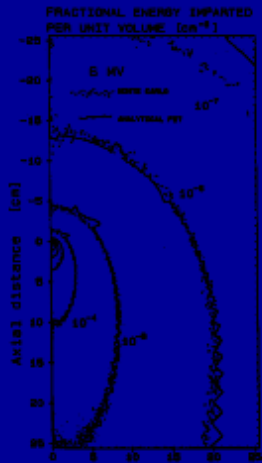


Solución del problema inverso de la radioterapia y su implementación en arquitecturas con SIMD.



$$D(\mathbf{r}) = \int \int \int \int T_E(s) h(E, \mathbf{r} - \mathbf{s}) d^3s dE.$$

```
#ifndef PENTIUM_ASM
// (C) NEVIS - Sao Paulo - Brasil.
// this is the ANSI C implementation, for maximum portability.

#define FAST_RND(x) ((int)((x) + 0.5))

float GetDensity3DFast(float x, float y, float z)
{
    long    index;

    if (x < xmin || x > xmax ||
        y < ymin || y > ymax ||
        z < zmin || z > zmax )
        return 0;

    index = FAST_RND((x - xmin) * facx); //x
    index += FAST_RND((y - ymin) * facy) * nx; //y
    index += FAST_RND((z - zmin) * facz) * plane_size; //z

    return (float)Den3D[ index ];
}

```

armando alaminos bouza

físico - medico. M.Sc.

- Serv. de Radiocirurgia CEPRO, São Paulo.

- MEVIS Informática Médica Ltda. Brasil.

```
#pragma aux sse2_mul_addf4 =
" SMA00:
"   movss    xmm1, dword ptr [eax]"
"   prefetchnta    byte ptr [eax+ebx]"
"   movss    xmm2, dword ptr [edx]"
"   mulss    xmm1, xmm0  "
"   add      eax, ebx    "
"   addss    xmm1, xmm2  "
"   movss    qword ptr [edx], xmm1"
"   add      edx, esi    "
"
"   movss    xmm3, dword ptr [eax]"
"   prefetchnta    byte ptr [eax+ebx]"
"   movss    xmm2, dword ptr [edx]"
"   mulss    xmm3, xmm0  "
"   add      eax, ebx    "
"   addss    xmm3, xmm2  "
"   movss    qword ptr [edx], xmm3"
"   add      edx, esi    "
"
"   movss    xmm1, dword ptr [eax]"
"   prefetchnta    byte ptr [eax+ebx]"
"   movss    xmm2, dword ptr [edx]"
"   mulss    xmm1, xmm0  "
"   add      eax, ebx    "
"   addss    xmm1, xmm2  "
"   movss    qword ptr [edx], xmm1"
"   add      edx, esi    "
"
"   movss    xmm3, dword ptr [eax]"
"   prefetchnta    byte ptr [eax+ebx]"
"   movss    xmm2, dword ptr [edx]"
"   mulss    xmm3, xmm0  "
"   add      eax, ebx    "
"   addss    xmm3, xmm2  "
"   movss    qword ptr [edx], xmm3"
"   add      edx, esi    "
"
"   sub      ecx, 4      "
"   ja      short SMA00 "
parm    [eax][ebx][edx][esi][ecx]
modify  [eax edx ecx];

```

Algoritmos presentes en los programas para planificación inversa con modulación de intensidad (IMRT).

- La planificación inversa se implementa como un proceso de optimización de los parámetros para cada haz de radiación. Cada conjunto de parámetros define un posible tratamiento para ser evaluado y comparado con los otros posibles.

- La función que mide la calidad del tratamiento se denomina "función objetivo". La función objetivo expresa numericamente la distancia entre la distribución de dosis "deseada" y la alcanzada por el plan que está siendo probado.

- En IMRT los haces de radiación tienen un perfil de fluencia complejo, generalmente dividido en pequeños sub-haces ("beamlet"), cuya distribución de intensidades debe ser investigada hasta conseguir el mejor resultado de la función objetivo.

- Un problema típico de planificación inversa (9 campos de 100 cm² cada) debe optimizar 900 intensidades. Dicho de otra forma: buscar un extremo de la función objetivo en un espacio de 900 dimensiones.

- Métodos de optimización empleados:

mínimos cuadrados iterativos.

Powell, Downhill-simplex, Simulated annealing.

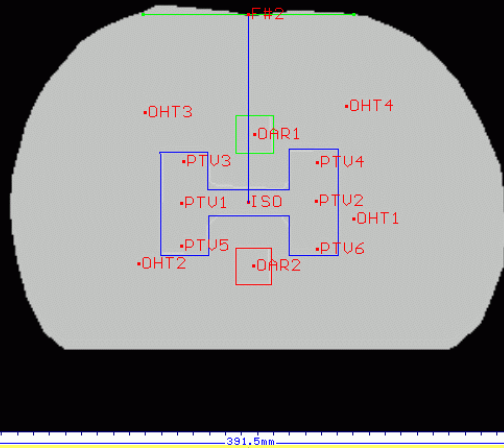
Davidon-Fletcher-Powell (en general, con evaluación de derivadas).

programación lineal.

entre otros, todos con virtudes y fragilidades.

El problema inverso de la RT no tiene solución analítica y debe ser formulado como una optimización, computacionalmente costosa.

Formulación matricial del modelo de distribución de dosis para un haz de fotones.



Plano axial con secciones del PTV y OARs. Algunos puntos de control como ilustración.

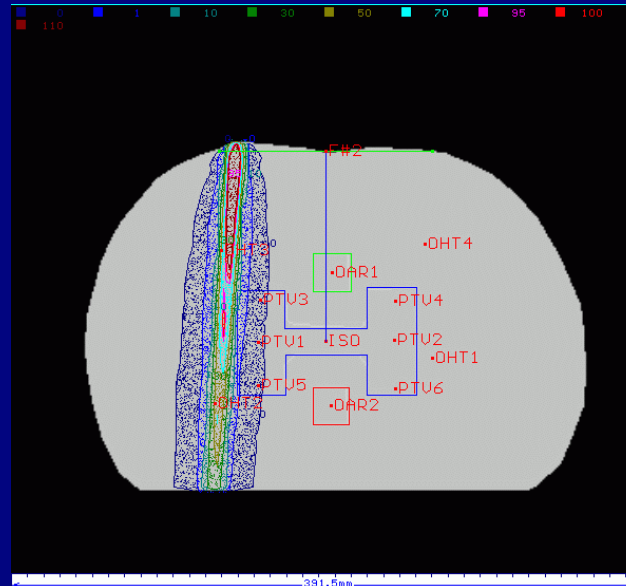
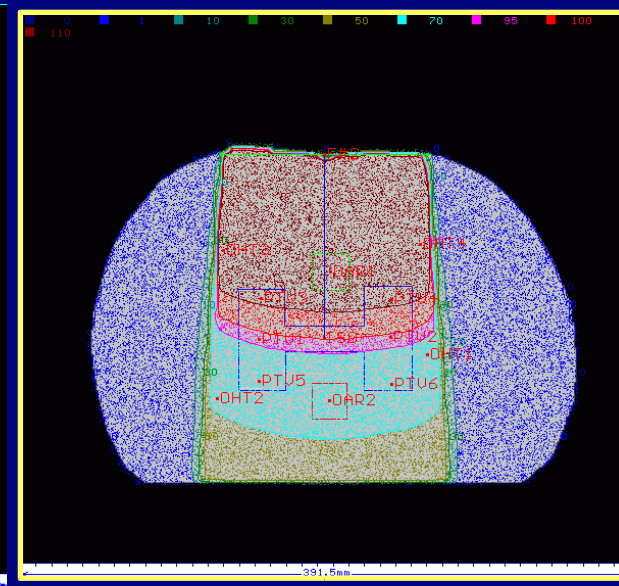


Ilustración de la contribución de dosis de una serie de "beamlets" sobre los puntos de control. Todos los beamlets con igual peso (no modulados)



Integración de contribuciones de todos los beamlets. Todos los pesos iguales, es decir campo sin modulación de intensidad.

