LOCALIZING NUMERICAL INTEGRATION BY 2^p SUBDIVISIONS IN p DIMENSIONS

Stephen L. Adler
Institute for Advanced Study, Princeton

SEE: arXiv: 1009.4691 "Parametrized Adaptive Multidimensional Integration Routines (PAMIR): Localization by Repeated 2^p Subdivision"

- PROBLEM ADDRESSED
- SIMPLEXES AND HYPERCUBES
- HYPERCUBE SUBDIVISION
- HIGHER ORDER INTEGRATION FORMULAS FOR HYPERCUBES
- PROGRAMMING TO UTILIZE MEMORY AND SPEED
- SOME SAMPLE RESULTS

- PROBLEM ADDRESSED

$\int d^p x \; f(\vec{x})$ $f(\vec{x})$ HAS LOCAL PEAKS AND VALLEYS

WOULD LIKE AN INTEGRATION METHOD THAT CAN FOLLOW LOCAL VARIATIONS

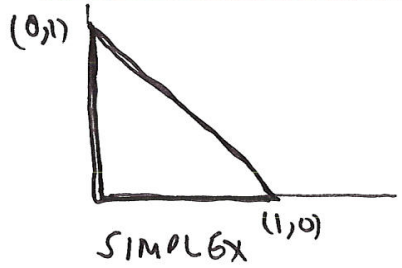
MOST CURRENT PROGRAMS RECURSIVELY SUBDIVIDE THE MOST RAPIDLY VARYING DIMENSION, RATHER THAN SIMULTANEOUSLY HALVING ALL DIMENSIONS (2^p SUBDIVISION)

COMPUTERS ARE NOW MUCH FASTER AND MEMORY MUCH CHEAPER: MEGA → GIGA

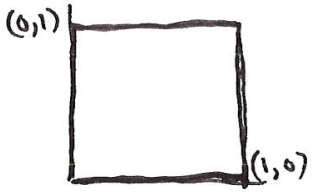
2^p SUBDIVISION NOW BECOMING FEASIBLE

- SIMPLEXES AND HYPERCUBES

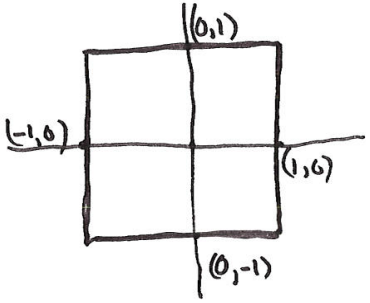
$p=2$



SIMPLEX



HYPERCUBE
SIDE 1



HALF-SIDE 1

• SIMPLEXES AND HYPERCUBES: GENERAL p

- $p+1$ POINTS IN p DIMENSIONS DEFINES A SIMPLEX

  VERTICES X_{ai} $\quad a = 0, \cdots, p \quad i = 1, \cdots, p$

  INTEGRATION FORMULAS THAT SUM OVER VERTICES:
  FUNCTION CALLS HAVE POLYNOMIAL GROWTH IN p

- HYPERCUBE HAS 2^p VERTICES, SO DON'T WANT ~~TO~~ USE VERTICES TO CONSTRUCT INTEGRATION FORMULAS

  AXIS PARALLEL HYPERCUBE SPECIFIED BY $p+1$ REAL NUMBERS
  HALF-SIDE LENGTH s, p COORDINATES OF CENTROID X_c

  $\tilde{X} = X - X_c$ CONSTRUCT INTEGRATION FORMULAS USING THE $2p$ CENTROIDS OF THE MAXIMAL BOUNDARY HYPERCUBES:

  $\tilde{X}_1 = (s, 0, 0 \cdots, 0)$ $\quad \tilde{X}_{p+1} = (-s, 0, 0, \cdots, 0)$

