3]University of Insubria, Como, Italy

Outline

Mathematical and physical context

Time discretization

Solver for simple operators

Plasma Physics

- **Thermonuclear fusion**: Nuclear reaction between deuterium and tritium (high energy physic phenomena), which can generate energy. For these very high temperatures, the gas are ionized and gives a plasma.
- **Tokamak** : The plasma is confined in a toroidal room (Tokamak) by powerful magnetic field.
- In the Tokamak some instabilities can appear in the plasma. The simulation of these instabilities is an important subject for ITER.
- The instabilities like ELM's (periodic edge instabilities) are linked to the very large gradient of pressure and very large current at the edge.

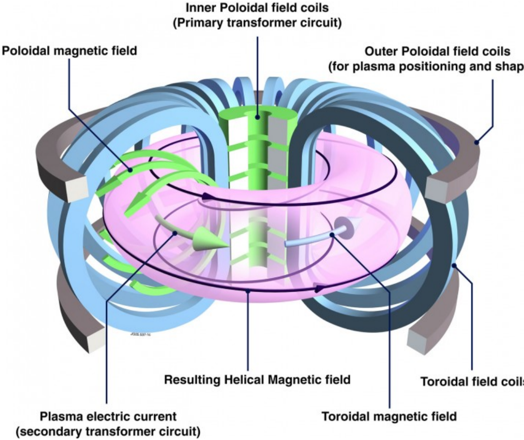
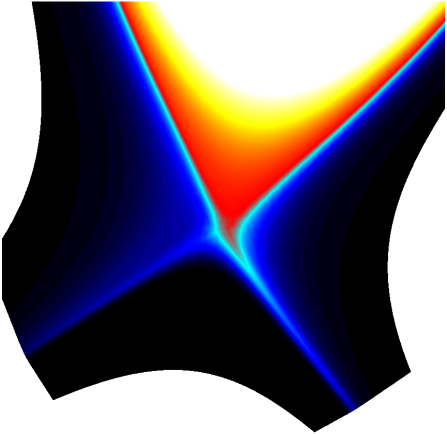


Figure: Tokamak

Plasma Physics

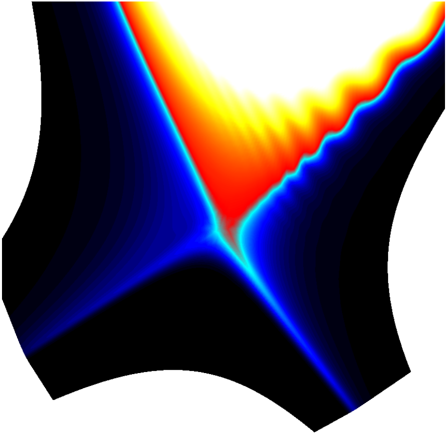
- **Thermonuclear fusion**: Nuclear reaction between deuterium and tritium (high energy physic phenomena), which can generate energy. For these very high temperatures, the gas are ionized and gives a plasma.
- **Tokamak** : The plasma is confined in a toroidal room (Tokamak) by powerful magnetic field.
- In the Tokamak some instabilities can appear in the plasma. The simulation of these instabilities is an important subject for ITER.
- The instabilities like ELM's (periodic edge instabilities) are linked to the very large gradient of pressure and very large current at the edge.

- ELM simulation



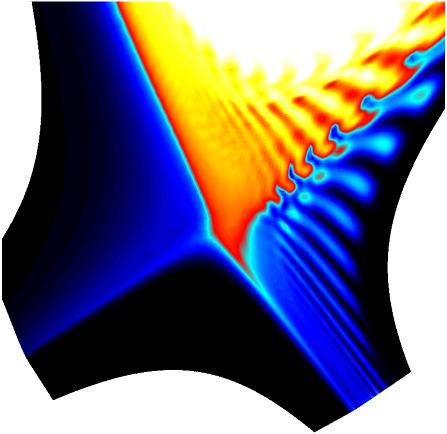
Plasma Physics

- **Thermonuclear fusion**: Nuclear reaction between deuterium and tritium (high energy physic phenomena), which can generate energy. For these very high temperatures, the gas are ionized and gives a plasma.
- **Tokamak** : The plasma is confined in a toroidal room (Tokamak) by powerful magnetic field.
- In the Tokamak some instabilities can appear in the plasma. The simulation of these instabilities is an important subject for ITER.
- The instabilities like ELM's (periodic edge instabilities) are linked to the very large gradient of pressure and very large current at the edge.



