BINARY (N number F fraction S sign)Fixed Point: $((-1)^S)$(NNNN.FFFF), stored as SNNNNFFFF
Floating Point: $((-1)^S)(1 + .FFFF) * 2^{((NNNN - 1023)}$, stored as SNNNNFFFF
ERROR (p value, p' approx, $p = p' + e_1$, $q = q' + e_2$) Absolute: $|p - p'|$, Relative: $|p - p'| / |p|$, $p \neq 0$
$p + q = p' + q' + (e_1 + e_2)$, $pq = p' + q' + (q'e_1 + p'e_2 + e_1e_2)$
[F(x) = 0] Monotone: no sign change, Oscillating = changes signs each iteration
Bisection Method (start with A and B); C = (A + B)/2; If(A)f(C) < 0, C = A, Else C = B
Regula Falsi Method (start with A and B); C = B – [(f(B)(B – A))/(f(B) – F(A))], same checks/loop
Newton-Raphson Method (from 1st order Taylor Series, start with p_0): $p_k = p_{k-1} - (f(p_{k-1})/f'(p_{k-1}))$
$f'(x) = (f(x + E) - f(x))/E$ for some very small E; converges order 2 if simple root, 1 if multiple
Secant Method (start with p_0 and p_1), $p_{k+1} = p_k - [((f(p_k)(p_k - p_{k-1}))/(f(p_k) - f(p_{k-1}))]$
Horizontal convergence when $| f(x) | <$ eps, vertical when $| x_n - p | <$ del, f(p) = 0
Well-conditioned = clear/straight near zero, ill = multi-crossing, close to x-axis
[Ax = B] Upper triangular: only 0 below diagonal, Back substituting: solve last equation, substitute up
Can multiply by scalar, add multiples, move order of rows in a matrix; pivot = largest left in column
Gaussian Elimination: Make upper triangular, back substitute (use if only 1 system)
LU factorization (place next to identity matrix, perform same on both, substitute) (use for >1 system)
ITERATIVE [Ax=B] (use on big sparse matrices); Gauss-Seidel Method: Get x_k, y_k from x_{k-1} and y_{k-1}
Jacobi Method: Get x_k, y_k from x_k and y_k, (x_{k-1} and y_{k-1} when not possible)
Converges for strictly diagonal (| a[k][k] | > ∑[j=1;j≠k;j->n] | a[k][j] |, k goes from 1 to n)
POLYNOMIAL APPROX. Horner's Method: derivative of (∑[j=n;j->1] $(a_j x^{j+1})/(j+1)$) to approx. p(x)
Lagrange: have n points, p[n](x) = ∑[k=0;n] y_k*(π[j=0;j≠k;j->n]$(x-x_j)$)/(π[j=0;j≠k;j->n](x_k-x_j))
Newton Method (using "divided differences", f[g(x)]); $P_n(x) = P_{n-1}(x) + a_n$(π[i=0;n]$(x-x_i)$)
$f[x_k] = f(x_k)$, $f[x_{k-1},x_k] = (f[x_k] - f[x_{k-1}])/(x_k - x_{k-1})$, get a_i from $f[x_{i-1},x_i]$ or $f[x_1]$
Padé Approximation: $R_{n,m}(x) = P_n(x)/Q_m(x)$, $P_n(x) = p_0 + p_1x + \ldots p_nx^n$, $Q_m(x) = 1 + q_1x + \ldots q_mx^m$
$f(x)Q_m(x) - P_n(x) = 0$; $a_0-p_0 = 0$, $a_1+a_0q_1-p_1 = 0$, $a_2+a_1q_1+a_0q_2-p_2 = 0$, etc., get a_i from Taylor exp. of f(x)