El cálculo del aporte de dosis por parte de un beamlet a un punto de control puede realizarse con diversos algoritmos: integral de Clarkson, Monte Carlo o Pencil Beam de sección finita. Los resultados que mostraremos utilizan el modelo de Pencil Beam divergente y finito.

Ejemplo de una formulación del problema de la planificación inversa en IMRT

(*) :

- el volumen 3D es dividido en N voxels.
- J es el número de haces empleados.
- M es el número de beamlets por cada campo.

La dosis calculada en el voxel n del volumen tratado es ($D_c(n)$):

$$D_c(n) = \sum_{j=1}^J \sum_{m=1}^M w_{jm}(n) I_{jm},$$

donde: I_{jm} es la fluencia relativa de energía del beamlet m del haz j . w_{jm} es la contribución de dosis del beamlet m del haz j al voxel n suponiendo que el beamlet m tenga fluencia relativa unitaria (1).

Sea la distribución de dosis prescrita $D_0(n)$. Matemáticamente, la proximidad entre la prescripción y el cálculo para un conjunto de intensidades de fluencia de energía es:

$$F = \frac{1}{N} \sum_{n=1}^N r_{\sigma} [D_c(n) - D_0(n)]^2,$$

donde r_{σ} es un factor que pondera la importancia de cumplir la condición para los voxels de la estructura σ . F es la función objetivo, que en esta formulación es cuadrática.

(*) - Xing L., et.al. Med.Phys 25(10), October 1998 .

Crooks S.M., et.al. Int.J.Radiat.Oncol.Biol.Phys. Vol. 54 No.4, 2002.

Función objetivo articulada.

$$F = r_{\alpha} E_{PTV} + r_{\beta} E_{OAR} + r_{\mu} E_{OHT}$$

donde :

$$E_{PTV} = 1/N \sum (Dc_i - Dp_i)^2_i$$

$$E_{OAR} = 1/N \sum [(Dc_i > Dp_i) ? (Dc_i - Dp_i)^2 : 0.0]$$

Restricciones (*constrains*) :

Los valores de fluencia relativa son no negativos. En la práctica las fluencias están acotadas, pues la modulación de intensidad enfrenta limitaciones físicas.



Cada punto del dominio que el método de optimización investiga genera una evaluación de la función objetivo y un chequeo de las restricciones.

Cual es el costo de evaluar la distribución de dosis para $\langle m \rangle$ puntos de control en un plan con $\langle n \rangle$ beamlets ?

$$\begin{array}{c}
 \begin{array}{|c|} \hline d_1 \\ \hline d_2 \\ \hline d_3 \\ \hline \cdot \\ \hline \cdot \\ \hline \cdot \\ \hline d_m \\ \hline \end{array}
 =
 \begin{array}{cccccccc}
 w_{11} & w_{12} & w_{13} & \cdot & \cdot & \cdot & \cdot & w_{1n} \\
 w_{21} & w_{22} & w_{23} & \cdot & \cdot & \cdot & \cdot & w_{2n} \\
 w_{31} & w_{32} & w_{33} & \cdot & \cdot & \cdot & \cdot & w_{3n} \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 w_{m1} & w_{m2} & w_{m3} & \cdot & \cdot & \cdot & \cdot & w_{mn}
 \end{array}
 \cdot
 \begin{array}{|c|} \hline I_1 \\ \hline I_2 \\ \hline I_3 \\ \hline \cdot \\ \hline \cdot \\ \hline \cdot \\ \hline I_n \\ \hline \end{array}
 \end{array}$$

del modelo matricial anterior se deduce que son necesarias (como mínimo) :

| | | |
|------------------|---|---------------|
| multiplicaciones | : | $m * n$ |
| adiciones | : | $m * (n - 1)$ |
| memory read | : | $(m * n) + n$ |
| memory write | : | m |

Para el caso 3, con 491 beamlets y 11500 puntos de control: 5.646.500 multiplicaciones y 5.635.000 adiciones por evaluación. Solo 5 campos !!

Por que investigamos en algoritmos e implementaciones para minimizar el tiempo de cálculo ?

- La planificación de IMRT aún es un proceso de “prueba y error” en que varios parámetros deben ser ajustados hasta alcanzar un plan clinicamente satisfactorio. Sin un programa rápido no se consigue un ambiente interactivo para planificación inversa (Crooks, Xing; I.J. Radiat. Oncol. Biol Phys. 2002).

- Otros parámetros pueden investigarse automáticamente si la optimización para modulación de intensidad es suficientemente rápida. Por ejemplo la selección del número y angulación de los campos (Gaede, Wong, Rasmussen; “An algorithm for systematic selection of beam directions for IMRT”, Med. Phys. 31 (2) 2004).

Qué recursos nos brindan las nuevas CPUs con arquitectura SIMD ?

SIMD = Single-Instruction Multiple-Data

- Incorporaron nuevo repertorio de instrucciones que pueden operar sobre datos escalares (lo tradicional) o vectoriales (packed). También llamados de procesadores vectoriales, constituyen una evolución del “array processor”. La idea del procesamiento vectorial nació en los años 60 en el proyecto Solomon de Westinghouse. La tecnología vectorial fue mejorando hasta ser adoptada en la Cray-1 (1976), hasta hoy es el paradigma de las supercomputadoras.

- Los procesadores más comunes en las PCs denominan sus repertorios de instrucciones para procesamiento vectorial como :

INTEL : MMX, SSE (P3), SSE2 (P4) , SSE3 (P4 + HT)

AMD : 3DNow!

- Permiten realizar 4 operaciones aritméticas paralelas en flotantes de simple precisión (32bits) o 2 de doble precisión (64bits), en el mismo tiempo que realizan una escalar.

Ejemplo A:

```
addps xmm1, xmm0
```

puede equivaler a:

$$x = x + u$$

$$y = y + v$$

$$z = z + w$$

$$r = r + p$$

Ejemplo B:

```
mulps xmm1, xmm0
```

puede equivaler a:

$$x = x * u$$

$$y = y * v$$

$$z = z * w$$

$$r = r * p$$

Obsérvese que nuestro modelo matricial para calcular la dosis en el punto de control k se reduce a :

$$d_k = w_{k1} * I_1 + w_{k2} * I_2 + w_{k3} * I_3 + w_{k4} * i_4 + \dots w_{kn} * I_n$$

