Opis pakietów `opacity` i `eostate`

Błażej L. Popielski*

Lipiec 1998

1 Pakiet `eostate`

- **cel:** uzyskanie wartości funkcji termodynamicznych dla gazu o określonym składzie chemicznym (X,Z) dla temperatury i gęstości zmieniających się w pewnych zakresach
- **środek:** generowanie „tablicy”[1] funkcji termodynamicznych dla określonego Z (`eostate.dat`) metodą interpolacji tablic dla innych Z, a następnie znajdowanie metodą interpolacji wewnątrz tablicy wartości funkcji termodynamicznych dla konkretnej gęstości i temperatury
- **składniki, wymagania:** 15·`peos??z?`[2] (28MB), `eostate.f` (12KB), kompilator Fortran-a 77, do obsługi wygenerowanego zbioru także: `eostate.04` (55KB) i `data.04` (12KB)
- **autor, źródło danych:** Pakiet (jak wynika z enigmatycznych nagłówków) jest autorstwa *dr.Ryszarda Sienkiewicza.* Dane i część procedur pochodzą natomiast z *Lawrence Livermore National Laboratory.*

1.1 `eostate.dat`

- **uzyskiwanie:** Skompilowany program uruchomić w katalogu zawierającym zbiory `peos??z?`[3] i podać Z; rozmiar zbioru to 9.38MB.
- **czas:** 5 – 10 minut w zależności od komputera
- **zawartość:** Każdej z dyskretnych wartości temperatury, gęstości i zawartości wodoru odpowiada szereg poniższych wielkości (w systemie cgs):
  - wielkość pomocnicza: $\log R$, gdzie $R = \rho/T_6^3, T_6 = T \cdot 10^{-6}$

*e-mail: `popielsk@alkor.astrouw.edu.pl`

[1] W istocie jest to tablica 3-wymiarowa, bo dla stałego Z pozostaje zależność od X,ρ i T.

[2] To są „fabryczne” równania stanu do interpolacji.

[3] Dwa pierwsze znaki zapytania odpowiadają cyfrom Z po przecinku, natomiast ostatni – dziesiętnym częściom X. Komplet danych do interpolacji w obszarze zainteresowania ($Z = 0.00, 0.02, 0.04$ oraz $X = 0.0, 0.2, 0.4, 0.6, 0.8$) wyjaśnia liczbę zbiorów `peos??z?`

- ciśnienie: P/P_0, gdzie $P_0 = T_6 * \rho$
- energia wewnętrzna grama materii na stopień: E/T_6, gdzie E jest w jednostkach $U = 10^{12} erg/g$
- entropia S w jednostkach $U_s = U/(E + 6K)$
- pochodna energii wewnętrznej: $dE/d\rho \cdot \sqrt{P_0}$
- ciepło właściwe: C_v
- pochodne ciśnienia: $(d \log P/d \log \rho)_{T_6}$ oraz $(d \log P/d \log T_6)_\rho$
- wykładniki adiabaty: Γ_1, $\Gamma_2/(\Gamma_2 - 1)$ oraz $\Gamma_3 - 1$

Każdej z tych wielkości odpowiada kolumna tabeli, której wiersze numeruje T_6; maksymalny zakres jej zmienności to $100 \ldots 0.005$. Krok jest zmienny, ale krok względny oscyluje wokół wartości $1/10$. Takiej postaci tabel jest $5 \cdot 121$ ($X = 0.0, 0.2, 0.4, 0.6, 0.8$ oraz 121 gestości). Gęstość zmienia się od $1 \cdot 10^{-14} g/cm^3$ do 10^5 znów ze zmiennym krokiem, przy czym wraz z gęstością rośnie dolna granica T_6 tabel.