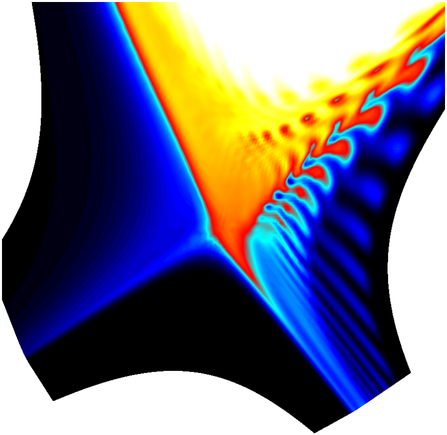
Plasma Physics

- **Thermonuclear fusion**: Nuclear reaction between deuterium and tritium (high energy physic phenomena), which can generate energy. For these very high temperatures, the gas are ionized and gives a plasma.
- **Tokamak** : The plasma is confined in a toroidal room (Tokamak) by powerful magnetic field.
- In the Tokamak some instabilities can appear in the plasma. The simulation of these instabilities is an important subject for ITER.
- The instabilities like ELM's (periodic edge instabilities) are linked to the very large gradient of pressure and very large current at the edge.



Plasma Physics

- **Thermonuclear fusion**: Nuclear reaction between deuterium and tritium (high energy physic phenomena), which can generate energy. For these very high temperatures, the gas are ionized and gives a plasma.
- **Tokamak** : The plasma is confined in a toroidal room (Tokamak) by powerful magnetic field.
- In the Tokamak some instabilities can appear in the plasma. The simulation of these instabilities is an important subject for ITER.
- The instabilities like ELM's (periodic edge instabilities) are linked to the very large gradient of pressure and very large current at the edge.



Equilibrium

Shift

- Tokamak equilibrium ($\boldsymbol{u} = 0$):

$$\boldsymbol{J} \times \boldsymbol{B} = \nabla p$$

- In a Tokamak we assume that

$$\boldsymbol{B} = \mu_0 \frac{F(\psi, Z)}{R}\mathbf{e}_\phi + \frac{1}{R}(\nabla\psi \times \mathbf{e}_\phi)$$

- Equation defining the equilibrium : Grad-Shafranov

$$\Delta^*\psi = -\mu_0 R^2 \frac{dp(\psi)}{d\psi} - \mu_0^2 F(\psi)\frac{dF(\psi)}{d\psi}$$

with

$$\Delta^*\psi = R^2\partial_R\left(\frac{1}{R^2}\partial_R\psi\right) + \partial_{ZZ}\psi$$

- **Instabilities study**: perturbation of the axisymmetric equilibrium.

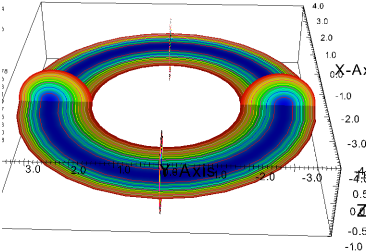


Figure: 3D equilibrium

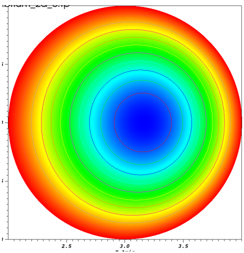


Figure: poloidal cut of equilibrium

MHD model

Single fluid resistive MHD

$$
\begin{cases}
\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0, \\
\rho \partial_t \boldsymbol{u} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p = \boldsymbol{J} \times \boldsymbol{B} - \nabla \cdot \overline{\overline{\boldsymbol{\Pi}}}, \\
\partial_t p + \boldsymbol{u} \cdot \nabla p + p \nabla \cdot \boldsymbol{u} + \nabla \cdot (K \nabla T) = 0 \\
\partial_t \boldsymbol{B} = -\nabla \times (-\boldsymbol{u} \times \boldsymbol{B} + \eta \boldsymbol{J}), \\
\nabla \cdot \boldsymbol{B} = 0, \quad \nabla \times \boldsymbol{B} = \boldsymbol{J}.
\end{cases}
$$

Spatial discretization

- **Parabolic problems with free-divergence** ===> Compatible Finite element methods.
- **Strongly anisotropic problem** ===> high-order methods and aligned grids.

