10th FreeFem++ days 2018, 13 Dec.

Dissection sparse direct solver for indefinite finite element matrices and application to a semi-conductor problem

Atsushi Suzuki[1]

[1] Cybermedia Center, Osaka University
atsushi.suzuki@cas.cmc.osaka-u.ac.jp

joint work with
François-Xavier Roux, ONERA, LJLL Sorbonne Université

Outline

Dissection solver for indefinite matrix

- overview of nested-dissection algorithm for sparse matrix
- LDU factorization with symmetric pivoting
- computation of a solution of singular linear system
- extension with 2x2 pivoting and unsymmetric pivoting
- kernel detection algorithm

semi-conductor problem

- mixed finite element formulation of a semi-conductor problem
- Newton iteration to find a solution of stationary problem
- example of 3D computation

abstract framework

V: Hilbert space with inner product $(\cdot,\cdot)$ and norm $||\cdot||$.
bilinear form $a(\cdot,\cdot) : V \times V \to \mathbb{R}$

- continuous : $\exists \gamma > 0 \quad |a(u,v)| \leq \gamma ||u|| \, ||v|| \; \forall u,\, v \in V$.
- $\exists \alpha_1 > 0 \qquad \sup_{v \in V, v \neq 0} \dfrac{a(u,v)}{||v||} \geq \alpha_1 ||u|| \; \forall u \in V$.
- $\exists \alpha_2 > 0 \qquad \sup_{u \in V, u \neq 0} \dfrac{a(u,v)}{||u||} \geq \alpha_2 ||v|| \; \forall v \in V$.

find $u \in V$ s.t. $a(u,v) = F(v) \quad \forall v \in V$ has a unique solution.

$\forall U \subset V$ subspace

$$\text{find } u \in U \text{ s.t. } a(u,v) = F(v) \quad \forall v \in U$$

in general, inf-sup condition in subspace U is unclear.
in discretized problem : $V_h \subset V$?
in linear solver (subspace of V_h) ?

matrix form of mixed finite element method

$\vec{u} \in \mathbb{R}^{N_u}$, $\vec{p} \in \mathbb{R}^{N_p}$

$$K \begin{bmatrix} \vec{u} \\ \vec{p} \end{bmatrix} = \begin{bmatrix} A & B^T \\ B & -\epsilon I \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{p} \end{bmatrix} = \begin{bmatrix} \vec{f} \\ \vec{g} \end{bmatrix}$$

- A: coercive $(A\vec{x}, \vec{x}\,) > 0 \;\forall \vec{x} \neq \vec{0}$
- B^T : satisfies discrete inf-sup condition, i.e. ker$B^T = \{\vec{0}\}$
- $\epsilon > 0$

Schur complement matrix $S = \epsilon I + BA^{-1}B^T$

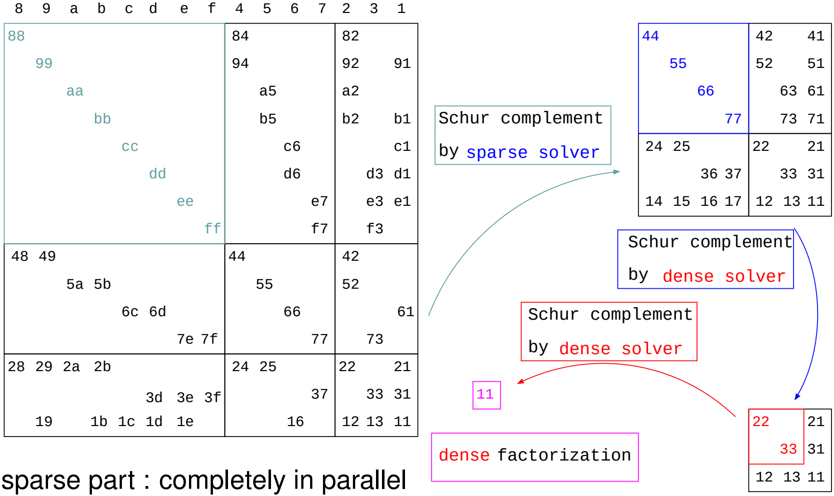
$$(S\,\vec{p}, \vec{p}\,) = \epsilon(\vec{p}, \vec{p}\,) + (BA^{-1}B^T\vec{p}, \vec{p}\,) \geq \epsilon(\vec{p}, \vec{p}\,) > 0$$

- LDL^T-factorization is possible for any ordering
- we need to set appropriate regularization $\epsilon > 0$

recursive generation of Schur complement by nested-dissection

$$\begin{bmatrix} A_{11} & A_{21} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & S_{22} \end{bmatrix} \begin{bmatrix} I_1 & A_{11}^{-1}A_{12} \\ 0 & I_2 \end{bmatrix}$$

$S_{22} = A_{22} - A_{21}A_{11}^{-1}A_{12} = A_{22} - (A_{21}U_{11}^{-1})D_{11}^{-1}L_{11}^{-1}A_{12}$: recursively computed



