Application of Kronecker products in Fusion Applications *

E.F. D'Azevedo[†]***, M.D. Carter***[‡]***, E.F. Jaeger***[§]

1 Introduction

We describe the application of Kronecker product formulation in speeding up key calculations in fusion codes used in the modeling of wave-plasma interaction within the Department of Energy SciDAC (Scientific Discovery through Advanced Computing)[1] program. By taking advantage of the compact representation and efficient matrix-matrix calculations, the Kronecker product formulation leads to an order of magnitude speedup in the matrix assembly in RANT3D (Three Dimensional Recesses Antenna Model) code [1]. Interpolation computed as Kronecker products leads to significant speedup in the 'WDOT' power calculation in AORSA2D (All-Orders Spectral Algorithm in Two Dimensions) [4, 5].

2 Kronecker Product

Kronecker product (also known as outer product or tensor product) has been successfully used as a framework for understanding different variants of the Fast Fourier Transform [7]. Van Loan [8, 9] has described various interesting properties of Kronecker products and their applications. We shall only briefly review the properties of Kronecker product of matrices.

*The submitted manuscript has been authored by a contractor of the U.S. Government under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

[†]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, e6d@ornl.gov.

[‡]Fusion Energy Division, Oak Ridge National Laboratory, cartermd@ornl.gov.

[§]Fusion Energy Division, Oak Ridge National Laboratory, jaegeref@ornl.gov.

[1]For details see www.osti.gov/scidac.

Let matrice A be $mA \times nA$ and B be $mB \times nB$. For convenience, let them be indexed as $A(ia, ja)$ and $B(ib, jb)$. Let $C = A \otimes B$ (or $\mathrm{kron}(A, B)$ 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-vector multiply can be written as very efficient matrix-matrix operations[2],

$$\begin{aligned} Y([ib, ia]) &= C([ib, ia], [jb, ja]) * X([jb, ja]) \\ &= A(ia, ja) * B(ib, jb) * X([jb, ja]) \\ &= B(ib, jb) * X(jb, ja) * A(ia, ja) \\ Y &= B * X * A^t \end{aligned}$$

Other interesting properties of Kronecker products are summarized below,

$$(A \otimes B) * (E \otimes F) = (A * E) \otimes (B * F) \tag{1}$$
$$(A + B) \otimes E = A \otimes E + B \otimes E \tag{2}$$
$$(A \otimes B) \otimes E = A \otimes (B \otimes E) \tag{3}$$
$$(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1}) \tag{4}$$
$$(A \otimes B)^t = (A^t \otimes B^t) \tag{5}$$

3 Interpolation

Kronecker products also arise from interpolation of tabulated function values. Let a matrix $F = (F_{ij})$ represent tabulated function values for $F_{ij} = F(x_i, y_j)$. The function $F(x, y)$ can be approximated as $F(x, y) = \sum_{k,\ell} C_{k\ell}\phi_k(x)\phi_\ell(y)$, where the basis functions $\phi_k(x)$ may be chosen for example to be B-splines or $(k-1)^{th}$ degree Chebyshev polynomials. The coefficients $C_{k\ell}$ can be computed to satisfy the interpolation conditions

$$F_{ij} = \sum_{k,\ell} C_{k\ell}\phi_k(x_i)\phi_\ell(y_j) \tag{6}$$

The interpolation conditions can be expressed as a Kronecker product, $F = (T_y \otimes T_x) * C$, where

$$T_x = \begin{pmatrix} \phi_1(x_1) & \cdots & \phi_n(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_n(x_n) \end{pmatrix}, \quad T_y = \begin{pmatrix} \phi_1(y_1) & \cdots & \phi_n(y_1) \\ \vdots & \ddots & \vdots \\ \phi_1(y_n) & \cdots & \phi_n(y_n) \end{pmatrix}. \tag{7}$$

[2]We blur the distinction between the matrix $X(jb, ja)$ and vector $X([jb, ja])$ with composite index.

The columns of T_x and T_y contain the values of the basis functions evaluated as the interpolation knots. The coefficients $C_{k\ell}$ can be efficiently computed using the property of Kronecker products as

$$C = (T_y \otimes T_x)^{-1} * F = \left(T_y^{-1} \otimes T_x^{-1}\right) * F = T_x^{-1} * F * T_y^{-t} \tag{8}$$

