Application of Kronecker Products in Fusion Applications

Ed D'Azevedo (dazevedoef@ornl.gov)
Mark Carter (cartermd@ornl.gov)
Fred Jaeger (jaegeref@ornl.gov)
Oak Ridge National Laboratory

Overview

- **Main Idea: Kronecker Product is a neat tool.**
- Refresher on properties of Kronecker Product.
- Kronecker Product is used in variable separable interpolation on rectangular grids.
- Kronecker Product is used in fusion codes.

Kronecker Product

- Let matrices A be $mA \times nA$ and B be $mB \times nB$ be indexed as $A(ia, ja)$ and $B(ib, jb)$. Let $C = A \otimes B$ (or kron(A, B) in MATLAB notation), then matrix C is size $(mA * mB) \times (nA * nB)$.
- If matrix A is 3×3, then

$$C = \begin{bmatrix} a_{11}B & a_{12}B & a_{13}B \\ a_{21}B & a_{22}B & a_{23}B \\ a_{31}B & a_{32}B & a_{33}B \end{bmatrix} .$$

- Matrix C can be interpreted as a 4-index array $C([ib, ia], [jb, ja]) = A(ia, ja) * B(ib, jb)$, where the composite index $[ib, ia] = ib + (ia - 1) * mB$ is the index in Fortran column-wise order.

Matrix Multiply

- Efficient ($O(2n^3)$) matrix multiply without expensive construction of $C = A \otimes B$ that require $O(n^4)$ storage and work,

$$\begin{aligned} Y([ib, ia]) &= C([ib, ia], [jb, ja]) * X([jb, ja]) \\ &= A(ia, ja) * B(ib, \mathbf{jb}) * X([\mathbf{jb}, ja]) \\ &= (B(ib, jb) * X([jb, \mathbf{ja}])) * A(ia, \mathbf{ja}) \\ Y &= B * X * A^t \quad O(2n^3) \text{ work.} \end{aligned}$$

Nice Properties

- Fast matrix multiply as $(A \otimes B) * vec(X) = B * X * A^t$.
- Other properties:

$$(A \otimes B) * (E \otimes F) = (A * E) \otimes (B * F) \quad (1)$$

$$(A + B) \otimes E = A \otimes E + B \otimes E \quad (2)$$

$$(A \otimes B) \otimes E = A \otimes (B \otimes E) \quad (3)$$

$$(A \otimes B)^t = (A^t \otimes B^t) \quad (4)$$

- Fast Solver based on $(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1})$.
- Fast Solver for $A_1 \otimes B_1 + A_2 \otimes B_2$ via generalized eigen decomposition of (A_1, A_2) and (B_1, B_2) to find matrices U_A,V_A (and U_B, V_B) such that $D_1 = U_A A_1 V_A$ and $D_2 = U_A A_2 V_A$ are both diagonal matrices.

Resources about Kronecker Products

- *The Ubiquitous Kronecker Product* by C. Van Loan, Journal of Computational and Applied Mathematics, 123(2000), pp. 85-100.
- *Approximation with Kronecker Products* by C. Van Loan and N. Pitsianis in Linear Algebra for Large Scale and Real Time Applications, M. S. Moonen and G. H. Golub, eds., Kluwer Publications, 1993, pp. 293-314. (See also `http://www.cs.duke.edu/~nikos/KP/home.html`).
- *Computational Frameworks for the Fast Fourier Transform* by C. Van Loan, SIAM, 1992.

1D Interpolation

- Approximate function $f(x) \approx \sum_{j=1}^{n} c_j \phi_j(x)$.
- Basis function $\phi_j(x)$ may be Chebyshev polynomials, B-splines, wavelets or Fourier basis $e^{\alpha x \sqrt{-1}}$.
- Interpolation condition of tabulated values is used to find c_j's, $f(x_i) = \sum_{j=1}^{n} c_j \phi_j(x_i)$.
- Coefficient c_j's are obtained by solving the linear system $[f(x_i)] = T_x * [c_j]$.
- Note T_x may be sparse if $\{\phi_j(x)\}$ have compact support

$$T_x = \begin{pmatrix} \phi_1(x_1) & \cdots & \phi_n(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_n(x_n) \end{pmatrix} .$$

1D Interpolation

- Evaluation at new set of points $[f(\tilde{x}_i)]$ is computed as matrix multiply, $[f(\tilde{x}_i)] = T_{\tilde{x}} * [c_j]$,

$$T_{\tilde{x}} = \begin{pmatrix} \phi_1(\tilde{x}_1) & \cdots & \phi_n(\tilde{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\tilde{x}_m) & \cdots & \phi_n(\tilde{x}_m) \end{pmatrix} .$$

2D Interpolation

- Tabulated values $f(x_i, y_j)$ on a rectangular grid (variable separable) is approximated as $f(x, y) \approx \sum_{k,\ell} c_{k\ell} \phi_k(x) \phi_\ell(y)$.
- Interpolation conditions lead to

$$\begin{aligned} vec(F) &= (T_y \otimes T_x)\, vec(C) \\ vec(C) &= \left(T_y^{-1} \otimes T_x^{-1}\right) vec(F) \\ C &= T_x^{-1} F T_y^{-t}\,, \end{aligned}$$

  where $F = [f(x_i, y_j)]$ $C = [c_{k\ell}]$.