  $\cdots$ $\quad \cdots$

  $\tilde{X}_p = (0, 0, \cdots, s)$ $\quad \tilde{X}_{2p} = (0, 0, \cdots, -s)$

• HYPERCUBE SUBDIVISION

START FROM HYPERCUBE: CENTROID X_c, HALF-SIDE s, SIDES PARALLEL TO UNIT AXIS VECTORS

$\hat{u}_1 = (1, 0, 0, \cdots, 0)$

$\hat{u}_2 = (0, 1, 0, \cdots, 0)$

$\cdots$

$\hat{u}_p = (0, 0, 0, \cdots, 1)$

SUBDIVISION ALGORITHM: NEW HALF-SIDE $= s/2$

NEW CENTROIDS $X_{c;k}$ $k = 0, 1, \cdots, 2^p - 1$

SCAN ALONG BINARY REPRESENTATION OF k

CONTAINS p DIGITS $1 \le j \le p$ EXTRACT WITH IBITS

IF j^{th} DIGIT IS 0 ADD $\frac{1}{2} s \hat{u}_j$ TO X_c

IF j^{th} DIGIT IS 1 ADD $-\frac{1}{2} s \hat{u}_j$ TO X_c

SIMPLEX SUBDIVISION: ANALOGOUS METHOD IN COMPUTER GRAPHICS LITERATURE (D. MOORE)

• HIGHER ORDER INTEGRATION FORMULAS FOR HYPERCUBES

FOR SMOOTH INTEGRANDS, ADVANTAGES IN USING HIGHER ORDER INTEGRATION

MOMENT INTEGRALS

$$m(\nu) = \int_{-s}^{s} dx_1 \cdots \int_{-s}^{s} dx_p \; \tilde{x}_1^{\nu_1} \cdots \tilde{x}_p^{\nu_p} = 0 \quad \text{ANY } \nu_\ell \text{ ODD}$$

$$= V \prod_{\ell=1}^{p} \frac{s^{\nu_\ell+1}}{\nu_\ell+1} \quad \text{ALL } \nu_\ell \text{ EVEN}$$

$V = (2s)^p$ = VOLUME

DEFINE $$\delta_{\lambda_1 \cdots \lambda_n} = \sum_{\gamma=1}^{2p} \frac{1}{\ell_\gamma!} \tilde{x}_{j\lambda_1} \cdots \tilde{x}_{j\lambda_n}$$

FIND $$\frac{1}{V} \int_{\text{hypercube}} dx_1 \cdots dx_p \; \tilde{x}_{\lambda_1} \tilde{x}_{\lambda_2} = \frac{1}{6} \delta_{\lambda_1 \lambda_2}$$

$$\frac{1}{V} \int_{\text{hypercube}} dx_1 \cdots dx_p \; \tilde{x}_{\lambda_1} \tilde{x}_{\lambda_2} \tilde{x}_{\lambda_3} \tilde{x}_{\lambda_4} = \frac{1}{36} \left(\delta_{\lambda_1\lambda_2}\delta_{\lambda_3\lambda_4} + \delta_{\lambda_1\lambda_3}\delta_{\lambda_2\lambda_4} + \delta_{\lambda_1\lambda_4}\delta_{\lambda_2\lambda_3} \right) - \frac{1}{15} \delta_{\lambda_1\lambda_2\lambda_3\lambda_4}$$

MORE COMPLICATED FOR 6^{th}, 8^{th} ORDER

THROUGH q^{th} ORDER

$$f(\tilde{x}) = A + B_{\lambda_1}\tilde{x}_{\lambda_1} + C_{\lambda_1\lambda_2}\tilde{x}_{\lambda_1}\tilde{x}_{\lambda_2} + D_{\lambda_1\lambda_2\lambda_3}\tilde{x}_{\lambda_1}\tilde{x}_{\lambda_2}\tilde{x}_{\lambda_3} + E_{\lambda_1\lambda_2\lambda_3\lambda_4}\tilde{x}_{\lambda_1}\cdots\tilde{x}_{\lambda_4}$$

$$+ \cdots + J_{\lambda_1\cdots\lambda_q}\tilde{x}_{\lambda_1}\cdots\tilde{x}_{\lambda_q} + \cdots$$

DEFINE $\quad C_{\lambda_1\lambda_2}\delta_{\lambda_1\lambda_2} \equiv C_2$

$$E_{\lambda_1\cdots\lambda_4}\left(\delta_{\lambda_1\lambda_2}\delta_{\lambda_3\lambda_4} + \delta_{\lambda_1\lambda_3}\delta_{\lambda_2\lambda_4} + \delta_{\lambda_1\lambda_4}\delta_{\lambda_2\lambda_3}\right) \equiv G_{2+2}$$

$$E_{\lambda_1\cdots\lambda_4}\delta_{\lambda_1\lambda_2\lambda_3\lambda_4} \equiv E_4$$