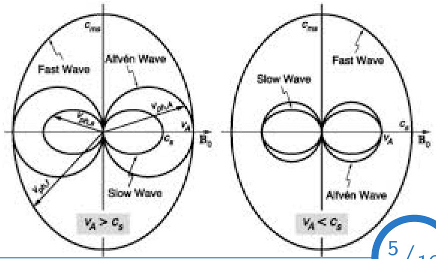
Time problem

- low Mach and Low β regime

$$0 \approx \| \mathbf{u} \| << c << V_a$$

with $c = \sqrt{T}$ and $V_a = \frac{|\boldsymbol{B}|}{\sqrt{\rho \mu_0}}$.

- Direction of $\boldsymbol{B}$: $\lambda_{min} \approx \| \mathbf{u} \| << \lambda_{\max} \approx V_a$
- Direction of $\boldsymbol{B}^{\perp}$: $\lambda_{min} \approx \| \mathbf{u} \| << \lambda_{\max} \approx c$



Exemple of Anisotropic problem: diffusion

- **Model** :

$$\partial_t T - \nabla \cdot \Big((k_\parallel - k_\perp)(\mathbf{b} \otimes \mathbf{b}) \nabla T + k_\perp \nabla T \Big) = 0$$

  with $k_\parallel >> k_\perp$.

- The magnetic field is construct solving the equilibrium.
- In this case $k_\parallel = 100$ and $k_\perp = 0$ .The diffusion is along the magnetic lines.

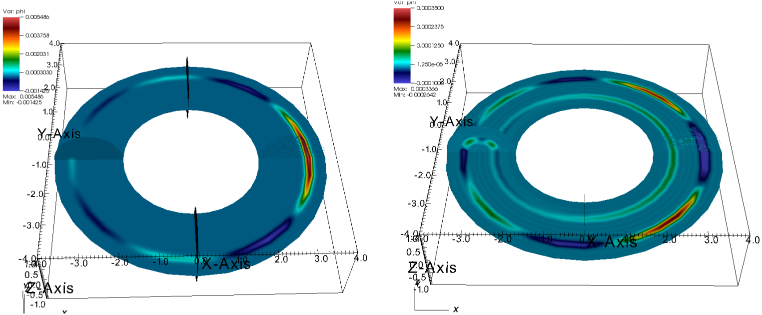


Figure: Left: solution after time $T = 0.5$. Right: solution after time $T = 7$

- **Aligned grids**: the actual physic code aligne the poloidal grid with the magnetic surfaces. In the future we want 3D meshes aligned to magnetic lines.

Euler linearized and implicit scheme

- We solve hyperbolic systems with small diffusion using implicit schemes.
- **Ill-conditioned systems when** $\Delta t >> 1$ since
  - $\frac{\lambda_{min}}{\lambda_{max}} >> 1$ in the Jacobian,
  - $\lambda_{min} \approx 0$ in the Jacobian.

Idea

- **Idea**: Use operator splitting and reformulation to approximate the Jacobian by a series of suitable simple problems (advection, diffusion or mass problems).

Linearized Euler equation

$$\begin{cases} \frac{1}{c}\partial_t \boldsymbol{u} + M\boldsymbol{a}\cdot\nabla \boldsymbol{u} + \nabla p & = 0 \\ \frac{1}{c}\partial_t p + M\boldsymbol{a}\cdot\nabla p + \nabla\cdot\boldsymbol{u} & = 0 \end{cases}$$

with $M \in]0,1]$, and $\mid \mathbf{a} \mid = 1$.