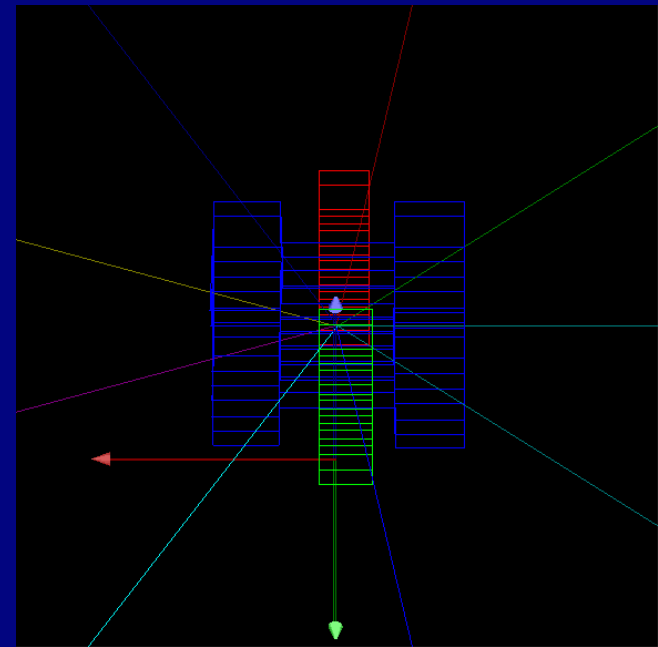
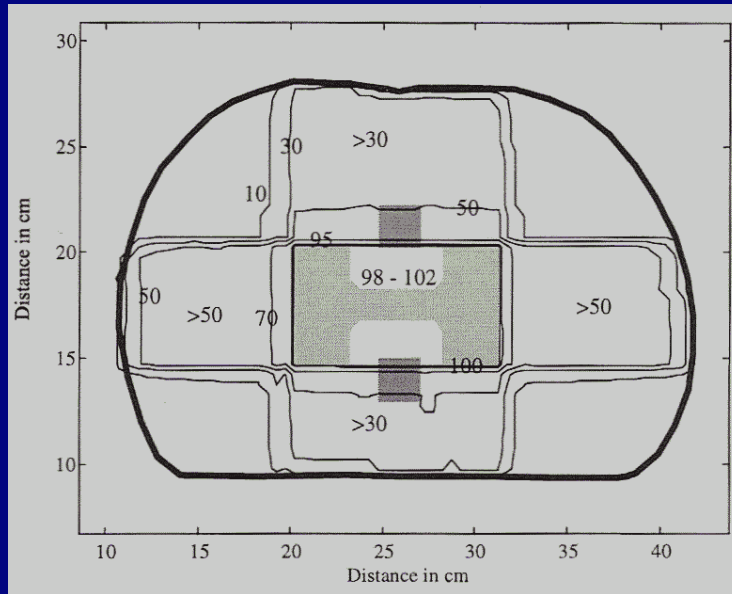
Lo que configura una formulación ideal para instrucciones del tipo SIMD, cuya implementación en SSE podría ser:

```
movaps  xmm0, Zero_Val ; limpiar acumulador
movaps  xmm1, w_k1      ; cargar primeros 4 valores [w_k1.. w_k4]
mulps   xmm1, I_1       ; multiplicar con los primeros 4 I [I_1 .. I_4]
addps   xmm0, xmm1     ; sumar al acumulador los primeros 4 resultados
movaps  xmm1, w_k5     ; cargar próximos 4 valores [w_k5.. w_k8]
mulps   xmm1, I_5      ; multiplicar con los próximos 4 I [I_5 .. I_8]
addps   xmm0, xmm1     ; sumar al acumulador los proximos 4 resultados
...
```

Lo que genera 4 veces menos operaciones que la implementación con instrucciones escalares.
(ver: “**Emmerald: a fast matrix-matrix multiply using Intel’s SSE instructions**”,
in: **Concurrency and Computation: Practice and Experience. 2001; 13:103-119**)

Problema: Casi todos los compiladores de la familia PC son para procesadores escalares. Para las CPUs de Intel o AMD sólo existen en el momento tres compiladores vectoriales: Intel C/C++, Intel Fortran y el VectorC86 de CodePlay. Los compiladores de Intel ignoran los recursos de AMD y el VectorC86 aún no tiene soporte completo a lenguaje ni bibliotecas. El único camino para explotar todos los recursos, hasta el momento, es el Ensamblador (2005).

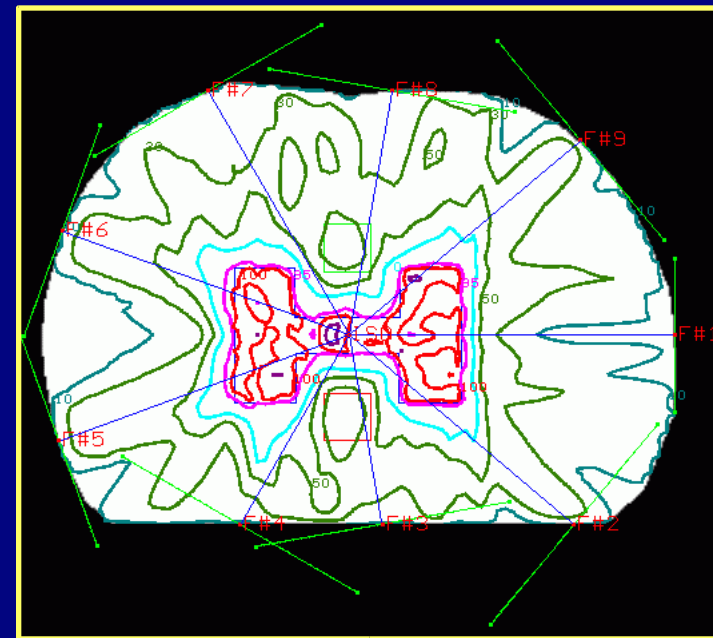
Caso 1: Solución de un problema "benchmark".



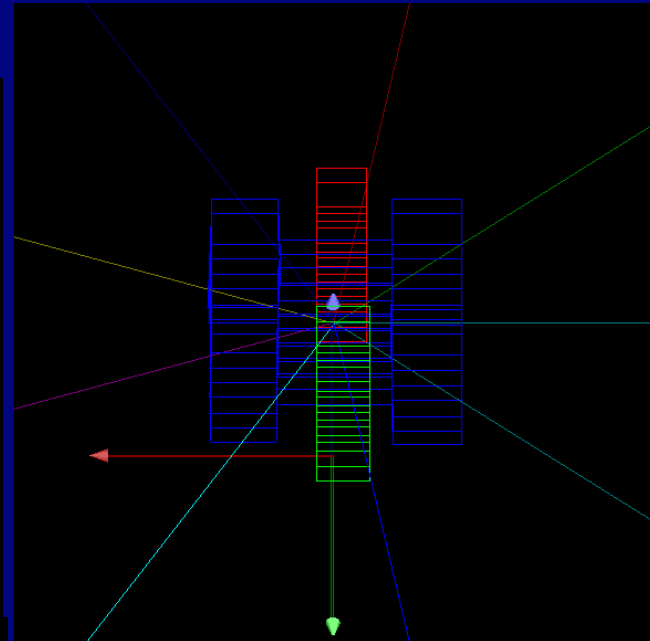
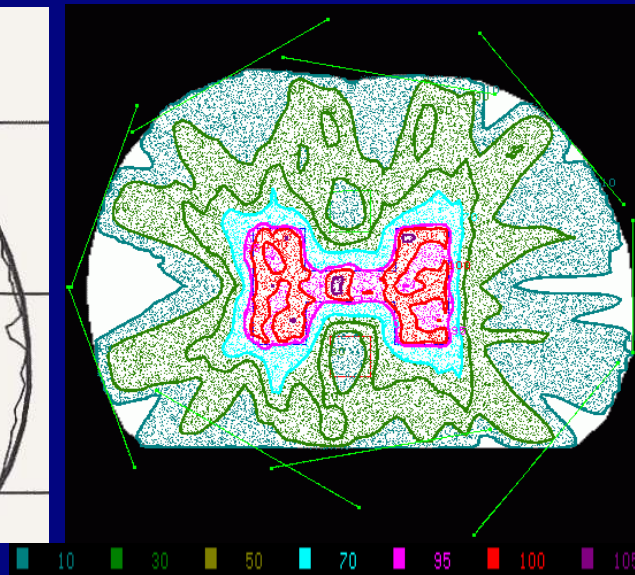
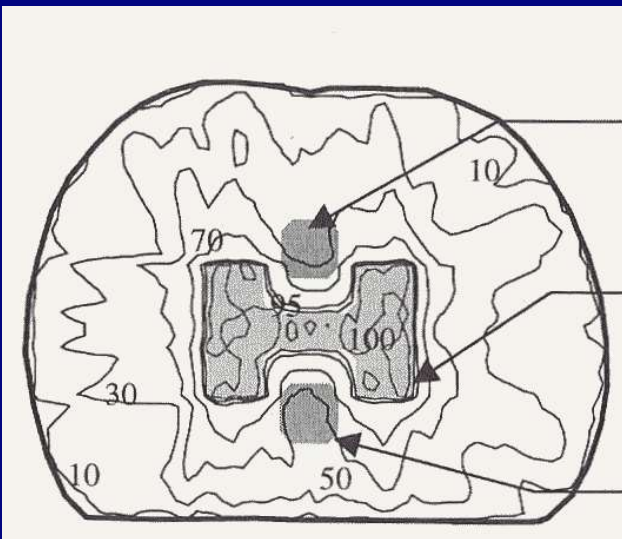
Caso con PTV en forma de una letra "H", lo que configura un volumen cóncavo. OARs anterior y posterior, invadiendo parcialmente las concavidades.