- Evaluation at new points $(\tilde{x}_i, \tilde{y}_j)$ is computed as

$$\left[\tilde{F}_{ij}\right] = (T_{\tilde{y}} \otimes T_{\tilde{x}})\, [c_{k\ell}] = T_{\tilde{x}} C {T_{\tilde{y}}}^t\,.$$

Higher Dimensions

- Interpolate tabulated values of 4-index function on rectangular (tensor product) grid,

$$f(w,x,y,z) \approx \sum_{ijk\ell} c_{ijk\ell}\phi_i(w)\phi_j(x)\phi_k(y)\phi_\ell(z) \ .$$

- Coefficients computed efficiently as

$$\begin{aligned} vec(F) &= (T_z \otimes T_y \otimes T_x \otimes T_w)\, vec(C) \\ vec(C) &= \left(T_z^{-1} \otimes T_y^{-1} \otimes T_x^{-1} \otimes T_w^{-1}\right) vec(F) \ . \end{aligned}$$

- Evaluation at new points is computed as

$$vec(\tilde{F}) = \left(T_{\tilde{z}} \otimes T_{\tilde{y}} \otimes T_{\tilde{x}} \otimes T_{\tilde{w}}\right) vec(C) \ .$$

Applications of Kronecker Products

- DKES transport code uses variable separable representation $f(r, \theta, \phi)$ in Tokamak geometry and lead to a linear system compactly represented as sum of 4 or more Kronecker products. Each product corresponds to a part of the differential operator. Fast matrix multiply by level 3 BLAS is available for iterative method. Question: Is there a fast solver for

$$A_1 \otimes B_1 + \cdots + A_k \otimes B_k \quad \text{where } k \geq 3 \text{ ?}$$

Applications of Kronecker Products

- 10X improvement in matrix construction in RANT3D Antenna code.
- 10X faster calculation of "WDOT" power absorption in plasma computed as $u^t * A * v$ where entries of matrix A are costly to compute. Full matrix A is interpolated from a smaller submatrix using Chebyshev polynomials and Kronecker product formulation.
- Transformation ((NU)FFT in 3D) between spectral and real representation in AORSA3D lead to 5X reduction in number of equations and about 100X speedup in dense linear solver in AORSA3D.

RANT3D

- Radio Frequency (RF) Antenna code solves Maxwell's equations in Cartesian geometry using spectral basis is used in the design of antenna for heating and current drive in tokamaks. Each cavity uses variable separable representation.

$$\nabla\left(\nabla\cdot\mathbf{E}\right)-\left(\nabla^2+\frac{\omega^2}{c^2}\right)\mathbf{E}=i\omega\mu_0\mathbf{J}\;. \tag{5}$$

- The constraints that enforce continuity of tangential electric and magnetic fields lead to repeated evaluations of the form

$$\begin{aligned} Y(m,n) &= \sum_{p,q} B(m,p)X(p,q)A(q,n) \quad \text{total } O(n^4) \text{ work} \\ Y &= (B*X)*A \quad O(2n^3) \text{ work} \\ vec(Y) &= \left(A^t\otimes B\right)*vec(X)\;. \end{aligned}$$

RANT3D

- On a model for NSTX (National Spherical Torus Experiment) with 37 recesses, the Kronecker product formulation reduced the time for impedance matrix assembly from about 682 sec to about 55 sec on a 1.3Ghz Power 4.
- We shall explore the use of Kronecker products as preconditioner.