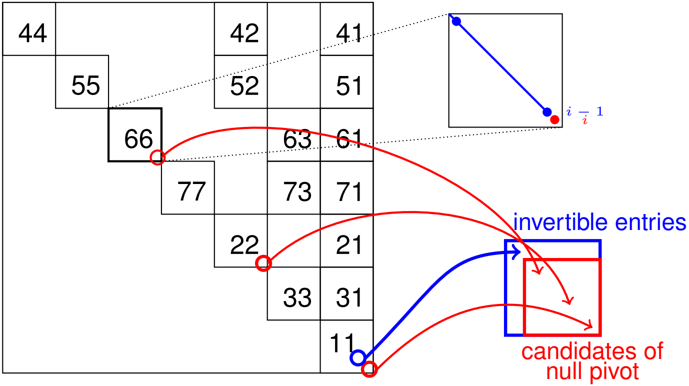
sparse part : completely in parallel

dense part : better use of BLAS 3; dgemm, dtrsm

Symmetric pivoting with postponing for block strategy : 1/2

- nested-dissection decomposition may produce singular sub-matrix for indefinite matrix

τ : given threshold for postponing, 10^{-2}
$|A(i,i)|/|A(i-1,i-1)| < \tau \;\Rightarrow \{A(k,j)\}_{i \leq k,j}$ are postponed

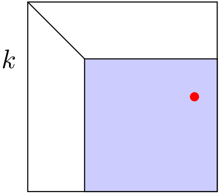


Schur complement matrix from postponed pivots is computed

Pivoting strategy

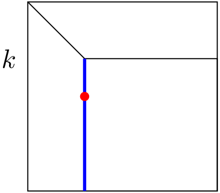
full pivoting : $A = \Pi_L^T L U \Pi_R$

find $\max_{k<i,j\leq n} |A(i,j)|$



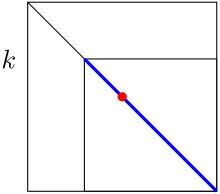
partial pivoting : $A = \Pi L U$

find $\max_{k<i\leq n} |A(i,k)|$



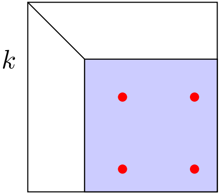
symmetric pivoting : $A = \Pi^T L D U \Pi$

find $\max_{k<i\leq n} |A(k,k)|$



2×2 pivoting : $A = \Pi^T \, L \, \tilde{D} \, U \Pi$

find $\max_{k<i,j\leq n} \det \begin{vmatrix} A(i,i) & A(i,j) \\ A(j,i) & A(j,j) \end{vmatrix}$



sym. pivoting is mathematically not always possible

Computation of a solution of singular linear system

- $A \in \mathbb{R}^{N\times N}$, $\quad k =$ dimKer$A$.
- index ordering $\{i_1, i_2, \cdots, i_N\}$ $\quad V_{N-k} =$ span$[\vec{e}_{i_1}, \vec{e}_{i_2}, \dots, \vec{e}_{i_{N-k}}]$

assumption 1 : $V_{N-k} \cap$ ker$A = \{\vec{0}\}$ $\quad \Rightarrow \quad \exists A_{11}^{-1}$
Solution in the space V_{N-k}

$$\vec{x} = A^{\dagger}\vec{f} \quad \Leftrightarrow \quad \vec{x} \in V_{N-k} \quad (A\vec{x} - \vec{f}, \vec{v}) = 0 \quad \forall \vec{v} \in V_{N-k}$$

$N_1 = \begin{bmatrix} A_{11}^{-1}A_{12} \\ -I_2 \end{bmatrix}$ $N_2 = \begin{bmatrix} A_{11}^{-T}A_{21}^{T} \\ -I_2 \end{bmatrix}$, ker$A =$ spanN_1, ker$A^T =$ sapnN_2

assumption 2 :

$\vec{u} = A^{\dagger}\vec{f}$ is computable by LDU-factorization with symmetric pivot.

$\vec{f} \in$ ImA

$\Leftrightarrow \vec{f} \perp$ Ker$A^T \Leftrightarrow N_2^T\vec{f} = \vec{0} \Leftrightarrow A_{21}A_{11}^{-1}\vec{f}_1 - \vec{f}_2 = \vec{0} \Rightarrow \; A\,A^{\dagger}\vec{f} = \vec{f}$

$\forall\vec{\zeta} \in \mathbb{R}^k, \quad A(\vec{u} + N_1\vec{\zeta}) = \vec{f}$

- $\vec{u} + N_1\vec{\zeta} \in$ (Ker$A)^{\perp}$ $\quad \Leftrightarrow \quad N_1^T(\vec{u} + N_1\vec{\zeta}) = \vec{0}$: algebraic inverse
- $\vec{u} + N_1\vec{\zeta} \in V_{N-k}$ $\quad \Leftrightarrow \quad \vec{\zeta} = \vec{0}$: computational
- $\vec{u} + N_1\vec{\zeta} \in$ ImA $\quad \Leftrightarrow \quad N_2^T(\vec{u} + N_1\vec{\zeta}) = \vec{0} \Leftarrow$ Im$A \cap$ Ker$A = \{\vec{0}\}$

Im$A \cap$ Ker$A = \{\vec{0}\}$ $\Rightarrow N_2^T N_1$ is invertible

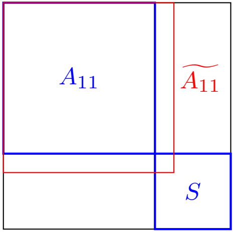
$N_1\vec{\eta} \in$ Im$A\cap$Ker$A \Leftrightarrow N_2^T N_1\vec{\eta} = \vec{0}$, Im$A\cap$Ker$A = \{\vec{0}\} \Leftrightarrow (N_2^T N_1\vec{\eta} = \vec{0} \Rightarrow \vec{\eta} = \vec{0})$

extension with 2x2 pivoting or unsymmetric pivoting

For matrix from PDE, most of part can be factorized with symmetric pivoting.

