

演習課題

課題 1

サンプルプログラムをコンパイル・実行して動作を確認せよ。さらに、適宜修正してその実行結果を確認せよ。

課題 2

任意の 1 変数関数 $f(x)$ およびその微分値を返す関数 $f'(x)$ を引数として受け取り、 $f(x) = 0$ の近似解を Newton 法により求めるサブルーチンを実装し、その動作を確認せよ。ただし以下の条件を満たすこと。

- ▷ 初期値を引数として受け取る。
- ▷ 無限ループにならないように最大の反復回数を引数として受け取る。
- ▷ 許容誤差を引数として受け取り、反復による近似解の変化が許容誤差以下となったら収束したとみなす。
- ▷ 収束判定のステータスを返す。
- ▷ 求まった近似解を返す (収束しなかった場合は最後の反復後の解の近似値)。

(二分法のモジュール `bisection.f90` の `mod_bisection` を参考にすればよい。)

課題 3 †

`list.f90` で定義されているリストを実装したモジュール `mod_list` へ以下の様な機能追加を試みよ。以下では `a` はリスト (`type(list_type)`) のポインタであるとする。

- ▷ 代入演算子が配列を受け取れるようにせよ。即ち、

```
a = (/0, 1, 2/)
```

のようにリストに配列を代入すると元のリストを破棄し (正しく `deallocate` し)、与えられた配列でリストを初期化すること。

- ▷ サンプルでは `append` によって単一の値をリストの終端に追加しているが、配列を与えた場合にはその各要素をリストに追加するように拡張せよ。ただしオーバーロードを用いることで、どちらも同じサブルーチン名 `append` で行うようにせよ。

```
call append(a, 3)
call append(a, (/4, 5/))
```

のようなコードが実行可能となるはずである。

- ▷ 同様に `insert` でも配列を追加できるようにせよ。

```
call insert(a, 1, -1)
call insert(a, 1, (/ -3, -2 /))
```

のようなコードが実行可能となるはずである。