A Sinc-Galerkin method for the Poisson Problem, Reformulated

Brian Bockelman
April 17, 2006

1 Tensor math

Definition 1.1 A block matrix $A$ can be represented as:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1\ell} \\ A_{21} & A_{22} & \cdots & A_{2\ell} \\ \vdots & \vdots & \ddots & \vdots \\ A_{k1} & A_{k2} & \cdots & A_{k\ell} \end{bmatrix}$$

where each submatrix $A_{ij}$ is of dimensions $m_i \times n_j$.

We may add, subtract, scalar multiply, and matrix multiply block matrices via submatrix operations. The adjoint of a block matrix $A$ is given by

$$A^* = \begin{bmatrix} A^*_{11} & A^*_{12} & \cdots & A^*_{1\ell} \\ A^*_{21} & A^*_{22} & \cdots & A^*_{2\ell} \\ \vdots & \vdots & \ddots & \vdots \\ A^*_{k1} & A^*_{k2} & \cdots & A^*_{k\ell} \end{bmatrix}$$

Definition 1.2 Let $A$ be an $m \times n$ matrix and $B$ be a $p \times q$ matrix. The Kronecker or tensor product of $A$ and $B$ is the $mp \times nq$ matrix

$$A \otimes B := \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}$$

The following theorem is a summary of various properties of the Kronecker product:

Theorem 1.3 Let $A, B, C$, and $D$ be the correct dimensions. Then,

1. $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha (A \otimes B)$ for $\alpha$ a scalar.
2. \((A + B) \otimes C = A \otimes C + B \otimes C\)

3. \(A \otimes (B \otimes C) = (A \otimes B) \otimes C\)

4. \((A \otimes B)(C \otimes D) = AC \otimes BD\)

5. \((A \otimes B)^* = A^* \otimes B^*\)

6. \(\text{rank}(A \otimes B) = \text{rank}(A)[\text{rank}(B)]\)

7. \((A \otimes B)^{-1} = A^{-1} \otimes B^{-1}\)

8. There exists a permutation matrix \(P\) such that

\[B \otimes A = P^T (A \otimes B) P.\]

The following two theorems relate the eigenvalues / eigenvectors of \(A, B,\) and \(A \otimes B:\)

**Theorem 1.4** Let the eigenpairs of \(A\) be \(\{\alpha_i, \vec{x}_i\}\) and those of \(B\) be \(\{\beta_j, \vec{y}_j\}\). Then the eigenpairs of \(A \otimes B\) are \(\{\alpha_i \beta_j, \vec{x}_i \otimes \vec{y}_j\}\)

**Theorem 1.5** If \(A\) and \(B\) are both

1. normal,
2. Hermitian,
3. positive definite,
4. unitary, then so is \(A \otimes B\).

We additionally want one other important matrices for our section:

**Definition 1.6** Let \(A\) be \(m \times m\) and \(B\) be \(n \times n\). The Kronecker sum of \(A\) and \(B\) is the \(mn \times mn\) matrix

\[A = I_m \otimes B + A \otimes I_n\]

In other words, the Kronecker sum of \(A\) and \(B\) is \(A = [C_{ij}]\) where

\[C_{ij} = a_{ij} I_n + \delta_{ij}^{(0)} B.\]

where \(\delta_{ij}^{(0)}\) is the Kronecker delta.

**Theorem 1.7** Let the eigenpairs of \(A\) be \(\{\alpha_i, \vec{x}_i\}\) and those of \(B\) be \(\{\beta_j, \vec{y}_j\}\). Then the eigenpairs of the Kronecker sum \(I_m \otimes B + A \otimes I_n\) are \(\{\alpha_i + \beta_j, \vec{x}_i \otimes \vec{y}_j\}\).

**Theorem 1.8** If \(A\) and \(B\) are both

1. normal,
2. Hermitian,
3. positive definite then so is the Kronecker sum of $A$ and $B$.