$\widetilde{A_{11}}$ has LDU-factorization with symmetric pivot

entries of S $\Leftarrow$ postponed pivots + some invertible entries of $\widetilde{A_{11}}$



$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & S \end{bmatrix} \begin{bmatrix} I_{11} & A_{11}^{-1} A_{12} \\ 0 & I_2 \end{bmatrix}$$

$$S = A_{22} - A_{21} A_{11}^{-1} A_{12}$$

- applying full pivot when S is (highly) unsymmetric, $\bar{S} = \Pi_L S \Pi_R$
- applying 2x2 pivot when S is symmetric $+$ indefinite

with full pivoting Π_L / Π_R

$$\bar{S} = \Pi_L S \Pi_R = \begin{bmatrix} \bar{S}_{22} & \bar{S}_{23} \\ \bar{S}_{32} & \bar{S}_{33} \end{bmatrix} = \begin{bmatrix} \bar{S}_{22} & 0 \\ \bar{S}_{32} & E_{33} \end{bmatrix} \begin{bmatrix} I_2 & \bar{S}_{22}^{-1} \bar{S}_{23} \\ 0 & I_3 \end{bmatrix}$$

$E_{33} = 0 \ \Rightarrow$ Ker$\bar{S}$ = span $\begin{bmatrix} \bar{S}_{22}^{-1} \bar{S}_{23} \\ -I_3 \end{bmatrix}$, Ker$S$ = span$\Pi_R \begin{bmatrix} \bar{S}_{22}^{-1} \bar{S}_{23} \\ -I_3 \end{bmatrix}$

$$\mathrm{Ker} A = \begin{bmatrix} A_{11}^{-1} A_{12}^{(2)} & A_{11}^{-1} A_{12}^{(3)} \\ -I_2^{(2)} & 0 \\ 0 & -I_2^{(3)} \end{bmatrix} \Pi_R \begin{bmatrix} \bar{S}_{22}^{-1} \bar{S}_{23} \\ -I_3 \end{bmatrix}$$

novel kernel detection algorithm based on LDU

$A : N \times N$ unsymmetric, dimKer$A = k \geq 1$, dimIm$A \geq m$.
two parameters: l, n, which define size of factorization,

$$\begin{array}{r} N-n \updownarrow \\ n \updownarrow \end{array} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & S_{22} \end{bmatrix} \begin{bmatrix} I_1 & A_{11}^{-1} A_{12} \\ 0 & I_2 \end{bmatrix} \qquad \widetilde{\mathrm{Im}}_n = \mathrm{span} \begin{bmatrix} \bar{A}_{11}^{-1} A_{12} \\ -I_2 \end{bmatrix}^{\perp}.$$

- projection : $P_n^{\perp} : \mathbb{R}^N \to \widetilde{\mathrm{Im}}_n$
- solution in subspace, $\bar{A}_{N-l}^{\dagger} \vec{b} = \begin{bmatrix} \bar{A}_{11}^{-1} \vec{b}_1 \\ \vec{0} \end{bmatrix}$, $b = \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix} \quad \begin{array}{l} \updownarrow N-l \\ \updownarrow l \end{array}$

$\bar{A}_{11}^{-1}$: computed in quadruple-precision with perturbation to simulate double-precision round-off error.

$\boxed{\text{kernel detection algorithm}}$

DOI: 10.1002/nme.4729

n : candidate of dimension of the kernel
compute for $l = n-1, n, n+1$

$$\mathrm{err}_l^{(n)} := \max \left\{ \max_{\vec{x} = [\vec{0}\ \vec{x}_l] \neq \vec{0}} \frac{||P_n^{\perp}(\bar{A}_{N-l}^{\dagger} A \vec{x} - \vec{x})||}{||\vec{x}||}, \max_{\vec{x} = [\vec{x}_{N-l}\ \vec{0}] \neq \vec{0}} \frac{||\bar{A}_{N-l}^{\dagger} A \vec{x} - \vec{x}||}{||\vec{x}||} \right\}$$

$$\begin{array}{cccccccc} n = k+1 & \Leftrightarrow & \mathrm{err}_{n-1} \approx 0 & \wedge & \mathrm{err}_n \approx 0 & \wedge & \mathrm{err}_{n+1} \sim 1 \\ n = k & \Leftrightarrow & \mathrm{err}_{n-1} \gg 0 & \wedge & \mathrm{err}_n \approx 0 & \wedge & \mathrm{err}_{n+1} \sim 1 \\ n = k-1 & \Leftrightarrow & \mathrm{err}_{n-1} \gg 0 & \wedge & \mathrm{err}_n \gg 0 & \wedge & \mathrm{err}_{n+1} \sim 1 \end{array}$$

Theoretically $\neg A_{N-k+1}^{-1}$, but $\mathrm{err}_{k-1}^{(k)}$ is computable; $\sim ||\bar{A}_N^{-1} A_N - I_N||$.