9 campos. Genera 1824 beamlets efectivos, con 16004 puntos de control. Sólo la matrix de w_{ik} ocupa 116 MegaBytes ! El tiempo de acceso a memoria no es despreciable. Optimización máxima en ~10 minutos en P4 HT a 3.0 GHz, 512 MB RAM.

Tomado de: Haas Olivier, "Radiotherapy Treatment Planning, New System Approaches", Springer-Verlag, 1999. Página 111.)



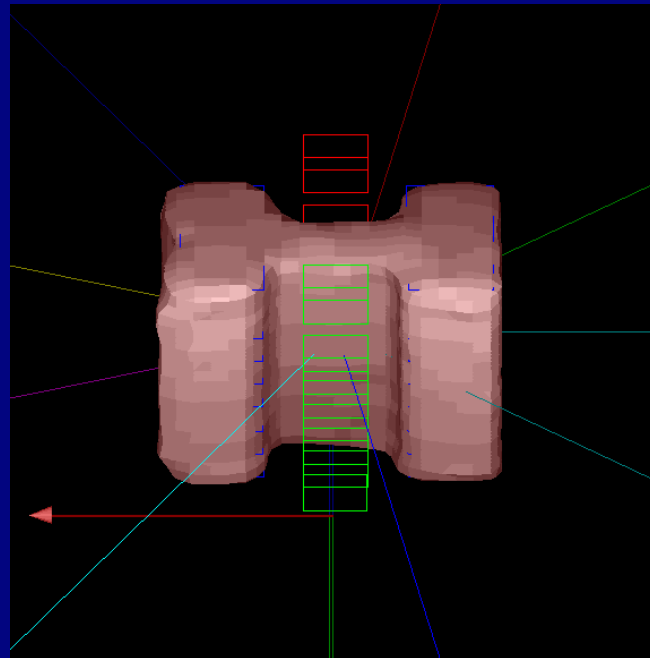
Caso 1: Solución de un problema "benchmark".

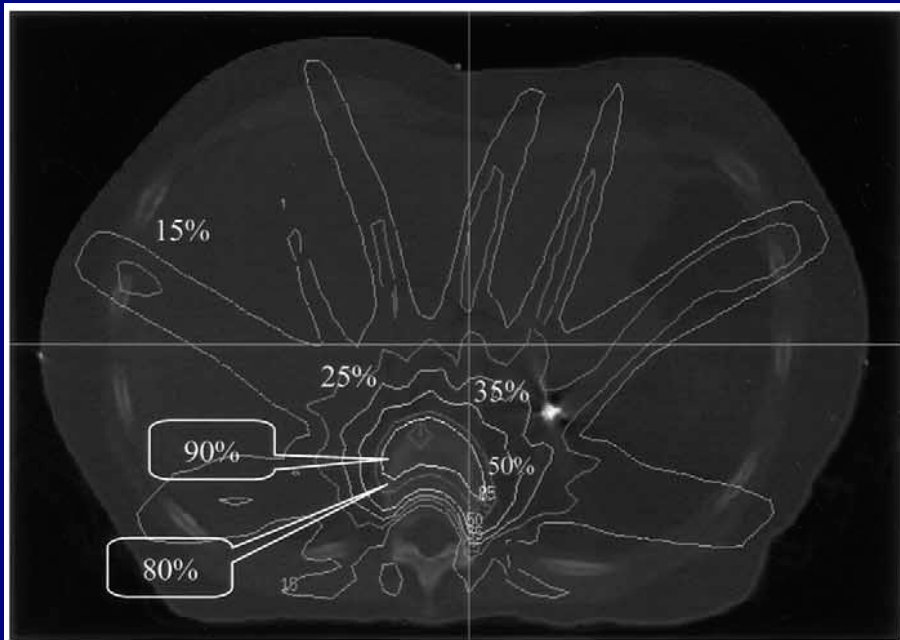


Solución de Haas (*) en blanco y negro. Solución del CAT3D en figuras coloridas.

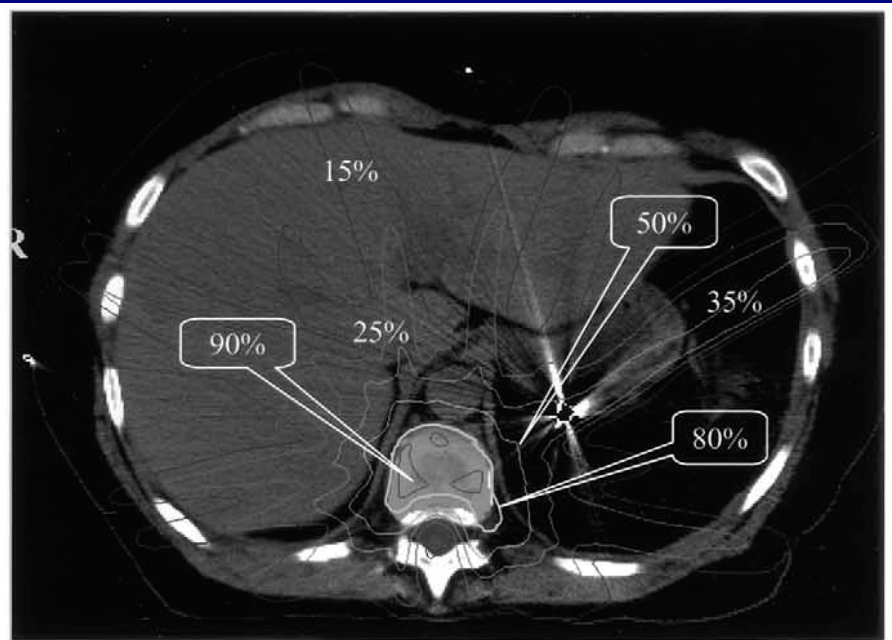
NOTA: Puede demostrarse que, en la ausencia de modulación de intensidad es posible producir planes conformados que se ajusten a PTVs convexos, utilizando suficientes haces. Para el caso de PTVs con superficies concavas, sólo puede conseguirse buena conformación de la dosis empleando modulación de intensidad.

(*) Haas Olivier, "Radiotherapy Treatment Planning, New System Approaches", Springer-Verlag, 1999. Página 25.





