Moment Questions

dreamable

March 2024

1 Problem

In the book *IC Interconnect Analysis*[1], it is stated that the first $2k$ moments may be preserved after net reduction. For example, Theorem 6.3 (subsection 6.4.1, page 199) demonstrates that the block Arnoldi method preserves the first $2k$ moments for RLC circuits with a symmetric formulation. Additionally, Theorem 6.4 (subsection 6.4.2, page 202) asserts that the block Lanczos method preserves the first $2k$ block moments. Furthermore, subsection 6.8.1 (page 228) indicates that PRIMA produces the same results for symmetric RC circuits as the block Lanczos and block Arnoldi methods, and therefore, the first $2k$ moments are preserved.

However, our implementation of PRIMA only preserves the first k moments, even for symmetric RC circuits. The moments from the $(k+1)$-th to the $2k$-th are significantly different, exceeding the range of numerical calculation errors. Why is this the case?

2 Counter Example

We have meticulously checked the code multiple times and are quite confident that the implementation is correct. Consequently, we have revisited the theoretical properties to ascertain their accuracy. We employed a straightforward circuit as a counterexample, enabling manual calculation of net reduction.

The case in point is a simple 4-stage RC circuit, as depicted in Figure 1. All resistors and capacitors are of equal value 1 ($r_1 = r_2 = r_3 = r_4 = 1, c_1 = c_2 = c_3 = c_4 = 1$).

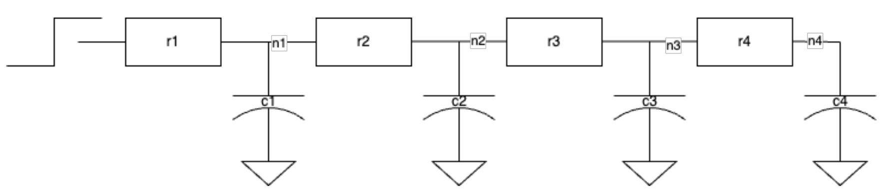

Figure 1: 4-stage RC circuit

For this circuit, the following applies:

$$\begin{aligned} GV + C\frac{dV}{dt} &= b \cdot V_{in} && (1) \\ G &= \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ b &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

which translates to:

$$\begin{aligned} V + A\frac{dV}{dt} &= r \cdot V_{in} && (2) \\ A &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix} \\ r &= \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \end{aligned}$$

This configuration clearly represents a symmetric RC circuit.

Focusing on the last node n_4, we obtain:

$$\begin{aligned} V_{out} &= l^T \cdot V && (3) \\ l &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

We calculate the moments of the original system with the transfer function:

$$\begin{aligned} H(s) &= \frac{V_{out}(s)}{V_{in}(s)} && (4) \\ &= l^T \cdot (I + sA)^{-1} \cdot r \end{aligned}$$

The first four moments are $[m_0, m_1, m_2, m_3] = [1, 10, 85, 707]$.

We then reduce the 4-dimensional system to 1-dimensional using the block Arnoldi method (a special case with $N = 1$ for each block). The results are manually determined as:

$$
\begin{aligned}
V &= [V_0] = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \\
H &= [7.5]
\end{aligned} \tag{5}
$$

Upon computing the moments of the reduced systems, we find $[m_0, m_1] = [1, 7.5]$. It is evident that m_0 is preserved, but not m_1.

Furthermore, we reduced the system to 2-dimensional, and the moments calculated were $[m_0, m_1, m_2, m_3] = [1, 10, 84.3571, 700.5510]$. Once again, only the first k moments are preserved, not $2k$.

This discrepancy suggests that while the block Arnoldi method is supposed to preserve the first $2k$ moments, in our implementations for reducing the system dimensions, it only preserves the first k moments. This raises questions about the specific conditions under which the theoretical guarantees of moment preservation apply, or if there might be an overlooked aspect in our implementation or the theoretical framework.