Once the coefficients are known, extrapolation at the set of new values $F(\tilde{x}_m, \tilde{y}_n) = \left(\tilde{F}_{mn}\right)$ can be computed as Kronecker products,

$$\tilde{F} = T_{\tilde{x}} * C * T_{\tilde{y}}{}^t = T_{\tilde{x}} * \left(T_x^{-1} * F * T_y^{-t}\right) * T_{\tilde{y}}{}^t \ , \quad \text{from (8)}$$

$$\tilde{F} = \left(T_{\tilde{x}} * T_x^{-1}\right) * F * \left(T_{\tilde{y}} * T_y^{-1}\right)^t \tag{9}$$

where

$$T_{\tilde{y}} = \begin{pmatrix} \phi_1(\tilde{y}_1) & \cdots & \phi_n(\tilde{y}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\tilde{y}_n) & \cdots & \phi_n(\tilde{y}_n) \end{pmatrix} . \tag{10}$$

The above can be generalized to higher dimentions. For a 4-variable function indexed as $F(w, x, y, z)$, the corresponding interpolation scheme would be

$$F(w, x, y, z) = \sum_{i,j,k,\ell} C_{ijk\ell} \phi_i(w) \phi_j(x) \phi_k(y) \phi_\ell(z). \tag{11}$$

The interpolation conditions can be written as

$$F = ((T_z \otimes T_y) \otimes (T_x \otimes T_w)) * C \tag{12}$$

so that coefficients $C_{ijk\ell}$ can be efficiently computed as

$$C = (T_x \otimes T_w)^{-1} * F * (T_z \otimes T_y)^{-t} \tag{13}$$

$$= \left(T_x^{-1} \otimes T_w^{-1}\right) * F * \left(T_z^{-t} \otimes T_y^{-t}\right) \ . \tag{14}$$

4 RANT3D

The RANT3D [1, 2] antenna modeling code solves Maxwell's equations in two and three dimensions. The antenna geometry is specified in Cartesian coordinates through a series of rectangular recesses and current straps. The code is used in the design of radio-frequency (rf) antenna for heating and current drive in tokamaks. The Kronecker product algebra implemented as matrix-matrix multply was used for calculating strap impedance matrices. This reduced the time for multiple strap antenna arrays by an order of magnitude.

RANT3D solves Maxwell's equation in vacuum with a generalized plama boundary

$$\nabla (\nabla \cdot \mathbf{E}) - \left(\nabla^2 + \frac{\omega^2}{c^2}\right) \mathbf{E} = i\omega\mu_0 \mathbf{J} \ . \tag{15}$$

Each component of $\mathbf{E}$ is represented in a variable separable manner by Fourier basis

$$E_c(x,y,z) = \sum_{m,n} E_c^{mn}(x)\eta_c^m(y)\eta_c^n(z)\ , \quad J_c(x,y,z) = \sum_{m,n} J_c^{mn}\eta_c^m(y)\eta_c^n(z) \tag{16}$$

where subscript $c = x, y,$ or z, and the basis functions η_c^m are of the form sin, cos or exp.

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

$$Y(m,n) = \sum_{i,j} B(m,i)X(i,j)A(n,j)\ , \quad \text{for each } m, n. \tag{17}$$

The above computation can be seen to be $Y = (A \otimes B) * X$ can this form can be evaluated in $O(2N^3)$ instead of $O(N^4)$ work. 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.

5 Power Calculations

The AORSA2D code [3, 4, 5] 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,

$$-\nabla \times \nabla \times \mathbf{E} + \frac{\omega^2}{c^2}\left(\mathbf{E} + \frac{i}{\omega\epsilon_0}\mathbf{J}_p\right) = -i\omega\mu_0\mathbf{J}_{ant}\ , \tag{18}$$

where $\mathbf{E}$ is the wave electric field and $\mathbf{J}_{ant}$ is a specified external antenna current. Most of the complication in (18) arises from the response of the plasma to the electromagnetic wave field, which is included through the plasma current, $\mathbf{J}_p$. The rf electric field $\mathbf{E}$ and plasma current $\mathbf{J}_p$ are expanded in Fourier harmonics of the radial dimension as

$$\mathbf{E}(x,y) = \sum_{n,m} \mathbf{E}_{nm} e^{i(k_n x + k_m y)} = \sum_{n,m} \mathbf{E}_{nm} e^{i\vec{k}_n \cdot r}\ , \tag{19}$$

$$\mathbf{J}_p(x) = \sum_{n,m} \sigma(x,y,k_n,k_m)\cdot \mathbf{E}_{nm} e^{i(k_n x + k_m y)}\ . \tag{20}$$