- Implicit problem after time discretization:

$$\begin{pmatrix} I_d + M\lambda\boldsymbol{a}\cdot\nabla & \lambda\nabla\cdot \\ \lambda\nabla & I_d + M\lambda\boldsymbol{a}\cdot\nabla \end{pmatrix}\begin{pmatrix} p^{n+1} \\ \mathbf{u}^{n+1} \end{pmatrix} = \begin{pmatrix} I_d - M\lambda\boldsymbol{a}\cdot\nabla & \lambda\nabla\cdot \\ \lambda\nabla & I_d - M\lambda\boldsymbol{a}\cdot\nabla \end{pmatrix}\begin{pmatrix} p^{n} \\ \mathbf{u}^{n} \end{pmatrix}$$

- with $\lambda = 0.5c\Delta t$ the numerical acoustic length.

Schur preconditioning method

- **Example of Algorithm** : Schur preconditioning.
- The implicit system after linearization is given by

$$\begin{pmatrix} p^{n+1} \\ \mathbf{u^{n+1}} \end{pmatrix} = \begin{pmatrix} A_{\mathbf{B},p} & Div \\ Grad & A_{\mathbf{u}} \end{pmatrix}^{-1} \begin{pmatrix} R_p \\ R_{\mathbf{u}} \end{pmatrix}$$

- with A_p and $A_{\mathbf{u}}$ the advection terms linked to p (resp $\mathbf{u}$), Div and Grad the coupling terms which gives the acoustic waves.
- Applying the Schur decomposition we obtain

$$\begin{pmatrix} p^{n+1} \\ \mathbf{u^{n+1}} \end{pmatrix} = \begin{pmatrix} I_d & A_{\mathbf{B},p}^{-1} Div \\ 0 & I_d \end{pmatrix} \begin{pmatrix} A_p^{-1} & 0 \\ 0 & P_{schur}^{-1} \end{pmatrix} \begin{pmatrix} I_d & 0 \\ -Grad A_p^{-1} & I_d \end{pmatrix} \begin{pmatrix} R_p \\ R_{\mathbf{u}} \end{pmatrix}$$

- Using the previous Schur decomposition, we obtain the following algorithm:

$$\begin{cases} \text{Predictor :} \quad A_p p^* = R_p \\ \text{Velocity evolution :} \quad P_{schur}\mathbf{u}^{n+1} = (-Grad\, p^{n+1} + R_{\mathbf{u}}) \\ \text{Corrector :} \quad A_p p^{n+1} = A_p p^* - Div\, \mathbf{u}_{n+1} \end{cases}$$

- with $P_{schur} = A_u - Grad(A_p^{-1})Div \approx A_u - GradDiv$. The approximation is valid in the low Mach regime.

Numerical results

- First test case. We compare the PC for different mesh and different time step.
- $c = 1$ and $\mathbf{a} = (0, 0)$. Number of iteration to converge :

	Gmres			Gmres + PBPC		
	$\Delta t = 0.1$	$\Delta t = 0.5$	$\Delta t = 1$	$\Delta = 0.1$	$\Delta = 0.5$	$\Delta = 1$
64*64	25	4000	1.0E+5	4	35	60
128*128	30	7800	2.0E+5	4	50	75

- The method allows to reduce the number of iteration to converge. The method is efficient if the sub-steps are treat efficiently.
- The algorithm depend of the boundary conditions. Additional optimization mus be add.
- Now we study the Mach dependency. We take a mesh 64*64 and $\Delta t = 0.25$

Mach	$M = 10^{-5}$	$M = 10^{-3}$	$M = 10^{-2}$	$M = 10^{-1}$	$M = 0.5$
	10	11	12	35	80

- **Conclusion** : the algorithm is less efficient for Mach around one, since the approximation of the Schur complement is less good.

Simple operators

- Applying the algorithm in time (Schur preconditioning or other splitting and reformulation methods) we obtain simples operator to solve

$$A = I_d + M\lambda \mathbf{a}\cdot\nabla, \qquad L = I_d + \lambda\Delta, \qquad D_d = I_d + \lambda\nabla(\nabla\cdot), \qquad D_c = I_d + \lambda\nabla\times(\nabla\times)$$

- with $M << 1$ and $\lambda >> 1$.
- **Numerical problems** :
  - At the limit $\lambda >> 1$, D_d and D_c have a infinite dimensional kernel. Therefore the operators are ill-conditioned for large λ.
  - Numerical example for D_d with 3-order Hdiv B-Splines

