Bilişimsel Elektromanyetikte İç İçe Çözücülerin Etkin Öniyileştiriciler Olarak Kullanılması†

Tahir Malas[1] ve Levent Gürel[1, 2]
[1]Elektrik ve Elektronik Mühendisliği Bölümü
[2]Bilişimsel Elektromanyetik Araştırma Merkezi (BiLCEM)
Bilkent Üniversitesi, Ankara, 06800
E-posta: tmalas@ee.bilkent.edu.tr, lgurel@bilkent.edu.tr

Özet: *Bu çalışmada, elektrik alan integral denklemi (EAİD) tarafından oluşturulan* $\overline{\mathbf{Z}} \cdot \mathbf{x} = \mathbf{b}$ *şeklindeki sistemlerin etkili öniyileştirilmesi üzerinde yoğunlaşılmıştır. EAİD yüksek derecede kötü durumlu matrisler oluşturmaktadır, fakat açık geometriler için kullanılabilen tek formülasyondur. Bu sistemlerin çok seviyeli hızlı çokkutup yöntemi ile çözümünde, sadece dolu katsayı matrisinin oldukça seyrek bir kısmına denk gelen yakın alan etkileşimleri hafızada tutulmaktadır. Kuvvetli bir öniyileştirici elde edebilmek için yakın alan matrisi* $\overline{\mathbf{Z}}^{\mathrm{YA}}$ *'nin içerdiği bilginin verimli ve etkili bir şekilde kullanılması gerekmektedir. Bu amaçla, yakın alan matrisinin iteratif çözümünü öniyileştirici olarak kullanmak, öniyileştiricilerin iterasyondan iterasyona değişmesine izin veren bir 'esnek' çözücü kullanmak koşuluyla mümkündür. Çeşitli geometrilerle yapılan deneyler, bu yöntemin büyük problemlerde etkili çalıştığını ve çözüm zamanını önemli miktarlarda düşürdüğünü ortaya koymuştur.*

1. Giriş

Elektrik alan integral denklemi (EAİD) ile modellenen, dalga boyu cinsinden büyük problemlerin iteratif çözümü, oluşan matrislerin giderek tekilliğe yaklaşmalarından ötürü zorlaşmaktadır. Oluşan matrisler aynı zamanda yoğun olduğu için, sistem matrisi $\overline{\mathbf{Z}}$ 'nin sadece yakın etkileşimlerine denk gelen seyrek $\overline{\mathbf{Z}}^{\mathrm{YA}}$ matrisini bellekte tutan çok seviyeli hızlı çokkutup yöntemi (ÇSHÇY) (MLFMA: multilevel fast multipole algorithm) [1] kullanılmaktadır. İterasyon sayısını düşürebilmek için $\overline{\mathbf{Z}}^{\mathrm{YA}}$ 'nın içerdiği bilginin verimli bir şekilde öniyileştirici olarak kullanılmasına ihtiyaç vardır. Bu amaçla, $\overline{\mathbf{Z}} \cdot \mathbf{a} = \mathbf{v}$ yerine $(\overline{\mathbf{Z}}^{\mathrm{YA}})^{-1} \cdot \overline{\mathbf{Z}} \cdot \mathbf{a} = (\overline{\mathbf{Z}}^{\mathrm{YA}})^{-1} \cdot \mathbf{v}$ sisteminin çözümü mantıklı gözükse de, seyrek matrislerin LU ayrışımı esnasında oluşan çarpanların seyreklik özelliğini kaybetmesinden dolayı çok pahalı olmaktadır.

Bu zorluğun üstesinden gelebilmek için $\overline{\mathbf{Z}}^{\mathrm{YA}}$ 'yı yaklaşık çarpanlarına ayıran eksik LU (ELU) öniyileştiricilerini kullanmak mümkündür. Paralel gerçekleştirimlerde ise $\overline{\mathbf{Z}}^{\mathrm{YA}}$ 'nın tersine direk yaklaşan seyrek yaklaşık ters (SYT) (SAI: sparse approximate inverse) öniyileştiricisi kullanılmaktadır [2]. Ancak, seyrek sistemlerin iteratif çözümü ucuz olduğu için, bir diğer yaklaşım, her iterasyonda $\overline{\mathbf{Z}}^{\mathrm{YA}} \cdot \mathbf{x} = \mathbf{y}$ sistemini iteratif çözmektir. Bu çalışmada EAİD ile modellenen problemlerde bu yöntemin diğer öniyileştirme yöntemlerinden (ELU veya SYT) daha etkili olduğu sayısal deneylerle gösterilmiştir.