The tensor $\sigma(x,y,k_n,k_m)$ can be derived from the first-order rf distribution function given by Stix [6]. It depends on the Fourier mode k_n,k_m and is a complicated function of modified Bessel functions and plasma Z functions. AORSA2D uses the method of collocation and constructs a large dense complex linear system that is solved in parallel by ScaLAPACK. For example, with 200×200 Fourier modes, it is necessary to solve $120,000$ coupled complex equations, and the storage required for the resulting matrix is about 230 GBytes and require about 1.3 Tflops-hour of 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,

$$\dot{W} = \frac{\partial W(\vec{k}_n, \vec{k}_m)}{\partial t} \tag{21}$$

$$= \tfrac{1}{2}\mathrm{Re}\sum_{n,m} e^{i(\vec{k}_n - \vec{k}_m)\cdot\vec{r}} \sum_{\ell=-\infty}^{\infty} \mathbf{E}_m^* \cdot W_\ell \cdot \mathbf{E}_n \tag{22}$$

$$= \tfrac{1}{2}\mathrm{Re}\sum_{n,m} e^{-i\vec{k}_m\cdot\vec{r}}\mathbf{E}_m^* \cdot \left(\sum_{\ell=-\infty}^{\infty} W_\ell\right) \cdot \mathbf{E}_n e^{i\vec{k}_n\cdot\vec{r}} \tag{23}$$

where $W(x,y,\vec{k}_n,\vec{k}_m)$ involve costly evaluaton of modified Bessel functions,

$$W_\ell = C * \begin{pmatrix} \frac{\ell^2 I_\ell}{\tilde{\Gamma}} Z_\ell & -i\ell\left(\frac{\Gamma_n}{\tilde{\Gamma}} I_\ell - I'_\ell\right) Z_\ell & -k_{\perp,n}\frac{\ell I_\ell}{\tilde{\Gamma}}\frac{\alpha Z'_\ell}{2\Omega} \\ i\ell\left(\frac{\Gamma_m}{\tilde{\Gamma}} I_\ell - I'_\ell\right) Z_\ell & \left[\frac{\ell^2}{\tilde{\Gamma}} I_\ell + 2\tilde{\Gamma} I_\ell - 2\bar{\Gamma} I'_\ell\right] Z_\ell & -i(k_{\perp,m} I_\ell - k_{\perp,n} I'_\ell)\frac{\alpha Z'_\ell}{2\Omega} \\ -k_{\perp,m}\frac{\ell I_\ell}{\tilde{\Gamma}}\frac{\alpha Z'_\ell}{2\Omega} & i(k_{\perp,n} I_\ell - k_{\perp,m} I'_\ell)\frac{\alpha Z'_\ell}{2\Omega} & -\zeta_\ell I_\ell Z'_\ell \end{pmatrix},$$

$$I_\ell = I_\ell(\tilde{\Gamma}) \ , \quad I'_\ell = I'_\ell(\tilde{\Gamma}) \ , \quad C = -i\epsilon_0 \frac{\omega_p^2}{k_\perp \alpha} e^{-\bar{\Gamma}} \tag{24}$$

$$\tilde{\Gamma} = \sqrt{\Gamma_n \Gamma_m} \ , \quad \bar{\Gamma} = \tfrac{1}{2}(\Gamma_n + \Gamma_m) \ , \quad \Gamma_n = \tfrac{1}{2}(k_{\perp,n}\alpha/\Omega)^2 \ . \tag{25}$$

Here I_ℓ is the modified Bessel function of order ℓ, Z_ℓ is the plasma dispersion function with arguments $\zeta_\ell = (\omega - \ell\Omega)/|k_\perp|\alpha$, and derivative of Z_ℓ is $Z'_\ell(\zeta_\ell) = -2[1 + \zeta_\ell Z(\zeta_\ell)]$. There is an additional 'Swanson's rotation' to transform $\mathbf{E}(x,y)$ and other quantities between the 'Stix frame' and local coordinate aligned to the magnetic field.

Two simplifying assumptions are used to reduce the cost for computing $\dot{W}$. The first is to truncate the expansion of W_ℓ typically to $-2 \le \ell \le 2$. The second is to restrict the evaluation to a smaller number of local Fourier modes (say 32 modes instead of 96 modes) and with some overlap with neigboring nodes. Even with these simplications for the evaluation of $\dot{W}$, the relative error in evaluation is typically less than 5%.

This costly evaluation can be further reduced by using interpolation and extrapolation expressed as Kronecker products in (9). We shall consider, for example, the evaluation of the $(1,1)$ component in $\dot{W}$. The evaluation of other entries can be similarly derived. The main contribution is

$$\sum_{m,n} u_m \left(\sum_\ell \ell^2 I_\ell Z_\ell\right) v_n \ , \tag{26}$$

where $u_m = e^{-i\vec{k}_m\cdot\vec{r}}\mathbf{E}_m/\sqrt{\Gamma_m}$, $v_n = e^{i\vec{k}_n\cdot\vec{r}}\mathbf{E}_n/\sqrt{\Gamma_n}$ (with $\tilde{\Gamma} = \sqrt{\Gamma_n\Gamma_m}$). This can be interpreted as the algebraic evaluation of a vector product, $u^t * \tilde{F} * v$ where entries in matrix $\tilde{F}$, $\tilde{F}_{n,m} = \sum_\ell \ell^2 I_\ell(\sqrt{\Gamma_n\Gamma_m}) Z_\ell$ are costly to compute.