Solution of singular system in FreeFem++ : 1/2

Navier-Stokes equations with full Dirichlet boundary conditions

```
load "Dissection"
defaulttoDissection();
solve NavierStokes([u1,u2,p],[v1,v2,q],
    solver=sparsesolver,tolpivot=1.0e-2) =
  int2d(Th)( [u1,u2]'*[v1,v2]*rho/dt +
             UgradV(u1,u2,up1,up2)'*[v1,v2]*rho
     + 2*nu *(Eps(u1,u2):Eps(v1,v2))
     - p * div(v1,v2) - q *div(u1,u2)) // - eps*p*q )
     - int2d(Th)( [up1,up2]'*[v1,v2]*rho/dt - rho*v2 )
  +on(1,u1=0,u2=0)
```

- without penalization for indefinite system
- without penalization for Dirichlet boundary condition by `tgv`

Solution of singular system in FreeFem++ : 2/2

Accessing to kernel vectors in cavity driven flow problem

```
load "Dissection"
defaulttoDissection();
varf vDNS ([u1,u2,p],[v1,v2,q]) = int2d(Th)(
     + nu * ( dx(u1)*dx(v1) + dy(u1)*dy(v1)
            + dx(u2)*dx(v2) + dy(u2)*dy(v2) )
     + p*dx(v1)+ p*dy(v2) - dx(u1)*q- dy(u2)*q
     + (Ugrad(u1,u2,up1,up2)'*[v1,v2] +
     + Ugrad(up1,up2,u1,u2)'*[v1,v2])
     + on(1,2,3,4,u1=1,u2=1)
matrix Ans=vDNS(XXMh,XXMh,tgv=1e+30);
real[int] b = vNS(0,XXMh,tgv=1e+30);
real[int,int] kernn(1,1), kernt(1,1); // N x kerndim
int kerndim;
set(Ans,solver=sparsesolver,strategy=2, tolpivot=1.0e-2);
dissectionkernel(Ans,kerneldim=kerndim,
                 kerneln=kernn,kernelt=kernt);
cout << "kern dim = " << kerndim << endl;
```

Thanks to Pierre Jolivet,

- `GetSolver()` added in `MarticeCreuse.pp`, github: develop

Numerical simulation of semi-conductor problem

- governing equation
  non-linear system of Drift-Diffusion equations
- discretization scheme to satisfy conservation law
  FVM : Scharfetter-Gummel method, 1969
  mixed FEM + hybridization + mass lumping = FVM
  M-matrix property Brezzi-Marini-Pietra, 1987
- nonlinear solver
  Gummel map (a kind of fixed point method), 1964
  full Newton iteration
- linear solver to tread large condition number
  Bi-CGSTAB : van der Vorst, 1992
  Pardiso : Schenk-Gärtner-Schmidthüsen-Fichtner, 1999

- mixed formulation with $H(\text{div}) \times L^2$ directly written by FreeFem++
- indefinite unsymmetric system is solved by Dissection stably.

Drift-Diffusion system at stationary state

- φ : electrostatic potential
- n : electron concentration
- p : hole concentration

$$
\begin{aligned}
\mathrm{div}(\varepsilon E) &= q(p - n + C(x)) \qquad & E &= -\nabla\varphi \\
-\mathrm{div} J_n &= 0 & J_n &= -q(\mu_n n \nabla\varphi - D_n \nabla n) \\
\mathrm{div} J_p &= 0 & J_p &= -q(\mu_p p \nabla\varphi + D_p \nabla p)
\end{aligned}
$$

- q : positive electron charge
- ε : dielectric constant of the materials
- $D_n = \mu_n\theta$, $D_p = \mu_p\theta$: Einstein's relation

Maxwell-Boltzmann statistics : $p = N_i \mathrm{exp}(\frac{\varphi_p - \varphi}{\theta})$leads to $J_p = -q\mu_p p \nabla\varphi_p$

- φ_p : quasi-Fermi level
- N_i : intrinsic concentration of the semiconductor
- $\theta = kT/q$: thermal voltage
- k : Boltzmann constant, T : lattice temperature