ETC (i.e. LABEL BY PARTITIONS OF n)

WE FIND, THROUGH q^{th} ORDER, HYPERCUBE INTEGRATION FORMULA

$$\frac{1}{V}\int_{\text{hypercube}} dx_1\cdots dx_n\, f(\tilde{x}) = A + \frac{1}{6}C_2 + \frac{1}{36}G_{2+2} - \frac{1}{15}E_4$$

$$+ \frac{1}{216}G_{2+2+2} - \frac{1}{90}G_{4+2} + \frac{8}{63}G_6$$

$$+ \frac{1}{1296}I_{2+2+2+2} + \frac{1}{225}I_{4+4} - \frac{1}{540}I_{4+2+2} + \frac{4}{189}I_{6+2} - \frac{8}{15}I_8$$

$$+ \cdots$$

TO GET NUMERICAL QUADRATURE FORMULAS:

USE THE DISCRETE SUMS

$$\Pi_1(\lambda) = \sum_a f(\lambda \tilde{x}_a) \qquad 0 \le \lambda \le 1$$

$$\Pi_2(\lambda,\sigma) = \sum_{a,b}{}' f(\lambda \tilde{x}_a + \sigma \tilde{x}_b) \qquad 0 \le \lambda,\sigma, \quad \lambda+\sigma \le 1$$

$$\Pi_3(\lambda,\sigma,\mu) = \sum_{a,b,c}{}' f(\lambda \tilde{x}_a + \sigma \tilde{x}_b + \mu \tilde{x}_c) \qquad 0 \le \lambda,\sigma,\mu, \quad \lambda+\sigma+\mu \le 1$$

$$\Pi_4(\lambda,\sigma,\mu,\kappa) = \sum_{a,b,c,d}{}' f(\lambda \tilde{x}_a + \sigma \tilde{x}_b + \mu \tilde{x}_c + \kappa \tilde{x}_d) \qquad 0 \le \lambda,\sigma,\mu,\kappa, \quad \lambda+\sigma+\mu+\kappa \le 1$$

THESE CAN BE USED TO EXTRACT N, C_2, E_{2+2}, ... NEEDED FOR INTEGRATION UP TO q^{th} ORDER

MATCHING EQUATIONS ALWAYS TAKE FORM OF A VANDERMONDE SYSTEM: $\sum_{i=1}^{N} x_i^{k-1} w_i = q_k \qquad k = 1, \ldots N$

SOLVABLE IN CLOSED FORM

- PROGRAMMING TO UTILIZE MEMORY AND SPEED

THREE SETS OF PROGRAMS:

ONE STAGE: SUBDIVIDE (WITH THINNING)  ETC

UNTIL MEMORY LIMITED

RECIRCULATING OR TWO STAGE: AT END OF FIRST STAGE, SEQUENTIALLY APPLY THE ONE STAGE ALGORITHM TO EACH REMAINING SUBREGION  , THEN ETC

SPEED LIMITED

MPI PARALLEL: AT SECOND STAGE, SEND GROUPS OF REMAINING SUBREGIONS TO PROCESSES IN CLUSTER

SPEED GAIN $\propto N_{processes} - 1$

• SOME SAMPLE RESULTS

DOUBLE GAUSSIAN : TWO GAUSSIANS DISPLACED ALONG DIAGONAL OF HYPERCUBE

MACBOOK PRO

$p=5$	5^{th} ORDER	5 LEVELS THINNING AFTER LEVEL 2	0.9999986 7 PLACE ACCURACY	0.63×10^{8} FUNCTION CALLS 29 SECONDS
$p=5$	7^{th} ORDER	6 LEVELS THINNING AFTER LEVEL 2	0.999993926 9 PLACE ACCURACY	0.39×10^{10} FUNCTION CALLS 1700 SECONDS

64 PROCESS CLUSTER

$p=7$	7^{th} ORDER	5 LEVELS NO THINNING	0.9999915 0 8 PLACE ACCURACY	0.27×10^{12} FUNCTION CALLS 760 SECONDS
$p=7$	7^{th} ORDER	6 LEVELS NO THINNING	0.9999915003 10 PLACE ACCURACY	0.35×10^{14} FUNCTION CALLS 98,000 SECONDS

HAVE GONE UP TO $p=9$ ON CLUSTER 5 LEVELS