Finally, we define two more operations which allow us to work with multi-dimensional arrays:

**Definition 1.9** If $b$ is a one-dimensional array $b = (b_i), 1 \leq i \leq m$, then the **concatenation** of $b$ is the vector

$$
\text{co}(b) := \begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_m
\end{bmatrix}.
$$

Similarly, if $B = (b_{ij}), 1 \leq i \leq m, 1 \leq j \leq n$, is a two-dimensional array, then the **concatenation** of $B$ is the $mn \times 1$ vector

$$
\text{co}(B) = 
\begin{bmatrix}
    \text{co}(b_{i1}) \\
    \text{co}(b_{i2}) \\
    \vdots \\
    \text{co}(b_{in})
\end{bmatrix}
$$

In general, if $B = (b_{i_1i_2\ldots i_n}), 1 \leq i_j \leq m_j, 1 \leq j \leq n$, is an $n$-dimensional array, then the **concatenation** of $B$ is recursively defined by

$$
\text{co}(B) = \text{co}((b_{i_1i_2\ldots i_n})) := 
\begin{bmatrix}
    \text{co}((b_{i_1i_2\ldots i_{n-1}1})) \\
    \text{co}((b_{i_1i_2\ldots i_{n-1}2})) \\
    \vdots \\
    \text{co}((b_{i_1i_2\ldots i_{n-1}m_n})) 
\end{bmatrix}
$$

Thus, $\text{co}(B)$ is a vector of dimension $\prod_{j=1}^{n} m_j \times 1$.

On the other hand, the **matrix** operation, “mat,” allows us to represent an $n$-dimensional array as a matrix.

**Definition 1.10** If $B = (b_{ij}), 1 \leq i \leq m, 1 \leq j \leq n$, is a two-dimensional array, then the **matrix** of $B$ is the $m \times n$ matrix

$$
\text{mat}(B) = [\text{co}(b_{i1}), \text{co}(b_{i2}), \ldots, \text{co}(b_{in})].
$$

Similarly, if $B = (b_{ij\ell}), 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq \ell \leq p$, is a three-dimensional array, then the matrix of $B$ is the $mn \times p$ matrix

$$
\text{mat}(B) = \text{mat}((b_{ij\ell})) := [\text{co}(b_{ij1}), \text{co}(b_{ij2}), \ldots, \text{co}(b_{ijp})].
$$

In general, if $B = (b_{i_1i_2\ldots i_n}), 1 \leq i_j \leq m_j, 1 \leq j \leq n$, is an $n$-dimensional array ($n \geq 2$), then the **matrix** of $B$ is recursively defined by

$$
\text{mat}(B) = \text{mat}((b_{i_1i_2\ldots i_n})) := [\text{co}(b_{i_1i_2\ldots i_{n-1}1}), \text{co}(b_{i_1i_2\ldots i_{n-1}2}), \ldots, \text{co}(b_{i_1i_2\ldots i_{n-1}m_n})].
$$

Thus the matrix operation unravels the first $n-1$ indices of the array leaving a matrix with $\prod_{j=1}^{n-1} m_j$ rows and $m_n$ columns.
Two useful properties of concatenation are given:

**Theorem 1.11** If $A$ and $B$ are arrays of identical dimension and $\alpha$ and $\beta$ are scalars, then

$$\text{co}(\alpha A + \beta B) = \alpha \text{co}(A) + \beta \text{co}(B)$$

**Theorem 1.12** Let $A$ be $m \times m$, $X$ be $m \times n$, and $B$ be $n \times n$. Then,

$$\text{co}(AXB) = (B^T \otimes A)\text{co}(X)$$

This leads to the following more general statement:

**Theorem 1.13** The linear system for the unknown matrix $X$ is given as

$$A_1XB_1 + A_2XB_2 + \cdots + A_kXB_k = C$$

where $A_i$ are $m \times m$; $X$, $C$ are $m \times n$; and $B_i$ are $n \times n$. It is equivalent to

$$G\text{co}(X) = \text{co}(C)$$

where

$$G := B_1^T \otimes A + B_2^T \otimes A_2 + \cdots + B_k^T \otimes A_k.$$

### 2 An example.

From the last theorem, we gain a powerful method for solving tough systems. Here, we will apply this method to the case with $k = 2$:

$$A_1XB_1 + A_2XB_2 = C$$

We can express the solution of this equation as

$$G\text{co}(X) = \text{co}(C)$$

$$G := B_1^T \otimes A_1 + B_2 \otimes A_2.$$

While being a system which we know how to solve, this transformed system is much larger. However, we may express the diagonalization in terms of the diagonalization of $A_i$ and $B_i$.