(a)



(b)

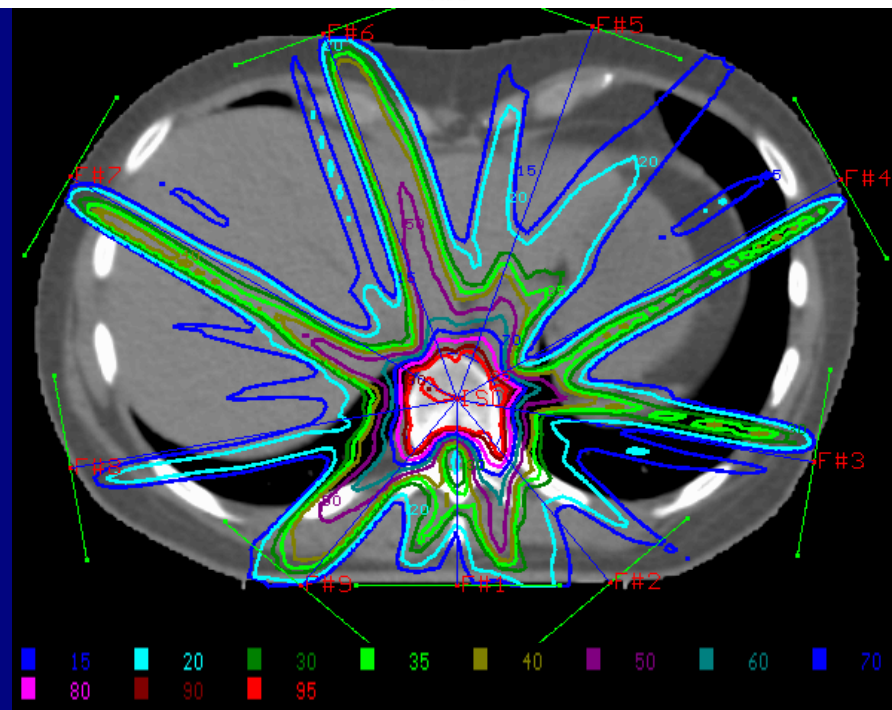
Caso 2: Tratar cuerpo vertebral, protegiendo médula.

Fig a. - método CLS, de Crooks and Xing. SGI O2 R5000, $t \sim 1$ min, beamlets 10x10mm (*)

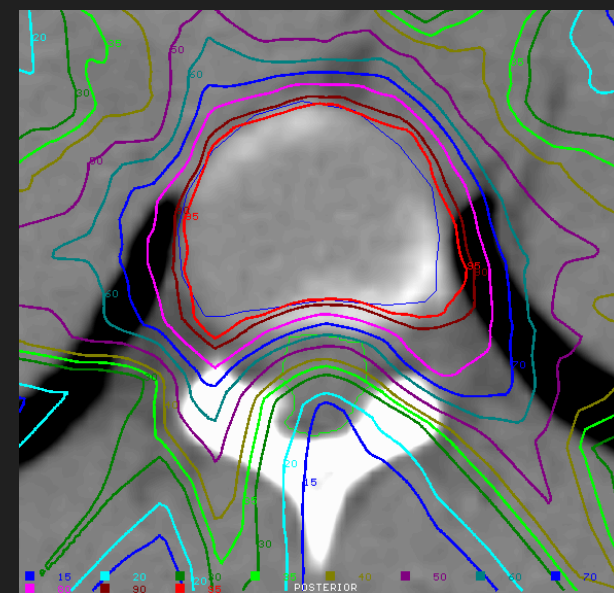
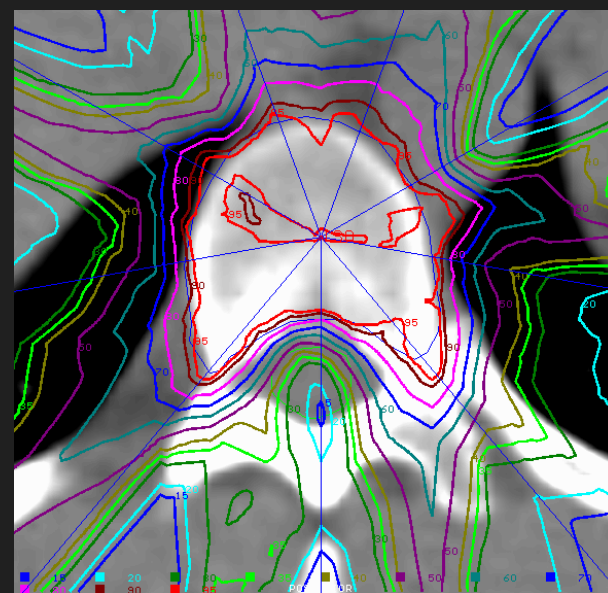
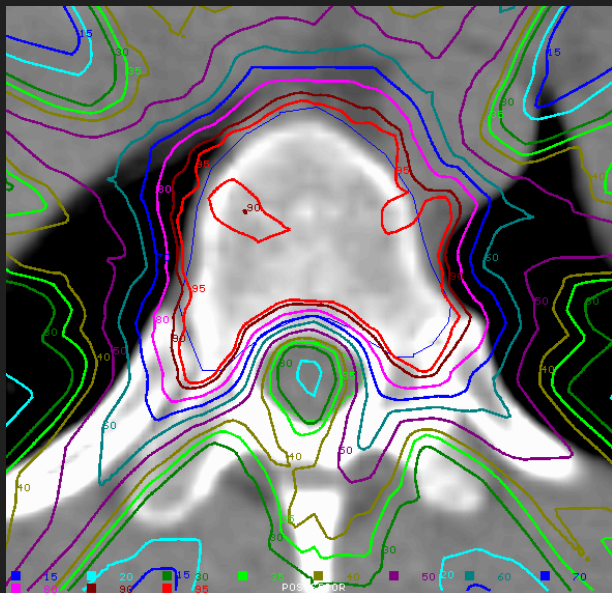
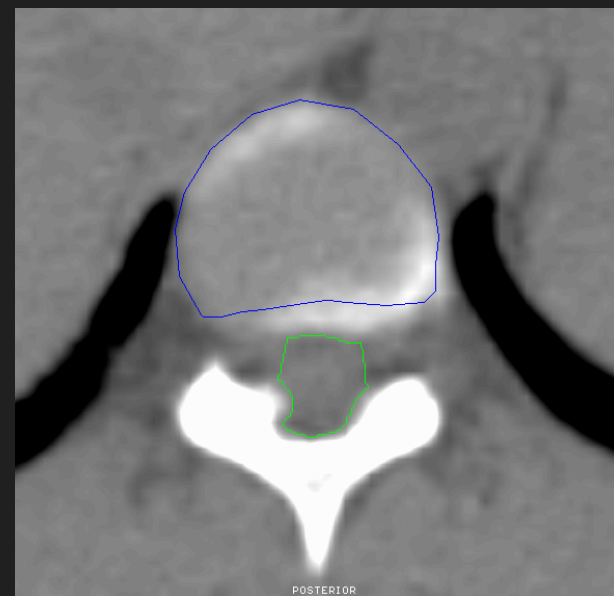
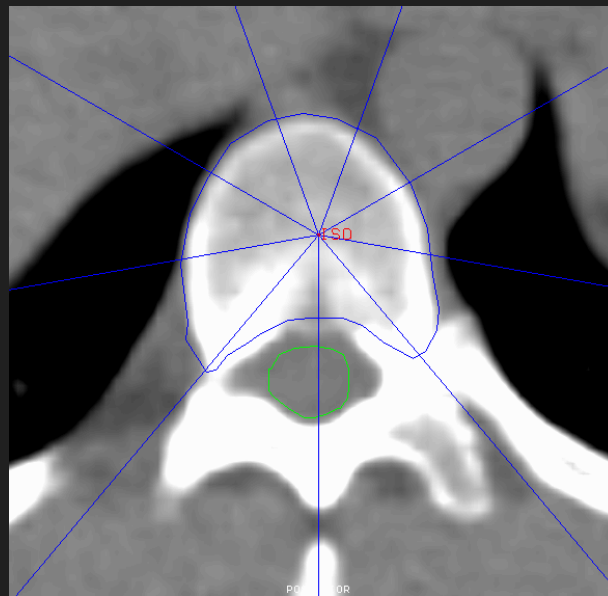
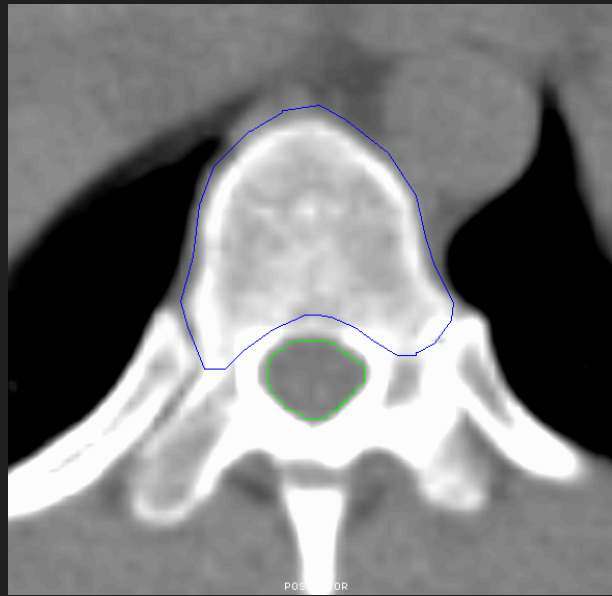
Fig b. - CORVUS (Nomos Corporation), motherboard con 4 pentiums pro en paralelo, 400MHz cada. $t \sim 30$ min. (*)