3 Proof Question

We have scrutinized the proof of Theorem 6.3 (there is no proof provided for Theorem 6.4). The proof for Lemma 6.1 appears to be valid, and the results from the reduction align with it. However, our findings conflict with equation 6.61 from the proof of Theorem 6.3. The equation in question is:

$$B^T A^i = B^T V_q H_q^i V_q^T \tag{6}$$

Taking a special case where $i = 0$, we identify that the left term is $l^T = [0, 0, 0, 1]$, but the right term becomes:

$$
\begin{aligned}
V \cdot V^T &= \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} \\
l^T \cdot V \cdot V^T &= [0.25, 0.25, 0.25, 0.25]
\end{aligned} \tag{7}
$$

According to the proof, Equation 6.61 is supposed to follow from manipulations based on Lemma 6.1. However, the connection between Lemma 6.1 and Equation 6.61 remains unclear to us.

Reviewing the proof of Lemma 6.1 (on page 234), we question whether Equation 6.153 can be universally applied. While it is true that $r = VV^T r$, the assertion $l^T = l^T VV^T$ seems invalid, given $r = [1, 1, 1, 1]$ and $l = [0, 0, 0, 1]$. As demonstrated in Equation 7, each row of VV^T sums up to 1, but the vector l only selects one element from these. If Equation 6.153 cannot be universally applied, then the same logic may not extend to Equation 6.61 neither.

4 Conclusion

Based on our analysis, there might be an error in Theorem 6.3, and possibly by extension, in Theorem 6.4 as well. The net reduction method appears to preserve the first k moments, but it may not preserve the first $2k$ moments, even in the case of symmetric RC circuits. This discrepancy necessitates a further review of the underlying proofs and the conditions under which these theorems are assumed to hold.

5 Matlab Code

Here are the Matlab codes we use.

5.1 Arnoldi

```
% Standard Arnoldi to reduce kr(A,b)
function [Qn,Hn,Q,H]= arnoldi(A,r,order)
n = size(A,1); order=min(order,n);
Q = zeros(n,order+1); H = zeros(order+1,order);
Q(:,1) = r/norm(r);
for i=1:order
    v=A*Q(:,i);
    % for j =1:i
    for j =1:max(i,2) % optimize to j=1:2 for symmetric case
        H(j,i) = Q(:,j)'*v;
        v = v - H(j,i)*Q(:,j);
    end
    H(i+1,i)=norm(v);
    Q(:,i+1)=v/H(i+1,i);
end
Hn = H(1:order,:); Qn=Q(:,1:order);
```

5.2 Circuit formulation

```
function [A,r,G,C]=rc_network(n)
res = ones(n,1); caps = ones(n,1); srcs=(0:n-1)';
g = 1./res;
Ag = eye(n);  Ag(n+1:n+1:end)=-1;
G = Ag*diag(g)*Ag';C=diag(caps);
```

```
b = zeros(n,1); b(1) = 1/res(1); L=eye(n);
A = G\C; r=G\b;
```

5.3 Testing script

```
% try with 4 serial RC with all value=1
n = 4;
[A,r]=rc_network(n);
% original momements
L = eye(n); l = L(:,end); % work on last node
ms = zeros(n,1);
for i=1:n
    ms(i) = l'*A^(i-1)*r;
end

% reduce to order=1/2
for q=1:2
    [V,H]=arnoldi(A,r,q);
    Ar = H; rr = V'*r; lr = V'*l;
    % reduced system moments
    msr = zeros(2*q,1);
    for i=1:2*q
        msr(i) = lr'*Ar^(i-1)*rr;
    end
    % First q match
    assert(max(abs(msr(1:q)-ms(1:q)))<1e-12);
    % 2*q doesn't match
    max(abs(msr(q+1:2*q)-ms(q+1:2*q)))

    % verify A^i*r = V*H^i*V'*r
    for i=0:q-1
        diff = A^i*r - V*H^i*V'*r;
        assert(max(abs(diff))<1e-12);
    end
    % verify l'*A^i = l'*V*H^i*V'
    for i=0:q-1
        diff = l'*A^i - l'*V*H^i*V';
        max(abs(diff))
    end
end
```

References

[1] Mustafa Celik, Lawrence Pileggi, and Altan Odabasioglu. *IC interconnect analysis*. Springer Science & Business Media, 2002.