Assume $A_1$ and $B_2$ are symmetric with eigenvalues $\{\alpha_i\}_{i=1}^n$ and $\{\beta_j\}_{j=1}^n$, respectively, and that $A_2$ and $B_1$ are positive definite. Then, $G$ becomes:

$$G := B_1 \otimes A_1 + B_2 \otimes A_2(*)$$

We know that there exists $S_1$ and $S_2$ that simultaneously diagonalize the pairs $(A_1, A_2)$ and $(B_2, B_1)$, respectively. So,

$$S_1^T A_1 S_1 = D(\alpha), \quad S_1^T A_2 S_1 = I$$
and
\[ S_2^T B_2 S_2 = D(\beta), \quad S_2^T B_1 S_2 = I. \]

Then, we can transform our problem:
\[
S_1^T A_1 S_1^{-1} X (S_2^T)^{-1} S_2^T B_1 S_2 + S_1^T A_2 S_1^{-1} X (S_2^T)^{-1} S_2^T B_2 S_2 = S_1^T F S_2.
\]

Taking
\[
Y := S_1^{-1} X (S_2^T)^{-1},
\]
\[
F := S_1^T C S_2,
\]
we can greatly simplify our system:
\[
D(\alpha)Y + Y D(\beta) = F
\]

We can then equivalently write our tensor-based system as:
\[
[I \otimes D(\alpha) + D(\beta) \otimes I] \text{co}(Y) = \text{co}(F)
\]

While this system works out cleanly, we can’t always simultaneously diagonalize the pairs \((A_1, A_2)\) and \((B_2, B_1)\). In this case, we use a Schur factorization:

**Theorem 2.1 Generalized Schur Theorem** Let \( A \) and \( B \) be \( n \times n \) matrices. Then, there exist unitary matrices \( Q \) and \( R \) such that
\[
Q^* A R = T_1, \quad Q^* B R = T_2
\]

where \( T_1 \) and \( T_2 \) are upper triangular.

Suppose there are unitary matrices \( Q_1, R_1, Q_2, \) and \( R_2 \) so that
\[
Q_1^* A_1 R_1 = T_1, \quad Q_1^* A_2 R_1 = T_2
\]

and
\[
Q_2^* B_1^* R_2 = T_3, \quad Q_2^* B_2^* R_2 = T_4.
\]

Then, our problem reduces - through similar means as before - to
\[
T_1 Y T_3^* + T_2 Y T_4^* = F
\]

where
\[
Y = R_1^* X Q_2
\]
\[
F = Q_1 C R_2
\]
2.1 Sylvester’s equation

We will often run into Sylvester’s equation in our work:

\[ AX + XB = C. \]

In terms of Kronecker products, we have:

\[ (I_n \otimes A + B^T \otimes I_m)\text{co}(X) = \text{co}(C) \]

If \( A \) and \( B \) are diagonalizable, then this can be rewritten as

\[ D(\alpha)Y + YD(\beta) = F \]

where

\[ Y = P^{-1}XQ \]
\[ F = P^{-1}CQ. \]

We can easily solve for \( Y \):

\[ y_{ij} := \frac{f_{ij}}{\alpha_i + \beta_j} \]

and then recover \( X \) with \( X = PYQ^{-1} \).