Figura a derecha : CAT3D, Pentium HT 3.0GHz. 512 MB, $t \sim 20$ s. Beamlets de 8x8mm

(*) Application of constrained least-square technique to IMRT treatment planning. Int. J. Radiation Oncol. Biol. Phys, vol. 54, no.4, 2002. Crooks and Xing, Dept. Radiat. Oncol., Stanford Univ. School of Medicine, CA.



Caso 2: Detalles de tres planos diferentes de la solución con 9 campos. 295 beamlets de 8x8mm. Considera radiación dispersa. PTV = 100, W=1.0; Médula < 30, W=0.6; OHT < 43, W=0.5



Caso 2: DVHs , representación de una posible solución en 3D

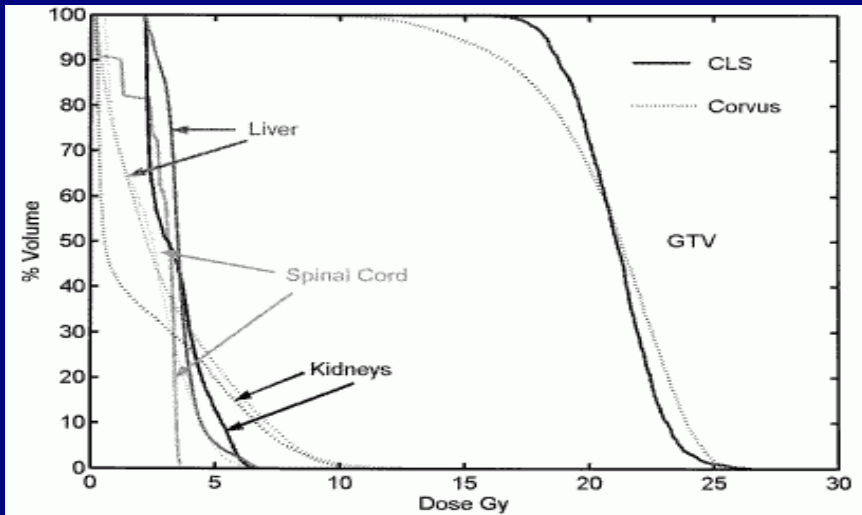
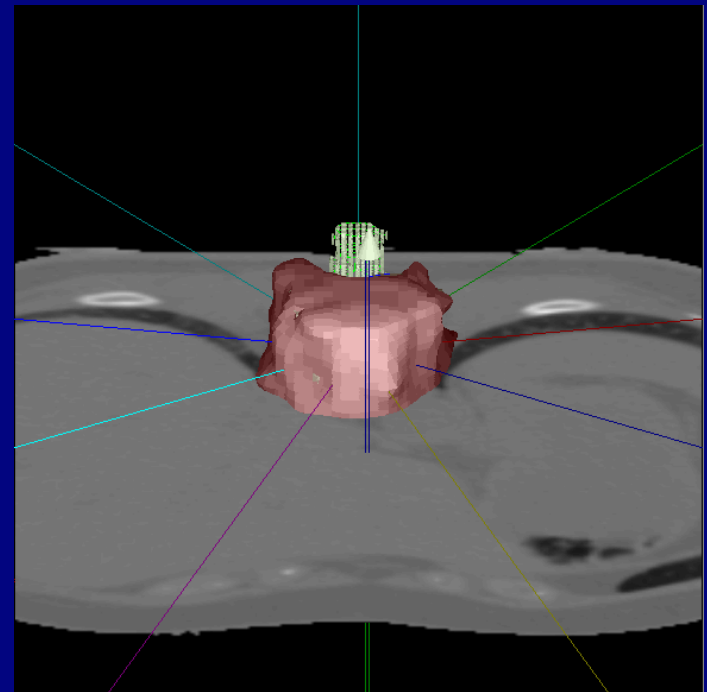
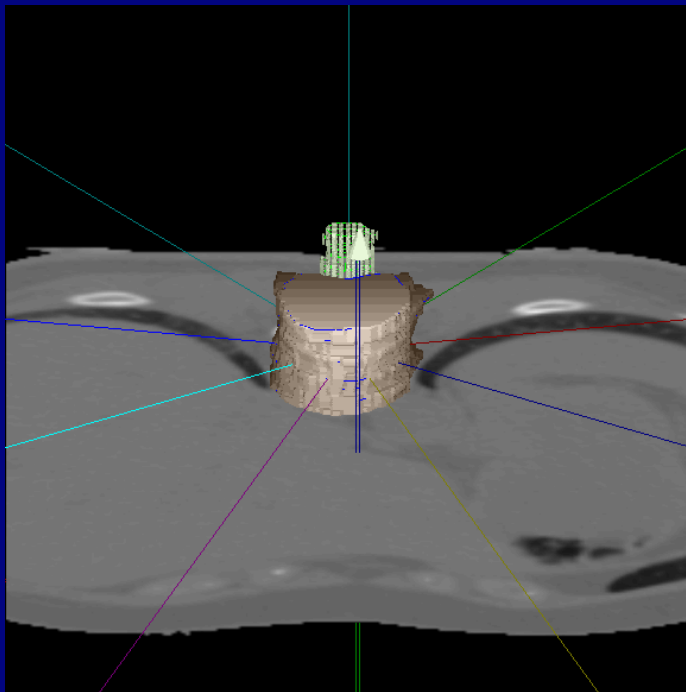
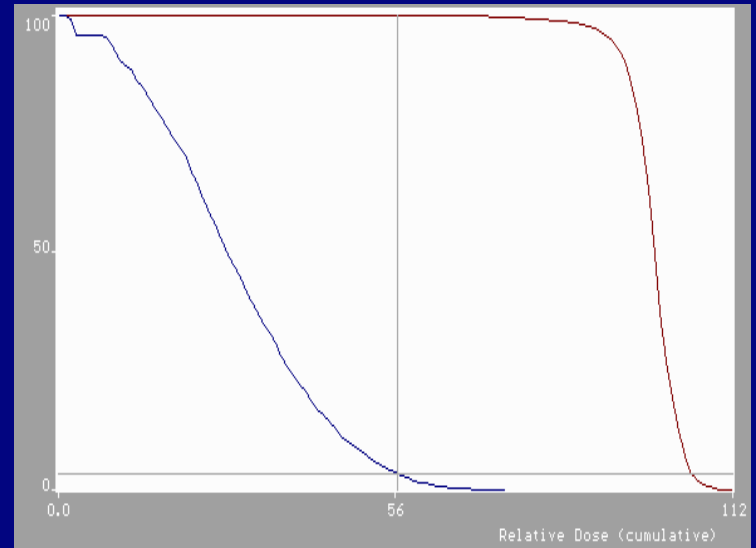
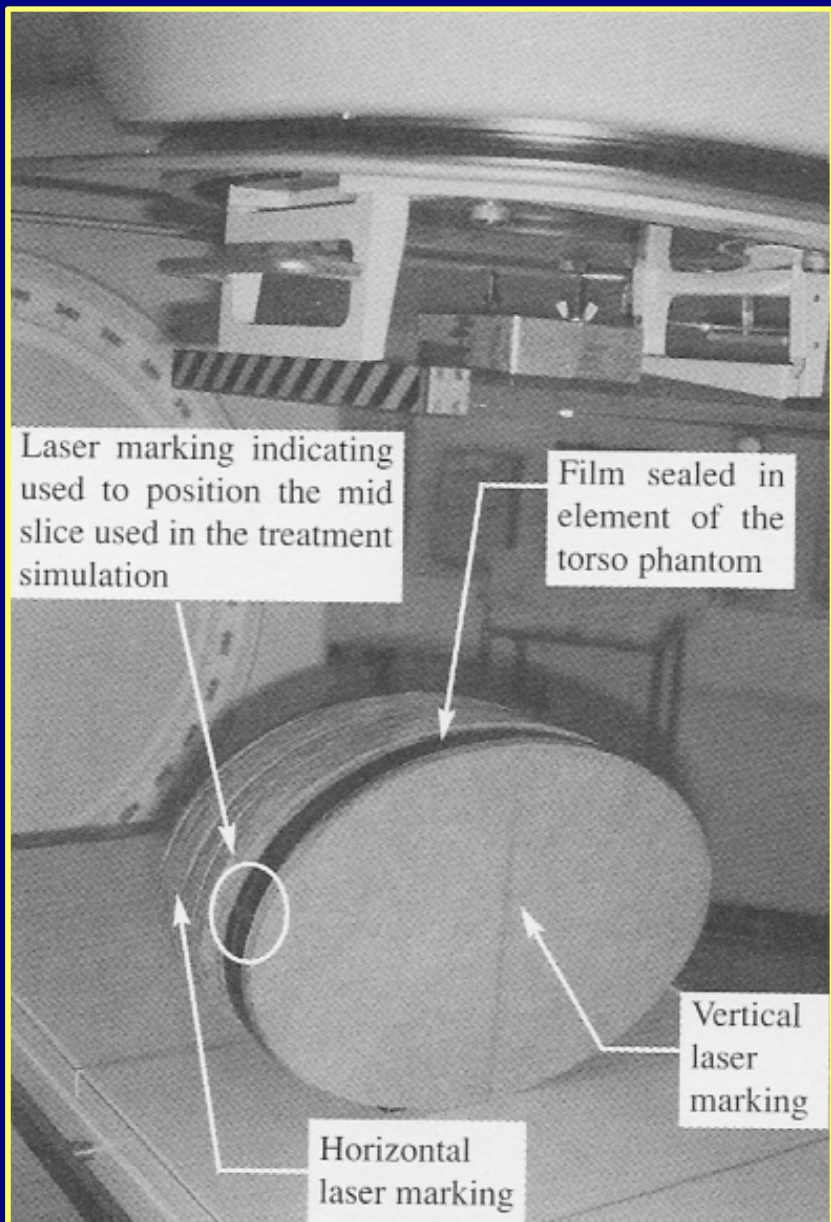


