KSN — III FK — zestaw 12

Minimalizacja

Poszukujemy (globalnego) minimum funkcji $f(\mathbf{x})$, gdzie $\mathbf{x}$ jest N-wymiarowym wektorem $\mathbf{x} = (x_1, x_2, \cdots, x_N)$. Zacznijmy od próby określenia minimum funkcji jednej zmiennej $f(x)$. Poszukiwanie $x_{\min}$ zbliżone jest do poszukiwania zer funkcji $f(x)$, z tym, że do określenia minimum potrzeba wartości funkcji w trzech punktach $a < b < c$, takich by $f(b) < f(a)$ oraz $f(b) < f(c)$. Wówczas minimum jest gdzieś wewnątrz przedziału (a, c). Analogiem bisekcji jest wskazanie punktu x spełniającego $a < x < b$ lub $b < x < c$. Wówczas jeśli $f(b) < f(x)$ trójkę $(x_1, x_2, x_3) = (a, b, c)$ zastępujemy trójką $(x_1, x_2, x_3) = (a, b, x)$, jeśli zaś $f(b) > f(x)$ — trójką $(x_1, x_2, x_3) = (b, x, c)$. Procedurę tą powtarzamy sukcesywnie, aż do uzyskania zadowalająco małej różnicy $x_3 - x_1$.

Poszczególne metody różnią się metodą wyboru kolejnych punktów b wewnątrz przedziału (a, c)...

W przypadku funkcji wielu zmiennych jednymi z najpopularniejszych technik numerycznych służących znajdowaniu minimów funkcji $f(\mathbf{x})$ są *metody gradientowe* wykorzystujące jednak minimalizację 1-D w odpowiednich kierunkach. Występujące w nazwie *sprzężenie* polega na tym, że poprawiając jedną ze składowych wektora $\mathbf{x}$ nie pogarszamy innych.

W bibliotece *Numerical Recipes* minimalizacji funkcji jednej zmiennej można dokonać

- metodą *złotego podziału* (`golden`),
- metodą *odwrotnej interpolacji parabolicznej* rozszerzonej przez *Brenta* (`brent`),
- czy dodatkowo uzupełnionej informacją o pochodnej (`dbrent`).

Użytkownik tej ostatniej

```
FUNCTION dbrent(ax,bx,cx,func,deriv,TOL,xmin)
...
END
```

minimalizującej funkcję $f(x)$ musi zatroszczyć się o przygotowanie funkcji

```
REAL FUNCTION func(x)
...
END
```

zwracającej wartość funkcji $f(x)$ oraz procedury

```
REAL FUNCTION deriv(x)
...
END
```

liczącej wartość pochodnej $f'(x)$. Funkcja `dbrent` zwraca wartość funkcji `func` w minimum `xmin`.

Procedura `mnbrak` wstępnie określa odpowiednie a, b, c jeśli mamy pewność, że minimum funkcji $f(x)$ znajduje się w przedziale (a, b). Przy korzystaniu z tej procedury funkcja licząca pochodną funkcji **musi** nazywać się `dfunc`.

Procedurę minimalizującą funkcję wielu zmiennych realizuje np. `frprmn`.

Dostępne w bibliotece *GSL* metody minimalizacji funkcji jednej zmiennej:

- złotego podziału (`gsl_min_fminimizer_goldensection`) oraz
- Brenta (`gsl_min_fminimizer_brent`)

obsługiwane są poprzez wspólną funkcję realizującą pojedynczy krok wybranej procedury:

```
int gsl_min_fminimizer_iterate (gsl_min_fminimizer * s)
```

przy czym obiekt `s` musi zostać wcześniej przygotowany funkcjami `gsl_min_fminimizer_alloc` oraz `gsl_min_fminimizer_set` (lub `gsl_min_fminimizer_set_with_values`).

Kolejne kroki minimalizacji należy powtarzać aż do uzyskania założonej dokładności (co łatwo sprawdzać funkcją `gsl_min_test_interval`) lub też do osiągnięcia maksymalnej założonej liczby iteracji.

Krzysztof Malarz & Maciej Wołoszyn, Kraków, 12 stycznia 2004