If \( A \) and \( B \) are not diagonalizable, then we can again turn to Schur factorization. If we pick unitary \( Q \) and \( R \) such that

\[ Q^* AQ = S \]
\[ R^* BR = T \]

where \( S \) and \( T \) are upper triangular, then we can rewrite Sylvester’s equation as:

\[ SY + YT^* = F \]

where

\[ Y = Q^* XR \]
\[ F = Q^* CR \]

Note \( T^* \) is lower triangular. Let \( A_j \) be the column \( j \) of matrix \( A \). The, we can write out our system as:

\[ S\tilde{y}_k + \sum_{j=k}^m t^*_{jk} y_j = \tilde{f}_k, \]

for \( k = m, m - 1, \ldots, 1 \). Remember that \( t^*_{jk} = t_{kj} \), so

\[(S + t_{kk}I)\tilde{y}_k = \tilde{f}_k - \sum_{j=k+1}^m t_{kj} \tilde{y}_k, \]

for \( k = m, m - 1, \ldots, 1 \). Note that the last column, \( m \), does not have a term from \( Y \) in the right hand side. This means that we can efficiently solve this system with back-substitution. We can recover \( X \) from \( Y \) with:

\[ X = QYR^* \]
3 Formulation of 2d Poisson Problem

From the formula (5.84) with weight function choices in (5.88), we have derived the solution to the Poisson problem using sinc methods. Further, we have been able to derive it as an example of Sylvester’s equation:

\[ D(\phi'_x)A((\phi'_x)^{-1/2})D(\phi'_x)V(2) + V(2)[D(\phi'_y)A((\phi'_y)^{-1/2})D(\phi'_y)]^T = G(2) \]

This can be more compactly written by defining

\[ A_x := D(\phi'_x)A((\phi'_x)^{-1/2})D(\phi'_x) \]

and similarly defining \( A_y \). Our system becomes:

\[ A_x V(2) + V(2) A_y^T = G(2) \]

where

\[ V(2) := D((\phi'_x)^{-1/2})U(2)D((\phi'_y)^{-1/2}) \]
\[ G(2) := D((\phi'_x)^{-1/2})F(2)D((\phi'_y)^{-1/2}) \]

Notice that \( A_x \) and \( A_y \) are symmetric for this selection of the weight. We can rewrite this again as a large sparse system using the Kronecker sums as we did in the previous section:

\[ A^{(2)} co(V^{(2)}) = co(G^{(2)}) \]

where \( A^{(2)} \) is a \( m_x m_y \times m_x m_y \) matrix:

\[ A^{(2)} := I_{m_y} \otimes D(\phi'_x)A((\phi'_x)^{-1/2})D(\phi'_x) + D(\phi'_y)A((\phi'_y)^{-1/2})D(\phi'_y) \otimes I_{m_x}. \]

The vectors have form:

\[ co(V^{(2)}) = co(D(v)U^{(2)}D(w) = [D(w) \otimes D(v)]co(U^{(2)}) \]
\[ := D_{wv}co(U^{(2)}) \]
\[ co(G^{(2)}) = D_{wv}co(F^{(2)}) \]

and have dimensions \( m_x m_y \times 1 \). Figure 5.2 on page 206 of the book illustrates what the matrix \( A^{(2)} \) may look like.

We have two methods for solution - diagonalization or using Schur factorization. We have shown each of these above for the general Sylvester’s formula. In fact, the second method is more general and efficient to calculate, given a good method for calculating the Schur factorization.
4 Formulation of the 3d Poisson Problem.

We consider the Poisson problem in $\mathbb{R}^3$, given by:

$$-\Delta^{(3)} u(x, y, z) = f(x, y, z)$$

$$(x, y, z) \in S = (0, 1) \times (0, 1) \times (0, 1)$$

$$u(x, y, z) = 0, \quad (x, y, z) \in \partial S.$$ 

The approximate solution is given by

$$u_{m_x, m_y, m_z}^S(x, y, z) = \sum_{k=-M_x}^{N_x} \sum_{j=-M_y}^{N_y} \sum_{i=-M_z}^{N_z} u_{ijk}^S S_{ijk}(x, y, z)$$