2. Esnek Öniyileştirme

Öniyileştirme metodlarında sistem matrisi $\overline{\mathbf{Z}}$ 'nin tersine yaklaşan bir $\overline{\mathbf{M}}$ operatörü seçilerek $\overline{\mathbf{Z}} \cdot \mathbf{a} = \mathbf{v}$ yerine $\overline{\mathbf{M}}^{-1} \cdot \overline{\mathbf{Z}} \cdot \mathbf{a} = \overline{\mathbf{M}}^{-1} \cdot \mathbf{v}$ (soldan öniyileştirme), veya $(\overline{\mathbf{Z}} \cdot \overline{\mathbf{M}}^{-1}) \cdot (\overline{\mathbf{M}} \cdot \mathbf{a}) = \mathbf{v}$ (sağdan öniyileştirme) çözülebilmektedir. Bu ifadelerde öniyileştirici, ELU ve ya SYT'de olduğu gibi bir matris olabileceği gibi, doğrusal sistem operatörüne yaklaşan bir başka *operatör* de olabilmektedir [2]. Örneğin, ÇSHÇY bağlamında, pahalı bir yöntem olan $\overline{\mathbf{Z}}^{\mathrm{YA}}$ 'nın doğrudan tersini almak yerine, Krylov altuzayı çözücülerine her adımda

† Bu çalışma, TÜBİTAK (105E172), Türkiye Bilimler Akademisi (LG/TÜBA-GEBİP/2002-1-12), ASELSAN ve SSM tarafından desteklenmektedir.

$\overline{\mathbf{Z}}^{\mathrm{YA}} \cdot \mathbf{x} = \mathbf{y}$ sisteminin iteratif çözümünü vermek mümkündür. Öniyileştirici operatörü sabit olmadığı, her adımda değişebileceği için bu yönteme "değişken öniyileştirme" (variable preconditioning) denmektedir.

Değişken öniyileştirmede, öniyileştirme operatörü sabit olmadığı için, Krylov altuzayını geren vektörlerin öniyileştirme operatörü uygulanmış hallerinin hafızada tutulması gerekmektedir. GMRES'in bu formuna "FGMRES" ismi verilmiştir [2]. İteratif öniyileştirme işlemi için de yine ilk iterasyonlarda hızlı düşüş sağlayan GMRES çözücüsü kullanılmıştır. Ayrıca, iteratif öniyileştirme işleminde de yine sabit bir öniyileştirici kullanmak münkündür. Bu sayede iç içe çözücülerin kullanıldığı bir çözüm metodu ortaya çıkmıştır. Şekil 1'de bu yöntem gösterilmiştir.

Dış çözücü **FGMRES:** $\overline{\mathbf{Z}} \cdot \mathbf{a} = \mathbf{v}$ çözümü
Matris-vektör çarpımı: ÇSHÇY
İç çözücü (öniyileştirici) **GMRES:** $\overline{\mathbf{Z}}^{\mathrm{YA}} \cdot \mathbf{x} = \mathbf{y}$ 'nın yaklaşık çözümü
$\overline{\mathbf{Z}}^{\mathrm{YA}}$ ile seyrek matris-vektör çarpımı
Sabit öniyileştirici: **ELU** veya **SYT**

Şekil 1. İç içe çözücülerin kullanımı.

Bu yöntemde önemli bir konu iç sistemin nasıl çözüleceğinin belirlenmesidir. İç çözücü asıl sistemin öniyileştiricisi olarak kullanılacağı için, genellikle çok iyi bir çözüme ihtiyaç bulunmamaktadır. Yakın alan sisteminin çözüldüğü bu örnekte, hata vektörünün onda birine indirilmesi yeterli olmuştur. Daha iyi çözümler dış iterasyon sayısını düşürmemiş, ancak öniyileştirme işleminin maliyetinin artmasından dolayı çözüm zamanını artırmıştır.

3. Sayısal Deneyler

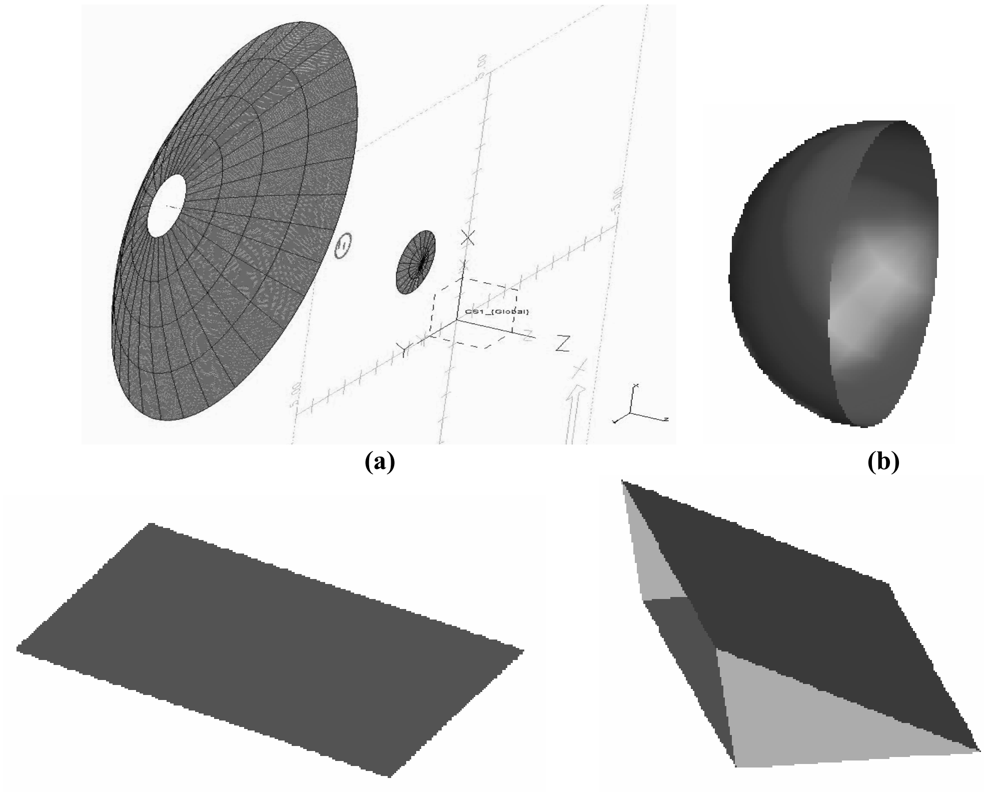
Esnek öniyileştirme yönteminin performansını test etmek amacı ile Şekil 2'de gösterilen geometrilerin çözümleri yapılmıştır. İterasyonlara ilk tahmin olarak sıfır vektörüyle başlanmış, kalan vektörü $\|\mathbf{r}_0\| / \|\mathbf{r}_k\| < 10^{-6}$ şartını sağladığında iterasyon durdurulmuştur. Tüm öniyileştiriciler sağdan uygulanmıştır. İç çözücü için ise durdurma kriteri $\|\mathbf{r}_0\| / \|\mathbf{r}_k\| < 0.1$ alınmış, aynı zamanda yapılan en fazla matris-vektör çarpımı da beşle sınırlandırılmıştır.