Drift-Diffusion system at stationary state : 2/3

dimensionless system (by De Mari scaling or unit scaling)

$$-\mathrm{div}(\lambda^2 \nabla \varphi) = p - n + C(x)$$

$$-\mathrm{div} J_n = 0 \qquad J_n = \nabla n - n \nabla \varphi$$

$$\mathrm{div} J_p = 0 \qquad J_p = -(\nabla p + p \nabla \varphi)$$

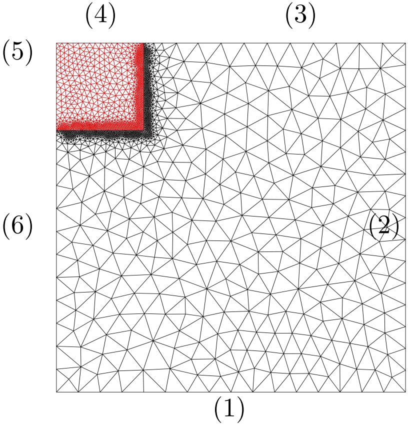
with boundary conditions

$$\varphi = f \text{ on } \Gamma_D \qquad \frac{\partial \varphi}{\partial \nu} = 0 \text{ on } \Gamma_N$$

$$n = g \text{ on } \Gamma_D \qquad J_n \cdot \nu = 0 \text{ on } \Gamma_N \Leftarrow \frac{\partial n}{\partial \nu} = 0$$

$$p = h \text{ on } \Gamma_D \qquad J_p \cdot \nu = 0 \text{ on } \Gamma_N \Leftarrow \frac{\partial p}{\partial \nu} = 0$$

Boundary conditions for a diode device (thanks to Dr. Sho)



N region : $x < 1/4 \wedge y > 3/4$,

$C(x,y) = n_d = 10^{17}$

P region : others

$C(x,y) = -n_a = -10^{20}$

$n\,p = n_i^2,\ n_i = 1.08 \times 10^{10}$

charge neutrality

$p - n + C(x,y) = 0$ on (1), (4).

N region :

$n = \sqrt{n_i^2 + C^2/4} + C/2 \simeq n_d$

P region :

$p = \sqrt{n_i^2 + C^2/4} - C/2 \simeq n_a$

(1)	$\varphi = -\log(\frac{n_a}{n_i}) + \frac{1}{V_{th}}\varphi_{\text{appl}}$	$n = \dfrac{n_i}{n_a}$	$p = \dfrac{n_a}{n_i}$
(2), (3), (5), (6)	$\partial_\nu \varphi = 0$	$\partial_\nu n = 0$	$\partial_\nu p = 0$
(4)	$\varphi = \log(\frac{n_d}{n_i})$	$n = \dfrac{n_d}{n_i}$	$p = \dfrac{n_i}{n_d}$

Finite volume discretization with Scharfetter-Gummel method

Slotboom variables, ξ : $p = \xi e^{-\varphi}$

$$\nabla p = \nabla \xi (e^{-\varphi}) - \xi e^{-\varphi} \nabla \varphi = e^{-\varphi} \nabla \xi - p \nabla \varphi$$

$$J_p = -\nabla p - p \nabla \varphi = -e^{-\varphi} \nabla \xi$$

approximation of $e^{\varphi} J_p = -\nabla \xi$ in an interval $[x_i, x_{i+1}]$, $h = x_{i+1} - x_i$

$$\int_{x_i}^{x_{i+1}} e^{\varphi} dx \, J_{p\, i+1/2} = -h \nabla \xi_{i+1/2} \simeq -(\xi_{i+1} - \xi_i)$$

- φ : assumed to be linear in the interval $[x_i, x_{i+1}]$ with φ_i and φ_{i+1}

$$\int_{x_i}^{x_{i+1}} e^{\varphi} dx = \frac{1}{\frac{d\varphi}{d\, x}} \left[e^{\varphi(x)} \right]_{x_i}^{x_{i+1}} = \frac{h}{\varphi_{i+1} - \varphi_i} (e^{\varphi_{i+1}} - e^{\varphi_i})$$

$$\begin{aligned} J_{p\, i+1/2} &\simeq -(\xi_{i+1} - \xi_i) \frac{\varphi_{i+1} - \varphi_i}{h} \frac{1}{e^{\varphi_{i+1}} - e^{\varphi_i}} \\ &= -\frac{\varphi_{i+1} - \varphi_i}{h} \frac{e^{\varphi_{i+1}} p_{i+1} - e^{\varphi_i} p_i}{e^{\varphi_{i+1}} - e^{\varphi_i}} \\ &= -\frac{1}{h} \left(B(\varphi_i - \varphi_{i+1}) p_{i+1} - B(\varphi_{i+1} - \varphi_i) p_i \right) \end{aligned}$$

- $B(x) = x/(e^x - 1)$: Bernoulli function

Mixed variational formulation : 1 / 2

Slotboom variable ξ : $p = \xi e^{-\varphi}$

$$\begin{aligned} \mathrm{div}(J_p) &= 0 \qquad \text{in } \Omega, \\ e^{\varphi} J_p &= -\nabla \xi \quad \text{in } \Omega. \end{aligned}$$

function space :

$$\begin{aligned} H(\mathrm{div}) &= \{v \in L^2(\Omega)^2 \,;\, \mathrm{div}\, v \in L^2(\Omega)\}, \\ \Sigma &= \{v \in H(\mathrm{div})\,;\, v \cdot \nu = 0 \text{ on } \Gamma_N\} \end{aligned}$$

integration by parts leads to

$$\int_\Omega e^{\varphi} J_p \cdot v = -\int_\Omega \nabla \xi \cdot v = \int_\Omega \xi \nabla \cdot v - \int_{\partial\Omega} \xi v \cdot \nu$$