where

$$S_{ijk}(x, y, z) = (S(i, h_x) \circ \phi(x))(S(j, h_y) \circ \phi(y))(S(k, h_z) \circ \phi(z))$$

and

$$m_q = M_q + N_q + 1$$

for $q = x, y, z$. The map for the basis functions is denoted by

$$\phi(z) = \ln \left( \frac{z}{1-z} \right).$$

The inner product is defined to be

$$\langle f, g \rangle = \int_0^1 \int_0^1 \int_0^1 f(x, y, z)g(x, y, z)w(x, y, z)dxdydz.$$ 

where

$$w(x, y, z) = \frac{1}{\sqrt{\phi'(x)\phi'(y)\phi'(z)}}$$

As before, we orthogonalize the residual:

$$\langle -\Delta^{(3)} u - fS_{pqr} \rangle = 0$$

We get rid of the derivatives on $u$ in this equation by using one of Green’s identities:

$$\int_0^1 \int_0^1 \int_0^1 -\Delta^{(3)} u(x, y, z)S_{pqr}(x, y, z)w(x, y, z)dxdydz$$

$$= \int_0^1 \int_0^1 \int_0^1 -u(x, y, z)\Delta^{(3)}(S_{pqr}(x, y, z)w(x, y, z))dxdydz + B_{T_3}$$

where $B_{T_3}$ consists of boundary terms, which we assume to be 0. This leads to the following discretization:

$$A^{(2)\text{mat}}(V^{(3)}) + \text{mat}(V^{(3)})A_z = \text{mat}(G^{(3)})$$
where 
\[ A^{(2)} := I_{m_y} \otimes A_x + A_y \otimes I_{m_z} \]
and \( A_z \) is defined by
\[ A_z := D(\phi'_x)A((\phi'_z)^{-1/2})D(\phi'_y) \]
The unknowns are given by
\[ \text{mat}(V^{(3)}) = D\left((\phi'_y)^{-1/2}\right) \otimes D\left((\phi'_x)^{-1/2}\right) \text{mat}(U^{(3)})D\left((\phi'_z)^{-1/2}\right) \]
\[ := D_{yz}\text{mat}(U^{(3)})D_z. \]
and the forcing term:
\[ \text{mat}(G^{(3)}) = D\left((\phi'_y)^{-1/2}\right) \otimes D\left((\phi'_x)^{-1/2}\right) \text{mat}(F^{(3)})D\left((\phi'_z)^{-1/2}\right) \]
\[ := D_{yz}\text{mat}(F^{(3)})D_z. \]
Finally, we have the term \( \text{mat}(U^{(3)}) \):
\[ \text{mat}(U^{(3)}) = \text{mat}((u^*_i,j)) = [\text{co}(u^*_i,j,-M_z), \text{co}(u^*_i,j,-M_z+1), \ldots, \text{co}(u^*_i,j,N_z)] \]
Note that \( \text{mat}(U^{(3)}) \) is a \( m_xm_y \times m_z \) matrix. Rewriting in terms of Kronecker sums, we have:
\[ (I_{m_z} \otimes A^{(2)} + A_z \otimes I_{m_ym_z})\text{co}(V^{(3)}) = \text{co}(G^{(3)}) \]
\[ \text{co}(\text{mat}(V^{(3)})) = \text{co}(V^{(3)}) \]
\[ = [D((\phi'_y)^{-1/2}) \otimes D((\phi'_x)^{-1/2}) \otimes D((\phi'_z)^{-1/2})]\text{co}(U^{(3)}) \]
\[ := D_{zyz}\text{co}(U^{(3)}) \]
\[ \text{co}(\text{mat}(G^{(3)})) = D_{zyz}\text{co}(F^{(3)}) \]
Setting:
\[ A^{(3)} := I_{m_z}A^{(2)} + A_z \otimes I_{m_ym_z} \]
\[ = I_{m_z} \otimes I_{m_y} \otimes A_x + I_{m_z} \otimes A_y \otimes I_{m_z} + A_z \otimes I_{m_y} \otimes I_{m_z} \]
Then,
\[ A^{(3)}\text{co}(V^{(3)}) = \text{co}(G^{(3)}). \]
We can solve this problem through similar methods as before.
5 The Heat Equation