As entries in the matrix $\tilde{F}$ (size $M \times M$) are smooth functions of modified Bessel functions and plasma Z dispersion functions, we can consider approximating matrix $\tilde{F}$ by evaluating only a $m \times m$ submatrix F and performing interpolation. Using equations (8) and (9), we have

$$u^t \tilde{F} v = u^t * \left(T_{\tilde{x}} * T_x^{-1}\right) * F * \left(T_{\tilde{y}} * T_y^{-1}\right)^t * v \tag{27}$$

$$= \left(u^t * T_{\tilde{x}} * T_x^{-1}\right) * F * \left(T_y^{-t} T_{\tilde{y}}{}^t v\right) \tag{28}$$

$$= \tilde{u}^t * F * \tilde{v}\ , \quad \text{where } \tilde{u} = T_x^{-t} * T_{\tilde{x}}{}^t * u,\ \tilde{v} = T_y^{-t} * T_{\tilde{y}}{}^t * v. \tag{29}$$

Since the evaluation of $\dot{W}$ is repeated many times at each grid point in the plasma, we can amortize the cost for precomputing the transformation matrices

$$\hat{T}_y = T_y^{-t} * T_{\tilde{y}}{}^t\ , \quad \hat{T}_x = T_x^{-t} * T_{\tilde{x}}{}^t\ . \tag{30}$$

The overall approximate computation of $u^t * \tilde{F} * v$ can be arranged as (i) evaluation of a submatrix F, (ii) transform vectors $\tilde{u} = \hat{T}_y * u$, $\tilde{v} = \hat{T}_x * v$ (in $O(2mM)$ work) and (iii) final evaluation of $\hat{u}^t * F * \hat{v}$ (in $O(2m^2)$ work). Note that explicit computation of the interpolation coefficients by (8) (in $O(2m^3)$ work) and evaluation of $\tilde{F}$ by (9) (in $O(mM^2)$ work) to evaluate $u^t * \tilde{F} * v$ (in $O(2M^2)$ work) would be more costly.

This approximate evaluation of $\dot{W}$ has been incorporated into the AORSA2D code. For one typical computation with 96×96 modes, the time for evaluating $\dot{W}$ using 32 local Fourier modes required about 78.3 min while the Kronecker version that evaluated a 9×9 submatrix (instead of 32×32 matrix for each component) reduced the time to about 7.1 minutes with less than 3% difference in the results.

There is an on-going effort to incorporate similar Kronecker technology for evaluate of $\dot{W}$ in the three-dimensional version of AORSA.

Bibliography

[1] M. D. Carter, D. A. Rasmussen, P. M. Ryan, G. R. Hanson, D. C. Stallings, D. B. Batchelor, T. S. Bigelow, A. C. England, D. J. Hoffman, M. Murakami, C. Y. Wang, J. B. Wilgen, J. H. Rogers, J. R. Wilson, R. Majeski, and G. Schilling, *Three dimensional modelling of ICRF launchers for fusion devices*, Nuclear Fusion, 36 (1996), pp. 209–223.

[2] A. Griffith, M. Carter, P. Ryan, K. White, and D. Swain, *Introduction to RANT3D - The Three-dimensional Recessed Antenna Model*, Tech. Rep. ORNL/FED, Fusion Energy Division, Oak Ridge National Laboratory, 2002. (see also http://www.ornl.gov/fed/Theory/rantv4_webpage/).

[3] E. F. Jaeger, L. A. Berry, and D. B. Batchelor, *Full-wave calculation of sheared poloidal flow driven by high-harmonic ion Bernstein waves in tokamak plasmas*, Physics of Plasmas, 7 (2000), pp. 3319–3329.

[4] E. F. Jaeger, L. A. Berry, E. D'Azevedo, D. B. Batchelor, and M. D. Carter, *All-orders spectral calculation of radio-frequency heating in two-dimensional toroidal plasmas*, Physics of Plasmas, 8 (2001), pp. 1573–1583.

[5] E. F. Jaeger, L. A. Berry, E. D'Azevedo, D. B. Batchelor, M. D. Carter, K. F. White, and H. Weitzner, *Advances in full-wave modeling of radio frequency heated, multidimensional plasmas*, Physics of Plasmas, 9 (2002), pp. 1873–19881.

[6] T. H. Stix, *Waves in Plasma*, American Institute of Physics, New York, 1992.

[7] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.

[8] ———, *The ubiquitous kronecker product*, Journal of Computational and Applied Mathematics, 123 (2000), pp. 85–100.

[9] C. Van Loan and N. Pitsianis, *Approximation with kronecker products*, 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).