

Lab 11: Otimização sem restrição

INF1608 – Análise Numérica

Leonardo Seperuelo Duarte

lduarte@tecgraf.puc-rio.br

Departamento de Informática, PUC-Rio

1. O método de otimização sem restrição, conhecido como **Método da Interpolação Parabólica Sucessiva** (mips), parte de 3 estimativas iniciais r , s e t , ajusta uma parábola interpolante e estima que o mínimo da parábola, x , é um próximo candidato a ser o ponto de mínimo da função:

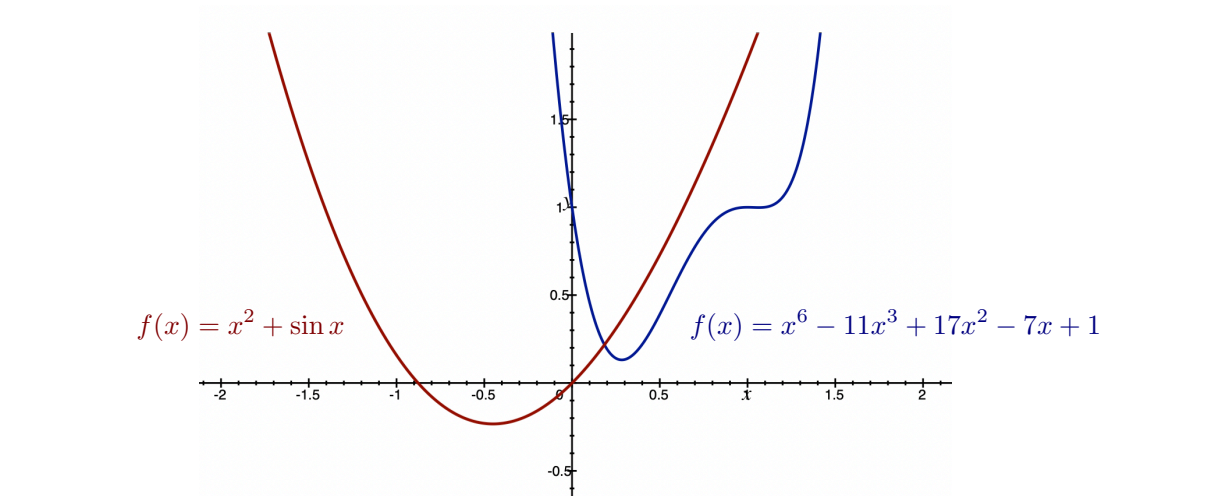
$$x = \frac{r + s}{2} - \frac{(f(s) - f(r))(t - r)(t - s)}{2[(s - r)(f(t) - f(s)) - (f(s) - f(r))(t - s)]}$$

Se o denominador da expressão acima for zero (na prática se $< 10^{-10}$), faz-se $x = (r + s + t)/3$. Essa estimativa substitui a menos recente, isto é, r passa a ter o valor de s , s de t e t de x . O método continua iterando até que o critério de convergência seja atendido ou que um número máximo de iterações seja alcançado. *Após 3 iterações*, pode-se adotar como critério de convergência a diferença entre o valor da função nas duas últimas estimativas: $|f(s) - f(t)| \leq \epsilon$, onde $\epsilon = 10^{-6}$ representa a tolerância requerida. O valor de x onde a função tem valor mínimo é então expresso pela média das estimativas: $x_{min} = \frac{s+t}{2}$.

Implemente uma função que codifica o Método da Interpolação Parabólica Sucessiva como descrito acima. Sua função deve receber como parâmetros a primeira estimativa inicial, r , e um valor de incremento, δ . As duas outras estimativas iniciais devem ser dadas por: $s = r - \delta$ e $t = r + \delta$. A função também recebe como parâmetros a *função objetivo*, $f(x)$, e um ponteiro x_{min} que deve ser preenchido com o valor de x onde a função tem valor mínimo dado pelo método. A função deve ainda retornar o número de iterações executadas para se alcançar o resultado, seguindo o protótipo abaixo. Se o método não convergir até 50 iterações, a função deve retornar zero. *A implementação deve minimizar o número de avaliações da função $f(x)$.*

```
int mips (double r, double delta, double (*f) (double x), double* xmin);
```

2. Para testar seu código, determine os valores mínimos das funções ilustradas abaixo. Compare o uso de diferentes estimativas e o número de iterações necessário para o método convergir.



Agrupe os protótipos das funções pedidas em um módulo “otimizacao.h” e as implementações em um módulo “otimizacao.c”. Escreva o teste em outro módulo “main.c”.

Entrega: O código fonte deste trabalho (isto é, os arquivos “otimizacao.c”, “otimizacao.h” e “main.c”) deve ser enviado via página da disciplina no EAD até o final da aula. O sistema receberá trabalhos com atraso (com perda de 1 ponto na avaliação) até 2 horas após o final da aula.