$$\int_\Omega e^{\varphi} J_p \cdot v - \int_\Omega \xi \nabla \cdot v = -\int_{\Gamma_D} h e^{\varphi} v \cdot \nu - \int_{\Gamma_N} \xi v \cdot \nu \quad \forall v \in \Sigma$$

$$\int_\Omega \nabla \cdot J_p \, q = 0 \quad \forall q \in L^2(\Omega)$$

Mixed variational formulation : 2 / 2

mixed-type weak formulation

find $(J_p, \xi) \in \Sigma \times L^2(\Omega)$

$$\int_\Omega e^\varphi J_p \cdot v - \int_\Omega \xi \nabla \cdot v = - \int_{\Gamma_D} h e^\varphi v \cdot \nu \qquad \forall v \in \Sigma$$

$$- \int_\Omega \nabla \cdot J_p \, q = 0 \qquad \forall q \in L^2(\Omega)$$

symmetric indefinite

replacing $\xi = e^\varphi p$ again,

find $(J_p, p) \in \Sigma \times L^2(\Omega)$

$$\int_\Omega e^\varphi J_p \cdot v - \int_\Omega e^\varphi p \nabla \cdot v = - \int_{\Gamma_D} h e^\varphi v \cdot \nu \qquad \forall v \in \Sigma$$

$$- \int_\Omega \nabla \cdot J_p \, q = 0 \qquad \forall q \in L^2(\Omega)$$

unsymmetric indefinite

hybridization of mixed formulation + mass lumping $\Rightarrow$ FVM

FreeFem++ script

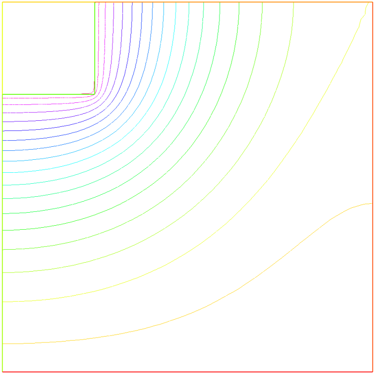
Finite element approximation

- $J_p \in H(\text{div})$: Raviar-Thomas : $RT0(K) = (P0(K))^2 + \vec{x}P0(K)$,
- $p \in L^2(\Omega)$: piecewise linear : $P1(K)$,
- $\varphi \in H^1(\Omega)$: piecewise linear : $P1(K)$.

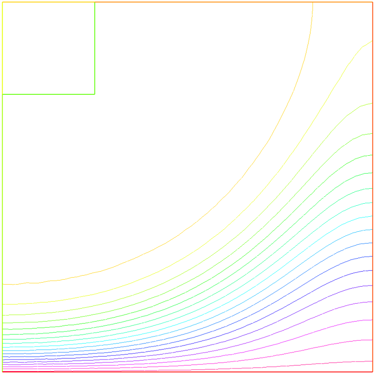
```
load "Dissection"
defaulttoDissection();
fespace Vh(Th, RT0);      fespace Ph(Th, P1);
fespace Xh(Th, P1);
Vh [up1, up2], [v1, v2];  Ph pp, q;
Xh phi; // obtained in Gummel map
problem DDp([up1, up2, pp],[v1, v2, q],
            solver=sparsesolver,strategy=2,
            tolpivot=1.0e-2,tgv=1.0e+30) =
  int2d(Th,qft=qf5pT)(exp(phi) * (up1 * v1 + up2 * v2)
                  - exp(phi) * pp * (dx(v1) + dy(v2))
                  + q * (dx(up1) + dy(up2)))
  +int1d(Th, 1)(gp1 * exp(phi) * (v1 * N.x + v2 * N.y))
  +int1d(Th, 4)(gp4 * exp(phi) * (v1 * N.x + v2 * N.y))
  + on(2, 3, 5, 6, up1 = 0, up2 = 0);
```

Numerical result of a semi-conductor problem

- compute thermal equilibrium with $p = n_i \text{exp}(-\varphi)V_{th}$ and $n = n_i \text{exp}(\varphi)V_{th}$
- Newton iteration for the potential equation and fixed point iteration for the whole system : a kind of Gummel map



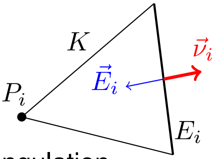
electrostatic potential



hole concentration

Raviart-Thomas finite element

$RT0(K) = (P0(K))^2 + \vec{x}P0(K) \subset (P1(K))^2$.



