

Przedstawione w ramkach kody tworzą macierz Hilberta: kwadratową tablicę liczb $H_{ij} = 1/(i + j - 1)$ (red.).

Wbudowana funkcja Octave hilb ma niemal identyczną implementację jak mhillb (red.).

Najdroższa!

Romeo MONTECCHI*

Julio, od kiedy się poznaliśmy, nie potrafię przestać myśleć o Tobie. . .

Ty wiesz, że przed Tobą były inne, że spędziłem długie lata w związku z Octawą, nie zważając na to, że ona wciąż była zapatrzona w muskularnego MATLAB-a. Nie, nie żałuję tych lat, kiedy ona powoli uczyła mnie wektoryzacji kodu i sprawiła, że w końcu zerwałem z nałogiem nadużywania pętli w programach:

```
1 function H = mhillb(n)
2   H = zeros(n);
3   for i = 1:n
4     for j = 1:n
5       H(i,j) = 1/(i+j-1);
6     end
7   end
8 end
```

Kod w Octave bez wektoryzacji: dwie pętle.
Czas wykonania (dla $n = 2048$): 16,1 s.

```
1 function H = mhillbv(n)
2   H = zeros(n);
3   row = 1:n;
4   for i = 1:n
5     H(i, :) = 1.0 ./ row;
6     row = row+1;
7   end
8 end
```

Kod w Octave zwektoryzowany: jedna pętla.
Czas wykonania (dla $n = 2048$): 0,06 s, to jest 270 razy szybciej.

Przeżywałem piekielne katusze, widząc, jak MATLAB imponował jej (i nie będę Cię oszukiwał, Ukochana – mnie też) tym, że potrafił sam skompilować nasze wielokrotnie zagnieżdżone pętle, wykorzystując kompilator JIT (*just in time*). I był przy tym taki niedostępny, co – o, serce nierozsądne! – tym bardziej przyciągało ich oboje. . .

```
1 function H = mhillb(n)
2   H = zeros(n);
3   for i = 1:n
4     for j = 1:n
5       H(i,j) = 1/(i+j-1);
6     end
7   end
8 end
```

Kod w MATLAB-ie bez wektoryzacji, identyczny z pierwszym.
Czas wykonania (dla $n = 2048$): 0,08 s, to jest 200 razy szybciej.

```
1 function mhillb(n)
2   H = zeros(n,n);
3   for i = 1:n
4     for j = 1:n
5       H[i,j] = 1/(i+j-1)
6     end
7   end
8 end
```

Kod w Julii bez wektoryzacji.
Czas wykonania (od drugiego wywołania, dla $n = 2048$): 0,04 s, to jest 400 razy szybciej.

I to właśnie wtedy nagle Ty pojawiłaś się w moim życiu; i od pierwszej chwili wiedziałem, że nic już nigdy nie będzie takie, jak dawniej. Nasze dusze są niczym dwie połówki jednego owocu. Zawsze mogę napisać Ci to, co mam w głowie, a Ty natychmiast zrealizujesz nasze nawet najbardziej zapętlone pomysły. Wszystkie obliczenia wydają się proste w te dni, gdy jesteśmy razem, a do osiągnięcia szczytu wydajności wystarcza nam zwykle kilka magicznych słów. Choć wiem, że skrywasz w sobie tajemne kompilacje,

o których wolałabyś nie mówić, codziennie cieszy mnie Twoja prostota i radość z prowadzenia błyskawicznych rachunków. Uwielbiam Twoje subtelne, niewymuszone deklaracje typów! Ascetyczną elegancję! I to, że – jak ja, jak Octava, jak wielu innych z nas – wybrałaś życie w wolnym świecie wolnego oprogramowania.

Tęsknię i niecierpliwie czekam na kolejne spotkanie. Tam gdzie zawsze, przy klawiaturze – OK?

Twój R.

Ale za to wiemy, że w każdym z systemów (Octave, MATLAB-ie, Julii) można napisać *jeszcze szybszy* kod generujący macierz Hilberta.

Od redakcji: Powyższy manuskrypt, nadesłany w dwunastym dniu wiosny – zapewne omyłkowo na adres *Delty* – publikujemy bez skrótów, choć przyznajemy, że czasem używane przez zakochanych frazy są dla nas niezbyt zrozumiałe, zapewne wskutek nadmiernej afekcji autora. Niestety, do tej pory nie udało się zweryfikować ze stuprocentową pewnością, kim jest autor powyższego listu, gdyż – wedle posiadanych przez nas danych – z Akademią w Weronie nie jest związana żadna osoba o nazwisku Montecchi (choćaj wynajęci przez nas eksperci potwierdzają, że sam list jest autentyczny). Natomiast, z pomocą dobrze poinformowanych źródeł, udało się nam potwierdzić nie tylko tożsamość Julii, w wersji 1.0, ale nawet ustalić jej adres: www.julialang.org. Jednak aby zaoszczędzić na kosztach przesyłki (wszak honoraria ekspertów mocno nadwątlili nasz budżet), zdecydowaliśmy się opublikować list na łamach niniejszego popularnego periodyku, wierząc, że w ten sposób też dotrze do adresatki.