

Scilab 固有値取得関数 spec について

伊藤榮信

2023年7月25日

ϕ に関する微分方程式とそれに含まれる固有値に対し、定義域を N 分割し、未知関数を ϕ を ϕ_i , $i = 1, \dots, N + 1$ と離散化する:

$$\phi = (\phi_1, \phi_2, \dots, \phi_{N+1})^T \quad (1)$$

各離散点では微分演算子を微分演算行列 (例えば, Chebyshev differentiation matrices) で扱う場合、微分方程式内の各導関数の係数により微分演算行列と同じサイズの係数行列が生成される。微分方程式内に含まれる固有値を c とすると、離散化された微分方程式は

$$A\phi_i = cB\phi_i \quad (2)$$

となる。もしくは、 ϕ_i を省略して

$$A = cB \quad (3)$$

と表現もできる。このように既知の A, B から固有値 c を求めることを一般化固有値問題という。

これに、微分方程式が表す領域の境界で Dirichlet 条件を課した場合、係数行列の外枠となる第 1

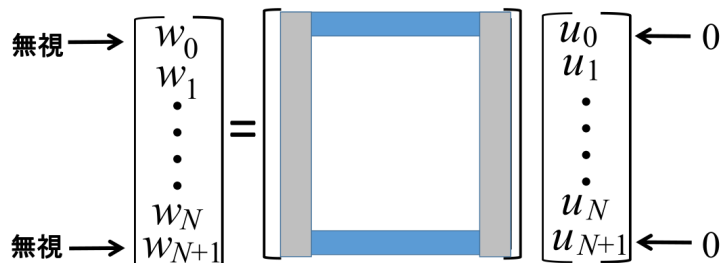


図 1: u_0 と u_{N+1} をゼロとする。その結果の w_0 と w_{N+1} は要素として考えなくてよい。

列と最後の列 (例えば第 $N + 1$ 列) には特定の数値が入られる。特に、境界で 0 を指定する場合についてここでは議論する。

図 1 で説明すると、いま未知関数 u (上の ϕ に相当) の定義区間の両端点の値を u_0 と u_{N+1} とする。 u の境界条件として定義区間の両端点で $u_0 = u_{N+1} = 0$ とした場合、最初の列と最後の列が消去され、未知数が 2 つ減少し、方程式と未知数の数に差が現れるから、行に関しても第 1 行と第 $N + 1$ 行をゼロとする。これにより $(N + 1, N + 1)$ サイズの行列は $(N - 1, N - 1)$ の行列となる。

よって、(3) 式で、両端で $\phi_i = 0, i = 1, N + 1$ となる境界条件を課した場合、サイズ $(N + 1, N + 1)$ の行列 A, B は、各々 $(N - 1, N - 1)$ の行列となる。したがって、一般化固有値を求める場合、元の行列 A, B により Scilab 関数 $\text{spec}(A, B)$ により固有値を求めるのではなく、 $(N - 1, N - 1)$ となった行列 A_1, B_1 により固有値を求めることとなる。

そこで、 $A, B \rightarrow A_1, B_1$ への変換であるが、プログラムとして考えられる方法は

```

for i=1:1:N-1,
for j=1:1:N-1,
A1(i,j)=A(i+1,j+1)
B1(i,j)=B(i+1,j+1)
end
end
end

```

とすることである。しかしこれは、 $(N-1, N-1)$ サイズの行列の全要素を一つずつ置きかえることで時間を要する。したがって、行列を縮小するのではなく、行列のサイズはそのまま $(N+1, N+1)$ で図 1 に示した通り、行列の外枠部分の要素を全て 0 にした行列の固有値で数値的に代用できないか。関数 `spec` の内容が分かっているならこの真偽は明らかだが、関数 `spec` をブラックボックスとして利用する場合、このような単純な疑問がわく。その外枠をゼロとするプログラムは

```

for i=1:1:N+1          //第 1 列と第 N + 1 列の要素をゼロにする
A(i,1)=0
A(i,N+1)=0
B(i,1)=0
B(i,N+1)=0
end
for j=2:1:N           //第 1 行と第 N + 1 行の要素をゼロにする
A(1,j)=0
A(N+1,j)=0
B(1,j)=0
B(N+1,j)=0
end
end

```

で、処理回数は圧倒的に少なく、行列を縮小するよりは外周をゼロにする方が効率的である。

$N = 128$ として、実際に Scilab により試行した結果を示す (図 2 参照) :

`spec(A, B) : 0.3486342045 ± 0.0059029453i, spec(A1, B1) : 0.3485763515 ± 0.0060462390i`

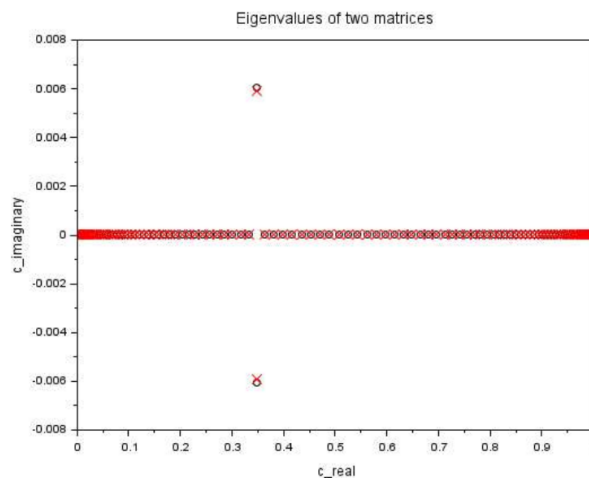


図 2: 赤の×が A, B 行列による固有値で黒○が A_1, B_1 による固有値。横：実軸，縦；虚軸