1.2 Wykorzystanie zbioru `eostate.dat`

Z danych zawartych w tym zbiorze można uzyskać wartości funkcji termodynamicznych dla X,T i ρ w zakresach „astrofizycznych". Wszystkie potrzebne do tego procedury w języku Fortran znajdują się w zbiorze `eostate.04`[4] dołączonym do pakietu. Oto przykładowy program:

```
implicit double precision(a-h,o-z)
include 'data.04'
dimension xvec(nx),extra(20)

xchem=...
zchem=...
t=...
rho=...

zed=zchem
xvec(1)=xchem
xvec(2)=0.
xvec(3)=1.-xchem-zchem

call state(rho,t,xvec,extra)
...
end

include 'eostate.04'
```

[4] Zbiór ten wymaga do poprawnej kompilacji obecności `data.04`

Korzystanie z tablicy jak z funkcji odbywa sie za pomoca procedury `state`. Powyższy fragment programu sprawia, że funkcje termodynamiczne (w cgs) stają się dostępne przez odwołania:

- `extra(1)` : ciśnienie całkowite (p)
- `extra(2)` : $(d\ln p/d\ln T)_\rho$
- `extra(3)` : $(d\ln p/d\ln \rho)_T$
- `extra(4)` : ciśnienie gazu (p_{gas})
- `extra(5)` : ciśnienie promieniowania (p_{rad})
- `extra(6)` : $(d\ln T/d\ln p)_{ad}$
- `extra(7)` : $(dQ/dT)_\rho = C_v$ (?)
- `extra(8)` : $(dQ/d\rho)_T$ (?)
- `extra(10)` : $(dU/dT)_p = C_p$ (?)
- `extra(11)` : $-(d\ln \rho/d\ln T)_p$ (?)

Wektor `xvec` zawiera obfitości: wodoru (1), helu-3 (2) i helu-4 (3). Linia `zed=zchem` konieczna jest ze względu na użycie poza pakietem ewolucyjnym, do którego opisywane kody były tworzone.

Ostrzeżenie: Błąd wyniku interpolacji w przypadku tego pakietu jest największy poniżej $\rho = 10^{-8} g/cm^3$; zazwyczaj nie przekracza jednak 10% dla wielkości nie będących pochodnymi i 20% dla pochodnych.

Dodatek 1.

Istnieje wersja pakietu okrojona do zastosowań np. pulsacyjnych. Korzysta z tych samych zbiorów `peos??z?`, co `eostate` (choć w tym przypadku dane te są nadmiarowe). Poza tym w skład pakietu wchodzą: `zfseos5.f` (12KB) do generowania tablicy oraz `eostate.s35` (50KB) i `data.s35` (10KB) do jej wykorzystania.

Jedyny input programu to Z. Wynikiem działania jest zbior `eosdata5` (5.2MB) o strukturze identycznej co `eostate.dat`. Zawiera on jednak 4 wybrane funkcje termodynamiczne: ciśnienie, ciepło właściwe C_v oraz pochodne logarytmiczne ciśnienia po ρ i T. Użycie tych wielkości w określonym zakresie temperatury i gęstości dla otoczki[5] ilustruje przykładowy program:

```
      implicit double precision(a-h,o-z)
      include 'data.s35'
      parameter(neqs=6,nres=20)
      dimesion res(nres)
```

[5] Jeśli chcemy modelować wnętrze gwiazdy zamiast procedury `termo` używać powinniśmy – `state`. W tym przypadku przydatne może się okazać posiadanie zbioru `electron.gas` (39KB).

```
double precision ionful

xchem=...
zchem=...
t=...
rho=...

ionful=...

call termo(rho,t,xchem,zchem,ionful,res)
...
end

include 'eostate.s35'
```

Pojawia się tutaj zmienna `ionful`. Przypisanie jej wartosci 1 powoduje, że zaniedbana zostanie częściowa jonizacja materii; podanie 0 – przyjęta całkowita jonizacja.