Tablo 1'de Şekil 2'deki problemlerin çözümlerinin iterasyon sayıları ve çözüm zamanları verilmiştir. Bu tabloda "LU" yakın alan matrisinin tam olarak çarpanlarına ayırımı ile elde edilmiştir. Bu öniyleştiricinin hem bellek kullanımı, hem de çözüm zamanı açısından çok pahalı olduğu için pratik kullanımı yoktur. Sınanan matrislerin yakın alan matrisine ne derece iyi bir yaklaşım sunduğunu anlamak amacı ile denektaşı (benchmark) olarak kullanılmıştır. Ayrıca, büyük problemlerde "LU" çözümü çok fazla bellek gerektirdiği için sunulamamıştır. "YA/SYT" ise yakın alan matrisinin SYT öniyileştiricisi kullanılarak iteratif çözümü ile elde edilmiştir. Bu yüzden SYT'nin ve YA/SYT'nin yapım zamanları ortaktır. Diyagonal öniyileştirici benzeri basit öniyileştiriciler, problemlerin çoğunda yakınsama sağlamadıklarından tabloya konmamıştır.

Tablo 1'de elde edilen sonuçlar YA/SYT'nin SYT'ye oranla daha kısa sürede çözüme ulaştığını göstermektedir. Ayrıca YA/SYT'nin yakın alan matrisinin tam çözümünü ifade eden LU öniyileştiricisine çok yakın iterasyon sayıları ürettiği için, yakın alan matrisi kullanılarak elde edilebilecek en verimli öniyileştiricilerden biri olduğu ortaya çıkmıştır.

Sonuç

Açık yüzeye sahip ışınım ve saçılım problemlerinin çözümünde, çözümü hedeflenen nesnelerin dalga boyu cinsinden büyüklükleri arttıkça, oluşan büyük ve yoğun matrislerin iterative çözümü zorlaşmaktadır. Bu çalışmada, iç içe iterasyonlar kullanılarak bu tarz büyük problemlerin kısa sürede çözümü sağlanmıştır. Ayrıca bu yaklaşım, yakın alan matrisinin iyice seyreldiği daha da büyük problemlerin çözümünde ÇSHÇY'nin de öniyileştirici olarak kullanımına imkan sağlamıştır.



(c) (d)

Şekil 2. Çözümü yapılan geometriler: (a) Yansıtıcı anten, (b) yarım küre, (c) levha, (d) açık prizma.

Geometri	Bilinmeyen Sayısı	LU	SYT Yapım Zamanı (s)	SYT		YA/SYT	
		İter		İter	Çözüm Zamanı (s)	İter	Çözüm Zamanı (s)
Levha	12,249	26	4	44	12	29	9
	137,792	53	52	91	336	59	253
	719,000	BSA	214	190	4,091	141	3,369
Açık Prizma	11,351	46	21	80	33	52	25
	127,925	112	156	172	628	120	520
	409,514	BSA	336	389	4,601	209	3,476
Yarım Küre	9,911	38	7	60	24	40	17
	116,596	93	77	156	510	103	383
Yansıtıcı Anten	356,439	BSA	952	125	878	71	646

Tablo 1. Öniyileştiricilerin Şekil 1'deki geometriler için çözüm performanslarının karşılaştırılması. "BSA" bellek sınırının aşıldığı durumları göstermektedir.

Kaynaklar

[1] Lu C.-C. ve Chew W.C., "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propagat.*, cilt 45, no. 10, s. 1488-1493, Ekim 1997.

[2] Saad Y., *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, ABD, 2003.