One of the classic problems in PDEs is the heat equation:

\[ P^{(2)} u(x,t) := \frac{\partial u(x,t)}{\partial t} - \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t) \]

\[ u(0,t) = u(1,t) = 0, t > 0 \]

\[ u(x,0) = 0, 0 < x < 1 \]

We will solve this with sinc methods. Assume that we have an approximate solution, defined by:

\[ u_{m_x,m_t}(x,t) = \sum_{j=-M_x}^{N_x} \sum_{i=-M_t}^{N_t} u_{ij} S_{ij}(x,t) \]

where \( m_x = M_x + 1 \), \( m_t = M_t + N_t + 1 \). Also,

\[ S_{ij}(x,t) = (S(i,h_x) \circ \phi(x))(S(j,h_t) \circ \tau(t)) \]

For this section, we chose:

\[ \phi(z) = \ln \left( \frac{z}{1-z} \right) \]

\[ \tau(t) = \ln(t) \]

I diverge from the book’s hand-waving derivation, and consider the two one-dimensional problems:

\[ u_{xx}(x,t_\ell) = f(x,t_\ell) + u(x,t_\ell) := f_\ell(x), 0 < x < 1 \]

\[ u(0,t_\ell) = u(1,t_\ell) = 0, -M_t \leq \ell \leq N_t \]

and

\[ -u_t(x_k,t) = f(x_k,t) - u_{xx}(x_k,t) := g_k(t), 0 < t \]

\[ u(x_k,0) = \lim_{t \to \infty} u(x_k,t) = 0, -M_x \leq k \leq N_x. \]

Define the one-dimension approximations \( u_{m_x,m_y}(x,t_\ell) \) and \( u_{m_x,m_y}(x_k,t) \) appropriately. We have already solved these problems once before. For \(-M_t \leq \ell \leq N_t,\)

\[ A(v)D(\phi' v)\bar{u}(\ell) = D \left( \frac{f_\ell v}{\phi'} \right) \bar{1}. \]

For \(-M_x \leq k \leq N_x:\)

\[ B(w)D(\tau' w)\bar{u}(k) = D \left( \frac{g_k w}{\tau'} \right) \bar{1}. \]
Following the derivations on pages 136 and 201, we end up with the following system:

\[
D(\phi')A(v)D(\phi')V^{(2)} + V^{(2)}D(\sqrt{\tau'})B(w)D(\sqrt{\tau'}) = G^{(2)}
\]

\[
V^{(2)} = D(v)U^{(2)}D\left(\frac{u}{\sqrt{\tau'}}\right)
\]

\[
G^{(2)} = D(w)F^{(2)}D\left(\frac{u}{\sqrt{\tau'}}\right)
\]

Note: In my derivation, I get terms of \(D\left(\frac{1}{\tau'}\right)\). Not sure how to resolve this.

This is a system of the form

\[
A_xV^{(2)} + V^{(2)}B_t^T = G^{(2)}
\]

where \(B_t\) is defined appropriately, and we chose:

\[
v(x)w(t) = \sqrt{\frac{\tau'(t)}{\phi'(x)}}
\]

Under appropriate conditions, we can show that

\[
\|u - u_{m,r,m_t}\|_\infty \leq KM^2_x\exp\left(-\pi d\alpha_s M_x^{1/2}\right)
\]

We can reformulate this in terms of Kronecker sums:

\[
\mathcal{B}^{(2)}co(V^{(2)}) = co(G^{(2)})
\]

where

\[
\mathcal{B}^{(2)} := I_{m_t} \otimes A_x + B_t \otimes I_{m_x}
\]

and

\[
\text{co}(V^{(2)}) = co(D((\phi')^{-1/2})U^{(2)})
\]

\[
= \left(I_{m_t} \otimes D((\phi')^{-1/2})\right)co(U^{(2)}).
\]