λ / size mesh	32*32	64*64	128*128
$\lambda = 0.01$	480	1060	3000
$\Delta t = 0.1$	2250	7500	14000
$\Delta t = 1$	7500	29000	112000
$\Delta t = 10$	27000	280000	nc

  - When the polynomial ordre is large all the operators are ill-conditioned.
  - Advection diffusion problem with $M = 0.1$, $\lambda = 1$ (Gmres + Jacobi) :

λ / size mesh	$p = 3$	$p = 5$	$p = 7$	$p = 9$
Mesh $32 * 32$	60	260	2200	70000

GLT principle

- PDE : $Lu = g$ after discretization gives $L_n \mathbf{u}_n = \mathbf{g}_n$ with $\{L_n\}_n$ a sequence of matrices.
- It is often the case that the matrix L_n is a **linear combination, product, inversion or conjugation** of these two simple kinds of matrices
  - $T_n(f)$, i.e., a Toeplitz matrix obtained from the Fourier coefficient of $f : [-\pi, \pi] \rightarrow C$, with $f \in L^1([-\pi, \pi])$.
  - $D(a)$, i.e., a diagonal matrix such that $(D_n(a))_{ii} = a(\frac{i}{n})$ with $a : [0, 1] \rightarrow C$ Riemann integrable function.

  In such a case $\{L_n\}_n$ is called a **GLT sequence**.

Fundamental property

- Each GLT sequence $\{L_n\}_n$ is equipped with a "symbol", a function $\chi : [0, 1] \times [-\pi, \pi] \rightarrow C$ which describes the asymptotical spectral behaviour of $\{L_n\}_n$:

$$\{L_n\}_n \sim \chi$$

E.g.: if $L_n = D_n(a) T_n(f)$, then $\{L_n\}_n \sim \chi = a \cdot f$

- **Advantage of this tool**: studying the symbol we retrieve information on the conditioning and propose new preconditioning based on this symbol.

GLT for stiffness matrix

- **Application**: B-Splines discretization of the model

$$-\Delta u = f, \quad \text{in } [0,1]^d.$$

- The basis functions are given by $\phi_i(\mathbf{x})$ a tensor product of 1D B-Splines functions.

Symbol of the problem

$$\left\{ n^{d-2} L_n \right\}_n \sim \frac{1}{n} \left(\mathbf{\Pi}_{k=1}^{d} m_{p_k-1}(\theta_k) \right) \left(\sum_{k=1}^{d} \mu_k^2 (2 - 2\cos(\theta_k)) \mathbf{\Pi}_{j=1, j\neq k}^{d} w_{p_j}(\theta_j) \right)$$

with $\theta_k \in [-\pi, \pi]$ and $w_p(\theta) := m_p(\theta)/m_{p-1}(\theta)$.

- $\left(\frac{4}{\pi^2}\right)^p \leq m_{p-1}(\theta) \leq m_{p-1}(0) = 1.$
- **Remark 1**: The symbol has a zero in $\theta = (0, \dots, 0) \Rightarrow n^{d-2} L_n$ is ill-conditioned in the low frequencies. Classical problem solved by MG preconditioning.
- **Remark 2**: The symbol has infinitely many exponential zeros at the points θ with $\theta_j = \pi$ for some j when $p_j \to \infty \Rightarrow n^{d-2} L_n$ is ill-conditioned in the high frequencies. Non-canonical problem solvable by GLT theory.
- **Preconditioning**: Using the symbol we can construct a smoother for MG valid for high-frequencies. (i.e. CG preconditioned with a Kronecker product whose jth factor is $T_{\mu_j n + p_j - 2}(m_{p_j-1})$).
- **Extension**: the method can be extended to the case with mapping (general geometries) and more general operators.

Numerical results for GLT

GLT for curl-curl problem: a 2D example

- **Application**: compatible B-Splines discretization based on a discrete De Rham sequence of the variational problem:

Find $\boldsymbol{u} \in H(curl, [0,1]^2)$ such that

$$(\nabla \times \boldsymbol{u}, \nabla \times \boldsymbol{v}) + \nu\,(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}), \quad \forall \boldsymbol{v} \in H(curl, [0,1]^2),$$

where $\nu \geq 0$ and $H(curl, [0,1]^2) := \{\boldsymbol{u} \in (L^2([0,1]^2))^2 \text{ s.t. } \nabla \times \boldsymbol{u} \in L^2([0,1]^2)\}$.