Tak obliczony wektor `res` zawiera podstawowe parametry równania stanu:

- `res(1)`: ciśnienie całkowite (p)
- `res(2)`: $(d\ln p/d\ln\rho)_T$
- `res(3)`: $(d\ln p/d\ln T)_\rho$
- `res(4)`: energia wewnętrzna na jednostkę masy (u)
- `res(5)`: $(dQ/dT)_p = C_p$
- `res(6)`: $(dQ/dT)_\rho = C_v$
- `res(7)`: $-(d\ln\rho/d\ln T)_p$
- `res(8)`: gradient adiabatyczny (∇_{ad})

Uwaga: Każdy z przedstawionych powyżej pakietów posiada ograniczenie na dokładność Z; jest to 10^{-4}. Zawartość tablicy równania stanu uzyskanej tymi pogramami będzie dobra, w jej nagłówku jednak pojawi się Z z obciętą przed 5-tym miejscem częścią ułamkową (skrajny przypadek dla np. $Z = 0.00001$). To może prowadzić do rozbieżności uniemożliwiających korzystanie z tablic w dotychczasowej formie.
Aby stosować większą dokładność należy zmienić format zapisywanej do pliku wynikowego tabeli i odczytywanej z niego wartości Z. W tej wersji pakietu jest to `f6.4`. Poniżej przedstawiam numery linii, w których trzeba dokonać poprawek formatu drugiej zmiennej w instrukcjach odpowiednio `write` i `read`: `eostate.f` – 222, `eostate.04` – 652 (dla pakietu pełnego) oraz `zfseos5.f` – 216, `eostate.s35` – 490 (dla pakietu okrojonego). Warto wspomnieć, że schodząc z dokładnością poniżej 10^{-6} należy zmniejszyć również dokładność sprawdzania zbieżności Z w okolicznych procedurach.

2 Pakiet `opacity`

- **zadanie:** dostęp do nieprzezroczystości (i pierwszych pochodnych) jako funkcji gęstości, temperatury, obfitości wodoru i obfitości pierwiastków ciężkich
- **sposób:** stworzenie „tablicy”[6] `opacity.dat` (160KB) zawierającej współczynniki nieprzezroczystości (opcjonalnie z pochodnymi po ρ i T) dla konkretnego Z, a następnie interpolacja wartości współczynnika dla dowolnych parametrów (X, ρ, T) za pomocą procedury dołączanej do programu (przydatne do modelowania zjawisk, w których potrzebna jest zmienność X np. pakietów ewolucyjnych – z czasem maleje X)
- **składniki pakietu, wymagania:** zbiory: `opacity.f` (90KB), `GN93hz.dat`[7] (375KB) oraz `Alexande.dat`[8] (1.22MB), kompilator standardu Fortran 77; do obsługi też: `opacity.04` (6KB) i `data.04` (11KB)
- **autor, źródło:** Autorem pakietu jest *dr.Ryszard Sienkiewicz*, dane pochodzą od 2 zespołów wymienionych w przypisach, a zrąb procedur – z pakietu *LLNL Opal*.

2.1 `opacity.dat`

- **tworzenie:**

  1. *same nieprzezroczystości* : Należy upewnić się, że w 1989. linii lub na początku procedury `opacity2` w zbiorze `opacity.f` jest przypisanie `ipul=0`. Zbiór powstaje po uruchomieniu skompilowanego `opacity.f` i podaniu obfitości pierwiastków ciężkich.
  2. *nieprzezroczystości i pochodne po ρ i T* : W zbiorze `opacity.f` powinno być `ipul=3`. Następnie uruchamiamy skompilowany program.

  Przy prawidłowym uruchomieniu kodu powstaje też 7 zbiorów (`x??`), przydatnych podczas interpolacji.[9]

  Uwaga: Dla zwiększenia dokładności tablice mogą być generowane z mniejszym krokiem w ρ i T. Uzyskuje się to przez zwiększenie wartości całkowitej zmiennej `igrid` w kodzie `opacity.f`[10] (procedura `opacity2`). Należy wtedy pamiętać, że program generujący zbiór `opacity.dat` podaje wartości zmiennych, które muszą być ustawione w procedurach interpolujących w `opacity.04`[11], aby zawartość tablicy była poprawnie odczytana. Zwykle

[6] Jest to znów zależność 3-wymiarowa : ρ, T, X. Tym razem jednak zbiór jest binarny.
[7] nieprzezroczystości OPAL – Rogers,F.J. & Iglesias,C.A., 1995 ASP Conf.
[8] wkład do nieprzezroczystości od cząsteczek – Alexander,D.R. & Ferguson,J.W., 1994 ApJ,437,879
[9] Nie wiadomo mi nic o ich wykorzystywaniu potem.
[10] Do większości zastosowań wystarcza `igrid=1`.
[11] ...lub w programie wywołującym procedurę `opacity(...)` przed jej wywołaniem.

są to zmienne `ntt` i `nr` oznaczające ilość kroków zmienności odpowiednio temperatury i gęstości przypadających na ten sam przedział.