Fig. 2. DVHs for the paraspinal case optimized using the CLS and CORVUS, respectively.

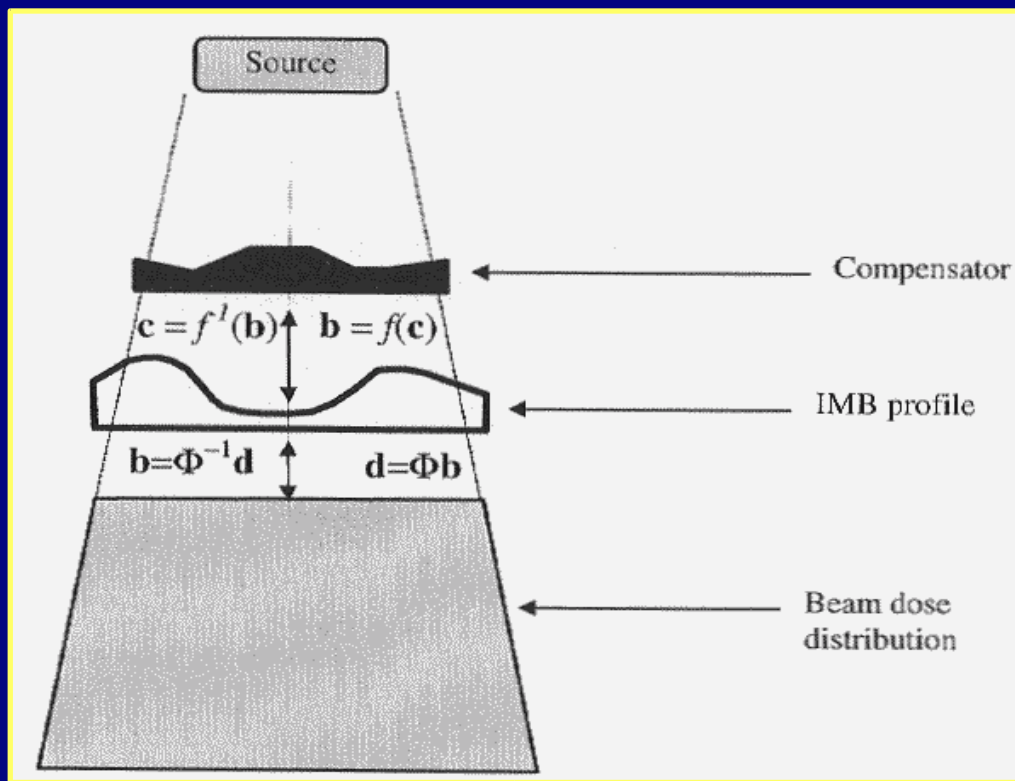


90%

Caso 3: Fantomas en forma de torso con PTV concavo y OAR que penetra en concavidad. Solución y verificación con film y gel (*).



(*) bis-acrylamide-nitrogen + gelatina, RM, T2



Modulación con compensadores

Tomado de: Olivier Haas, "Radiotherapy Treatment Planning. New System Approach". Desarrollado en Walsgrave Hospital, NHS Trust, Coventry.