- **Coefficient matrix** $\mathcal{A}_{\boldsymbol{n}}^{\nu}$: is a 2×2 block matrix.
- **Spectral symbol** f^{ν}:
  - 2D problem $\Rightarrow$ f^{ν} is bivariate;
  - vectorial problem $\Rightarrow$ f^{ν} is 2×2 matrix-valued function. In such cases, we have to look at the eigenvalue functions of f^{ν}.

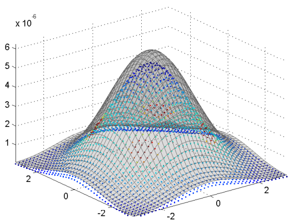
$$\lambda_1\left(f^{\nu}(\theta_1, \theta_2)\right) \approx \frac{1}{\mu_1 \mu_2} m_{p-1}(\theta_1) m_{p-1}(\theta_2) \frac{\nu}{n^2}$$

$$\lambda_2\left(f^{\nu}(\theta_1, \theta_2)\right) \approx \frac{1}{\mu_1 \mu_2} m_{p-1}(\theta_1) m_{p-1}(\theta_2) \left[\mu_2^2(2 - 2\cos(\theta_2)) + \mu_1^2(2 - 2\cos(\theta_1)) + \frac{\nu}{n^2}\right]$$

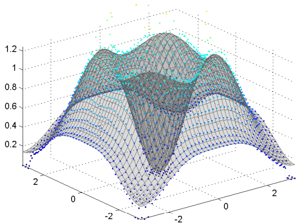
- **Continuum**: the curl-curl operator has infinite dimensional kernel and on the complement behaves as a second order operator.
- **Spectral counterpart**: when $\nu = 0$, $\lambda_1(f^{\nu}) \equiv 0$, while $\lambda_2(f^{\nu})$ is the symbol of the 2D Laplacian operator.

GLT for curl-curl problem

- **How to use our spectral analysis?**: an equispaced sampling of the eigenvalues functions in $[-\pi, \pi]^2$ gives an approximation of the eigenvalues of $\mathcal{A}_n^\nu$.



$\lambda_1(f^\nu)$



$\lambda_2(f^\nu)$

Comparison between the eigenvalues of $\mathcal{A}_{\boldsymbol{n}}^\nu$ (colored dots) and $\lambda_k(f^\nu)$, $k = 1, 2$, when $n = 40$, $p = 3$, $\nu = 10^{-2}$ (matrix-size 3612).

- **As for IgA stiffness matrices**: $\lambda_k(f^\nu)$, $k = 1, 2$ satisfy the following properties
  - for $\nu = 0$, $\lambda_2(f^\nu)$ has an analytic zero in $(\theta_1, \theta_2) = (0, 0)$ of order 2;
  - both $\lambda_1(f^\nu)$ and $\lambda_2(f^\nu)$ possess infinitely many numerical exponential zeros at the points (θ_1, θ_2) with $\theta_j = \pi$ when p becomes large.
- **Solver proposal**: Using the symbol we can construct a smoother for MG valid for high-frequencies. (i.e. CG preconditioned with a direct sum of Toeplitz matrices generated by the mass symbol $m_{p-1}(\theta_1)m_{p-1}(\theta_2)$).

Conclusion

Conclusion about time-scheme

- **Schur preconditioning**: Very efficient in the low-mach regime. Less when the mach is close to one.
- **Other possibilities**:
  - Coupling implicit acoustic scheme (with Schur pc) and explicit transport.
  - Linearization and decoupled approximate model adding variables and using splitting.
- **General remark**: these algorithms are efficient if we have efficient solvers for simple models.

Conclusion about simple solvers

- **GLT**: the method allows to understand the problem of conditioning linked to different operators discretized with B-Splines.
- **GLT + MG**: the method allows to design smoother for Multi-grids methods for these operators.
- **Vectorial elliptic operators**: coupling GLT and auxiliary spaces method allows to design solver for div-div and curl-curl operators.