- **skład**[12]**:** Współczynniki nieprzezroczystości zawarte są w 7 tabelach odpowiadających różnym X $(0.0, 0.1, 0.2, 0.35, 0.5, 0.7, 0.8)$. W przypadku, gdy generowane są również pochodne, liczba tabel potraja się. Każda z nich zawiera współczynniki odpowiadające różnym wartościom $\log T$ oraz $\log R$, gdzie $R = \rho/T_6^3$. Pierwsza z tych wielkości zmienia się następująco: $3, 3 + 0.05/\texttt{igrid}, \ldots, 6, 6 + 0.1/\texttt{igrid}, \ldots, 8.1, 8.1 + 0.2/\texttt{igrid}, \ldots, 8.7$, natomiast $\log R$: $-7, -7 + 0.5/\texttt{igrid}, \ldots, 1$. Wszystkie wartości graniczne i kroki pomiędzy nimi są zdefiniowane na początku procedury `opacity2` (około linii 1990. kodu) i tam też mogą być modyfikowane.

- **czas:** 5 – 10 minut (w zależności od komputera)

2.2 Wykorzystanie `opacity.dat` w programie

Zbiór `opacity.dat` zawiera wszystkie dane potrzebne do interpolacji. Procedury w Fortranie, napisane do jej przeprowadzenia, znajdują się w zbiorze opacity.04[13].
Oto przykład użycia danych i procedur interpolujących:

```
implicit double precision (a-h,o-z)
include 'data.04'
common /comkappa/ opr,opt,ideriv

ipul=1
ideriv=0
ntt=85
nr=17

t=...
rh=...
xchem=...
zchem=...

zed=zchem
call opacity(rh,t,xchem,fkap)
...
end

include 'opacity.04'
```

Współczynnik nieprzezroczystości(κ) dostępny jest przez zmienną `fkap`(nazwa może być zmieniona), natomiast pochodne: `opt` $= (d\kappa/dT)_\rho$ i `opr` $= (d\kappa/d\rho)_T$;

[12] Przypominam - zbiór binarny - ale może komuś to się przyda.
[13] Użycie go wymaga dołączenia zbioru ze stałymi `data.04` w kodzie programu.

nazwy tych zmienych pojawiają się w innych procedurach pakietu ewolucyjnego, a przypisania im wartości następują niejawnie — nie powinny być zmieniane.

Uwaga: Istnieje możliwość ustalenia zachowania programu w przypadku, gdy parametry wywołania nie mieszczą się w przedziale interpolacji, ale zasadniczo w takiej sytuacji następuje wyjście z procedur.

Dodatek 2.

Podobnie i dla tego pakietu istnieje wersja przystosowana do zastosowań pulsacyjnych. Pakiet funkcjonujący pod nazwą `opac` składa się z pliku `opac.f` (91KB), tych samych tablic `Alexande.dat`, `GN93hz.dat` oraz opac.s35 (14KB) i `data.s35` (10KB). Najistotniejsze różnice w porównaniu z `opacity` to:

- automatyczne generowanie tablic z pochodnymi współczynnika nieprzezroczystości
- tworzenie tablic w standardzie `ASCII` (nie – binarnie) w zbiorze `opac_env.dat` (181KB)
- jawne przekazywanie pochodnych nieprzezroczystości w programie, jak również obfitości pierwiastków ciężkich.

Poniżej zamieszczam przykład użytkowania już wygenerowanej tablicy:

```
implicit double precision (a-h,o-z)
include 'data.s35'

t=...
rh=...
xchem=...
zchem=...

call opacit(rh,t,xchem,zchem,fkap,opr,opt)
...
end

include 'opac.s35'
```

KONIEC.