- K : triangle element
- $\{E_i\}$: edges of K
- $\vec{\nu}_i$: outer normal of K on E_i
- $\vec{E}_i$: normal to edge E_i given by whole triangulation

$$\vec{v} \in RT0(K) \ \Rightarrow \ \vec{v}|_{E_i} \cdot n_i \in P0(E_i), \quad \text{div}\vec{v} \in P0(K)$$

finite element basis

$$\vec{\Psi}_i(\vec{x}) = \sigma_i \frac{|E_i|}{2|K|}(\vec{x} - \vec{P}_i) \quad \sigma_i = \vec{E}_i \cdot \vec{\nu}_i, \quad P_i : \text{ node of } K$$

finite element vector value is continuous on the middle point on E_i.
inner finite element approximation of $H(\text{div}\,;\,\Omega)$.

$$\int_K e^{\varphi_h} \vec{\Psi}_j \cdot \vec{\Psi}_i \ \leftarrow \ \int_K e^{\varphi_1\lambda_1+\varphi_2\lambda_2+\varphi_3\lambda_3}\lambda_k\lambda_l \quad \text{by exact quadrature}$$

$$\int_K e^{\varphi_h} \vec{\Psi}_j \cdot \vec{\Psi}_i \ \simeq \ \frac{1}{|K|}\int_K e^{\sum_k \varphi_k\lambda_k}\int_K \vec{\Psi}_j \cdot \vec{\Psi}_i \quad \text{: exponential fitting}$$

$\{\lambda_1, \lambda_2, \lambda_3\}$: barycentric coordinates of K

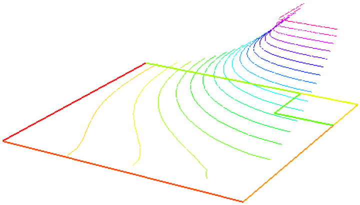
Numerical comparison on accuracy of quadrature

relative error of each entry of the stiffness matrix $m_{i\,j} = \int_K e^{\varphi_h} \vec{\Psi}_j \cdot \vec{\Psi}_i$

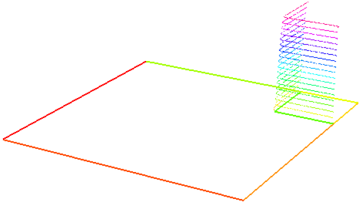
during 4 iterations of the Gummel map #nnz=31,249

	# 1		# 2	
quadrature	$\geq 1\%$	max	$\geq 1\%$	max
`qf5pT`	484	12.0995	290	1.31964
`qf7pT`	0	0.00894496	4	0.017345
`qf9pT`	0	0.0025877	0	0.00256528

	# 3		# 4	
quadrature	$\geq 1\%$	max	$\geq 1\%$	max
`qf5pT`	268	0.783141	268	1.04263
`qf7pT`	2	0.00872897	0	0.00852419
`qf9pT`	0	0.00190486	0	0.000325378



φ_h



e^{φ_h}

Nonlinear iteration to obtain the stationary state

a variant of Gummel map
φ^m, n^m, p^m : given $\to \varphi^{m+1}, n^{m+1}, p^{m+1}$ by a fixed point method
Slotboom variable ξ^m, η^m

$$p^m = \xi^m e^{-\varphi^m} \quad n^m = \eta^m e^{\varphi^m}$$

to find a solution of the nonlinear eq. : $-\mathrm{div}(\lambda^2 \nabla\varphi) = p - n + C(x)$

$$F(\eta^m, \xi^m\,;\,\varphi, \psi) = \lambda^2 \int_\Omega \nabla\varphi\cdot\nabla\psi - \int_\Omega (\xi^m e^{-\varphi} - \eta^m e^{-\varphi} + C)\psi = 0$$

differential calculus with $\delta\varphi \in \{H^1(\Omega)\,;\,\psi = 0 \text{ on } \Gamma_D\}$ leads to

$$\begin{aligned}
&F(\eta^m, \xi^m\,;\,\varphi^m + \delta\varphi, \psi) - F(\eta^m, \xi^m\,;\,\varphi^m, \psi)\\
&\quad = \lambda^2 \int_\Omega \nabla\delta\varphi\cdot\nabla\psi - \int_\Omega \xi^m\left(e^{-\varphi^m - \delta\varphi} - e^{-\varphi^m}\right)\psi - \eta^m\left(e^{\varphi^m + \delta\varphi} - e^{\varphi^m}\right)\psi\\
&\quad = \lambda^2 \int_\Omega \nabla\delta\varphi\cdot\nabla\psi + \int_\Omega \left(\xi^m e^{-\varphi^m} + \eta^m e^{\varphi^m}\right)\delta\varphi\,\psi\\
&\quad = \lambda^2 \int_\Omega \nabla\delta\varphi\cdot\nabla\psi + \int_\Omega (p^m + n^m)\delta\varphi\,\psi \qquad \varphi^{m+1} = \varphi^m + \delta\varphi
\end{aligned}$$

mixed-type weak formulation

find $(\varphi, J_n, n, J_p, p) \in H^1(\Omega) \times H(\mathrm{div};\Omega) \times L^2(\Omega) \times H(\mathrm{div};\Omega) \times L^2(\Omega)$