AORSA2D

- The All-ORders Spectral Analysis code in 2D (AORSA2D) uses a spectral representation to model the response of plasma to radio frequency (RF) waves in a tokamak geometry by solving the inhomogeneous wave equation or Helmholtz equation,

$$-\nabla \times \nabla \times \mathbf{E} + \frac{\omega^2}{c^2}\left(\mathbf{E} + \frac{i}{\omega\epsilon_0}\mathbf{J}_p\right) = -i\omega\mu_0\mathbf{J}_{ant} \ .$$

- The RF electric field $\mathbf{E}$ and plasma current $\mathbf{J}_p$ are expanded in Fourier harmonics of the radial dimension as

$$\begin{aligned} \mathbf{E}(x,y) &= \sum_{n,m} \mathbf{E}_{nm} e^{i(k_n x + k_m y)} = \sum_{n,m} \mathbf{E}_{nm} e^{i\vec{k}_n \cdot r} \ , \\ \mathbf{J}_p(x) &= \sum_{n,m} \sigma(x,y,k_n,k_m) \cdot \mathbf{E}_{nm} e^{i(k_n x + k_m y)} \ . \end{aligned}$$

WDOT Power Computation

- One costly computation is the calculation of the local energy absorption at every grid point in the plasma, after $\mathbf{E}(x, y)$ is available,

$$\begin{aligned}
\dot{W} &= \frac{\partial W(\vec{k}_n, \vec{k}_m)}{\partial t} \\
&= \tfrac{1}{2}\mathrm{Re}\sum_{n,m} e^{i(\vec{k}_n - \vec{k}_m)\cdot\vec{r}} \sum_{\ell=-\infty}^{\infty} \mathbf{E}_m^* \cdot W_\ell \cdot \mathbf{E}_n \\
&= \tfrac{1}{2}\mathrm{Re}\sum_{n,m} e^{-i\vec{k}_m\cdot\vec{r}} \mathbf{E}_m^* \cdot \left(\sum_{\ell=-\infty}^{\infty} W_\ell\right) \cdot \mathbf{E}_n e^{i\vec{k}_n\cdot\vec{r}} ,
\end{aligned}$$

  where $W(x, y, \vec{k}_n, \vec{k}_m)$ involve costly function evaluations.

WDOT

- The evaluation of "WDOT" can be viewed as computing the product $u^t \tilde{F} v$ at each grid point. We interpolate $\tilde{F}$ from a small submatrix F.

$$
\begin{aligned}
u^t \tilde{F} v &= u^t (T_{\tilde{y}} \otimes T_{\tilde{x}}) C v, \quad \text{where } C = (T_y^{-1} \otimes T_x^{-1}) F \\
&= u^t T_{\tilde{x}} C T_{\tilde{y}}{}^t v = u^t T_{\tilde{x}} \boxed{T_x^{-1} F T_y^{-t}} T_{\tilde{y}}{}^t v \\
&= \left(u^t T_{\tilde{x}} T_x^{-1}\right) F \left(T_y^{-t} T_{\tilde{y}}{}^t v\right) \\
&= \tilde{u}^t F \tilde{v} \,, \quad \text{where } \tilde{u} = \left(T_x^{-t} T_{\tilde{x}}{}^t\right) u, \, \tilde{v} = \left(T_y^{-t} T_{\tilde{y}}{}^t\right) v.
\end{aligned}
$$

- The cost of precomputing $V_x = T_x^{-t} T_{\tilde{x}}{}^t$ and $V_y = T_y^{-t} T_{\tilde{y}}{}^t$ can be amortized across many grid points. Note we require only matrix-vector operations with V_x, V_y, and coefficient matrix C is not constructed.

WDOT

- On a 96×96 mode problem and 32 local modes, this reduced the time for computing WDOT from about 78 min to about 7 min by evaluating only a 9×9 submatrix of 32×32 with less than 3% difference in results.

AORSA3D

- A $34 \times 34 \times 64$ mode problem requires the solution of a large dense complex linear system with 220,000 equations. This requires about 788GBytes and 358min on 1936 power3 processors at NERSC.
- By transforming the system from Fourier representation back to physical space, we can eliminate grid points outside of plasma to solve a smaller 40,000 system.
- The row transformations are computed using ZGEMM in Kronecker products. The formulation can easily accommodate non-uniform grid spacing. Actual computation time for transformation is small compared to data redistribution.
- The reduced system requires about 26GBytes and about 100X speedup in linear solver and 27X overall speedup.

Questions

- Is there a fast solver for

$$A_1 \otimes B_1 + \cdots + A_k \otimes B_k, \quad \text{where } k \geq 3?$$

- What iterative method (GMRES?) is suitable for unsymmetric (complex) system with large imaginary eigen components?
- Nearest Kronecker Product (NKP): Given matrix C, find A and B such that $\min \|C - A \otimes B\|$ by computing SVD of $\tilde{C}$. We want $C([ib, ia], [jb, ja]) \approx A(ia, ja)B(ib, jb)$. Let $\tilde{C}$ be a rearrangement (not permutation) of C, $\tilde{C}([\mathbf{ia}, \mathbf{ja}], [ib, jb]) = C([ib, \mathbf{ia}], [jb, \mathbf{ja}])$
- Generalized NKP: Given a high order "tensor" or tabulated function $C(x_1, \ldots, x_k)$ find a good approximation by product of simpler low order tensors (similar to higher order SVD).