プログラムのページ

担 当　森 口 繁 一

6305.　3項方程式（1）

藤川洋一郎（立大理学部）

係数行列の対角線要素と，それをはさむ両側の要素を除くすべての要素が0であるような連立一次方程式

$$a_k x_{k-1} + b_k x_k + c_k x_{k+1} = d_k \quad (k=1,\ 2,\ \cdots\cdots,\ n)$$

$$a_1 = c_n = 0$$

を解く．係数ベクトル $[a_k]$，$[b_k]$，$[c_k]$，$[d_k]$ をそれぞれ array A, B, C, D に置いて，この procedure を実行すれば解ベクトル $[x_k]$ が array D にでき上る．解法の方針は，まず対角線要素の左側にある要素 $a_k (k=1, 2, \cdots\cdots, n)$ を，そのすぐ上の式を用いて順々に消去し，対角線より左側にあるすべての要素が0になるように連立一次方程式を変形する．次に，この方程式の第 n 式から逐次代入計算で，x_n，x_{n-1}，……，x_1 の順に解を求める．この二つの段階の前者を前進部，後者を後進部ということにする．

この解法の前進部で，対角線要素に0が現れないことを仮定しているが，0が現われない場合でも，対角線要素が桁落ちによって，絶対値が小さくなる場合には解の精度が落ちることがある．対角線要素の絶対値がその両側の要素の絶対値より大きいほどこの危険性は少なく，精度の良い解が得られるであろう．なお，このプログラムで array A とCの値は最初の状態のままで最後まで変らない．変化するのはDだけである．

```
procedure TRIDG (A, B, C, D, N);
  value N; integer N; array A, B, C, D;
begin integer K; real W;
  for K:=2 step 1 unitl N do
    begin W:=A[K]/B[K-1];
      B[K]:=B[K]-(W×C[K-1]);
      D[K]:=D[K]-W×D[K-1]
    end;
  D[N]:=D[N]/B[N];
  for K:=N-1 step-1 until 1 do
  D[K]:=(D[K]-C[K]×D[K+1])/B[K]
end
```

〔注〕

　　for K:=N-1 **step**-1 **until** 1 **do** ……

の代りに，

　　for K:=N **step**-1 **until** 1 **do** ……

として，この procedure を用いる前に D[N+1]:=0 を行なっておくことにすれば，D[N]:=D[N]/B[N] を省くことができる．またこの procedure によって D[N+1] は何等変化しないから，この procedure を何度も使うときでも D[N+1]:=0 は1度だけ main program で行なっておけばよい．

6306.　3項方程式（2）

藤川洋一郎（立大理学部）

プログラム 6305 と異なる点は，係数ベクトル $[a_k]$，$[b_k]$，$[c_k]$ が不変で，非斉次項ベクトル $[d_k]$ をかえては，それに対応する解ベクトル $[x_k]$ を求めるのに便利なようにしたことである．そのために procedure を TRIDGA と TRIDGB の二つに分けてある．TRIDGA では係数ベクトルについてだけ前進部を適用する．TRIDGB では，TRIDGA で得た新しい係数ベクトルを用いて，与えられた非斉次項ベクトルに前進部と後進部を適用して解ベクトルを array D の上につくる．その他，細かい点であるが後進部で除算を省くためにプログラム 6305 と若干計算の手順を変えてある．その結果，このプログラムでは array C が変化を受けるようになっている．

```
procedure TRIDGA (A, B, C, N);
  value N; integer N; array A, B, C;
  begin integer K; real W;
    for K:=2 step 1 until N do
      begin W:=D[K-1]:=C[K-1]/B[K-1];
          D[K]:=D[K]-A[K]×W
      end
  end
procedure TRIDGB (A, B, C, D, N);
  value N; integer N; array A, B, C, D;
  begin integer K; real W;
    for K:=2 step 1 until N do
      begin W:=D[K-1]:=D[K-1]/B[K-1];
          D[K]:=D[K]-A[K]×W
      end;
   D[N]:=D[N]/B[N];
      for K:=N-1 step-1 until 1 do
      D[K]:=D[K]-C[K]×D[K+1]
  end
```

［応用例］ 拡散方程式の差分近似解法で，次のような3項方程式を反復して解きたいことがある．

$$a_k x_{k-1}{}^{(p+1)}+b_k x_k{}^{(p+1)}+c_k x_{k+1}{}^{(p+1)}$$
$$=c_k x_{k-1}{}^{(p)}+f_k x_k{}^{(p)}+g_k x_{k+1}{}^{(p)}$$
$$k=1, 2, \cdots\cdots, n$$
$$p=0, 1, 2, \cdots\cdots, m$$

ただし $a_1=c_n=e_1=g_n=0$ とする．p は反復回数で，与えられた初期解ベクトル $[x_k{}^{(0)}]$ から出発して，上の3項方程式を解いて $[x_k{}^{(1)}]$, $[x_k{}^{(2)}]$, …… と次々に求めていく．係数ベクトル $[a_k]$, $[b_k]$, $[c_k]$, $[e_k]$, $[f_k]$, $[g_k]$ と初期解ベクトル $[x_k{}^{(0)}]$ を array A,B, C, D, E, F, G, X に読みこんでおいて，反復が1回すむごとに $[x_k{}^{(p)}]$ をタイプするプログラムを作ると次のようになる．ただし $n=10$, $m=100$ とする．

```
begin interge K, P:
    array A, B, C, D, E, F, G [1 : 10];
  <TRIDGA の宣言文全体をここに入れる>;
  <TRIDGB の宣言文全体をここに入れる>;
 for K:=1 step 1 until 10 do
   begin READREAL(A[K]); READREAL
          (B[K]);
          READREAL(C[K]); READREAL
          (E[K]);
          READREAL(F[K]); READREAL
          (G[K]);
          READREAL(X[K])
   end;
 TRIDGA(A, B, C, 10);
 for P:=1 step 1 until 100 do
   begin
     for K:=1 step 1 until 10 do
       D[K]:=E[K]×X[K−1]+F[K]×X[K]
         +G[K]×X[K+1];
       TRIDGB (A, B, C, D, 10);
     for K:=1 step 1 until 10 do
       begin X[K]:=D[K];
         CRLF; PRINTREAL(X[K])
       end
   end
end
```

［注］ TRIDGB の procedure を，解ベクトルが array D の上にでなく array X の上にでき上るように改造しておけば，**array** D の内容を **array** X に移すための **statement** X[K]:=D[K] を省くことができる．

（昭和38年8月17日受付）