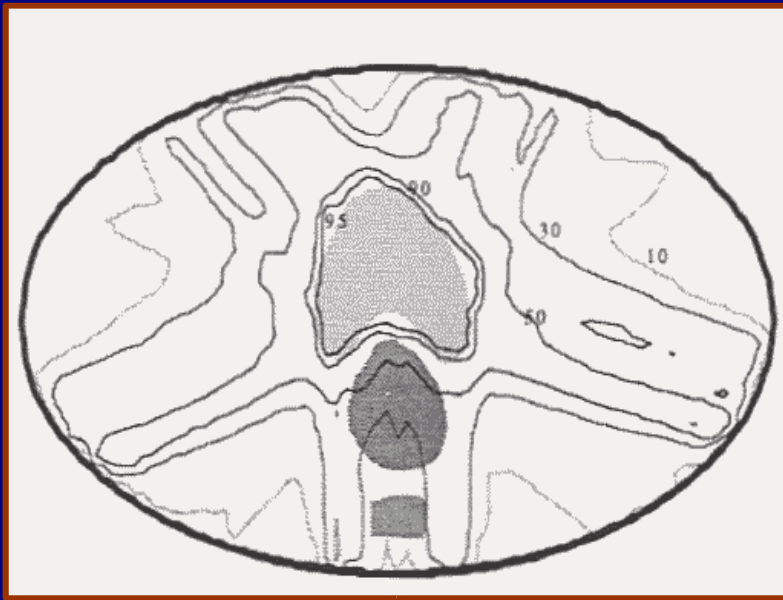
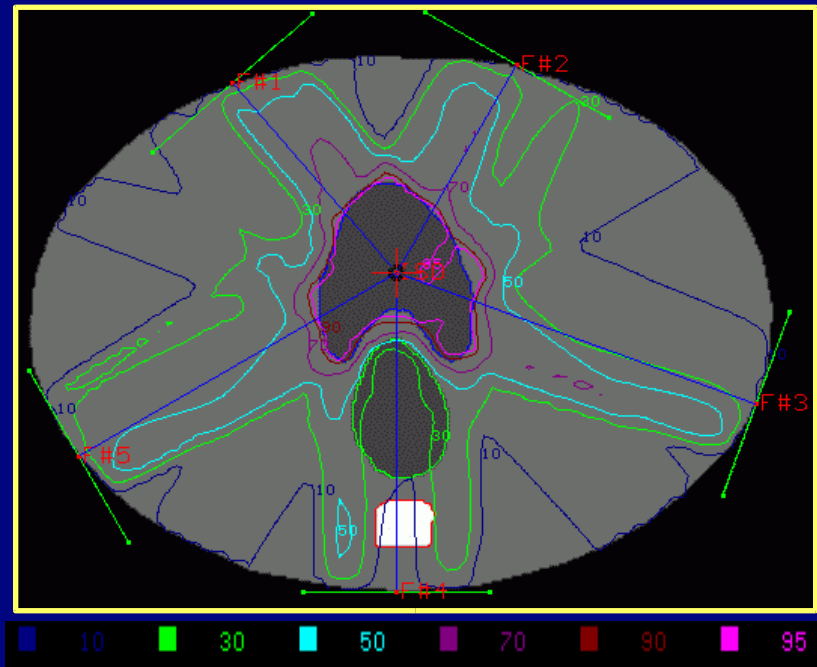
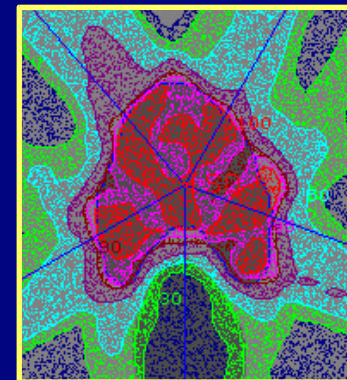
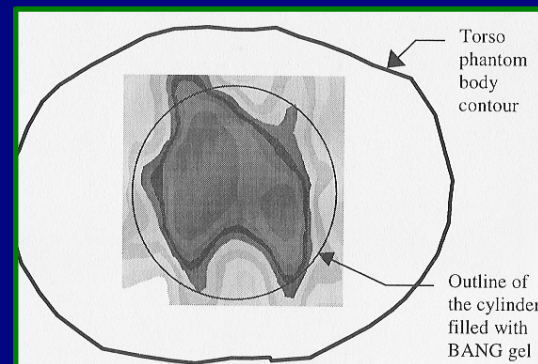
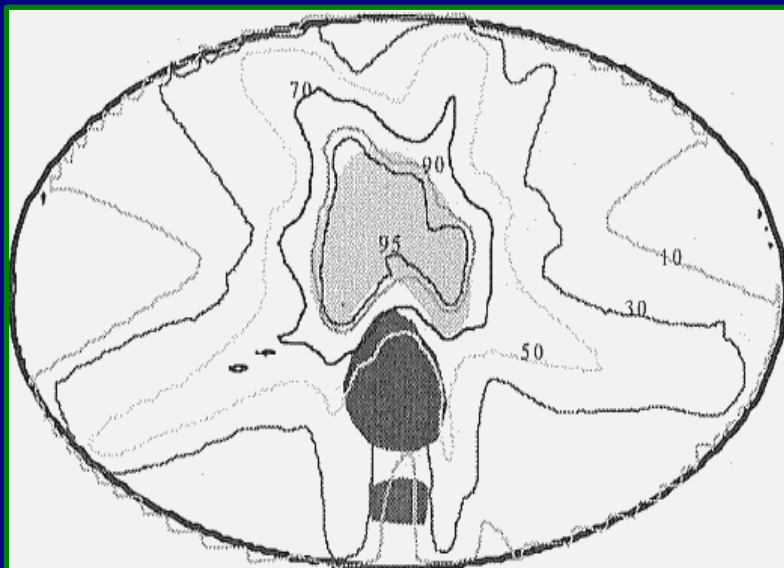


Fig. 6.19 Predicted isodose contour plot, for the hardboard torso phantom, with use being made of a divergent beam and MOGA



Solución con CAT3D, 491 beamlets de 8x8mm, 11500 pto control. $t \sim 65s$, P4 HT 3.0GHz.
PTV=100, ROI1<40, ROI2<30.



Dosimetría con gel (BAND)

Fig. 6.20 Digitised isodose contour plot, for the hardboard torso phantom, obtained from the development of the film used to demonstrate the feasibility of conformal radiotherapy

Otras posibilidades:

- Instrucciones especiales del conjunto de SIMD con gran impacto en algoritmos para TPS.

RSQRTPS , RSQRTSS - cálculo rápido del recíproco de la raíz cuadrada, vectorial y escalar. (Error Rel. < 0.000366). Fundamental en el ray tracing y con precisión suficiente. Unas 15 veces más rápidas que la codificación tradicional de FPU Intel.

CVTTSS2SI, CVTTSS2SI- convertir truncando de flotante a entero, vectorial y escalar, ideal para interpolación en tablas.

- “Dual Core CPU” : procesadores con dos unidades completas en el mismo encapsulado. El primero en el mercado fue el Power4 de IBM, lanzado en 2001. Sólo se beneficiarán con esta tecnología programas implementados para multi-thread, por ejemplo dividiendo el procesamiento del producto matricial en dos mitades, una para cada thread. En el caso de la tecnología HT, el sistema reconoce 2 procesadores “virtuales” y no puede paralelizar muchas tareas. Los “dual-core” pueden paralelizar cualquier tarea desde que el software esté diseñado como multi-thread.

Parece que la tendencia futura no será el aumento de la frecuencia de las CPUs; sino el procesamiento paralelo (multi-core), vectorial (SIMD) y aumento de las memorias internas de alta velocidad (cache L1). Las razones son consideraciones energéticas y por la distancia cada vez mayor entre la velocidad de CPU y la de acceso a memoria RAM.

Conclusiones :

- 1 La planificación inversa aún recorre un camino de “prueba y error” en la búsqueda del plan más apropiado para cada paciente. Diversos parámetros pueden ser modificados para evaluar la influencia en el plan, por eso un ambiente de trabajo interactivo, con respuestas rápidas, es lo ideal.
- 2 La optimización de distribución de intensidad en IMRT continúa siendo de alto costo, en términos de tiempo de procesamiento, para las estaciones de trabajo que las instituciones de salud pueden pagar.
- 3 La investigación en algoritmos y modelos físicos ofrece una vía para disminuir los tiempos de cálculo en una misma estación de trabajo.
- 4 La inclusión de recursos de SIMD en las implementaciones de algoritmos para optimización de IMRT puede disminuir en 2 o 3 veces el tiempo de procesamiento sin aumentar el precio de la CPU.

Muchas Gracias,

Armando Alaminos