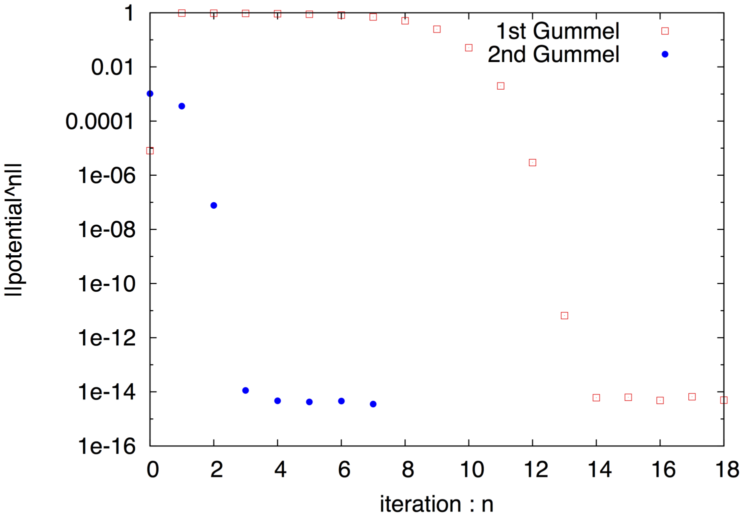
$$
\begin{aligned}
\lambda^2 \int_\Omega \nabla\varphi \cdot \nabla\psi &= \int_\Omega (p - n + C)\psi && \forall \psi \in H^1(\Omega), \\
\int_\Omega e^{-\varphi} J_n \cdot v + \int_\Omega e^{-\varphi} n \nabla \cdot v &= \int_{\Gamma_D} g_n e^{-\varphi} v \cdot \nu && \forall v \in H(\mathrm{div};\Omega), \\
\int_\Omega \nabla \cdot J_n \, q &= 0 && \forall q \in L^2(\Omega), \\
\int_\Omega e^{\varphi} J_p \cdot v - \int_\Omega e^{\varphi} p \nabla \cdot v &= -\int_{\Gamma_D} g_p e^{\varphi} v \cdot \nu && \forall v \in H(\mathrm{div};\Omega), \\
-\int_\Omega \nabla \cdot J_p \, q &= 0 && \forall q \in L^2(\Omega).
\end{aligned}
$$

Newton method + initial guess given by a solution by Gummel map
coding by `FreeFem++` and `Dissection`

- 1st order : P1/RT0/P1/RT0/P1 instead of P1/RT0/P0/RT0/P0

Newton iteration

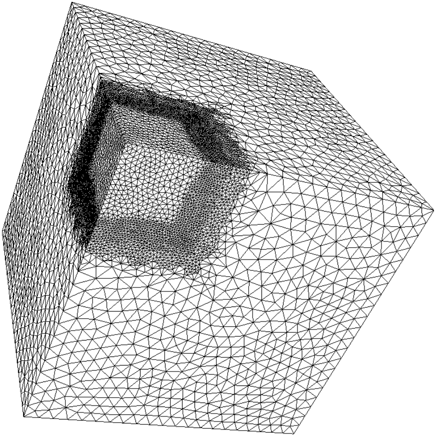
1st order scheme

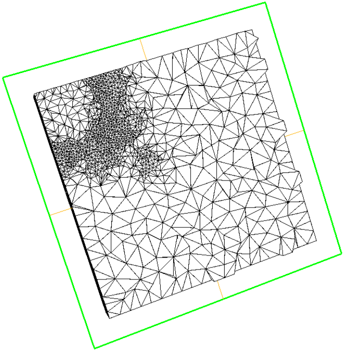


twice repeated iteration of Gummel map produces a good initial guess for Newton iteration

3D computation : 1/2

tetrahedral mesh with refinement by `tetgen`





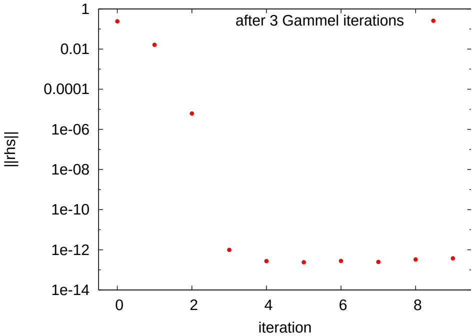
nodes = 29,107   # elements = 165,163

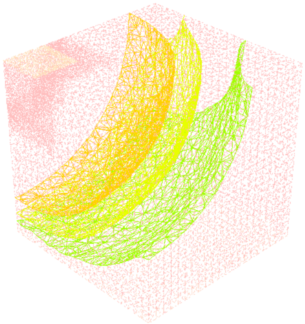
1st order : P1/RT0/P1/RT0/P1

DOF for φ = 27,942, (J_n, n) = 354,756, full Newton = 73,7454

3D computation : 2/2

- computing thermal equilibrium by Newton iteration for φ
- Gummel map : fixed point iteration for (J_n, n) and (J_p, p) + Newton iteration for φ, 3 iterations
- full Newton iteration





Dissection solver for the 4th Newton iteration

- postponed entries = 8 with $\tau = 10^{-2}$
- error = 4.71758548707057 05e-09, residual = 2.4167459916234529e-16

Summary

- Full pivoting procedure works as a preconditioner for kernel detection algorithm based on symmetric pivoting
- Kernel detection routine in Dissection is accessible from FreeFem++, github develop
- tangent matrix of Newton method with high condition number for highly nonlinear semi-conductor problem is well factorized by Dissection

ongoing

- merging in FreeFem++ ver. 4
- quadruple precision solution from given matrix data by double precision with inner iterative refinement by Dissection

source code of Dissection is accessible within FreeFem++ repository
`https://github.com/FreeFem/FreeFem-sources/tree/master/download/dissection`

under GPL linking-exception / CeCILL-C licenses